

**Técnicas de aproximación
de *throughput* en
redes de Petri estocásticas**

Carlos José Pérez Jiménez

TESIS DOCTORAL

Departamento de Informática e Ingeniería de Sistemas

Universidad de Zaragoza

Director: Dr. D. Javier Campos Laclaustra

Abril 2002

Agradecimientos

Mi agradecimiento en primer lugar a Javier Campos, director de esta tesis, por sus enseñanzas, su constante disponibilidad para todo y por dejarme trabajar a mi manera. Sin su trabajo, esta tesis no se habría podido realizar.

En segundo lugar, me gustaría agradecer a todos los profesores, becarios y personal de administración del Departamento de Informática e Ingeniería de Sistemas la ayuda, consejos y apoyo que me han dado siempre que la he necesitado.

Gracias a mi familia y amigos por su apoyo moral y sus chistes durante los últimos años. Se vive mejor con humor.

Por último, me gustaría agradecer el apoyo financiero de las diversas instituciones y proyectos que han colaborado durante la realización de esta tesis: Beca predoctoral de investigación BIT1192 de la Diputación General de Aragón; proyectos TIC 0354/91, TIC 0242/94 y TAP 0679/98 de la Comisión Interministerial de Ciencia y Tecnología; acción integrada Hispano-Italiana HI 1998-0196 del Ministerio de Educación y Cultura; proyecto ESPRIT BRA 7269 (QMIPS) y contrato HCM CHRX-CT94-0452 (MATCH) de la Comunidad Europea.

Resumen

Los constantes avances tecnológicos conllevan la construcción y utilización de sistemas reales cada vez más complejos. Para el estudio y comprensión del funcionamiento de estos sistemas, se hace necesario el uso de técnicas formales. Las redes de Petri constituyen un formalismo adecuado para atacar estos problemas. Añadiendo una interpretación temporal estocástica, es posible realizar estudios de evaluación de prestaciones de los sistemas modelados. El principal inconveniente de estas técnicas es el conocido como problema de la explosión del espacio de estados. Para reducir el efecto de este problema, se utiliza una estrategia de divide y vencerás.

En esta memoria se ataca el problema de evaluar el número de disparos por unidad de tiempo (*throughput*) en estado estacionario de las transiciones de una red de Petri estocástica. Para ello, se realiza una descomposición del modelo original en varias componentes y, a partir de ellas, se construyen varios sistemas agregados. Con los sistemas agregados, por medio de un algoritmo iterativo de aproximación del tiempo de respuesta, se calcula un valor aproximado del *throughput* de las transiciones del modelo original. Al operar con modelos más pequeños, se pueden estudiar casos más complejos que con las técnicas clásicas a costa de perder precisión en los resultados (cálculo aproximado frente a análisis exacto).

Primero se estudian clases particulares de redes como los grafos marcados, los grafos marcados con pesos o los sistemas deterministas de procesos secuenciales. Por último, se ataca el caso general de las redes de Petri estocásticas. El caso general permite desarrollar una visión estructurada del grafo de alcanzabilidad de cualquier red de Petri a partir de una descomposición estructural suya. Esta visión estructurada del grafo de alcanzabilidad puede utilizarse también para otro tipo de estudios, en particular la generación o almacenamiento eficiente del grafo de alcanzabilidad del modelo original.

Índice General

1	Introducción	1
2	Conceptos básicos y notaciones	9
2.1	Elementos de análisis y álgebra	9
2.2	Redes de Petri	11
2.2.1	Estructura de una red lugar/transición	11
2.2.2	Marcado	13
2.2.3	Propiedades básicas de redes de Petri	15
2.2.4	Técnicas de análisis	16
2.2.5	Lugares implícitos	20
2.3	Redes de Petri estocásticas	21
2.4	Subclases de redes de Petri	30
2.4.1	Grafos marcados	30
2.4.2	Máquinas de estados	32
2.4.3	Grafos marcados con pesos	33
2.4.4	Sistemas deterministas de procesos secuenciales	34
3	Aproximación de throughput en grafos marcados	37
3.1	Literatura previa	38
3.2	Grafos marcados y lugares implícitos	41
3.3	Descomposición estructural de grafos marcados	43
3.4	Aproximación iterativa del throughput	67
3.5	Conclusiones	81
4	Aproximación de throughput en grafos marcados con pesos	83
4.1	Introducción	84
4.2	Ganancia, marcado ponderado y resistencia	86
4.3	Descomposición estructural de WTS's	106

4.4	Aproximación iterativa del throughput	143
4.5	Conclusiones	155
5	Aproximación de throughput en DSSP's	157
5.1	Introducción	158
5.2	Reducción parcial de SM's	161
5.3	Reducción total de SM's	172
5.4	Descomposición estructural de DSSP's	183
5.5	Aproximación iterativa del throughput	187
5.6	Conclusiones	192
6	Descomposición del grafo de alcanzabilidad para el análisis numérico de redes de Petri estocásticas	195
6.1	Introducción	196
6.2	Descomposición estructural de redes de Petri	199
6.2.1	Visión estructurada de redes de Petri	199
6.2.2	Regla de reducción y vistas abstractas	200
6.3	Garantizando la ergodicidad de los sistemas agregados	205
6.4	Visión estructurada del grafo de alcanzabilidad	209
6.5	Eliminación de marcados espurios	231
6.6	Reducción de secuencias de disparo espurias	236
6.7	Aplicación a la aproximación de throughput	244
6.8	Conclusiones	250
7	Comparación numérica de técnicas	253
7.1	Grafos marcados	254
7.2	Grafos marcados con pesos	269
7.3	DSSP's	295
7.4	Redes generales	330
7.5	Conclusiones	341
8	Conclusiones	343

Índice de Figuras

2.1	Seis transformaciones que preservan vivacidad y k-limitación.	18
2.2	Grafo marcado (MG).	31
2.3	Máquina de estados (SM).	32
2.4	Grafo marcado con pesos (WTS).	34
2.5	Sistema determinista de procesos secuenciales (DSSP).	35
3.1	Corte en un grafo marcado.	45
3.2	\mathcal{ES} del MG de la figura 3.1	63
3.3	(a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del MG de la figura 3.1.	65
4.1	Camino de ganancia 1/3.	87
4.2	Camino de marcado ponderado 3.	88
4.3	(a) Un WTS y sus (b) \mathcal{LS}_1 , (c) \mathcal{LS}_2 y (d) \mathcal{BS} .	93
4.4	Camino de resistencia 10.	94
4.5	Resistencia entre transiciones.	96
4.6	Contraejemplo para el recíproco del teorema 4.21.	106
5.1	Cambio de estructura de un conflicto.	162
5.2	Reducción total de una SM.	182
6.1	Un SAM.	200
6.2	(a) \mathcal{ES} y (b) \mathcal{LS}_1 del SAM de la figura 6.1.	206
6.3	(a) Un SAM y sus (b) \mathcal{LS}_1 y (c) \mathcal{BS} .	209
6.4	Producto de grafos dirigidos etiquetados.	214
6.5	Un SAM.	238
6.6	RG del SAM de la figura 6.5.	239
7.1	Ejemplo 1 de MG's.	256
7.2	\mathcal{ES} del MG de la figura 7.1	260

7.3	(a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del MG de la figura 7.1.	261
7.4	Ejemplo 2 de MG's.	263
7.5	\mathcal{ES} del MG de la figura 7.4	266
7.6	(a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del MG de la figura 7.4.	267
7.7	Ejemplo de WTS's.	270
7.8	\mathcal{ES} del WTS de la figura 7.7 (técnica de WTS's).	284
7.9	(a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del WTS de la figura 7.7 (técnica de WTS's).	285
7.10	\mathcal{ES} del WTS de la figura 7.7 (técnicas generales).	289
7.11	(a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del WTS de la figura 7.7 (técnicas generales).	290
7.12	Sistema de fabricación.	296
7.13	DSSP que modela el sistema de la figura 7.12.	297
7.14	(a) Una SM y (b) su reducción parcial.	300
7.15	\mathcal{ES} del DSSP de la figura 7.13 (reducción parcial).	302
7.16	(a) \mathcal{LS}_1 y (b) \mathcal{LS}_2 de la figura 7.13 (reducción parcial).	303
7.17	(a) \mathcal{LS}_3 y (b) \mathcal{BS} de la figura 7.13. (reducción parcial).	304
7.18	(a) Una SM en un DSSP y (b) su reducción total.	308
7.19	(a) \mathcal{LS}_1 y (b) \mathcal{LS}_2 de la figura 7.13. (reducción total).	310
7.20	(a) \mathcal{LS}_3 y (b) \mathcal{BS} de la figura 7.13. (reducción total).	311
7.21	\mathcal{ES} del DSSP de la figura 7.13 (técnicas generales).	314
7.22	(a) \mathcal{LS}_1 y (b) \mathcal{LS}_2 de la figura 7.13. (técnicas generales).	315
7.23	(a) \mathcal{LS}_3 y (b) \mathcal{BS} de la figura 7.13. (técnicas generales).	316
7.24	DSSP con SM's totalmente reducibles.	318
7.25	(a) Reducción total de SM's de la figura 7.24 y (b) \mathcal{ES} del WTS de (a) (técnica de WTS's).	320
7.26	(a) \mathcal{LS}_1 y (b) \mathcal{LS}_2 de la figura 7.24. (reducción total).	321
7.27	\mathcal{BS} de la figura 7.24 (reducción total).	322
7.28	\mathcal{ES} del DSSP de la figura 7.24 (técnicas generales).	324
7.29	(a) \mathcal{LS}_1 y (b) \mathcal{LS}_2 de la figura 7.24. (técnicas generales).	326
7.30	\mathcal{BS} de la figura 7.24 (técnicas generales).	327
7.31	Ejemplo 1 de SAM's.	330
7.32	\mathcal{ES} del SAM de la figura 7.31.	333
7.33	(a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{LS}_3 del SAM de la figura 7.31.	334
7.34	\mathcal{BS} del SAM de la figura 7.31.	335
7.35	Ejemplo 2 de SAM's.	336
7.36	\mathcal{ES} del SAM de la figura 7.35.	337
7.37	(a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del SAM de la figura 7.35.	339

Índice de Tablas

7.1	Aproximación de \mathcal{X} para Fig. 7.1. Tasas 1.0	260
7.2	Aproximación de \mathcal{X} para Fig. 7.1. Tasas variables.	262
7.3	Aproximación de \mathcal{X} para Fig. 7.4. Tasas 1.0	268
7.4	Aproximación de \mathcal{X} para Fig. 7.4. Tasas 1 en \mathcal{N}_1 y 2 en \mathcal{N}_2	268
7.5	Aproximación de \mathcal{X} para Fig. 7.4. Tasas 2 en T_{11} , T_{12} , T_{21} , T_{22} , I_{11} , I_{12} , I_{21} e I_{22} . Resto tasas 1.	269
7.6	Aproximación de \mathcal{X} para Fig. 7.7. Tasas 1.0 (técnica WTS's).	286
7.7	Aproximación de \mathcal{X} para Fig. 7.7. Tasas 10 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 (técnica de WTS's).	287
7.8	Aproximación de \mathcal{X} para Fig. 7.7. Tasas variables (técnica de WTS's).	287
7.9	Aproximación de \mathcal{X} para Fig. 7.7. Tasas 1.0 (técnica general 1).	291
7.10	Aproximación de \mathcal{X} para Fig. 7.7. Tasas 1.0 (técnicas gene- rales 2 y 3).	292
7.11	Aproximación de \mathcal{X} para Fig. 7.7. Tasas 10 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 (técnica general 1).	292
7.12	Aproximación de \mathcal{X} para Fig. 7.7. Tasas 10 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 (técnicas generales 2 y 3).	292
7.13	Aproximación de \mathcal{X} para Fig. 7.7. Tasas variables (técnica general 1).	293
7.14	Aproximación de \mathcal{X} para Fig. 7.7. Tasas variables (técnicas generales 2 y 3).	293
7.15	Comparación de técnicas para el WTS de la Fig. 7.7.	294
7.16	Tasas de las transiciones del DSSP de la Fig. 7.13.	298
7.17	Aproximación de \mathcal{X} para Fig. 7.13 ($n = 17$, reducción parcial).	302
7.18	Aproximaciones para diversos valores de n (reducción parcial).	305
7.19	Aproximación de \mathcal{X} para Fig. 7.13 ($n = 17$, reducción total).	309
7.20	Aproximaciones para diversos valores de n (reducción total).	312

7.21	Aproximación de \mathcal{X} para Fig. 7.13 ($n = 17$, técnicas generales).	314
7.22	Aproximaciones para diversos valores de n (técnicas generales).	317
7.23	Tasas de las transiciones del DSSP de la Fig. 7.24.	319
7.24	Aproximación de \mathcal{X} para Fig. 7.24. Tasas 1.0.	322
7.25	Aproximación de \mathcal{X} para Fig. 7.24. Tasas 100.0 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2	323
7.26	Aproximación de \mathcal{X} para Fig. 7.24. Tasas de la tabla 7.23. . .	323
7.27	Aproximación de \mathcal{X} para Fig. 7.24. Tasas 1.0 (técnicas gene- rales 1 y 2).	325
7.28	Aproximación de \mathcal{X} para Fig. 7.24. Tasas 1.0 (técnica general 3).	327
7.29	Aproximación de \mathcal{X} para Fig. 7.24. Tasas 100 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 (técnicas generales 1 y 2).	328
7.30	Aproximación de \mathcal{X} para Fig. 7.24. Tasas 100 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 (técnica general 3).	328
7.31	Aproximación de \mathcal{X} para Fig. 7.24. Tasas de la tabla 7.23 (técnicas generales 1 y 2).	329
7.32	Aproximación de \mathcal{X} para Fig. 7.24. Tasas de la tabla 7.23 (técnica general 3).	329
7.33	Tasas de las transiciones del SAM de la Fig. 7.31.	331
7.34	Aproximación de \mathcal{X} para Fig. 7.31 (técnicas generales).	335
7.35	Aproximación de \mathcal{X} para Fig. 7.35 (técnica general 1).	338
7.36	Aproximación de \mathcal{X} para Fig. 7.35 (técnica general 2).	338
7.37	Aproximación de \mathcal{X} para Fig. 7.35 (técnica general 3).	340

Capítulo 1

Introducción

Los constantes avances tecnológicos conllevan la construcción y utilización de sistemas reales cada vez más complejos. Estos sistemas se pueden observar en campos tan diferentes como los sistemas de fabricación, sistemas informáticos, sistemas de telecomunicación, etc. Esta creciente complejidad de los sistemas dificulta todas las etapas de su ciclo de vida, desde el diseño, pasando por el mantenimiento, modificación, actualización, mejora de su rendimiento, etc. Por lo tanto se hace necesario el empleo de técnicas formales que ayuden al analista a estudiar el funcionamiento de estos sistemas. Desde el punto de vista teórico todos estos sistemas pueden verse como proveedores de servicios compuestos internamente por componentes o subsistemas más simples que evolucionan en el tiempo de forma concurrente y que tienen que compartir una serie de recursos. El uso de estos recursos provoca la aparición de fenómenos de competencia y cooperación entre las distintas componentes. Además, la evolución de las componentes del sistema debe sincronizarse para conseguir prestar los servicios globales del sistema. Si el sistema no ha sido construido todavía entonces es necesario tener un modelo del mismo y poder estudiar a partir de él sus principales propiedades. El modelado de un sistema depende del tipo de estudio que quiera realizarse, ya que un modelo es una simplificación del sistema real en el que se eliminan los aspectos irrelevantes del mismo. El estudio de las propiedades de un sistema a partir de un modelo del mismo es una tarea que puede ser automatizada si se disponen de las técnicas adecuadas.

El analista de sistemas necesita estudiar básicamente dos tipos de propiedades, las propiedades cualitativas y las cuantitativas. Las propiedades cualitativas del sistema tienen que ver con un funcionamiento correcto desde

el punto de vista lógico. Propiedades de este tipo son por ejemplo la ausencia de bloqueos, que todas sus componentes puedan evolucionar permanentemente, que no se desborde ninguna componente, etc. Las propiedades cuantitativas (evaluación de prestaciones) del sistema tienen que ver con el funcionamiento temporal del sistema, por ejemplo la velocidad a la que el sistema puede operar, su fiabilidad, etc. Este trabajo se centra en la evaluación de prestaciones de los modelos por lo que se asumirá que ya tienen un comportamiento cualitativo correcto.

En [Ho89] se distingue entre dos grandes grupos de sistemas, los sistemas dinámicos de eventos discretos (DEDS), y los sistemas dinámicos de variable continua (CVDS). En los CVDS el estado del sistema cambia continuamente en el tiempo y estos cambios dependen de entradas o variables continuas. La evolución de los CVDS se describe tradicionalmente por medio de ecuaciones diferenciales. En cambio, los DEDS evolucionan en base a la ocurrencia de un determinado evento. De esta forma, un DEDS permanece en el mismo estado hasta que un evento le hace cambiar a otro estado. Para los DEDS no hay un formalismo único universalmente aceptado como las ecuaciones diferenciales para los CVDS. Entre los formalismos empleados para el estudio de los DEDS se tienen las redes de Petri (RdP) [Pet66], redes de colas [Kan92], redes de autómatas estocásticas [PS00], álgebras de procesos estocásticas [HHM95], etc.

Las RdP fueron diseñadas originalmente para realizar análisis cualitativo principalmente. Estas redes pueden extenderse posteriormente con distintas interpretaciones [ST96] dependiendo del objeto de estudio. Por ejemplo, si el propósito es realizar evaluación de prestaciones se les asocia a cada transición un tiempo de disparo estocástico. Para otras aplicaciones se añaden tiempos de disparo deterministas, etc. De esta forma todas las extensiones comparten las técnicas de análisis cualitativo de las RdP. En esta memoria se estudian DEDS modelados con RdP a las que se le añade una interpretación temporal estocástica. Además no se estudia el modelado y especificación de sistemas, sino únicamente métodos de análisis partiendo de modelos con funcionamiento cualitativo correcto.

En cuanto a la extensión estocástica empleada, se asocian tiempos de evolución de las redes distribuidos exponencialmente obteniéndose las redes de Petri estocásticas (SPN) o las redes de Petri estocásticas generalizadas (GSPN). Con las SPN's y GSPN's se obtiene a bajo nivel procesos estocásticos que son cadenas de Markov en tiempo continuo (CTMC). La ventaja de utilizar este tipo de temporización reside en la gran variedad de

técnicas de análisis y herramientas existentes para las CTMC's [Chi87]. Para cualquier CTMC es posible calcular su comportamiento transitorio (en tiempo finito) o, bajo ciertas condiciones (condiciones de ergodicidad), su comportamiento en estado estacionario (comportamiento límite cuando el tiempo tiende a infinito). En esta memoria se analizarán comportamientos en estado estacionario de los modelos, asumiendo que estos comportamientos existen (ergodicidad). De esta forma el análisis numérico de SPN's y GSPN's se realiza en cuatro etapas; generación del grafo de alcanzabilidad de la SPN o GSPN, generación de la CTMC asociada, cálculo del comportamiento estacionario π de la CTMC y cálculo de los índices de prestaciones deseados a partir de π . En esta memoria el índice de prestaciones que se analiza es el número de disparos por unidad de tiempo de las transiciones de la red inicial, es decir, el *throughput*¹ de las transiciones. El problema de este tipo de análisis numérico consiste en que es necesario calcular todos los estados alcanzables de la red de Petri, y este número de estados crece, en general, exponencialmente con el tamaño de la red. Este hecho es conocido como el problema de la explosión del espacio de estados.

Una forma de reducir el efecto del problema de la explosión del espacio de estados es usar una estrategia de *divide y vencerás*. Para ello es necesario disponer de un método de descomposición del modelo original en componentes de menor complejidad y de una técnica de composición de la solución del modelo original a partir de las soluciones de las componentes. Las técnicas desarrolladas en esta memoria se basan en una descomposición del modelo inicial.

Existen diversas técnicas en la literatura para el cálculo de índices de prestaciones basadas en técnicas de descomposición. Estas técnicas pueden clasificarse con respecto a distintos criterios (propuestos en [SC98]).

- Calidad de los resultados obtenidos.
- Información del entorno en las componentes.
- Existencia de una visión abstracta de alto nivel del modelo.
- Partición del modelo por lugares o transiciones.

¹Se mantiene el término inglés *throughput* por su difícil traducción corta al español, porque es ampliamente conocido por los expertos en el tema y porque se utiliza con frecuencia a lo largo de la memoria.

El primer criterio de clasificación (calidad de los resultados obtenidos) puede emplearse para todas las técnicas de análisis existentes, en particular para técnicas de descomposición. En base a este criterio se tienen técnicas exactas, aproximaciones y cotas. Las técnicas de descomposición exactas consiguen la solución del modelo original a partir de las soluciones de las componentes. Normalmente son las técnicas de mayor complejidad computacional tanto en espacio como en tiempo, pero obviamente las que mejores resultados consiguen en términos de exactitud. Un ejemplo de este tipo de técnicas se puede ver en [Don94] donde se utiliza una descomposición del modelo (normalmente subredes generadas por p-semiflujos) definida por el analizador o por construcción del modelo para el cálculo exacto de la distribución en estado estacionario del modelo original a partir de la información obtenida de las componentes. Las técnicas de aproximación no consiguen en general la solución exacta del modelo original sino únicamente una aproximación. La ventaja de este tipo de técnicas reside en la posibilidad de reducir la complejidad computacional de los algoritmos de cálculo. Ejemplos de este tipo de técnicas pueden verse en [CCJS94, LW95, PJC99a]. Las técnicas de cálculo de cotas son las más eficientes, y también las que más se alejan en general de la solución del modelo. En algunos casos el cálculo de cotas es suficiente. Por ejemplo, en las primeras etapas de diseño de un sistema, el cálculo de cotas permite eliminar las alternativas que no permitan obtener las prestaciones deseadas del sistema. En [CCS91] se utiliza una descomposición del modelo original en p-semiflujos para el cálculo de cotas de throughput de las transiciones del modelo, asociando distribuciones de probabilidad arbitrarias al tiempo de disparo de las transiciones. Atendiendo a este criterio de clasificación, las técnicas desarrolladas en esta memoria son técnicas de aproximación.

Respecto al segundo criterio de clasificación (información del entorno contenida en las componentes) hay técnicas en las que las componentes son directamente las subredes generadas por la descomposición del modelo, es decir, ninguna componente tiene información adicional del resto del modelo. Ejemplos de este tipo de técnicas son: el análisis exacto de GSPN superpuestas (una subclase de GSPN) en [Don94], el análisis aproximado de SPN por medio de agregación de flujo equivalente en [JD91], o el cálculo de cotas basado en técnicas de programación lineal en [CCS91]. En otros casos, las subredes se complementan con objeto de resumir el comportamiento del resto del modelo. Ejemplos de este tipo de técnicas son [CDS99] donde se emplea una visión estructurada del modelo original para generar las compo-

mentes y después realizar un análisis exacto, o [PJC98] que emplea la misma visión pero para análisis aproximado. Otros ejemplos son [CS93] para la mejora de cotas o [LW95] para análisis aproximado. En las técnicas de esta memoria, todos los sistemas agregados contienen información del entorno en las componentes.

Para poder emplear una técnica de descomposición se debe disponer también de una técnica de composición de la solución del modelo original a partir de las soluciones de las componentes. Por lo tanto, otro criterio de clasificación puede ser la existencia o no de una visión abstracta de alto nivel del modelo en la fase de integración. Ejemplos de técnicas sin esta visión abstracta de alto nivel son: el cálculo de cotas presentado en [CCS91] y su mejora en [CS93], o la técnica de aproximación presentada en [LW95]. Ejemplos de técnicas con la visión abstracta de alto nivel son: el análisis aproximado de SPN por medio de agregación de flujo equivalente en [JD91], la técnica de aproximación basada en forma producto en [BD96], el análisis exacto de SPN con visión estructurada de [CDS99], y las técnicas de análisis aproximado para WTS [PJCS96b], DSSP [PJCS96c, PJCS96a], o SPN [PJC98, PJC99a]. Todas las técnicas de esta memoria emplean una visión abstracta de alto nivel del sistema inicial.

Como la estructura de una red de Petri tiene dos tipos de nodos (lugares y transiciones), otro criterio de clasificación de técnicas de descomposición puede ser la selección de lugares o transiciones para realizar la descomposición del modelo. Existen técnicas en la literatura que descomponen el modelo por lugares o por transiciones. Las técnicas que se presentan en esta memoria descomponen las redes por medio de lugares.

El trabajo desarrollado en esta memoria tiene su punto de partida en [CCJS94] donde se desarrolla una técnica de aproximación del throughput de las transiciones de un grafo marcado. La técnica se basa en una descomposición estructural del modelo inicial en varias subredes, a partir de las cuales se construyen varios sistemas agregados. Con estos sistemas agregados, por medio de un algoritmo numérico iterativo de aproximación del tiempo de respuesta de las subredes reducidas, se aproxima el throughput de las transiciones del grafo marcado inicial. El objetivo consistía en extender la técnica a clases de redes cada vez más generales. Siguiendo con técnicas de descomposición estructural del modelo original en varias subredes se consiguió extender la técnica a grafos marcados con pesos [PJCS96b] y a sistemas deterministas de procesos secuenciales [PJCS96c, PJCS96a]. Para descomponer RdP generales, se ha adaptado la técnica aparecida en

[CDS99]. En este trabajo, por medio de una descomposición estructural del modelo inicial en varias componentes es posible resolver la CTMC de la SPN inicial a partir de las CTMC's de sus componentes. De esta manera se reduce drásticamente la complejidad en espacio del problema. En [PJC98, PJC99a] se adaptan los sistemas agregados de [CDS99] para poder realizar cálculos aproximados. Recientemente, en [FZ01] ha aparecido una técnica de aproximación del throughput de las transiciones de una SPN fuertemente inspirada en este tipo de técnicas. Allí se incluyen ligeras modificaciones en el algoritmo numérico de aproximación. En [FZ98] se extiende el algoritmo de aproximación del tiempo de respuesta a una subclase de redes de Petri coloreadas para obtener aproximaciones de throughput.

La memoria está estructurada en los siguientes 7 capítulos:

En el capítulo 2 se introducen formalmente los conceptos básicos, notaciones y técnicas utilizadas a lo largo de la memoria. Se define el formalismo de las redes de Petri y se exponen las principales técnicas de análisis empleadas en la memoria, así como la interpretación temporal de los modelos y las subclases de redes que se estudian en capítulos posteriores.

En el capítulo 3 se explica en detalle el trabajo desarrollado en [CCJS94] sobre aproximación de throughput en grafos marcados. Este es el trabajo que motiva el desarrollo de esta memoria y se aprovecha la exposición para introducir algunas mejoras técnicas y el análisis de funcionamiento del algoritmo numérico de aproximación.

En el capítulo 4 se extiende la técnica de aproximación de throughput a la subclase de los grafos marcados con pesos (WTS). Esta técnica, presentada en [PJCS96b], se basa igualmente en una descomposición estructural del WTS original, la generación de varios WTS's más pequeños y la aproximación del throughput de las transiciones del WTS original por medio de un algoritmo iterativo que opera sobre los WTS's agregados.

En el capítulo 5 se desarrollan dos técnicas de reducción de máquinas de estados que permiten extender las técnicas anteriores a la subclase de los sistemas deterministas de procesos secuenciales (DSSP). Estas técnicas se presentaron en [PJCS95, PJCS96c, PJCS96a] y también se basan en una descomposición estructural del modelo original.

En el capítulo 6 se aborda la generalización a cualquier red de Petri del problema de descomposición y generación de los sistemas agregados. Son las técnicas presentadas en [PJC98, PJC99a, PJC99b]. Para algunas subclases de redes es posible realizar la descomposición del modelo inicial y la generación de los sistemas agregados por medio de un análisis basado en

la estructura y marcado inicial del modelo. Para realizar la misma tarea en sistemas generales es necesario operar a nivel de su grafo de alcanzabilidad. En este capítulo se estudia cómo una descomposición estructural del modelo inicial induce una estructura en su grafo de alcanzabilidad que permite su almacenamiento de forma descompuesta y sin tener que generar todos los estados alcanzables. Esta descripción del grafo de alcanzabilidad se puede emplear para propósitos más generales que el cálculo aproximado del throughput de las transiciones del modelo inicial. De esta forma, se expone un algoritmo de generación del grafo de alcanzabilidad del sistema original más eficiente que el clásico, tanto en memoria como en tiempo. Posteriormente, la descripción descompuesta del grafo de alcanzabilidad se utiliza para generar directamente los sistemas agregados. Se exponen tres técnicas distintas de generación de los sistemas agregados dependiendo del tipo de propiedades que se deseen preservar en los mismos. Por último, las tres técnicas de generación de sistemas agregados se aplican al cálculo aproximado del throughput de las transiciones de la red inicial. Para ello es necesario adaptar el algoritmo numérico a la nueva situación.

En el capítulo 7 se desarrolla una batería de ejemplos con todas las técnicas desarrolladas, comparando en algunos de ellos todas las técnicas de descomposición que les sean aplicables.

Por último, en el capítulo 8 se exponen las conclusiones de la memoria y el trabajo que habría que realizar en el futuro.

Capítulo 2

Conceptos básicos y notaciones

En este capítulo se introducen los conceptos básicos, notaciones y técnicas utilizadas a lo largo de la memoria. La distribución del capítulo es la siguiente. En la sección 2.1 se exponen los principales resultados de álgebra y análisis empleados en la memoria, en la sección 2.2 se introducen las notaciones, definiciones, principales propiedades y técnicas de análisis de las redes lugar/transición autónomas. En la sección 2.3 se introduce la interpretación temporal que se le va a dar a las redes a lo largo de esta memoria. Por último, en la sección 2.4 se definen y exponen las principales propiedades de las distintas subclases de redes que aparecen en la memoria.

2.1 Elementos de análisis y álgebra

En esta sección se indicarán las notaciones, definiciones y resultados elementales de análisis y álgebra que aparecen en la memoria. Se asume que el lector tiene conocimientos elementales de estas materias (ver [Kla86, Gre76] para una introducción).

Notación 2.1 *Se denotarán por \mathbb{N} , \mathbb{Z} , \mathbb{Q} y \mathbb{R} a los conjuntos de números naturales, enteros, racionales y reales respectivamente. Se entiende que 0 no es un número natural. Además, se denotarán por \mathbb{Z}^+ , \mathbb{Q}^+ y \mathbb{R}^+ a los conjuntos de enteros, racionales y reales no negativos respectivamente.*

Notación 2.2 Sea P un conjunto. Se denota por $|P|$ al cardinal del conjunto (número de elementos).

Teorema 2.3 [Kla86] Axioma del supremo.

Sea $V \subseteq \mathbb{R}$, $V \neq \emptyset$ acotado superiormente. Entonces V posee supremo.

Corolario 2.4 [Kla86] Sea $V \subseteq \mathbb{R}$, $V \neq \emptyset$ acotado inferiormente. Entonces V posee ínfimo.

Teorema 2.5 [Kla86] Algoritmo de la división.

Sean $a, b \in \mathbb{Z}$ con $b > 0$. Entonces existen $q, r \in \mathbb{Z}$ tales que $a = bq + r$ y $0 \leq r < b$.

Notación 2.6 División entera.

Sean $a, b \in \mathbb{Z}$ con $b > 0$. Se denota por $\lfloor \frac{a}{b} \rfloor$ al mayor entero q tal que $q \leq \frac{a}{b}$.

Definición 2.7 [Kla86] Norma infinito.

Sea $\mathbf{v} = (v_1, v_2, \dots, v_n) \in \mathbb{R}^n$ un vector de n componentes reales. Se define la norma infinito de \mathbf{v} como $\|\mathbf{v}\|_\infty = \max_{i=1}^n \{|v_i|\}$.

Teorema 2.8 [Kla86] Teorema de la función inversa.

Sea $f : D \rightarrow \mathbb{R}$ continua e inyectiva en D compacto. Entonces existe $f^{-1} : f(D) \rightarrow D$ y es continua.

Teorema 2.9 [Kla86] Propiedad de Darboux.

Sea $f : [a, b] \rightarrow \mathbb{R}$ continua y $h \in \mathbb{R}$ cumpliendo $f(a) < h < f(b)$ ó $f(b) < h < f(a)$. Entonces existe $x \in (a, b)$ tal que $f(x) = h$.

Teorema 2.10 [Sma74] Teorema del punto fijo de Brouwer.

Sea $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ una función continua en D compacto, convexo y no vacío tal que $f(D) \subseteq D$. Entonces existe $x \in D$ tal que $f(x) = x$.

Notación 2.11 Sea V un K -espacio vectorial (K cuerpo). Se denota por $\dim(V)$ a la dimensión de V .

Notación 2.12 Sea $f : V \rightarrow W$ aplicación lineal entre dos K -espacios vectoriales (K cuerpo). Se utilizan las siguientes notaciones:

i) $\text{Im}f = \{f(v) \mid v \in V\}$. $\text{rank}f = \dim(\text{Im}f)$.

ii) $\text{Ker}f = \{v \in V \mid f(v) = 0\}$.

Teorema 2.13 [Gre76] *Sea V un K -espacio vectorial (K cuerpo) de dimensión n . Entonces V es isomorfo a K^n (denotado por $V \cong K^n$).*

Teorema 2.14 [Gre76] **Primer teorema de isomorfía.**

Sean V, W K -espacios vectoriales (K cuerpo) y $f : V \rightarrow W$ aplicación lineal. Entonces $V/\text{Ker}f \cong \text{Im}f$.

Corolario 2.15 [Gre76] *Sean V, W K -espacios vectoriales (K cuerpo) y $f : V \rightarrow W$ aplicación lineal. Entonces $\dim(V) = \dim(\text{Ker}f) + \dim(\text{Im}f)$.*

2.2 Redes de Petri

En esta memoria se asume que el lector está familiarizado con los conceptos básicos de *redes de Petri* (ver [Pet81, Sil85, Mur89, DHP⁺93] para una introducción). En esta sección se introducen las definiciones, notaciones y principales propiedades que serán de utilidad en los siguientes capítulos.

2.2.1 Estructura de una red lugar/transición

En esta sección se exponen las definiciones, notaciones y principales propiedades de las redes de Petri no temporizadas.

Existen dos definiciones de *redes lugar/transición*, una orientada a grafos y otra a matrices.

Definición 2.16 **P/T red orientada a grafos.**

Una red lugar/transición (P/T red) es una cuádrupla $\mathcal{N} = \langle P, T, F, W \rangle$ donde P y T son conjuntos disjuntos, finitos no vacíos de lugares y transiciones, $F \subseteq (P \times T) \cup (T \times P)$ es el conjunto de arcos dirigidos y $W : F \rightarrow \mathbb{N}$ asigna a cada arco un peso. A los elementos del conjunto $P \cup T$ se les denomina nodos.

Definición 2.17 **P/T red orientada a matrices.**

Una P/T red es una cuádrupla $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$ donde P y T son como en la definición 2.16 y $\mathbf{Pre}, \mathbf{Post}$ son las matrices de incidencia anterior e incidencia posterior respectivamente (ambas de enteros no negativos de tamaño $|P| \times |T|$).

Se utilizará una definición u otra dependiendo de las necesidades. Una red de Petri se representa gráficamente como un grafo bipartito orientado.

Los lugares se representan mediante círculos y las transiciones mediante rectángulos o barras. Por cada elemento $\mathbf{Pre}(p, t) \neq 0$ hay un arco dirigido del lugar p a la transición t de peso $\mathbf{Pre}(p, t) = W(p, t)$. Análogamente, por cada elemento $\mathbf{Post}(p, t) \neq 0$ hay un arco dirigido de la transición t al lugar p de peso $\mathbf{Post}(p, t) = W(t, p)$. Un arco sin peso tiene asociado por defecto peso 1. Se puede suponer, sin pérdida de generalidad, que las P/T redes son conexas, es decir, que se puede ir de un nodo a cualquier otro de la red atravesando arcos sin tener en cuenta el sentido de los mismos. En el fondo una P/T red que no es conexa se puede descomponer en tantas redes como componentes conexas tenga y estudiar cada una por separado.

Definición 2.18 Una P/T red \mathcal{N} es ordinaria si y sólo si todos sus arcos tienen peso asociado 1. Se denotará por $\mathcal{N} = \langle P, T, F \rangle$.

Se suele utilizar notaciones con puntos para los conjuntos de incidencia anterior y posterior de un nodo o conjunto de nodos.

Notación 2.19 Sea \mathcal{N} una P/T red y $v \in P \cup T$ un nodo de \mathcal{N} . Se denota por:

- i) $\bullet v = \{u \in P \cup T \mid (u, v) \in F\}$.
- ii) $v \bullet = \{u \in P \cup T \mid (v, u) \in F\}$.

Notación 2.20 Sea \mathcal{N} una P/T red y $V \subseteq P \cup T$ un subconjunto de nodos de \mathcal{N} . Se denota por:

- i) $\bullet V = \{u \in P \cup T \mid \exists v \in V \text{ tal que } (u, v) \in F\}$.
- ii) $V \bullet = \{u \in P \cup T \mid \exists v \in V \text{ tal que } (v, u) \in F\}$.

Definición 2.21 Sea \mathcal{N} una P/T red.

- i) Un lugar $p \in P$ se dice selección si y sólo si $|p \bullet| > 1$.
- ii) Un lugar $p \in P$ se dice atribución si y sólo si $|\bullet p| > 1$.
- iii) Una transición $t \in T$ se dice distribución si y sólo si $|t \bullet| > 1$.
- iv) Una transición $t \in T$ se dice conjunción si y sólo si $|\bullet t| > 1$.

Definición 2.22 Una P/T red \mathcal{N} es pura si ninguna transición contiene un lugar que sea simultáneamente de entrada y salida.

Definición 2.23 Sea \mathcal{N} una P/T red. La matriz de incidencia de \mathcal{N} es $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$. Las filas (columnas) de \mathbf{C} se denominan vectores de incidencia del lugar (transición) correspondiente.

En la matriz de incidencia \mathbf{C} de una P/T red, los elementos positivos se corresponden con los de la matriz \mathbf{Post} de incidencia posterior mientras que los elementos negativos se corresponden con los de la matriz \mathbf{Pre} de incidencia anterior.

Notar que una red pura está completamente caracterizada por su matriz de incidencia, mientras que si la red no es pura, parte de su estructura no aparece en la matriz de incidencia (la correspondiente a los lugares que son simultáneamente de entrada y salida de una misma transición).

Definición 2.24 Sean $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$ y $\mathcal{N}' = \langle P', T', \mathbf{Pre}', \mathbf{Post}' \rangle$ dos P/T redes. \mathcal{N}' se dice subred de \mathcal{N} (denotado por $\mathcal{N}' \subseteq \mathcal{N}$) si:

- i) $P' \subseteq P$ y $T' \subseteq T$.
- ii) $\mathbf{Pre}' = \mathbf{Pre}[P', T']$ y $\mathbf{Post}' = \mathbf{Post}[P', T']$.

Definición 2.25 Sea $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$ una P/T red y $V \subseteq P \cup T$ un subconjunto de nodos. La subred $\mathcal{N}' = \langle P', T', \mathbf{Pre}', \mathbf{Post}' \rangle$ de \mathcal{N} generada por V es una subred tal que:

- i) $P' = V \cap P$ y $T' = V \cap T$.
- ii) $\mathbf{Pre}' = \mathbf{Pre}[P', T']$ y $\mathbf{Post}' = \mathbf{Post}[P', T']$.

Si V contiene sólo un tipo de nodos, se entiende que \mathcal{N}' es la red generada por $V \cup \bullet V \cup V \bullet$.

2.2.2 Marcado

La estructura de una red es estática. Para poder modelar sistemas dinámicos es necesario definir un estado inicial en el modelo y una regla de evolución de este estado. Este es el contenido de esta sección.

Definición 2.26 Marcado y P/T sistema.

Sea \mathcal{N} una P/T red. Un marcado de \mathcal{N} es un vector $\mathbf{m} \in \{\mathbf{Z}^+\}^{|P|}$ que asigna a cada lugar $p \in P$ un entero no negativo. Un sistema lugar/transición (P/T sistema) es un par $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ donde \mathbf{m}_0 es el marcado inicial.

El número $\mathbf{m}[p]$ asociado al lugar $p \in P$ constituye el estado local del lugar. El estado del P/T sistema se compone de los estados individuales de cada lugar. Por lo tanto, el vector \mathbf{m} constituye el estado del sistema modelado por el P/T sistema \mathcal{S} . El estado local del lugar p se representa gráficamente con $\mathbf{m}[p]$ *marcas* (puntos) dentro del círculo que representa al lugar p .

La evolución de un P/T sistema viene definido por la *regla de disparo* de sus transiciones.

Definición 2.27 Sensibilización y disparo.

Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m} \rangle$ un P/T sistema. Una transición $t \in T$ está sensibilizada en \mathbf{m} si y sólo si $\mathbf{m} \geq \mathbf{Pre}[P, t]$. Se define el grado de sensibilización de t en \mathbf{m} como $\mathbf{e}(\mathbf{m})[t] = \max\{k \in \mathbb{Z}^+ \mid \mathbf{m} \geq k \cdot \mathbf{Pre}[P, t]\}$. Una transición t sensibilizada en un marcado \mathbf{m} puede dispararse produciendo un nuevo marcado $\mathbf{m}' = \mathbf{m} + \mathbf{C}[P, t]$. Este disparo se denota por $\mathbf{m} \xrightarrow{t} \mathbf{m}'$.

En un P/T sistema puede haber varias transiciones sensibilizadas simultáneamente y por lo tanto sería posible el disparo simultáneo de varias transiciones. En la evolución de un P/T sistema se supondrá que en un determinado momento sólo se dispara una transición, es decir, que el observador puede distinguir cada disparo de cada transición de forma independiente. Esta interpretación se conoce como *semántica de entrelazado* (“interleaving semantics”).

Definición 2.28 Secuencias de disparo.

Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema. Una secuencia de disparo desde \mathbf{m}_0 es una sucesión finita $\sigma = \{t_i\}_{i=1}^n$ de transiciones tales que $\mathbf{m}_{i-1} \xrightarrow{t_i} \mathbf{m}_i$ para $1 \leq i \leq n$. En ese caso se dice que \mathbf{m}_n es un marcado alcanzable desde \mathbf{m}_0 y se denota por $\mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}_n$.

Definición 2.29 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema y σ una secuencia de disparo desde \mathbf{m}_0 . El vector característico o vector contador de disparos de σ es σ tal que $\sigma[t]$ es el número de veces que la transición t aparece en σ .

Si en un P/T sistema $\mathbf{m} \xrightarrow{t} \mathbf{m}'$, entonces se tiene que $\mathbf{m}' = \mathbf{m} + \mathbf{C}[t]$. Generalizando esta expresión a secuencias de disparo se obtiene la *ecuación de estado* de un P/T sistema.

Teorema 2.30 [Sil85] Ecuación de estado de un P/T sistema.

Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema y σ una secuencia de disparo desde \mathbf{m}_0 de vector característico σ . Si $\mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$ entonces $\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma$.

Definición 2.31 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema. El lenguaje de \mathcal{S} es el conjunto $L(\mathcal{S})$ de secuencias de disparo desde \mathbf{m}_0 .

Definición 2.32 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema. El conjunto de estados alcanzables de \mathcal{S} es $R(\mathcal{S}) = \{\mathbf{m} \mid \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m} \text{ con } \sigma \in L(\mathcal{S})\}$.

Definición 2.33 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema. El grafo de alcanzabilidad $RG(\mathcal{S})$ es un grafo dirigido etiquetado cuyos vértices son los elementos de $R(\mathcal{S})$ y hay una arista dirigida etiquetada con t del vértice \mathbf{m} al vértice \mathbf{m}' si y sólo si $\mathbf{m} \xrightarrow{t} \mathbf{m}'$.

2.2.3 Propiedades básicas de redes de Petri

Definición 2.34 Vivacidad.

Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema. Una transición $t \in T$ es viva si y sólo si para todo $\mathbf{m} \in R(\mathcal{S})$ existe $\sigma \in L(\mathcal{N}, \mathbf{m})$ tal que $t \in \sigma$. \mathcal{S} es vivo si y sólo si todas sus transiciones son vivas.

La propiedad de vivacidad asegura la posible repetición futura de cada acción individual del sistema.

Definición 2.35 Un P/T sistema \mathcal{S} se dice libre de bloqueo si y sólo si para todo $\mathbf{m} \in R(\mathcal{S})$ existe $t \in T$ sensibilizada en \mathbf{m} .

La propiedad de ser libre de bloqueo es claramente necesaria para la vivacidad.

Definición 2.36 Limitación.

Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema. Un lugar $p \in P$ se dice k -limitado si y sólo si $\mathbf{m}[p] \leq k$ para todo $\mathbf{m} \in R(\mathcal{S})$. El P/T sistema es k -limitado si y sólo si todos sus lugares son k -limitados. Un lugar o un P/T sistema son limitados si y sólo si son k -limitados para algún $k \in \mathbb{N}$.

La propiedad de limitación impide desbordamientos debidos al crecimiento ilimitado del contenido de cualquier componente del sistema.

Definición 2.37 Estados recurrentes y reversibilidad.

Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema. Un marcado alcanzable $\mathbf{m} \in R(\mathcal{S})$ se dice estado recurrente (“home state”) si y sólo si es alcanzable desde cualquier marcado de $R(\mathcal{S})$, es decir, si y sólo si para todo $\mathbf{m}' \in R(\mathcal{S})$ se tiene $\mathbf{m}' \xrightarrow{\sigma} \mathbf{m}$ con $\sigma \in L(\mathcal{N}, \mathbf{m}')$. Un P/T sistema \mathcal{S} se dice reversible si y sólo si \mathbf{m}_0 es un estado recurrente.

La existencia de estados recurrentes indica la posibilidad de regresar siempre a ciertos estados del sistema y la reversibilidad indica la posibilidad de regresar siempre a cualquier estado del sistema.

Definición 2.38 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema. Dos transiciones t y t' están en conflicto efectivo en \mathcal{S} si y sólo si existe un marcado alcanzable \mathbf{m} tal que $\mathbf{m} \geq \mathbf{Pre}[P, t]$ y $\mathbf{m} \geq \mathbf{Pre}[P, t']$ pero no se verifica que $\mathbf{m} \geq \mathbf{Pre}[P, t] + \mathbf{Pre}[P, t']$.

Dos transiciones están en conflicto efectivo si están sensibilizadas en un marcado alcanzable y el disparo de una desensibiliza la otra.

Definición 2.39 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema. Dos transiciones t y t' son concurrentes en \mathcal{S} si y sólo si existe un marcado alcanzable \mathbf{m} tal que $\mathbf{m} \geq \mathbf{Pre}[P, t] + \mathbf{Pre}[P, t']$.

Dos transiciones son concurrentes si el disparo de una no desensibiliza a la otra.

Definición 2.40 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m} \rangle$ un P/T sistema. Un paso es un multiconjunto de transiciones que podrían dispararse simultáneamente en \mathcal{S} . Puede representarse por un vector \mathbf{s} de $|T|$ enteros no negativos de forma que $\mathbf{s}[t]$ es el número de veces que la transición t aparece en \mathbf{s} . En este caso, el paso \mathbf{s} está sensibilizado en \mathbf{m} si y sólo si $\mathbf{m} \geq \mathbf{Pre} \cdot \mathbf{s}$. El disparo de \mathbf{s} se denota por $\mathbf{m} \xrightarrow{\mathbf{s}} \mathbf{m}'$ o por $\mathbf{m} \xrightarrow{\sigma} \mathbf{m}'$ donde σ es cualquier secuencialización de \mathbf{s} .

Definición 2.41 Sea \mathcal{N} una P/T red. Un camino en \mathcal{N} es una sucesión $\{x_i\}_{i=1}^n$ de nodos de \mathcal{N} tales que $(x_i, x_{i+1}) \in F$ para $1 \leq i < n$. Un camino se dice ciclo si $(x_n, x_1) \in F$. Un camino (ciclo) se dice simple si todos sus elementos son diferentes.

Definición 2.42 Sea \mathcal{N} una P/T red. \mathcal{N} es fuertemente conexa si y sólo si para cualesquiera $x, y \in P \cup T$ existe un camino en \mathcal{N} que conecta x con y .

2.2.4 Técnicas de análisis

En esta sección se van a exponer las técnicas de análisis utilizadas a lo largo de la memoria. Las técnicas para el análisis de los P/T sistemas (*análisis cualitativo* de redes) se clasifican normalmente en tres grupos.

En primer lugar se tienen las *técnicas enumerativas*. Se basan en la generación del grafo de alcanzabilidad para sistemas limitados o del *grafo de cobertura* para sistemas no limitados [Fin93]. Estas técnicas se pueden aplicar en teoría, pero en la práctica están limitadas a sistemas “pequeños” debido a su elevada complejidad computacional (altamente exponencial normalmente). Este problema se conoce con el nombre de *problema de la explosión del espacio de estados*.

En segundo lugar se tienen las *técnicas de transformación*. En este grupo de técnicas el objetivo es reducir el tamaño de los modelos mediante de reglas de reducción que preserven las propiedades que se quieren estudiar (ver [Sil85, Ber86] para ejemplos de este tipo de técnicas).

En tercer lugar se tienen las *técnicas estructurales*. En este grupo de técnicas el objetivo es obtener la máxima información del modelo utilizando únicamente su estructura y marcado inicial.

En esta memoria se emplean principalmente técnicas de análisis estructural, pero en alguna ocasión es necesario conocer alguna técnica de transformación. En la figura 2.1 puede observarse un conjunto sencillo de seis reglas de reducción o refinamiento que preservan vivacidad y k -limitación tomadas de [Sil85]. Con este conjunto de reglas es posible reducir la complejidad del cálculo de la vivacidad y limitación de un P/T sistema.

A continuación se expondrán las técnicas de análisis estructural que se utilizarán a lo largo de la memoria.

Definición 2.43 Una P/T red \mathcal{N} es estructuralmente viva si y sólo si existe un marcado \mathbf{m} tal que $\langle \mathcal{N}, \mathbf{m} \rangle$ es vivo.

La vivacidad estructural de una P/T red asegura la existencia de un marcado inicial que produce un P/T sistema vivo. No se conoce una caracterización algebraica general para la vivacidad estructural.

Definición 2.44 Una P/T red \mathcal{N} es estructuralmente limitada si y sólo si para todo marcado \mathbf{m} $\langle \mathcal{N}, \mathbf{m} \rangle$ es limitado.

La limitación estructural de una P/T red asegura la limitación de un P/T sistema independientemente del marcado inicial que se utilice. Existe una caracterización algebraica para la limitación estructural que se verá en el siguiente teorema. Pero antes se va a tratar una cuestión sobre la notación que se va a emplear.

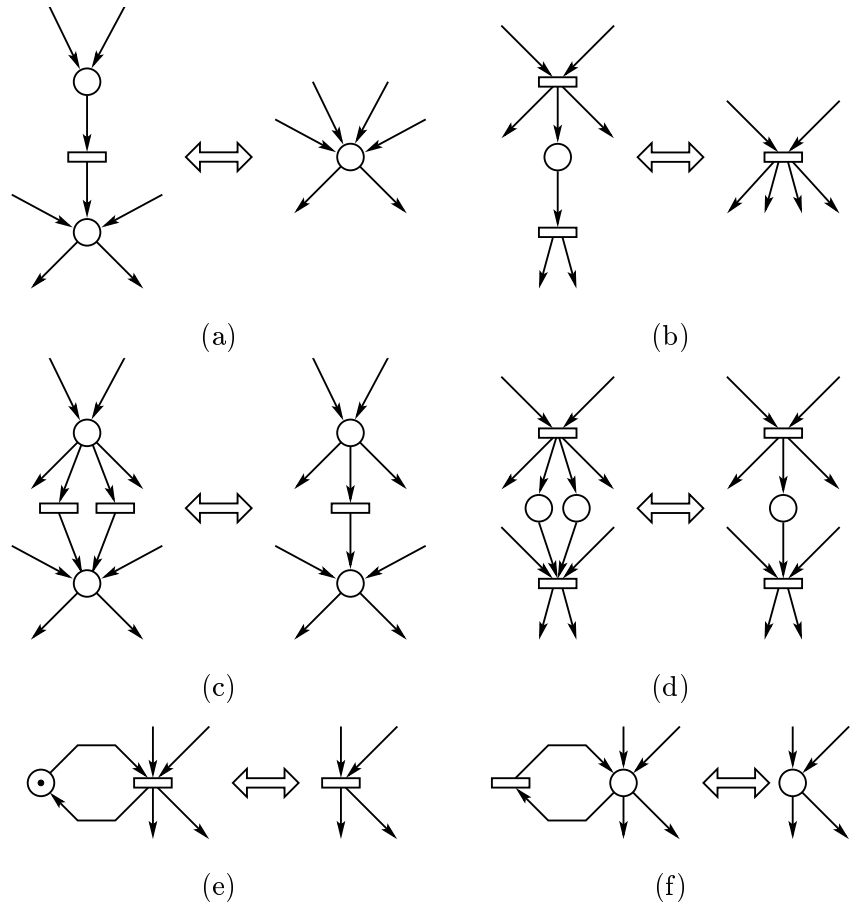


Figura 2.1: Seis transformaciones que preservan vivacidad y k -limitación.

Es habitual en las técnicas estructurales operar con productos de una matriz por un vector tanto a izquierda como a derecha. La notación matemática habitual para estos productos consiste en considerar los vectores como matrices columna y por lo tanto los productos vector por matriz se denotan de la forma $\mathbf{y}^t \cdot \mathbf{C}$ (\mathbf{y} es el vector y \mathbf{C} la matriz). En el campo concreto que se está desarrollando, esta notación supone poner constantemente superíndices a los vectores, por lo que la comunidad internacional ha decidido cambiar la notación y eliminar estos superíndices, entendiéndose que cuando un vector columna multiplica una matriz por la izquierda hay que trasponerlo. De esta forma el producto de vector por matriz se denotará a

lo largo de toda la memoria como $\mathbf{y} \cdot \mathbf{C}$.

El siguiente teorema da una caracterización algebraica de la limitación estructural.

Teorema 2.45 [Mur89] *Una P/T red \mathcal{N} es estructuralmente limitada si y sólo si existe un vector $\mathbf{y} > \mathbf{0}$ tal que $\mathbf{y} \cdot \mathbf{C} \leq \mathbf{0}$.*

Definición 2.46 *Una P/T red \mathcal{N} es conservativa si y sólo si existe un vector $\mathbf{y} \geq \mathbf{1}$ tal que $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$.*

Definición 2.47 *Una P/T red \mathcal{N} es consistente si y sólo si existe un vector $\mathbf{x} \geq \mathbf{1}$ tal que $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$.*

El siguiente teorema establece la existencia de un invariante de marcado para cualquier P/T sistema consistente.

Teorema 2.48 [Mur89] *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema con \mathcal{N} consistente. Entonces existe $\mathbf{x} \geq \mathbf{1}$ tal que $\mathbf{m} \cdot \mathbf{x} = \mathbf{m}_0 \cdot \mathbf{x}$ para todo $\mathbf{m} \in \mathbf{R}(\mathcal{S})$.*

Los siguientes teoremas establecen relaciones entre algunas propiedades definidas anteriormente válidas para cualquier P/T red o P/T sistema.

Teorema 2.49 [Mur89] *Sea \mathcal{N} una P/T red estructuralmente viva y estructuralmente limitada. Entonces \mathcal{N} es consistente y conservativa.*

Teorema 2.50 [Shi87, Ter94] *Sea \mathcal{S} un P/T sistema vivo y limitado. Entonces \mathcal{S} es fuertemente conexo y consistente.*

Definición 2.51 *Sea \mathcal{N} una P/T red. Un p-semiflujo o componente conservativa de \mathcal{N} es un vector $\mathbf{y} \geq \mathbf{0}$, $\mathbf{y} \neq \mathbf{0}$ tal que $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$.*

Definición 2.52 *Sea \mathcal{N} una P/T red. Un t-semiflujo o componente repetitiva de \mathcal{N} es un vector $\mathbf{x} \geq \mathbf{0}$, $\mathbf{x} \neq \mathbf{0}$ tal que $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$.*

Definición 2.53 *Se denomina soporte de un vector \mathbf{v} al conjunto de índices de sus componentes no nulas. $\|\mathbf{v}\| = \{i \mid \mathbf{v}[i] \neq 0\}$.*

Definición 2.54 *Un p-semiflujo (t-semiflujo) es mínimo si y sólo si su soporte no es un superconjunto propio de otro p-semiflujo (t-semiflujo) y el máximo común divisor de sus componentes es 1.*

Algoritmo 2.55 [MS82] **Cálculo de p-semiflujos mínimos.****input:** Matriz \mathbf{C} de incidencia de \mathcal{N} . $n := |P|; m := |T|$ $\mathbf{A} := \mathbf{C}$ $\mathbf{D} := \mathbf{I}_n$ (matriz identidad de dimensión n).**for** $i := 1$ **to** n **do** Añadir a $[\mathbf{D}|\mathbf{A}]$ todas las filas que resulten de una combinación lineal positiva de pares de filas de $[\mathbf{D}|\mathbf{A}]$ y que anulen la columna i de \mathbf{A} . Eliminar de $[\mathbf{D}|\mathbf{A}]$ todas las filas cuya columna i de \mathbf{A} sea no nula.**end for**Las filas de \mathbf{D} son p-semiflujos de \mathcal{N} . Tras una eventual normalización se obtienen los p-semiflujos mínimos.**output:** Matriz \mathbf{D} **2.2.5 Lugares implícitos****Notación 2.56** Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema y $p \notin P$ un lugar que se quiere añadir a \mathcal{N} con vector de incidencia \mathbf{l}_p y marcado inicial $\mathbf{m}_0[p]$. Se denota por $\mathcal{S}_p = \langle \mathcal{N}_p, \mathbf{m}_0 \cup \mathbf{m}_0[p] \rangle$ al P/T sistema extendido en el lugar p .**Definición 2.57** [Sil85] **Lugar implícito (IP).**Sea \mathcal{S}_p un P/T sistema. El lugar p es implícito si y sólo si $L(\mathcal{S}_p) = L(\mathcal{S})$. Esto equivale a que para todo \mathbf{m} marcado alcanzable en \mathcal{S} , si $\mathbf{m} \geq \mathbf{Pre}[t_k]$ entonces $\mathbf{m}[p] \geq \mathbf{Pre}[t_k]$.

Por lo tanto, un lugar implícito nunca es el único lugar que impide la sensibilización de una transición. Un lugar puede ser implícito o no dependiendo de su marcado inicial.

Definición 2.58 [CS91] **Lugar estructuralmente implícito (SIP).**Sea \mathcal{N}_p una P/T red. El lugar p es estructuralmente implícito si y sólo si para todo \mathbf{m}_0 de \mathcal{N} existe $\mathbf{m}_0[p] \in \mathbb{Z}^+$ tal que p es implícito en el P/T sistema $\langle \mathcal{N}_p, \mathbf{m}_0 \cup \mathbf{m}_0[p] \rangle$.

Existe una caracterización algebraica para los SIP's.

Teorema 2.59 [CS91] Un lugar p es SIP en \mathcal{N}_p si y sólo si existe un vector $\mathbf{y} \geq \mathbf{0}$ tal que $\mathbf{y} \cdot \mathbf{C} \leq \mathbf{l}_p$.

Todo SIP admite un marcado inicial que lo hace implícito. Sería interesante entonces conocer el mínimo marcado inicial que hace implícito a un SIP. En general sólo puede calcularse por medio de técnicas enumerativas, pero empleando un *problema de programación lineal* es posible calcular una cota superior de este marcado mínimo inicial.

Algoritmo 2.60 [CS91] **Cota superior para $\mathbf{m}_0[p]$ (p un SIP).**

input: $\mathcal{S}, \mathbf{l}_p$

$$\begin{aligned} v = \min \quad & \mathbf{y} \cdot \mathbf{m}_0 + \mu \text{ sujeto a} \\ & \mathbf{y} \cdot \mathbf{C} \leq \mathbf{l}_p \\ & \mathbf{y} \geq \mathbf{0} \\ & \mathbf{y} \cdot \mathbf{Pre}[t_k] + \mu \geq \mathbf{Pre}[p, t_k] \quad \forall t_k \in p^\bullet \end{aligned}$$

output: $\mathbf{m}_0[p] \geq \max\{v, 0\}$

Definición 2.61 [CS91] **Lugar SIP marcado (MSIP).**

Sea \mathcal{S} un P/T sistema y p un lugar con vector de incidencia \mathbf{l}_p . El lugar p es un lugar estructuralmente implícito marcado (MSIP) en \mathcal{S}_p si y sólo si existe $\mathbf{y} \geq \mathbf{0}$ tal que $\mathbf{y}[p] = 0$ y $\mathbf{l}_p = \mathbf{y} \cdot \mathbf{C}$.

En cualquier marcado alcanzable de \mathcal{S}_p , el marcado de un MSIP es una función lineal del marcado de un conjunto de lugares de \mathcal{S} , por lo que el marcado de un MSIP es una variable de estado redundante.

Definición 2.62 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un P/T sistema y $p, q, r \in P$ tres lugares del mismo. Se dice que p es implícito respecto a q y r si y sólo si p es un lugar implícito en el sistema generado por p, q y r .

2.3 Redes de Petri estocásticas

En la sección anterior se han desarrollado los principales conceptos y propiedades de las redes de Petri no temporizadas. Si se quieren realizar estudios sobre las prestaciones de los sistemas, es necesario introducir una interpretación temporal a los modelos. Este será el propósito de esta sección.

En su definición original, las redes de Petri no incluían la noción de tiempo y por lo tanto, sólo se podía modelar el comportamiento lógico de los sistemas. Históricamente ha habido dos formas de introducir una interpretación temporal a las redes de Petri; asociando el tiempo a los lugares [Sif78] o a las transiciones [Ram74]. Como normalmente las transiciones modelan

las actividades del sistema, parece más natural asociar la duración de las actividades a las transiciones del modelo. Esta será la interpretación que se empleará a lo largo de la memoria.

La inclusión de tiempos en redes de Petri permite dos interpretaciones diferentes para la regla de disparo; el *disparo en tres fases* o el *disparo atómico*. En el disparo en tres fases, cuando una transición está sensibilizada toma las marcas necesarias de sus lugares de entrada y los retiene el tiempo que dura su actividad. Pasado este tiempo se produce el disparo de la transición y por lo tanto la salida de marcas a los lugares de salida. En el disparo atómico, las marcas de los lugares de entrada de la transición sensibilizada permanecen en su sitio mientras dure la actividad de la transición. Transcurrido este tiempo se produce el disparo de la transición y el cambio de marcas en tiempo 0. En esta memoria se utilizará la interpretación de disparo atómico.

También existen diferentes interpretaciones por lo que respecta a la resolución de conflictos [AMBB⁺89]. La primera consiste en considerar *inmediatas* (se disparan en tiempo 0) las transiciones en conflicto. De esta forma las transiciones inmediatas tienen prioridad de disparo sobre las temporizadas. Para resolver los conflictos entre transiciones inmediatas se asocia a cada transición un peso que permite calcular su probabilidad de disparo (*redes de Petri estocásticas generalizadas* [AMBC84, AMBCC87]). Con esta política la resolución de conflictos se separa de la duración de las actividades. La segunda política de resolución de conflictos se hace entre transiciones temporizadas y se llama *política de carrera*. En esta interpretación dadas dos transiciones en conflicto, se dispara primero la que antes termine su actividad. A lo largo de esta memoria se emplearán las dos políticas. Si hay un conflicto entre transiciones temporizadas se supone que se resuelve por política de carrera y si el conflicto es entre transiciones inmediatas se emplean los pesos asociados a cada transición para resolverlo.

La última ambigüedad que puede aparecer en la interpretación temporal de las redes de Petri tiene que ver con el grado de sensibilización de las transiciones (ver definición 2.27). Se puede suponer que cada transición tiene un solo servidor por lo que trabaja una velocidad independiente del grado de sensibilización, o que tiene varios servidores por lo que su velocidad de trabajo depende del grado de sensibilización, es decir, del marcado de sus lugares de entrada. En el primer caso se tiene *semántica de un solo servidor* [Mol82, FN85a] y en el segundo caso semántica de varios servidores o, en caso límite, *semántica de infinitos servidores* [RP84, HV85, Zub85]. Si se

utiliza semántica de infinitos servidores, es posible modelar una transición de k servidores añadiendo un lugar de entrada y salida a la transición con k marcas. Por lo tanto, desde el punto de vista de modelado, la semántica más general es la de infinitos servidores, que es la que se asumirá en todos los desarrollos teóricos de la memoria. En los ejemplos prácticos del capítulo 7 se empleará semántica de un solo servidor por motivos de implementación.

Por lo que respecta a la duración de las actividades, en la literatura aparecen diversas formas de asociar tiempos a las transiciones, normalmente dependiendo del campo de aplicación de los modelos y el tipo de estudio que se pretende realizar. Se pueden asociar duraciones deterministas o aleatorias. En el caso de duraciones aleatorias, se pueden emplear diversas distribuciones. Un caso particular es el que se emplea a lo largo de esta memoria, en el que las transiciones temporizadas tienen asociados *tiempos aleatorios distribuidos exponencialmente*. Al asociar variables aleatorias exponenciales a las transiciones del modelo se pueden utilizar técnicas de resolución basadas en *cadena de Markov en tiempo continuo* (CTMC).

Siguiendo las interpretaciones anteriores, a lo largo de esta memoria se emplearán *redes de Petri estocásticas* (SPN) [Mo181, FN85b, BT81] o *redes de Petri estocásticas generalizadas* (GSPN) [AMBC84, AMBCC87] cuyas definiciones formales se exponen ahora.

Definición 2.63 Redes de Petri estocásticas (SPN).

Una red de Petri estocástica (SPN) es un par $\langle \mathcal{S}, w \rangle$ donde \mathcal{S} es un P/T sistema y $w : T \rightarrow (0, \infty)$ es una función real estrictamente positiva que asocia a cada transición $t \in T$ un tiempo de disparo distribuido exponencialmente con tasa $w(t)$.

Una SPN es un P/T sistema en el que se asocia un tiempo de disparo distribuido exponencialmente a cada transición. Una variable aleatoria exponencial X se especifica por su *tasa* λ o por su *valor medio* $E[X] = 1/\lambda$ y es la única variable aleatoria continua sin memoria.

Definición 2.64 SPN's generalizadas (GSPN).

Una red de Petri estocástica generalizada (GSPN) es una ocho-tupla

$$\langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{Inh}, \mathbf{m}_0, \mathbf{\Pi}, w \rangle$$

donde P y T son conjuntos disjuntos, finitos no vacíos de lugares y transiciones, \mathbf{Pre} , \mathbf{Post} e \mathbf{Inh} son las matrices de incidencia anterior, incidencia

posterior e inhibición respectivamente (todas de enteros no negativos de tamaño $|P| \times |T|$). El vector $\mathbf{m}_0 \in \{\mathbf{Z}^+\}^{|P|}$ es el marcado inicial, que asocia a cada lugar un entero no negativo, el vector $\mathbf{\Pi} \in \{0, 1\}^{|T|}$ asocia a cada transición una prioridad en $\{0, 1\}$ y $w : T \rightarrow (0, \infty)$ es una función real estrictamente positiva que asigna a cada transición $t \in T$ un peso $w(t)$. Las transiciones de prioridad 0 se llaman temporizadas y las de prioridad 1 inmediatas.

Las GSPN's se representan gráficamente como las P/T redes diferenciando las transiciones inmediatas de las temporizadas. Las transiciones inmediatas se representan por medio de barras delgadas negras y las temporizadas con cajas rectangulares. Los nuevos *arcos inhibidores* (matriz \mathbf{Inh}) se representan como los arcos de una P/T red añadiendo un círculo en el extremo correspondiente a la transición del arco. En la definición de GSPN hay que notar una interpretación de la función w ligeramente diferente a la de SPN's. Si la transición $t \in T$ es temporizada, $w(t)$ es la tasa de la exponencial asociada a la duración de su actividad y si t es inmediata, $w(t)$ es el peso asociado a su disparo para la resolución de conflictos.

La inclusión de prioridades en las transiciones modifica ligeramente la regla de sensibilización y disparo en GSPN's. Las transiciones inmediatas tienen prioridad de disparo sobre las temporizadas, por lo que si en un marcado alcanzable \mathbf{m} hay una transición inmediata t_1 sensibilizada, todas las transiciones sensibilizadas en \mathbf{m} deben ser inmediatas. Denotando este conjunto de transiciones sensibilizadas en \mathbf{m} por $\{t_i\}_{i=1}^k$, la probabilidad de que se dispare la transición inmediata t_j con $1 \leq j \leq k$ es

$$\text{Prob}\{t_j \text{ se dispara en } \mathbf{m}\} = \frac{w(t_j)}{\sum_{i=1}^k w(t_i)}.$$

Las GSPN's suponen una extensión de las SPN's por lo que respecta a las transiciones inmediatas y arcos inhibidores. En realidad, a lo largo de la memoria se trabaja con SPN's o con extensiones de ellas por medio de transiciones inmediatas para la resolución de conflictos, es decir, no se trabaja con arcos inhibidores, lo que permitiría definir una subclase de GSPN's más sencilla.

Una vez se añade una interpretación temporal a los sistemas, es posible realizar estudios sobre la evolución en el tiempo de los mismos e intentar calcular *medidas* o *índices de prestaciones* sobre los mismos. Se van a exponer los índices de prestaciones habitualmente empleados en SPN's y GSPN's.

Definición 2.65 Throughput de una transición.

Dada una SPN o GSPN, el throughput de la transición $t \in T$, denotado por $\mathcal{X}(t)$, es el número medio de disparos por unidad de tiempo que realiza la transición t .

Definición 2.66 Throughput relativo o ratio de visita.

Dada una SPN o GSPN y $t_1 \in T$ una transición de referencia, la ratio de visita o throughput relativo de la transición $t \in T$ normalizada para la transición t_1 y denotada por $v(t)^{(1)}$, es $v(t)^{(1)} = \mathcal{X}(t)/\mathcal{X}(t_1)$. Se denota por $\mathbf{v} \in (\mathbb{R}^+)^{|T|}$ al vector de ratios de visita.

Normalmente los ratios de visita se supondrán normalizadas para la primera transición de la red y se eliminará el superíndice.

La diferencia entre el vector de ratios de visita y un t-semiflujo es que los t-semiflujos son vectores de enteros no negativos y el de ratios de visita es de reales no negativos en general.

Teorema 2.67 [Cam90] Dada una SPN limitada y \mathbf{v} su vector de ratios de visita, se cumple $\mathbf{C} \cdot \mathbf{v} = 0$.

Definición 2.68 Utilización de una transición.

Dada una SPN o GSPN, la utilización de la transición $t \in T$, denotado por $\mathcal{U}(t)$, es la probabilidad de encontrar la transición sensibilizada.

Definición 2.69 Dada una SPN o GSPN, la demanda media de servicio de la transición $t \in T$, denotado por $D(t)$, es $D(t) = v(t)s(t) = v(t)/w(t)$ donde $s(t) = 1/w(t)$ es el tiempo medio de servicio de la transición t .

La ventaja de asociar a las transiciones un tiempo de disparo distribuido exponencialmente consiste en que la evolución temporal de la SPN o GSPN es equivalente al de una cadena de Markov en tiempo continuo. Se van a dar ahora las principales definiciones y propiedades de las cadenas de Markov en tiempo continuo.

Definición 2.70 Un proceso estocástico es una familia de variables aleatorias $\{X(t) \mid t \in \Theta\}$ sobre el mismo espacio de probabilidad, indexadas por el parámetro t que varía en un conjunto ordenado de índices Θ . Si Θ es discreto (continuo) el proceso estocástico se dice de tiempo discreto (continuo).

Normalmente el conjunto Θ es \mathbb{R} y se interpreta como tiempo. La variable aleatoria $X(t)$ es la observación del proceso estocástico en el instante t . Los valores que pueden tomar las variables aleatorias de un proceso estocástico constituyen el espacio de estados del proceso. Este espacio de estados puede ser continuo o discreto. En esta memoria se asumirá que los espacios de estados son finitos, por lo tanto discretos.

Definición 2.71 *Un proceso estocástico $\{X(t) \mid t \in \Theta\}$ cumple la propiedad de Markov si para cualesquiera $t_0 < t_1 < \dots < t_n < t$ se cumple:*

$$\text{Prob}\{X(t) \leq x \mid X(t_i) \leq x_i, 0 \leq i \leq n\} = \text{Prob}\{X(t) \leq x \mid X(t_n) \leq x_n\}$$

La evolución de un proceso estocástico con la propiedad de Markov depende únicamente de la última observación realizada y no de las anteriores.

Definición 2.72 (CTMC).

Una cadena de Markov en tiempo continuo (CTMC) es un proceso estocástico de tiempo continuo que cumple la propiedad de Markov.

Notación 2.73 *En una cadena de Markov en tiempo continuo se denota por $p_{ij}(t, s)$ a la probabilidad $\text{Prob}\{X(s) = j \mid X(t) = i\}$.*

Definición 2.74 *Una cadena de Markov en tiempo continuo se dice homogénea si y sólo si $p_{ij}(t, s) = p_{ij}(t + u, s + u)$ para todo $t, s, u \in \Theta$.*

Las CTMC's que aparecen a lo largo de la memoria son homogéneas, por lo que se hablará de CTMC's para referirse a CTMC's homogéneas. En las CTMC's homogéneas la probabilidad de paso de un estado a otro no depende, por la definición anterior, del tiempo que se lleva en el estado origen, por lo que el tiempo de paso de un estado a otro está distribuido exponencialmente (por la propiedad de falta de memoria de la variable aleatoria exponencial). Por lo tanto, la evolución de una CTMC homogénea viene determinada por las tasas de paso entre estados.

Notación 2.75 *En una cadena de Markov en tiempo continuo homogénea se denota por p_{ij} a la tasa de paso del estado i al j .*

Una CTMC se puede representar gráficamente por medio de un grafo dirigido etiquetado cuyos vértices son los estados de la CTMC y cuyas aristas describen los cambios de estado. Las aristas van etiquetadas con las tasas de paso entre estados. Desde un estado la CTMC puede evolucionar a distintos estados con distintas tasas de paso. En estas condiciones, el tiempo de permanencia en un estado i está distribuido exponencialmente con tasa $\sum_j p_{ij}$ (porque el mínimo de variables aleatorias exponenciales tiene distribución exponencial de tasa la suma de las tasas de las variables que intervienen). Las transiciones de un estado a él mismo pueden eliminarse fácilmente debido a la propiedad de Markov, lo que permite representar una CTMC por medio de una matriz \mathbf{Q} con las tasas de paso entre estados, llamada *generador infinitesimal* de la CTMC.

Definición 2.76 *El generador infinitesimal de una CTMC homogénea de n estados ($n \in \mathbb{N}$) es la matriz \mathbf{Q} de tamaño $n \times n$ tal que $\mathbf{Q}[i, j] = p_{ij}$ si $i \neq j$ y $\mathbf{Q}[i, i] = -\sum_{j=1}^n p_{ij}$ con $1 \leq i, j \leq n$. Por lo tanto $\mathbf{Q} \cdot \mathbf{1} = \mathbf{0}$.*

En las CTMC's es posible realizar estudios de su *estado transitorio* (tiempo finito) y en *estado estacionario* (comportamiento límite cuando el tiempo tiende a infinito). En esta memoria se estudia únicamente el comportamiento límite o en estado estacionario de los sistemas, por lo que sólo interesa conocer los fundamentos de la evolución en estado estacionario de una CTMC. La exposición que se hace a continuación no es totalmente formal para reducir su extensión (ver [KS76] para un desarrollo en profundidad).

Definición 2.77 *Sea $\{X(t) | t \in \mathbb{R}^+\}$ una CTMC homogénea de espacio de estados $1 \leq i \leq n$ ($i \in \mathbb{N}$). El estado i se dice transitorio si y sólo si $\text{Prob}\{X(t) \neq i \ \forall t > 0 \mid X(0) = i\} > 0$.*

Un estado de una CTMC es transitorio si existe una probabilidad estrictamente positiva de que la CTMC no regrese al estado.

Definición 2.78 *Sea $\{X(t) | t \in \mathbb{R}^+\}$ una CTMC homogénea de espacio de estados $1 \leq i \leq n$ ($i \in \mathbb{N}$). El estado i se dice absorbente si y sólo si $p_{ij} = 0$ para todo $1 \leq j \leq n$ y $j \neq i$.*

Si una CTMC tiene un estado absorbente, una vez llega al estado la CTMC no puede salir de él, por lo que en ese caso es sencillo calcular la distribución en estado estacionario.

Teorema 2.79 [KS76] *Sea i un estado transitorio de una cadena de Markov en tiempo continuo homogénea. Entonces $\lim_{t \rightarrow \infty} \text{Prob}\{X(t) = i\} = 0$.*

Desde el punto de vista de la evolución en estado estacionario de una CTMC, los estados transitorios no suponen ningún problema, ya que la CTMC los abandona con probabilidad 1, es decir, la probabilidad de que la CTMC se encuentre en el estado transitorio cuando el tiempo es suficientemente grande tiende a 0.

Definición 2.80 *Una CTMC homogénea se dice ergódica si y sólo si su grafo asociado es fuertemente conexo.*

Las CTMC's ergódicas tienen un comportamiento en estado estacionario único que es posible calcular.

Teorema 2.81 [KS76] *Sea $\{X(t) | t \in \mathbb{R}^+\}$ una CTMC ergódica de espacio de estados $1 \leq i \leq n$ ($i \in \mathbb{N}$). En estas condiciones existen los límites $\pi_i = \lim_{t \rightarrow \infty} \text{Prob}\{X(t) = i\}$ y son estrictamente positivos para $1 \leq i \leq n$.*

Definición 2.82 *Sea $\{X(t) | t \in \mathbb{R}^+\}$ una CTMC ergódica de espacio de estados $1 \leq i \leq n$ ($i \in \mathbb{N}$). La probabilidad en estado estacionario del estado i es el valor π_i del teorema anterior. Al vector $\boldsymbol{\pi}$ con las probabilidades en estado estacionario de todos los estados se le llama distribución en estado estacionario de la CTMC.*

Teorema 2.83 [KS76] *Sea $\{X(t) | t \in \mathbb{R}^+0\}$ una CTMC ergódica de estados $1 \leq i \leq n$ ($i \in \mathbb{N}$). La distribución en estado estacionario $\boldsymbol{\pi}$ cumple $\boldsymbol{\pi} \cdot \mathbf{Q} = \mathbf{0}$ y $\boldsymbol{\pi} \cdot \mathbf{1} = 1$.*

Por lo tanto, es posible calcular la distribución en estado estacionario de una CTMC ergódica resolviendo un sistema de $n + 1$ ecuaciones (siempre hay una ecuación redundante) con n incógnitas donde n es el número de estados de la CTMC. En general es posible calcular la distribución en estado estacionario en CTMC's que no sean ergódicas siempre y cuando su grafo asociado tenga unos estados transitorios que desemboquen todos en una única componente fuertemente conexa terminal (que la CTMC no pueda abandonar). En ese caso las probabilidades límite son 0 para los estados transitorios y estrictamente positivas para el resto.

Con la distribución en estado estacionario de una CTMC ergódica es posible calcular el valor en estado estacionario de cualquier índice de prestaciones que pueda ponerse como función de los elementos de π .

Se está en condiciones de exponer la relación entre SPN's, GSPN's y CTMC's.

Definición 2.84 *Sea $\langle \mathcal{S}, w \rangle$ una SPN. La CTMC isomorfa a $\langle \mathcal{S}, w \rangle$ es la que tiene como grafo dirigido etiquetado asociado $\text{RG}(\mathcal{S})$ sustituyendo en cada etiqueta $\mathbf{m} \xrightarrow{t} \mathbf{m}'$ la transición $t \in T$ por $w(t)$ si se utiliza semántica de un solo servidor o por $\mathbf{e}(\mathbf{m})[t]w(t)$ si se utiliza semántica de infinitos servidores.*

Teorema 2.85 [AMBC⁺95] *Sea $\langle \mathcal{S}, w \rangle$ una SPN limitada y con estados recurrentes. Entonces su CTMC isomorfa es finita y tiene una única distribución en estado estacionario. Además, si \mathcal{S} es reversible entonces su CTMC isomorfa es ergódica.*

Por lo tanto, para poder calcular el comportamiento en estado estacionario de una SPN es necesario que tenga estados recurrentes, lo que asegura la existencia de una única componente fuertemente conexa terminal (que no puede abandonarse) en su RG y por lo tanto en su CTMC isomorfa. Como mucho habrá algunos estados transitorios que desembocan, después del disparo de algunas transiciones, en un estado recurrente \mathbf{m} . Como es evidente que $\langle \mathcal{N}, \mathbf{m} \rangle$ es reversible entonces $\text{RG}(\mathcal{N}, \mathbf{m})$ es fuertemente conexo y por lo tanto su CTMC isomorfa es ergódica. La hipótesis de limitación se añade para que la CTMC isomorfa sea finita y sea posible el cálculo de la distribución en estado estacionario.

Definición 2.86 *Sea $\langle \mathcal{S}, w \rangle$ una SPN limitada y con estados recurrentes. La distribución en estado estacionario de $\langle \mathcal{S}, w \rangle$ es el vector $\pi \in (\mathbb{R}^+)^{|\mathbb{R}(\mathcal{S})|}$ de probabilidades en estado estacionario de su CTMC isomorfa.*

En el caso de GSPN's el problema del cálculo de su CTMC isomorfa es un poco más complicado, ya que su RG contiene *estados tangibles* (en los que todas las transiciones sensibilizadas son temporizadas) y *estados intangibles* (en los que hay alguna transición inmediata temporizada). Desde el punto de vista de evaluación de prestaciones pueden eliminarse los estados intangibles porque el tiempo que el sistema pasa en ellos es 0. Por ello para generar la CTMC isomorfa a una GSPN hay que eliminar los estados intangibles

del RG y posteriormente calcular las probabilidades de cada camino en el RG formado por transiciones inmediatas de acuerdo con las probabilidades de disparo de cada transición inmediata.

A lo largo de toda la memoria se supondrá que las SPN's o GSPN's admiten distribución en estado estacionario, es decir, que las admiten sus CTMC's isomorfas.

Una vez generada la CTMC isomorfa a una SPN o GSPN y su distribución estado estacionario es posible calcular diversos índices de prestaciones. Ahora se exponen las fórmulas de los índices de prestaciones más frecuentemente utilizados.

Teorema 2.87 [AMBC⁺95] *Sea $\langle \mathcal{S}, w \rangle$ una SPN con distribución en estado estacionario π . El throughput en estado estacionario de la transición $t \in T$ es:*

$$\mathcal{X}(t) = w(t) \cdot \sum_{\substack{\mathbf{m} \in \mathbb{R}(\mathcal{S}) \\ e(\mathbf{m})[t] > 0}} \pi[\mathbf{m}].$$

Teorema 2.88 [AMBC⁺95] *Sea $\langle \mathcal{S}, w \rangle$ una SPN con distribución en estado estacionario π . La utilización en estado estacionario de la transición $t \in T$ es:*

$$\mathcal{U}(t) = \sum_{\substack{\mathbf{m} \in \mathbb{R}(\mathcal{S}) \\ e(\mathbf{m})[t] > 0}} \pi[\mathbf{m}].$$

De forma análoga se puede definir cualquier índice de prestaciones en función de los estados alcanzables del sistema y calcular su valor en estado estacionario a partir de π . A lo largo de toda la memoria, cuando se emplee el nombre de cualquier índice de prestaciones estará referido a su valor en estado estacionario.

2.4 Subclases de redes de Petri

En esta sección se van a definir las clases de redes que se van a utilizar a lo largo de la memoria, así como sus principales propiedades.

2.4.1 Grafos marcados

Definición 2.89 [CHEP71] **Grafo marcado (MG).**

Un grafo marcado (MG) es una P/T red ordinaria tal que para todo lugar $p \in P$ se verifica $|\bullet p| = |p \bullet| = 1$.

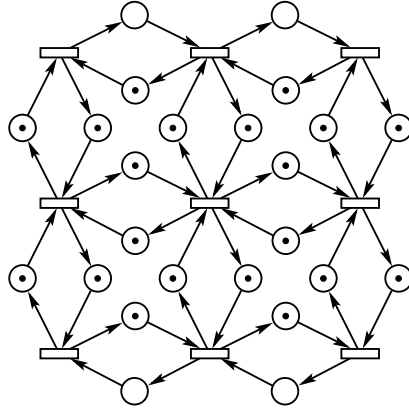


Figura 2.2: Grafo marcado (MG).

En la figura 2.2 puede verse un ejemplo de MG. La vivacidad de un MG es fácil de comprobar a partir del siguiente resultado.

Teorema 2.90 [Mur89] *Sea \mathcal{N} un MG. Entonces \mathbf{m}_0 es un marcado inicial que lo hace vivo si y sólo si \mathbf{m}_0 tiene marcas en cualquier ciclo de \mathcal{N} .*

Corolario 2.91 *Los MG's son estructuralmente vivos.*

Teorema 2.92 [Sil85] *Sea \mathcal{N} un MG. Entonces se cumple:*

1. *Son equivalentes:*
 - i) *\mathcal{N} es fuertemente conexo.*
 - ii) *\mathcal{N} es estructuralmente limitado.*
 - iii) *\mathcal{N} es conservativo.*
2. *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ vivo. Entonces $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ es limitado si y sólo si \mathcal{N} es estructuralmente limitado.*

El siguiente resultado es evidente a partir de la definición de MG.

Teorema 2.93 *Sea \mathcal{N} un MG. Entonces $\mathbf{1}$ es un t -semiflujo de \mathcal{N} , es decir, $\mathbf{C} \cdot \mathbf{1} = \mathbf{0}$. Por lo tanto, todo MG es consistente (ver definición 2.47).*

Una consecuencia inmediata de este teorema es el siguiente corolario.

Corolario 2.94 Sea $\langle \mathcal{S}, w \rangle$ un MG estocástico y $t_1, t_2 \in T$ dos transiciones del mismo. Entonces $\mathcal{X}(t_1) = \mathcal{X}(t_2)$.

Demostración:

Por los teoremas 2.67 y 2.93, todo MG estocástico tiene como vector de ratios de visita $\mathbf{1}$. Por la definición 2.66 de ratios de visita se sigue que el throughput de dos transiciones cualesquiera debe ser el mismo. \diamond

2.4.2 Máquinas de estados

Definición 2.95 Máquina de estados (SM).

Una máquina de estados (SM) es una P/T red ordinaria tal que para toda transición $t \in T$ se verifica $|\bullet t| = |t \bullet| = 1$.

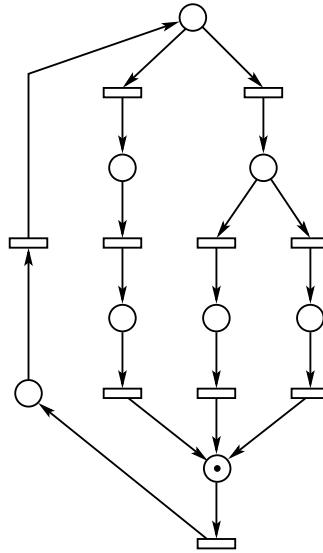


Figura 2.3: Máquina de estados (SM).

En la figura 2.3 puede verse un ejemplo de SM. En el siguiente teorema se exponen las principales propiedades de las SM's que se van a emplear a lo largo de la memoria.

Teorema 2.96 *Sea \mathcal{N} una SM. Se cumple:*

1. \mathcal{N} es conservativa y $\text{rank}(\mathbf{C}) = |P| - 1$.
2. \mathcal{N} es fuertemente conexa si y sólo si \mathcal{N} es consistente.
3. $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ es vivo si y sólo si \mathcal{N} es fuertemente conexa y \mathbf{m}_0 tiene al menos una marca.
4. Si \mathcal{N} es fuertemente conexa, sus t -semiflujos mínimos son sus ciclos.

2.4.3 Grafos marcados con pesos

Definición 2.97 Grafo marcado con pesos (WTS).

Un grafo marcado con pesos (WTS) es un P/T sistema tal que para todo lugar $p \in P$ se verifica $|\bullet p| = |p \bullet| = 1$.

En la figura 2.4 puede verse un ejemplo de WTS. Las principales propiedades de los WTS's que se van a emplear a lo largo de la memoria son las siguientes:

Teorema 2.98 [TCWCS92] *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS. Se cumple:*

1. \mathcal{N} es consistente si y sólo si $\text{rank}(\mathbf{C}) = |T| - 1$.
2. Si \mathcal{N} es consistente entonces tiene un único t -semiflujo mínimo.
3. Son equivalentes:
 - i) \mathcal{N} es estructuralmente viva y estructuralmente limitada.
 - ii) \mathcal{N} es consistente y conservativa.
 - iii) \mathcal{N} es fuertemente conexa y consistente.
4. Si \mathcal{N} es fuertemente conexa entonces \mathcal{S} es vivo si y sólo si es libre de bloqueo.

Como consecuencia del apartado 2 se tiene que el vector de ratios de visita de un WTS estocástico viene determinado por la estructura de la red, y por lo tanto también la relación entre los throughput de sus transiciones.

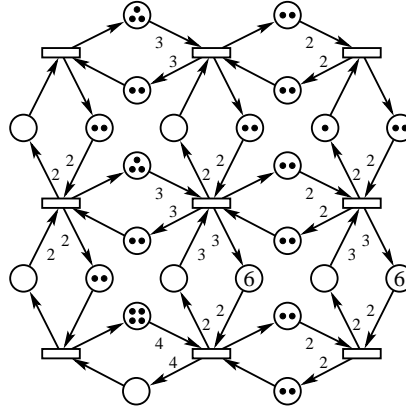


Figura 2.4: Grafo marcado con pesos (WTS).

Teorema 2.99 [Ter94] *Sea \mathcal{S} un WTS consistente. Entonces \mathcal{S} es vivo si y sólo si es reversible.*

Este resultado puede aplicarse también a MG's ya que todo MG es un WTS.

2.4.4 Sistemas deterministas de procesos secuenciales

Los *sistemas deterministas de procesos secuenciales* (DSSP) se obtienen por medio de un sencillo diseño modular. Varias unidades funcionales (en este caso modeladas por medio de máquinas de estados vivos y 1-limitadas) funcionan de forma concurrente y cooperan entre ellas por medio de un mecanismo de paso de mensajes a través de canales. Los canales se modelan por medio de lugares con arcos de entrada y salida a los que se les puede añadir pesos. De esta forma se puede modelar el envío de mensajes en lotes. Los canales son de destino privado en el sentido de que sus mensajes van a una única unidad funcional. De esta forma se evita la competencia por los mensajes entre las distintas unidades funcionales. Además los canales no afectan a la resolución de los conflictos internos que pueda haber en las unidades funcionales, es decir, los conflictos son libres.

En la literatura han aparecido diversas clases siguiendo este diseño modular [Rei82, Sou93, TSCC95, RTS95]. En esta memoria se adopta la siguiente clase.

Definición 2.100 (DSSP).

Un P/T sistema $\langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{m}_0 \rangle$ es un sistema determinista de procesos secuenciales (DSSP) si cumple:

- i) $P = B \cup (\bigcup_{i=1}^n P_i)$ con $P_i \cap P_j = \emptyset$ y $P_i \cap B = \emptyset$ para todo $1 \leq i, j \leq n$, $i \neq j$.
- ii) $T = \bigcup_{i=1}^n T_i$ con $T_i \cap T_j = \emptyset$ para todo $1 \leq i, j \leq n$, $i \neq j$.
- iii) Si $\mathcal{N}_i = \langle P_i, T_i, \mathbf{Pre}[P_i, T_i], \mathbf{Post}[P_i, T_i] \rangle$ entonces $\langle \mathcal{N}_i, \mathbf{m}_0[P_i] \rangle$ es una SM viva y 1-limitada para todo $1 \leq i \leq n$.
- iv) $\mathbf{Pre}[P_i, T_j] = \mathbf{Post}[P_i, T_j] = \mathbf{0}$ para todo $1 \leq i, j \leq n$, $i \neq j$.
- v) Para cada canal $b \in B$:
 1. Existe $1 \leq j \leq n$ tal que $b^\bullet \subseteq T_j$.
 2. Para todo $p \in P \setminus B$, si $t, t' \in p^\bullet$ entonces $\mathbf{Pre}[b, t] = \mathbf{Pre}[b, t']$.

Los cuatro primeros apartados establecen que un DSSP está compuesto por un conjunto de SM's disjuntas y un conjunto de canales (B). El último apartado establece que los canales tienen una única SM de salida (apartado v.1) y que su marcado no altera la resolución de los conflictos internos de las SM's (apartado v.2). Esta definición supone una ligera generalización de la de [Sou93] ya que en este trabajo las SM's debían tener también una única SM de entrada.

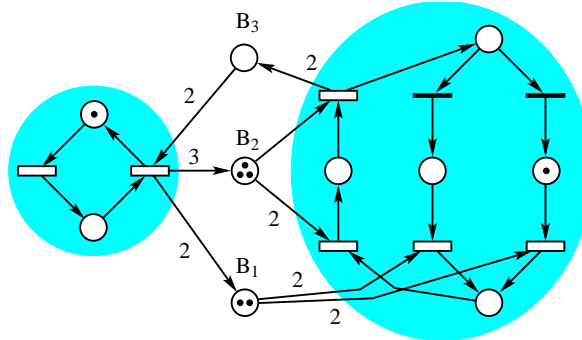


Figura 2.5: Sistema determinista de procesos secuenciales (DSSP).

En la figura 2.5 puede verse un ejemplo de DSSP compuesto por dos SM's y tres canales (los lugares etiquetados B_1 , B_2 y B_3). La principal

propiedad de los DSSP's que se va a emplear a lo largo de la memoria es la siguiente:

Teorema 2.101 [RTS98] *Todo DSSP vivo y limitado tiene estados recurrentes.*

Capítulo 3

Aproximación de throughput en grafos marcados

En este capítulo se va a explicar detalladamente el tipo de técnicas que se van a desarrollar en esta memoria. Para ello se tomará como punto de partida la técnica de [CCJS94] para el cálculo aproximado del throughput de transiciones en grafos marcados (MG's). Se presentará con algunas mejoras técnicas que permiten demostrar el fundamento teórico de su buen comportamiento. En este capítulo se puede hablar de “throughput de un MG” ya que en esta clase de redes todas las transiciones tienen el mismo throughput (ver corolario 2.94).

Para que tenga sentido aproximar el throughput de un MG es necesario exigir algunas condiciones. Por un lado el MG debe ser conexo porque en caso contrario se tienen varios MG's independientes que se pueden estudiar por separado (uno por cada componente conexa). Se restringirá el estudio a MG's limitados con el objeto de obtener cadenas de Markov en tiempo continuo (CTMC) finitas. Además, es necesario que el MG sea vivo porque si no su throughput es nulo y en ese caso no tiene sentido desarrollar una técnica de aproximación. Por último, el MG debe tener estados recurrentes, lo que asegura la existencia de distribución en estado estacionario única para su CTMC isomorfa (teorema 2.85). Aplicando los resultados de teoría estructural de MG's se tiene que, por el apartado 2 del teorema 2.92, todo MG vivo es limitado si y sólo si es estructuralmente limitado y esto es equivalente a ser fuertemente conexo (por el apartado 1 del teorema 2.92). Por lo tanto, para MG's vivos son equivalentes ser limitado y fuertemente

conexo. Por otro lado, todo MG es consistente (teorema 2.93), y un MG consistente es vivo si y sólo si es reversible (teorema 2.99), lo que asegura la ergodicidad de su CTMC isomorfa (teorema 2.85). Por lo tanto, todo MG fuertemente conexo y vivo es limitado y reversible, por lo que se desarrollará una técnica de aproximación de throughput para MG's fuertemente conexos y vivos.

Esta técnica fue presentada en el caso de corte del modelo original en dos subredes, a partir de las cuales se generaban tres sistemas agregados. Los dos primeros sistemas agregados, los de bajo nivel, estaban formados cada uno por una subred producto del corte y un resumen de la otra por medio de lugares. El tercer sistema agregado, el de alto nivel, estaba formado por el resumen de las dos subredes. Aquí se presentará de forma más general, permitiendo un corte múltiple del modelo en un número finito arbitrario de subredes. Además, la descomposición se realizará de forma que los sistemas agregados resultantes tengan la estructura más sencilla posible en cuanto a número de nodos.

Posteriormente, estos sistemas agregados se empleaban en [CCJS94] para aproximar el throughput de las transiciones del MG original por medio de un algoritmo iterativo. Por lo que respecta a esta parte numérica del método, en este capítulo se demostrará la existencia de un punto fijo para el algoritmo iterativo, lo que supone la existencia de solución para el esquema.

El capítulo está organizado de la siguiente manera. La sección 3.1 relaciona la técnica con otras existentes en la literatura. En la sección 3.2 se presentan algunas propiedades básicas sobre lugares implícitos en MG's, en la sección 3.3 se desarrolla la técnica de descomposición estructural para MG's y en la sección 3.4 el método iterativo de resolución junto con la demostración de la existencia de punto de convergencia para el mismo. Por último, en la sección 3.5 se indican las conclusiones finales del capítulo.

3.1 Literatura previa

Los MG's son una subclase de redes de Petri bien conocida. Permiten el modelado de concurrencia y sincronizaciones, pero no de decisiones. Desde el punto de vista de redes de colas, los MG's fuertemente conexos son isomorfos a las redes de colas distribución-conjunción (“fork-join”) con sincronizaciones [DLT90].

Para esta clase de redes se han desarrollado diversas técnicas de análisis.

Si se asocian tiempos aleatorios exponenciales a los disparos de las transiciones, la evolución temporal de la red es isomorfa al de una CTMC. Para el cálculo exacto de índices de prestaciones es necesario resolver esta CTMC isomorfa, lo que impide el estudio de sistemas con gran número de estados. No se pueden aplicar técnicas analíticas exactas eficientes basadas en solución en forma producto [DS92] ya que, en general, no se satisface la propiedad de balance local. En [BL92, CCCS92] se pueden encontrar técnicas basadas en el cálculo de cotas.

Por lo que respecta a técnicas de aproximación, también hay diversas aportaciones. En [AI89], se presenta un método para el análisis de redes que admitan una descomposición en su escala de tiempos (partes del modelo con velocidades muy diferentes) basada en la descomposición casi-completa de CTMC's. El mismo tipo de descomposición se emplea en [CT91] para una técnica de aproximación iterativa de redes débilmente conectadas. En [BD90a], algunas redes de colas con subredes sujetas a restricciones de población son analizadas por medio de la técnica del servidor equivalente ("flow-equivalent aggregation"). En [Mar79] se expone el método de Marie, en el que la idea es reemplazar una subred por una estación equivalente con un tiempo de servicio distribuido exponencialmente con tasa dependiente de la carga. Estas tasas se calculan a partir del estudio cada subred por separado sometida a un proceso de llegadas tipo Poisson con tasa dependiente de la carga. Otra técnica alternativa se presenta en [LW91a] para el cálculo aproximado del throughput en MG's. En este trabajo, el sistema original se divide en sistemas agregados y para obtener la aproximación del throughput se emplea un criterio de equivalencia de retardo. Las tasas de servicio de los sistemas agregados son dependientes del marcado. En [JSS92] se emplea una aproximación del tiempo de respuesta para una técnica iterativa de cálculo aproximado del throughput en MG's. Las principales diferencias con respecto a [LW91a] son dos: el corte del sistema original se basa firmemente en las propiedades cualitativas de los MG's y las tasas de servicio en los sistemas agregados son constantes.

En [Jun92] se aplican y comparan las técnicas expuestas anteriormente a SPN's. Las principales conclusiones son las siguientes. La técnica del servidor equivalente es la más eficiente (no es un método iterativo) siempre que pueda emplearse. Para ello es necesario que el comportamiento del subsistema que se agrega sea independiente del proceso de llegadas y que dependa únicamente del número de clientes en el subsistema. En muchos casos, esta suposición no es cierta (ver [JSS92]) y por lo tanto no puede

aplicarse.

El método de Marie funciona bien en muchos casos. La principal objeción es que en ocasiones la convergencia representa un problema [BD90b].

Por lo que respecta a la técnica de equivalencia de retardo de [LW91a], la convergencia también puede suponer un problema. El método se mejoró en [LW91b], donde las tasas de servicio de los sistemas agregados se hacen constantes. Algunos problemas de esta aproximación han aparecido en [JSS92] donde se demuestra que la velocidad de convergencia depende de los valores iniciales estimados para las tasas de servicio que representan a los sistemas agregados. Además, para algunos modelos los autores de [LW91a] y [LW91b] no pudieron encontrar valores iniciales para los que el método convergiera. Finalmente, el método de aproximación del tiempo de respuesta introducido en [JSS92] muestra una precisión similar al método de equivalencia de retardo, pero con menor complejidad computacional. La aproximación obtenida con este método parece no depender, según los autores, de los valores iniciales de las tasas de servicio. La principal desventaja de este método es que el modelo original debe descomponerse en dos subredes cada una de las cuales debe tener un único lugar de entrada y un único lugar de salida, lo cual limita la clase de sistemas que pueden analizarse. Una posible generalización de este tipo de corte consiste en permitir que las subredes tengan más de un lugar de entrada o de salida. Se han propuesto cortes de este tipo pero presentan serios problemas en cuanto a la precisión de los resultados [Jun92].

La técnica que se va a presentar en este capítulo se basa en el esquema iterativo de aproximación del tiempo de respuesta presentado en [CCJS94]. Esta técnica evita los problemas de aproximación de la propuesta en [JSS92], permitiendo un corte general en el modelo original. Esta técnica de descomposición se basa en la teoría estructural de MG's. En concreto, dado un corte general en el MG original (subconjunto de lugares), se desarrolla una técnica de descomposición estructural que permite dividir el MG original en K subredes y construir, a partir de ellas, K sistemas de bajo nivel y un sistema de alto nivel o esqueleto básico. Cada sistema de bajo nivel contiene completamente una de las K subredes producto del corte y un "resumen" o visión abstracta del resto de las subredes. El esqueleto básico está formado por los "resúmenes" de las K subredes y constituye una visión de alto nivel de la red original. La principal característica de la técnica de descomposición estructural es que el comportamiento de los sistemas agregados (tanto los de bajo nivel como el esqueleto básico), por lo que respecta a

marcados alcanzables y secuencias de disparo, es equivalente al del sistema original (proyectado este último sobre los nodos preservados en cada sistema agregado).

Se aprovechará esta exposición para introducir algunas mejoras. Por un lado, se permitirá el corte del sistema original en un número finito arbitrario de subredes, no sólo en dos. Por otro lado, se reducirá al máximo la estructura de los sistemas agregados sin aumentar por ello el tiempo o la memoria requerida para los cálculos. Y por último, se demostrará la existencia de un punto de convergencia para el algoritmo numérico de aproximación.

3.2 Grafos marcados y lugares implícitos

La técnica estructural de descomposición de MG's se basa en añadir al modelo original una serie de lugares implícitos [CS91] (que no modifican el comportamiento del modelo) con el objeto de resumir el comportamiento de cada subred producto del corte. Por ello es necesario estudiar las principales propiedades de los lugares implícitos.

En la sección 2.2.5 se introducen las definiciones de lugares implícitos (IP), estructuralmente implícitos (SIP), lugares estructuralmente implícitos marcados (MSIP) así como sus principales características. En esta sección se van a estudiar características particulares de los lugares implícitos en relación con MG's.

Para la técnica de descomposición de MG's es suficiente que los lugares implícitos tengan una única transición de entrada y una única de salida. Por ello se denotarán como TT-MSIP's (MSIP's de transición a transición). De esta manera se logra que la red resultante de añadir estos TT-MSIP siga siendo un MG fuertemente conexo, vivo y con el mismo comportamiento que el original (sólo difieren en la codificación de los marcados alcanzables).

Las propiedades de los TT-MSIP's que se van a utilizar aquí se pueden demostrar razonando sobre caminos y ciclos simples (ver definición 2.41) en MG's. Todas las demostraciones están en [CCJS94].

Notación 3.1 [CCJS94] *Sea \mathcal{N} una P/T red y $x, y \in P \cup T$ dos nodos suyos. Se denota con $\mathcal{P}(x, y)$ al conjunto de caminos y ciclos simples de x a y (ver definición 2.41). La noción puede extenderse a conjuntos de nodos.*

$$\mathcal{P}(X, Y) = \bigcup_{\substack{x \in X \\ y \in Y}} \mathcal{P}(x, y).$$

El vector de incidencia de un TT-MSIP se puede obtener sumando los vectores de incidencia de los lugares de cualquier camino que una las transiciones de entrada y de salida del TT-MSIP. Se denotará con \mathbf{l}_p al vector de incidencia de un lugar p .

Teorema 3.2 [CCJS94] *Sea $\mathcal{N} = \langle P, T, F \rangle$ un MG fuertemente conexo y $p \notin P$ un lugar con $\bullet p = t_i$ y $p^\bullet = t_o$. Entonces p es un TT-MSIP y para todo $\pi \in \mathcal{P}(t_i, t_o)$ se tiene que $\mathbf{l}_p = \sum_{p_j \in \pi} \mathbf{l}_{p_j}$.*

En el caso de MG's se simplifica el problema de cálculo del marcado inicial necesario para que un TT-MSIP sea implícito. El siguiente teorema caracteriza este marcado mínimo, que se calcula a partir de los marcados de todos los caminos que unen la transición de entrada con la de salida del TT-MSIP.

Teorema 3.3 [CCJS94] *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un MG fuertemente conexo y vivo y $p \notin P$ un TT-MSIP con $\bullet p = t_i$ y $p^\bullet = t_o$. El mínimo marcado de p que lo hace implícito es:*

$$\mathbf{m}_0^{\min}[p] = \min \left\{ \sum_{p_j \in \pi} \mathbf{m}_0[p_j] \mid \pi \in \mathcal{P}(t_i, t_o) \right\}$$

Corolario 3.4 [CCJS94] *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un MG fuertemente conexo y vivo y $p \notin P$ un TT-MSIP con $\bullet p = t_i$ y $p^\bullet = t_o$. Entonces p es implícito para cualquier marcado inicial $\mathbf{m}_0[p] \geq \mathbf{m}_0^{\min}[p]$.*

El cálculo del marcado mínimo de un TT-MSIP para hacerlo implícito puede hacerse aplicando el algoritmo clásico de Floyd (ver [AHU83]) para el cálculo del camino de coste mínimo entre cualesquiera dos vértices de un grafo en el que cada arista tiene asociado un peso. En la siguiente sección se abordará este problema.

Para la reducción de MG's se calculan conjuntos de lugares implícitos que resumen el comportamiento de ciertas subredes del MG. Es interesante que el conjunto de lugares implícitos sea lo más pequeño posible para reducir la estructura de los sistemas agregados y por lo tanto la complejidad de su estudio. Para poder eliminar lugares implícitos que aportan la misma información que otros lugares implícitos del conjunto se necesita el siguiente teorema.

Teorema 3.5 *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un MG fuertemente conexo y vivo, t_i, t_j y t_k tres transiciones del mismo y p_{ik}, p_{kj} y p_{ij} tres TT-MSIP's de t_i a t_k , de t_k a t_j y de t_i a t_j respectivamente, con el marcado mínimo para hacerlos implícitos. Entonces, p_{ij} es implícito respecto a p_{ik} y p_{kj} (ver definición 2.62) si y sólo si existe un camino en $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ de t_i a t_j de peso mínimo pasando por t_k .*

Demostración:

Por ser p_{ij}, p_{ik} y p_{kj} TT-MSIP's, es evidente que el vector de incidencia de p_{ij} es la suma de los de p_{ik} y p_{kj} (por teorema 3.2). En estas condiciones p_{ij} es implícito respecto a p_{ik}, p_{kj} si y sólo si $\mathbf{m}_0[p_{ij}] \geq \mathbf{m}_0[p_{ik}] + \mathbf{m}_0[p_{kj}]$ (por definición 2.57).

\Rightarrow) Si p_{ik} es TT-MSIP con el marcado mínimo para hacerlo implícito, por el teorema 3.3, existe un camino $\pi_1 \in \mathcal{P}(t_i, t_k)$ tal que $\mathbf{m}_0[p_{ik}] = \sum_{p \in \pi_1} \mathbf{m}_0[p]$. Análogamente para p_{kj} , existe $\pi_2 \in \mathcal{P}(t_k, t_j)$ tal que $\mathbf{m}_0[p_{kj}] = \sum_{p \in \pi_2} \mathbf{m}_0[p]$. Sea π el camino formado por π_1 seguido de π_2 . Es claro que π pasa por t_k y $\sum_{p \in \pi} \mathbf{m}_0[p] = \mathbf{m}_0[p_{ik}] + \mathbf{m}_0[p_{kj}]$. Si p_{ij} es implícito respecto a p_{ik} y p_{kj} , entonces $\mathbf{m}_0[p_{ij}] \geq \mathbf{m}_0[p_{ik}] + \mathbf{m}_0[p_{kj}] = \sum_{p \in \pi} \mathbf{m}_0[p]$, luego π es un camino en $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ de t_i a t_j de peso mínimo que pasa por t_k .

\Leftarrow) Si existe π camino en $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ de t_i a t_j de peso mínimo que pasa por t_k , por teorema 3.3, $\mathbf{m}_0[p_{ij}] = \sum_{p \in \pi} \mathbf{m}_0[p]$. Sean π_1, π_2 los subcaminos de π que van de t_i a t_k y de t_k a t_j respectivamente. Por ser π camino de peso mínimo de t_i a t_j , π_1 y π_2 son caminos de peso mínimo de t_i a t_k y de t_k a t_j respectivamente. Por teorema 3.3, $\mathbf{m}_0[p_{ik}] = \sum_{p \in \pi_1} \mathbf{m}_0[p]$ y $\mathbf{m}_0[p_{kj}] = \sum_{p \in \pi_2} \mathbf{m}_0[p]$, es decir, $\mathbf{m}_0[p_{ij}] = \mathbf{m}_0[p_{ik}] + \mathbf{m}_0[p_{kj}]$. Luego p_{ij} es implícito respecto a p_{ik} y p_{kj} . \diamond

3.3 Descomposición estructural de grafos marcados

Una vez estudiadas las principales características de los lugares implícitos en la clase de los MG's, se va a desarrollar la técnica estructural de descomposición. En [CCJS94] la descomposición de MG's se realiza en dos subredes. Se aprovechará ahora para introducir las primeras generalizaciones de este capítulo. La primera consiste en descomponer el MG en un número finito arbitrario de subredes y la segunda en reducir la estructura de los sistemas agregados en cuanto al número de lugares implícitos necesarios para resumir el comportamiento de las subredes.

La idea básica es la siguiente. Un MG fuertemente conexo y vivo se corta en K subredes por medio de un *corte* B definido sobre lugares. A partir del corte y de las subredes, se construyen K sistemas de bajo nivel \mathcal{LS}_i con $1 \leq i \leq K$ y un sistema de alto nivel o esqueleto básico \mathcal{BS} . Cada \mathcal{LS}_i contiene completamente una de las subredes producto del corte y agregaciones del resto de las subredes. El \mathcal{BS} se compone de las agregaciones de todas las subredes. Las agregaciones de cada subred se consiguen por medio de TT-MSIP's, junto con un marcado inicial que los hace implícitos. Se puede demostrar que el comportamiento de los \mathcal{LS}_i y \mathcal{BS} en cuanto a marcados alcanzables y secuencias de disparo es una proyección exacta de los del MG original sobre los lugares y transiciones mantenidos en cada subsistema.

En [CCJS94] puede ocurrir que entre los lugares implícitos calculados en la reducción de las subredes haya algunos redundantes, es decir, que sean implícitos respecto a otros TT-MSIP's en los \mathcal{LS}_i y \mathcal{BS} y por tanto puedan ser eliminados. Aquí se desarrollará la técnica de manera diferente, eliminando los lugares implícitos redundantes.

Definición 3.6 Sea $\mathcal{N} = \langle P, T, F \rangle$ un MG fuertemente conexo. Un subconjunto de lugares $B \subseteq P$ se dice K -corte ($K \geq 2$) de \mathcal{N} si existen K subredes $\mathcal{N}_i = \langle P_i, T_i, F_i \rangle$ con $1 \leq i \leq K$ de \mathcal{N} tales que:

- i) $\bigcup_{i=1}^K T_i = T$ y $T_i \cap T_j = \emptyset$ para $1 \leq i, j \leq K, i \neq j$.
- ii) $P_i = \bullet T_i \cup T_i \bullet$ para $1 \leq i \leq K$.
- iii) $\bigcup_{i=1}^K P_i = P$ y $\bigcup_{i \neq j} (P_i \cap P_j) = B$ para $1 \leq i, j \leq K$.
- iv) $F_i = F \cap ((P_i \times T_i) \cup (T_i \times P_i))$ para $1 \leq i \leq K$.

Las transiciones del conjunto $TI = \bullet B \cup B \bullet$ se llaman transiciones de interfaz. El resto se llaman transiciones internas.

En la figura 3.1 puede verse el ejemplo de un MG en el que se ha definido un corte (lugares B_i) en dos subredes. La subred \mathcal{N}_i ($i = 1, 2$) producto del corte es la generada por los lugares etiquetados P_{ij} y las transiciones T_{ij} . En este caso la subred \mathcal{N}_1 es la de la izquierda y la subred \mathcal{N}_2 la de la derecha. Las transiciones internas son las blancas y las de interfaz las sombreadas.

Una vez definido un corte sobre un MG hay que construir los \mathcal{LS}_i y el \mathcal{BS} . Para realizar esta tarea se calcula, para cada subred producto del corte, un conjunto de lugares implícitos (concretamente TT-MSIP's) que

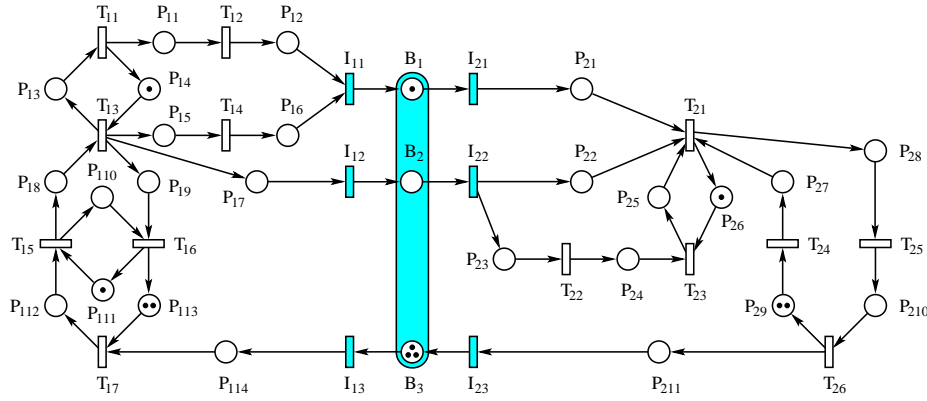


Figura 3.1: Corte en un grafo marcado.

resumen el comportamiento interno de la subred. Por el teorema 3.3, el marcado mínimo necesario para hacer implícitos a los TT-MSIP's se basa en el cálculo de caminos de marcado mínimo en el MG. Para realizar este cálculo se puede aplicar el algoritmo clásico de Floyd [AHU83] para el cálculo de los caminos de coste mínimo entre vértices de un grafo dirigido con pesos.

Algoritmo 3.7 Algoritmo de Floyd.

input: Matriz \mathbf{D} de pesos entre vértices de un grafo
 $\mathbf{K} := \mathbf{0}$ Matriz de vértices de los caminos mínimos
for $k := 1$ **to** n **do** n es el número de vértices del grafo
 for $i := 1$ **to** n **do**
 for $j := 1$ **to** n **do**
 if $\mathbf{D}(i, j) > \mathbf{D}(i, k) + \mathbf{D}(k, j)$ **then**
 $\mathbf{D}(i, j) := \mathbf{D}(i, k) + \mathbf{D}(k, j)$
 $\mathbf{K}(i, j) := k$
 end if
 end for
 end for
end for
output: Matrices \mathbf{D}, \mathbf{K}

El funcionamiento del algoritmo de Floyd es fácil de entender si se piensa que los pesos asociados a las aristas son distancias entre vértices. El algoritmo puede aplicarse a grafos con pesos positivos cualesquiera entre vértices

(e incluso puede haber pesos negativos siempre que no existan ciclos con suma de pesos negativa). Dado un grafo G dirigido con pesos, se denota con $V = 1, \dots, n$ al conjunto de vértices numerados y con A al conjunto de aristas. Como entrada para el algoritmo de Floyd se construye una matriz \mathbf{D} de tamaño $n \times n$ con los pesos asociados a cada arista de G . Si no existe una arista entre dos vértices i y j , entonces $\mathbf{D}(i, j) = \infty$. Tras la iteración k del bucle exterior, $\mathbf{D}(i, j)$ proporciona la distancia (peso) del camino mínimo del vértice i al j que está compuesto únicamente por vértices de índice menor o igual que k . Por lo tanto, al acabar el algoritmo, en \mathbf{D} se tienen las distancias (pesos) de los caminos mínimos entre cualesquiera dos vértices del grafo.

La matriz \mathbf{K} sirve para guardar de forma eficiente los caminos mínimos. Cada vez que se encuentra un camino de menor peso entre dos transiciones dadas (instrucciones del interior de los tres bucles encajados) se almacena en la matriz \mathbf{D} el nuevo peso mínimo y en la matriz \mathbf{K} el vértice por el que se unen los dos subcaminos que forman el nuevo camino mínimo. De esta forma, al finalizar el algoritmo, por medio de la matriz \mathbf{K} se pueden reconstruir de forma recursiva todos los caminos mínimos teniendo en cuenta el principio de optimalidad (si un camino entre dos transiciones dadas es mínimo, cualquier subcamino suyo también lo es). Si no se requiere conocer exactamente cuáles son los caminos mínimos se puede eliminar la matriz \mathbf{K} del algoritmo 3.7.

La complejidad en tiempo del algoritmo de Floyd es $O(n^3)$ y $O(n^2)$ en espacio, siendo n el número de vértices del grafo.

Los MG's pueden traducirse a grafos dirigidos con pesos de forma sencilla. Dado un MG, se puede construir un grafo con un vértice por cada transición y una arista dirigida uniendo dos vértices v_1 y v_2 del grafo si y sólo si existe un lugar en el MG que conecta la transición correspondiente a v_1 con la transición correspondiente a v_2 . El peso que se asigna a cada arista es el número de marcas en el mercado inicial del lugar que representa. Lo mismo se aplica a lugares que tienen la misma transición de entrada y de salida. Si una transición t_i del MG no tiene un lugar de entrada y salida a la vez debe ponerse inicialmente $\mathbf{D}(i, i) = \infty$.

Aplicando el algoritmo de Floyd a la matriz \mathbf{D} correspondiente a un MG definida de esta manera se obtiene, para cualesquiera dos transiciones de la subred, el número de marcas del camino de menos marcas que las une y por el teorema 3.3 este número es el mercado mínimo necesario para que un TT-MSIP que une las dos transiciones sea implícito. Si después de aplicar

el algoritmo de Floyd se obtiene una cierta entrada $\mathbf{D}(i, j) = \infty$, quiere decir que no hay ningún camino en la red que una la transición t_i con la transición t_j (situación que no es posible en MG's fuertemente conexos) y por lo tanto no hay que poner ningún lugar implícito de t_i a t_j . Por el contrario, si $\mathbf{D}(i, j) < \infty$, quiere decir que el camino de menos marcas en el marcado inicial que va de la transición t_i a la transición t_j tiene $\mathbf{D}(i, j)$ marcas.

Para la fase de reducción hay cuestiones particulares que hay que tener en cuenta. No interesan todos los caminos de menos marcas, sino sólo aquellos que conectan entre sí transiciones de interfaz de una subred ya que se pretende resumir el comportamiento de cada subred por separado. Serán los lugares implícitos asociados a estos caminos de peso mínimo entre transiciones de interfaz de una subred los que resuman el comportamiento de esa subred. La segunda cuestión es que se puede aprovechar el cálculo para reducir el número de lugares implícitos necesarios para resumir las subredes si se cambia ligeramente la función de coste. En efecto, en [CCJS94] se utiliza el algoritmo de Floyd para el cálculo de lugares implícitos que resumen el comportamiento de cada subred. Si se procede así, el número de lugares implícitos calculados para resumir el comportamiento de una subred puede ser mayor del necesario, es decir, algunos lugares implícitos pueden ser implícitos respecto a otros. Para asegurar que ningún lugar implícito sea a su vez implícito respecto a otros, se hará uso del teorema 3.5. Este teorema asegura que si un camino de peso mínimo de la transición de interfaz t_i a la transición de interfaz t_j pasa por otra transición de interfaz t_k , el lugar implícito correspondiente al camino de t_i a t_j es implícito respecto a los lugares correspondientes a los caminos de t_i a t_k y de t_k a t_j . En este caso el lugar implícito de t_i a t_j puede eliminarse ya que los lugares implícitos de t_i a t_k y de t_k a t_j resumen también el camino mínimo de t_i a t_j . Entonces interesa que de todos los caminos de peso mínimo que unen dos transiciones de interfaz, el algoritmo de Floyd se quede con alguno que pase por una transición de interfaz si existe. Esto asegura la eliminación de un lugar implícito al construir los sistemas agregados. Para conseguir este objetivo hay que modificar la función de coste. Y para poder demostrar la corrección del algoritmo modificado se empleará el marco general para la resolución de problemas de caminos en grafos dirigidos desarrollado en [CLR90].

La modificación de la función de coste es muy simple y la complejidad del algoritmo resultante la misma que la del algoritmo de Floyd, pero la demostración de la corrección es algo extensa. Por ello, primero se va a

dar la idea de la modificación, después la demostración de la corrección del algoritmo y por último una codificación eficiente de esta modificación.

La modificación consiste en sustituir los pesos de las aristas del grafo por etiquetas en las que se incluye una variable lógica que indique si el camino que tiene asociada esta etiqueta pasa por una transición de interfaz o no. Esto obliga a distinguir los vértices del grafo entre los que corresponden a transiciones de interfaz y los que no. Inicialmente las etiquetas de las aristas del grafo se corresponden con lugares del MG, por lo que sus etiquetas iniciales tienen esta variable lógica a 0 (un camino formado por dos transiciones y un lugar no puede pasar por una transición de interfaz). Posteriormente, según se vayan conectando aristas para formar caminos más largos, las variables lógicas de las etiquetas correspondientes a los caminos pueden tomar valor 1 en el momento en que se detecte que el camino formado pasa por un vértice correspondiente a una transición de interfaz. A la hora de seleccionar entre dos caminos el de peso mínimo, el primer criterio de selección consiste en elegir el de menor número de marcas y en caso de igualdad, se mira el indicador lógico. Si dos caminos tienen el mismo número de marcas, pero uno pasa por una transición de interfaz y el otro no, se coge como mínimo el que pasa por la transición de interfaz.

Para demostrar la existencia y corrección de un algoritmo que codifique esta idea se va a exponer a continuación el marco general para la resolución de problemas de caminos en grafos dirigidos propuesto en [CLR90].

La resolución de problemas de caminos en grafos dirigidos se basa en la estructura algebraica de semianillo cerrado. Aquí está su definición formal.

Definición 3.8 *Un semianillo cerrado es una 5-tupla $(S, \oplus, \odot, \bar{0}, \bar{1})$, donde S es un conjunto de elementos, \oplus (operador resumen) y \odot (operador extensión) operaciones binarias en S y $\bar{0}$ y $\bar{1}$ elementos de S que satisfacen las siguientes propiedades:*

i) $(S, \oplus, \bar{0})$ es un monoide, es decir:

- 1. \oplus es cerrada en S : $a \oplus b \in S$ para todo $a, b \in S$.*
- 2. \oplus es asociativa: $a \oplus (b \oplus c) = (a \oplus b) \oplus c$ para todo $a, b, c \in S$.*
- 3. $\bar{0}$ elemento neutro para \oplus : $a \oplus \bar{0} = \bar{0} \oplus a = a$ para todo $a \in S$.*

Análogamente, $(S, \odot, \bar{1})$ es un monoide.

ii) $\bar{0}$ es anulador para \odot : $a \odot \bar{0} = \bar{0} \odot a = \bar{0}$ para todo $a \in S$.

- ii) \oplus es conmutativa: $a \oplus b = b \oplus a$ para todo $a, b \in S$.
- iv) \oplus es idempotente: $a \oplus a = a$ para todo $a \in S$.
- v) \odot es distributiva respecto a \oplus : $a \odot (b \oplus c) = (a \odot b) \oplus (a \odot c)$ y $(b \oplus c) \odot a = (b \odot a) \oplus (c \odot a)$ para todo $a, b, c \in S$.
- vi) Si $\{a_i\}_{i=1}^{\infty}$ es una sucesión de elementos de S , $\bigoplus_{i=1}^{\infty} a_i$ está bien definida y en S .
- vii) Se cumplen las propiedades conmutativa, asociativa e idempotencia en el resumen de sucesiones, es decir, el resumen de una sucesión puede reescribirse como un resumen en el que cada elemento aparece una sola vez y es independiente del orden de operación.
- viii) El operador \odot es distributivo respecto de resúmenes de sucesiones: $a \odot (\bigoplus_{i=1}^{\infty} b_i) = \bigoplus_{i=1}^{\infty} (a \odot b_i)$ y $(\bigoplus_{i=1}^{\infty} b_i) \odot a = \bigoplus_{i=1}^{\infty} (b_i \odot a)$ para todo $a, b_i \in S$ con $i \in \mathbb{N}$.

Las propiedades de un semianillo cerrado están muy relacionadas con el cálculo de caminos en grafos dirigidos. Sea $G = (V, A)$ un grafo dirigido y $\lambda : V \times V \rightarrow S$ una función de etiquetado que asocia a cada par de vértices una etiqueta del dominio S . La etiqueta de una arista $(u, v) \in A$ del grafo se denota con $\lambda(u, v)$. Como λ está definida en $V \times V$, se toma $\lambda(u, v) = \bar{0}$ si (u, v) no es un arista del grafo.

El operador extensión \odot sirve para extender la noción de etiqueta a caminos. La etiqueta de un camino $p = \langle v_1, v_2, \dots, v_k \rangle$ es

$$\lambda(p) = \bigodot_{i=1}^{k-1} \lambda(v_i, v_{i+1}).$$

El elemento neutro $\bar{1}$ de \odot representa la etiqueta del camino vacío.

Como caso particular para explicar el marco general se tomará el algoritmo de Floyd. El dominio S es $\mathbb{R}^+ \cup \{\infty\}$ y $\lambda(i, j) = w_{ij}$ para todo $i, j \in V$. El operador extensión \odot es en este caso la suma aritmética $+$, y la etiqueta de un camino $p = \langle v_1, v_2, \dots, v_k \rangle$ es

$$\lambda(p) = \bigodot_{i=1}^{k-1} \lambda(v_i, v_{i+1}) = \sum_{i=1}^{k-1} w_{v_i, v_{i+1}} = w(p).$$

El papel del elemento neutro $\bar{1}$ de \odot lo cumple el 0, que es el elemento neutro de la suma aritmética $+$. Se denotará con ε al camino vacío, y su etiqueta es $\lambda(\varepsilon) = w(\varepsilon) = 0 = \bar{1}$.

La etiqueta de un camino que sea la concatenación de otros subcaminos debe ser independiente del orden en el que se vayan concatenando los subcaminos, de ahí que se exija asociatividad para el operador extensión \odot . La existencia de un elemento neutro para el operador extensión es necesaria porque si a un camino se le extiende con el camino vacío su etiqueta no debe variar.

El operador resumen \oplus se usa para resumir etiquetas de varios caminos, es decir, $\lambda(p_1) \oplus \lambda(p_2)$ resume las etiquetas de los caminos p_1 y p_2 . La semántica del resumen de etiquetas de caminos depende de la aplicación. En el algoritmo de Floyd, el operador resumen \oplus es el ínfimo (inf) y su interpretación es que el resumen de las etiquetas de dos caminos es la etiqueta del camino de menor peso.

El objetivo es calcular, para cada par de vértices de un grafo $i, j \in V$, el resumen de las etiquetas de todos los caminos que van de i a j :

$$l_{ij} = \bigoplus_{i \overset{p}{\rightsquigarrow} j} \lambda(p) \quad (3.1)$$

Es necesaria la conmutatividad y asociatividad del operador resumen porque el orden en que se resumen las etiquetas de un conjunto caminos no debe alterar el resultado. Al emplear $\bar{0}$ como etiqueta de los pares ordenados (u, v) que no son aristas del grafo, cualquier camino que intente extenderse por un par que no es arista del grafo debe tener etiqueta $\bar{0}$, es decir, $\bar{0}$ debe ser anulador para \odot .

En el algoritmo de Floyd, ∞ es el elemento neutro del operador resumen inf y es anulador para el operador extensión $+$: $a + \infty = \infty + a = \infty$ para todo $a \in \mathbb{R}^+ \cup \{\infty\}$.

Se necesita que el operador resumen sea idempotente porque al resumir un camino consigo mismo el resultado debe ser la misma etiqueta del camino.

Como en un grafo puede haber caminos que no sean simples, es posible que el número de caminos entre dos vértices sea infinito numerable (por ejemplo si hay ciclos). Entonces, el operador resumen \oplus debe poder aplicarse a sucesiones de caminos. Además, el resultado de este resumen no debe depender del orden en que se resumen. Por ello, hace falta asociatividad, conmutatividad e idempotencia de resúmenes de sucesiones.

En el caso del algoritmo de Floyd estas propiedades se cumplen debido a la existencia del ínfimo de cualquier sucesión de números reales acotada inferiormente (ver corolario 2.4).

Por lo que respecta a la distributividad del operador extensión \odot respecto del operador resumen \oplus , es necesaria para asegurar que las etiquetas de caminos que puedan tener una parte en común no varíe por la forma en que hayan sido calculadas. Si se tienen caminos $u \xrightarrow{p_1} v$, $v \xrightarrow{p_2} x$ y $v \xrightarrow{p_3} y$, es posible resumir los caminos $p_1 \circ p_2$ y $p_1 \circ p_3$ de dos formas que deben dar el mismo resultado: $\lambda(p_1) \odot (\lambda(p_2) \oplus \lambda(p_3))$ o bien $(\lambda(p_1) \odot \lambda(p_2)) \oplus (\lambda(p_1) \odot \lambda(p_3))$. Y como puede haber resúmenes de sucesiones de caminos, \odot también debe poseer la propiedad distributiva respecto a estos resúmenes de sucesiones (sirven por ejemplo para resumir caminos con ciclos, que pueden ser atravesados un número arbitrario de veces).

En este marco algebraico, existe un algoritmo de programación dinámica [CLR90] para el cálculo del resumen de las etiquetas de todos los caminos entre cualesquiera dos vértices de un grafo. En el algoritmo de Floyd, este resumen supone calcular, para cada par ordenado de vértices del grafo, el peso del camino de menor peso que los une.

Dados $i, j \in V$ se quiere calcular la expresión 3.1. l_{ij} es el resultado de resumir las etiquetas de todos los caminos de i a j . El algoritmo de programación dinámica es una generalización del algoritmo de Floyd y emplea el algoritmo de clausura transitiva. Sea $Q_{ij}^{(k)}$ el conjunto de caminos de i a j cuyos vértices intermedios están en el conjunto $\{1, 2, \dots, k\}$. Se define

$$l_{ij}^{(k)} = \bigoplus_{p \in Q_{ij}^{(k)}} \lambda(p).$$

Se pueden calcular recursivamente los $l_{ij}^{(k)}$ con la fórmula

$$l_{ij}^{(k)} = l_{ij}^{(k-1)} \oplus \left(l_{ik}^{(k-1)} \odot (l_{kk}^{(k-1)})^* \odot l_{kj}^{(k-1)} \right) \quad (3.2)$$

Esta ecuación es prácticamente la misma del algoritmo de Floyd pero con el factor adicional $(l_{kk}^{(k-1)})^*$. Este factor representa el resumen de todos los ciclos que pasan por el vértice k y que tienen todos sus vértices intermedios en el conjunto $\{1, 2, \dots, k-1\}$. En el algoritmo de Floyd este factor vale $0 = \bar{1}$ (el peso del ciclo vacío). El punto de partida para poder calcular la expresión recursiva es:

$$l_{ij}^{(0)} = \begin{cases} \lambda(i, j) & \text{si } i \neq j \\ \bar{1} \oplus \lambda(i, j) & \text{si } i = j \end{cases}$$

Este valor se puede entender de la siguiente manera. La etiqueta de un par ordenado de vértices (i, j) con $i \neq j$ es simplemente $\lambda(i, j)$ (que es igual a $\bar{0}$ si (i, j) no es una arista del grafo). Si $i = j$ se resume la etiqueta de la arista de i a i con la etiqueta del camino vacío.

El algoritmo de programación dinámica para calcular los valores $l_{ij}^{(k)}$ en orden creciente de k es el siguiente:

Algoritmo 3.9 Algoritmo general de resúmenes.

input: $G = (V, A)$ grafo dirigido, λ función de etiquetado

L matriz de etiquetas entre vértices del grafo

K matriz de vértices de paso de los caminos mínimos

$n :=$ cardinal de V

for $i := 1$ **to** n **do**

for $j := 1$ **to** n **do**

K $(i, j) := 0$

if $i = j$ **then**

L $(i, j) := \bar{1} \oplus \lambda(i, j)$

else

L $(i, j) := \lambda(i, j)$

end if

end for

end for

for $k := 1$ **to** n **do**

for $i := 1$ **to** n **do**

for $j := 1$ **to** n **do**

$c := \mathbf{L}(i, k) \odot \mathbf{L}(k, k)^* \odot \mathbf{L}(k, j)$

L $(i, j) := \mathbf{L}(i, j) \oplus c$

if $\mathbf{L}(i, j) = c$ **then** **K** $(i, j) := k$

end for

end for

end for

output: Matrices **L**, **K**

En este algoritmo, la matriz **K** se incluye por la misma razón que en el algoritmo de Floyd, es decir, para almacenar de forma eficiente cuáles

son los caminos mínimos de acuerdo con la función de coste. Si se observa el algoritmo y el tipo de operaciones que se hacen con las etiquetas, se pueden relajar algunas de las condiciones de semianillo cerrado expuestas en el marco general. En efecto, el resumen de caminos siempre se aplica a etiquetas de caminos con los mismos vértices origen y destino, o como mucho a las del camino vacío ($\bar{1}$) y/o al inexistente ($\bar{0}$). Por lo tanto, es suficiente que el operador resumen \oplus esté definido para las etiquetas de conjuntos de caminos con el mismo origen y destino junto con el camino vacío y/o el inexistente.

Definición 3.10 *Un semianillo cerrado en sentido laxo es un semianillo cerrado en el que el operador resumen \oplus se define únicamente para etiquetas de caminos con el mismo origen y destino junto con el camino vacío y/o el inexistente.*

Una vez explicado el marco general para la resolución de problemas de caminos en grafos dirigidos, se va a desarrollar ahora el algoritmo necesario para la reducción de MG's. Para ello es necesario definir un semianillo cerrado en sentido laxo de acuerdo con la definición 3.10.

Definición 3.11 *Sea $S = (\mathbb{Z}^+ \times \{0, 1\} \times \{0, 1\}) \cup \bar{0} \cup \bar{1}$ ($\bar{0}$ y $\bar{1}$ son dos elementos especiales). Se definen las siguientes operaciones en S :*

i) *Operador extensión \odot*

$$\begin{aligned} (n_1, b_1, v_1) \odot (n_2, b_2, v_2) &= (n_1 + n_2, \max\{b_1, b_2, v_1\}, v_2), \\ (n, b, v) \odot \bar{0} &= \bar{0} \odot (n, b, v) = \bar{0}, \quad (n, b, v) \odot \bar{1} = \bar{1} \odot (n, b, v) = (n, b, v), \\ \bar{0} \odot \bar{1} &= \bar{1} \odot \bar{0} = \bar{0}, \quad \bar{0} \odot \bar{0} = \bar{0} \quad \text{y} \quad \bar{1} \odot \bar{1} = \bar{1}. \end{aligned}$$

ii) *Operador resumen \oplus*

$$\begin{aligned} (n_1, b_1, v) \oplus (n_2, b_2, v) &= \begin{cases} (n_1, b_1, v) & \text{si } \begin{matrix} n_1 < n_2 & \text{o} \\ n_1 = n_2 & \text{y } b_1 \geq b_2 \end{matrix} \\ (n_2, b_2, v) & \text{si } \begin{matrix} n_1 > n_2 & \text{o} \\ n_1 = n_2 & \text{y } b_1 < b_2 \end{matrix} \end{cases} \\ (n, b, v) \oplus \bar{0} &= \bar{0} \oplus (n, b, v) = (n, b, v), \quad (n, b, v) \oplus \bar{1} = \bar{1} \oplus (n, b, v) = \bar{1}, \\ \bar{0} \oplus \bar{1} &= \bar{1} \oplus \bar{0} = \bar{1}, \quad \bar{0} \oplus \bar{0} = \bar{0} \quad \text{y} \quad \bar{1} \oplus \bar{1} = \bar{1} \end{aligned}$$

El dominio de las etiquetas S está formado por el conjunto de las ternas de enteros no negativos cuyas segunda y tercera componente son variables lógicas, es decir, enteros en $\{0, 1\}$, y dos elementos especiales $\bar{0}$ y $\bar{1}$. Los

elementos $\bar{0}$ y $\bar{1}$ no tienen expresión fácilmente interpretable en el conjunto $\mathbb{Z}^+ \times \{0, 1\} \times \{0, 1\}$. Por ello se les denota de forma diferente y se muestra cómo operan con el resto de elementos de S .

La interpretación de las etiquetas es sencilla. $\bar{0}$ es la etiqueta de los pares ordenados (u, v) de vértices que no son aristas del grafo y $\bar{1}$ es la etiqueta del camino vacío (como en el marco general). Por lo que respecta a los elementos de $\mathbb{Z}^+ \times \{0, 1\} \times \{0, 1\}$, la primera componente es el número de marcas del camino que lleva la etiqueta, la segunda es una variable lógica que indica si el camino pasa (1) o no (0) por una transición de interfaz (descontando las transiciones inicial y final), y la tercera es otra variable lógica que indica si el vértice final del camino corresponde (1) o no (0) a una transición de interfaz.

El operador extensión \odot calcula, a partir de las etiquetas (n_1, b_1, v_1) y (n_2, b_2, v_2) de un par ordenado de caminos, la etiqueta (n, b, v) del camino formado por la concatenación de los dos caminos manteniendo el orden. Por ello, el número de marcas del camino resultante es la suma de las marcas de los dos caminos, es decir, $n = n_1 + n_2$. El camino resultante pasa por una transición de interfaz si lo hacía cualquiera de los dos que lo componen o si el vértice por el que se unen corresponde a una transición de interfaz, es decir, $b = \max\{b_1, b_2, v_1\}$. Y el último vértice del camino resultante es el mismo que el último vértice del segundo camino, es decir, $v = v_2$. La etiqueta correspondiente a la extensión de cualquier camino con el camino vacío ($\bar{1}$) o el inexistente ($\bar{0}$) explica el resto de la definición del operador extensión.

El operador resumen \oplus está definido en sentido laxo (definición 3.10), es decir, sólo para etiquetas correspondientes a caminos con el mismo origen y destino junto con las del camino vacío y/o el inexistente. Por ello se exige que la tercera componente (tipo de vértice final del camino) sea la misma para todas las etiquetas diferentes de $\bar{0}$ y $\bar{1}$. Este operador resumen \oplus calcula, a partir de las etiquetas de dos caminos con el mismo vértice de origen y destino, la etiqueta del camino de menor número de marcas, y en caso de igualdad, la del que pase por una transición de interfaz, es decir, al operar con etiquetas se coge la etiqueta de menor primera componente (número de marcas del camino) y en caso de igualdad la de máxima segunda componente (indicador de paso por una transición de interfaz). El resumen de otras etiquetas con las del camino vacío o el inexistente son fáciles de explicar con esta interpretación.

Se va a demostrar ahora que el dominio de etiquetas con las operaciones

extensión y resumen así definidas tiene estructura de semianillo cerrado en sentido laxo.

Propiedad 3.12 *Sea $S = (\mathbb{Z}^+ \times \{0, 1\} \times \{0, 1\}) \cup \bar{0} \cup \bar{1}$ el dominio de etiquetas y \oplus y \odot los operadores extensión y resumen de la definición 3.11. Entonces $\langle S, \oplus, \odot, \bar{0}, \bar{1} \rangle$ tiene estructura de semianillo cerrado en sentido laxo (ver definición 3.10).*

Demostración:

i.1) $\langle S, \oplus, \bar{0} \rangle$ es un monoide.

\oplus es una operación interna en S por definición ya que el min es interna en \mathbb{Z}^+ y el max es interna en $\{0, 1\}$. Que $\bar{0}$ es elemento neutro para \oplus también es evidente por la definición. Sólo falta demostrar la propiedad asociativa. Sean $s_1, s_2, s_3 \in S$. Se distinguen tres casos:

Caso 1) Algún s_i es $\bar{0}$. Nuevamente hay tres casos. Aplicando que $\bar{0}$ es elemento neutro de \oplus , se tiene:

$$\bar{0} \oplus (s_2 \oplus s_3) = s_2 \oplus s_3 = (\bar{0} \oplus s_2) \oplus s_3.$$

$$s_1 \oplus (\bar{0} \oplus s_3) = s_1 \oplus s_3 = (s_1 \oplus \bar{0}) \oplus s_3.$$

$$s_1 \oplus (s_2 \oplus \bar{0}) = s_1 \oplus s_2 = (s_1 \oplus s_2) \oplus \bar{0}.$$

Caso 2) Ningún s_i es $\bar{0}$ y algún s_i es $\bar{1}$. Otros tres casos. Aplicando que $\bar{1}$ es elemento anulador de \oplus se tiene:

$$\bar{1} \oplus (s_2 \oplus s_3) = \bar{1} \quad \text{y} \quad (\bar{1} \oplus s_2) \oplus s_3 = \bar{1} \oplus s_3 = \bar{1}.$$

$$s_1 \oplus (\bar{1} \oplus s_3) = s_1 \oplus \bar{1} = \bar{1} \quad \text{y} \quad (s_1 \oplus \bar{1}) \oplus s_3 = \bar{1} \oplus s_3 = \bar{1}.$$

$$s_1 \oplus (s_2 \oplus \bar{1}) = s_1 \oplus \bar{1} = \bar{1} \quad \text{y} \quad (s_1 \oplus s_2) \oplus \bar{1} = \bar{1}.$$

Caso 3) Todos los s_i son distintos de $\bar{0}$ y $\bar{1}$. Este caso se demuestra teniendo en cuenta que el min en \mathbb{Z}^+ y el max en $\{0, 1\}$ son asociativos.

i.2) $\langle S, \odot, \bar{1} \rangle$ es un monoide.

\odot es una operación interna en S por definición ya que la suma aritmética $+$ es interna en \mathbb{Z}^+ y el max es interna en $\{0, 1\}$. Que $\bar{1}$ es elemento neutro para \odot también es evidente por la definición. Sólo falta demostrar la propiedad asociativa. Sean $s_1, s_2, s_3 \in S$. Se distinguen tres casos:

Caso 1) Algún s_i es $\bar{0}$. Aplicando que $\bar{0}$ es elemento anulador de \odot , se demuestra como el caso 2 del apartado *i.1*.

Caso 2) Ningún s_i es $\bar{0}$ y algún s_i es $\bar{1}$. Aplicando que $\bar{1}$ es elemento neutro de \odot , se demuestra igual que el caso 1 del apartado *i.1*.

Caso 3) Todos los s_i son distintos de $\bar{0}$ y $\bar{1}$. Este caso se demuestra teniendo en cuenta que la suma aritmética $+$ es asociativa en \mathbb{Z}^+ y el max es asociativo en $\{0, 1\}$.

ii) $\bar{0}$ es anulador de \odot . Por definición.

iii) \oplus es conmutativa. Por definición.

iv) \oplus es idempotente. Por definición.

v) \odot es distributiva respecto de \oplus . Sean $s_1, s_2, s_3 \in S$. Se distinguen tres casos:

Caso 1) Algún s_i es $\bar{0}$. Nuevamente hay tres casos:

$$\bar{0} \odot (s_2 \oplus s_3) = \bar{0} \quad \text{y} \quad (\bar{0} \odot s_2) \oplus (\bar{0} \odot s_3) = \bar{0} \oplus \bar{0} = \bar{0}.$$

$$(s_2 \oplus s_3) \odot \bar{0} = \bar{0} \quad \text{y} \quad (s_2 \odot \bar{0}) \oplus (s_3 \odot \bar{0}) = \bar{0} \oplus \bar{0} = \bar{0}.$$

$$s_1 \odot (\bar{0} \oplus s_3) = s_1 \odot s_3 \quad \text{y} \quad (s_1 \odot \bar{0}) \oplus (s_1 \odot s_3) = \bar{0} \oplus (s_1 \odot s_3) = s_1 \odot s_3.$$

$$(\bar{0} \oplus s_3) \odot s_1 = s_3 \odot s_1 \quad \text{y} \quad (\bar{0} \odot s_1) \oplus (s_3 \odot s_1) = \bar{0} \oplus (s_3 \odot s_1) = s_3 \odot s_1.$$

$$s_1 \odot (s_2 \oplus \bar{0}) = s_1 \odot s_2 \quad \text{y} \quad (s_1 \odot s_2) \oplus (s_1 \odot \bar{0}) = (s_1 \odot s_2) \oplus \bar{0} = s_1 \odot s_2.$$

$$(s_2 \oplus \bar{0}) \odot s_1 = s_2 \odot s_1 \quad \text{y} \quad (s_2 \odot s_1) \oplus (\bar{0} \odot s_1) = (s_2 \odot s_1) \oplus \bar{0} = s_2 \odot s_1.$$

Caso 2) Ningún s_i es $\bar{0}$ y algún s_i es $\bar{1}$. Otra vez hay tres casos:

$$\bar{1} \odot (s_2 \oplus s_3) = s_2 \oplus s_3 \quad \text{y} \quad (\bar{1} \odot s_2) \oplus (\bar{1} \odot s_3) = s_2 \oplus s_3.$$

$$(s_2 \oplus s_3) \odot \bar{1} = s_2 \oplus s_3 \quad \text{y} \quad (s_2 \odot \bar{1}) \oplus (s_3 \odot \bar{1}) = s_2 \oplus s_3.$$

$$s_1 \odot (\bar{1} \oplus s_3) = s_1 \odot \bar{1} = s_1 \quad \text{y} \quad (s_1 \odot \bar{1}) \oplus (s_1 \odot s_3) = s_1 \oplus (s_1 \odot s_3) = s_1. \text{ Esta}$$

última igualdad se debe a lo siguiente. Para que tenga sentido el resumen de s_1 con $s_1 \odot s_3$, los caminos correspondientes deben tener el mismo vértice origen y destino, es decir, el camino cuya etiqueta es s_3 debe ser un ciclo (con vértice de origen y destino el vértice destino del camino de etiqueta s_1).

Ahora, cualquier ciclo en un MG fuertemente conexo y vivo tiene marcas (ver teorema 2.90), luego el camino de etiqueta $s_1 \odot s_3$ tiene un número de marcas mayor que el camino de etiqueta s_1 .

$$(\bar{1} \oplus s_3) \odot s_1 = \bar{1} \odot s_1 = s_1 \quad \text{y} \quad (\bar{1} \odot s_1) \oplus (s_3 \odot s_1) = s_1 \oplus (s_3 \odot s_1) = s_1$$

por la misma razón.

$$s_1 \odot (s_2 \oplus \bar{1}) = s_1 \odot \bar{1} = s_1 \quad \text{y} \quad (s_1 \odot s_2) \oplus (s_1 \odot \bar{1}) = (s_1 \odot s_2) \oplus s_1 = s_1$$

por la misma razón.

$$(s_2 \oplus \bar{1}) \odot s_1 = \bar{1} \odot s_1 = s_1 \quad \text{y} \quad (s_2 \odot s_1) \oplus (\bar{1} \odot s_1) = (s_2 \odot s_1) \oplus s_1 = s_1$$

por la misma razón.

Caso 3) Todos los s_i son distintos de $\bar{0}$ y $\bar{1}$. Sean $s_i = (n_i, b_i, v_i)$ para $i = 1, 2, 3$. Deben ser $v_2 = v_3$ para que tengan sentido los resúmenes involucrados.

$$\begin{aligned} s_1 \odot (s_2 \oplus s_3) &= \begin{cases} s_1 \odot s_2 & \text{si } n_2 < n_3 \text{ o } (n_2 = n_3 \text{ y } b_2 \geq b_3) \\ s_1 \odot s_3 & \text{si } n_2 > n_3 \text{ o } (n_2 = n_3 \text{ y } b_2 < b_3) \end{cases} \\ &= \begin{cases} (n_1 + n_2, \max\{b_1, b_2, v_1\}, v_2) & \text{si } n_2 < n_3 \text{ o } (n_2 = n_3 \text{ y } b_2 \geq b_3) \\ (n_1 + n_3, \max\{b_1, b_3, v_1\}, v_3) & \text{si } n_2 > n_3 \text{ o } (n_2 = n_3 \text{ y } b_2 < b_3) \end{cases} \end{aligned}$$

Por otro lado:

$$\begin{aligned}
(s_1 \odot s_2) \oplus (s_1 \odot s_3) &= (n_1 + n_2, \max\{b_1, b_2, v_1\}, v_2) \oplus (n_1 + n_3, \max\{b_1, b_3, v_1\}, v_3) \\
&= \begin{cases} (n_1 + n_2, \max\{b_1, b_2, v_1\}, v_2) & \text{si } n_1 + n_2 < n_1 + n_3 \text{ o} \\ & n_1 + n_2 = n_1 + n_3 \text{ y } \max\{b_1, b_2, v_1\} \geq \max\{b_1, b_3, v_1\} \\ (n_1 + n_3, \max\{b_1, b_3, v_1\}, v_3) & \text{si } n_1 + n_2 > n_1 + n_3 \text{ o} \\ & n_1 + n_2 = n_1 + n_3 \text{ y } \max\{b_1, b_2, v_1\} < \max\{b_1, b_3, v_1\} \end{cases}
\end{aligned}$$

Hay que demostrar que para cualesquiera s_1, s_2, s_3 las dos expresiones coinciden. Es claro que $n_1 + n_2 < n_1 + n_3$ si y sólo si $n_2 < n_3$ y también $n_1 + n_2 = n_1 + n_3$ es equivalente a $n_2 = n_3$ para todo $n_1, n_2, n_3 \in \mathbb{Z}^+$ luego si $n_2 \neq n_3$ las expresiones coinciden. Si $n_2 = n_3$ y $b_2 \geq b_3$ entonces $n_1 + n_2 = n_1 + n_3$ y $\max\{b_1, b_2, v_1\} \geq \max\{b_1, b_3, v_1\}$ luego también coinciden las expresiones. Por último, si $n_2 = n_3$ y $b_2 < b_3$ entonces $n_1 + n_2 = n_1 + n_3$ y $\max\{b_1, b_2, v_1\} < \max\{b_1, b_3, v_1\}$. Si la última desigualdad es estricta, las expresiones coinciden y si $\max\{b_1, b_2, v_1\} = \max\{b_1, b_3, v_1\}$ entonces los dos casos de la segunda expresión son iguales al caso de la primera. Por lo tanto, $s_1 \odot (s_2 \oplus s_3) = (s_1 \odot s_2) \oplus (s_1 \odot s_3)$.

$$\begin{aligned}
(s_2 \oplus s_3) \odot s_1 &= \begin{cases} s_2 \odot s_1 & \text{si } n_2 < n_3 \text{ o } (n_2 = n_3 \text{ y } b_2 \geq b_3) \\ s_3 \odot s_1 & \text{si } n_2 > n_3 \text{ o } (n_2 = n_3 \text{ y } b_2 < b_3) \end{cases} \\
&= \begin{cases} (n_2 + n_1, \max\{b_2, b_1, v_2\}, v_1) & \text{si } n_2 < n_3 \text{ o } (n_2 = n_3 \text{ y } b_2 \geq b_3) \\ (n_3 + n_1, \max\{b_3, b_1, v_3\}, v_1) & \text{si } n_2 > n_3 \text{ o } (n_2 = n_3 \text{ y } b_2 < b_3) \end{cases}
\end{aligned}$$

Por otro lado:

$$\begin{aligned}
(s_2 \odot s_1) \oplus (s_3 \odot s_1) &= (n_2 + n_1, \max\{b_2, b_1, v_2\}, v_1) \oplus (n_3 + n_1, \max\{b_3, b_1, v_3\}, v_1) \\
&= \begin{cases} (n_2 + n_1, \max\{b_2, b_1, v_2\}, v_1) & \text{si } n_2 + n_1 < n_3 + n_1 \text{ o} \\ & n_2 + n_1 = n_3 + n_1 \text{ y } \max\{b_2, b_1, v_2\} \geq \max\{b_3, b_1, v_3\} \\ (n_3 + n_1, \max\{b_3, b_1, v_3\}, v_1) & \text{si } n_2 + n_1 > n_3 + n_1 \text{ o} \\ & n_2 + n_1 = n_3 + n_1 \text{ y } \max\{b_2, b_1, v_2\} < \max\{b_3, b_1, v_3\} \end{cases}
\end{aligned}$$

Hay que demostrar que para cualesquiera s_1, s_2, s_3 las dos expresiones coinciden. Es claro que $n_2 < n_3$ si y sólo si $n_2 + n_1 < n_3 + n_1$ y $n_2 = n_3$ si y sólo si $n_2 + n_1 = n_3 + n_1$ para todo $n_1, n_2, n_3 \in \mathbb{Z}^+$ luego si $n_2 \neq n_3$ las expresiones coinciden. Si $n_2 = n_3$ y $b_2 \geq b_3$ entonces $n_2 + n_1 = n_3 + n_1$ y como $v_2 = v_3$ entonces $\max\{b_2, b_1, v_2\} \geq \max\{b_3, b_1, v_3\}$ luego también coinciden las expresiones. Por último, si $n_2 = n_3$ y $b_2 < b_3$ entonces $n_2 + n_1 = n_3 + n_1$ y como $v_2 = v_3$ entonces $\max\{b_2, b_1, v_2\} < \max\{b_3, b_1, v_3\}$. Si la última desigualdad es estricta, las expresiones coinciden y si $\max\{b_2, b_1, v_2\} = \max\{b_3, b_1, v_3\}$ entonces los dos casos de la segunda expresión son iguales al caso de la primera. Por lo tanto, $(s_2 \oplus s_3) \odot s_1 = (s_2 \odot s_1) \oplus (s_3 \odot s_1)$.

vi) Si $\{s_i\}_{i=1}^{\infty}$ es una sucesión de etiquetas, $\bigoplus_{i=1}^{\infty} s_i$ está bien definida y en S : Si algún camino de la sucesión tiene etiqueta $\bar{0}$, se puede eliminar del resumen. Si alguno tiene etiqueta $\bar{1}$, el resultado del resumen es $\bar{1}$. En el

resto de casos, el resumen de sucesiones de caminos está bien definido por estar bien definido el min de una sucesión de enteros no negativos y el max de una sucesión de números en $\{0, 1\}$.

vii) Se cumplen las propiedades conmutativa, asociativa e idempotencia en el resumen de sucesiones: Si algún camino de la sucesión tiene etiqueta $\bar{0}$, se puede eliminar del resumen, lo que equivale a hacer un resumen con un único camino de etiqueta $\bar{0}$. Si alguno tiene etiqueta $\bar{1}$, el resultado del resumen es $\bar{1}$, lo que equivale a hacer un resumen con un único camino de etiqueta $\bar{1}$. En el resto de casos, debido a la conmutatividad e idempotencia del operador \oplus , las etiquetas de la forma (n, b, v) pueden conmutar con el resto de etiquetas y eliminarse las instancias repetidas.

viii) \odot es distributiva respecto de resúmenes de sucesiones. Sean $a, b_i \in S$ con $i \in \mathbb{N}$. Se distinguen tres casos:

Caso 1) Todos los $b_i = \bar{0}$. Entonces $a \odot (\bigoplus_{i=1}^{\infty} b_i) = a \odot (\bigoplus_{i=1}^{\infty} \bar{0}) = a \odot \bar{0} = \bar{0}$ y $\bigoplus_{i=1}^{\infty} (a \odot b_i) = \bigoplus_{i=1}^{\infty} (a \odot \bar{0}) = \bigoplus_{i=1}^{\infty} \bar{0} = \bar{0}$.

$(\bigoplus_{i=1}^{\infty} b_i) \odot a = (\bigoplus_{i=1}^{\infty} \bar{0}) \odot a = \bar{0} \odot a = \bar{0}$ y $\bigoplus_{i=1}^{\infty} (b_i \odot a) = \bigoplus_{i=1}^{\infty} (\bar{0} \odot a) = \bigoplus_{i=1}^{\infty} \bar{0} = \bar{0}$.

Caso 2) Algún $b_i = \bar{1}$. Por el apartado vii, se puede suponer que $b_1 = \bar{1}$ y $b_i \neq \bar{1}$ para todo $i > 1$. Entonces, $a \odot (\bigoplus_{i=1}^{\infty} b_i) = a \odot (\bar{1} \oplus (\bigoplus_{i=2}^{\infty} b_i)) = a \odot \bar{1} = a$ y $\bigoplus_{i=1}^{\infty} (a \odot b_i) = (a \odot \bar{1}) \oplus (\bigoplus_{i=2}^{\infty} (a \odot b_i)) = a \oplus (\bigoplus_{i=2}^{\infty} (a \odot b_i)) = a$ por la misma razón que en caso 2 del apartado v.

$(\bigoplus_{i=1}^{\infty} b_i) \odot a = (\bar{1} \oplus (\bigoplus_{i=2}^{\infty} b_i)) \odot a = \bar{1} \odot a = a$ y

$\bigoplus_{i=1}^{\infty} (b_i \odot a) = (\bar{1} \odot a) \oplus (\bigoplus_{i=2}^{\infty} (b_i \odot a)) = a \oplus (\bigoplus_{i=2}^{\infty} (b_i \odot a)) = a$ por la misma razón.

Caso 3) $b_i \neq \bar{1}$ para todo $i \in \mathbb{N}$ y algún $b_i \neq \bar{0}$. Por el apartado vii, se puede suponer también que los $b_i \neq \bar{0}$ para todo $i \in \mathbb{N}$. El apartado vii también permite reordenar los $b_i = (n_i, b_i, v_i)$ de forma que $n_i \leq n_{i+1}$ para todo $i \in \mathbb{N}$ y si $n_i = n_{i+1}$ para algún i entonces $b_i \geq b_{i+1}$. Todos los v_i con $i \in \mathbb{N}$ deben ser iguales para que tenga sentido el resumen. Después de esta reordenación se tiene: $a \odot (\bigoplus_{i=1}^{\infty} b_i) = a \odot b_1$ por definición de \oplus y $\bigoplus_{i=1}^{\infty} (a \odot b_i) = (a \odot b_1)$ ya que si $a = (n, b, v)$ entonces $n + n_i \leq n + n_{i+1}$ para todo $i \in \mathbb{N}$ y si $n + n_i = n + n_{i+1}$ para algún $i \in \mathbb{N}$ entonces $\max\{b, b_i, v\} \geq \max\{b, b_{i+1}, v\}$.

Por otra parte, por definición de \oplus , se tiene que $(\bigoplus_{i=1}^{\infty} b_i) \odot a = b_1 \odot a$ y $\bigoplus_{i=1}^{\infty} (b_i \odot a) = (b_1 \odot a)$ ya que, por la reordenación anterior, se tiene que $n_i + n \leq n_{i+1} + n$ para todo $i \in \mathbb{N}$ y si $n_i + n = n_{i+1} + n$ para algún $i \in \mathbb{N}$ entonces $\max\{b_i, b, v_i\} \geq \max\{b_{i+1}, b, v_{i+1}\}$ ya que $v_i = v_{i+1}$ y $b_i \geq b_{i+1}$. \diamond

Una vez demostrada la estructura de semianillo cerrado en sentido laxo

para el dominio de etiquetas S con las operaciones \oplus y \odot , siguiendo el marco general para la resolución de problemas de caminos en grafos dirigidos de [CLR90], el algoritmo 3.9 con los operadores \oplus y \odot de la definición 3.11 permite calcular el resumen de todos los caminos que unen dos vértices cualesquiera del grafo inicial. Ahora se expone un algoritmo eficiente para el cálculo de marcados mínimos entre transiciones de un MG's.

Algoritmo 3.13 Algoritmo de cálculo de TT-MSIP's.

input: $G := (V, A)$, \mathbf{m}_0 grafo dirigido asociado al MG y marcado inicial

L Matriz de marcas entre vértices del grafo

B Matriz de indicadores de paso por transiciones de interfaz

n := número de transiciones del MG

n' := número de transiciones internas del MG

for $i := 1$ **to** n **do**

for $j := 1$ **to** n **do**

$\mathbf{B}(i, j) := 0$

if $i = j$ **then** $\mathbf{L}(i, j) := 0$; $\mathbf{B}(i, j) = 1$

else if $(i, j) \in A$ **then** $\mathbf{L}(i, j) := \mathbf{m}_0[p_{ij}]$

else $\mathbf{L}(i, j) := \infty$

end if

end for

end for

for $k := 1$ **to** n **do**

for $i := 1$ **to** n , $i \neq k$ **do**

for $j := 1$ **to** n , $j \neq k$ **do**

if $\mathbf{L}(i, j) > \mathbf{L}(i, k) + \mathbf{L}(k, j)$ **then** $\mathbf{L}(i, j) := \mathbf{L}(i, k) + \mathbf{L}(k, j)$

if $k > n'$ **then** $\mathbf{B}(i, j) := 1$

else $\mathbf{B}(i, j) := \max\{\mathbf{B}(i, k), \mathbf{B}(k, j)\}$

end if

else if $\mathbf{L}(i, j) = \mathbf{L}(i, k) + \mathbf{L}(k, j)$ **then**

if $k > n'$ **then** $\mathbf{B}(i, j) := 1$

else $\mathbf{B}(i, j) := \max\{\mathbf{B}(i, j), \mathbf{B}(i, k), \mathbf{B}(k, j)\}$

end if

end if

end for

end for

end for

output: Matrices **L**, **B**

A la hora de codificar el algoritmo en la práctica, no es necesario mantener las etiquetas de tres componentes como en el desarrollo teórico, sino únicamente el número de marcas del camino mínimo entre cada dos transiciones y una variable lógica que indique si el camino mínimo pasa o no por una transición de interfaz. Como no es necesario conocer explícitamente cuáles son los caminos mínimos, se puede eliminar del algoritmo la matriz \mathbf{K} del algoritmo general de resúmenes (ver algoritmo 3.9).

Se denota con n al número de transiciones del MG y con n' al número de transiciones internas. Se numeran las transiciones de forma que primero aparezcan las transiciones internas. De esta forma se tiene un test inmediato de si una transición es o no de interfaz. Además, con esta numeración se pueden obtener de forma eficiente todos los marcados mínimos entre transiciones de interfaz al final de la ejecución del algoritmo, ya que todas estas transiciones aparecen de forma consecutiva en la numeración.

La entrada de este algoritmo consta de la descripción de la estructura y marcado inicial del MG por medio de su grafo dirigido asociado G y \mathbf{m}_0 . A partir de esta descripción, se construyen dos matrices \mathbf{L} y \mathbf{B} . La matriz \mathbf{L} sirve para almacenar el número de marcas de los caminos de peso mínimo en el marcado inicial. Inicialmente contiene entradas 0 para los elementos de la diagonal. Este 0 es el resultado correspondiente a la operación $\bar{1} \oplus \lambda(i, i)$ del algoritmo general de resúmenes. En este caso, el camino vacío tiene 0 marcas y cualquier camino con origen y destino el mismo vértice debe tener marcas (por el teorema 2.90 ya que el MG es vivo), por lo que el resumen es el de un camino con 0 marcas. Para elementos distintos de la diagonal, el valor inicial puede ser el número de marcas del lugar que une los vértices o ∞ si este lugar no existe (indicando que de momento el camino de peso mínimo es el camino inexistente).

Por lo que respecta a la matriz \mathbf{B} , contiene información sobre si los caminos de peso mínimo pasan (1) o no (0) por una transición de interfaz. Inicialmente todas sus entradas, excepto las de la diagonal, son 0 indicando que ningún camino tiene un vértice intermedio que sea una transición de interfaz, puesto que de momento los caminos considerados tienen sólo una transición origen y otra destino. Los elementos de la diagonal se inicializan a 1 para evitar que el algoritmo calcule un lugar implícito con una única transición de entrada y salida a la vez. Esos lugares no tienen sentido en la reducción de MG's.

El paso siguiente (los tres bucles encajados) consiste en realizar los cálculos de caminos de peso mínimo. Se eliminan de los bucles las com-

paraciones triviales en MG's, que son los casos en que $i = k$ o $j = k$. En el algoritmo general de resúmenes esta operación consiste en calcular la ecuación (3.2). En este caso, la clausura $(l_{kk}^{(k-1)})^*$ es $\bar{1} \oplus (\bigoplus_{i=1}^{\infty} (\bigodot_{j=1}^i l_{kk}^{(k-1)})) = \bar{1}$ por lo que puede desaparecer del resumen, simplificando la expresión. La expresión está codificada en las instrucciones del interior de los tres bucles encajados. Si el camino de peso mínimo del vértice i al k concatenado con el que va del vértice k al j tiene menos marcas ($\mathbf{L}(i, k) + \mathbf{L}(k, j)$) que el que va de i al j ($\mathbf{L}(i, j)$) se almacena este nuevo camino de peso mínimo (para ello se almacena en la matriz \mathbf{L} el nuevo número mínimo de marcas y en la matriz \mathbf{B} el indicador de paso por transición de interfaz). Si los dos caminos tienen el mismo número de marcas, entonces se comparan sus indicadores de paso por transición de interfaz, almacenándose el que sea mayor, es decir, si uno de los dos pasa por una transición de interfaz, el algoritmo se queda con él. Para calcular si un camino que es la concatenación de otros dos pasa o no por una transición de interfaz, basta tomar los indicadores de los dos subcaminos (elementos $\mathbf{B}(i, k)$ y $\mathbf{B}(k, j)$) y por último saber si el vértice k corresponde o no a una transición de interfaz (esto es inmediato realizando la comparación $k > n'$ debido a la numeración de las transiciones).

De esta manera, tras la iteración k del bucle exterior, $\mathbf{L}(i, j)$ proporciona el número de marcas del camino con menos marcas que une el vértice i con el j y que está compuesto únicamente por vértices de índice menor o igual que k . Por su parte, $\mathbf{B}(i, j)$ es (también tras la iteración k) una variable lógica (valor 0 ó 1) que indica si este camino de peso mínimo pasa o no por una transición de interfaz.

Al finalizar el algoritmo, se tiene en $\mathbf{L}(i, j)$ el número de marcas del camino de menos marcas que une el vértice i con el j y en $\mathbf{B}(i, j)$ un indicador de si existe un camino de entre los de peso mínimo que pase por una transición de interfaz.

La complejidad de este algoritmo es de $O(n^3)$ en tiempo y $O(n^2)$ en espacio, donde n es el número de transiciones del MG original. Esta complejidad es la misma que la del algoritmo de Floyd, por lo que el desarrollo teórico realizado no aumenta el coste computacional del algoritmo resultante. Por lo tanto, se ha desarrollado un algoritmo que permite calcular, para dos transiciones cualesquiera del MG original, el número de marcas del camino de menos marcas que las une, eligiendo de entre todos los caminos mínimos uno que pase por una transición de interfaz, si existe.

Notar que para la reducción de MG's sólo interesan los caminos mínimos entre transiciones de interfaz, es decir, los valores $\mathbf{L}(i, j)$ y $\mathbf{B}(i, j)$ para

$i, j > n'$.

Se tienen ya los elementos teóricos necesarios para realizar la descomposición estructural de MG's.

Definición 3.14 *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un MG fuertemente conexo y vivo, $B \subseteq P$ un K -corte de \mathcal{N} y $\mathcal{N}_i = \langle P_i, T_i, F_i \rangle$ con $1 \leq i \leq K$ las subredes producto de B . Sean \mathbf{L} y \mathbf{B} las matrices resultado de aplicar a $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ el algoritmo 3.13. Para cada $1 \leq i \leq K$ se construye el conjunto H_i de la siguiente manera:*

$$H_i = \{ p_{kj} \text{ lugar} \mid t_k, t_j \in T_i \cap \text{TI}, \mathbf{B}(k, j) = 0, \bullet p_{kj} = t_k, p_{kj} \bullet = t_j \text{ y } \mathbf{m}_0[p_{kj}] = \mathbf{L}(k, j) \}$$

Se define el conjunto H como $H = \bigcup_{i=1}^K H_i$.

El conjunto H_i es el conjunto de lugares TT-MSIP que resumen los caminos de peso mínimo entre transiciones de interfaz de la subred \mathcal{N}_i . El conjunto H es el conjunto unión de los H_i . Estos conjuntos se construyen directamente a partir de la salida del algoritmo 3.13 observando los valores de las matrices \mathbf{L} y \mathbf{B} entre transiciones de interfaz. Dadas dos transiciones de interfaz diferentes t_k y t_j , en este caso $\mathbf{L}(k, j) < \infty$ ya que el MG es fuertemente conexo y por lo tanto siempre existe un camino que une la transición t_k con la t_j . Si $\mathbf{B}(k, j) = 1$, quiere decir que existe un camino de t_k a t_j de peso mínimo que pasa por otra transición de interfaz t_l . Aplicando el teorema 3.5, el lugar implícito p_{kj} puede ser eliminado porque es implícito respecto a los lugares p_{il} y p_{lj} . Por ello el lugar p_{kj} no se incluye en el conjunto H .

Al construir los conjuntos H_i sólo interesan los lugares que conectan transiciones de interfaz de la misma subred. La razón es que si se toma el camino de peso mínimo entre dos transiciones de interfaz de distintas subredes hay dos posibilidades. La primera es que este camino de peso mínimo no pase por otra transición de interfaz. Entonces, el camino está formado únicamente por un lugar y las dos transiciones consideradas, es decir, el lugar es un lugar del corte que ya forma parte de la red original (no hace falta añadirlo). La segunda posibilidad es que el camino de peso mínimo pase por otra transición de interfaz. En este caso, el lugar que lo resume es implícito con respecto a otros lugares y puede ser eliminado del conjunto H .

En consecuencia, el conjunto H_i está formado únicamente por lugares TT-MSIP que conectan transiciones de interfaz de la subred \mathcal{N}_i y que no

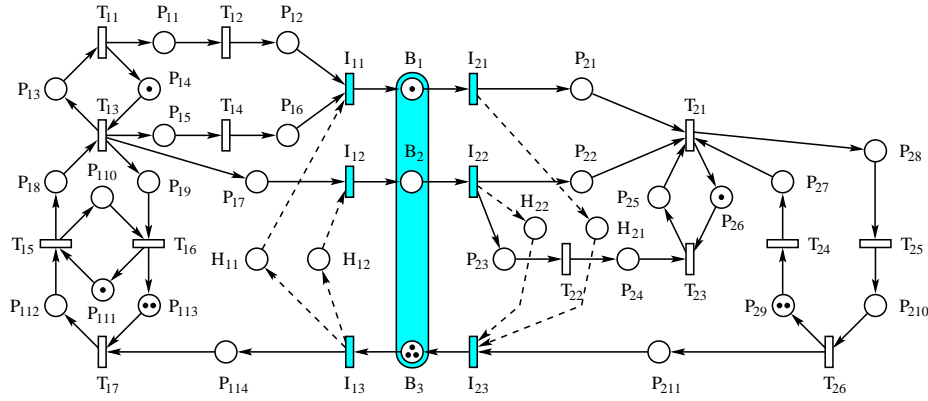


Figura 3.2: \mathcal{ES} del MG de la figura 3.1

son implícitos respecto a otros lugares de H_i . Estos lugares constituyen el resumen del comportamiento interno de la subred \mathcal{N}_i .

Ahora se pueden construir los sistemas agregados.

Definición 3.15 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un MG fuertemente conexo y vivo y $B \subseteq P$ un K -corte de \mathcal{N} . El sistema extendido $\mathcal{ES} = \langle \mathcal{EN}, \mathbf{m}_0^{\mathcal{EN}} \rangle$ de $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ es el sistema resultante de añadir a $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ los lugares del conjunto H con los arcos de entrada y salida y el marcado inicial de la definición 3.14.

En la figura 3.2 puede verse el \mathcal{ES} correspondiente al MG de la figura 3.1. Se han añadido en este caso 4 lugares implícitos (denotados con H_{ij} y arcos discontinuos) para resumir el comportamiento interno de las subredes.

Definición 3.16 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle P, T, F \rangle$ un MG fuertemente conexo y vivo, $B \subseteq P$ un K -corte de \mathcal{N} y \mathcal{ES} su sistema extendido.

- i) El sistema de bajo nivel $\mathcal{LS}_i = \langle \mathcal{LN}_i, \mathbf{m}_0^i \rangle$ con $1 \leq i \leq K$ es el sistema que se obtiene eliminando de \mathcal{ES} los lugares $\bigcup_{j \neq i} P_j \setminus B$ y las transiciones $\bigcup_{j \neq i} T_j \setminus \text{TI}$ y sus arcos adyacentes.
- ii) El sistema de alto nivel (o esqueleto básico) $\mathcal{BS} = \langle \mathcal{BN}, \mathbf{m}_0^{\mathcal{BN}} \rangle$ es el sistema que se obtiene eliminando de \mathcal{ES} los lugares $\bigcup_{i=1}^K P_i \setminus B$ y las transiciones $\bigcup_{i=1}^K T_i \setminus \text{TI}$ y sus arcos adyacentes.

En cada \mathcal{LS}_i , todas las subredes \mathcal{N}_j con $j \neq i$ se reducen a sus transiciones de interfaz y a los lugares implícitos del conjunto H_j , mientras que la subred \mathcal{N}_i se preserva completamente. En el \mathcal{BS} se reducen todas las subredes.

En la figura 3.3 pueden verse los sistemas agregados correspondientes al MG de la figura 3.1. En este caso hay dos sistemas de bajo nivel (\mathcal{LS}_1 y \mathcal{LS}_2) y uno de alto nivel (\mathcal{BS}). El \mathcal{LS}_1 preserva la subred \mathcal{N}_1 y el \mathcal{LS}_2 la subred \mathcal{N}_2 .

El siguiente teorema demuestra las propiedades que tienen los sistemas agregados recién construidos.

Teorema 3.17 *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un MG fuertemente conexo y vivo, $B \subseteq P$ un K -corte de \mathcal{N} . Entonces, \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} son MG's fuertemente conexos, vivos y cumplen:*

1. $L(\mathcal{S})|_{T_i \cup T_i} = L(\mathcal{LS}_i)$ para $1 \leq i \leq K$.
2. $L(\mathcal{S})|_{T_i} = L(\mathcal{BS})$.
3. $R(\mathcal{S})|_{P_i \cup B \cup H} = R(\mathcal{LS}_i)$ para $1 \leq i \leq K$.
4. $R(\mathcal{S})|_{B \cup H} = R(\mathcal{BS})$.

Demostración:

Los lugares del conjunto H tienen una única transición de entrada y una única de salida, por lo que el \mathcal{ES} es un MG. El \mathcal{ES} se forma a partir de \mathcal{S} añadiendo únicamente lugares implícitos, luego sigue siendo fuertemente conexo. Además, \mathcal{ES} es vivo porque $L(\mathcal{ES}) = L(\mathcal{S})$. A partir del \mathcal{ES} , para formar los sistemas agregados se eliminan nodos con todos sus arcos de entrada y salida, por lo que los lugares que quedan deben tener un único arco de entrada y uno solo de salida, es decir, son MG's. También es evidente por construcción que los sistemas agregados son fuertemente conexos ya que la subred de \mathcal{ES} generada por los lugares de $B \cup H$ y las transiciones de interfaz contiene los resúmenes de todos los caminos de peso mínimo entre transiciones de interfaz, en particular mantiene las conexiones entre transiciones de interfaz de \mathcal{N} . Por ello, al eliminar subredes para formar los sistemas agregados no se pierden conexiones entre los nodos que se preservan. Eliminar lugares para formar los sistemas agregados quita restricciones al disparo de las transiciones que quedan, por lo que siguen siendo vivos.

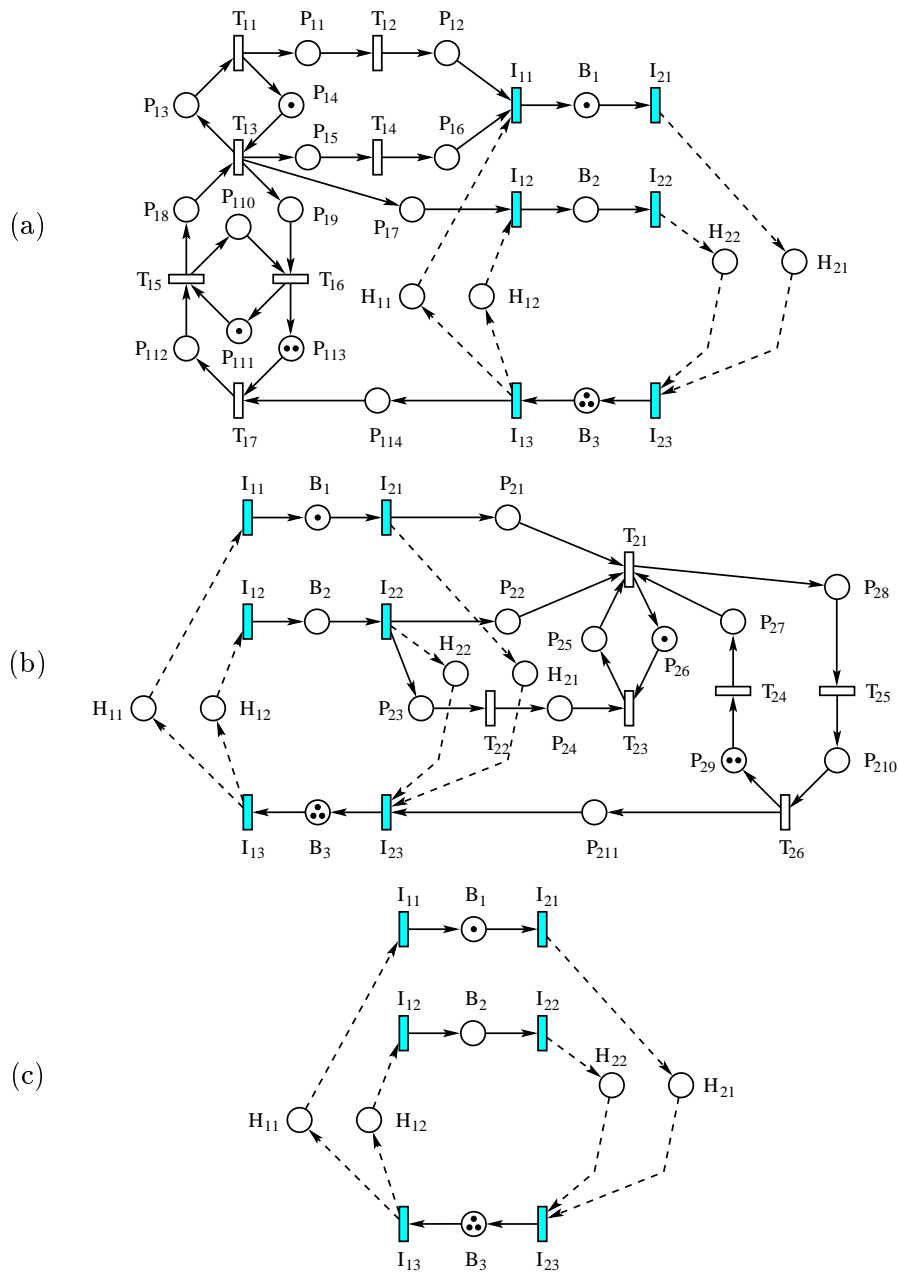


Figura 3.3: (a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del MG de la figura 3.1.

1) Es claro que $L(\mathcal{ES})|_{T_i \cup \text{TI}} \subseteq L(\mathcal{LS}_i)$, porque para construir el \mathcal{LS}_i a partir del \mathcal{ES} se eliminan transiciones y lugares internos a subredes, es decir, restricciones para el disparo de las transiciones de interfaz que quedan en las subredes.

Falta demostrar que $L(\mathcal{LS}_i) \subseteq L(\mathcal{ES})|_{T_i \cup \text{TI}}$. Se procederá por reducción al absurdo. Suponer que existe una secuencia de disparo $\sigma \in L(\mathcal{LS}_i)$ para la que no existe $\sigma' \in L(\mathcal{ES})$ tal que $\sigma'|_{T_i \cup \text{TI}} = \sigma$. Sea σ_0 el máximo prefijo de σ para el que existe $\sigma'_0 \in L(\mathcal{ES})$ verificando que $\sigma'_0|_{T_i \cup \text{TI}} = \sigma_0$. Entonces si $\mathbf{m}_0^{\mathcal{EN}} \xrightarrow{\sigma'_0} \mathbf{m}'$ y $\mathbf{m}_0^i \xrightarrow{\sigma_0} \mathbf{m}$, es claro que $\mathbf{m}'|_{P_i \cup H} = \mathbf{m}$. La transición t de σ que sigue a σ_0 debe ser una transición de alguna subred \mathcal{N}_j con $j \neq i$ porque éstas son las únicas que tienen restricciones de disparo adicionales en \mathcal{ES} . Estas restricciones adicionales vienen de los lugares P_j y no de los lugares implícitos H_j (por la definición 2.57 de lugar implícito). En \mathcal{ES} , a partir de \mathbf{m}' , cualquier secuencia de disparo σ'_1 formada únicamente por transiciones internas de \mathcal{N}_j no puede sensibilizar a la transición t ya que σ_0 es el máximo prefijo de σ para el que existe secuencia $\sigma'_0 \in L(\mathcal{ES})$ tal que $\sigma'_0|_{T_i \cup \text{TI}} = \sigma_0$. Sea \mathbf{m}'_1 un marcado alcanzable en $\langle \mathcal{EN}, \mathbf{m}' \rangle$ disparando una secuencia maximal σ'_1 de transiciones de $T_j \setminus \text{TI}$. En \mathbf{m}'_1 todas las transiciones de $T_j \setminus \text{TI}$ deben tener al menos un lugar de entrada sin marcas. Entonces, es posible construir un camino en \mathcal{N}_j uniendo dos transiciones de interfaz de \mathcal{N}_j que contiene a t y tal que todos sus lugares están sin marcas. Esto significa que los lugares implícitos que resumen este camino están desmarcados lo que contradice la hipótesis de que t es disparable en $\langle \mathcal{LN}_i, \mathbf{m} \rangle$.

2) Se demuestra igual que el apartado 1.

3) Por construcción del \mathcal{ES} , $R(\mathcal{ES})|_P = R(\mathcal{S})$. Ahora, $\mathbf{m} \in R(\mathcal{ES})$ si y sólo si existe $\sigma \in L(\mathcal{ES})$ tal que $\mathbf{m}_0^{\mathcal{EN}} \xrightarrow{\sigma} \mathbf{m}$. Por el apartado 1, esto ocurre si y sólo si $\sigma' = \sigma|_{T_i \cup \text{TI}} \in L(\mathcal{LS}_i)$. Sea $\mathbf{m}_0^i \xrightarrow{\sigma'} \mathbf{m}'$. Es evidente que $\mathbf{m}' \in R(\mathcal{LS}_i)$ (por construcción) y que $\mathbf{m}|_{P_i \cup B \cup H} = \mathbf{m}'$ ya que las transiciones del conjunto $T_i \cup \text{TI}$ sólo producen cambios en los lugares del conjunto $P_i \cup B \cup H$.

4) Se demuestra igual que el apartado 2. \diamond

A modo de resumen, en esta sección se ha desarrollado una técnica estructural eficiente que permite descomponer un MG fuertemente conexo y vivo en un número finito arbitrario K de subredes y a partir de ellas construir K sistemas de bajo nivel (en los que se mantiene una de las subredes y se reducen el resto) y un sistema de alto nivel o esqueleto básico (en el que se reducen todas las subredes). Estos $K + 1$ sistemas agregados son también MG's y mantienen las proyecciones de los estados alcanzables y de

las secuencias de disparo del sistema original sobre los nodos preservados en cada sistema agregado. Además, los sistemas agregados tienen la estructura más simple posible, en el sentido de que no se han empleado lugares implícitos redundantes en la reducción de las subredes producto del corte. Esta reducción de la estructura de los sistemas agregados se ha conseguido sin incrementar la complejidad de los algoritmos de cálculo.

El siguiente paso consiste en emplear estos sistemas agregados para aproximar el throughput de las transiciones del MG original. Este será el trabajo que se desarrolla en la siguiente sección.

3.4 Aproximación iterativa del throughput

En la sección anterior se ha desarrollado una técnica para descomponer un MG fuertemente conexo y vivo en K subredes, a partir de las cuales se construyen K sistemas de bajo nivel y uno de alto nivel o esqueleto básico. Cada sistema de bajo nivel contiene una de las subredes producto del corte, los lugares del corte, las transiciones de interfaz y los lugares implícitos que resumen el comportamiento del resto de subredes. El sistema de alto nivel contiene los lugares del corte, las transiciones de interfaz y los lugares implícitos que resumen el comportamiento de todas las subredes. Ahora se van a emplear estos $K + 1$ sistemas agregados para aproximar el throughput de las transiciones del MG original.

En [ABS84] se desarrolla una técnica de preservación del tiempo de respuesta en redes de colas. La técnica que se va a exponer en esta sección, que es básicamente la misma de [CCJS94], es un método de *aproximación del tiempo de respuesta*, es decir, es una variante aproximada de la primera. Las transiciones de interfaz de cada subred reducida en cada sistema agregado aproximarán el tiempo de respuesta de toda la subred. En esta memoria se incluyen dos generalizaciones; la adaptación del método para más de dos sistemas de bajo nivel y resultados parciales relativos a la convergencia del método.

Para aproximar el throughput, primero se intentó elaborar un método directo que asignaba tasas constantes a cada transición de interfaz, pero se conseguían malas aproximaciones. Por ello, se desarrolló un método iterativo en el que se iban actualizando las tasas de las transiciones de interfaz en las sucesivas iteraciones del método, hasta lograr la convergencia. El primer intento ([CCJS94]) se hizo sin la inclusión en el método del sistema de alto

nivel, lo cual dio lugar a problemas de convergencia. La explicación la da el hecho de que es posible calcular distintos valores para las tasas de las transiciones de interfaz (por ejemplo acelerando la parte reducida), de forma que los throughput de todos los sistemas de alto nivel sean iguales entre sí, pero diferentes al del sistema original. En ese caso, las tasas de las transiciones de interfaz no tendrán relación con el tiempo de respuesta de la subred que resumen y esto da problemas de convergencia en los métodos. Por ello, se decidió incluir en el método iterativo el sistema de alto nivel. El objetivo de este sistema es cruzar la información relativa a las distintas subredes producto del corte que se va obteniendo del estudio de los sistemas de alto nivel. El algoritmo utilizado es el siguiente:

Algoritmo 3.18 [CCJS94]

input: $\langle \mathcal{S}, w \rangle$ MG estocástico fuertemente conexo y vivo
seleccionar un K -corte $B \subseteq P$ de \mathcal{N}
construir \mathcal{LS}_i y \mathcal{BS} para $1 \leq i \leq K$
seleccionar tasas iniciales $\mu_t^{(0)}$ para cada $t \in T_j \cap \text{TI}$ con $2 \leq j \leq K$
 $n := 0$ (contador de iteraciones)
repeat
 $n := n + 1$
 for $i := 1$ **to** K **do**
 solve \mathcal{LS}_i **con:**
 In: tasas $\mu_l^{(n)}$ para las transiciones de $T_l \cap \text{TI}$ con $1 \leq l < i$
 tasas $\mu_l^{(n-1)}$ para las transiciones de $T_l \cap \text{TI}$ con $i < l \leq K$
 Out: proporciones μ_i entre tasas de las transiciones de $T_i \cap \text{TI}$ y
 throughput $\chi_i^{(n)}$ de las transiciones de $T_i \cap \text{TI}$
 solve \mathcal{BS} (encontrar factor de escala λ) **con:**
 In: tasas $\mu_l^{(n)}$ para las transiciones de $T_l \cap \text{TI}$ con $1 \leq l < i$
 tasas $\mu_l^{(n-1)}$ para las transiciones de $T_l \cap \text{TI}$ con $i < l \leq K$
 proporciones entre tasas μ_i para las transiciones de $T_i \cap \text{TI}$
 throughput $\chi_i^{(n)}$ de las transiciones de $T_i \cap \text{TI}$
 Out: tasas reales $\mu_i^{(n)} = \lambda \cdot \mu_i$ de las transiciones de $T_i \cap \text{TI}$
 tales que $\chi_{\mathcal{BS}}^{(n)} = \chi_i^{(n)}$
 end for
 until convergencia de $\{\chi_i^{(n)}\}_{i=1}^K$
output: throughput $\chi_1^{(n)}$ de las transiciones de $\langle \mathcal{S}, w \rangle$

En este algoritmo iterativo se van resolviendo uno por uno todos los sistemas de bajo nivel \mathcal{LS}_i (con $1 \leq i \leq K$). Cuando en la iteración n del esquema iterativo se resuelve el \mathcal{LS}_i , se obtienen para las transiciones de $T_i \cap \text{TI}$ los vectores $\chi_i^{(n)}$ de throughput y μ_i de proporciones entre tasas de servicio. Con esta información se busca, por medio del \mathcal{BS} , un factor de escala λ para las proporciones entre las tasas de disparo, de forma que el throughput de \mathcal{BS} y el de \mathcal{LS}_i coincidan. El cálculo de este factor de escala puede realizarse por medio de una búsqueda lineal en \mathcal{BS} . Se va resolviendo el \mathcal{BS} cambiando todas las tasas de las transiciones de $T_i \cap \text{TI}$ de forma que se mantengan las proporciones μ_i hasta que se consiga el mismo throughput en \mathcal{BS} y en \mathcal{LS}_i . Notar que \mathcal{BS} tiene muchos menos estados que el sistema original e incluso que los \mathcal{LS}_i , por lo que el coste de esta búsqueda lineal es despreciable comparado con la complejidad del estudio de los otros sistemas.

Después de este doble paso se obtiene una nueva aproximación $\mu_i^{(n)}$ para las tasas de disparo de las transiciones de $T_i \cap \text{TI}$ que se mantienen en el estudio posterior de otros \mathcal{LS}_j y en \mathcal{BS} .

Quedan por aclarar algunos puntos de este algoritmo iterativo. Por un lado, cuestiones respecto a las restricciones del corte. Por otro lado, la selección de las tasas de disparo iniciales $\mu_j^{(0)}$ de las transiciones de $T_j \cap \text{TI}$ con $2 \leq j \leq K$, que son necesarias para resolver por primera vez el sistema \mathcal{LS}_1 e iniciar el algoritmo iterativo. Otra cuestión a tener en cuenta es la forma de calcular, al resolver el \mathcal{LS}_i , las proporciones μ_i entre las tasas de disparo de las transiciones de $T_i \cap \text{TI}$. Por último, consideraciones acerca de la búsqueda lineal del factor de escala λ en \mathcal{BS} para el cálculo de las tasas de disparo reales de las transiciones de interfaz en cada iteración y el test de convergencia para decidir el final del algoritmo iterativo.

Respecto a las restricciones sobre el corte, en la sección anterior se podía definir sobre cualquier conjunto de lugares que induzca una partición en el conjunto de transiciones y de lugares (excluidos los del corte) del MG original. Si se tienen en cuenta ahora las cuestiones numéricas, no es deseable que aparezca ninguna transición de interfaz que sea inmediata. En este caso podrían aparecer errores de división por cero al estimar las proporciones entre las tasas de disparo de las transiciones de interfaz como se verá en el párrafo dedicado a la estimación de estas proporciones.

Respecto a la selección de las tasas iniciales para las transiciones de interfaz, el algoritmo iterativo no se ve afectado, en lo que respecta a la aproximación del throughput obtenida, por la selección de estos valores.

Por lo tanto, en principio se pueden seleccionar valores arbitrarios. Pero es razonable poner unos valores iniciales que tengan relación con el modelo. Una selección adecuada consiste en poner a cada transición de interfaz la tasa que tiene en el modelo original. De esta forma se reduce el número de iteraciones necesarias para alcanzar la convergencia del algoritmo. Esta elección dará sistemas agregados más rápidos en la primera iteración del método pero se corrige en las siguientes.

Por lo que respecta a la estimación de las proporciones μ_i entre las tasas de disparo de las transiciones de interfaz de la subred \mathcal{N}_i , se utilizan los lugares implícitos calculados en la fase de descomposición estructural. Como se explicó en la sección anterior, para cada subred \mathcal{N}_i producto del corte del MG original se calculaba un conjunto H_i de lugares TT-MSIP que resumen su comportamiento interno. Son estos lugares los que se emplean para realizar la estimación. El sistema de bajo nivel \mathcal{LS}_i contiene completamente la subred \mathcal{N}_i y los lugares del conjunto H_i . Por lo tanto, al resolver su CTMC isomorfa, se puede calcular la utilización de cada transición de interfaz si sólo se tienen en cuenta los lugares del conjunto $H_i \cup B$ (es decir, se calcula la probabilidad de que la transición esté sensibilizada una vez se eliminan los nodos internos de la subred \mathcal{N}_i). De esta manera, se está relacionando el funcionamiento completo de la subred \mathcal{N}_i con el de su resumen para poder exportar esta información al resto de sistemas de bajo nivel. Posteriormente, basta emplear la fórmula del cálculo del throughput del teorema 2.87 para encontrar las proporciones entre las tasas de disparo de las transiciones de interfaz. En la iteración n del algoritmo, la tasa proporcional de disparo de la transición $t \in T_i \cap \text{TI}$ es:

$$\mu_i[t] = \frac{\chi_i^{(n)}[t]}{P\{\mathbf{m}[p] > 0 \quad \forall p \in \bullet t \cap (H_i \cup B)\}}$$

Esta expresión es válida si se emplea semántica de un solo servidor. Si se emplea semántica de infinitos servidores, hay que sustituir la utilización de las transiciones por su grado medio de sensibilización, es decir, la tasa proporcional de disparo es en este caso:

$$\mu_i[t] = \frac{\chi_i^{(n)}[t]}{E[\min\{\mathbf{m}[p] \quad \forall p \in \bullet t \cap (H_i \cup B)\}]}$$

En cuanto al algoritmo de búsqueda lineal del factor de escala en \mathcal{BS} para calcular las tasas reales de disparo de las transiciones de interfaz, hay que te-

ner en cuenta que a lo largo del algoritmo iterativo se va modificando el valor de las tasas de las transiciones de interfaz de los \mathcal{LS}_i y de \mathcal{BS} . El throughput de estos sistemas es una función racional (cociente de polinomios) con respecto a estas variables. Como se han calculado unas proporciones entre las tasas de disparo que se quieren mantener en \mathcal{BS} , si se quiere calcular un factor de escala λ de forma que \mathcal{BS} tenga el mismo throughput que \mathcal{LS}_i , lo que se está haciendo en el fondo es plantear una ecuación racional de la forma:

$$\chi_{\mathcal{BS}}^{(n)}(\lambda) = \chi_{\mathcal{LS}_i}^{(n)}$$

En esta ecuación, $\chi_{\mathcal{BS}}^{(n)}(\lambda)$ es la función racional en λ (no se conoce explícitamente) y $\chi_{\mathcal{LS}_i}^{(n)}$ es el throughput de \mathcal{LS}_i calculado justo antes de empezar el estudio en \mathcal{BS} . Este problema numérico puede resolverse con algoritmos como el de la secante.

Por último, queda por aclarar el test de convergencia. Este algoritmo numérico calcula unas aproximaciones del throughput aceptables para una técnica de aproximación (por debajo del 5%). Teniendo en cuenta la aproximación que se va a obtener al final, no es necesario que el test de convergencia sea muy exigente ya que esto provocaría el cálculo de un mayor número de iteraciones sin que por ello se aumente la precisión de los resultados obtenidos. Si se pone un test de convergencia poco exigente, entonces el algoritmo finalizará antes de lo debido, aumentando en general el error de la aproximación. El test de convergencia que se utiliza consiste en iterar hasta que la diferencia entre los throughput de todos los \mathcal{LS}_i en dos iteraciones consecutivas del método sea menor que el 0.1%. Con este test el algoritmo converge (en todas las pruebas realizadas) en muy pocas iteraciones (entre 3 y 5 iteraciones).

Una vez explicado el método iterativo que se emplea para calcular aproximaciones del throughput del sistema original, se van a demostrar algunas de sus características. La demostración formal de estas características se basa en algunos resultados de otros autores que se van a exponer ahora.

El primer resultado (ver [MT95]) establece, para cualquier SPN, la continuidad del vector π de probabilidades en estado estacionario con respecto a las entradas del generador infinitesimal \mathbf{Q} de la CTMC isomorfa.

Teorema 3.19 [MT95] *Sea $\langle \mathcal{S}, w \rangle$ una SPN, \mathbf{Q} el generador infinitesimal de su CTMC isomorfa y π el vector de probabilidades en estado estacionario. Los elementos de π son funciones continuas de las entradas de \mathbf{Q} .*

Como las entradas de \mathbf{Q} son funciones continuas de las tasas de disparo de las transiciones de la red, en particular, $\boldsymbol{\pi}$ es continuo con respecto a las tasas de las transiciones de la red.

Corolario 3.20 *Sea $\langle \mathcal{S}, w \rangle$ una SPN y $\boldsymbol{\pi}$ el vector de probabilidades en estado estacionario. Los elementos de $\boldsymbol{\pi}$ son funciones continuas de las tasas de disparo de las transiciones de \mathcal{N} .*

Otra consecuencia inmediata del teorema anterior es la continuidad respecto de las entradas de \mathbf{Q} de todos los los índices de prestaciones de $\langle \mathcal{S}, w \rangle$ que puedan definirse como funciones continuas de las entradas del vector $\boldsymbol{\pi}$, en particular el throughput de las transiciones.

Corolario 3.21 *Sea $\langle \mathcal{S}, w \rangle$ una SPN, \mathbf{Q} el generador infinitesimal de su CTMC isomorfa y $\boldsymbol{\pi}$ el vector de probabilidades en estado estacionario. Sea X un índice de prestaciones de la SPN que puede expresarse como $X = f(\boldsymbol{\pi})$ con f continua. Entonces X es función continua de las entradas de \mathbf{Q} y de las tasas de las transiciones de $\langle \mathcal{S}, w \rangle$. En particular, el throughput de cualquier transición de $\langle \mathcal{S}, w \rangle$ es una función continua de las entradas de \mathbf{Q} y de las tasas de las transiciones de $\langle \mathcal{S}, w \rangle$.*

Otros resultados que se necesitan para demostrar las propiedades del método iterativo son las referentes a monotonía de throughput en la clase de los MG's. Estos resultados pueden verse en [BL92].

Teorema 3.22 [BL92] Monotonía respecto a las tasas de disparo.
Sea $\langle \mathcal{S}, w \rangle$ un MG estocástico, fuertemente conexo y vivo y $t \in T$ una transición de \mathcal{N} de tasa λ variable. El throughput de $\langle \mathcal{S}, w \rangle$ en función de λ , $\chi_{\mathcal{S}}(\lambda)$ es una función monótona creciente, es decir, si $\lambda' \leq \lambda \Rightarrow \chi_{\mathcal{S}}(\lambda') \leq \chi_{\mathcal{S}}(\lambda)$.

Teorema 3.23 [BL92] Monotonía respecto a la topología.
Sean $\langle \mathcal{S}, w \rangle$ y $\langle \mathcal{S}', w' \rangle$ dos MG's estocásticos, fuertemente conexos y vivos tales que $P \subseteq P'$, $T \subseteq T'$ y $F \subseteq F'$. Suponer además que $\langle \mathcal{S}, w \rangle$ y $\langle \mathcal{S}', w' \rangle$ tienen el mismo marcado inicial y tasas de disparo en los lugares y transiciones comunes a las dos redes. En estas condiciones se tiene que $\chi_{\mathcal{S}} \geq \chi_{\mathcal{S}'}$.

Por último, se necesita un resultado sobre el cálculo de una cota superior para el throughput de las transiciones de un MG. El siguiente resultado es mucho más general.

Propiedad 3.24 [CS93] Cota superior de throughput.

Sea $\langle \mathcal{S}, w \rangle$ una SPN viva y limitada. Una cota inferior para el tiempo medio entre disparos $\Gamma^{(i)}$ de la transición t_i es el resultado del siguiente problema de programación lineal:

$$\begin{aligned} \Gamma^{(i)} \geq & \max \quad \mathbf{Y}^T \cdot \mathbf{Pre} \cdot \mathbf{D}^{(i)} \\ \text{sujeto a} & \quad \mathbf{Y}^T \cdot \mathbf{C} = \mathbf{0} \\ & \quad \mathbf{Y}^T \cdot \mathbf{m}_0 = 1 \\ & \quad \mathbf{Y} \geq \mathbf{0} \end{aligned}$$

Donde \mathbf{Pre} es la matriz de preincidencia de \mathcal{N} , \mathbf{C} es la matriz de incidencia de \mathcal{N} y $\mathbf{D}^{(i)}$ es el vector de demandas medias de servicio de las transiciones ($\mathbf{D}^{(i)}(t_j) = s_j v^{(i)}(t_j)$ con s_j el tiempo medio de servicio de la transición t_j y $v^{(i)}(t_j)$ la ratio de visita de la transición t_j normalizada para t_i , es decir, $v^{(i)}(t_i) = 1$).

A partir de una cota inferior del tiempo medio entre disparos de una transición se obtiene, calculando el inverso, una cota superior del throughput de la misma. La idea de la demostración de esta propiedad es la siguiente (se pone para dar una idea intuitiva del cálculo de estas cotas). Toda SPN viva y limitada puede descomponerse en un conjunto de p -componentes (máquinas de estados fuertemente conexas). Entonces, la red puede verse como un conjunto de p -componentes que se sincronizan a través de sus transiciones. El tiempo medio entre disparos de una determinada transición siempre será mayor o igual que el tiempo medio entre disparos de la transición en cualquiera de las p -componentes que la incluyen. Como toda p -componente define un p -semiflujo mínimo de \mathcal{N} , el tiempo medio entre disparos de la transición también será mayor o igual que el tiempo medio entre disparos de la transición en cualquier p -semiflujo mínimo de \mathcal{N} (de ahí el máximo sobre p -semiflujos que aparece en el problema de programación lineal). Para que la cota superior del throughput de una transición sea significativa, debe ser menor que ∞ , lo que equivale a que la solución del problema de programación lineal sea mayor que 0. Esto siempre es posible conseguirlo para cualquier transición por la que pase un ciclo con alguna transición temporizada. En este caso, como un ciclo en una red es un p -semiflujo y para un ciclo con una transición temporizada el tiempo medio entre disparos de cualquier transición es mayor que 0 se asegura que el problema de programación lineal tiene solución estrictamente positiva.

Para demostrar las propiedades del método iterativo conviene describir el algoritmo 3.18 por medio de ecuaciones y funciones. La forma más sencilla

de describir el algoritmo es fijando como variables las tasas de las transiciones de interfaz.

Notación 3.25 Sea $\langle \mathcal{S}, w \rangle$ un MG estocástico, fuertemente conexo y vivo y $B \subseteq P$ un K -corte de \mathcal{N} . Se utilizarán las siguientes notaciones:

- i) $n_i = |T_i \cap \text{TI}|$ con $1 \leq i \leq K$ es el número de transiciones de interfaz de la subred \mathcal{N}_i .
- ii) $M = \sum_{i=1}^K n_i$ es el número de transiciones de interfaz de \mathcal{N} .
- iii) I_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ es la transición de interfaz j de la subred \mathcal{N}_i .
- iv) λ_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ es la tasa de disparo de I_{ij} .
- v) $\boldsymbol{\lambda}_i = (\lambda_{i1}, \dots, \lambda_{in_i})$ con $1 \leq i \leq K$ es el vector de tasas de disparo de las transiciones de interfaz de \mathcal{N}_i .
- vi) $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_K)$ es el vector de tasas de disparo de todas las transiciones de interfaz.
- vii) χ_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ es el throughput de la transición I_{ij} .
- viii) $\boldsymbol{\chi}_i = (\chi_{i1}, \dots, \chi_{in_i})$ con $1 \leq i \leq K$ es el vector de throughput de las transiciones de interfaz de \mathcal{N}_i (en MG's todas estas transiciones tienen el mismo throughput).
- ix) μ_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ es la tasa proporcional de disparo de transición I_{ij} calculada en \mathcal{LS}_i .
- x) $\boldsymbol{\mu}_i = (\mu_{i1}, \dots, \mu_{in_i})$ con $1 \leq i \leq K$ es el vector de tasas proporcionales de disparo de las transiciones de interfaz de \mathcal{N}_i .
- xi) λ'_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ es la nueva tasa de disparo de la transición I_{ij} calculada en \mathcal{BS} .
- xii) $\boldsymbol{\lambda}'_i = (\lambda'_{i1}, \dots, \lambda'_{in_i})$ con $1 \leq i \leq K$ es el vector de nuevas tasas de disparo de las transiciones de $T_i \cap \text{TI}$ calculadas en \mathcal{BS} .
- xiii) $\boldsymbol{\lambda}^{(i)} = (\boldsymbol{\lambda}'_1, \dots, \boldsymbol{\lambda}'_i, \boldsymbol{\lambda}_{i+1}, \dots, \boldsymbol{\lambda}_K)$ con $1 \leq i \leq K$ es el vector intermedio de tasas de disparo de las transiciones de TI después del estudio de \mathcal{LS}_i y el correspondiente estudio de \mathcal{BS} .

Se va a dar una explicación de estas notaciones. Cada vez que se resuelve \mathcal{LS}_i con $1 \leq i \leq K$ se calcula, para cada transición I_{ij} con $1 \leq j \leq n_i$, su throughput χ_{ij} y su tasa relativa μ_{ij} . Agrupando en vectores los índices de prestaciones calculados en \mathcal{LS}_i para todas las transiciones de $T_i \cap \text{TI}$, se obtienen $\boldsymbol{\chi}_i$ y $\boldsymbol{\mu}_i$. El siguiente paso consiste en estudiar el \mathcal{BS} , para obtener las nuevas tasas de disparo λ'_{ij} de las transiciones de $T_i \cap \text{TI}$. En notación vectorial, se obtiene el vector $\boldsymbol{\lambda}'_i$. El siguiente paso consiste en estudiar el \mathcal{LS}_{i+1} . El vector de tasas de transiciones de interfaz que se emplea para el estudio de este sistema no es el vector $\boldsymbol{\lambda}$ porque ya han sido actualizadas las tasas de las transiciones de $T_j \cap \text{TI}$ con $j \leq i$. El vector de tasas es el $\boldsymbol{\lambda}^{(i)}$.

Ahora se van a dar las notaciones de las funciones que intervienen en el algoritmo iterativo.

Notación 3.26 Sea $\langle \mathcal{S}, w \rangle$ un MG estocástico, fuertemente conexo y vivo y $B \subseteq P$ un K -corte de \mathcal{N} . Se denotan a las siguientes funciones:

- i) $p_i : (\mathbb{R}^+)^M \longrightarrow (\mathbb{R}^+)^{n_i}$ con $1 \leq i \leq K$ tal que $p_i(\boldsymbol{\lambda}) = \boldsymbol{\lambda}_i$.
- ii) $g_i : (\mathbb{R}^+)^M \longrightarrow (\mathbb{R}^+)^{n_i}$ con $1 \leq i \leq K$ tal que $g_i(\boldsymbol{\lambda}) = \boldsymbol{\lambda}'_i$.
- iii) $G : (\mathbb{R}^+)^M \longrightarrow (\mathbb{R}^+)^M$ tal que $G = (g_1, \dots, g_K)$.
- iv) $\chi_{ij} : (\mathbb{R}^+)^M \longrightarrow \mathbb{R}^+$ con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ tal que $\chi_{ij}(\boldsymbol{\lambda})$ es el throughput en \mathcal{LS}_i de la transición I_{ij} si se ponen como tasas de disparo de las transiciones de $T_k \cap \text{TI}$ con $k \neq i$ las que marcan los vectores $\boldsymbol{\lambda}_k$ de $\boldsymbol{\lambda}$.
- v) $\mu_{ij} : (\mathbb{R}^+)^M \longrightarrow \mathbb{R}^+$ con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ tal que $\mu_{ij}(\boldsymbol{\lambda})$ es la tasa proporcional de disparo en \mathcal{LS}_i de la transición I_{ij} si se ponen como tasas de disparo de las transiciones de $T_k \cap \text{TI}$ con $k \neq i$ las que marcan los vectores $\boldsymbol{\lambda}_k$ de $\boldsymbol{\lambda}$.
- vi) $\mu_i : (\mathbb{R}^+)^M \longrightarrow (\mathbb{R}^+)^{n_i}$ con $1 \leq i \leq K$ tal que $\mu_i = (\mu_{i1}, \dots, \mu_{in_i})$.
- vii) $\chi_{\mathcal{BS}} : (\mathbb{R}^+)^M \setminus \{0\} \longrightarrow \mathbb{R}^+$ tal que $\chi_{\mathcal{BS}}(\boldsymbol{\lambda})$ es el throughput de las transiciones de \mathcal{BS} si se ponen como tasas de disparo de las transiciones de $T_i \cap \text{TI}$ las que marcan los vectores $\boldsymbol{\lambda}_i$ de $\boldsymbol{\lambda}$.
- viii) $\chi_{\mathcal{BS}_i}[\boldsymbol{\lambda}_1, \dots, \boldsymbol{\mu}_i, \dots, \boldsymbol{\lambda}_K] : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ con $1 \leq i \leq K$ tales que:
 $\chi_{\mathcal{BS}_i}[\boldsymbol{\lambda}_1, \dots, \boldsymbol{\mu}_i, \dots, \boldsymbol{\lambda}_K](\lambda) = \chi_{\mathcal{BS}}(\boldsymbol{\lambda}_1, \dots, \lambda \boldsymbol{\mu}_i, \dots, \boldsymbol{\lambda}_K)$.

Se va a dar una explicación de estas notaciones. La función p_i con $1 \leq i \leq K$ extrae el vector λ_i de tasas de las transiciones de $T_i \cap \text{TI}$ a partir del vector λ de tasas de transiciones de interfaz (es una función proyección). La función g_i con $1 \leq i \leq K$ calcula las nuevas tasas λ'_i de las transiciones de $T_i \cap \text{TI}$ en una iteración parcial (hasta el estudio del \mathcal{LS}_i) del algoritmo 3.18, si se toma λ como vector de tasas iniciales para las transiciones de interfaz. La función G calcula las nuevas tasas λ' de las transiciones de interfaz tras una iteración completa del algoritmo 3.18, si se toma λ como vector de tasas iniciales para las transiciones de interfaz. Las funciones χ_{ij} y μ_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ calculan el throughput y la tasa proporcional de disparo de la transición de interfaz I_{ij} en \mathcal{LS}_i , si se toma λ_k como vector de tasas para las transiciones de $T_k \cap \text{TI}$ para todo $k \neq i$. La función μ_i con $1 \leq i \leq K$ calcula el vector μ_i de tasas proporcionales de disparo de las transiciones de $T_i \cap \text{TI}$ que se obtiene en \mathcal{LS}_i , si se toma λ_j como vector de tasas para las transiciones de $T_j \cap \text{TI}$ para todo $j \neq i$. La función $\chi_{\mathcal{BS}}$ calcula el throughput de las transiciones de \mathcal{BS} , si se toma λ como vector de tasas para sus transiciones. La función $\chi_{\mathcal{BS}_i}[\lambda_1, \dots, \mu_i, \dots, \lambda_K]$ con $1 \leq i \leq K$ calcula el throughput de las transiciones de \mathcal{BS} , si se ponen λ_j con $j \neq i$ como vectores de tasas de disparo de las transiciones de $T_j \cap \text{TI}$ y $\lambda\mu_i$ para las transiciones de $T_i \cap \text{TI}$ (λ es el parámetro de la función).

El siguiente resultado expone las relaciones que hay entre las funciones y vectores de las notaciones anteriores.

Proposición 3.27 *Sea $\langle \mathcal{S}, w \rangle$ un MG estocástico, fuertemente conexo y vivo y $B \subseteq P$ un K -corte de \mathcal{N} . Se cumplen las siguientes relaciones:*

1. $g_1(\lambda) = \lambda'_1 = \chi_{\mathcal{BS}_1}^{-1}(\chi_{11}(\lambda))\mu_1(\lambda)$.
2. $\lambda^{(i)} = (g_1(\lambda), \dots, g_i(\lambda), p_{i+1}(\lambda), \dots, p_K(\lambda))$ para $1 \leq i < K$.
3. $g_i(\lambda) = \lambda'_i = \chi_{\mathcal{BS}_i}^{-1}(\chi_{i1}(\lambda^{(i-1)}))\mu_i(\lambda^{(i-1)})$ para $1 < i \leq K$.

Demostración:

Evidente a partir del algoritmo 3.18 y de las notaciones anteriores. Sólo falta demostrar que existen las funciones inversas de $\chi_{\mathcal{BS}_i}$, que se hará en la proposición 3.30. \diamond

Con las notaciones y relaciones anteriores se puede describir el algoritmo iterativo de la siguiente manera.

Notación 3.28 Sea $\langle \mathcal{S}, w \rangle$ un MG estocástico, fuertemente conexo y vivo y $B \subseteq P$ un K -corte de \mathcal{N} . Sean $\lambda_{ij}^{(0)}$ con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ las tasas de disparo de las transiciones I_{ij} en $\langle \mathcal{S}, w \rangle$, $\boldsymbol{\lambda}_i^{(0)} = (\lambda_{i1}, \dots, \lambda_{in_i})$ y $\boldsymbol{\lambda}^{(0)} = (\boldsymbol{\lambda}_1^{(0)}, \dots, \boldsymbol{\lambda}_K^{(0)})$. El algoritmo iterativo 3.18 puede escribirse de la siguiente manera:

Algoritmo 3.29

input: $\boldsymbol{\lambda}^{(0)}$ tasas iniciales de las transiciones de interfaz.

$n := 0$ (contador de iteraciones)

repeat

$n := n + 1$

$\boldsymbol{\lambda}^{(n)} := G(\boldsymbol{\lambda}^{(n-1)})$

until convergencia de $\boldsymbol{\lambda}^{(n)}$

output: $(\chi_{11}(\boldsymbol{\lambda}^{(n)}), \dots, \chi_{Kn_K}(\boldsymbol{\lambda}^{(n)}))$ vector de aproximaciones de χ .

Con esta forma alternativa de ilustrar el método iterativo se pueden demostrar algunas propiedades referentes al mismo. Lo primero que hay que demostrar es que las búsquedas lineales en \mathcal{BS} siempre acaban con éxito.

Proposición 3.30 Sea $\langle \mathcal{S}, w \rangle$ un MG estocástico, fuertemente conexo y vivo y $B \subseteq P$ un K -corte de \mathcal{N} . Sea $\boldsymbol{\lambda}$ un vector de tasas asociadas a las transiciones de interfaz de los \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} . Sea $\chi_{i1}(\boldsymbol{\lambda})$ el throughput de la transición de interfaz I_{i1} y $\boldsymbol{\mu}_i(\boldsymbol{\lambda})$ las tasas proporcionales de disparo de las transiciones de $T_i \cap \text{TI}$ calculadas en \mathcal{LS}_i con las tasas indicadas en $\boldsymbol{\lambda}$. La función $\chi_{\mathcal{BS}_i}[\boldsymbol{\lambda}_1, \dots, \boldsymbol{\mu}_i, \dots, \boldsymbol{\lambda}_K] : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ de la notación 3.26 es continua, monótona creciente y acotada, por lo que existe su función inversa en el intervalo de acotación. En particular, existe $\lambda \in \mathbb{R}^+$ tal que $\chi_{\mathcal{BS}_i}[\boldsymbol{\lambda}_1, \dots, \boldsymbol{\mu}_i, \dots, \boldsymbol{\lambda}_K](\lambda) = \chi_{i1}(\boldsymbol{\lambda})$.

Demostración:

Se denotará a la función $\chi_{\mathcal{BS}_i}[\boldsymbol{\lambda}_1, \dots, \boldsymbol{\mu}_i, \dots, \boldsymbol{\lambda}_K]$ por $\chi_{\mathcal{BS}_i}$ ya que no hay confusión sobre las tasas que se asocian a las transiciones de interfaz. Por el corolario 3.21, el throughput de \mathcal{BS} es una función continua de las tasas de las transiciones. En particular, $\chi_{\mathcal{BS}_i}$ es continua en \mathbb{R}^+ ya que las tasas asociadas a las transiciones de \mathcal{BS} son polinomios de grado 1 en λ (son constantes o múltiplos constantes de λ), es decir, las tasas son funciones continuas de λ . Como la composición de funciones continuas es continua, se sigue que $\chi_{\mathcal{BS}_i}$ es continua en \mathbb{R}^+ .

Por otro lado, aplicando el teorema 3.22 sobre monotonía de throughput en MG's con respecto a las tasas de las transiciones, se tiene que $\chi_{\mathcal{BS}_i}$ es una función monótona creciente en \mathbb{R}^+ .

Es evidente que $\chi_{\mathcal{BS}_i}(0) = 0$ ya que si $\lambda = 0$, las transiciones de $T_i \cap \text{TI}$ tienen tasa 0. Una transición de tasa 0 nunca se dispara, es decir, tiene throughput nulo. Como por el corolario 2.94 todas las transiciones de un MG tienen el mismo throughput, entonces $\chi_{\mathcal{BS}_i}(0) = 0$. Por otro lado, $\chi_{\mathcal{BS}_i}$ debe ser acotada en \mathbb{R}^+ . Esto se puede demostrar aplicando la propiedad 3.24. Si se hacen inmediatas las transiciones de interfaz de \mathcal{N}_i (esto se corresponde con $\lim_{\lambda \rightarrow \infty} \chi_{\mathcal{BS}_i}(\lambda)$), como \mathcal{N} es fuertemente conexo, se puede construir un ciclo que pase por alguna transición temporizada t de \mathcal{N}_j con $j \neq i$. Como t es una transición temporizada, el tiempo medio entre disparos de t en este ciclo es una cierta constante $a > 0$. Aplicando el problema de programación lineal de la propiedad 3.24, se obtiene que $\lim_{\lambda \rightarrow \infty} \chi_{\mathcal{BS}_i}(\lambda) \leq 1/a$. Por lo tanto, $\chi_{\mathcal{BS}_i}$ es monótona creciente acotada en \mathbb{R}^+ .

Falta demostrar que existe solución a la ecuación $\chi_{\mathcal{BS}_i}(\lambda) = \chi_{i1}(\boldsymbol{\lambda})$. La única diferencia entre \mathcal{LS}_i y \mathcal{BS} es que la subred \mathcal{N}_i ha sido eliminada en \mathcal{BS} . Dicho de otra manera, \mathcal{LS}_i contiene a \mathcal{BS} . Por el teorema 3.23 (monotonía de throughput respecto a la topología), se tiene que $\chi_{i1}(\boldsymbol{\lambda}) \leq \chi_{\mathcal{BS}}[\boldsymbol{\lambda}]$ (el throughput de \mathcal{LS}_i es menor o igual que el throughput de \mathcal{BS} si se mantienen las tasas de las transiciones de $T_i \cap \text{TI}$). Ahora, tomando el vector $\boldsymbol{\mu}_i$ de tasas proporcionales de disparo de las transiciones de $T_i \cap \text{TI}$, es evidente que existe un $\lambda' > 0$ tal que $\lambda' \mu_{ij} > \lambda_{ij}$ para todo $1 \leq j \leq n_i$ (basta tomar $\lambda' > \max_j \lambda_{ij} / \min_j \mu_{ij}$). Por el teorema 3.22, $\chi_{i1}(\boldsymbol{\lambda}) \leq \chi_{\mathcal{BS}}[\boldsymbol{\lambda}] \leq \chi_{\mathcal{BS}_i}(\lambda')$, luego $\chi_{\mathcal{BS}_i}(0) < \chi_{i1}(\boldsymbol{\lambda}) \leq \chi_{\mathcal{BS}}(\lambda')$. Aplicando la propiedad de Darboux (teorema 2.9), existe $\lambda \in (0, \lambda']$ tal que $\chi_{\mathcal{BS}_i}(\lambda) = \chi_{i1}$. \diamond

Este teorema asegura que en todas las iteraciones del algoritmo y en el estudio de todos los \mathcal{LS}_i existe un factor de escala λ tal que, al poner como tasas de disparo para las transiciones de $T_i \cap \text{TI}$ en \mathcal{BS} el vector $\lambda \boldsymbol{\mu}_i$, el throughput que se obtiene para todas las transiciones de \mathcal{BS} es el mismo que el que se había calculado para las transiciones de \mathcal{LS}_i . La codificación numérica del cálculo de este factor de escala se realiza por medio de algoritmos como el de la secante, ya que este es un problema numérico de cálculo de una solución para una ecuación no lineal con una incógnita. Se utiliza un algoritmo iterativo en el que se van probando distintos factores de escala y en función del throughput obtenido en \mathcal{BS} , teniendo en cuenta la propiedad 3.22 de monotonía de throughput respecto a las tasas de disparo de las

transiciones de un MG, se va incrementando o decrementando este factor de escala hasta que el throughput de \mathcal{BS} y el de \mathcal{LS}_i están suficientemente cerca. Una vez calculado el factor de escala λ , multiplicando el vector $\boldsymbol{\mu}_i$ de proporciones entre tasas de disparo por λ , se obtiene una nueva aproximación para las tasas de disparo de las transiciones de interfaz de la subred \mathcal{N}_i , que son las que aproximan el tiempo de respuesta de toda esta subred.

El siguiente paso consiste en demostrar la existencia de solución para el método iterativo. En [MT95] se demuestra la existencia de punto fijo para funciones que definen algoritmos iterativos del estilo del que se utiliza en esta memoria, pero sus resultados no son directamente aplicables en este caso debido a la existencia del \mathcal{BS} . Aquí se demostrará la existencia de un punto fijo para la función G del algoritmo 3.29. Un punto fijo de la función G se corresponde con una solución del método numérico, es decir, al poner como tasas para las transiciones de interfaz de los \mathcal{LS}_i y \mathcal{BS} los valores numéricos del punto fijo de G se obtienen los mismos throughput para todas las transiciones de interfaz de todos los \mathcal{LS}_i y \mathcal{BS} . Para demostrar la existencia de este punto fijo se utilizará el teorema del punto fijo de Brouwer (ver teorema 2.10).

Teorema 3.31 *Sea $\langle \mathcal{S}, w \rangle$ un MG estocástico, fuertemente conexo y vivo y $B \subseteq P$ un K -corte de \mathcal{N} . Sea $G : (\mathbb{R}^+)^M \rightarrow (\mathbb{R}^+)^M$ la función del algoritmo 3.29 según la notación 3.26. G es continua y existe un conjunto $S \subset (\mathbb{R}^+)^M$ compacto, conexo y no vacío tal que $G(S) \subseteq S$.*

Demostración:

La primera propiedad que hay que demostrar es la continuidad de G . La función G queda descrita por las funciones $g_i : (\mathbb{R}^+)^M \rightarrow (\mathbb{R}^+)^{n_i}$ con $1 \leq i \leq K$ de la notación 3.26. Por lo tanto, basta demostrar que las funciones g_i son continuas. Esto se demostrará por inducción sobre i .

Caso $i = 1$. $g_1(\boldsymbol{\lambda}) = \boldsymbol{\lambda}'_1 = \chi_{\mathcal{BS}_1}^{-1}(\chi_{11}(\boldsymbol{\lambda}))\boldsymbol{\mu}_1(\boldsymbol{\lambda})$ (la última igualdad es por la proposición 3.27). Las tasas relativas de disparo $\boldsymbol{\mu}_1(\boldsymbol{\lambda})$ calculadas en \mathcal{LS}_1 para las transiciones de $T_1 \cap \text{TI}$ son el cociente del throughput de \mathcal{LS}_1 (función continua en $\boldsymbol{\lambda}$ por corolario 3.21) por la utilización de la transición de interfaz (caso de semántica de un solo servidor) o por el grado medio de sensibilización de la misma (caso de semántica de infinitos servidores). En cualquier caso, estos índices de prestaciones también son continuos en $\boldsymbol{\lambda}$ por el corolario 3.21. Además, como las transiciones de interfaz de \mathcal{LS}_1 son temporizadas y \mathcal{LS}_1 es vivo, tanto la utilización como el grado medio

de sensibilización son no nulos, lo que garantiza la continuidad de $\mu_1(\boldsymbol{\lambda})$. Por la proposición 3.30, $\chi_{\mathcal{BS}_1}^{-1}[\boldsymbol{\mu}_1, \boldsymbol{\lambda}_2, \dots, \boldsymbol{\lambda}_K]$ es una función continua y χ_{11} también es continua en $\boldsymbol{\lambda}$, luego g_1 es una función continua en $\boldsymbol{\lambda}$ (por ser producto y composición de funciones continuas en $\boldsymbol{\lambda}$).

Suponer que g_1, \dots, g_{i-1} son continuas en $\boldsymbol{\lambda}$. Se verá que g_i es continua en $\boldsymbol{\lambda}$. $g_i(\boldsymbol{\lambda}) = \boldsymbol{\lambda}'_i = \chi_{\mathcal{BS}_i}^{-1}(\chi_{i1}(\boldsymbol{\lambda}^{(i-1)}))\mu_i(\boldsymbol{\lambda}^{(i-1)})$ (la última igualdad es por la proposición 3.27). Ahora $\boldsymbol{\lambda}^{(i-1)} = (g_1(\boldsymbol{\lambda}), \dots, g_{i-1}(\boldsymbol{\lambda}), p_i(\boldsymbol{\lambda}), \dots, p_K(\boldsymbol{\lambda}))$. Por hipótesis de inducción, las $i-1$ primeras componentes de $\boldsymbol{\lambda}^{(i-1)}$ son funciones continuas en $\boldsymbol{\lambda}$ y el resto de componentes son proyecciones de $\boldsymbol{\lambda}$, luego también funciones continuas en $\boldsymbol{\lambda}$. Por lo tanto, $\boldsymbol{\lambda}^{(i-1)}$ es una función continua en $\boldsymbol{\lambda}$. Ahora, con los mismos argumentos que en el caso 1, se sigue que g_i es una función continua en $\boldsymbol{\lambda}$.

Falta demostrar la existencia de un conjunto S compacto, convexo no vacío tal que $G(S) \subseteq S$. Como las transiciones de interfaz son temporizadas (ver párrafo sobre restricciones en el corte), si se inicia el algoritmo iterativo con vector inicial de tasas de las transiciones de interfaz $\boldsymbol{\lambda}^{(0)} = (\infty, \dots, \infty)$ (es lo mismo que hacer inmediatas las transiciones de interfaz de las subredes \mathcal{N}_j con $j \geq 2$), el throughput de \mathcal{LS}_1 es finito (ya que hay al menos una transición temporizada) y distinto de 0 (por ser \mathcal{LS}_1 vivo). Además, es máximo en el sentido de que es imposible conseguir con cualquier otra combinación inicial de tasas $\boldsymbol{\lambda}^{(0)}$ un throughput mayor para \mathcal{LS}_1 (consecuencia directa del teorema 3.22 sobre monotonía de throughput respecto a las tasas de las transiciones). El mismo análisis puede hacerse para cualquier \mathcal{LS}_i renumerando adecuadamente los sistemas agregados. Sea χ el máximo throughput que puede obtenerse de esta manera para cualquier \mathcal{LS}_i . Es claro que $\chi \in (0, \infty)$. Tomando ahora \mathcal{BS} y cualquier transición I_{ij} suya, si se pone como tasa $\lambda = 1/\chi$ y se hacen inmediatas el resto, el throughput de \mathcal{BS} es χ . Por construcción, $\lambda \in (0, \infty)$. Además, χ es el máximo valor del throughput que puede aparecer en el esquema iterativo para \mathcal{BS} y, por el teorema 3.22, λ es el máximo valor de la tasa de disparo de cualquier transición de interfaz que puede aparecer en el esquema iterativo. Sea $S = [0, \lambda]^M \subseteq (\mathbb{R}^+)^M$. Claramente S es un conjunto compacto, convexo y no vacío (porque λ es finito y distinto de 0). Por construcción de S , es evidente que $G((\mathbb{R}^+)^M) \subseteq S$, en particular, $G(S) \subseteq S$. \diamond

Corolario 3.32 *Sea $\langle \mathcal{S}, w \rangle$ un MG estocástico, fuertemente conexo y vivo y $B \subseteq P$ un K -corte de \mathcal{N} . El algoritmo iterativo 3.29 aplicado a los sistemas \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} tiene punto fijo.*

Demostración:

Inmediato aplicando el teorema 2.10 a la función G que es la que describe una iteración del algoritmo 3.29. \diamond

Con los últimos resultados, se ha conseguido demostrar la existencia de punto fijo para el esquema iterativo utilizado, y por lo tanto, la posibilidad de convergencia del esquema a uno de esos puntos fijos. Se ha intentado estudiar más en detalle las características del esquema para asegurar convergencia, unicidad de solución e incluso estimar el error cometido en la aproximación, pero no se ha llegado a resultados generales debido a la gran complejidad de los sistemas agregados que se emplean en el esquema iterativo. Desde el punto de vista del autor de esta memoria, es posible que algún día se pueda demostrar que el esquema converge (utilizando otros teoremas de punto fijo), e incluso que este punto de convergencia sea único. Para ello sería necesario demostrar propiedades mucho más exigentes que la continuidad de las funciones empleadas en el esquema (por ejemplo la contractividad). Este tipo de propiedades parecen claras si se ven los resultados experimentales pero extremadamente difíciles de demostrar teóricamente. Lo que no parece posible es estimar el error cometido en la aproximación debido a que este error es inducido por la descomposición estructural del MG original y a que las agregaciones de estados que se producen al construir los sistemas agregados no son exactas desde el punto de vista de CTMC's.

A falta de un mejor soporte teórico, la experimentación numérica con diferentes ejemplos sugiere que el esquema converge en muy pocos pasos (entre 3 y 5 iteraciones), converge a la misma solución independientemente de las tasas iniciales elegidas para iniciar el algoritmo, y el error cometido al final del mismo (inferior al 5%) es admisible para técnicas de aproximación en evaluación de prestaciones de sistemas concurrentes basada en modelos formales.

3.5 Conclusiones

En este capítulo se ha desarrollado en detalle el trabajo de [CCJS94] sobre aproximación de throughput en MG's, ya que este trabajo es el punto de partida para las aportaciones de esta memoria. Se ha aprovechado la ocasión para introducir algunas mejoras técnicas que permiten afianzar cuestiones teóricas sobre la técnica así como posibles codificaciones de la misma. En concreto, las aportaciones de este capítulo son:

- Ampliación de la técnica para permitir un corte múltiple del MG original en un número finito arbitrario de subredes.
- Reducción del número de lugares implícitos necesarios para resumir el comportamiento de las subredes en los sistemas agregados. La reducción se logra manteniendo exactamente las proyecciones de estados alcanzables y secuencias de disparo del MG original sobre los nodos preservados en cada sistema agregado.
- Demostración formal de existencia de punto de convergencia para el algoritmo de aproximación del tiempo de respuesta.

Queda como trabajo futuro demostrar la convergencia del algoritmo numérico de aproximación del tiempo de respuesta a algún punto de convergencia de los que ya se sabe que existen. Se ha intentado demostrar sin éxito completo hasta el momento.

Capítulo 4

Aproximación de throughput en grafos marcados con pesos

En este capítulo se va a desarrollar una técnica de aproximación del throughput de las transiciones de un grafo marcado con pesos (WTS a partir de aquí, que son las iniciales de su nombre inglés, *weighted T-systems*).

Esta técnica es una extensión de la desarrollada en el capítulo anterior para grafos marcados (MG's), en el sentido de que las dos son exactamente la misma si se aplican a un MG (todo MG es un WTS).

Todo lo expuesto en este capítulo son aportaciones de esta memoria. Esta técnica fue presentada en [PJCS96b] y aquí se expondrá introduciendo alguna mejora técnica tal y como se ha hecho en el capítulo anterior con MG's. Estas mejoras técnicas consisten en la reducción de la estructura de los sistemas agregados que se construyen a partir del modelo original y la demostración de existencia de punto fijo para el algoritmo numérico de aproximación del throughput de las transiciones.

El capítulo está organizado de la siguiente manera. La sección 4.1 describe a grandes rasgos la técnica y la relaciona con otras existentes en la literatura, en la sección 4.2 se desarrollan las propiedades estructurales de los WTS's que es necesario conocer para la fase de descomposición. En concreto se definen los conceptos de ganancia, marcado ponderado y resistencia y se demuestran sus principales propiedades. La sección 4.3 se dedica a la técnica de descomposición estructural de WTS's y la sección 4.4 al método iterativo para la aproximación del throughput de sus transiciones. Por último, en la sección 4.5 se indican las conclusiones finales del capítulo.

4.1 Introducción

Los WTS's son una subclase de P/T sistemas que incluye a la de los MG's. De hecho son la extensión con pesos de los MG's. Permiten el modelado de concurrencia, sincronizaciones y movimiento de tareas por bloques, pero no de decisiones. En la sección 2.4.3 se definen los WTS's y sus principales propiedades.

Los WTS's pueden emplearse para el modelado de sistemas de ensamblado y desensamblado. En estos sistemas se realizan tareas de ensamblado de diversas piezas simples para formar otras más complejas o de desensamblado de piezas complejas en varias simples.

Para que tenga sentido aproximar el throughput de las transiciones de un WTS es necesario exigir algunas condiciones. Por un lado el WTS debe ser conexo porque en caso contrario se tienen varios WTS's independientes que se pueden estudiar por separado (uno por cada componente conexa). Se restringirá el estudio a WTS's limitados con el objeto de obtener cadenas de Markov en tiempo continuo (CTMC) finitas. Además, es necesario que el WTS sea vivo porque si no el throughput de sus transiciones es nulo y en ese caso no tiene sentido desarrollar una técnica de aproximación. Por último, el WTS debe tener estados recurrentes, lo que asegura la existencia de distribución en estado estacionario única para su CTMC isomorfa (teorema 2.85). Aplicando la teoría estructural de redes se tiene que todo P/T sistema vivo y limitado es fuertemente conexo y consistente (teorema 2.50). Todo WTS consistente tiene un único t-semiflujo que cubre a todas las transiciones [Ter94] por lo que las ratios de visita de sus transiciones vienen determinadas por la estructura de la red, y por lo tanto también los throughput de sus transiciones. Como los WTS's fuertemente conexos y vivos son libres de bloqueo (apartado 4 del teorema 2.98), entonces los throughput de sus transiciones son no nulos. Por último, para WTS's consistentes son equivalentes vivacidad y reversibilidad (teorema 2.99). Por lo tanto, todo WTS vivo y limitado es fuertemente conexo y reversible lo que asegura la ergodicidad de su CTMC isomorfa (teorema 2.85). En consecuencia, se desarrollará una técnica de aproximación del throughput de las transiciones de WTS's vivos y limitados.

La técnica que se presenta en este capítulo, al igual que todas las de esta memoria, se basa en una descomposición estructural del modelo original en varias subredes. A partir de estas subredes se construyen varios sistemas de bajo nivel (en los que hay una subred producto de la descomposición y

un resumen del resto de las subredes) y un sistema de alto nivel o esqueleto básico (formado por los resúmenes de todas las subredes). Esta técnica es una generalización de la presentada en el capítulo anterior para MG's, en el sentido de que la técnica de aproximación para WTS's aplicada a un MG es exactamente la misma que la del capítulo anterior.

En el caso de MG's era posible construir los sistemas de alto y bajo nivel de forma que estos mantenían las proyecciones de los marcados alcanzables y las secuencias de disparo del modelo original sobre los nodos preservados en cada sistema agregado. Esta característica no se va a mantener en la técnica de este capítulo. La descomposición estructural que se realiza construye unos sistemas agregados que preservan un conjunto de propiedades más débil que en el caso de MG's, pero que son suficientes en esta clase de redes para obtener buenas aproximaciones del throughput de las transiciones del WTS original. La razón de no mantener las proyecciones de marcados alcanzables y secuencias de disparo en los sistemas agregados es reducir el número de nodos necesarios para resumir las subredes producto del corte. En el capítulo 6 se desarrollará una técnica de descomposición aplicable a cualquier red de Petri que opera a nivel del grafo de alcanzabilidad y que permitirá mantener al menos las proyecciones de los marcados alcanzables en los sistemas agregados.

Las propiedades que es necesario preservar en la descomposición estructural son vivacidad y limitación. Preservando estas propiedades en los sistemas agregados, se asegura la obtención de CTMC's finitas y ergódicas, lo que permite su resolución. Sin embargo, preservando únicamente estas dos propiedades no se obtienen, en general, buenas aproximaciones para el throughput de las transiciones de un WTS's, como se verá en la siguiente sección. Para resolver este problema es necesario conocer algunas propiedades estructurales de los WTS's, como son la *ganancia*, el *mercado ponderado* y la *resistencia* de caminos con pesos. Estos conceptos se introducen en la siguiente sección y serán necesarios para la fase de descomposición estructural. Esta fase se podrá realizar con una complejidad en espacio y tiempo de orden polinomial en el tamaño de la red original.

Por lo que respecta a la parte numérica, no hay grandes cambios respecto a la expuesta en el capítulo anterior para MG's. Se sigue pudiendo asegurar la existencia de punto de convergencia para el algoritmo numérico en esta clase de redes y obviamente se sigue sin poder demostrar la convergencia del método. En cualquier caso, los experimentos numéricos realizados aseguran buen comportamiento del método, con convergencia en pocas iteraciones

(entre 3 y 5) y error asumible en una técnica de aproximación (inferior al 5%).

Otros autores han estudiado las prestaciones de los WTS's cuando a las transiciones del modelo se les asocia tiempos de disparo determinista (ver por ejemplo [Mun93]), pero nadie ha estudiado el caso de tiempos estocásticos.

4.2 Ganancia, mercado ponderado y resistencia

Los WTS's son P/T sistemas en los que los lugares tienen una única transición de entrada y de salida (ver definición 2.97), pero a diferencia de los MG's, ahora puede haber pesos en los arcos. Los pesos en los arcos inducen sincronizaciones en las transiciones. Si una determinada transición tiene un único lugar de entrada conectado por un arco de peso 1 sólo se necesita una marca en el lugar para sensibilizar la transición, pero si el peso del arco es n , todas las marcas que lleguen al lugar quedan paradas hasta que éste contenga n marcas. Desde este punto de vista los arcos con pesos inducen sincronizaciones en los modelos.

Para realizar la descomposición estructural de WTS's y generar los sistemas agregados con las propiedades deseadas (vivacidad y limitación) se necesita conocer algunas propiedades estructurales sobre el funcionamiento de los WTS's. Este es el trabajo que se va a desarrollar en esta sección. Estas propiedades se definen sobre caminos cualesquiera del WTS que comienzan y terminan en transición, y son la ganancia, el mercado ponderado y la resistencia. La ganancia y el mercado ponderado sirven para preservar la vivacidad y limitación de los sistemas agregados. Como se verá en esta sección, preservar estas propiedades no es suficiente para obtener buenos resultados numéricos. La resistencia sirve para mejorarlos. Para cada concepto se dará una idea intuitiva de su significado, después se definirá formalmente y se demostrarán sus propiedades.

La primera propiedad estructural a tener en cuenta es la ganancia de un camino. El concepto de ganancia fue introducido en [TCWCS92] para ciclos en un WTS y es un concepto que ayuda a calcular la vivacidad y limitación de los WTS's. Para que un WTS sea vivo y limitado cualquier ciclo suyo debe tener ganancia 1. En esta memoria se extiende el concepto de ganancia a caminos de un WTS que conectan transiciones. Este concepto extendido de ganancia representa la proporción media de disparos entre la transición final del camino y la inicial.

Definición 4.1 Ganancia de un camino.

Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle P, T, F, W \rangle$ un WTS, $\pi = (t_0 p_1 t_1 p_2 \dots p_n t_n)$ un camino dirigido en \mathcal{N} que conecta las transiciones t_0 y t_n . Sean $r_i = W(t_{i-1}, p_i)$ y $s_i = W(p_i, t_i)$ con $1 \leq i \leq n$. La ganancia del camino π es:

$$G(\pi) = \prod_{i=1}^n \frac{r_i}{s_i} \in \mathbb{Q}^+ \setminus \{0\}.$$

La ganancia de un camino sólo depende de los pesos de los arcos del camino. Notar que en el caso de MG's la ganancia de cualquier camino es 1. La siguiente proposición es inmediata a partir de la definición.

Proposición 4.2 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS, π_1 y π_2 dos caminos en \mathcal{N} tales que la transición final de π_1 es la transición inicial de π_2 . Sea $\pi_3 = \pi_1 \circ \pi_2$ el camino concatenación de los otros dos. Se tiene que $G(\pi_3) = G(\pi_1) \cdot G(\pi_2)$.

Demostración:

Sean r_i, s_i con $1 \leq i \leq n$ los pesos de los arcos de π_1 (r_i para arcos de transición a lugar y s_i para arcos de lugar a transición) y r'_i, s'_i con $1 \leq i \leq m$ los de π_2 . Por la definición 4.1 se tiene:

$$G(\pi_3) = \frac{(\prod_{i=1}^n r_i) \cdot (\prod_{i=1}^m r'_i)}{(\prod_{i=1}^n s_i) \cdot (\prod_{i=1}^m s'_i)} = (\prod_{i=1}^n \frac{r_i}{s_i}) \cdot (\prod_{i=1}^m \frac{r'_i}{s'_i}) = G(\pi_1) \cdot G(\pi_2). \quad \diamond$$

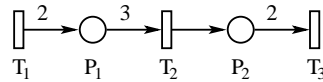


Figura 4.1: Camino de ganancia 1/3.

Por ejemplo, en la figura 4.1, el camino de T_1 a T_2 tiene arcos de pesos 2 y 3, luego la ganancia del camino de T_1 a T_2 es $2/3$ y el comportamiento del camino es tal que por cada 3 disparos de T_1 , la transición T_2 puede dispararse 2 veces. Análogamente, la ganancia del camino de T_2 a T_3 es $1/2$ y la del camino de T_1 a T_3 es $1/3$ (por la proposición 4.2).

Si el objetivo es resumir ciertas subredes para generar los sistemas agregados, algunos caminos serán sustituidos por otros más cortos. Al sustituir cualquier camino en el WTS original por otro, es necesario que el nuevo camino tenga la misma ganancia que el original, para asegurar la vivacidad y limitación de los sistemas agregados (ver [TCWCS92]).

La extensión natural a WTS's de la técnica de descomposición de MG's consistiría en reducir las subredes producto del corte de un WTS preservando la ganancia de los caminos entre transiciones de interfaz. De esta manera se obtienen sistemas agregados que son WTS's vivos y limitados. En el caso de MG's todos los caminos tienen ganancia 1 y por lo tanto las dos técnicas serían equivalentes. Sólo faltaría calcular los marcados iniciales de los lugares que resumen los caminos. En MG's el problema se reducía a calcular la suma de las marcas de los lugares que componen un camino, pero en el caso de WTS's no puede hacerse de forma tan sencilla porque los pesos de los arcos hacen que las marcas de un determinado lugar de un camino no tengan la misma importancia que los de otro. Por ejemplo, en la figura 4.2 hay un camino con dos lugares; P_1 con 1 marca y P_2 con 2 marcas. Es evidente que para que en P_2 haya una marca se ha tenido que disparar antes la transición T_2 y por lo tanto tuvo que haber antes dos marcas en P_1 . Dicho de otra manera, cada marca en P_2 equivale a dos marcas en P_1 debido a los pesos de los arcos de entrada y de salida de la transición T_2 .

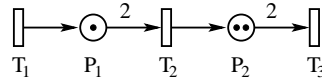


Figura 4.2: Camino de marcado ponderado 3.

Por lo tanto, es necesario extender el concepto de suma de marcas de un camino en un MG a un concepto similar en WTS's. Este concepto se llama marcado ponderado de un camino. La idea intuitiva del marcado ponderado de un camino es el número de veces que hace falta disparar la primera transición del camino para alcanzar la disposición de marcas de los lugares del camino (suponiendo que se parte del camino totalmente desmarcado).

Definición 4.3 Marcado ponderado de un camino.

Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS, $\pi = (t_0 p_1 t_1 p_2 \dots p_n t_n)$ un camino en \mathcal{N} que conecta dos transiciones t_0 y t_n con $r_i = F(t_{i-1}, p_i)$ y $s_i = F(p_i, t_i)$ para $1 \leq i \leq n$. Si $\mathbf{m} \in R(\mathcal{S})$, el marcado ponderado del camino π en \mathbf{m} es:

$$M(\pi, \mathbf{m}) = \sum_{i=1}^n \left(\frac{\mathbf{m}[p_i]}{r_i} \prod_{j=1}^{i-1} \frac{s_j}{r_j} \right) \in \mathbb{Q}^+.$$

En general, el marcado ponderado es un racional no negativo que depende de los pesos de los arcos y de las marcas de los lugares que lo componen.

Teniendo en cuenta la idea intuitiva del mercado ponderado se puede entender la fórmula de la definición, ya que cada sumando representa el mercado de cada lugar del camino en términos del número de disparos de la transición inicial necesarios para obtener las marcas del lugar.

Notar que en el caso de MG's, el mercado ponderado de un camino coincide con la suma de las marcas de los lugares del camino ya que todos los arcos tienen peso 1.

El camino de la figura 4.2 tiene mercado ponderado 3 que coincide con el número de disparos de la transición T_1 que es necesario realizar para que se pueda llegar a esa disposición de marcas.

A continuación se van a demostrar algunas propiedades de la ganancia y del mercado ponderado de caminos que ayudan a entender el funcionamiento de los WTS's.

Proposición 4.4 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS, π un camino en \mathcal{N} que conecta dos transiciones t_0 y t_n con $r_i = F(t_{i-1}, p_i)$ y $s_i = F(p_i, t_i)$ para $1 \leq i \leq n$. Sean $\mathbf{m}, \mathbf{m}' \in \mathbb{R}(\mathcal{S})$ de forma que existe una transición t tal que $\mathbf{m} \xrightarrow{t} \mathbf{m}'$. Se cumplen las siguientes propiedades:

1. Si $t \neq t_0, t_n$ entonces $M(\pi, \mathbf{m}') = M(\pi, \mathbf{m})$.
2. Si $t = t_0$ entonces $M(\pi, \mathbf{m}') = M(\pi, \mathbf{m}) + 1$.
3. Si $t = t_n$ entonces $M(\pi, \mathbf{m}') = M(\pi, \mathbf{m}) - 1/G(\pi)$.

Demostración:

1) Si se dispara la transición interna t_i con $1 \leq i < n$, sólo cambian $\mathbf{m}[p_i]$ y $\mathbf{m}[p_{i+1}]$. Se tiene $\mathbf{m}'[p_i] = \mathbf{m}[p_i] - s_i$ y $\mathbf{m}'[p_{i+1}] = \mathbf{m}[p_{i+1}] + r_{i+1}$, por lo tanto, $M(\pi, \mathbf{m}') - M(\pi, \mathbf{m}) = \frac{\mathbf{m}'[p_i] - \mathbf{m}[p_i]}{r_i} \prod_{j=1}^{i-1} \frac{s_j}{r_j} + \frac{\mathbf{m}'[p_{i+1}] - \mathbf{m}[p_{i+1}]}{r_{i+1}} \prod_{j=1}^i \frac{s_j}{r_j} = \frac{-s_i}{r_i} \prod_{j=1}^{i-1} \frac{s_j}{r_j} + \frac{r_{i+1}}{r_{i+1}} \prod_{j=1}^i \frac{s_j}{r_j} = -\prod_{j=1}^i \frac{s_j}{r_j} + \prod_{j=1}^i \frac{s_j}{r_j} = 0$.

2) Si se dispara t_0 , sólo cambia $\mathbf{m}[p_1]$. Se tiene que $\mathbf{m}'[p_1] = \mathbf{m}[p_1] + r_1$, es decir, $M(\pi, \mathbf{m}') - M(\pi, \mathbf{m}) = \frac{\mathbf{m}'[p_1] - \mathbf{m}[p_1]}{r_1} = \frac{r_1}{r_1} = 1$.

3) Si se dispara t_n , sólo cambia $\mathbf{m}[p_n]$. Se tiene que $\mathbf{m}'[p_n] = \mathbf{m}[p_n] - s_n$, es decir, $M(\pi, \mathbf{m}') - M(\pi, \mathbf{m}) = \frac{\mathbf{m}'[p_n] - \mathbf{m}[p_n]}{r_n} \prod_{j=1}^{n-1} \frac{s_j}{r_j} = \frac{-s_n}{r_n} \prod_{j=1}^{n-1} \frac{s_j}{r_j} = -\prod_{j=1}^n \frac{s_j}{r_j} = -1/G(\pi)$. \diamond

Esta proposición establece que el mercado ponderado de un camino sólo varía al dispararse las transiciones inicial o final del camino. Además, el mercado ponderado aumenta por unidades y disminuye por cantidades inversas

a la ganancia del camino. Este resultado tiene consecuencias inmediatas sobre el comportamiento de los WTS's vivos y limitados.

Teorema 4.5 *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado y π un ciclo (ver definición 2.41) en \mathcal{N} . Se cumplen las siguientes propiedades:*

1. [TCWCS92] $G(\pi) = 1$.
2. $M(\pi, \mathbf{m}) = M(\pi, \mathbf{m}_0) \geq 1$ para cualquier $\mathbf{m} \in \mathbf{R}(\mathcal{S})$.

Demostración:

Sea t_i una transición cualquiera del ciclo, que se tomará como transición inicial y final del camino y $\mathbf{m}, \mathbf{m}' \in \mathbf{R}(\mathcal{S})$ tales que existe una transición t de \mathcal{N} de forma que $\mathbf{m} \xrightarrow{t} \mathbf{m}'$. Se denotará por $M(\mathbf{m}) = M(\pi, \mathbf{m})$ para todo $\mathbf{m} \in \mathbf{R}(\mathcal{S})$. Por la proposición 4.4, $M(\mathbf{m}') = M(\mathbf{m})$ si $t \neq t_i$ y $M(\mathbf{m}') = M(\mathbf{m}) + 1 - 1/G(\pi)$ si $t = t_i$ (por ser t_i la transición inicial y final del camino).

1) Si $G(\pi) < 1$ entonces $1/G(\pi) > 1$ por lo que $1 - 1/G(\pi) < 0$. En este caso el valor de $M(\mathbf{m}')$ se va decrementando con los disparos de la transición t_i . Como \mathcal{S} es vivo, t puede dispararse indefinidamente y $M(\mathbf{m}')$ puede tomar valores negativos, lo que es absurdo, por la definición 4.3. Si $G(\pi) > 1$ entonces $1/G(\pi) < 1$ por lo que $1 - 1/G(\pi) > 0$. En este caso el valor de $M(\mathbf{m}')$ se va incrementando con los disparos de la transición t , por lo que $M(\mathbf{m}')$ puede tomar valores tan grandes como se quiera, lo que es incompatible con la limitación del modelo inicial. Por lo tanto, $G(\pi) = 1$.

2) Si $G(\pi) = 1$ y $t = t_i$, $M(\mathbf{m}') = M(\mathbf{m}) + 1 - 1/G(\pi) = M(\mathbf{m})$, luego $M(\mathbf{m}') = M(\mathbf{m})$ para cualquier t . Extendiendo a secuencias de disparo, se obtiene que $M(\mathbf{m}) = M(\mathbf{m}_0)$ para cualquier $\mathbf{m} \in \mathbf{R}(\mathcal{S})$. Queda por demostrar que $M(\mathbf{m}_0) \geq 1$. Sea ahora t_i la transición final del camino y $\mathbf{m} \in \mathbf{R}(\mathcal{S})$ tal que t_i está sensibilizada en \mathbf{m} . Por el apartado 3 de la proposición 4.4, debe ser $M(\mathbf{m}) \geq 1/G(\pi)$. En estas condiciones, se tiene que $M(\mathbf{m}_0) = M(\mathbf{m}) \geq 1/G(\pi) = 1$. \diamond

Este teorema establece que todos los ciclos de un WTS vivo y limitado deben tener ganancia 1 (resultado ya demostrado en [TCWCS92]) y que su marcado ponderado es una constante (mayor o igual que 1) en todos los marcados alcanzables. Por lo tanto, en todo ciclo de cualquier WTS vivo y limitado ni entran ni salen marcas. Lo único que puede ocurrir es que las marcas que tenga el ciclo en el marcado inicial circulen permanentemente por los lugares del ciclo.

Teorema 4.6 *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado, t_i, t_o dos transiciones de \mathcal{N} y π_1, π_2 dos caminos en \mathcal{N} que conectan t_i con t_o . Se cumplen las siguientes propiedades:*

1. $G(\pi_1) = G(\pi_2)$.
2. Existe $C \in \mathbb{Q}$ tal que $\forall \mathbf{m} \in \mathbf{R}(\mathcal{S}), M(\pi_1, \mathbf{m}) = M(\pi_2, \mathbf{m}) + C$.

Demostración:

Sean $\mathbf{m}, \mathbf{m}' \in \mathbf{R}(\mathcal{S})$ tales que existe una transición t de \mathcal{N} de forma que $\mathbf{m} \xrightarrow{t} \mathbf{m}'$. En esta demostración se denotarán por $M_1(\mathbf{m})$ a $M(\pi_1, \mathbf{m})$, por $M_2(\mathbf{m})$ a $M(\pi_2, \mathbf{m})$ y por $M(\mathbf{m})$ a $M_2(\mathbf{m}) - M_1(\mathbf{m})$ para todo $\mathbf{m} \in \mathbf{R}(\mathcal{S})$.

1) Se demostrará el resultado por reducción al absurdo. Si en las condiciones del enunciado se tiene que $G(\pi_1) \neq G(\pi_2)$ se puede suponer sin pérdida de generalidad que $G(\pi_1) < G(\pi_2)$, es decir, $\frac{1}{G(\pi_1)} > \frac{1}{G(\pi_2)}$.

Si $t = t_i$, por el apartado 2 de la proposición 4.4, $M_1(\mathbf{m}') = M_1(\mathbf{m}) + 1$ y $M_2(\mathbf{m}') = M_2(\mathbf{m}) + 1$, es decir, $M(\mathbf{m}') = M(\mathbf{m})$. Si $t = t_o$, por el apartado 3 de la proposición 4.4, $M_1(\mathbf{m}') = M_1(\mathbf{m}) - \frac{1}{G(\pi_1)}$ y $M_2(\mathbf{m}') = M_2(\mathbf{m}) - \frac{1}{G(\pi_2)}$, es decir, $M(\mathbf{m}') = M(\mathbf{m}) + \frac{1}{G(\pi_1)} - \frac{1}{G(\pi_2)} > M(\mathbf{m})$. Si t es cualquier otra transición del WTS, $M_1(\mathbf{m}') = M_1(\mathbf{m})$ y $M_2(\mathbf{m}') = M_2(\mathbf{m})$, es decir, $M(\mathbf{m}') = M(\mathbf{m})$. Por lo tanto, $M(\mathbf{m}') = M(\mathbf{m})$ si $t \neq t_o$ y $M(\mathbf{m}') > M(\mathbf{m})$ si $t = t_o$. Como \mathcal{S} es vivo, la transición t_o puede dispararse un número arbitrario de veces. Por lo tanto, $M(\mathbf{m})$ puede alcanzar valores tan grandes como se quiera, es decir, al menos un lugar del camino π_2 no es limitado, lo que contradice la hipótesis inicial de limitación de \mathcal{S} .

2) Si $G(\pi_1) = G(\pi_2)$ y $t = t_o$, $M(\mathbf{m}') = M(\mathbf{m}) + \frac{1}{G(\pi_1)} - \frac{1}{G(\pi_2)} = M(\mathbf{m})$, luego $M(\mathbf{m}') = M(\mathbf{m})$ para cualquier $\mathbf{m} \xrightarrow{t} \mathbf{m}'$. Extendiendo a secuencias de disparo se obtiene que $M(\mathbf{m}) = M(\mathbf{m}_0)$ para cualquier $\mathbf{m} \in \mathbf{R}(\mathcal{S})$. Sea $C = -M(\mathbf{m}_0) = M_1(\mathbf{m}_0) - M_2(\mathbf{m}_0) \in \mathbb{Q}$. Entonces $M(\mathbf{m}) = -C$ luego $M_1(\mathbf{m}) = M_2(\mathbf{m}) + C$ para todo $\mathbf{m} \in \mathbf{R}(\mathcal{S})$. \diamond

Este resultado asegura que en WTS's vivos y limitados, todos los caminos con el mismo origen y destino deben tener la misma ganancia. Por ello, se puede hablar de ganancia entre dos transiciones. Además, la diferencia entre el mercado ponderado de dos caminos con el mismo origen y destino debe permanecer constante en todos los mercados alcanzables. Esto quiere decir que el camino con mayor mercado ponderado tiene algunas marcas permanentemente atrapadas, debido a la sincronización de los dos caminos en la transición final.

Definición 4.7 Ganancia entre transiciones.

Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado y t_i, t_o dos transiciones del mismo. La ganancia de t_i a t_o es la ganancia de cualquier camino de t_i a t_o y se denota por $G(t_i, t_o)$.

Esta definición es correcta ya que todos los caminos de t_i a t_o tienen la misma ganancia (teorema 4.6) y el WTS es fuertemente conexo (teorema 2.50) por lo que existe un camino de t_i a t_o .

Proposición 4.8 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado y t_a, t_b, t_c tres transiciones de \mathcal{N} . Se tiene que $G(t_a, t_c) = G(t_a, t_b) \cdot G(t_b, t_c)$.

Demostración:

Sean π_1 y π_2 caminos en \mathcal{N} de t_a a t_b y de t_b a t_c respectivamente (existen por el teorema 2.50). Sea $\pi_3 = \pi_1 \circ \pi_2$. Por la definición 4.7 y la proposición 4.2, $G(t_a, t_c) = G(\pi_3) = G(\pi_1) \cdot G(\pi_2) = G(t_a, t_b) \cdot G(t_b, t_c)$. \diamond

Teorema 4.9 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado, \mathbf{C} su matriz de incidencia y $\mathbf{g} \in (\mathbb{Q}^+)^{|T|}$ tal que $\mathbf{g}(i) = G(t_1, t_i)$. Entonces $\mathbf{C} \cdot \mathbf{g} = 0$.

Demostración:

Basta demostrar que $\mathbf{C}[p] \cdot \mathbf{g} = 0$ para todo $p \in P$. Como \mathcal{N} es un WTS, $\bullet p = t_i$ y $p \bullet = t_o$. Sean $r = F(t_i, p)$ y $s = F(p, t_o)$. Entonces $G(t_i, t_o) = \frac{r}{s}$ y $\mathbf{C}[p] \cdot \mathbf{g} = r\mathbf{g}(t_i) - s\mathbf{g}(t_o) = rG(t_1, t_i) - sG(t_1, t_o) = s(G(t_1, t_i) \frac{r}{s} - G(t_1, t_o)) = s(G(t_1, t_i) \cdot G(t_i, t_o) - G(t_1, t_o)) = s(G(t_1, t_o) - G(t_1, t_o)) = 0$. \diamond

Este teorema establece que las ratios de visita (ver definición 2.66) de las transiciones de un WTS quedan determinadas por la estructura de la red.

Proposición 4.10 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS y π_1, π_2 dos caminos en \mathcal{N} tales que la transición final de π_1 coincide con la transición inicial de π_2 . Sea $\pi_3 = \pi_1 \circ \pi_2$. Entonces para cualquier $\mathbf{m} \in \mathbf{R}(\mathcal{S})$ se tiene:

$$M(\pi_3, \mathbf{m}) = M(\pi_1, \mathbf{m}) + \frac{M(\pi_2, \mathbf{m})}{G(\pi_1)}$$

Demostración:

Sean $\pi_1 = (t_a p_1 t_1 \dots p_n t_b)$ y $\pi_2 = (t_b p'_1 t'_1 \dots p'_m t_c)$ los dos caminos dados. Sean r_i, s_i con $1 \leq i \leq n$ los pesos de los arcos de π_1 siguiendo la notación habitual y r'_i, s'_i con $1 \leq i \leq m$ los de π_2 . Por la definición 4.3 se tiene:

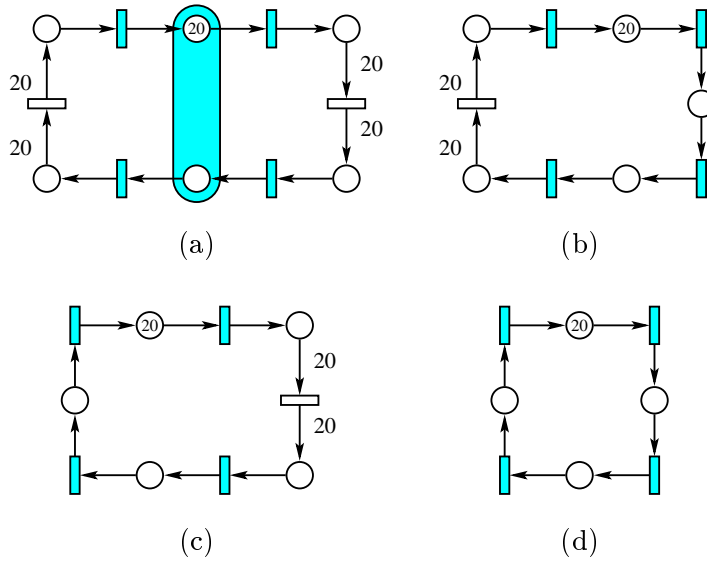


Figura 4.3: (a) Un WTS y sus (b) \mathcal{LS}_1 , (c) \mathcal{LS}_2 y (d) \mathcal{BS} .

$$M(\pi_3, \mathbf{m}) = \sum_{i=1}^n \left(\frac{\mathbf{m}[p_i]}{r_i} \prod_{j=1}^{i-1} \frac{s_j}{r_j} \right) + \sum_{i=1}^m \left(\frac{\mathbf{m}[p'_i]}{r'_i} \cdot \left(\prod_{j=1}^n \frac{s_j}{r_j} \right) \cdot \left(\prod_{j=1}^{i-1} \frac{s'_j}{r'_j} \right) \right) =$$

$$M(\pi_1, \mathbf{m}) + \left(\prod_{j=1}^n \frac{s_j}{r_j} \right) \cdot \sum_{i=1}^m \left(\frac{\mathbf{m}[p'_i]}{r'_i} \prod_{j=1}^{i-1} \frac{s'_j}{r'_j} \right) = M(\pi_1, \mathbf{m}) + \frac{M(\pi_2, \mathbf{m})}{G(\pi_1)}. \quad \diamond$$

Una vez definidos los conceptos de ganancia y marcado ponderado y conocidas sus principales propiedades, es posible extender a WTS's la técnica de descomposición de MG's del capítulo anterior. Pero ejemplos sencillos como el de la figura 4.3 demuestran que no se obtienen buenas aproximaciones. En la figura 4.3.a se tiene un ciclo con pesos muy altos en los arcos que conectan las transiciones internas (las blancas). Si los lugares del corte son los de la banda gris, los caminos entre transiciones de interfaz (las grises) tienen ganancia 1. Aplicando la técnica de descomposición de MG's, se obtendrían los sistemas agregados de las figuras 4.3.b (\mathcal{LS}_1), 4.3.c (\mathcal{LS}_2) y 4.3.d (\mathcal{BS}). Si las tasas de las transiciones son 1, al aplicar el algoritmo de aproximación de throughput se obtienen errores del orden del 50%, lo que no es aceptable para una técnica de aproximación. Además, el error aumenta si se incrementan la marcas que circulan por el ciclo y el peso de los arcos del WTS original.

El problema reside en que los pesos de los arcos inducen sincronizaciones

en las transiciones de los caminos internos a las subredes producto del corte y estas sincronizaciones desaparecen en los sistemas agregados si sólo se quiere conseguir que estos sean vivos y limitados (para lo que basta preservar la ganancia de los caminos a reducir). De esta manera, mientras en el sistema original las 20 marcas iniciales tienen que estar todas juntas para poder disparar cualquier transición interna (las blancas), en \mathcal{BS} tienen libertad total para moverse, aumentando la velocidad del sistema y por tanto el throughput de las transiciones. Los \mathcal{LS}_1 y \mathcal{LS}_2 tienen el mismo problema en la subred reducida. Al intentar aproximar el throughput de las transiciones del sistema original por medio de sistemas agregados como los calculados, se obtienen resultados muy pobres.

Por lo tanto, al descomponer un WTS no basta con construir sistemas agregados vivos y limitados. De alguna manera hay que preservar las sincronizaciones internas que contengan los caminos de las subredes a reducir. En el caso de la figura 4.3, hay que preservar en los sistemas agregados la sincronización de las 20 marcas. Este nivel de sincronización interna de un camino se reflejará en el concepto de resistencia.

La resistencia de un camino trata de aproximar el grado de sincronización interna del camino y, como se ha visto en el ejemplo anterior, es completamente independiente del concepto de ganancia. El concepto de resistencia es bastante más complejo que el de ganancia o el de marcado ponderado y para simplificar su exposición, se expondrá en dos fases. Primero se estudiará el caso de caminos aislados, que se resume en el concepto de resistencia estructural. Después se ampliará al caso de un conjunto de caminos, que se resume en el concepto general de resistencia.

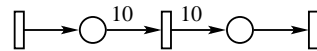


Figura 4.4: Camino de resistencia 10.

El caso de caminos aislados se puede ver con ayuda del ejemplo de la figura 4.4. Aquí se tiene un camino con ganancia 1, pero el alto peso de los arcos internos del camino inducen una sincronización en las marcas. Es necesario disparar 10 veces la transición inicial del camino para que se pueda disparar la transición intermedia, paso previo obligado para que se pueda disparar por primera vez la transición final del camino. Llegado ese momento la transición final se podrá disparar 10 veces, igualando su número medio

de disparos con el de la transición inicial.

La definición intuitiva de la resistencia de un camino es el número medio de disparos de la primera transición del camino que es necesario realizar para poder disparar la última, capturando de esta manera el nivel de sincronización interna del camino.

Definición 4.11 Resistencia estructural de un camino.

Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS, $\pi = (t_0 p_1 t_1 p_2 \dots p_n t_n)$ un camino en \mathcal{N} que conecta dos transiciones t_0 y t_n . Sean $r_i = F(t_{i-1}, p_i)$ y $s_i = F(p_i, t_i)$ con $1 \leq i \leq n$. La resistencia estructural del camino π es:

$$RE(\pi) = \max_{1 \leq j \leq n} \left\{ \prod_{i=1}^j \frac{s_i}{r_i} \right\} \in \mathbb{Q}^+ \setminus \{0\}.$$

La resistencia estructural de un camino sólo depende de los pesos de los arcos del camino y es un número racional estrictamente positivo. Formalmente, es el máximo producto parcial de los cocientes de los pesos de los arcos de salida de los lugares entre los pesos de los arcos de entrada a los mismos. La explicación de esta fórmula es la siguiente. El cociente entre los pesos de los arcos de salida y entrada de un lugar es la proporción entre el número de disparos de la transición de salida del lugar y los de la transición de entrada (si el lugar está inicialmente desmarcado). Cuando este cociente es mayor que 1, hacen falta varios disparos de la transición de entrada para poder disparar la transición de salida. Si el cociente es menor que 1 cuando se dispara la transición de entrada también es posible disparar la transición de salida. El producto parcial j es la proporción entre el número de disparos de la transición inicial t_0 y los de la transición t_j . Por lo tanto, el mayor producto parcial indica el número de disparos de la transición inicial del camino que es necesario realizar para poder disparar cualquier otra transición del camino y por lo tanto, poder disparar la transición final. De esta forma, la resistencia estructural recoge la idea intuitiva del concepto de resistencia en el caso de un camino aislado.

En el caso de la figura 4.4, la resistencia estructural del camino sería 10, puesto que el mayor producto parcial es el primero e igual a 10.

Proposición 4.12 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS, $\pi = (t_0 p_1 t_1 p_2 \dots p_n t_n)$ un camino en \mathcal{N} . Sea π_i con $1 \leq i \leq n$ el subcamino de π que une t_0 con t_i y π'_i con $1 \leq i \leq n$ el subcamino de π que une t_i con t_n . Entonces existe $1 \leq j \leq n$ tal que $\pi = \pi_j \circ \pi'_j$ y $RE(\pi_j) = 1/G(\pi_j) = RE(\pi)$.

Demostración:

Por la definición 4.11, $RE(\pi) = \max_{1 \leq j \leq n} \{\prod_{i=1}^n s_i/r_i\}$. Sea j tal que $RE(\pi) = \prod_{i=1}^j s_i/r_i$. Entonces $\pi = \pi_j \circ \pi_j'$ y el camino π_j cumple las condiciones de la proposición. \diamond

Esta proposición demuestra que en todo camino de un WTS hay una transición en la que se alcanza la resistencia estructural del camino. Este resultado será importante a la hora de reducir un camino preservando su resistencia.

En el caso de la resistencia estructural no se tiene en cuenta el marcado inicial de los lugares del camino. Esto es debido a que aunque las marcas que pueda haber inicialmente hacen que la transición final se pueda disparar antes, éste no es un comportamiento estacionario. Por ejemplo, en la figura 4.4, si el primer lugar del camino tiene 4 marcas, sólo es necesario disparar 6 veces la primera transición del camino para completar las 10 marcas necesarias para poder disparar la transición intermedia y por lo tanto 10 veces la transición final. Pero a partir de este comportamiento inicial transitorio, hace falta disparar 10 veces la transición inicial para que se pueda volver a disparar la transición final. Por ello se ha hablado de número medio de disparos al introducir de forma intuitiva el concepto de resistencia.

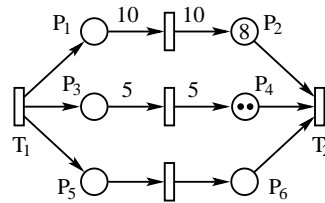


Figura 4.5: Resistencia entre transiciones.

La cuestión cambia si se tienen varios caminos conectando el mismo par ordenado de transiciones. Por ejemplo, en la figura 4.5 se tienen tres caminos entre las transiciones T_1 y T_2 . Si el modelo original es vivo y limitado, todos los caminos deben tener la misma ganancia (en el ejemplo todos tienen ganancia 1), pero pueden tener distinta resistencia estructural. Si se mira cada camino de la figura 4.5 por separado, el primer camino tiene resistencia estructural 10, el segundo 5 y el tercero 1. La diferencia con respecto al caso de un camino aislado es que ahora hay marcas que están permanentemente atrapadas en sus respectivos caminos. En efecto,

aplicando el teorema 4.6, la diferencia entre los mercados ponderados de dos caminos que conectan el mismo par ordenado de transiciones en un WTS vivo y limitado debe ser constante en todos los mercados alcanzables. Los tres caminos en cuestión tienen mercados ponderados iguales a 8, 2 y 0 respectivamente. Como el tercer camino tiene mercado ponderado igual a 0, las 8 marcas del lugar P_2 y las 2 marcas del lugar P_4 están permanentemente atrapadas en sus respectivos caminos. Una consecuencia directa de que un camino tenga marcas atrapadas es que la resistencia que ofrece el camino para que la transición final se pueda disparar es menor que su resistencia estructural (resistencia del camino aislado) y éste es un comportamiento permanente debido a que esas marcas atrapadas nunca abandonan el camino. Dicho de otra manera, cuando se tienen varios caminos conectando el mismo par ordenado de transiciones, la resistencia que ofrece cada camino para el disparo de la transición final depende no sólo de los pesos de sus arcos, sino también del mercado ponderado de todos los caminos que unen las dos transiciones. Y es el camino de menor mercado ponderado el que fija la cantidad de marcas atrapadas en cada camino individual y por lo tanto su resistencia.

En consecuencia, la resistencia de un camino en un WTS se definirá como su resistencia estructural disminuida en una cantidad igual al mercado ponderado atrapado del camino. Este mercado ponderado atrapado del camino es una cantidad que no depende únicamente de la estructura y mercado inicial del camino, sino que depende de otros caminos en el WTS, es decir, es una propiedad global a todo el WTS y no una propiedad local al camino en cuestión. Por ello no es posible dar una definición explícita formal y simple del concepto de resistencia de un camino. Esta definición formal se dará por medio de un algoritmo en la siguiente sección. Entonces será necesario tener en cuenta esta idea intuitiva de lo que se quiere reflejar con el concepto de resistencia de un camino para poder desarrollar el algoritmo.

En el caso de la figura 4.5, la resistencia que ofrece el primer camino es igual a su resistencia estructural (10) menos su mercado ponderado atrapado (8) dando como resultado una resistencia de 2. El mismo cálculo para el segundo camino da una resistencia de 3 y para el tercero 1.

Por lo tanto, el concepto de mercado ponderado de un camino es necesario en la técnica de descomposición estructural de WTS's para dos propósitos. Por un lado, para generalizar el concepto de suma de marcas empleado en MG's y por otro lado, para calcular las marcas que eventualmente pueden estar atrapadas en algunos caminos y así poder estimar la

resistencia de los mismos.

A continuación se va a demostrar una proposición que relaciona la resistencia estructural de un camino con la de dos subcaminos suyos, lo que permite el cálculo eficiente de la resistencia estructural de cualquier camino.

Proposición 4.13 *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS, π_1 y π_2 dos caminos en \mathcal{N} tales que la transición final de π_1 es la transición inicial de π_2 . Sea $\pi_3 = \pi_1 \circ \pi_2$. Se tiene que:*

$$RE(\pi_3) = \max \left\{ RE(\pi_1), \frac{RE(\pi_2)}{G(\pi_1)} \right\}.$$

Demostración:

Sean $\pi_1 = (t_a p_1 t_1 \dots p_n t_b)$ y $\pi_2 = (t_b p'_1 t'_1 \dots p'_m t_c)$ los dos caminos dados. Sean r_i, s_i con $1 \leq i \leq n$ los pesos de los arcos de π_1 siguiendo la notación habitual y r_{n+i}, s_{n+i} con $1 \leq i \leq m$ los de π_2 . Por las definiciones 4.11 y 4.1 se tiene:

$$\begin{aligned} RE(\pi_3) &= \max_{j \in [1, n+m]} \left\{ \prod_{i=1}^j \frac{s_i}{r_i} \right\} = \max \left\{ \max_{j \in [1, n]} \left\{ \prod_{i=1}^j \frac{s_i}{r_i} \right\}, \max_{j \in [n+1, n+m]} \left\{ \prod_{i=1}^j \frac{s_i}{r_i} \right\} \right\} = \\ &= \max \left\{ RE(\pi_1), \prod_{i=1}^n \frac{s_i}{r_i} \cdot \max_{j \in [1, m]} \left\{ \prod_{i=1}^j \frac{s_{n+i}}{r_{n+i}} \right\} \right\} = \max \left\{ RE(\pi_1), \frac{RE(\pi_2)}{G(\pi_1)} \right\}. \quad \diamond \end{aligned}$$

Otro resultado importante es el que relaciona la resistencia estructural de un ciclo con su marcado ponderado.

Proposición 4.14 *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado. Sea π un ciclo en \mathcal{N} con transición inicial y final t . Entonces $M(\pi, \mathbf{m}_0) \geq RE(\pi)$.*

Demostración:

Sean $\{p_i\}_{i=1}^n$ y $\{t_i\}_{i=1}^n$ los lugares y transiciones del ciclo y r_i, s_i con $1 \leq i \leq n$ los pesos de los arcos de π (r_i para arcos de transición a lugar y s_i para arcos de lugar a transición). Por la definición 4.11 y la proposición 4.12, sea i el entero tal que $RE(\pi) = \prod_{j=1}^i \frac{s_j}{r_j}$. Se observará el disparo de la transición t_i . Por ser el WTS vivo, a partir de \mathbf{m}_0 existe una secuencia de disparo σ tal que si $\mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$ entonces t_i está sensibilizada en \mathbf{m} . Entonces $\mathbf{m}[p_i] \geq s_i$ por lo que $M(\pi, \mathbf{m}) \geq \frac{\mathbf{m}[p_i]}{r_i} \prod_{j=1}^{i-1} \frac{s_j}{r_j} \geq \prod_{j=1}^i \frac{s_j}{r_j} = RE(\pi)$. Ahora, por el apartado 2 del teorema 4.5, $M(\pi, \mathbf{m}_0) = M(\pi, \mathbf{m})$ luego $M(\pi, \mathbf{m}_0) \geq RE(\pi)$. \diamond

La técnica estructural de descomposición de WTS's se basa en añadir al modelo original una serie de nodos con el objeto de resumir el comportamiento de cada subred producto del corte. Algunos de estos nodos son lugares implícitos (que no modifican el comportamiento del modelo). Por ello, es necesario estudiar las principales propiedades de los lugares implícitos en WTS's. Con las propiedades que se han demostrado sobre la ganancia y el mercado ponderado en WTS's se pueden dar ahora estos resultados.

En la sección 2.2.5 se introducen las definiciones de lugares implícitos (IP), estructuralmente implícitos (SIP), lugares estructuralmente implícitos marcados (MSIP) así como sus principales características. Ahora se van a estudiar características particulares de los lugares implícitos en relación con los WTS's.

Para la técnica de descomposición es suficiente que los lugares implícitos tengan una única transición de entrada y una única de salida. Por ello se denotarán como TT-MSIP's (MSIP's de transición a transición) al igual que en MG's. De esta manera se logra que el sistema resultante de añadir estos TT-MSIP siga siendo un WTS.

Propiedad 4.15 *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle P, T, F, W \rangle$ un WTS y $p \in P$ con $\bullet p = t_i$ y $p \bullet = t_o$. Sean $r = W(t_i, p)$, $s = W(p, t_o)$ y $d \in \mathbb{N}$ de forma que $r = r'd$ y $s = s'd$. Sea $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m}'_0 \rangle$ el sistema resultante de sustituir en \mathcal{S} los pesos $W(t_i, p)$ por r' y $W(p, t_o)$ por s' y el marcado inicial $\mathbf{m}_0[p]$ por $\lfloor \mathbf{m}_0[p]/d \rfloor$ (división entera, ver notación 2.6). Entonces se tiene que \mathcal{S} es equivalente a \mathcal{S}' en el sentido de que $\text{RG}(\mathcal{S}) \cong \text{RG}(\mathcal{S}')$ (los grafos de alcanzabilidad son isomorfos).*

Demostración:

Si $d = 1$ no hay simplificación y el resultado es evidente. Por lo tanto se estudiará el caso $d > 1$.

$\text{RG}(\mathcal{S})$ es un grafo dirigido cuyo conjunto de vértices es $\text{R}(\mathcal{S})$ y cuyo conjunto de aristas son los pares $(\mathbf{m}, \mathbf{m}_2)$ tales que $\mathbf{m}_1, \mathbf{m}_2 \in \text{R}(\mathcal{S})$ y existe $t \in T$ tal que $\mathbf{m}_1 \xrightarrow{t} \mathbf{m}_2$ en \mathcal{S} . Para establecer el isomorfismo buscado basta con dar una biyección $f : \text{R}(\mathcal{S}) \rightarrow \text{R}(\mathcal{S}')$ que preserve los disparos de transiciones, es decir, debe cumplir que para cualesquiera $\mathbf{m}_1, \mathbf{m}_2 \in \text{R}(\mathcal{S})$ y $t \in T$, se tenga que $\mathbf{m}_1 \xrightarrow{t} \mathbf{m}_2$ en \mathcal{S} si y sólo si $f(\mathbf{m}_1) \xrightarrow{t} f(\mathbf{m}_2)$ en \mathcal{S}' .

Sea $f : \text{R}(\mathcal{S}) \rightarrow \text{R}(\mathcal{S}')$ dado por $f(\mathbf{m}) = \mathbf{m}'$ con $\mathbf{m}'[q] = \mathbf{m}[q]$ si $q \neq p$ y $\mathbf{m}'[p] = \lfloor \mathbf{m}[p]/d \rfloor$ (división entera). Antes de demostrar que f es la biyección buscada, observar que existe $0 \leq c < d$ tal que $\mathbf{m}[p] \equiv c \pmod{d}$ para

todo $\mathbf{m} \in R(\mathcal{S})$. En efecto, por el algoritmo de la división (teorema 2.5), $\mathbf{m}_0[p] = m'd + c$ con $m' \geq 0$ y $0 \leq c < d$. Entonces $\mathbf{m}_0[p] \equiv c \pmod{d}$, y por definición de d , $r \equiv s \equiv 0 \pmod{d}$. Ahora, $\mathbf{m}[p]$ sólo se modifica ante el disparo de t_i o t_o , pero en cualquier caso no se modifica su resto al dividir por d (ya que $r \equiv s \equiv 0 \pmod{d}$), luego $\mathbf{m}[p] \equiv c \pmod{d}$ para todo $\mathbf{m} \in R(\mathcal{S})$. Dicho de otra forma, las c marcas (si $c > 0$) están permanentemente atrapadas en el lugar p y se pueden eliminar. Se procederá a demostrar que f cumple las propiedades deseadas. Hay que tener en cuenta que la única diferencia entre \mathbf{m} y $f(\mathbf{m})$ está en el marcado del lugar p para todo $\mathbf{m} \in R(\mathcal{S})$.

1) Ver que $\mathbf{m}_1 \xrightarrow{t} \mathbf{m}_2$ en $\langle \mathcal{N}, \mathbf{m}_1 \rangle$ si y sólo si $f(\mathbf{m}_1) \xrightarrow{t} f(\mathbf{m}_2)$ en $\langle \mathcal{N}', f(\mathbf{m}_1) \rangle$. Si $t \neq t_i$ y $t \neq t_o$, el resultado es evidente ya que $p \notin \bullet t \cup t \bullet$ y por lo tanto $\mathbf{m}_1[q] = f(\mathbf{m}_1)[q]$ y $\mathbf{m}_2[q] = f(\mathbf{m}_2)[q]$ para todo $q \in \bullet t \cup t \bullet$.

Si $t = t_i$ y $t_i \neq t_o$, $\mathbf{m}_1 \xrightarrow{t_i} \mathbf{m}_2$ en $\langle \mathcal{N}, \mathbf{m}_1 \rangle \Leftrightarrow t_i$ sensibilizada en $\langle \mathcal{N}, \mathbf{m}_1 \rangle$ y $\mathbf{m}_2[p] = \mathbf{m}_1[p] + r \Leftrightarrow t_i$ sensibilizada en $\langle \mathcal{N}', f(\mathbf{m}_1) \rangle$ (ya que $p \notin \bullet t_i$) y $f(\mathbf{m}_2)[p]d + c = f(\mathbf{m}_1)[p]d + c + r'd$ (por definición de f y r) $\Leftrightarrow t_i$ sensibilizada en $\langle \mathcal{N}', f(\mathbf{m}_1) \rangle$ y $f(\mathbf{m}_2)[p] = f(\mathbf{m}_1)[p] + r' \Leftrightarrow f(\mathbf{m}_1) \xrightarrow{t_i} f(\mathbf{m}_2)$ en $\langle \mathcal{N}', f(\mathbf{m}_1) \rangle$.

Si $t = t_o$ y $t_i \neq t_o$, $\mathbf{m}_1 \xrightarrow{t_o} \mathbf{m}_2$ en $\langle \mathcal{N}, \mathbf{m}_1 \rangle \Leftrightarrow \mathbf{m}_1[p] \geq s$ y $\mathbf{m}_2[p] = \mathbf{m}_1[p] - s \Leftrightarrow f(\mathbf{m}_1)[p]d + c \geq s'd$ y $f(\mathbf{m}_2)[p]d + c = f(\mathbf{m}_1)[p]d + c - s'd$ (por definición de f y s) $\Leftrightarrow f(\mathbf{m}_1)[p] \geq s'$ (por ser $c < d$) y $f(\mathbf{m}_2)[p] = f(\mathbf{m}_1)[p] - s' \Leftrightarrow f(\mathbf{m}_1) \xrightarrow{t_o} f(\mathbf{m}_2)$ en $\langle \mathcal{N}', f(\mathbf{m}_1) \rangle$.

Si $t = t_i = t_o$, $\mathbf{m}_1 \xrightarrow{t_o} \mathbf{m}_2$ en $\langle \mathcal{N}, \mathbf{m}_1 \rangle \Leftrightarrow \mathbf{m}_1[p] \geq s$ y $\mathbf{m}_2[p] = \mathbf{m}_1[p] + r - s \Leftrightarrow f(\mathbf{m}_1)[p]d + c \geq s'd$ y $f(\mathbf{m}_2)[p]d + c = f(\mathbf{m}_1)[p]d + c + r'd - s'd$ (por definición de f , r y s) $\Leftrightarrow f(\mathbf{m}_1)[p] \geq s'$ (por ser $c < d$) y $f(\mathbf{m}_2)[p] = f(\mathbf{m}_1)[p] + r' - s' \Leftrightarrow f(\mathbf{m}_1) \xrightarrow{t_o} f(\mathbf{m}_2)$ en $\langle \mathcal{N}', f(\mathbf{m}_1) \rangle$.

2) Ver que f está bien definida, es decir, $\mathbf{m} \in R(\mathcal{S})$ si y sólo si $f(\mathbf{m}) \in R(\mathcal{S}')$. El apartado anterior puede extenderse (por inducción sobre la longitud de secuencias de disparo) al siguiente resultado; para todo $\sigma \in L(\mathcal{N}, \mathbf{m}_1)$, $\mathbf{m}_1 \xrightarrow{\sigma} \mathbf{m}_2$ en $\langle \mathcal{N}, \mathbf{m}_1 \rangle$ si y sólo si $f(\mathbf{m}_1) \xrightarrow{\sigma} f(\mathbf{m}_2)$ en $\langle \mathcal{N}', f(\mathbf{m}_1) \rangle$. Y este resultado es válido para cualquier marcado \mathbf{m}_1 de \mathcal{N} , en particular para \mathbf{m}_0 . Ahora, $\mathbf{m} \in R(\mathcal{S}) \Leftrightarrow \exists \sigma \in L(\mathcal{S})$ tal que $\mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$ en $\mathcal{S} \Leftrightarrow f(\mathbf{m}_0) \xrightarrow{\sigma} f(\mathbf{m})$ en $\mathcal{S}' \Leftrightarrow f(\mathbf{m}) \in R(\mathcal{S}')$ y $\sigma \in L(\mathcal{S}')$ (porque $f(\mathbf{m}_0) = \mathbf{m}'_0$). \diamond

Esta propiedad asegura que en un WTS se pueden simplificar los pesos de los arcos de entrada y salida de cualquier lugar por cualquier divisor común de los mismos, en particular por su máximo común divisor, obteniendo de esta manera pesos que son primos entre sí. En cuanto al marcado

del lugar en el que se realiza la simplificación, se puede hacer la división entera del mercado que tenía por el factor de simplificación de los pesos de sus arcos ya que el posible resto de marcas (el valor c de la demostración) están permanentemente atrapadas en el lugar y no juegan ningún papel en el funcionamiento del WTS (únicamente en la codificación de los mercados alcanzables). En consecuencia, se puede suponer a partir de aquí que el WTS original es tal que los pesos de los arcos de entrada y salida de cualquier lugar son primos entre sí. De esta manera se eliminan marcas de lugares que no juegan ningún papel en el funcionamiento del WTS y que se deben únicamente a errores de modelado por parte del analista. El siguiente resultado es el equivalente al anterior, pero multiplicando los pesos de los arcos.

Corolario 4.16 *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle P, T, F, W \rangle$ un WTS y $p \in P$ con $\bullet p = t_i$ y $p \bullet = t_o$. Sean $r = W(t_i, p)$, $s = W(p, t_o)$ y $d, c \in \mathbb{Z}$ con $0 \leq c < d$. Sea $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m}'_0 \rangle$ el sistema resultante de sustituir en \mathcal{S} los pesos $W(t_i, p)$ por rd y $W(p, t_o)$ por sd y el mercado inicial $\mathbf{m}_0[p]$ por $\mathbf{m}_0[p]d + c$. Entonces se tiene que \mathcal{S} es equivalente a \mathcal{S}' en el sentido de que $\text{RG}(\mathcal{S}) \cong \text{RG}(\mathcal{S}')$ (los grafos de alcanzabilidad son isomorfos).*

Demostración:

Evidente, aplicando la propiedad 4.15 a \mathcal{S}' con factor de simplificación d . Al ser $0 \leq c < d$ se tiene que $\lfloor \mathbf{m}'_0[p]/d \rfloor = \lfloor (\mathbf{m}_0[p]d + c)/d \rfloor = \mathbf{m}_0[p]$. \diamond

Teorema 4.17 *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS y t, t' dos transiciones del mismo tales que existe un camino π en \mathcal{N} de t a t' con $G(\pi) = \frac{g_1}{g_2}$. Sea $p \notin P$ un lugar con $\bullet p = t$ y $p \bullet = t'$ tal que $W(t, p) = g_1$ y $W(p, t') = g_2$. Entonces p es un TT-MSIP.*

Demostración:

Sean $\{p_i\}_{i=1}^n$ y $\{t_i\}_{i=0}^n$ los lugares y transiciones de π (se asume que $t_0 = t$ y $t_n = t'$). Sean $r_i = W(t_{i-1}, p_i)$, $s_i = W(p_i, t_i)$ con $1 \leq i \leq n$ los pesos de los arcos de π . Por la definición 4.1 de ganancia, se tiene que $\frac{g_1}{g_2} = \prod_{i=1}^n \frac{r_i}{s_i}$. Por la propiedad 4.15 se puede suponer, sin pérdida de generalidad, que $g_1 = \prod_{i=1}^n r_i$ y $g_2 = \prod_{i=1}^n s_i$. Entonces, el vector de incidencia \mathbf{l}_p del lugar p es tal que $\mathbf{l}_p[t_0] = g_1$, $\mathbf{l}_p[t_n] = -g_2$ y $\mathbf{l}_p[t] = 0$ para cualquier otra transición $t \neq t_0, t_n$. Por la definición 2.61 de MSIP, hay que encontrar un vector $\mathbf{y} \geq \mathbf{0}$ tal que $\mathbf{y} \cdot \mathbf{C} = \mathbf{l}_p$ (\mathbf{C} es la matriz de incidencia de \mathcal{N}).

El vector \mathbf{y} tiene una componente por cada lugar de \mathcal{N} . Se denotará cada componente por el lugar al que corresponde. Sea \mathbf{y} tal que $\mathbf{y}[p] = 0$ si $p \notin \pi$ y $\mathbf{y}[p_i] = (\prod_{j=1}^{i-1} s_j) \cdot (\prod_{j=i+1}^n r_j)$ para $1 \leq i \leq n$ (se asume que los productos vacíos valen 1). Por construcción $\mathbf{y} \geq \mathbf{0}$. Ver que $\mathbf{y} \cdot \mathbf{C} = \mathbf{l}_p$.

Como \mathbf{y} sólo tiene componentes no nulas en los lugares de π , basta fijarse en las filas de \mathbf{C} correspondientes a los lugares p_i con $1 \leq i \leq n$.

Si $t \notin \pi$, entonces $\mathbf{C}[p_i, t] = 0$ para $1 \leq i \leq n$, es decir, $\mathbf{y} \cdot \mathbf{C}[t] = 0$.

Si $t = t_i$ con $1 \leq i < n$, $\mathbf{C}[p_i, t] = -s_i$, $\mathbf{C}[p_{i+1}, t] = r_{i+1}$ y $\mathbf{C}[p_j, t] = 0$ si $j \neq i, i+1$ y $1 \leq j \leq n$. Entonces $\mathbf{y} \cdot \mathbf{C}[t] = r_{i+1}\mathbf{y}[p_{i+1}] - s_i\mathbf{y}[p_i] = r_{i+1}(\prod_{j=1}^i s_j)(\prod_{j=i+2}^n r_j) - s_i(\prod_{j=1}^{i-1} s_j)(\prod_{j=i+1}^n r_j) = (\prod_{j=1}^i s_j)(\prod_{j=i+1}^n r_j) - (\prod_{j=1}^i s_j)(\prod_{j=i+1}^n r_j) = 0$.

Si $t = t_0$, entonces $\mathbf{C}[p_1, t] = r_1$ y $\mathbf{C}[p_i, t] = 0$ para $1 < i \leq n$, por lo que $\mathbf{y} \cdot \mathbf{C}[t] = r_1\mathbf{y}[p_1] = r_1(\prod_{j=2}^n r_j) = \prod_{j=1}^n r_j = g_1$.

Si $t = t_n$, entonces $\mathbf{C}[p_n, t] = -s_n$ y $\mathbf{C}[p_i, t] = 0$ para $1 \leq i < n$, por lo que $\mathbf{y} \cdot \mathbf{C}[t] = -s_n\mathbf{y}[p_n] = -s_n(\prod_{j=1}^{n-1} s_j) = -\prod_{j=1}^n s_j = -g_2$.

Por lo tanto, se ha demostrado que $\mathbf{y} \cdot \mathbf{C} = \mathbf{l}_p$. \diamond

Este teorema da la forma de añadir lugares TT-MSIP a un WTS. Se puede hacer entre transiciones conectadas por un camino en la red original. Basta con calcular la ganancia de un camino que una las transiciones que se quieren conectar por medio del lugar implícito. Como la ganancia es un número racional positivo se toma su numerador y denominador y estos son los pesos que hay que poner a los arcos del lugar.

Los siguientes resultados resuelven el problema del cálculo de un marcado inicial para que un TT-MSIP sea implícito.

Proposición 4.18 *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS y t_i, t_o dos transiciones del mismo tales que existe un camino π en \mathcal{N} de t_i a t_o con $G(\pi) = \frac{g_1}{g_2}$. Sea $p \notin P$ un TT-MSIP con $W(t_i, p) = g_1$, $W(p, t_o) = g_2$ y $\mathbf{m}_0[p] = \lfloor g_1 \cdot M(\pi, \mathbf{m}_0) \rfloor$. Entonces p es implícito.*

Demostración:

Sean r_i, s_i con $1 \leq i \leq n$ los pesos de los arcos de π siguiendo la notación habitual ($\{p_i\}_{i=1}^n$ son los lugares de π). Sea $\langle \mathcal{N}_p, \mathbf{m}_0^p \rangle$ el sistema que resulta al añadir a $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ el lugar $p \notin P$ del enunciado. Se demostrará que p es implícito, es decir, $L(\mathcal{N}, \mathbf{m}_0) = L(\mathcal{N}_p, \mathbf{m}_0^p)$.

Por la propiedad 4.15, se puede suponer sin pérdida de generalidad que $g_1 = \prod_{i=1}^n r_i$ y $g_2 = \prod_{i=1}^n s_i$.

Es evidente que $L(\mathcal{N}_p, \mathbf{m}_0^p) \subseteq L(\mathcal{N}, \mathbf{m}_0)$ porque para construir $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ a partir de $\langle \mathcal{N}_p, \mathbf{m}_0^p \rangle$ sólo se quita una restricción de disparo para t_o (la del lugar p).

Para demostrar que $L(\mathcal{N}, \mathbf{m}_0) \subseteq L(\mathcal{N}^q, \mathbf{m}_0^q)$ basta ver que p nunca es el único lugar que impide el disparo de t_o (porque ésta es la única diferencia entre $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ y $\langle \mathcal{N}_p, \mathbf{m}_0^p \rangle$). Formalmente, hay que ver que si $\mathbf{m}^p \in R(\mathcal{N}_p, \mathbf{m}_0^p)$ es tal que para todo $q \in \bullet t_o$ con $q \neq p$ se cumple $\mathbf{m}^p[q] \geq W(q, t_o)$, entonces se tiene que $\mathbf{m}^p[p] \geq g_2$. Sea π' el camino $t_i p t_o$. π y π' son dos caminos en $\langle \mathcal{N}_p, \mathbf{m}_0^p \rangle$ de t_i a t_o . Por construcción de $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, $\langle \mathcal{N}_p, \mathbf{m}_0^p \rangle$ y π' se tiene que $G(\pi) = G(\pi')$ y $M(\pi, \mathbf{m}_0) = M(\pi, \mathbf{m}_0^p) = M(\pi', \mathbf{m}_0^p)$. Por la proposición 4.4, se tiene que $M(\pi, \mathbf{m}^p) = M(\pi', \mathbf{m}^p)$ para todo $\mathbf{m}^p \in R(\mathcal{N}_p, \mathbf{m}_0^p)$. Si $\mathbf{m}^p \in R(\mathcal{N}_p, \mathbf{m}_0^p)$ es tal que para todo $q \in \bullet t_o$ con $q \neq p$ cumple $\mathbf{m}^p[q] \geq W(q, t_o)$, en particular $\mathbf{m}^p[p_n] \geq s_n$ (p_n es el último lugar del camino π). En este caso $M(\pi, \mathbf{m}^p) \geq 1/G(\pi') = \frac{g_2}{g_1}$, es decir, $M(\pi', \mathbf{m}_0^p) = M(\pi, \mathbf{m}_0^p) \geq \frac{g_2}{g_1}$. Como $M(\pi', \mathbf{m}^q[q]) = \frac{\mathbf{m}^q[q]}{g_1}$, entonces $\mathbf{m}^p[p] \geq g_2$, luego p es implícito. \diamond

Esta proposición demuestra que para encontrar un mercado inicial que haga implícito a un TT-MSIP en un WTS, basta calcular el mercado ponderado de un camino cualquiera que una las transiciones, multiplicarlo por el numerador de la ganancia y redondear por defecto.

En el caso de WTS's vivos y limitados todos los caminos que unen un mismo par ordenado de transiciones tienen la misma ganancia (teorema 4.6), por lo que se puede elegir cualquiera para los cálculos. En este caso se obtendrá un TT-MSIP con menor mercado inicial si el camino elegido para efectuar los cálculos es el de menor mercado ponderado.

Teorema 4.19 *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado y t_i, t_o dos transiciones de \mathcal{N} tales que $G(t_i, t_o) = \frac{g_1}{g_2}$. Sea $p \notin P$ un TT-MSIP con $W(t, p) = g_1$ y $W(p, t') = g_2$. Un mercado suficiente para que p sea implícito es:*

$$\mathbf{m}_0[p] = \left\lfloor g_1 \cdot \min \left\{ M(\pi, \mathbf{m}_0) \mid \pi \in \mathcal{P}(t, t') \right\} \right\rfloor$$

Demostración:

Basta con aplicar la proposición 4.18 al camino entre t_i y t_o de menor mercado ponderado inicial. Siempre existe un camino de menor mercado ponderado ya que, por el apartado 2 del teorema 4.5, los caminos con ciclos tienen mayor mercado ponderado que los caminos simples que contienen y sólo hay un número finito de caminos simples entre dos transiciones. \diamond

A diferencia del caso de MG's, en WTS's es mucho más complicado calcular el mínimo marcado inicial que hay que poner a un TT-MSIP para hacerlo implícito. Hay que tener en cuenta que el marcado ponderado de un camino no da información sobre la distribución interna de las marcas en los lugares y por lo tanto, de la posibilidad o no de disparar sus transiciones. Puede ocurrir que el camino de menor marcado ponderado entre t_i y t_o nunca llegue a perder todas sus marcas en todos los estados alcanzables del WTS lo que haría posible reducir $\mathbf{m}_0[p]$ sin que p deje de ser implícito. Una forma de calcular este marcado mínimo consiste en tomar el camino de menor marcado ponderado de t_i a t_o y con él calcular el menor marcado ponderado alcanzado en todos los estados alcanzables del sistema. Con ese valor (que podría ser estrictamente mayor que 0) y el numerador de la ganancia se calcula el marcado mínimo de p que lo hace implícito. Obviamente este cálculo no tiene sentido desde el punto de vista práctico por su elevada complejidad. Por lo tanto, habrá que conformarse con poner a los TT-MSIP un marcado inicial que los haga implícitos, aunque no sea el mínimo en algunos casos.

Corolario 4.20 *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado y t_i, t_o dos transiciones de \mathcal{N} tales que $G(t_i, t_o) = \frac{g_1}{g_2}$. Sea $p \notin P$ un TT-MSIP con $W(t_i, p) = g_1$ y $W(p, t_o) = g_2$. Entonces p es implícito para cualquier marcado inicial $\mathbf{m}_0[p] \geq \left\lceil g_1 \cdot \min \left\{ M(\pi, \mathbf{m}_0) \mid \pi \in \mathcal{P}(t_i, t_o) \right\} \right\rceil$.*

Demostración:

El lugar p es implícito para el $\mathbf{m}_0[p]$ del teorema 4.19. Además, por el apartado 2 del teorema 4.6, un eventual exceso de marcas en p permanece allí atrapado en todos los marcados alcanzables (ya que el camino de menor marcado ponderado entre t_i y t_o no cambia) por lo que aumentando $\mathbf{m}_0[p]$ no cambia $\text{RG}(\mathcal{N}, \mathbf{m}_0)$ excepto en el número de marcas de p . \diamond

El cálculo del marcado inicial de un TT-MSIP para hacerlo implícito puede hacerse aplicando una ligera modificación del algoritmo de Floyd (ver algoritmo 3.7). Para ello será necesario calcular la ganancia entre dos transiciones cualesquiera del WTS's. En la siguiente sección se abordarán estos problemas.

En la reducción de WTS's se calculan conjuntos de lugares implícitos que resumen el comportamiento de ciertas subredes del WTS. Es interesante que el conjunto de lugares implícitos sea lo más pequeño posible para reducir la estructura de los sistemas agregados y por lo tanto la complejidad de su estudio. Para poder eliminar lugares implícitos que aportan la misma

información que otros lugares implícitos del conjunto se necesita el siguiente teorema.

Teorema 4.21 *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado, t_i, t_j y t_k tres transiciones de \mathcal{N} y p_{ik}, p_{kj} y p_{ij} tres TT-MSIP's de t_i a t_k , de t_k a t_j y de t_i a t_j respectivamente, con el mercado inicial indicado en el teorema 4.19. Si existe un camino en \mathcal{N} de t_i a t_j de mercado ponderado mínimo pasando por t_k , entonces p_{ij} es implícito respecto a p_{ik}, p_{kj} (ver definición 2.62).*

Demostración:

Sea π_3 el camino de mercado ponderado mínimo de t_i a t_j pasando por t_k . Se denotarán por π_1 y π_2 a los subcaminos de π_3 que van de t_i a t_k y de t_k a t_j respectivamente. Es evidente que π_1 y π_2 son caminos de mercado ponderado mínimo de t_i a t_k y de t_k a t_j respectivamente.

Sean r_i, s_i con $1 \leq i \leq n$ los pesos de los arcos de π_1 siguiendo la notación habitual y r'_i, s'_i con $1 \leq i \leq m$ los de π_2 . Sean $r = \prod_{i=1}^n r_i$, $s = \prod_{i=1}^n s_i$, $r' = \prod_{i=1}^m r'_i$ y $s' = \prod_{i=1}^m s'_i$ para simplificar notación.

Por la propiedad 4.15, se puede suponer sin pérdida de generalidad que $W(t_i, p_{ik}) = r$, $W(p_{ik}, t_k) = s$ y $\mathbf{m}_0[p_{ik}] = \lfloor rM(\pi_1, \mathbf{m}_0) \rfloor = rM(\pi_1, \mathbf{m}_0)$ (ya que $rM(\pi_1, \mathbf{m}_0) \in \mathbf{Z}^+$). Análogamente, $W(t_k, p_{kj}) = r'$, $W(p_{kj}, t_j) = s'$ y $\mathbf{m}_0[p_{kj}] = r'M(\pi_2, \mathbf{m}_0)$. Por último, $W(t_i, p_{ij}) = rr'$, $W(p_{ij}, t_j) = ss'$ y $\mathbf{m}_0[p_{ij}] = rr'M(\pi_3, \mathbf{m}_0)$.

Sea π el camino $(t_i p_{ik} t_k p_{kj} t_j)$. Por el teorema 4.6, $G(\pi) = G(\pi_3) = \frac{rr'}{ss'}$ y $M(\pi, \mathbf{m}_0) = \frac{\mathbf{m}_0[p_{ik}]}{r} + \frac{s\mathbf{m}_0[p_{kj}]}{rr'} = \frac{rM(\pi_1, \mathbf{m}_0)}{r} + \frac{sr'M(\pi_2, \mathbf{m}_0)}{rr'} = M(\pi_1, \mathbf{m}_0) + \frac{M(\pi_2, \mathbf{m}_0)}{G(\pi_1)} = M(\pi_3, \mathbf{m}_0)$. Luego π es otro camino de mercado ponderado mínimo entre t_i y t_j . Por el teorema 4.19, el lugar p_{ij} también es implícito respecto al camino π . \diamond

Este teorema no es tan fuerte como el teorema 3.5 en el caso de MG's. En MG's se tenía una equivalencia. En WTS's sólo se tiene una condición suficiente para poder eliminar un determinado TT-MSIP. En cualquier caso, este resultado es suficiente para poder reducir el número de lugares implícitos.

El recíproco del teorema 4.21 no es cierto, como puede verse en la figura 4.6. Aquí se tienen las transiciones t_i, t_k y t_j (con la notación del teorema). El camino de mercado ponderado mínimo de t_i a t_k es el $(t_i p_1 t_1 p_2 t_k)$ que tiene ganancia $2/3$ y mercado ponderado $3/2$. El TT-MSIP correspondiente es el lugar p_{ik} con mercado inicial 3. El camino de mercado ponderado mínimo de t_k a t_j es el $(t_k p_3 t_2 p_4 t_j)$ que tiene ganancia $3/2$ y mercado

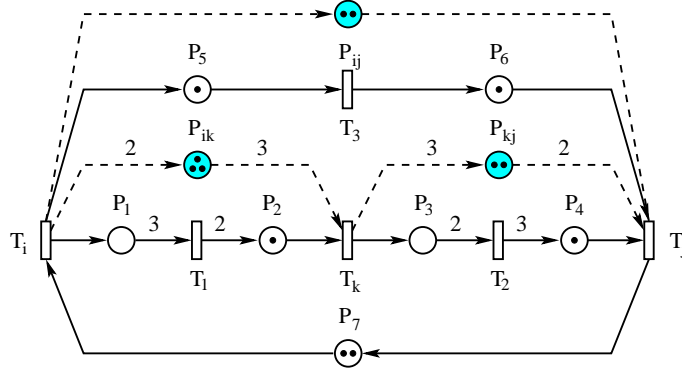


Figura 4.6: Contraejemplo para el recíproco del teorema 4.21.

ponderado $2/3$. El TT-MSIP correspondiente es el lugar p_{kj} con marcado inicial 2. Por último, el camino de marcado ponderado mínimo de t_i a t_j es el $(t_i p_5 t_3 p_6 t_j)$ que tiene ganancia 1 y marcado ponderado 2. El TT-MSIP correspondiente es el lugar p_{ij} con marcado inicial 2. Notar que el camino $(t_i p_{ik} t_k p_{kj} t_j)$ es de ganancia 1 y marcado ponderado $5/2$, por lo que el TT-MSIP que le corresponde es el mismo lugar p_{ij} , es decir, el lugar p_{ij} es implícito respecto a los lugares p_{ik} y p_{kj} . Sin embargo, el camino de menor marcado ponderado de t_i a t_j no pasa por t_k .

4.3 Descomposición estructural de WTS's

Una vez estudiadas las principales características estructurales de los WTS's, incluidos algunos resultados respecto a los lugares implícitos en la clase de los WTS's, se va a desarrollar la técnica estructural de descomposición. La descomposición de WTS's sigue la misma filosofía que la de MG's del capítulo anterior. Un WTS vivo y limitado se descompone en K subredes por medio de un corte B definido sobre lugares (como en MG's). A partir del corte y de las subredes se construyen K sistemas de bajo nivel \mathcal{LS}_i con $1 \leq i \leq K$ y un sistema de alto nivel o esqueleto básico \mathcal{BS} . Cada \mathcal{LS}_i contiene completamente una de las subredes producto del corte y agregaciones del resto de las subredes. El \mathcal{BS} se compone de las agregaciones de todas las subredes.

La forma de agregar subredes en WTS's es una extensión de la de MG's,

en el sentido de que la técnica de agregación que se va a exponer en esta sección aplicada a un MG da exactamente los mismos sistemas agregados que los que se conseguían con la técnica del capítulo anterior. Ahora las agregaciones de cada subred se consiguen por medio de TT-MSIP's y algunas transiciones inmediatas que intentan resumir la resistencia de ciertos caminos de las subredes. Para ello son necesarias las propiedades demostradas en la sección anterior.

En WTS's (a diferencia del caso de los MG's) no es cierto que el comportamiento de los \mathcal{LS}_i y \mathcal{BS} en cuanto a marcados alcanzables y secuencias de disparo sea una proyección exacta de los del WTS original sobre los nodos preservados en cada sistema agregado. Esto sólo será cierto en algunos WTS's, por ejemplo en los MG's.

En esta memoria se incluye una mejora parcial respecto a la técnica desarrollada en [PJCS96b], que consiste en reducir la estructura de los sistemas agregados calculados en esta fase de descomposición en cuanto al número de nodos necesarios para resumir el comportamiento de las subredes. En [PJCS96b] puede ocurrir que entre los nodos calculados en la reducción de las subredes haya algunos redundantes. Por ejemplo, algunos TT-MSIP's pueden ser implícitos respecto a otros TT-MSIP's en los \mathcal{LS}_i y \mathcal{BS} y por lo tanto, pueden ser eliminados. Aquí se desarrollará la técnica de manera diferente eliminando los nodos redundantes.

Definición 4.22 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS con $\mathcal{N} = \langle P, T, F, W \rangle$. Un subconjunto de lugares $B \subseteq P$ se dice K -corte ($K \geq 2$) de \mathcal{N} si existen K subredes $\mathcal{N}_i = \langle P_i, T_i, F_i, W_i \rangle$ con $1 \leq i \leq K$ de \mathcal{N} tales que:

- i) $\bigcup_{i=1}^K T_i = T$ y $T_i \cap T_j = \emptyset$ para $1 \leq i, j \leq K, i \neq j$.
- ii) $P_i = \bullet T_i \cup T_i \bullet$ para $1 \leq i \leq K$.
- iii) $\bigcup_{i=1}^K P_i = P$ y $\bigcup_{i \neq j} (P_i \cap P_j) = B$ para $1 \leq i, j \leq K$.
- iv) $F_i = F \cap ((P_i \times T_i) \cup (T_i \times P_i))$ y $W_i = W|_{F_i}$ para $1 \leq i \leq K$.

Las transiciones del conjunto $\text{TI} = \bullet B \cup B \bullet$ se llaman transiciones de interfaz. El resto se llaman transiciones internas.

Una vez definido un corte sobre un WTS, hay que calcular los \mathcal{LS}_i y el \mathcal{BS} . Para realizar esta tarea se calcula, para cada subred producto del corte, un conjunto de lugares implícitos (concretamente TT-MSIP's) y de

transiciones inmediatas que resumen el comportamiento interno de la subred. En cada subred se pretende mantener las transiciones de interfaz, con el fin de que en la parte numérica de la técnica de aproximación sean éstas las transiciones que resuman la evolución temporal de la subred (en concreto el tiempo de respuesta). Para reducir una subred producto del corte es necesario preservar por un lado la ganancia de los caminos de la subred (para que los sistemas agregados sean vivos y limitados) y por otro lado la máxima resistencia o nivel de sincronización interna de los caminos que conectan dos transiciones de interfaz dadas (para mejorar los resultados numéricos de la técnica de aproximación de throughput). Estos nodos se añadirán al WTS original para construir de forma eficiente un sistema extendido, \mathcal{ES} . Por construcción, \mathcal{ES} tiene el mismo lenguaje de secuencias de disparo que el WTS original. El grafo de estados alcanzables de \mathcal{ES} es isomorfo al del WTS original. Sólo difieren en la codificación de los marcados debido al mayor número de lugares de \mathcal{ES} . A partir de \mathcal{ES} es inmediato construir los \mathcal{LS}_i y el \mathcal{BS} .

Por lo tanto, el primer objetivo consiste en desarrollar los algoritmos necesarios para calcular la ganancia de los caminos de cada subred y la resistencia máxima entre transiciones de interfaz. Para calcular las resistencias máximas, es necesario conocer el marcado ponderado que tiene atrapado cada camino. Para ello es necesario calcular previamente los marcados ponderados mínimos entre cualesquiera dos transiciones del WTS. Posteriormente se utilizarán estas cantidades para calcular los nodos necesarios para reducir cada subred y generar los sistemas agregados.

El primer algoritmo que se va a tratar resuelve el cálculo de ganancias de cualquier camino en el WTS, incluidos los de cada subred producto del corte. Como los WTS's que se quieren descomponer son vivos y limitados, las ganancias de dos caminos con las mismas transiciones inicial y final son iguales (teorema 4.6). Para almacenar las ganancias entre cualesquiera dos transiciones del WTS es suficiente un vector \mathbf{g} de dimensión igual al número de transiciones del WTS que asocie un valor $\mathbf{g}(i)$ a cada transición t_i de forma que la ganancia de t_i a t_j sea $\mathbf{g}(j)/\mathbf{g}(i)$. Para este algoritmo se tomará como datos de entrada la estructura del WTS en forma de grafo dirigido con pesos. Dado un WTS, se puede construir un grafo dirigido con pesos con un vértice por cada transición del WTS y una arista del vértice v_1 al vértice v_2 si y sólo si existe un lugar p_{12} en el WTS que conecta la transición correspondiente a v_1 con la transición correspondiente a v_2 . El peso asociado a esta arista será la ganancia del camino elemental $(t_1 p_{12} t_2)$, es decir,

$W(t_1, p_{12})/W(p_{12}, t_2)$. Un grafo dirigido definido de esta manera contiene toda la información sobre la estructura del WTS relativa a la ganancia entre transiciones.

Algoritmo 4.23 Cálculo de ganancias.

input: (V, A) grafo dirigido asociado a un WTS vivo y limitado.

$L : A \rightarrow \mathbb{Q}^+ \setminus \{0\}$ función de etiquetado de las aristas tal que

$L(v_1, v_2) = G(v_1, v_2)$ (ganancias de caminos elementales)

Q cola de vértices por visitar

\mathbf{B} vector de vértices visitados (inicialmente false)

\mathbf{g} vector de ganancias

$Q := \{v_1\}$

$\mathbf{B}(v_1) := \text{true}$

$\mathbf{g}(1) := 1$

while $Q \neq \emptyset$ **do**

$v := \text{primero}(Q)$

$C := \{v' \in V \mid (v, v') \in A\}$

for each $v' \in C$ **do**

if $\mathbf{B}(v') = \text{false}$ **then**

$\mathbf{B}(v') := \text{true}$

inserta v' en Q

$\mathbf{g}(v') := \mathbf{g}(v) \cdot L(v, v')$

end if

end for

end while

output: Vector \mathbf{g} tal que $\mathbf{g}(i) = G(t_1, t_i)$

Este algoritmo realiza un recorrido en anchura del grafo asociado al WTS. Como el WTS es vivo y limitado, entonces es fuertemente conexo (teorema 2.50) lo que asegura que el recorrido pasa por todos los vértices del grafo. Se puede comenzar el recorrido por cualquier vértice, por ejemplo el v_1 (correspondiente a la transición t_1). En la iteración i del bucle exterior del algoritmo se visitan los vértices del grafo conectados a v_1 por medio de un camino de i aristas. Para guardar los vértices que hay que visitar, se utiliza una estructura de datos tipo cola (Q en el algoritmo). Al visitar un vértice, se calculan todos los vértices directamente conectados con él (conjunto C del algoritmo) por medio del conjunto A de aristas del grafo. Para evitar pasos múltiples por un mismo vértice se mantiene el vector \mathbf{B} de

variables lógicas en el que se marcan los vértices ya visitados. Este vector asegura la finalización del algoritmo, que ocurrirá cuando todos los vértices hayan sido visitados. El cálculo que se pretende realizar con este algoritmo es el vector \mathbf{g} tal que $\mathbf{g}(i) = G(t_1, t_i)$. Esto se consigue por medio de las etiquetas de las aristas del grafo (función L del algoritmo). El siguiente teorema demuestra la corrección del algoritmo.

Teorema 4.24 *Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado y \mathbf{g} el vector calculado por medio del algoritmo 4.23. Si t_i y t_j son dos transiciones de $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ entonces $G(t_i, t_j) = \mathbf{g}(j)/\mathbf{g}(i)$.*

Demostración:

Hay que tener en cuenta que el vértice v_i de (V, A) se corresponde con la transición t_i de \mathcal{N} . Para cada vértice v_i sea $n(v_i)$ el número mínimo de aristas que hay que atravesar en (V, A) para ir de v_1 a v_i . Se demostrará por inducción sobre $n(v_i)$ que $\mathbf{g}(i) = G(t_1, t_i)$ para cualquier vértice v_i . Para $n(v) = 0$ es evidente el resultado porque el único vértice que cumple $n(v) = 0$ es v_1 y el algoritmo calcula $\mathbf{g}(1) = 1 = G(t_1, t_1)$. Si el resultado es cierto para los vértices v_i tales que $n(v_i) \leq n$, ver que también lo es para los vértices v_i tales que $n(v_i) = n + 1$.

En efecto, sea v_i con $n(v_i) = n + 1$ y v_j el primer vértice que provoca la primera visita a v_i en el algoritmo 4.23. Entonces, $n(v_j) = n$ y $(v_j, v_i) \in A$. En este caso, en la iteración $n + 1$ del bucle exterior del algoritmo se hace cálculo $\mathbf{g}(i) = \mathbf{g}(j) \cdot L(v_j, v_i) = \mathbf{g}(j) \cdot G(t_j, t_i)$. Por hipótesis de inducción, $\mathbf{g}(j) = G(t_1, t_j)$, luego $\mathbf{g}(i) = G(t_1, t_j) \cdot G(t_j, t_i) = G(t_1, t_i)$ (proposición 4.8) lo que demuestra la corrección del algoritmo.

Como cualquier ciclo de $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ tiene ganancia 1 (apartado 1 del teorema 4.5) y \mathcal{N} es fuertemente conexo (teorema 2.50), entonces se tiene que, para cualesquiera t_i, t_j transiciones de \mathcal{N} , $G(t_i, t_j) \cdot G(t_j, t_i) = 1$. Por lo tanto, $\frac{\mathbf{g}(j)}{\mathbf{g}(i)} = \frac{G(t_1, t_j)}{G(t_1, t_i)} = G(t_i, t_1) \cdot G(t_1, t_j) = G(t_i, t_j)$ (proposición 4.8). \diamond

Las estructuras de datos del algoritmo 4.23 son de tamaño $|A|$ o $|V|$ (cardinal de A o de V). Como el WTS es vivo y limitado, por el teorema 2.50 también es fuertemente conexo y entonces $|A| \geq |V|$ por lo que la complejidad en espacio del algoritmo 4.23 es $O(|A|)$. La complejidad en tiempo es también $O(|A|)$ porque en el peor caso hay que recorrer todas las aristas del grafo. Traduciendo a notación de redes de Petri, la complejidad en espacio y tiempo del algoritmo 4.23 es $O(|P|)$ (cardinal del conjunto de lugares del WTS).

Una vez que se tienen calculadas las ganancias entre dos transiciones cualesquiera de un WTS vivo y limitado, se está en disposición de atacar el problema de el cálculo de los marcados ponderados mínimos. Este cálculo puede hacerse con una ligera modificación del algoritmo de Floyd para el cálculo de caminos de peso mínimo entre los vértices de un grafo dirigido con pesos. El algoritmo de Floyd se basa en el cálculo del peso total de un determinado camino sumando los pesos de subcaminos suyos. En el caso del marcado ponderado, es la proposición 4.10 la que relaciona el marcado ponderado de un camino con los de sus subcaminos. Para poder aplicar esta relación entre marcados ponderados es necesario disponer de las ganancias de los subcaminos, problema ya resuelto con el algoritmo 4.23. Se aprovechará para calcular cuándo un determinado camino de marcado ponderado mínimo pasa por una transición de interfaz o no lo que permitirá disminuir la cantidad de lugares implícitos necesarios para reducir una determinada subred (de acuerdo con el teorema 4.21). Todas estas cuestiones suponen ligeras modificaciones de la función de coste con respecto al algoritmo inicial de Floyd por lo que es necesario demostrar su corrección. Y para ello se utiliza el marco general para la resolución de problemas de caminos en grafos dirigidos con pesos de [CLR90] y que se ha explicado con detalle en el capítulo 3. Para el siguiente algoritmo se utiliza el mismo grafo dirigido que en el algoritmo 4.23. En lo único que se diferencian es en las etiquetas que se asocian a las aristas del mismo. Por ello, el primer paso consiste en definir el tipo de etiquetas que se van a utilizar ahora.

Definición 4.25 Sea $S = (\mathbb{Q}^+ \times \mathbb{Q}^+ \setminus \{0\} \times \{0, 1\} \times \{0, 1\}) \cup \bar{0} \cup \bar{1}$ ($\bar{0}$ y $\bar{1}$ son dos elementos especiales). Se definen las siguientes operaciones en S :

i) *Operador extensión* \odot

$$\begin{aligned} (w_1, g_1, b_1, v_1) \odot (w_2, g_2, b_2, v_2) &= (w_1 + w_2/g_1, g_1 g_2, \max\{b_1, b_2, v_1\}, v_2), \\ (w, g, b, v) \odot \bar{0} &= \bar{0} \odot (w, g, b, v) = \bar{0}, \bar{0} \odot \bar{1} = \bar{1} \odot \bar{0} = \bar{0}, \bar{0} \odot \bar{0} = \bar{0}, \\ (w, g, b, v) \odot \bar{1} &= \bar{1} \odot (w, g, b, v) = (w, g, b, v) \text{ y } \bar{1} \odot \bar{1} = \bar{1}. \end{aligned}$$

ii) *Operador resumen* \oplus

$$(w_1, g, b_1, v) \oplus (w_2, g, b_2, v) = \begin{cases} (w_1, g, b_1, v) & \text{si } w_1 < w_2 \text{ o} \\ & w_1 = w_2 \text{ y } b_1 \geq b_2 \\ (w_2, g, b_2, v) & \text{si } w_1 > w_2 \text{ o} \\ & w_1 = w_2 \text{ y } b_1 < b_2 \end{cases}$$

$$\begin{aligned} (w, g, b, v) \oplus \bar{0} &= \bar{0} \oplus (w, g, b, v) = (w, g, b, v), \bar{0} \oplus \bar{1} = \bar{1} \oplus \bar{0} = \bar{1}, \\ (w, g, b, v) \oplus \bar{1} &= \bar{1} \oplus (w, g, b, v) = \bar{1}, \bar{0} \oplus \bar{0} = \bar{0} \text{ y } \bar{1} \oplus \bar{1} = \bar{1}. \end{aligned}$$

El dominio de las etiquetas S está formado por cuaternas de números (un racional no negativo, un racional estrictamente positivo y dos variables lógicas) y dos elementos especiales $\bar{0}$ y $\bar{1}$. Los elementos $\bar{0}$ y $\bar{1}$ no tienen expresión fácilmente interpretable en el conjunto $\mathbb{Q}^+ \times \mathbb{Q}^+ \setminus \{0\} \times \{0, 1\} \times \{0, 1\}$ y por ello se les denota de forma diferente y se muestra cómo operan con el resto de elementos.

La interpretación de las etiquetas es sencilla. $\bar{0}$ es la etiqueta de los pares ordenados (u, v) de vértices que no son aristas del grafo y $\bar{1}$ es la etiqueta del camino vacío (como en el marco general). Por lo que respecta a los elementos de $\mathbb{Q}^+ \times \mathbb{Q}^+ \setminus \{0\} \times \{0, 1\} \times \{0, 1\}$, la primera componente es el marcado ponderado del camino que lleva la etiqueta (es un racional no negativo), la segunda es la ganancia (es un racional estrictamente positivo), la tercera una variable lógica que indica si el camino pasa (1) o no (0) por una transición de interfaz (descontando las transiciones inicial y final), y la cuarta es otra variable lógica que indica si el vértice final del camino corresponde (1) o no (0) a una transición de interfaz.

El operador extensión \odot calcula, a partir de las etiquetas (w_1, g_1, b_1, v_1) y (w_2, g_2, b_2, v_2) de un par ordenado de caminos, la etiqueta (w, g, b, v) del camino formado por la concatenación de los dos caminos manteniendo el orden. Por ello, el marcado ponderado resultante es $w = w_1 + w_2/g_1$ (proposición 4.10) y la ganancia es $g = g_1g_2$ (por proposición 4.2). El camino resultante pasa por una transición de interfaz si lo hacía cualquiera de los dos que lo componen o si el vértice por el que se unen corresponde a una transición de interfaz, es decir, $b = \max\{b_1, b_2, v_1\}$. Y el último vértice del camino resultante es el mismo que el último vértice del segundo camino, es decir, $v = v_2$. La etiqueta correspondiente a la extensión de cualquier camino con el camino vacío ($\bar{1}$) o el inexistente ($\bar{0}$) explica el resto de la definición del operador extensión.

El operador resumen \oplus calcula, a partir de las etiquetas de dos caminos con el mismo vértice de origen y destino, la etiqueta del camino de menor marcado ponderado, y en caso de igualdad, la del que pase por una transición de interfaz, es decir, al operar con etiquetas se coge la etiqueta de menor primera componente (marcado ponderado del camino) y en caso de igualdad la de máxima tercera componente (indicador de paso por una transición de interfaz). Notar que se exige que la segunda y cuarta componente (ganancia y tipo de vértice final del camino) sean la misma para todas las etiquetas diferentes de $\bar{0}$ y $\bar{1}$ ya que sólo se resumen caminos con los mismos vértices origen y destino y estos obviamente tienen el mismo tipo de vértice final y

la misma ganancia (por ser el WTS vivo y limitado y por el teorema 4.6). El resumen de otras etiquetas con las del camino vacío o el inexistente son fáciles de explicar con esta interpretación.

Si se sigue el marco general, el operador resumen debería estar definido para cualquier par de etiquetas (cosa que no ocurre en la definición 4.25). Pero al igual que en MG's, se pueden relajar algunas de las condiciones en este caso ya que el resumen de caminos siempre se aplica a etiquetas de caminos con el mismo vértice de origen y de destino, o como mucho a las del camino vacío ($\bar{1}$) y/o al inexistente ($\bar{0}$). Por lo tanto, es suficiente que el operador resumen \oplus esté definido para las etiquetas de conjuntos de caminos con el mismo origen y destino junto con el camino vacío y/o el inexistente. Y todos los caminos con el mismo origen y destino en un WTS vivo y limitado tienen a su vez la misma ganancia (teorema 4.6). Esta relajación afectará tanto al resumen finito de etiquetas como al resumen de sucesiones de caminos.

Definición 4.26 *Un semianillo cerrado en sentido laxo es un semianillo cerrado (ver definición 3.8) en el que el operador resumen \oplus se define únicamente para etiquetas de caminos con el mismo origen y destino junto con el camino vacío y/o el inexistente.*

Se va a demostrar ahora que el dominio de etiquetas con las operaciones extensión y resumen así definidas tiene estructura de semianillo cerrado en sentido laxo, es decir, salvo la relajación que se han explicado referente al operador resumen.

Propiedad 4.27 *Sea $S = (\mathbb{Q}^+ \times \mathbb{Q}^+ \setminus \{0\} \times \{0, 1\} \times \{0, 1\}) \cup \bar{0} \cup \bar{1}$ el dominio de etiquetas y \odot y \oplus los operadores extensión y resumen de la definición 4.25. Entonces $\langle S, \oplus, \odot, \bar{0}, \bar{1} \rangle$ tiene estructura de semianillo cerrado en sentido laxo (ver definición 4.26).*

Demostración:

i.1) $\langle S, \oplus, \bar{0} \rangle$ es un monoide.

\oplus es una operación interna en S por definición ya que el min es interna en \mathbb{Q}^+ y el max es interna en $\{0, 1\}$. Que $\bar{0}$ es elemento neutro para \oplus también es evidente por la definición. Sólo falta demostrar la propiedad asociativa. Sean $s_1, s_2, s_3 \in S$. Se distinguen tres casos:

Caso 1) Algún s_i es $\bar{0}$. Nuevamente hay tres casos. Aplicando que $\bar{0}$ es elemento neutro de \oplus , se tiene:

$$\begin{aligned}\bar{0} \oplus (s_2 \oplus s_3) &= s_2 \oplus s_3 = (\bar{0} \oplus s_2) \oplus s_3. \\ s_1 \oplus (\bar{0} \oplus s_3) &= s_1 \oplus s_3 = (s_1 \oplus \bar{0}) \oplus s_3. \\ s_1 \oplus (s_2 \oplus \bar{0}) &= s_1 \oplus s_2 = (s_1 \oplus s_2) \oplus \bar{0}.\end{aligned}$$

Caso 2) Ningún s_i es $\bar{0}$ y algún s_i es $\bar{1}$. Otros tres casos. Aplicando que $\bar{1}$ es elemento anulador de \oplus se tiene:

$$\begin{aligned}\bar{1} \oplus (s_2 \oplus s_3) &= \bar{1} \quad \text{y} \quad (\bar{1} \oplus s_2) \oplus s_3 = \bar{1} \oplus s_3 = \bar{1}. \\ s_1 \oplus (\bar{1} \oplus s_3) &= s_1 \oplus \bar{1} = \bar{1} \quad \text{y} \quad (s_1 \oplus \bar{1}) \oplus s_3 = \bar{1} \oplus s_3 = \bar{1}. \\ s_1 \oplus (s_2 \oplus \bar{1}) &= s_1 \oplus \bar{1} = \bar{1} \quad \text{y} \quad (s_1 \oplus s_2) \oplus \bar{1} = \bar{1}.\end{aligned}$$

Caso 3) Todos los s_i son distintos de $\bar{0}$ y $\bar{1}$. Este caso se demuestra teniendo en cuenta que el min en \mathbb{Q}^+ y el max en $\{0, 1\}$ son asociativos.

i.2) $\langle S, \odot, \bar{1} \rangle$ es un monoide.

\odot es una operación interna en S por definición ya que la suma aritmética $+$ y el producto \cdot son internas en \mathbb{Q}^+ , la ganancia es estrictamente positiva, luego existe $1/g_1 \in \mathbb{Q}^+ \setminus \{0\}$ y el producto es interna en $\mathbb{Q}^+ \setminus \{0\}$. Por último, el max es interna en $\{0, 1\}$. Que $\bar{1}$ es elemento neutro para \odot también es evidente por la definición. Sólo falta demostrar la propiedad asociativa. Sean $s_1, s_2, s_3 \in S$. Se distinguen tres casos:

Caso 1) Algún s_i es $\bar{0}$. Aplicando que $\bar{0}$ es elemento anulador de \odot , se demuestra como el caso 2 del apartado *i.1*.

Caso 2) Ningún s_i es $\bar{0}$ y algún s_i es $\bar{1}$. Aplicando que $\bar{1}$ es elemento neutro de \odot se demuestra igual que el caso 1 de l apartado *i.1*.

Caso 3) Todos los s_i son distintos de $\bar{0}$ y $\bar{1}$.

$$\begin{aligned}s_1 \odot (s_2 \odot s_3) &= (w_1, g_1, b_1, v_1) \odot \left(w_2 + \frac{w_3}{g_2}, g_2 g_3, \max\{b_2, b_3, v_2\}, v_3 \right) \\ &= \left(w_1 + \frac{1}{g_1} \left(w_2 + \frac{w_3}{g_2} \right), g_1 (g_2 g_3), \max\{b_1, b_2, b_3, v_2, v_1\}, v_3 \right) \\ &= \left(w_1 + \frac{w_2}{g_1} + \frac{w_3}{g_1 g_2}, (g_1 g_2) g_3, \max\{b_1, b_2, v_1, b_3, v_2\}, v_3 \right) \\ &= \left(w_1 + \frac{w_2}{g_1}, g_1 g_2, \max\{b_1, b_2, v_1\}, v_2 \right) \odot (w_3, g_3, b_3, v_3) \\ &= (s_1 \odot s_2) \odot s_3\end{aligned}$$

ii) $\bar{0}$ es anulador de \odot . Por definición.

iii) \oplus es conmutativa. Por definición.

iv) \oplus es idempotente. Por definición.

v) \odot es distributiva respecto de \oplus . Sean $s_1, s_2, s_3 \in S$. Se distinguen tres casos:

Caso 1) Algún s_i es $\bar{0}$. Nuevamente hay tres casos:

$$\bar{0} \odot (s_2 \oplus s_3) = \bar{0} \quad \text{y} \quad (\bar{0} \odot s_2) \oplus (\bar{0} \odot s_3) = \bar{0} \oplus \bar{0} = \bar{0}.$$

$$(s_2 \oplus s_3) \odot \bar{0} = \bar{0} \quad \text{y} \quad (s_2 \odot \bar{0}) \oplus (s_3 \odot \bar{0}) = \bar{0} \oplus \bar{0} = \bar{0}.$$

$$s_1 \odot (\bar{0} \oplus s_3) = s_1 \odot s_3 \quad \text{y} \quad (s_1 \odot \bar{0}) \oplus (s_1 \odot s_3) = \bar{0} \oplus (s_1 \odot s_3) = s_1 \odot s_3.$$

$$\begin{aligned}(\bar{0} \oplus s_3) \odot s_1 &= s_3 \odot s_1 \quad \text{y} \quad (\bar{0} \odot s_1) \oplus (s_3 \odot s_1) = \bar{0} \oplus (s_3 \odot s_1) = s_3 \odot s_1. \\s_1 \odot (s_2 \oplus \bar{0}) &= s_1 \odot s_2 \quad \text{y} \quad (s_1 \odot s_2) \oplus (s_1 \odot \bar{0}) = (s_1 \odot s_2) \oplus \bar{0} = s_1 \odot s_2. \\(s_2 \oplus \bar{0}) \odot s_1 &= s_2 \odot s_1 \quad \text{y} \quad (s_2 \odot s_1) \oplus (\bar{0} \odot s_1) = (s_2 \odot s_1) \oplus \bar{0} = s_2 \odot s_1.\end{aligned}$$

Caso 2) Ningún s_i es $\bar{0}$ y algún s_i es $\bar{1}$. Otra vez hay tres casos:

$$\bar{1} \odot (s_2 \oplus s_3) = s_2 \oplus s_3 \quad \text{y} \quad (\bar{1} \odot s_2) \oplus (\bar{1} \odot s_3) = s_2 \oplus s_3.$$

$$(s_2 \oplus s_3) \odot \bar{1} = s_2 \oplus s_3 \quad \text{y} \quad (s_2 \odot \bar{1}) \oplus (s_3 \odot \bar{1}) = s_2 \oplus s_3.$$

$s_1 \odot (\bar{1} \oplus s_3) = s_1 \odot \bar{1} = s_1 \quad \text{y} \quad (s_1 \odot \bar{1}) \oplus (s_1 \odot s_3) = s_1 \oplus (s_1 \odot s_3) = s_1$. Esta última igualdad se debe a lo siguiente. Para que tenga sentido el resumen de s_1 con $s_1 \odot s_3$ los caminos correspondientes deben tener el mismo vértice origen y destino, es decir, el camino cuya etiqueta es s_3 es un ciclo (con vértice de origen y destino el vértice destino del camino de etiqueta s_1). Ahora, cualquier ciclo en un WTS vivo y limitado tiene marcado ponderado estrictamente positivo (teorema 4.5), luego el camino de etiqueta $s_1 \odot s_3$ tiene marcado ponderado estrictamente mayor que el camino de etiqueta s_1 . $(\bar{1} \oplus s_3) \odot s_1 = \bar{1} \odot s_1 = s_1 \quad \text{y} \quad (\bar{1} \odot s_1) \oplus (s_3 \odot s_1) = s_1 \oplus (s_3 \odot s_1) = s_1$ por la misma razón.

$s_1 \odot (s_2 \oplus \bar{1}) = s_1 \odot \bar{1} = s_1 \quad \text{y} \quad (s_1 \odot s_2) \oplus (s_1 \odot \bar{1}) = (s_1 \odot s_2) \oplus s_1 = s_1$ por la misma razón.

$(s_2 \oplus \bar{1}) \odot s_1 = \bar{1} \odot s_1 = s_1 \quad \text{y} \quad (s_2 \odot s_1) \oplus (\bar{1} \odot s_1) = (s_2 \odot s_1) \oplus s_1 = s_1$ por la misma razón.

Caso 3) Todos los s_i son distintos de $\bar{0}$ y $\bar{1}$. Sean $s_i = (w_i, g_i, b_i, v_i)$ para $i = 1, 2, 3$. Deben ser $g_2 = g_3$ y $v_2 = v_3$ para que tengan sentido los resúmenes involucrados.

$$\begin{aligned}s_1 \odot (s_2 \oplus s_3) &= \begin{cases} s_1 \odot s_2 & \text{si } w_2 < w_3 \text{ o } (w_2 = w_3 \text{ y } b_2 \geq b_3) \\ s_1 \odot s_3 & \text{si } w_2 > w_3 \text{ o } (w_2 = w_3 \text{ y } b_2 < b_3) \end{cases} \\&= \begin{cases} \left(w_1 + \frac{w_2}{g_1}, g_1 g_2, \max\{b_1, b_2, v_1\}, v_2 \right) & \text{si } w_2 < w_3 \text{ o } (w_2 = w_3 \text{ y } b_2 \geq b_3) \\ \left(w_1 + \frac{w_3}{g_1}, g_1 g_3, \max\{b_1, b_3, v_1\}, v_3 \right) & \text{si } w_2 > w_3 \text{ o } (w_2 = w_3 \text{ y } b_2 < b_3) \end{cases}\end{aligned}$$

$$\begin{aligned}\text{Por otro lado, } (s_1 \odot s_2) \oplus (s_1 \odot s_3) &= \left(w_1 + \frac{w_2}{g_1}, g_1 g_2, \max\{b_1, b_2, v_1\}, v_2 \right) \oplus \\&\left(w_1 + \frac{w_3}{g_1}, g_1 g_3, \max\{b_1, b_3, v_1\}, v_3 \right) \\&= \begin{cases} \left(w_1 + \frac{w_2}{g_1}, g_1 g_2, \max\{b_1, b_2, v_1\}, v_2 \right) & \text{si } w_1 + \frac{w_2}{g_1} < w_1 + \frac{w_3}{g_1} \text{ o} \\ & w_1 + \frac{w_2}{g_1} = w_1 + \frac{w_3}{g_1} \text{ y } \max\{b_1, b_2, v_1\} \geq \max\{b_1, b_3, v_1\} \\ \left(w_1 + \frac{w_3}{g_1}, g_1 g_3, \max\{b_1, b_3, v_1\}, v_3 \right) & \text{si } w_1 + \frac{w_2}{g_1} > w_1 + \frac{w_3}{g_1} \text{ o} \\ & w_1 + \frac{w_2}{g_1} = w_1 + \frac{w_3}{g_1} \text{ y } \max\{b_1, b_2, v_1\} < \max\{b_1, b_3, v_1\} \end{cases}\end{aligned}$$

Hay que demostrar que para cualesquiera s_1, s_2, s_3 las dos expresiones coinciden. $w_1 + w_2/g_1 < w_1 + w_3/g_1$ si y sólo si $w_2 < w_3$ y $w_1 + w_2/g_1 = w_1 + w_3/g_1$

si y sólo si $w_2 = w_3$ para todo $w_1, w_2, w_3 \in \mathbb{Q}^+$ y para todo $g_1 \in \mathbb{Q}^+ \setminus \{0\}$, luego si $w_2 \neq w_3$ las expresiones coinciden. Si $w_2 = w_3$ y $b_2 \geq b_3$ entonces $w_1 + w_2/g_1 = w_1 + w_3/g_1$ y $\max\{b_1, b_2, v_1\} \geq \max\{b_1, b_3, v_1\}$ luego también coinciden las expresiones. Por último, si $w_2 = w_3$ y $b_2 < b_3$ entonces $w_1 + w_2/g_1 = w_1 + w_3/g_1$ y $\max\{b_1, b_2, v_1\} \leq \max\{b_1, b_3, v_1\}$. Si la última desigualdad es estricta, las expresiones coinciden y si es una igualdad ($\max\{b_1, b_2, v_1\} = \max\{b_1, b_3, v_1\}$) entonces los dos casos de la segunda expresión son iguales al caso de la primera. Por lo tanto, queda demostrado que $s_1 \odot (s_2 \oplus s_3) = (s_1 \odot s_2) \oplus (s_1 \odot s_3)$.

$$(s_2 \oplus s_3) \odot s_1 = \begin{cases} s_2 \odot s_1 & \text{si } w_2 < w_3 \text{ o } (w_2 = w_3 \text{ y } b_2 \geq b_3) \\ s_3 \odot s_1 & \text{si } w_2 > w_3 \text{ o } (w_2 = w_3 \text{ y } b_2 < b_3) \end{cases}$$

$$= \begin{cases} \left(w_2 + \frac{w_1}{g_2}, g_2 g_1, \max\{b_2, b_1, v_2\}, v_1 \right) & \text{si } w_2 < w_3 \text{ o } (w_2 = w_3 \text{ y } b_2 \geq b_3) \\ \left(w_3 + \frac{w_1}{g_3}, g_3 g_1, \max\{b_3, b_1, v_3\}, v_1 \right) & \text{si } w_2 > w_3 \text{ o } (w_2 = w_3 \text{ y } b_2 < b_3) \end{cases}$$

Por otro lado, $(s_2 \odot s_1) \oplus (s_3 \odot s_1) = \left(w_2 + \frac{w_1}{g_2}, g_2 g_1, \max\{b_2, b_1, v_2\}, v_1 \right) \oplus \left(w_3 + \frac{w_1}{g_3}, g_3 g_1, \max\{b_3, b_1, v_3\}, v_1 \right)$

$$= \begin{cases} \left(w_2 + \frac{w_1}{g_2}, g_2 g_1, \max\{b_2, b_1, v_2\}, v_1 \right) & \text{si } w_2 + \frac{w_1}{g_2} < w_3 + \frac{w_1}{g_3} \text{ o} \\ & w_2 + \frac{w_1}{g_2} = w_3 + \frac{w_1}{g_3} \text{ y } \max\{b_2, b_1, v_2\} \geq \max\{b_3, b_1, v_3\} \\ \left(w_3 + \frac{w_1}{g_3}, g_3 g_1, \max\{b_3, b_1, v_3\}, v_1 \right) & \text{si } w_2 + \frac{w_1}{g_2} > w_3 + \frac{w_1}{g_3} \text{ o} \\ & w_2 + \frac{w_1}{g_2} = w_3 + \frac{w_1}{g_3} \text{ y } \max\{b_2, b_1, v_2\} < \max\{b_3, b_1, v_3\} \end{cases}$$

Ahora, como $g_2 = g_3$, $w_2 < w_3$ si y sólo si $w_2 + w_1/g_2 < w_3 + w_1/g_3$ y $w_2 = w_3$ si y sólo si $w_2 + w_1/g_2 = w_3 + w_1/g_3$ para todo $w_1, w_2, w_3 \in \mathbb{Q}^+$ y para todo $g_2 = g_3 \in \mathbb{Q}^+ \setminus \{0\}$, luego si $w_2 \neq w_3$ las expresiones coinciden. Si $w_2 = w_3$ y $b_2 \geq b_3$ entonces $w_2 + w_1/g_2 = w_3 + w_1/g_3$ y como $v_2 = v_3$ entonces $\max\{b_2, b_1, v_2\} \geq \max\{b_3, b_1, v_3\}$ luego también coinciden las expresiones. Por último, si $w_2 = w_3$ y $b_2 < b_3$ entonces $w_2 + w_1/g_2 = w_3 + w_1/g_3$ y como $v_2 = v_3$ entonces $\max\{b_2, b_1, v_2\} \leq \max\{b_3, b_1, v_3\}$. Si la última desigualdad es estricta, las expresiones coinciden y si $\max\{b_2, b_1, v_2\} = \max\{b_3, b_1, v_3\}$ entonces los dos casos de la segunda expresión son iguales al caso de la primera. Por lo tanto, $(s_2 \oplus s_3) \odot s_1 = (s_2 \odot s_1) \oplus (s_3 \odot s_1)$.

vi) Si $\{s_i\}_{i=1}^{\infty}$ es una sucesión de etiquetas, $\bigoplus_{i=1}^{\infty} s_i$ está bien definida y en S : Si algún camino de la sucesión tiene etiqueta $\bar{0}$, se puede eliminar del resumen. Si alguno tiene etiqueta $\bar{1}$, el resultado del resumen es $\bar{1}$. El caso que falta es si en la sucesión todos los $s_i \neq \bar{0}, \bar{1}$. Para demostrar este caso se tendrá en cuenta la relajación comentada en la definición 4.26 respecto a las sucesiones de etiquetas que pueden aparecer en WTS. Dadas

dos transiciones t y t' , para cada $i \in \mathbb{N}$ sea π_i un camino de t a t' cuya etiqueta sea s_i . Para que tenga sentido el resumen, todos los s_i son de la forma $s_i = (w_i, g, b_i, v)$, es decir, tienen la misma ganancia y el mismo tipo de vértice final. Por lo tanto, el resumen de la sucesión tendrá las mismas segunda y cuarta componentes que todos los s_i . La tercera componente del resumen no crea problemas porque será 0 ó 1. El único problema que puede aparecer es en la primera componente, el marcado ponderado. La primera componente del resumen es $\min_{i=1}^{\infty} \{w_i\}$. En general, puede no existir el mínimo de una sucesión de números racionales. Se demostrará que en este caso si existe.

Cada camino π_i está compuesto por un conjunto de aristas del grafo correspondiente al WTS. Sea $\{a_i\}_{i=1}^m$ el conjunto de aristas del grafo. Por la definición 4.3, cada arista a_i del WTS aporta al marcado ponderado de cada camino que la contenga una cantidad fija $q_i \in \mathbb{Q}^+$ igual al marcado ponderado de la arista dividido por la ganancia entre el vértice de comienzo del camino y el primer vértice de la arista. Como todos los caminos del resumen tienen el mismo vértice de origen, los q_j son los mismos para todos los caminos. Para cada camino π_i , sea $\rho_i = (n_{i1}, \dots, n_{im})$ un vector de m enteros no negativos tales que n_{ij} es el número de veces que el camino π_i contiene a la arista a_j . Entonces, $M(\pi_i, \mathbf{m}_0) = \sum_{j=1}^m n_{ij} q_j$. Puede suponerse sin pérdida de generalidad que $q_i \neq 0$ para todo $1 \leq i \leq m$ (para ello es suficiente eliminar los elementos nulos del conjunto $\{q_i\}_{i=1}^m$, reenumerar los que quedan y disminuir el valor de m).

Como $M(\pi_1, \mathbf{m}_0) = w_1$, sea $K_j = \lfloor w_1/q_j \rfloor$ para cada $1 \leq j \leq m$. K_j es el número máximo de veces que cualquier camino del resumen π_i puede contener a la arista a_j para tener marcado ponderado menor o igual w_1 . Como $w_1 \in \mathbb{Q}^+$ y $q_i \in \mathbb{Q}^+ \setminus \{0\}$ para todo $1 \leq i \leq m$, entonces $K_i \in \mathbb{Z}^+$ para todo $1 \leq i \leq m$. Entonces, en el conjunto $\{w_i \mid i \in \mathbb{N}\}$ de los marcados ponderados de los caminos del resumen sólo puede haber un número finito de marcados ponderados menores o iguales que w_1 , todos contenidos en el conjunto $\{\sum_{i=1}^m k_i q_i \mid 0 \leq k_i \leq K_i, 1 \leq i \leq m\}$ que es de cardinal finito. Por lo tanto, $\min_{i=1}^{\infty} \{w_i\}$ existe y es un racional no negativo (se reduce a calcular el mínimo del conjunto de los marcados ponderados menores o iguales que w_1 que es un conjunto finito de racionales no negativos).

vii) Se cumplen las propiedades conmutativa, asociativa e idempotencia en el resumen de sucesiones: Si algún camino de la sucesión tiene etiqueta $\bar{0}$, se puede eliminar del resumen, lo que equivale a hacer un resumen con un único camino de etiqueta $\bar{0}$. Si alguno tiene etiqueta $\bar{1}$, el resultado del

resumen es $\bar{1}$, lo que equivale a hacer un resumen con un único camino de etiqueta $\bar{1}$. En el resto de casos, debido a la conmutatividad e idempotencia del operador \oplus , las etiquetas de la forma (w, g, b, v) pueden conmutar con el resto de etiquetas y eliminarse las instancias repetidas.

viii) \odot es distributiva respecto de resúmenes de sucesiones. Sean $a, b_i \in S$ con $i \in \mathbb{N}$. Se distinguen tres casos:

Caso 1) Todos los $b_i = \bar{0}$. Entonces $a \odot (\bigoplus_{i=1}^{\infty} b_i) = a \odot (\bigoplus_{i=1}^{\infty} \bar{0}) = a \odot \bar{0} = \bar{0}$ y $\bigoplus_{i=1}^{\infty} (a \odot b_i) = \bigoplus_{i=1}^{\infty} (a \odot \bar{0}) = \bigoplus_{i=1}^{\infty} \bar{0} = \bar{0}$.

$(\bigoplus_{i=1}^{\infty} b_i) \odot a = (\bigoplus_{i=1}^{\infty} \bar{0}) \odot a = \bar{0} \odot a = \bar{0}$ y $\bigoplus_{i=1}^{\infty} (b_i \odot a) = \bigoplus_{i=1}^{\infty} (\bar{0} \odot a) = \bigoplus_{i=1}^{\infty} \bar{0} = \bar{0}$.

Caso 2) Algún $b_i = \bar{1}$. Por el apartado *vii*, se puede suponer que $b_1 = \bar{1}$ y $b_i \neq \bar{1}$ para todo $i > 1$. Entonces, $a \odot (\bigoplus_{i=1}^{\infty} b_i) = a \odot (\bar{1} \oplus (\bigoplus_{i=2}^{\infty} b_i)) = a \odot \bar{1} = a$ y $\bigoplus_{i=1}^{\infty} (a \odot b_i) = (a \odot \bar{1}) \oplus (\bigoplus_{i=2}^{\infty} (a \odot b_i)) = a \oplus (\bigoplus_{i=2}^{\infty} (a \odot b_i)) = a$ por la misma razón que en caso 2 del apartado *v*.

$(\bigoplus_{i=1}^{\infty} b_i) \odot a = (\bar{1} \oplus (\bigoplus_{i=2}^{\infty} b_i)) \odot a = \bar{1} \odot a = a$ y

$\bigoplus_{i=1}^{\infty} (b_i \odot a) = (\bar{1} \odot a) \oplus (\bigoplus_{i=2}^{\infty} (b_i \odot a)) = a \oplus (\bigoplus_{i=2}^{\infty} (b_i \odot a)) = a$ por la misma razón.

Caso 3) $b_i \neq \bar{1}$ para todo $i \in \mathbb{N}$ y algún $b_i \neq \bar{0}$. Por el apartado *vii*, se puede suponer también que los $b_i \neq \bar{0}$ para todo $i \in \mathbb{N}$. El apartado *vii* también permite reordenar los $b_i = (w_i, g_i, b_i, v_i)$ de forma que $w_i \leq w_{i+1}$ para todo $i \in \mathbb{N}$ y si $w_i = w_{i+1}$ para algún i entonces $b_i \geq b_{i+1}$. Todos los v_i, g_i con $i \in \mathbb{N}$ deben ser iguales para que tenga sentido el resumen. Después de esta reordenación $a \odot (\bigoplus_{i=1}^{\infty} b_i) = a \odot b_1$ por definición de \oplus y $\bigoplus_{i=1}^{\infty} (a \odot b_i) = (a \odot b_1)$ como se demostrará ahora. Sea $a = (w, g, b, v)$. $w + w_i/g \leq w + w_{i+1}/g$ para todo $i \in \mathbb{N}$ por ser $w_i \leq w_{i+1}$. Además, si $w + w_i/g = w + w_{i+1}/g$ para algún $i \in \mathbb{N}$ entonces $w_i = w_{i+1}$. Por la reordenación, $b_i \geq b_{i+1}$ y por lo tanto $\max\{b, b_i, v\} \geq \max\{b, b_{i+1}, v\}$.

Por otra parte, por definición de \oplus , se tiene que $(\bigoplus_{i=1}^{\infty} b_i) \odot a = b_1 \odot a$ y $\bigoplus_{i=1}^{\infty} (b_i \odot a) = (b_1 \odot a)$ ya que, por la reordenación anterior, se tiene que $w_i + w/g_i \leq w_{i+1} + w/g_{i+1}$ para todo $i \in \mathbb{N}$ (por ser todos los g_i iguales) y si $w_i + w/g_i = w_{i+1} + w/g_{i+1}$ entonces $\max\{b_i, b, v_i\} \geq \max\{b_{i+1}, b, v_{i+1}\}$ ya que $v_i = v_{i+1}$ y $b_i \geq b_{i+1}$ en ese caso. \diamond

Una vez demostrada la estructura de semianillo cerrado en sentido laxo para $\langle S, \oplus, \odot, \bar{0}, \bar{1} \rangle$, siguiendo el marco general para la resolución de problemas de caminos en grafos dirigidos de [CLR90], el algoritmo 3.9 con los operadores \oplus y \odot de la definición 4.25 permite calcular el resumen de todos los caminos que unen dos vértices cualesquiera del grafo inicial.

Ahora se expone un algoritmo eficiente para el cálculo de marcados ponderados mínimos entre transiciones de un WTS.

Algoritmo 4.28 Algoritmo de cálculo de TT-MSIP's.

input: $G := (V, A)$, \mathbf{m}_0 grafo dirigido asociado al WTS y marcado inicial
 \mathbf{g} Vector de ganancias (salida del algoritmo 4.23)
 \mathbf{L} Matriz de marcados ponderados entre vértices del grafo
 \mathbf{B} Matriz de indicadores de paso por transiciones de interfaz
 $n :=$ número de transiciones del WTS
 $n' :=$ número de transiciones internas del WTS
for $i := 1$ **to** n **do**
 for $j := 1$ **to** n **do**
 $\mathbf{B}(i, j) := 0$
 if $i = j$ **then** $\mathbf{L}(i, j) := 0$; $\mathbf{B}(i, j) = 1$
 else if $(i, j) \in A$ **then** $\mathbf{L}(i, j) := \mathbf{m}_0[p_{ij}]/W(t_i, p_{ij})$
 else $\mathbf{L}(i, j) := \infty$
 end if
 end for
end for
for $k := 1$ **to** n **do**
 for $i := 1$ **to** n , $i \neq k$ **do**
 for $j := 1$ **to** n , $j \neq k$ **do**
 if $\mathbf{L}(i, j) > \mathbf{L}(i, k) + \frac{\mathbf{L}(k, j)\mathbf{g}(i)}{\mathbf{g}(k)}$ **then** $\mathbf{L}(i, j) := \mathbf{L}(i, k) + \frac{\mathbf{L}(k, j)\mathbf{g}(i)}{\mathbf{g}(k)}$
 if $k > n'$ **then** $\mathbf{B}(i, j) := 1$
 else $\mathbf{B}(i, j) := \max\{\mathbf{B}(i, k), \mathbf{B}(k, j)\}$
 end if
 else if $\mathbf{L}(i, j) = \mathbf{L}(i, k) + \frac{\mathbf{L}(k, j)\mathbf{g}(i)}{\mathbf{g}(k)}$ **then**
 if $k > n'$ **then** $\mathbf{B}(i, j) := 1$
 else $\mathbf{B}(i, j) := \max\{\mathbf{B}(i, j), \mathbf{B}(i, k), \mathbf{B}(k, j)\}$
 end if
 end if
 end for
end for
end for
output: Matrices \mathbf{L} , \mathbf{B}

A la hora de codificar el algoritmo en la práctica, no es necesario mantener las etiquetas de cuatro componentes como en el desarrollo teórico, sino

únicamente el marcado ponderado mínimo entre cada dos transiciones y una variable lógica que indique si el camino mínimo pasa o no por una transición de interfaz. Como no es necesario conocer explícitamente cuáles son los caminos de marcado ponderado mínimo, se puede eliminar del algoritmo la matriz \mathbf{K} del algoritmo general de resúmenes (ver algoritmo 3.9).

Se denota con n al número de transiciones del WTS y con n' al número de transiciones internas. Se numeran las transiciones de forma que primero aparezcan las transiciones internas. De esta forma se tiene un test inmediato de si una transición es o no de interfaz.

La entrada de este algoritmo consta de la descripción de la estructura y marcado inicial del WTS por medio de su grafo dirigido asociado G y \mathbf{m}_0 . También se incluye como entrada el vector \mathbf{g} de ganancias, para no repetir los cálculos del algoritmo 4.23. A partir de la descripción del WTS, se construyen dos matrices \mathbf{L} y \mathbf{B} . La matriz \mathbf{L} sirve para almacenar los marcados ponderados mínimos en el marcado inicial entre transiciones cualesquiera. Inicialmente contiene entradas 0 para los elementos de la diagonal. Este 0 es el resultado correspondiente a la operación $\bar{1} \oplus \lambda(i, i)$ del algoritmo general de resúmenes (algoritmo 3.9). En este caso, el camino vacío tiene marcado ponderado 0 y cualquier camino con origen y destino el mismo vértice debe tener marcado ponderado estrictamente positivo (apartado 2 del teorema 4.5), por lo que el resumen es el de un camino con marcado ponderado 0. Para elementos distintos de la diagonal, el valor inicial puede ser el marcado ponderado del camino simple que une los vértices o ∞ si este camino no existe (indicando que de momento el camino de marcado ponderado mínimo es el camino inexistente).

Por lo que respecta a la matriz \mathbf{B} , contiene información sobre si los caminos de marcado ponderado mínimo pasan (1) o no (0) por una transición de interfaz. Inicialmente todas sus entradas, excepto las de la diagonal, son 0 indicando que ningún camino tiene un vértice intermedio que sea una transición de interfaz, puesto que de momento los caminos considerados tienen sólo una transición origen y otra destino. Los elementos de la diagonal se inicializan a 1 para evitar que el algoritmo calcule un lugar implícito con una única transición de entrada y salida a la vez. Esos lugares no tienen sentido en la reducción de WTS's.

El paso siguiente (los tres bucles encajados) consiste en realizar los cálculos de caminos de marcado ponderado mínimo. Se eliminan de los bucles las comparaciones triviales en WTS's, que son los casos en que $i = k$ o $j = k$. En el algoritmo general de resúmenes esta operación consis-

te en calcular la ecuación (3.2). En este caso, la clausura $(l_{kk}^{(k-1)})^*$ es $\bar{1} \oplus (\bigoplus_{i=1}^{\infty} (\bigodot_{j=1}^i l_{kk}^{(k-1)})) = \bar{1}$ por lo que puede desaparecer del resumen, simplificando la expresión. La expresión está codificada en las instrucciones del interior de los tres bucles encajados. Si el camino de menor marcado ponderado del vértice i al k concatenado con el que va del vértice k al j tiene menor marcado ponderado $(\mathbf{L}(i, k) + \mathbf{L}(k, j)\mathbf{g}(k)/\mathbf{g}(i))$ que el que va de i al j $(\mathbf{L}(i, j))$ se almacena en \mathbf{L} este nuevo marcado ponderado mínimo y en \mathbf{B} el indicador de paso por transición de interfaz. Si los dos caminos tienen el mismo marcado ponderado, entonces se comparan sus indicadores de paso por transición de interfaz, almacenándose el que sea mayor, es decir, si uno de los dos pasa por una transición de interfaz, el algoritmo se queda con él. Para calcular si un camino que es la concatenación de otros dos pasa o no por una transición de interfaz, basta tomar los indicadores de los dos subcaminos (elementos $\mathbf{B}(i, k)$ y $\mathbf{B}(k, j)$) y por último saber si el vértice k corresponde o no a una transición de interfaz (esto es inmediato realizando la comparación $k > n'$ debido a la numeración de las transiciones).

De esta manera, tras la iteración k del bucle exterior, $\mathbf{L}(i, j)$ proporciona el menor marcado ponderado de entre todos los caminos que unen el vértice i con el j y que están compuestos únicamente por vértices de índice menor o igual que k . Por su parte, $\mathbf{B}(i, j)$ es (también tras la iteración k) una variable lógica (valor 0 ó 1) que indica si uno de estos caminos de marcado ponderado mínimo pasa o no por una transición de interfaz.

Al finalizar el algoritmo, se tiene en $\mathbf{L}(i, j)$ el marcado ponderado mínimo de entre los caminos que unen el vértice i con el j y en $\mathbf{B}(i, j)$ un indicador de si existe un camino de marcado ponderado mínimo que pase por una transición de interfaz.

La complejidad de este algoritmo es de $O(n^3)$ en tiempo y $O(n^2)$ en espacio, donde n es el número de transiciones del WTS original. Esta complejidad es la misma que la del algoritmo de Floyd, por lo que el desarrollo teórico realizado no aumenta el coste computacional del algoritmo resultante. Por lo tanto, se ha desarrollado un algoritmo que permite calcular, para dos transiciones cualesquiera del WTS original, el menor marcado ponderado de entre los caminos que las unen, eligiendo de entre todos los caminos de marcado ponderado mínimo uno que pase por una transición de interfaz, si existe.

Una vez resuelto el cálculo de las ganancias y los marcados ponderados mínimos entre transiciones de un WTS vivo y limitado, se está en condiciones

de calcular las resistencias máximas entre transiciones de interfaz. La resistencia de un camino depende del marcado ponderado atrapado en el mismo, que a su vez depende de los marcados ponderados mínimos ya calculados con el algoritmo 4.28. Este cálculo puede hacerse con otra ligera modificación del algoritmo de Floyd para el cálculo de caminos de peso mínimo entre los vértices de un grafo dirigido con pesos. En el caso de la resistencia no existe una fórmula sencilla que relacione la resistencia de un camino con la de dos subcaminos suyos, pero sí que existe para la resistencia estructural (proposición 4.13) y para el marcado ponderado (proposición 4.10). Para poder aplicar estas relaciones es necesario disponer de las ganancias de los subcaminos (algoritmo 4.23). La idea del algoritmo consiste en calcular para cada camino un par de valores. El primero de ellos representa la resistencia del camino y el segundo el marcado ponderado libre del mismo. Cada vez que se detecte una sincronización de este camino con otros se disminuirán los dos valores en una cantidad igual al marcado ponderado atrapado del mismo debido a la sincronización. De esta manera se mantiene la interpretación de los dos valores asociados a cada camino y es posible calcular de forma eficiente los caminos de resistencia máxima entre transiciones del WTS.

No es necesario calcular la resistencia máxima entre transiciones cualesquiera, sino únicamente entre transiciones de interfaz. También es necesario demostrar la corrección del algoritmo que se va a exponer a continuación y para ello se empleará de nuevo el marco general para la resolución de problemas de caminos en grafos dirigidos con pesos de [CLR90] y que se ha explicado con detalle en el capítulo 3. Ya se ha visto cómo construir un grafo a partir de la estructura de un WTS y lo único que cambia respecto al algoritmo de marcados ponderados mínimos son las etiquetas que se asocian a las aristas del mismo. Por ello el primer paso consiste en definir el tipo de etiquetas que se van a necesitar ahora.

Definición 4.29 Sea $S = (\mathbb{Q}^+ \times \mathbb{Q}^+ \times \mathbb{Q}^+ \setminus \{0\}) \cup \bar{0}$ ($\bar{0}$ es un elemento especial), y sea $\bar{1} = (0, 0, 1)$. Se definen las siguientes operaciones en S :

i) Operador extensión \odot

$$(r_1, w_1, g_1) \odot (r_2, w_2, g_2) = (\max\{r_1, r_2/g_1\}, w_1 + w_2/g_1, g_1 g_2)$$

$$(r, w, g) \odot \bar{0} = \bar{0} \odot (r, w, g) = \bar{0} \odot \bar{0} = \bar{0}$$

ii) Operador resumen \oplus

$$(r_1, w_1, g) \oplus (r_2, w_2, g) = \left(\max_{i=1,2} \{r_i - w_i\} + \min_{i=1,2} \{w_i\}, \min_{i=1,2} \{w_i\}, g \right)$$

$$(r, w, g) \oplus \bar{0} = \bar{0} \oplus (r, w, g) = (r, w, g) \text{ y } \bar{0} \oplus \bar{0} = \bar{0}$$

El dominio de las etiquetas S está formado por ternas de números (dos racionales no negativos y un racional estrictamente positivo) y un elemento especial $\bar{0}$ que no tiene expresión fácilmente interpretable en el conjunto $\mathbb{Q}^+ \times \mathbb{Q}^+ \times \mathbb{Q}^+ \setminus \{0\}$. Por ello se le denota de forma diferente y se muestra cómo opera con el resto de elementos.

La interpretación de las etiquetas es sencilla. $\bar{0}$ es la etiqueta de los pares ordenados (u, v) de vértices que no son aristas del grafo y $\bar{1}$ es la etiqueta del camino vacío (como en el marco general). Por lo que respecta a los elementos de $\mathbb{Q}^+ \times \mathbb{Q}^+ \times \mathbb{Q}^+ \setminus \{0\}$, la primera componente es la resistencia del camino que lleva la etiqueta, la segunda es el marcado ponderado libre del camino (es decir, el marcado ponderado no atrapado en el propio camino debido a sincronizaciones con otros caminos) y la tercera es la ganancia del camino. Esta tercera componente se incluye sólo para simplificar el desarrollo teórico, pero no es necesario incluirla en la codificación porque ya se han realizado los cálculos en el algoritmo 4.23.

El operador extensión \odot calcula, a partir de las etiquetas (r_1, w_1, g_1) y (r_2, w_2, g_2) de un par ordenado de caminos, la etiqueta (r, w, g) del camino formado por la concatenación de los dos caminos manteniendo el orden. Por la proposición 4.13, se tiene que la resistencia del camino concatenación es $\max\{r_1, r_2/g_1\}$, por la proposición 4.10 el marcado ponderado es $w = w_1 + w_2/g_1$ y por la proposición 4.2 la ganancia es $g = g_1g_2$. La etiqueta correspondiente a la extensión de cualquier camino con el camino inexistente ($\bar{0}$) explica el resto de la definición del operador extensión.

El operador resumen \oplus calcula, a partir las etiquetas de dos caminos con el mismo vértice de origen y destino, la etiqueta del camino de máxima resistencia. Para entender la operación hay que tener en cuenta que al hacer el resumen de dos caminos se produce una sincronización en la transición final de los mismos. Como ya se ha explicado en la sección 4.2, las sincronizaciones de caminos pueden atrapar marcas en los mismos y estas marcas atrapadas disminuyen la resistencia de un camino respecto a su resistencia estructural. De esta manera, el camino del resumen con menor marcado ponderado es el que indica el marcado ponderado que queda libre después de la sincronización de los dos caminos ($\min\{w_1, w_2\}$). Por ello se ha dado como interpretación de la segunda componente de las ternas el marcado ponderado libre y no el marcado ponderado total de un camino. Por lo que respecta a las resistencias, éstas se ven decrementadas respecto a la resistencia estructural por el marcado ponderado atrapado en cada camino. El marcado ponderado atrapado en el camino i con $i = 1, 2$ es $w_i - \min\{w_1, w_2\}$, por lo

que la resistencia del camino i será $r_i - w_i + \min\{w_1, w_2\}$. En consecuencia, el camino con mayor resistencia del resumen tiene una resistencia igual a $\max_{i=1,2}\{r_i - w_i + \min\{w_1, w_2\}\} = \max_{i=1,2}\{r_i - w_i\} + \min\{w_1, w_2\}$. Esto explica la primera componente del resumen. Por lo que respecta a la ganancia no hay que decir nada porque la ganancia los caminos que intervienen en un resumen siempre es la misma por tener las mismas transiciones origen y destino (por ser el WTS vivo y limitado y por el teorema 4.6). El resumen de otras etiquetas con la del camino inexistente son fáciles de explicar con esta interpretación.

Si se sigue el marco general, el operador resumen debería estar definido para cualquier par de etiquetas (cosa que no ocurre en la definición 4.29). Pero, al igual que en el caso del marcado ponderado se pueden relajar algunas de las condiciones en este caso ya que el resumen de caminos siempre se aplica a etiquetas de caminos con el mismo vértice de origen y de destino, o como mucho a las del camino vacío ($\bar{1}$) y/o al inexistente ($\bar{0}$). Por lo tanto es suficiente que el operador resumen \oplus esté definido para las etiquetas de conjuntos de caminos con el mismo origen y destino junto con el camino vacío y/o el inexistente. Y todos los caminos con el mismo origen y destino en un WTS tienen a su vez la misma ganancia. Esta relajación, expuesta formalmente en la definición 4.26, afectará tanto al resumen finito de etiquetas como al resumen de sucesiones de caminos.

Se va a demostrar ahora que el dominio de etiquetas con las operaciones extensión y resumen así definidas tiene estructura de semianillo cerrado en sentido laxo (ver definición 4.26).

Propiedad 4.30 *Sea $S = (\mathbb{Q}^+ \times \mathbb{Q}^+ \times \mathbb{Q}^+ \setminus \{0\}) \cup \bar{0}$ el dominio de etiquetas y \odot, \oplus los operadores extensión y resumen de la definición 4.29. Entonces $\langle S, \oplus, \odot, \bar{0}, \bar{1} \rangle$ tiene estructura de semianillo cerrado en sentido laxo (ver definición 4.26).*

Demostración:

i.1) $\langle S, \oplus, \bar{0} \rangle$ es un monoide.

Primero hay que demostrar es que \oplus es una operación interna en S . Esto es evidente si algún operando es $\bar{0}$ (por la definición de \oplus). Si se tiene un resumen de la forma $(r_1, w_1, g) \oplus (r_2, w_2, g)$, la etiqueta resultado tiene por tercera componente $g \in \mathbb{Q}^+ \setminus \{0\}$, y por segunda componente $\min\{w_1, w_2\} \in \mathbb{Q}^+$ (por ser el min interna en \mathbb{Q}^+). Falta ver que la primera componente del resumen es un racional no negativo. Sea $w_j = \min\{w_1, w_2\}$ (con $j = 1$ ó 2).

Entonces $\max_{i=1,2}\{r_i - w_i\} + \min_{i=1,2}\{w_i\} \geq r_j - w_j + w_j = r_j \geq 0$, por lo que queda demostrado que \oplus es una operación interna en S .

$\bar{0}$ es elemento neutro para \oplus por definición, por lo que sólo falta demostrar la propiedad asociativa. Sean $s_1, s_2, s_3 \in S$. Se distinguen dos casos:

Caso 1) Algún s_i es $\bar{0}$. Nuevamente hay tres casos. Aplicando que $\bar{0}$ es elemento neutro de \oplus , se tiene:

$$\bar{0} \oplus (s_2 \oplus s_3) = s_2 \oplus s_3 = (\bar{0} \oplus s_2) \oplus s_3.$$

$$s_1 \oplus (\bar{0} \oplus s_3) = s_1 \oplus s_3 = (s_1 \oplus \bar{0}) \oplus s_3.$$

$$s_1 \oplus (s_2 \oplus \bar{0}) = s_1 \oplus s_2 = (s_1 \oplus s_2) \oplus \bar{0}.$$

Caso 2) Todos los s_i son distintos de $\bar{0}$. Este caso se demuestra teniendo en cuenta que el min y el max son asociativas en \mathbb{Q}^+ .

i.2) $\langle S, \odot, \bar{1} \rangle$ es un monoide.

\odot es una operación interna en S por definición ya que la ganancia de un camino nunca es 0 luego existe $1/g_1 \in \mathbb{Q}^+ \setminus \{0\}$ y el max es interna en \mathbb{Q}^+ , la suma aritmética $+$ es interna en \mathbb{Q}^+ y el producto de ganancias es un racional estrictamente positivo.

$\bar{1}$ es elemento neutro para \odot ya que:

$$\bar{1} \odot \bar{0} = \bar{0} \odot \bar{1} = \bar{0}.$$

$$(r, w, g) \odot \bar{1} = (r, w, g) \odot (0, 0, 1) = (\max\{r, 0/g\}, w + 0/g, g \cdot 1) = (r, w, g).$$

$$\bar{1} \odot (r, w, g) = (0, 0, 1) \odot (r, w, g) = (\max\{0, r/1\}, 0 + w/1, 1 \cdot g) = (r, w, g).$$

Sólo falta demostrar la propiedad asociativa. Sean $s_1, s_2, s_3 \in S$. Se distinguen dos casos:

Caso 1) Algún s_i es $\bar{0}$. Nuevamente hay tres casos. Aplicando que $\bar{0}$ es elemento anulador de \odot , se tiene:

$$\bar{0} \odot (s_2 \odot s_3) = \bar{0} \quad \text{y} \quad (\bar{0} \odot s_2) \odot s_3 = \bar{0} \odot s_3 = \bar{0}.$$

$$s_1 \odot (\bar{0} \odot s_3) = s_1 \odot \bar{0} = \bar{0} \quad \text{y} \quad (s_1 \odot \bar{0}) \odot s_3 = \bar{0} \odot s_3 = \bar{0}.$$

$$s_1 \odot (s_2 \odot \bar{0}) = s_1 \odot \bar{0} = \bar{0} \quad \text{y} \quad (s_1 \odot s_2) \odot \bar{0} = \bar{0}.$$

Caso 2) Ningún s_i es $\bar{0}$. Sean $s_i = (r_i, w_i, g_i)$ para $i = 1, 2, 3$.

$$\begin{aligned} s_1 \odot (s_2 \odot s_3) &= (r_1, w_1, g_1) \odot \left(\max \left\{ r_2, \frac{r_3}{g_2} \right\}, w_2 + \frac{w_3}{g_2}, g_2 g_3 \right) = \\ &= \left(\max \left\{ r_1, \frac{1}{g_1} \max \left\{ r_2, \frac{r_3}{g_2} \right\} \right\}, w_1 + \frac{1}{g_1} \left(w_2 + \frac{w_3}{g_2} \right), g_1 (g_2 g_3) \right) = \\ &= \left(\max \left\{ r_1, \max \left\{ \frac{r_2}{g_1}, \frac{r_3}{g_1 g_2} \right\} \right\}, w_1 + \left(\frac{w_2}{g_1} + \frac{w_3}{g_1 g_2} \right), g_1 (g_2 g_3) \right) = \\ &= \left(\max \left\{ \max \left\{ r_1, \frac{r_2}{g_1} \right\}, \frac{r_3}{g_1 g_2} \right\}, \left(w_1 + \frac{w_2}{g_1} \right) + \frac{w_3}{g_1 g_2}, (g_1 g_2) g_3 \right) = \\ &= \left(\max \left\{ r_1, \frac{r_2}{g_1} \right\}, w_1 + \frac{w_2}{g_1}, g_1 g_2 \right) \odot (r_3, w_3, g_3) = (s_1 \odot s_2) \odot s_3. \end{aligned}$$

ii) $\bar{0}$ es anulador de \odot . Por definición.

iii) \oplus es conmutativa. Por definición.

iv) \oplus es idempotente. Por definición.

v) \odot es distributiva respecto de \oplus . Sean $s_1, s_2, s_3 \in S$. Se distinguen dos casos:

Caso 1) Algún s_i es $\bar{0}$. Nuevamente hay tres casos:

$$\bar{0} \odot (s_2 \oplus s_3) = \bar{0} \quad \text{y} \quad (\bar{0} \odot s_2) \oplus (\bar{0} \odot s_3) = \bar{0} \oplus \bar{0} = \bar{0}.$$

$$(s_2 \oplus s_3) \odot \bar{0} = \bar{0} \quad \text{y} \quad (s_2 \odot \bar{0}) \oplus (s_3 \odot \bar{0}) = \bar{0} \oplus \bar{0} = \bar{0}.$$

$$s_1 \odot (\bar{0} \oplus s_3) = s_1 \odot s_3 \quad \text{y} \quad (s_1 \odot \bar{0}) \oplus (s_1 \odot s_3) = \bar{0} \oplus (s_1 \odot s_3) = s_1 \odot s_3.$$

$$(\bar{0} \oplus s_3) \odot s_1 = s_3 \odot s_1 \quad \text{y} \quad (\bar{0} \odot s_1) \oplus (s_3 \odot s_1) = \bar{0} \oplus (s_3 \odot s_1) = s_3 \odot s_1.$$

$$s_1 \odot (s_2 \oplus \bar{0}) = s_1 \odot s_2 \quad \text{y} \quad (s_1 \odot s_2) \oplus (s_1 \odot \bar{0}) = (s_1 \odot s_2) \oplus \bar{0} = s_1 \odot s_2.$$

$$(s_2 \oplus \bar{0}) \odot s_1 = s_2 \odot s_1 \quad \text{y} \quad (s_2 \odot s_1) \oplus (\bar{0} \odot s_1) = (s_2 \odot s_1) \oplus \bar{0} = s_2 \odot s_1.$$

Caso 2) Todos los s_i son distintos de $\bar{0}$. Sean $s_i = (r_i, w_i, g_i)$ para $i = 1, 2, 3$. Deben ser $g_2 = g_3$ para que tengan sentido los resúmenes involucrados y se les denotará por g .

$$s_1 \odot (s_2 \oplus s_3) = (r_1, w_1, g_1) \odot \left(\max_{i=2,3} \{r_i - w_i\} + \min_{i=2,3} \{w_i\}, \min_{i=2,3} \{w_i\}, g \right) = \\ \left(\max \left\{ r_1, \frac{1}{g_1} \left(\max_{i=2,3} \{r_i - w_i\} + \min_{i=2,3} \{w_i\} \right) \right\}, w_1 + \frac{1}{g_1} \min_{i=2,3} \{w_i\}, g_1 g \right).$$

Por otro lado,

$$(s_1 \odot s_2) \oplus (s_1 \odot s_3) = \left(\max \left\{ r_1, \frac{r_2}{g_1} \right\}, w_1 + \frac{w_2}{g_1}, g_1 g \right) \oplus \left(\max \left\{ r_1, \frac{r_3}{g_1} \right\}, w_1 + \frac{w_3}{g_1}, g_1 g \right) = \\ \left(\max \left\{ \max \left\{ r_1, \frac{r_2}{g_1} \right\} - w_1 - \frac{w_2}{g_1}, \max \left\{ r_1, \frac{r_3}{g_1} \right\} - w_1 - \frac{w_3}{g_1} \right\} + \min \left\{ w_1 + \frac{w_2}{g_1}, w_1 + \frac{w_3}{g_1} \right\}, \right. \\ \left. \min \left\{ w_1 + \frac{w_2}{g_1}, w_1 + \frac{w_3}{g_1} \right\}, g_1 g \right).$$

Hay que demostrar que para cualesquiera s_1, s_2, s_3 las dos expresiones coinciden. Es evidente que la tercera componente es $g_1 g$ en las dos expresiones. También las segundas componentes coinciden ya que:

$$w_1 + \frac{1}{g_1} \min_{i=2,3} \{w_i\} = w_1 + \min \left\{ \frac{w_2}{g_1}, \frac{w_3}{g_1} \right\} = \min \left\{ w_1 + \frac{w_2}{g_1}, w_1 + \frac{w_3}{g_1} \right\}.$$

Sólo falta demostrar que también coinciden las primeras componentes. Partiendo de la primera componente de $(s_1 \odot s_2) \oplus (s_1 \odot s_3)$, se tiene:

$$\max \left\{ \max \left\{ r_1, \frac{r_2}{g_1} \right\} - w_1 - \frac{w_2}{g_1}, \max \left\{ r_1, \frac{r_3}{g_1} \right\} - w_1 - \frac{w_3}{g_1} \right\} + \min \left\{ w_1 + \frac{w_2}{g_1}, w_1 + \frac{w_3}{g_1} \right\} = \\ \max \left\{ \max \left\{ r_1, \frac{r_2}{g_1} \right\} - \frac{w_2}{g_1}, \max \left\{ r_1, \frac{r_3}{g_1} \right\} - \frac{w_3}{g_1} \right\} - w_1 + w_1 + \frac{1}{g_1} \min \{w_2, w_3\} = \\ \max \left\{ \max \left\{ r_1 - \frac{w_2}{g_1}, \frac{r_2 - w_2}{g_1} \right\}, \max \left\{ r_1 - \frac{w_3}{g_1}, \frac{r_3 - w_3}{g_1} \right\} \right\} + \frac{1}{g_1} \min_{i=2,3} \{w_i\} = \\ \max \left\{ \max \left\{ r_1 - \frac{w_2}{g_1}, r_1 - \frac{w_3}{g_1} \right\}, \max \left\{ \frac{r_2 - w_2}{g_1}, \frac{r_3 - w_3}{g_1} \right\} \right\} + \frac{1}{g_1} \min_{i=2,3} \{w_i\} = \\ \max \left\{ r_1 + \frac{1}{g_1} \max_{i=2,3} \{-w_i\}, \frac{1}{g_1} \max_{i=2,3} \{r_i - w_i\} \right\} + \frac{1}{g_1} \min_{i=2,3} \{w_i\} = \\ \max \left\{ r_1 - \frac{1}{g_1} \min_{i=2,3} \{w_i\}, \frac{1}{g_1} \max_{i=2,3} \{r_i - w_i\} \right\} + \frac{1}{g_1} \min_{i=2,3} \{w_i\} = \\ \max \left\{ r_1, \frac{1}{g_1} \left(\max_{i=2,3} \{r_i - w_i\} + \min_{i=2,3} \{w_i\} \right) \right\}.$$

Por lo tanto, $s_1 \odot (s_2 \oplus s_3) = (s_1 \odot s_2) \oplus (s_1 \odot s_3)$.

$$(s_2 \oplus s_3) \odot s_1 = \left(\max_{i=2,3} \{r_i - w_i\} + \min_{i=2,3} \{w_i\}, \min_{i=2,3} \{w_i\}, g \right) \odot (r_1, w_1, g_1) = \\ \left(\max_{i=2,3} \left\{ \max_{i=2,3} \{r_i - w_i\} + \min_{i=2,3} \{w_i\}, \frac{r_1}{g} \right\}, \min_{i=2,3} \{w_i\} + \frac{w_1}{g}, gg_1 \right).$$

Por otro lado,

$$(s_2 \odot s_1) \oplus (s_3 \odot s_1) = \left(\max \left\{ r_2, \frac{r_1}{g} \right\}, w_2 + \frac{w_1}{g}, gg_1 \right) \oplus \left(\max \left\{ r_3, \frac{r_1}{g} \right\}, w_3 + \frac{w_1}{g}, gg_1 \right) = \\ \left(\max \left\{ \max \left\{ r_2, \frac{r_1}{g} \right\} - w_2 - \frac{w_1}{g}, \max \left\{ r_3, \frac{r_1}{g} \right\} - w_3 - \frac{w_1}{g} \right\} + \min \left\{ w_2 + \frac{w_1}{g}, w_3 + \frac{w_1}{g} \right\}, \right. \\ \left. \min \left\{ w_2 + \frac{w_1}{g}, w_3 + \frac{w_1}{g} \right\}, gg_1 \right).$$

Hay que demostrar que para cualesquiera s_1, s_2, s_3 las dos expresiones coinciden. Es evidente que la tercera componente es gg_1 en las dos expresiones. También las segundas componentes coinciden ya que:

$$\min_{i=2,3} \{w_i\} + \frac{w_1}{g} = \min \left\{ w_2 + \frac{w_1}{g}, w_3 + \frac{w_1}{g} \right\}.$$

Sólo falta demostrar que también coinciden las primeras componentes. Partiendo de la primera componente de $(s_1 \odot s_2) \oplus (s_1 \odot s_3)$, se tiene:

$$\max \left\{ \max \left\{ r_2, \frac{r_1}{g} \right\} - w_2 - \frac{w_1}{g}, \max \left\{ r_3, \frac{r_1}{g} \right\} - w_3 - \frac{w_1}{g} \right\} + \min \left\{ w_2 + \frac{w_1}{g}, w_3 + \frac{w_1}{g} \right\} = \\ \max \left\{ \max \left\{ r_2, \frac{r_1}{g} \right\} - w_2, \max \left\{ r_3, \frac{r_1}{g} \right\} - w_3 \right\} - \frac{w_1}{g} + \frac{w_1}{g} + \min_{i=2,3} \{w_i\} = \\ \max \left\{ \max \left\{ r_2 - w_2, \frac{r_1}{g} - w_2 \right\}, \max \left\{ r_3 - w_3, \frac{r_1}{g} - w_3 \right\} \right\} + \min_{i=2,3} \{w_i\} = \\ \max \left\{ \max \{r_2 - w_2, r_3 - w_3\}, \max \left\{ \frac{r_1}{g} - w_2, \frac{r_1}{g} - w_3 \right\} \right\} + \min_{i=2,3} \{w_i\} = \\ \max \left\{ \max_{i=2,3} \{r_i - w_i\}, \frac{r_1}{g} + \max_{i=2,3} \{-w_i\} \right\} + \min_{i=2,3} \{w_i\} = \\ \max \left\{ \max_{i=2,3} \{r_i - w_i\}, \frac{r_1}{g} - \min_{i=2,3} \{w_i\} \right\} + \min_{i=2,3} \{w_i\} = \\ \max \left\{ \max_{i=2,3} \{r_i - w_i\} + \min_{i=2,3} \{w_i\}, \frac{r_1}{g} \right\}.$$

Por lo tanto, $(s_2 \oplus s_3) \odot s_1 = (s_2 \odot s_1) \oplus (s_3 \odot s_1)$.

vi) Si $\{s_i\}_{i=1}^{\infty}$ es una sucesión de etiquetas, $\bigoplus_{i=1}^{\infty} s_i$ está bien definida y en S : Si algún camino de la sucesión tiene etiqueta $\bar{0}$, se puede eliminar del resumen. Una vez eliminadas las etiquetas $\bar{0}$, si queda un resumen finito, el resumen existe y está en S por ser \oplus operación interna.

El caso que falta es si en la sucesión existen infinitos $s_i \neq \bar{0}$. Para demostrar este caso se tendrá en cuenta la relajación comentada en la definición 4.26 respecto a las sucesiones de etiquetas que pueden aparecer en WTS. Dadas dos transiciones t y t' , para cada $i \in \mathbb{N}$ sea π_i un camino de t a t' cuya etiqueta sea s_i . Para que tenga sentido el resumen, todos los s_i son de la forma $s_i = (r_i, w_i, g)$, es decir, tienen la misma ganancia. Por lo tanto, el

resumen de la sucesión tendrá la misma ganancia que todos los s_i .

La segunda componente del resumen es $\min_{i=1}^{\infty} \{w_i\}$. En general, puede no existir el mínimo de una sucesión de números racionales, pero como se ha demostrado en el apartado *vi* de la propiedad 4.27, en el resumen de las etiquetas de una sucesión de caminos en un WTS vivo y limitado con las mismas transiciones origen y destino siempre existe un camino con marcado ponderado mínimo y por lo tanto, este marcado ponderado mínimo es un racional no negativo.

Sea π_j un camino del resumen con marcado ponderado mínimo. Entonces $\max_{i=1}^{\infty} \{r_i - w_i\} + \min_{i=1}^{\infty} \{w_i\} = \max_{i=1}^{\infty} \{r_i - w_i\} + w_j \geq r_j - w_j + w_j = r_j \geq 0$. Por lo tanto, si existe la primera componente del resumen, es no negativa. Sólo queda demostrar que existe el $\max_{i=1}^{\infty} \{r_i - w_i\}$. Si el camino π_i tiene etiqueta asociada $s_i = (r_i, w_i, g)$, entonces π_i tiene ganancia g , marcado ponderado libre w_i y resistencia r_i . Como en todo resumen siempre hay un camino con marcado ponderado mínimo (ver apartado *vi* de la propiedad 4.27), existe $j \in \mathbb{N}$ tal que $w_j = 0$, es decir, $\max_{i=1}^{\infty} \{r_i - w_i\} \geq 0$. Por lo tanto, basta con demostrar que $\{r_i - w_i \mid r_i - w_i \geq 0, i \in \mathbb{N}\}$ es finito.

Por definición del operador \oplus , se tiene que la resistencia de un camino es su resistencia estructural disminuida por el marcado ponderado atrapado hasta el momento en el camino debido a su sincronización con otros caminos. Se denotará a partir de ahora $RE(\pi_i)$ (resistencia estructural) por RE_i y sea $W_i = RE_i - r_i$ para todo $i \in \mathbb{N}$ (notar que es diferente la notación W_i a la w_i de marcado ponderado libre). Se demostrará a continuación que $\{RE_i \mid i \in \mathbb{N}\}$, $\{W_i \mid W_i \leq RE_i, i \in \mathbb{N}\}$ y $\{w_i \mid w_i \leq r_i, i \in \mathbb{N}\}$ son finitos.

Cada camino π_i está compuesto por un conjunto de aristas del grafo correspondiente al WTS. Sea $\{a_i\}_{i=1}^m$ el conjunto de aristas del grafo. Por la definición 4.11, cada arista a_i del WTS aporta a la resistencia estructural de un camino que la contenga una cantidad fija $u_i \in \mathbb{Q}^+ \setminus \{0\}$ igual a la resistencia estructural de la arista dividida por la ganancia entre el vértice de comienzo del camino y el primer vértice de la arista. Como todos los caminos del resumen tienen el mismo vértice de origen, los u_j son los mismos para todos los caminos. Para cada camino π_i , sea $\rho_i = (n_{i1}, \dots, n_{im})$ un vector de m variables lógicas tales que $n_{ij} = 1$ si el camino π_i contiene a la arista a_j y $n_{ij} = 0$ en caso contrario. Entonces $RE_i = \max\{u_j \mid n_{ij} = 1\}$ para todo $i \in \mathbb{N}$. Como el conjunto de u_j es finito (como mucho el número de aristas del WTS), sólo hay un número finito de resistencias estructurales diferentes en todos los caminos de un WTS, en particular $\{RE_i \mid i \in \mathbb{N}\}$ es finito, luego existe $R = \max_{i=1}^{\infty} \{RE_i\}$.

En el apartado *vi* de la propiedad 4.27 se demuestra que en el resumen de cualquier sucesión de caminos sólo hay un número finito de marcados ponderados menores o iguales que cualquier cantidad dada. Como los w_i son marcados ponderados de ciertos caminos del resumen, $\{w_i \mid w_i \leq R, i \in \mathbb{N}\}$ es finito y como $\{w_i \mid w_i \leq r_i, i \in \mathbb{N}\} \subseteq \{w_i \mid w_i \leq R, i \in \mathbb{N}\}$, entonces el conjunto $\{w_i \mid w_i \leq r_i, i \in \mathbb{N}\}$ es finito. Análogamente, los W_i son marcados ponderados atrapados en caminos, es decir, diferencias entre marcados ponderados de caminos del resumen, por lo que $\{W_i \mid W_i \leq R \in \mathbb{N}\}$ es finito. Como $\{W_i \mid W_i \leq RE_i, i \in \mathbb{N}\} \subseteq \{W_i \mid W_i \leq R \in \mathbb{N}\}$, entonces $\{W_i \mid W_i \leq RE_i, i \in \mathbb{N}\}$ es finito.

Resumiendo, $\{RE_i \mid i \in \mathbb{N}\}$ y $\{W_i \mid W_i \leq RE_i, i \in \mathbb{N}\}$ son finitos, es decir, $\{RE_i - W_i \mid RE_i - W_i \geq 0, i \in \mathbb{N}\}$ es finito. Como $r_i = RE_i - W_i$ para todo $i \in \mathbb{N}$, $\{r_i \mid r_i \geq 0, i \in \mathbb{N}\}$ es finito. También $\{w_i \mid w_i \leq r_i, i \in \mathbb{N}\}$ es finito, luego $\{r_i - w_i \mid r_i - w_i \geq 0, i \in \mathbb{N}\}$ es finito y no vacío (como se ha visto antes). Por lo tanto $\max_{i=1}^{\infty} \{r_i - w_i\} = \max\{r_i - w_i \mid r_i - w_i \geq 0\}$. Además, $r_i - w_i \in \mathbb{Q}$ para todo $i \in \mathbb{N}$, luego $\max_{i=1}^{\infty} \{r_i - w_i\} \in \mathbb{Q}^+$.

vii) Se cumplen las propiedades conmutativa, asociativa e idempotencia en el resumen de sucesiones: Si algún camino de la sucesión tiene etiqueta $\bar{0}$ se puede eliminar del resumen, lo que equivale a hacer un resumen con un único camino de etiqueta $\bar{0}$. En el resto de casos, debido a la conmutatividad e idempotencia del operador \oplus , las etiquetas de la forma (r, w, g) pueden conmutar con el resto de etiquetas y eliminarse las instancias repetidas.

viii) \odot es distributiva respecto de resúmenes de sucesiones. Sean $a, b_i \in S$ con $i \in \mathbb{N}$. Se distinguen dos casos:

Caso 1) Todos los $b_i = \bar{0}$. Entonces $a \odot (\bigoplus_{i=1}^{\infty} b_i) = a \odot (\bigoplus_{i=1}^{\infty} \bar{0}) = a \odot \bar{0} = \bar{0}$ y $\bigoplus_{i=1}^{\infty} (a \odot b_i) = \bigoplus_{i=1}^{\infty} (a \odot \bar{0}) = \bigoplus_{i=1}^{\infty} \bar{0} = \bar{0}$.

$(\bigoplus_{i=1}^{\infty} b_i) \odot a = (\bigoplus_{i=1}^{\infty} \bar{0}) \odot a = \bar{0} \odot a = \bar{0}$ y $\bigoplus_{i=1}^{\infty} (b_i \odot a) = \bigoplus_{i=1}^{\infty} (\bar{0} \odot a) = \bigoplus_{i=1}^{\infty} \bar{0} = \bar{0}$.

Caso 2) Algún $b_i \neq \bar{0}$. Por el apartado *vii*, se puede suponer también que los $b_i \neq \bar{0}$ para todo $i \in \mathbb{N}$. En el apartado *vi* de la propiedad 4.27 se demuestra que en el resumen de cualquier sucesión de caminos siempre existe un camino con marcado ponderado mínimo y en el apartado *vi* de esta propiedad se ha demostrado que también existe un camino con resistencia máxima. Aplicando inducción al apartado *v*, se tiene que el operador \odot es distributivo respecto a resúmenes finitos. En estas condiciones, para el resumen $\bigoplus_{i=1}^{\infty} b_i$ existen $n, m \in \mathbb{N}$ tales que $r_n - w_n = \max_{i=1}^{\infty} \{r_i - w_i\}$ y $w_m = \min_{i=1}^{\infty} \{w_i\}$. Por lo tanto, $\bigoplus_{i=1}^{\infty} b_i = b_n \oplus b_m$. En particular, existe $k_1 \in \mathbb{N}$ tal que $\bigoplus_{i=1}^{\infty} b_i = \bigoplus_{i=1}^{k_1} b_i$ para todo $k \geq k_1$ (basta tomar en este

caso $k_1 = \max\{n, m\}$. De la misma forma, para el otro resumen existe $k_2 \in \mathbb{N}$ tal que $\bigoplus_{i=1}^{\infty} (a \odot b_i) = \bigoplus_{i=1}^k (a \odot b_i)$ para todo $k \geq k_2$. Sea ahora $k = \max\{k_1, k_2\}$. Entonces $a \odot (\bigoplus_{i=1}^{\infty} b_i) = a \odot (\bigoplus_{i=1}^k b_i) = \bigoplus_{i=1}^k (a \odot b_i) = \bigoplus_{i=1}^{\infty} (a \odot b_i)$.

Para el mismo valor de k , $(\bigoplus_{i=1}^{\infty} b_i) \odot a = (\bigoplus_{i=1}^k b_i) \odot a = \bigoplus_{i=1}^k (b_i \odot a) = \bigoplus_{i=1}^{\infty} (b_i \odot a)$. \diamond

Algoritmo 4.31 Algoritmo de cálculo de resistencias máximas.

input: $G := (V, A)$, \mathbf{m}_0 grafo dirigido asociado al WTS y marcado inicial
Vector \mathbf{g} de ganancias (salida del algoritmo 4.23)

R Matriz de resistencias entre vértices del grafo

L Matriz de marcados ponderados libres entre vértices del grafo

K Matriz de vértices de paso de los caminos de máxima resistencia

$n :=$ número de transiciones del WTS

$n' :=$ número de transiciones internas del WTS

for $i := 1$ **to** n **do**

for $j := 1$ **to** n **do**

$\mathbf{K}(i, j) := 0$

if $i = j$ **then** $\mathbf{R}(i, j) := 0$; $\mathbf{L}(i, j) := 0$

else if $(i, j) \in A$ **then**

$\mathbf{R}(i, j) := W(p_{ij}, t_j) / W(t_i, p_{ij})$

$\mathbf{L}(i, j) := \mathbf{m}_0[p_{ij}] / W(t_i, p_{ij})$

else $\mathbf{R}(i, j) := 0$; $\mathbf{L}(i, j) := \infty$

end for

end for

for $k := 1$ **to** n **do**

for $i := 1$ **to** n , $i \neq k$ **do**

for $j := 1$ **to** n , $j \neq k$ **do**

$l := \mathbf{L}(i, k) + \frac{\mathbf{L}(k, j)\mathbf{g}(j)}{\mathbf{g}(k)}$; $m := \min\{\mathbf{L}(i, j), l\}$

$r := \max\left\{\mathbf{R}(i, k), \frac{\mathbf{R}(k, j)\mathbf{g}(j)}{\mathbf{g}(k)}\right\} - l + m$

$\mathbf{R}(i, j) := \mathbf{R}(i, j) - \mathbf{L}(i, j) + m$

if $m < \mathbf{L}(i, j)$ **then** $\mathbf{L}(i, j) := m$

if $r > \mathbf{R}(i, j)$ **then** $\mathbf{R}(i, j) := r$; $\mathbf{K}(i, j) := k$

end for

end for

end for

output: Matrices **R**, **L**, **K**

Una vez demostrada la estructura de semianillo cerrado en sentido laxo para el dominio de etiquetas S con las operaciones \oplus y \odot , siguiendo el marco general para la resolución de problemas de caminos en grafos dirigidos de [CLR90], el algoritmo 3.9 con los operadores \oplus y \odot de la definición 4.29 permite calcular el resumen de todos los caminos que unen dos vértices cualesquiera del grafo inicial.

El algoritmo 4.31 muestra una codificación eficiente para el cálculo de resistencias máximas entre transiciones de interfaz en WTS's. No es necesario mantener las etiquetas de tres componentes como en el desarrollo teórico, sino únicamente el par resistencia, marcado ponderado libre. La información relativa a la ganancia entre transiciones se tiene almacenada en el vector \mathbf{g} de ganancias del algoritmo 4.23.

Se denota con n al número de transiciones del WTS y con n' al número de transiciones internas. Se numeran las transiciones de forma que primero aparezcan las transiciones internas. De esta forma se tiene un test inmediato de si una transición es o no de interfaz. Además, con esta numeración se pueden obtener de forma eficiente todas las resistencias máximas entre transiciones de interfaz al final de la ejecución del algoritmo, ya que todas estas transiciones aparecen de forma consecutiva en la numeración.

La entrada de este algoritmo consta de la descripción de la estructura y marcado inicial del WTS por medio de su grafo dirigido asociado G y \mathbf{m}_0 . También se incluye como entrada el vector \mathbf{g} de ganancias, para no repetir los cálculos del algoritmo 4.23. A partir de la descripción del WTS, se construyen dos matrices \mathbf{R} y \mathbf{L} . La matriz \mathbf{R} sirve para almacenar las resistencias máximas entre transiciones cualesquiera y la matriz \mathbf{L} para almacenar los marcados ponderados libres de los caminos de resistencia máxima. Inicialmente los elementos de la diagonal $\mathbf{R}(i, i)$ y $\mathbf{L}(i, i)$ son 0. Estos valores nulos son el resultado correspondiente a la operación $\bar{\mathbf{I}} \oplus \lambda(i, i)$ del algoritmo general de resúmenes (algoritmo 3.9). En este caso, el camino vacío tiene resistencia y marcado ponderado 0 (ya que $\bar{\mathbf{I}} = (0, 0, 1)$) y cualquier etiqueta $\lambda(i, i)$ del grafo asociado al WTS corresponde a lugar que tenga de entrada y salida la transición t_i , es decir, forma con ella un ciclo. Por la proposición 4.14, todo ciclo en un WTS vivo y limitado tiene marcado ponderado mayor o igual que su resistencia estructural, es decir, la etiqueta $\lambda(i, i)$ debe ser de la forma $(r, w, 1)$ con $w \geq r$ y por lo tanto $\bar{\mathbf{I}} = (0, 0, 1) \oplus (r, w, 1) = (0, 0, 1) = \bar{\mathbf{I}}$. Para elementos distintos de la diagonal, $\mathbf{R}(i, j)$ debe ser la resistencia estructural del camino simple que une los vértices ó 0 si este camino no existe (indicando que de momento el

camino de máxima resistencia es el camino inexistente que tiene la menor de las resistencias posibles). Y los elementos $\mathbf{L}(i, j)$ deben ser el marcado ponderado inicial del camino simple que une los vértices o ∞ si este camino no existe (indicando que de momento el camino de máxima resistencia tiene el máximo marcado ponderado posible).

El paso siguiente (los tres bucles encajados) consiste en realizar los cálculos de caminos de máxima resistencia. Se eliminan de los bucles las comparaciones triviales en WTS's, que son los casos en que $i = k$ o $j = k$. En el algoritmo general de resúmenes esta operación consiste en calcular la ecuación (3.2). En este caso, la clausura $(l_{kk}^{(k-1)})^*$ es $\bar{1} \oplus (\bigoplus_{i=1}^{\infty} (\bigodot_{j=1}^i l_{kk}^{(k-1)})) = \bar{1}$ por lo que puede desaparecer del resumen, simplificando la expresión. La expresión está codificada en las instrucciones del interior de los tres bucles encajados. Estas instrucciones tienen por objeto realizar el resumen de la etiqueta del camino de resistencia máxima del vértice i al j calculado hasta el momento con la de un nuevo camino, el formado por la concatenación del camino de resistencia máxima del vértice i al k con el de resistencia máxima del vértice k al j (también calculados hasta ese momento). Para ello, primero se calcula el marcado ponderado libre l del camino concatenación. Después se calcula m , que es el menor de los marcados ponderados libres de los dos caminos a reducir. El resultado del resumen tendrá como marcado ponderado libre m . El siguiente paso consiste en calcular las resistencias de los dos caminos involucrados, teniendo en cuenta que la sincronización de los mismos puede disminuir la resistencia de alguno de los dos. En el algoritmo, r es la resistencia del camino concatenación y $\mathbf{R}(i, j)$ es la nueva resistencia del resumen realizado hasta ese momento, teniendo en cuenta que la nueva sincronización puede hacer disminuir su valor. Por último, se actualizan los valores de $\mathbf{R}(i, j)$ y $\mathbf{L}(i, j)$ con la mayor de las resistencias y el marcado ponderado libre calculado.

De esta manera, tras la iteración k del bucle exterior, $\mathbf{R}(i, j)$ proporciona la resistencia del camino de máxima resistencia que une el vértice i con el j y que está compuesto únicamente por vértices de índice menor o igual que k . Por su parte, $\mathbf{L}(i, j)$ proporciona (también tras la iteración k) el marcado ponderado libre del camino de resistencia máxima calculado hasta ese momento, lo que equivale al menor marcado ponderado de entre todos los caminos que unen el vértice i con el j y que están compuestos únicamente por vértices de índice menor o igual que k .

Al finalizar el algoritmo, se tiene en $\mathbf{R}(i, j)$ la resistencia máxima de entre todos los caminos que unen el vértice i con el j y en $\mathbf{L}(i, j)$ el marcado

ponderado libre de ese camino de máxima resistencia, que equivale al menor marcado ponderado de entre todos los caminos que unen el vértice i con el j . Notar que este algoritmo de cálculo de resistencias máximas calcula también los marcados ponderados mínimos lo que simplifica la implementación práctica de la técnica que se está desarrollando. Se ha expuesto de esta manera para hacer más comprensibles los distintos algoritmos involucrados.

En este caso se ha incluido en el algoritmo la matriz \mathbf{K} (ver algoritmo 3.7 para una explicación detallada de su funcionamiento) para almacenar de forma eficiente los vértices de paso de los caminos de máxima resistencia. Esta es una matriz de tamaño $n \times n$ que se pone inicialmente a 0 y en la que cada vez que se encuentra un camino de mayor resistencia entre dos transiciones dadas que el que se tenía previamente, se almacena un vértice de paso del mismo. De esta forma, al finalizar el algoritmo es posible reconstruir de forma recursiva todos los vértices de paso de cualquier camino de resistencia máxima.

La complejidad de este algoritmo es de $O(n^3)$ en tiempo y $O(n^2)$ en espacio, donde n es el número de transiciones del WTS original. Esta complejidad es la misma que la del algoritmo de Floyd, por lo que el desarrollo teórico realizado no aumenta el coste computacional del algoritmo resultante. Por lo tanto, se ha desarrollado un algoritmo que permite calcular, para dos transiciones cualesquiera del WTS original, el camino de resistencia máxima que las une. Notar que para la reducción de WTS's sólo interesan las resistencias máximas entre transiciones de interfaz, es decir, los valores $\mathbf{R}(i, j)$ para $i, j > n'$.

Para realizar la descomposición estructural de los WTS's es necesario preservar los caminos de máxima resistencia. Si se mantienen todos los nodos de estos caminos, la reducción de las subredes puede ser muy pobre. Por ello, es conveniente reducir estos caminos preservando su nivel de resistencia. Por la proposición 4.12, todo camino alcanza su resistencia estructural en alguna transición. Si se descompone el camino original en dos subcaminos partiéndolo por esa transición, es posible realizar una reducción de todo el camino por medio de una transición y dos lugares TT-MSIP preservando la ganancia y la resistencia del mismo. En consecuencia, es necesario conocer, para cada camino de máxima resistencia entre transiciones de interfaz, cuál es la transición en la que se alcanza su resistencia estructural. Además será necesario conocer el marcado ponderado de los dos subcaminos en que queda dividido, para poder reducirlos. Estos cálculos se puede realizar con el siguiente algoritmo empleando la matriz \mathbf{K} del algoritmo 4.31.

Algoritmo 4.32 Algoritmo de reducción de caminos.

input: $G := (V, A)$, \mathbf{m}_0 grafo dirigido asociado al WTS y marcado inicial
 \mathbf{g} Vector de ganancias (salida del algoritmo 4.23)

\mathbf{K} Matriz del algoritmo 4.31 (caminos de máxima resistencia)

Q pila de pares de vértices por visitar

\mathbf{V} vector de vértices de un camino

\mathbf{E} , \mathbf{D}_1 , \mathbf{D}_2 matrices para resultados (inicialmente a 0)

$n :=$ número de transiciones del WTS

$n' :=$ número de transiciones internas del WTS

for $i := n' + 1$ **to** n **do**

$ns := l$ tal que $t_i \in \mathcal{N}_l$

for $j := n' + 1$ **to** n , $j \neq i$ **do**

$c := 1$; $\mathbf{V}(1) := i$; inserta (i, j) en Q

while $Q \neq \emptyset$ **do**

$(v_1, v_2) :=$ desapila(Q)

if $\mathbf{K}(v_1, v_2) \neq 0$ **then**

inserta $(\mathbf{K}(v_1, v_2), v_2)$ en Q

inserta $(v_1, \mathbf{K}(v_1, v_2))$ en Q

else

$c := c + 1$; $ns_2 := l$ tal que $t_j \in \mathcal{N}_l$

if $ns = ns_2$ **then** $\mathbf{V}(c) := v_2$

else vacía(Q); $ns_2 := 0$

end if

end while

if $ns_2 > 0$ **then**

$a := \infty$; $w := 0$

for $k := 2$ **to** c **do**

$v_1 := \mathbf{V}(k - 1)$; $v_2 := \mathbf{V}(k)$; $w := w + \frac{\mathbf{m}_0[p_{v_1 v_2}] \mathbf{g}(i)}{W(t_{v_1}, p_{v_1 v_2}) \mathbf{g}(v_1)}$

if $\mathbf{g}(v_2)/\mathbf{g}(i) < a$ **then** $m := v_2$; $a := \mathbf{g}(m)/\mathbf{g}(i)$; $d := w$

end for

if $m \leq n'$ and $a < \frac{\mathbf{g}(j)}{\mathbf{g}(i)}$ **then**

$\mathbf{E}(i, j) := m$; $\mathbf{D}_1(i, j) := d$; $\mathbf{D}_2(i, j) := (w - d) \frac{\mathbf{g}(m)}{\mathbf{g}(i)}$

end if

end if

end for

end for

output: Matrices \mathbf{D}_1 , \mathbf{D}_2 , \mathbf{E}

Este algoritmo calcula, para cada camino de resistencia máxima entre transiciones de interfaz, toda la información necesaria para su reducción. Esta información consta de tres valores. Por un lado el índice de la transición en la que alcanza su resistencia estructural. Esta información se almacena en la matriz \mathbf{E} . Tomando esta transición como punto de división del camino de resistencia máxima, las matrices \mathbf{D}_1 y \mathbf{D}_2 proporcionan el marcado ponderado de los dos subcaminos resultantes de la división. Estos marcados ponderados son necesarios para calcular el marcado inicial de los lugares TT-MSIP que resumirán el comportamiento del camino. Se aprovecha el recorrido de los caminos de resistencia máxima para eliminar algunos caminos cuyo resumen puede construirse a partir de los resúmenes de otros más cortos.

La entrada de este algoritmo consta de la descripción de la estructura y marcado inicial del WTS por medio de su grafo dirigido asociado G y \mathbf{m}_0 . Además, se incluyen como entrada el vector \mathbf{g} de ganancias entre transiciones (salida del algoritmo 4.23) y la matriz \mathbf{K} (salida del algoritmo 4.31) con información sobre los vértices de paso de los caminos de resistencia máxima del WTS. Como sólo interesa hacer los cálculos para caminos entre transiciones de interfaz, el par de bucles exteriores recorren sólo los índices correspondientes a las transiciones de interfaz ($n' < i, j \leq n$). Dado un par (i, j) de transiciones de interfaz, la primera tarea consiste en encontrar los vértices de paso del camino de máxima resistencia de i a j . Este cálculo se realiza en el bucle de tipo “while”, que no es más que un recorrido en profundidad de las aristas del camino en cuestión dirigido por la matriz \mathbf{K} . Para hacer un recorrido en profundidad es necesaria una estructura de tipo pila (Q en el algoritmo) en la que se almacenan los subcaminos que quedan por recorrer. Si un cierto elemento $\mathbf{K}(i, j) = 0$, quiere decir que el camino de resistencia máxima de i a j es la arista (i, j) del grafo G . Esta arista se almacena en el vector temporal \mathbf{V} que contiene las aristas del camino que se pretende recorrer. Por el contrario, si $\mathbf{K}(i, j) \neq 0$ quiere decir que el camino de resistencia máxima de i a j no es elemental y que pasa por el vértice $\mathbf{K}(i, j)$, es decir, es la concatenación del camino de resistencia máxima de i a $\mathbf{K}(i, j)$ con el de resistencia máxima de $\mathbf{K}(i, j)$ a j . Por lo tanto, estos dos subcaminos se apilan para su exploración posterior. El orden de apilado es el adecuado para conseguir el recorrido ordenado de las aristas del camino de resistencia máxima. Al finalizar el bucle de tipo “while”, el vector \mathbf{V} contiene por orden los vértices por los que pasa el camino de resistencia máxima de i a j y la variable c contiene el número de vértices del camino. Según

se realiza el recorrido de un camino de resistencia máxima se comprueba que todos sus vértices pertenezcan a la misma subred (valores de ns y ns_2). Si un camino de resistencia máxima pasa por más de una subred se para el recorrido ya que el resumen de este camino se podrá construir a partir de los resúmenes de caminos que pertenezcan a una sola subred (estos son algunos de los caminos que se eliminan).

El siguiente paso consiste en calcular para este camino el vértice en el que se alcanza su resistencia estructural y los marcados ponderados de los subcaminos en que se divide el camino original. Este cálculo se realiza en el último bucle interno. Para ello se emplea el vector \mathbf{g} de ganancias y se simplifican los cálculos. Por la proposición 4.12, existe un subcamino del de máxima resistencia con el mismo origen cuya ganancia es el inverso de la resistencia estructural del camino. Se recorren los vértices del camino para encontrar el subcamino con el mismo origen de menor ganancia (que equivale a mayor resistencia estructural). A la vez que se recorre el camino se va calculando también el marcado ponderado del mismo (valor que se almacena en la variable temporal w). A la salida del recorrido del camino, la variable m contiene el vértice de división del camino inicial en dos subcaminos, la variable d el marcado ponderado del primer subcamino y la variable w el marcado ponderado total del camino de resistencia máxima.

Para no descartar el camino estudiado se exigen dos condiciones. La primera consiste en que alcance su resistencia estructural en una transición interna de \mathcal{N} , no en una de interfaz (test $m \leq n'$ en el algoritmo). En caso contrario, sería un camino cuyo resumen se puede formar a partir de los resúmenes de sus dos subcaminos, que también conectan transiciones de interfaz. Y la segunda condición consiste en que tenga una resistencia estructural estrictamente mayor que el inverso de su ganancia (test $a < \mathbf{g}(j)/\mathbf{g}(i)$ en el algoritmo). Si el camino tiene como resistencia estructural el inverso de su ganancia, un TT-MSIP que conecte sus transiciones inicial y final preserva tanto la ganancia como la resistencia del camino. Es más, en estas condiciones el camino debe ser forzosamente un camino de marcado ponderado mínimo y por lo tanto ya se tiene calculada toda la información necesaria para reducirlo con el algoritmo 4.28. En cualquiera de los dos casos pueden desecharse los cálculos realizados, porque no se van a traducir en la existencia de nodos para la reducción del camino en cuestión.

Si el vértice de división del camino corresponde a una transición interna, entonces se almacenan en las matrices de salida del algoritmo los valores necesarios para la reducción del camino (vértice de división en la matriz \mathbf{E} y

marcados ponderados de los dos subcaminos que se forman en las matrices \mathbf{D}_1 y \mathbf{D}_2).

La complejidad en tiempo de este algoritmo es $O(n^3)$ (porque el máximo número de vértices de un camino de resistencia máxima es $n + 1 = O(n)$ y en espacio $O(n^2)$, donde n es el número de transiciones del WTS original, la misma que la de algoritmos anteriores.

Una vez desarrollados los algoritmos necesarios para calcular las ganancias, marcados ponderados mínimos y resistencias máximas entre transiciones de interfaz, se tienen ya los elementos teóricos necesarios para realizar la descomposición estructural de WTS's.

Definición 4.33 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado, $B \subseteq P$ un K -corte de \mathcal{N} y $\mathcal{N}_i = \langle P_i, T_i, F_i, W_i \rangle$ con $1 \leq i \leq K$ las subredes producto de B . Sean \mathbf{L} , \mathbf{B} , \mathbf{E} , \mathbf{D}_1 y \mathbf{D}_2 las matrices resultado de aplicar a $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ los algoritmos 4.28 y 4.32. Para cada $1 \leq i \leq K$ se construyen los conjuntos H_i , I_i y H'_i de la siguiente manera:

$$\begin{aligned}
H_i &= \{ p_{kj} \text{ lugar} \mid t_k, t_j \in T_i \cap \text{TI}, \mathbf{B}(k, j) = 0 \text{ y si } G(t_k, t_j) = \frac{p}{q} \text{ entonces} \\
&\quad W(t_k, p_{kj}) = p, W(p_{kj}, t_j) = q \text{ y } \mathbf{m}_0[p_{kj}] = \lfloor p \cdot \mathbf{L}(k, j) \rfloor \} \\
I_i &= \{ t_{im} \text{ transición inmediata} \mid \exists t_k, t_j \in T_i \cap \text{TI} \text{ tales que} \\
&\quad \mathbf{E}(k, j) = m > 0 \text{ y } t_m \in T_i \} \\
H'_i &= \{ p_{kj}^1, p_{kj}^2 \text{ lugares} \mid t_k, t_j \in T_i \cap \text{TI}, \mathbf{E}(k, j) = m > 0, t_m \in T_i \text{ y si} \\
&\quad G(t_k, t_m) = \frac{p_1}{q_1}, G(t_m, t_j) = \frac{p_2}{q_2} \text{ entonces} \\
&\quad W(t_k, p_{kj}^1) = p_1, W(p_{kj}^1, t_m) = q_1 \text{ y } \mathbf{m}_0[p_{kj}^1] = \lfloor p_1 \cdot \mathbf{D}_1(k, j) \rfloor \\
&\quad W(t_m, p_{kj}^2) = p_2, W(p_{kj}^2, t_j) = q_2 \text{ y } \mathbf{m}_0[p_{kj}^2] = \lfloor p_2 \cdot \mathbf{D}_2(k, j) \rfloor \}
\end{aligned}$$

Se definen los conjuntos H , I y H' como $H = \bigcup_{i=1}^K H_i$, $I = \bigcup_{i=1}^K I_i$ y $H' = \bigcup_{i=1}^K H'_i$.

El conjunto H_i es el conjunto de lugares TT-MSIP que resumen los caminos de marcado ponderado mínimo entre transiciones de interfaz de la subred \mathcal{N}_i . El conjunto H es el conjunto unión de los H_i . Estos conjuntos se construyen directamente a partir de la salida del algoritmo 4.28 observando los valores de las matrices \mathbf{L} y \mathbf{B} entre transiciones de interfaz. Dadas dos transiciones de interfaz diferentes t_k y t_j , en este caso $\mathbf{L}(k, j) < \infty$ ya que el WTS es fuertemente conexo (teorema 2.50) y por lo tanto siempre

existe un camino que une la transición t_k con la t_j . Si $\mathbf{B}(k, j) = 1$, quiere decir que existe un camino de t_k a t_j de marcado ponderado mínimo que pasa por otra transición de interfaz t_l . Aplicando el teorema 4.21, el lugar implícito p_{kj} puede ser eliminado porque es implícito respecto a los lugares p_{il} y p_{lj} . Por ello el lugar p_{kj} no se incluye en el conjunto H . Al construir los conjuntos H_i sólo interesan los lugares que conectan transiciones de interfaz de la misma subred. La razón es que si se toma el camino de marcado ponderado mínimo entre dos transiciones de interfaz de distintas subredes hay dos posibilidades. La primera es que este camino de marcado mínimo no pase por otra transición de interfaz. Entonces, el camino está formado únicamente por un lugar y las dos transiciones consideradas, es decir, el lugar es un lugar del corte que ya forma parte de la red original (no hace falta añadirlo). La segunda posibilidad es que el camino de marcado ponderado mínimo pase por otra transición de interfaz. En este caso, el lugar que lo resume es implícito con respecto a otros lugares (teorema 4.21) y puede ser eliminado del conjunto H . En consecuencia, el conjunto H_i está formado únicamente por lugares TT-MSIP que conectan transiciones de interfaz de la subred \mathcal{N}_i y que no son implícitos respecto a otros lugares de H_i . Estos lugares constituyen el resumen de todos los caminos de marcado ponderado mínimo entre transiciones de interfaz de la subred \mathcal{N}_i .

El conjunto I_i es el de transiciones inmediatas necesarias para reducir los caminos de máxima resistencia entre transiciones de interfaz de la subred \mathcal{N}_i . Sólo interesan estos caminos, porque los caminos de resistencia máxima entre transiciones de interfaz de distintas subredes se pueden descomponer en subcaminos que conectan transiciones de interfaz de la misma subred más algunos lugares del corte. El resumen de estos caminos puede conseguirse por medio de los resúmenes de los caminos entre transiciones de interfaz de la misma subred más algunos lugares del corte. El conjunto I es el conjunto unión de los I_i . El cálculo de estas transiciones es inmediato a partir de la salida del algoritmo 4.32 observando los valores de la matriz \mathbf{E} entre transiciones de interfaz de la subred \mathcal{N}_i . Si $\mathbf{E}(k, j) = 0$ el camino no requiere transición inmediata para su resumen o bien no es necesario su resumen por quedar contenido en otros. Si $\mathbf{E}(k, j) = m > 0$ con t_k y t_j transiciones de interfaz de la subred \mathcal{N}_i , quiere decir que la reducción del camino de resistencia máxima entre t_k y t_j alcanza su resistencia estructural en la transición $t_m \in T_i \setminus \text{TI}$. Entonces, el resumen del camino de resistencia máxima de t_k a t_j se compone de la transición inmediata t_{im} (que es una réplica inmediata de t_m) y de los resúmenes de sus subcaminos que van de

t_k a t_m y de t_m a t_j . En principio, puede haber varios caminos de resistencia máxima entre transiciones de interfaz de la misma subred que alcancen su resistencia estructural en la misma transición interna t_m . En ese caso, no es necesario poner varias réplicas inmediatas de t_m . Es suficiente poner una, de ahí la notación conjuntista de la definición 4.33.

El conjunto H'_i es el de los lugares TT-MSIP necesarios para reducir los caminos de máxima resistencia entre transiciones de interfaz de la subred \mathcal{N}_i y que necesitan de más de un nodo para ser reducidos. Por los mismos motivos que en los conjuntos anteriores, sólo interesan caminos entre transiciones de interfaz de la misma subred. El conjunto H' es el conjunto unión de los H'_i . El cálculo de estos lugares es inmediato a partir de la salida del algoritmo 4.32 observando los valores de la matrices \mathbf{E} , \mathbf{D}_1 y \mathbf{D}_2 entre transiciones de interfaz de la subred \mathcal{N}_i . Si el camino de resistencia máxima entre t_k y t_j requería la inserción de una réplica inmediata de la transición t_m en el conjunto I_i ($\mathbf{E}(k, j) = m > 0$), ahora hay que insertar en el conjunto H'_i dos lugares TT-MSIP p_{kj}^1 y p_{kj}^2 correspondientes a los subcaminos en que la transición t_m divide el camino en cuestión. Para ello es necesario conocer las ganancias de t_k a t_m y de t_m a t_j (información contenida en el vector \mathbf{g} de ganancias) y los marcados ponderados de los mismos, información que se tiene calculada ya en las matrices \mathbf{D}_1 y \mathbf{D}_2 . En principio, puede haber varios caminos de resistencia máxima entre transiciones de interfaz de la misma subred que tengan alguno de sus lugares de resumen en común, es decir, en los conjuntos H_i pueden aparecer lugares repetidos (denotados de forma diferente pero con los mismos arcos de entrada y salida y el mismo marcado inicial). Con poner uno de ellos es suficiente.

Ahora se pueden construir los sistemas agregados.

Definición 4.34 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado y $B \subseteq P$ un K -corte de \mathcal{N} . El sistema extendido $\mathcal{ES} = \langle \mathcal{EN}, \mathbf{m}_0^{\mathcal{EN}} \rangle$ de $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ es el sistema resultante de añadir a $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ los nodos de los conjuntos H , I y H' con sus arcos de entrada y salida y el marcado inicial de la definición 4.33.

Teorema 4.35 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado, $B \subseteq P$ un K -corte de \mathcal{N} y \mathcal{ES} su sistema extendido. Entonces \mathcal{ES} es un WTS vivo y limitado que cumple:

1. $L(\mathcal{ES})|_T = L(\mathcal{S})$.
2. $R(\mathcal{ES})|_P = R(\mathcal{S})$.

Demostración:

La demostración se hará en dos pasos por medio de un sistema extendido intermedio \mathcal{ES}' . Sea \mathcal{ES}' el sistema que se forma a partir de \mathcal{ES} uniendo cada transición inmediata t_{im} del conjunto I (ver definición 4.33) con la transición interna t_m de \mathcal{N} de la que es réplica (por unir se entiende aquí sustituir las dos transiciones por una sola con todos los arcos de entrada y de salida de las transiciones iniciales). El sistema \mathcal{ES}' está formado a partir de \mathcal{S} añadiendo únicamente lugares TT-MSIP (teorema 4.17) con un marcado inicial que los hace implícitos (teorema 4.19) por lo que \mathcal{ES}' sigue siendo un WTS vivo y limitado que cumple $L(\mathcal{ES}') = L(\mathcal{S})$ y $R(\mathcal{ES}')|_P = R(\mathcal{S})$.

Ahora, para construir \mathcal{ES} a partir de \mathcal{ES}' simplemente hay que duplicar las transiciones internas con réplica inmediata en el conjunto I . Por lo tanto, es evidente que el \mathcal{ES} sigue siendo un WTS. Como \mathcal{S} es limitado, el marcado ponderado de cualquier camino en \mathcal{S} es finito en cualquier marcado alcanzable. Por construcción, los nuevos nodos \mathcal{ES} forman nuevos caminos en \mathcal{S} con el mismo marcado ponderado que otros ya existentes. Por el teorema 4.6, estos nuevos caminos tienen marcado ponderado finito en cualquier marcado alcanzable, luego \mathcal{ES} es acotado.

Ahora se demostrará que $L(\mathcal{ES})|_T = L(\mathcal{S})$, después que $R(\mathcal{ES})|_P = R(\mathcal{S})$ y por último que \mathcal{ES} es vivo.

$L(\mathcal{ES})|_T \subseteq L(\mathcal{S})$. Evidente porque \mathcal{S} puede construirse a partir de \mathcal{ES} eliminando restricciones a las transiciones del conjunto T .

$L(\mathcal{S}) \subseteq L(\mathcal{ES})|_T$. Por ser $L(\mathcal{ES}') = L(\mathcal{S})$, es suficiente demostrar que $L(\mathcal{ES}') \subseteq L(\mathcal{ES})|_T$. Es evidente que $\mathbf{m}_0^{\mathcal{EN}} = \mathbf{m}_0^{\mathcal{EN}'}$ ya que \mathcal{ES} y \mathcal{ES}' tienen el mismo conjunto de lugares. Sea $\sigma' \in L(\mathcal{ES}')$. Se construye la secuencia σ sustituyendo en σ' cada transición interna t_m con réplica inmediata t_{im} en el conjunto I por el par de transiciones (t_{im}, t_m) . Se demostrará, por inducción sobre el número de pares (t_{im}, t_m) , que σ es una secuencia de transiciones que se puede disparar en \mathcal{ES} . Si σ no tiene ningún par de la forma (t_{im}, t_m) , quiere decir que $\sigma = \sigma'$ y que es una secuencia de transiciones sin réplica en el conjunto I . Todas estas transiciones tienen exactamente las mismas precondiciones en \mathcal{ES} y en \mathcal{ES}' por lo que σ puede dispararse en \mathcal{ES} . Si el resultado es cierto para σ con n pares de la forma (t_{im}, t_m) , ver que también se cumple para σ con $n + 1$ pares. Sea σ_1 el prefijo de σ hasta el último par de la forma (t_{im}, t_m) (sin incluirlo) y σ'_1 el prefijo de σ' hasta la última transición t_m con réplica inmediata en I (sin incluirla). Si $\mathbf{m}_0^{\mathcal{EN}'} \xrightarrow{\sigma'_1} \mathbf{m}^{\mathcal{EN}'}$, entonces t_m está sensibilizada en $\langle \mathcal{EN}', \mathbf{m}^{\mathcal{EN}'} \rangle$. Por hipótesis de inducción,

σ_1 es disparable en \mathcal{ES} produciendo un marcado $\mathbf{m}^{\mathcal{EN}}$. Por construcción de \mathcal{ES} , \mathcal{ES}' y de las secuencias de disparo σ_1 y σ'_1 , como $\mathbf{m}_0^{\mathcal{EN}} = \mathbf{m}_0^{\mathcal{EN}'}$, entonces $\mathbf{m}^{\mathcal{EN}} = \mathbf{m}^{\mathcal{EN}'}$ y como t_m está sensibilizada en $\langle \mathcal{EN}', \mathbf{m}^{\mathcal{EN}'} \rangle$, entonces t_m y t_{im} están sensibilizadas en $\langle \mathcal{EN}, \mathbf{m}^{\mathcal{EN}} \rangle$ (ya que las precondiciones de t_m en \mathcal{ES}' son la unión de las precondiciones de t_m y t_{im} en \mathcal{ES}). Por lo tanto, el par (t_{im}, t_m) es disparable en $\langle \mathcal{EN}, \mathbf{m}^{\mathcal{EN}} \rangle$. Sea ahora \mathbf{m}_2 el marcado que se produce al disparar en $\langle \mathcal{EN}, \mathbf{m}^{\mathcal{EN}} \rangle$ el par (t_{im}, t_m) y $\mathbf{m}^{\mathcal{EN}'} \xrightarrow{t_m} \mathbf{m}_2^{\mathcal{EN}'}$. Nuevamente $\mathbf{m}_2^{\mathcal{EN}} = \mathbf{m}_2^{\mathcal{EN}'}$, por lo que el resto σ_2 de la secuencia σ es disparable en $\langle \mathcal{EN}, \mathbf{m}_2^{\mathcal{EN}} \rangle$ por ser igual al resto σ'_2 de la secuencia σ' y ser σ'_2 disparable en $\langle \mathcal{EN}', \mathbf{m}_2^{\mathcal{EN}'} \rangle$. Se ha demostrado que para cada $\sigma' \in L(\mathcal{ES}')$ existe una secuencia σ de transiciones de \mathcal{ES} que es disparable en \mathcal{ES} . Como las transiciones del conjunto I son inmediatas no se puede asegurar que $\sigma \in L(\mathcal{ES})$ ya que, en caso de conflicto las transiciones de I se deben disparar antes que las de T . Reordenando las transiciones inmediatas de σ y manteniendo el orden del resto de forma que se disparen en el primer momento en el que estén sensibilizadas (primero se reordena la transición inmediata del primer par, luego la del segundo, etc.), se obtiene una nueva secuencia φ que sigue siendo disparable en \mathcal{ES} y cumple que $\varphi \in L(\mathcal{ES})$ por construcción. Además, $\varphi|_T = \sigma|_T = \sigma'$. Por lo tanto, $L(\mathcal{S}) = L(\mathcal{ES}') \subseteq L(\mathcal{ES})|_T$.

$R(\mathcal{ES})|_P = R(\mathcal{S})$. $\mathbf{m} \in R(\mathcal{S})$ si y sólo si existe $\sigma \in L(\mathcal{S})$ tal que $\mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$. Por ser $L(\mathcal{ES})|_T = L(\mathcal{S})$, esto ocurre si y sólo si existe $\varphi \in L(\mathcal{ES})$ tal que $\varphi|_T = \sigma$. Sea $\mathbf{m}_0^{\mathcal{EN}} \xrightarrow{\varphi} \mathbf{m}^{\mathcal{EN}}$. Como $\varphi|_T = \sigma$, entonces $\mathbf{m}^{\mathcal{EN}}|_P = \mathbf{m}$. Luego $\mathbf{m} \in R(\mathcal{S})$ si y sólo si existe $\mathbf{m}^{\mathcal{EN}} \in R(\mathcal{ES})$ tal que $\mathbf{m}^{\mathcal{EN}}|_P = \mathbf{m}$.

\mathcal{ES} es vivo. Se ha demostrado que las transiciones de interfaz y las internas sin réplica en I tienen el mismo comportamiento (en cuanto a las secuencias de disparo) en \mathcal{ES} y \mathcal{ES}' por lo que son vivas en \mathcal{ES} , por ser \mathcal{ES}' vivo. Y los pares de la forma (t_{im}, t_m) se comportan en \mathcal{ES} como t_m en \mathcal{ES}' por lo que también son vivas. Por lo tanto, \mathcal{ES} es vivo. \diamond

Definición 4.36 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle P, T, F, W \rangle$ un WTS vivo y limitado, $B \subseteq P$ un K -corte de \mathcal{N} y \mathcal{ES} su sistema extendido.

- i) El sistema de bajo nivel $\mathcal{LS}_i = \langle \mathcal{LN}_i, \mathbf{m}_0^i \rangle$ con $1 \leq i \leq K$ es el sistema que se obtiene eliminando de \mathcal{ES} los lugares $\bigcup_{j \neq i} P_j \setminus B$ y las transiciones $\bigcup_{j \neq i} T_j \setminus TI$ y sus arcos adyacentes.
- ii) El sistema de alto nivel (o esqueleto básico) $\mathcal{BS} = \langle \mathcal{BN}, \mathbf{m}_0^{\mathcal{BN}} \rangle$ es el sistema que se obtiene eliminando de \mathcal{ES} los lugares $\bigcup_{i=1}^K P_i \setminus B$ y las transiciones $\bigcup_{i=1}^K T_i \setminus TI$ y sus arcos adyacentes.

En cada \mathcal{LS}_i , todas las subredes \mathcal{N}_j con $j \neq i$ se reducen a sus transiciones de interfaz y a los nodos de los conjuntos H_j , I_j y H'_j (ver definición 4.33), mientras que la subred \mathcal{N}_i se preserva completamente. En el \mathcal{BS} se reducen todas las subredes.

El siguiente teorema demuestra las propiedades que tienen los sistemas agregados recién construidos.

Teorema 4.37 *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un WTS vivo y limitado, $B \subseteq P$ un K -corte de \mathcal{N} . Entonces, \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} son WTS's vivos y limitados.*

Demostración:

Por el teorema 4.35, \mathcal{ES} es un WTS vivo y limitado. Para formar a partir de \mathcal{ES} los sistemas agregados se eliminan nodos con todos sus arcos de entrada y salida. Por lo tanto, los lugares que quedan deben tener un único arco de entrada y salida, es decir, los sistemas agregados son WTS's. En [TCWCS92] se demuestra que un WTS es vivo y limitado si y sólo si todos sus ciclos tienen ganancia 1 y tienen suficientes marcas para hacerlos vivos. Como la construcción de los sistemas agregados preserva la ganancia de cualquier camino y \mathcal{ES} es vivo y limitado, todos los ciclos de los sistemas agregados tienen ganancia 1. Además, todo ciclo de un sistema agregado es también ciclo en \mathcal{ES} . Como \mathcal{ES} es vivo, todos sus ciclos tienen marcas suficientes para ser vivos, por lo que todos los ciclos de los sistemas agregados tienen marcas suficientes para ser vivos. \diamond

A modo de resumen, en esta sección se ha desarrollado una técnica estructural eficiente que permite descomponer un WTS vivo y limitado en un número finito arbitrario K de subredes y a partir de ellas construir K sistemas de bajo nivel (en los que se mantiene una de las subredes y se reducen el resto) y un sistema de alto nivel o esqueleto básico (en el que se reducen todas las subredes). Estos $K + 1$ sistemas agregados son también WTS's vivos y limitados (por lo tanto reversibles, ver sección 4.1), por lo que sus cadenas de Markov isomorfas son ergódicas y poseen distribución en estado estacionario única (teorema 2.85). Además, los sistemas agregados tienen la estructura más simple posible, en el sentido de que no se han empleado lugares implícitos redundantes en la reducción de las subredes producto del corte. Esta reducción de la estructura de los sistemas agregados se ha conseguido sin incrementar la complejidad de los algoritmos de cálculo.

En la siguiente sección se emplearán estos sistemas agregados para aproximar el throughput de las transiciones del WTS original.

4.4 Aproximación iterativa del throughput

El contenido de esta sección es similar al de la sección 3.4. Se incluye aquí con las adaptaciones necesarias para WTS's con objeto de aumentar la legibilidad de la técnica completa de aproximación.

En la sección anterior se ha desarrollado una técnica para descomponer un WTS vivo y limitado en K subredes, a partir de las cuales se construyen K sistemas de bajo nivel y uno de alto nivel o esqueleto básico. Cada sistema de bajo nivel contiene una de las subredes producto del corte, los lugares del corte, las transiciones de interfaz y un conjunto de nodos (lugares implícitos y transiciones inmediatas) que resumen el comportamiento del resto de subredes. El sistema de alto nivel contiene los lugares del corte, las transiciones de interfaz y los nodos que resumen el comportamiento de todas las subredes. Ahora se van a emplear estos $K + 1$ sistemas agregados para aproximar el throughput de las transiciones del WTS original.

Si se observa el teorema 4.9, en WTS's vivos y limitados los ratios de visita o los throughput relativos de las transiciones (ver definición 2.66) están determinados por estructura de la red. En concreto, el teorema 4.9 establece que el vector \mathbf{g} de ganancias del algoritmo 4.23 es a su vez el vector de ratios de visita de las transiciones del WTS. Así, si la ganancia entre t_i y t_j es g_{ij} entonces se cumple que $\chi(t_j) = \chi(t_i)g_{ij}$. Por lo tanto, basta con aproximar el throughput de una transición cualquiera del WTS original, porque por medio de las ganancias se pueden obtener aproximaciones para el throughput de cualquier otra transición con el mismo porcentaje de error. Además, como los sistemas agregados generados en la sección anterior preservan la ganancia entre transiciones, los vectores de ratios de visita de cualquier sistema agregado coinciden con el del sistema original en las transiciones que se preserven.

La técnica que se va a exponer en esta sección es básicamente la misma que la del capítulo 3 para MG's. Por ello, en esta sección se expondrán las cuestiones referentes a la adaptación de la técnica para WTS's y se añadirán algunos resultados parciales relativos a la convergencia del método. Esta técnica es un método de aproximación del tiempo de respuesta. Las transiciones de interfaz de cada subred reducida en cada sistema agregado aproximarán el tiempo de respuesta de toda la subred. La técnica es iterativa y utiliza un sistema de alto nivel o esqueleto básico por las mismas razones que en el caso de MG's, para no tener problemas de convergencia y de malas aproximaciones. El algoritmo utilizado es el siguiente:

Algoritmo 4.38

input: $\langle \mathcal{S}, w \rangle$ WTS estocástico vivo y limitado
seleccionar un K -corte $B \subseteq P$ de \mathcal{N}
construir \mathcal{LS}_i y \mathcal{BS} para $1 \leq i \leq K$
seleccionar tasas iniciales $\mu_t^{(0)}$ para cada $t \in T_j \cap \text{TI}$ con $2 \leq j \leq K$
 $n := 0$ (contador de iteraciones)
repeat
 $n := n + 1$
 for $i := 1$ **to** K **do**
 solve \mathcal{LS}_i **con:**
 In: tasas $\mu_l^{(n)}$ para las transiciones de $T_l \cap \text{TI}$ con $1 \leq l < i$
 tasas $\mu_l^{(n-1)}$ para las transiciones de $T_l \cap \text{TI}$ con $i < l \leq K$
 Out: proporciones μ_i entre tasas de las transiciones de $T_i \cap \text{TI}$ y
 throughput $\chi_i^{(n)}$ de las transiciones de $T_i \cap \text{TI}$
 solve \mathcal{BS} (encontrar factor de escala λ) **con:**
 In: tasas $\mu_l^{(n)}$ para las transiciones de $T_l \cap \text{TI}$ con $1 \leq l < i$
 tasas $\mu_l^{(n-1)}$ para las transiciones de $T_l \cap \text{TI}$ con $i < l \leq K$
 proporciones entre tasas μ_i para las transiciones de $T_i \cap \text{TI}$
 throughput $\chi_i^{(n)}$ de las transiciones de $T_i \cap \text{TI}$
 Out: tasas reales $\mu_i^{(n)} = \lambda \cdot \mu_i$ de las transiciones de $T_i \cap \text{TI}$
 tales que $\chi_{\mathcal{BS}}^{(n)} = \chi_i^{(n)}$
 end for
until convergencia de $\{\chi_i^{(n)}\}_{i=1}^K$
output: throughput $\{\chi_i^{(n)}\}_{i=1}^K$ de las transiciones de $\langle \mathcal{S}, w \rangle$

En este algoritmo iterativo se van resolviendo uno por uno todos los sistemas de bajo nivel \mathcal{LS}_i (con $1 \leq i \leq K$). Cuando en la iteración n del esquema iterativo se resuelve el \mathcal{LS}_i , se obtienen para las transiciones de $T_i \cap \text{TI}$ los vectores $\chi_i^{(n)}$ de throughput y μ_i de proporciones entre tasas de servicio. Con esta información se busca, por medio del \mathcal{BS} , un factor de escala λ para las proporciones entre las tasas de disparo, de forma que los throughput de las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i coincidan. El cálculo de este factor de escala puede realizarse por medio de una búsqueda lineal en \mathcal{BS} . Se va resolviendo el \mathcal{BS} cambiando todas las tasas de las transiciones de $T_i \cap \text{TI}$ de forma que se mantengan las proporciones μ_i hasta que se consigan los mismos throughput en las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i .

Notar que \mathcal{BS} tiene muchos menos estados que el sistema original e incluso que los \mathcal{LS}_i , por lo que el coste de esta búsqueda lineal es despreciable comparado con la complejidad del estudio de los otros sistemas.

Después de este doble paso se obtiene una nueva aproximación $\mu_i^{(n)}$ para las tasas de disparo de las transiciones de $T_i \cap \text{TI}$ que se mantienen en el estudio posterior de otros \mathcal{LS}_j y en \mathcal{BS} .

Quedan por aclarar algunos puntos de este algoritmo iterativo. Por un lado, cuestiones respecto a las restricciones del corte. Por otro lado, la selección de las tasas de disparo iniciales $\mu_j^{(0)}$ de las transiciones de $T_j \cap \text{TI}$ con $2 \leq j \leq K$, que son necesarias para resolver por primera vez el sistema \mathcal{LS}_1 e iniciar el algoritmo iterativo. Otra cuestión a tener en cuenta es la forma de calcular, al resolver el \mathcal{LS}_i , las proporciones μ_i entre las tasas de disparo de las transiciones de $T_i \cap \text{TI}$. Por último, consideraciones acerca de la búsqueda lineal del factor de escala λ en \mathcal{BS} para el cálculo de las tasas de disparo reales de las transiciones de interfaz en cada iteración y el test de convergencia para decidir el final del algoritmo iterativo.

Respecto a las restricciones sobre el corte, en la sección anterior se podía definir sobre cualquier conjunto de lugares que induzca una partición en el conjunto de transiciones y de lugares (excluidos los del corte) del WTS original. Si se tiene en cuenta ahora las cuestiones numéricas, no es deseable que aparezca ninguna transición de interfaz que sea inmediata. Si éste es el caso, podría haber errores de ejecución al poder obtener como medida de utilización de alguna transición de interfaz 0 y por lo tanto divisiones por 0 al calcular su tasa relativa de disparo.

Respecto a las restricciones sobre el corte, en la sección anterior se podía definir sobre cualquier conjunto de lugares que induzca una partición en el conjunto de transiciones y de lugares (excluidos los del corte) del WTS original. Si se tienen en cuenta ahora las cuestiones numéricas, no es deseable que aparezca ninguna transición de interfaz que sea inmediata. En este caso podrían aparecer errores de división por cero al estimar las proporciones entre las tasas de disparo de las transiciones de interfaz como se verá en el párrafo dedicado a la estimación de estas proporciones.

Respecto a la selección de las tasas iniciales para las transiciones de interfaz, el algoritmo iterativo no se ve afectado, en lo que respecta a las aproximaciones de los throughput obtenidas, por la selección de estos valores. Por lo tanto, en principio se pueden seleccionar valores arbitrarios. Pero es razonable poner unos valores iniciales que tengan relación con el modelo.

Una selección adecuada consiste en poner a cada transición de interfaz la tasa que tiene en el modelo original. De esta forma se reduce el número de iteraciones necesarias para alcanzar la convergencia del algoritmo. Esta elección dará sistemas agregados más rápidos en la primera iteración del método pero se corrige en las siguientes.

Por lo que respecta a la estimación de las proporciones μ_i entre las tasas de disparo de las transiciones de interfaz de la subred \mathcal{N}_i , se utilizan los nodos calculados en la fase de descomposición estructural. Como se explicó en la sección anterior, para cada subred \mathcal{N}_i producto del corte del WTS original se calculaba un conjunto $H_i \cup I_i \cup H'_i$ de nodos que resumen su comportamiento interno. Son estos nodos los que se emplean para realizar la estimación. El sistema de bajo nivel \mathcal{LS}_i contiene completamente la subred \mathcal{N}_i y los nodos del conjunto $H_i \cup I_i \cup H'_i$. Por lo tanto, al resolver su CTMC isomorfa, se puede calcular la utilización de cada transición de interfaz si sólo se tienen en cuenta los lugares del conjunto $H_i \cup H'_i \cup B$ (es decir, se calcula la probabilidad de que la transición esté sensibilizada una vez se eliminan los nodos internos de la subred \mathcal{N}_i). De esta manera, se está relacionando el funcionamiento completo de la subred \mathcal{N}_i con el de su resumen para poder exportar esta información al resto de sistemas de bajo nivel. Posteriormente, basta emplear la fórmula del cálculo del throughput del teorema 2.87 para encontrar las proporciones entre las tasas de disparo de las transiciones de interfaz. En la iteración n del algoritmo, la tasa proporcional de disparo de la transición $t \in T_i \cap \text{TI}$ es:

$$\mu_i[t] = \frac{\chi_i^{(n)}[t]}{P\{\mathbf{m}[p] \geq W(p, t) \quad \forall p \in \bullet t \cap (H_i \cup H'_i \cup B)\}}$$

Esta expresión es válida si se emplea semántica de un solo servidor. Si se emplea semántica de infinitos servidores, hay que sustituir la utilización de las transiciones por su grado medio de sensibilización, es decir, la tasa proporcional de disparo es en este caso:

$$\mu_i[t] = \frac{\chi_i^{(n)}[t]}{E[\max\{k \mid \mathbf{m}[p] \geq kW(p, t) \quad \forall p \in \bullet t \cap (H_i \cup H'_i \cup B)\}]}$$

En cuanto al algoritmo de búsqueda lineal del factor de escala en \mathcal{BS} para calcular las tasas reales de disparo de las transiciones de interfaz, hay que tener en cuenta lo siguiente. Se quiere conseguir que coincidan los

throughput de las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i , es decir, que coincidan los throughput de las transiciones de interfaz. A diferencia de MG's, en WTS's los throughput de las transiciones pueden ser diferentes. Por lo tanto, no se puede hablar de throughput de un WTS sino de throughput de una transición de un WTS. Entonces, conseguir que los throughput de todas las transiciones de interfaz coincidan en los dos sistemas podría suponer un problema. Como en WTS's las ratios de visita o throughput relativos de las transiciones vienen determinados por la ganancia entre transiciones (teorema 4.9) y los sistemas agregados construidos preservan estas ganancias, entonces desaparece el problema y basta que el throughput de una transición t de interfaz cualquiera coincida en \mathcal{BS} y \mathcal{LS}_i para poder asegurar que coinciden todos los throughput de todas las transiciones de interfaz. A lo largo del algoritmo iterativo se va modificando el valor de las tasas de las transiciones de interfaz de los \mathcal{LS}_i y de \mathcal{BS} . El throughput de las transiciones de estos sistemas es una función racional (cociente de polinomios) con respecto a estas variables. Como se han calculado unas proporciones entre las tasas de disparo que se quieren mantener en \mathcal{BS} , si se quiere calcular un factor de escala λ de forma que coincidan los throughput de las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i , lo que se está haciendo en el fondo es plantear una ecuación racional de la forma:

$$\chi_{\mathcal{BS}}^{(n)}[t](\lambda) = \chi_{\mathcal{LS}_i}^{(n)}[t]$$

En la ecuación anterior, t es una transición de interfaz cualquiera que se fija como referencia en \mathcal{BS} y \mathcal{LS}_i . Entonces, $\chi_{\mathcal{LS}_i}^{(n)}[t]$ es el throughput en \mathcal{LS}_i de la transición t calculado justo antes de empezar el estudio en \mathcal{BS} . Por su parte, $\chi_{\mathcal{BS}}^{(n)}[t](\lambda)$ es una función de λ , que devuelve el throughput en \mathcal{BS} de la misma transición t de referencia. Esta función es racional en λ y no se conoce de forma explícita. Calculando una solución λ de la ecuación anterior, se puede asegurar que el throughput de todas las transiciones de interfaz coincide en \mathcal{BS} y \mathcal{LS}_i . El problema numérico de resolución de esta ecuación puede resolverse con algoritmos como el de la secante.

Por último, queda por aclarar el test de convergencia. Este algoritmo numérico calcula unas aproximaciones del throughput aceptables para una técnica de aproximación (por debajo del 5%). Teniendo en cuenta la aproximación que se va a obtener al final, no es necesario que el test de convergencia sea muy exigente ya que esto provocaría el cálculo de un mayor número de iteraciones sin que por ello se aumente la precisión de los resultados obteni-

dos. Si se pone un test de convergencia poco exigente, entonces el algoritmo finalizará antes de lo debido, aumentando en general el error de la aproximación. El test de convergencia que se utiliza consiste en iterar hasta que la diferencia entre los throughput de las transiciones de interfaz de todos los \mathcal{LS}_i en dos iteraciones consecutivas del método sea menor que el 0.1%. Con este test el algoritmo converge (en todas las pruebas realizadas) en muy pocas iteraciones (entre 3 y 5 iteraciones).

Una vez explicado el método iterativo que se emplea para calcular aproximaciones del throughput de las transiciones del sistema original, se van a demostrar algunas de sus características. La demostración formal de estas características se basa en algunos resultados de otros autores que se van a exponer ahora.

El primer resultado (ver [MT95]) establece, para cualquier SPN, la continuidad del vector π de probabilidades en estado estacionario con respecto a las entradas del generador infinitesimal \mathbf{Q} de la CTMC isomorfa.

Teorema 4.39 [MT95] *Sea $\langle \mathcal{S}, w \rangle$ una SPN, \mathbf{Q} el generador infinitesimal de su CTMC isomorfa y π el vector de probabilidades en estado estacionario. Los elementos de π son funciones continuas de las entradas de \mathbf{Q} .*

Como las entradas de \mathbf{Q} son funciones continuas de las tasas de disparo de las transiciones de la red, en particular, π es continuo con respecto a las tasas de las transiciones de la red.

Corolario 4.40 *Sea $\langle \mathcal{S}, w \rangle$ una SPN y π el vector de probabilidades en estado estacionario. Los elementos de π son funciones continuas de las tasas de disparo de las transiciones de \mathcal{N} .*

Otra consecuencia inmediata del teorema anterior es la continuidad respecto de las entradas de \mathbf{Q} de todos los los índices de prestaciones de $\langle \mathcal{S}, w \rangle$ que puedan definirse como funciones continuas de las entradas del vector π , en particular el throughput de las transiciones.

Corolario 4.41 *Sea $\langle \mathcal{S}, w \rangle$ una SPN, \mathbf{Q} el generador infinitesimal de su CTMC isomorfa y π el vector de probabilidades en estado estacionario. Sea X un índice de prestaciones de la SPN que puede expresarse como $X = f(\pi)$ con f continua. Entonces X es función continua de las entradas de \mathbf{Q} y de las tasas de las transiciones de $\langle \mathcal{S}, w \rangle$. En particular, el throughput de cualquier transición de $\langle \mathcal{S}, w \rangle$ es una función continua de las entradas de \mathbf{Q} y de las tasas de las transiciones de $\langle \mathcal{S}, w \rangle$.*

Otros resultados que se necesitan para demostrar las propiedades del método iterativo son las referentes a monotonía de throughput en la clase de los WTS's. En concreto, se tiene monotonía de throughput respecto a las tasas de disparo de las transiciones y respecto a la topología de las redes. Estos resultados pueden verse en [BLS96].

Teorema 4.42 [BLS96] Monotonía respecto a las tasas de disparo.
 Sea $\langle \mathcal{S}, w \rangle$ un WTS estocástico, vivo y limitado y $t \in T$ una transición de \mathcal{N} de tasa λ variable. El throughput de t en $\langle \mathcal{S}, w \rangle$ en función de λ , $\chi_{\mathcal{S}}[t](\lambda)$ es una función monótona creciente, es decir, si $\lambda' \leq \lambda \Rightarrow \chi_{\mathcal{S}}[t](\lambda') \leq \chi_{\mathcal{S}}[t](\lambda)$.

Teorema 4.43 [BLS96] Monotonía respecto a la topología.
 Sean $\langle \mathcal{S}, w \rangle$ y $\langle \mathcal{S}', w' \rangle$ dos WTS's estocásticos, vivos y limitados tales que $P \subseteq P'$, $T \subseteq T'$, $F \subseteq F'$ y $W = W'|_F$. Suponer además que $\langle \mathcal{S}, w \rangle$ y $\langle \mathcal{S}', w' \rangle$ tienen el mismo marcado inicial y tasas de disparo en los lugares y transiciones comunes a las dos redes. En estas condiciones se tiene que $\chi_{\mathcal{S}}[t] \geq \chi_{\mathcal{S}'}[t]$ para cualquier $t \in T$.

Por último, se necesita un resultado sobre el cálculo de una cota superior para el throughput de las transiciones de un WTS. El siguiente resultado es mucho más general.

Propiedad 4.44 [CS93] Cota superior de throughput.
 Sea $\langle \mathcal{S}, w \rangle$ una SPN viva y limitada. Una cota inferior para el tiempo medio entre disparos $\Gamma^{(i)}$ de la transición t_i es el resultado del siguiente problema de programación lineal:

$$\begin{aligned} \Gamma^{(i)} \geq & \max \quad \mathbf{Y}^T \cdot \mathbf{Pre} \cdot \mathbf{D}^{(i)} \\ \text{sujeto a} & \quad \mathbf{Y}^T \cdot \mathbf{C} = \mathbf{0} \\ & \quad \mathbf{Y}^T \cdot \mathbf{m}_0 = 1 \\ & \quad \mathbf{Y} \geq \mathbf{0} \end{aligned}$$

Donde \mathbf{Pre} es la matriz de preincidencia de \mathcal{N} , \mathbf{C} es la matriz de incidencia de \mathcal{N} y $\mathbf{D}^{(i)}$ es el vector de demandas medias de servicio de las transiciones ($\mathbf{D}^{(i)}(t_j) = s_j v^{(i)}(t_j)$ con s_j el tiempo medio de servicio de la transición t_j y $v^{(i)}(t_j)$ la ratio de visita de la transición t_j normalizada para t_i , es decir, $v^{(i)}(t_i) = 1$).

A partir de una cota inferior del tiempo medio entre disparos de una transición se obtiene, calculando el inverso, una cota superior del throughput de la misma.

Para demostrar las propiedades del método iterativo conviene describir el algoritmo 4.38 por medio de ecuaciones y funciones. La forma más sencilla de describir el algoritmo es fijando como variables las tasas de las transiciones de interfaz.

Notación 4.45 Sea $\langle \mathcal{S}, w \rangle$ un WTS estocástico, vivo y limitado y $B \subseteq P$ un K -corte de \mathcal{N} . Se utilizarán las siguientes notaciones:

- i) $n_i = |T_i \cap \text{TI}|$ con $1 \leq i \leq K$ es el número de transiciones de interfaz de la subred \mathcal{N}_i .
- ii) $M = \sum_{i=1}^K n_i$ es el número de transiciones de interfaz de \mathcal{N} .
- iii) I_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ es la transición de interfaz j de la subred \mathcal{N}_i .
- iv) λ_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ es la tasa de disparo de I_{ij} .
- v) $\boldsymbol{\lambda}_i = (\lambda_{i1}, \dots, \lambda_{in_i})$ con $1 \leq i \leq K$ es el vector de tasas de disparo de las transiciones de interfaz de \mathcal{N}_i .
- vi) $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_K)$ es el vector de tasas de disparo de todas las transiciones de interfaz.
- vii) χ_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ es el throughput de la transición I_{ij} .
- viii) $\boldsymbol{\chi}_i = (\chi_{i1}, \dots, \chi_{in_i})$ con $1 \leq i \leq K$ es el vector de throughput de las transiciones de interfaz de \mathcal{N}_i .
- ix) μ_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ es la tasa proporcional de disparo de transición I_{ij} calculada en \mathcal{LS}_i .
- x) $\boldsymbol{\mu}_i = (\mu_{i1}, \dots, \mu_{in_i})$ con $1 \leq i \leq K$ es el vector de tasas proporcionales de disparo de las transiciones de interfaz de \mathcal{N}_i .
- xi) λ'_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ es la nueva tasa de disparo de la transición I_{ij} calculada en \mathcal{BS} .
- xii) $\boldsymbol{\lambda}'_i = (\lambda'_{i1}, \dots, \lambda'_{in_i})$ con $1 \leq i \leq K$ es el vector de nuevas tasas de disparo de las transiciones de $T_i \cap \text{TI}$ calculadas en \mathcal{BS} .
- xiii) $\boldsymbol{\lambda}^{(i)} = (\boldsymbol{\lambda}'_1, \dots, \boldsymbol{\lambda}'_i, \boldsymbol{\lambda}_{i+1}, \dots, \boldsymbol{\lambda}_K)$ con $1 \leq i \leq K$ es el vector intermedio de tasas de disparo de las transiciones de TI después del estudio de \mathcal{LS}_i y el correspondiente estudio de \mathcal{BS} .

Se va a dar una explicación de estas notaciones. Cada vez que se resuelve \mathcal{LS}_i con $1 \leq i \leq K$ se calcula, para cada transición I_{ij} con $1 \leq j \leq n_i$, su throughput χ_{ij} y su tasa relativa μ_{ij} . Agrupando en vectores los índices de prestaciones calculados en \mathcal{LS}_i para todas las transiciones de $T_i \cap \text{TI}$, se obtienen $\boldsymbol{\chi}_i$ y $\boldsymbol{\mu}_i$. El siguiente paso consiste en estudiar el \mathcal{BS} , para obtener las nuevas tasas de disparo λ'_{ij} de las transiciones de $T_i \cap \text{TI}$. En notación vectorial, se obtiene el vector $\boldsymbol{\lambda}'_i$. El siguiente paso consiste en estudiar el \mathcal{LS}_{i+1} . El vector de tasas de transiciones de interfaz que se emplea para el estudio de este sistema no es el vector $\boldsymbol{\lambda}$ porque ya han sido actualizadas las tasas de las transiciones de $T_j \cap \text{TI}$ con $j \leq i$. El vector de tasas es el $\boldsymbol{\lambda}^{(i)}$.

Ahora se van a dar las notaciones de las funciones que intervienen en el algoritmo iterativo.

Notación 4.46 Sea $\langle \mathcal{S}, w \rangle$ un WTS estocástico, vivo y limitado y $B \subseteq P$ un K -corte de \mathcal{N} . Se denotan a las siguientes funciones:

- i) $p_i : (\mathbb{R}^+)^M \longrightarrow (\mathbb{R}^+)^{n_i}$ con $1 \leq i \leq K$ tal que $p_i(\boldsymbol{\lambda}) = \boldsymbol{\lambda}_i$.
- ii) $g_i : (\mathbb{R}^+)^M \longrightarrow (\mathbb{R}^+)^{n_i}$ con $1 \leq i \leq K$ tal que $g_i(\boldsymbol{\lambda}) = \boldsymbol{\lambda}'_i$.
- iii) $G : (\mathbb{R}^+)^M \longrightarrow (\mathbb{R}^+)^M$ tal que $G = (g_1, \dots, g_K)$.
- iv) $\chi_{ij} : (\mathbb{R}^+)^M \longrightarrow \mathbb{R}^+$ con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ tal que $\chi_{ij}(\boldsymbol{\lambda})$ es el throughput en \mathcal{LS}_i de la transición I_{ij} si se ponen como tasas de disparo de las transiciones de $T_k \cap \text{TI}$ con $k \neq i$ las que marcan los vectores $\boldsymbol{\lambda}_k$ de $\boldsymbol{\lambda}$.
- v) $\mu_{ij} : (\mathbb{R}^+)^M \longrightarrow \mathbb{R}^+$ con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ tal que $\mu_{ij}(\boldsymbol{\lambda})$ es la tasa proporcional de disparo en \mathcal{LS}_i de la transición I_{ij} si se ponen como tasas de disparo de las transiciones de $T_k \cap \text{TI}$ con $k \neq i$ las que marcan los vectores $\boldsymbol{\lambda}_k$ de $\boldsymbol{\lambda}$.
- vi) $\mu_i : (\mathbb{R}^+)^M \longrightarrow (\mathbb{R}^+)^{n_i}$ con $1 \leq i \leq K$ tal que $\mu_i = (\mu_{i1}, \dots, \mu_{in_i})$.
- vii) $\chi_{\mathcal{BS}}[t] : (\mathbb{R}^+)^M \setminus \{0\} \longrightarrow \mathbb{R}^+$ tal que $\chi_{\mathcal{BS}}(\boldsymbol{\lambda})$ es el throughput de la transición t de \mathcal{BS} si se ponen como tasas de disparo de las transiciones de $T_i \cap \text{TI}$ las que marcan los vectores $\boldsymbol{\lambda}_i$ de $\boldsymbol{\lambda}$.
- viii) $\chi_{\mathcal{BS}_i}[t][\boldsymbol{\lambda}_1, \dots, \boldsymbol{\mu}_i, \dots, \boldsymbol{\lambda}_K] : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ con $1 \leq i \leq K$ tales que:
 $\chi_{\mathcal{BS}_i}[t][\boldsymbol{\lambda}_1, \dots, \boldsymbol{\mu}_i, \dots, \boldsymbol{\lambda}_K](\lambda) = \chi_{\mathcal{BS}}[t](\boldsymbol{\lambda}_1, \dots, \lambda \boldsymbol{\mu}_i, \dots, \boldsymbol{\lambda}_K)$.

Se va a dar una explicación de estas notaciones. La función p_i con $1 \leq i \leq K$ extrae el vector λ_i de tasas de las transiciones de $T_i \cap \text{TI}$ a partir del vector λ de tasas de transiciones de interfaz (es una función proyección). La función g_i con $1 \leq i \leq K$ calcula las nuevas tasas λ'_i de las transiciones de $T_i \cap \text{TI}$ en una iteración parcial (hasta el estudio del \mathcal{LS}_i) del algoritmo 4.38, si se toma λ como vector de tasas iniciales para las transiciones de interfaz. La función G calcula las nuevas tasas λ' de las transiciones de interfaz tras una iteración completa del algoritmo 4.38, si se toma λ como vector de tasas iniciales para las transiciones de interfaz. Las funciones χ_{ij} y μ_{ij} con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ calculan el throughput y la tasa proporcional de disparo de la transición de interfaz I_{ij} en \mathcal{LS}_i , si se toma λ_k como vector de tasas para las transiciones de $T_k \cap \text{TI}$ para todo $k \neq i$. La función μ_i con $1 \leq i \leq K$ calcula el vector μ_i de tasas proporcionales de disparo de las transiciones de $T_i \cap \text{TI}$ que se obtiene en \mathcal{LS}_i , si se toma λ_j como vector de tasas para las transiciones de $T_j \cap \text{TI}$ para todo $j \neq i$. La función $\chi_{\mathcal{BS}}[t]$ calcula el throughput de la transición t de \mathcal{BS} , si se toma λ como vector de tasas para sus transiciones. La función $\chi_{\mathcal{BS}_i}[t][\lambda_1, \dots, \mu_i, \dots, \lambda_K]$ con $1 \leq i \leq K$ calcula el throughput de la transición t de \mathcal{BS} , si se ponen λ_j con $j \neq i$ como vectores de tasas de disparo de las transiciones de $T_j \cap \text{TI}$ y $\lambda \mu_i$ para las transiciones de $T_i \cap \text{TI}$ (λ es el parámetro de la función).

El siguiente resultado expone las relaciones que hay entre las funciones y vectores de las notaciones anteriores.

Proposición 4.47 *Sea $\langle \mathcal{S}, w \rangle$ un WTS estocástico, vivo y limitado y $B \subseteq P$ un K -corte de \mathcal{N} . Se cumplen las siguientes relaciones:*

1. $g_1(\lambda) = \lambda'_1 = (\chi_{\mathcal{BS}_1}^{-1}[I_{11}])(\chi_{11}(\lambda)) \cdot \mu_1(\lambda)$.
2. $\lambda^{(i)} = (g_1(\lambda), \dots, g_i(\lambda), p_{i+1}(\lambda), \dots, p_K(\lambda))$ para $1 \leq i < K$.
3. $g_i(\lambda) = \lambda'_i = (\chi_{\mathcal{BS}_i}^{-1}[I_{i1}])(\chi_{i1}(\lambda^{(i-1)})) \cdot \mu_i(\lambda^{(i-1)})$ para $1 < i \leq K$.

Demostración:

Evidente a partir del algoritmo 4.38 y de las notaciones anteriores. Se han tomado como transiciones de referencia en cada \mathcal{LS}_i la primera transición de interfaz I_{i1} . Sólo falta demostrar que existen las funciones inversas de $\chi_{\mathcal{BS}_i}$, que se hará en la proposición 4.50. \diamond

Con las notaciones y relaciones anteriores se puede describir el algoritmo iterativo de la siguiente manera.

Notación 4.48 Sea $\langle \mathcal{S}, w \rangle$ un WTS estocástico, vivo y limitado y $B \subseteq P$ un K -corte de \mathcal{N} . Sean $\lambda_{ij}^{(0)}$ con $1 \leq i \leq K$ y $1 \leq j \leq n_i$ las tasas de disparo de las transiciones I_{ij} en $\langle \mathcal{S}, w \rangle$, $\lambda_i^{(0)} = (\lambda_{i1}, \dots, \lambda_{in_i})$ y $\lambda^{(0)} = (\lambda_1^{(0)}, \dots, \lambda_K^{(0)})$. El algoritmo iterativo 4.38 puede escribirse de la siguiente manera:

Algoritmo 4.49

input: $\lambda^{(0)}$ tasas iniciales de las transiciones de interfaz.

$n := 0$ (contador de iteraciones)

repeat

$n := n + 1$

$\lambda^{(n)} := G(\lambda^{(n-1)})$

until convergencia de $\lambda^{(n)}$

output: $(\chi_{11}(\lambda^{(n)}), \dots, \chi_{Kn_K}(\lambda^{(n)}))$ vector de aproximaciones de χ .

Con esta forma alternativa de ilustrar el método iterativo se pueden demostrar algunas propiedades referentes al mismo. Lo primero que hay que demostrar es que las búsquedas lineales en \mathcal{BS} siempre acaban con éxito.

Proposición 4.50 Sea $\langle \mathcal{S}, w \rangle$ un WTS estocástico, vivo y limitado y $B \subseteq P$ un K -corte de \mathcal{N} . Sea λ un vector de tasas asociadas a las transiciones de interfaz de los \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} . Sea $\chi_{i1}(\lambda)$ el throughput de la transición de interfaz I_{i1} y $\mu_i(\lambda)$ las tasas proporcionales de disparo de las transiciones de $T_i \cap \text{TI}$ calculadas en \mathcal{LS}_i con las tasas indicadas en λ . La función $\chi_{\mathcal{BS}_i[I_{i1}]}[\lambda_1, \dots, \mu_i, \dots, \lambda_K] : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ de la notación 4.46 es continua, monótona creciente y acotada, por lo que existe su función inversa en el intervalo de acotación. En particular, existe $\lambda \in \mathbb{R}^+$ tal que $\chi_{\mathcal{BS}_i[I_{i1}]}[\lambda_1, \dots, \mu_i, \dots, \lambda_K](\lambda) = \chi_{i1}(\lambda)$.

Demostración:

La demostración es la misma que la de la proposición 3.30 aplicando en este caso los resultados similares para WTS's, en concreto los teoremas 4.9, 4.42 y 4.43, la propiedad 4.44 y el corolario 4.41. \diamond

Este teorema asegura que en todas las iteraciones del algoritmo y en el estudio de todos los \mathcal{LS}_i existe un factor de escala λ tal que, al poner como tasas de disparo para las transiciones de $T_i \cap \text{TI}$ en \mathcal{BS} el vector $\lambda\mu_i$, el vector de throughput que se obtiene para las transiciones de \mathcal{BS} es el mismo que el que se había calculado para las transiciones de interfaz \mathcal{LS}_i . La

codificación numérica del cálculo de este factor de escala se realiza por medio de algoritmos como el de la secante (como en el caso de MG's), ya que este es un problema numérico de cálculo de una solución para una ecuación no lineal con una incognita. Se utiliza un algoritmo iterativo en el que se van probando distintos factores de escala y en función de los throughput obtenidos para las transiciones de \mathcal{BS} , teniendo en cuenta la propiedad 4.42 de monotonía de throughput respecto a las tasas de disparo de las transiciones de un WTS, se va incrementando o decrementando este factor de escala hasta que el vector de throughput de las transiciones de \mathcal{BS} y el de \mathcal{LS}_i están suficientemente cerca. Una vez calculado el factor de escala λ , multiplicando el vector μ_i de proporciones entre tasas de disparo por λ , se obtiene una nueva aproximación para las tasas de disparo de las transiciones de interfaz de la subred \mathcal{N}_i , que son las que aproximan el tiempo de respuesta de toda esta subred.

El siguiente paso consiste en demostrar la existencia de solución para el método iterativo. Al igual que en el caso de MG's, puede demostrarse la existencia de punto fijo para la función G del algoritmo 4.49. Un punto fijo de la función G se corresponde con una solución del método numérico, es decir, al poner como tasas para las transiciones de interfaz de los \mathcal{LS}_i y \mathcal{BS} los valores numéricos del punto fijo de G se obtienen los mismos throughput para todas las transiciones de interfaz de todos los \mathcal{LS}_i y \mathcal{BS} . Para demostrar la existencia de este punto fijo se utilizará el teorema del punto fijo de Brouwer (ver teorema 2.10).

Teorema 4.51 *Sea $\langle \mathcal{S}, w \rangle$ un WTS estocástico, vivo y limitado y $B \subseteq P$ un K -corte de \mathcal{N} . Sea $G : (\mathbb{R}^+)^M \rightarrow (\mathbb{R}^+)^M$ la función del algoritmo 4.49 según la notación 4.46. G es continua y existe un conjunto $S \subset (\mathbb{R}^+)^M$ compacto, convexo y no vacío tal que $G(S) \subseteq S$.*

Demostración:

La demostración es la misma que la del teorema 3.31 aplicando en este caso el teorema 4.42, las proposiciones 4.47 y 4.50 y el corolario 4.41. \diamond

Corolario 4.52 *Sea $\langle \mathcal{S}, w \rangle$ un WTS estocástico, vivo y limitado y $B \subseteq P$ un K -corte de \mathcal{N} . El algoritmo iterativo 4.49 aplicado a los sistemas \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} tiene punto fijo.*

Demostración:

Inmediato aplicando el teorema 2.10 a la función G que es la que describe una iteración del algoritmo 4.49. \diamond

Con los últimos resultados, se ha conseguido demostrar la existencia de punto fijo para el esquema iterativo utilizado, y por lo tanto, la posibilidad de convergencia del esquema a uno de esos puntos fijos. Se ha intentado estudiar más en detalle las características del esquema para asegurar convergencia, unicidad de solución e incluso estimar el error cometido en la aproximación, pero no se ha llegado a resultados generales debido a la gran complejidad de los sistemas agregados que se emplean en el esquema iterativo. Desde el punto de vista del autor de esta memoria, es posible que algún día se pueda demostrar que el esquema converge (utilizando otros teoremas de punto fijo), e incluso que este punto de convergencia sea único. Para ello sería necesario demostrar propiedades mucho más exigentes que la continuidad de las funciones empleadas en el esquema (por ejemplo la contractividad). Este tipo de propiedades parecen claras si se ven los resultados experimentales pero extremadamente difíciles de demostrar teóricamente. Lo que no parece posible es estimar el error cometido en la aproximación debido a que este error es inducido por la descomposición estructural del WTS original y a que las agregaciones de estados que se producen al construir los sistemas agregados no son exactas desde el punto de vista de CTMC's.

A falta de un mejor soporte teórico, la experimentación numérica con diferentes ejemplos sugiere que el esquema converge en muy pocos pasos (entre 3 y 5 iteraciones), converge a la misma solución independientemente de las tasas iniciales elegidas para iniciar el algoritmo, y el error cometido al final del mismo (inferior al 5%) es admisible para técnicas de aproximación en evaluación de prestaciones de sistemas concurrentes basada en modelos formales.

4.5 Conclusiones

En este capítulo se ha desarrollado en detalle el trabajo de [PJCS96b] sobre aproximación de throughput en WTS's. Se ha aprovechado la ocasión para introducir algunas mejoras técnicas que permiten afianzar cuestiones teóricas sobre la técnica así como posibles codificaciones de la misma.

La técnica de descomposición estructural de WTS's desarrollada en este capítulo es una extensión de la del capítulo 3 para MG's, en el sentido de que las dos técnicas de descomposición aplicadas a un MG son equivalentes.

El algoritmo iterativo de aproximación es, salvo ligeras adaptaciones a la nueva clase de redes, el mismo de [CCJS94] para MG's que se ha explicado

en detalle en el capítulo 3.

Todo lo expuesto en este capítulo son aportaciones de esta memoria. En concreto estas aportaciones son:

- La descomposición estructural de WTS's en un número finito arbitrario de componentes.
- Reducción del número de nodos necesarios para resumir el comportamiento de las subredes en los sistemas agregados.
- Adaptación del algoritmo iterativo de aproximación a WTS's.
- Demostración formal de la existencia de punto de convergencia para el algoritmo de aproximación del tiempo de respuesta.

A diferencia del caso de MG's, en WTS's no es posible mantener exactamente las proyecciones de estados alcanzables y secuencias de disparo del WTS original sobre los nodos preservados en cada sistema agregado con este tipo de técnicas de descomposición basadas en la estructura de la red original. En el capítulo 6 se expone una técnica basada en el estudio del espacio de estados que permite preservar al menos las proyecciones de los estados alcanzables en los sistemas agregados de cualquier red de Petri, en particular de los WTS's. Las propiedades que se preservan en los sistemas agregados por medio de esta técnica de descomposición son la vivacidad y limitación (por lo tanto, también la reversibilidad) Para mejorar las aproximaciones obtenidas con la técnica se desarrolla el concepto de resistencia de un camino. Preservando en los sistemas agregados los caminos de máxima resistencia entre transiciones de interfaz se consigue mejorar las aproximaciones.

Queda como trabajo futuro demostrar la convergencia del algoritmo numérico de aproximación del tiempo de respuesta a algún punto de convergencia de los que ya se sabe que existen. Se ha intentado demostrar sin éxito completo hasta el momento.

Capítulo 5

Aproximación de throughput en DSSP's

En este capítulo se va a desarrollar una técnica de aproximación del throughput de las transiciones de un sistema determinista de procesos secuenciales (DSSP a partir de aquí, que son las iniciales de su nombre inglés, *deterministic system of sequential processes*).

Los DSSP's están formados por un conjunto de máquinas de estados (SM's) conectadas entre sí por una serie de lugares llamados canales. Los arcos de entrada y salida de los canales pueden tener pesos. La clase de los DSSP's incluye a la de los grafos marcados con pesos (WTS's) y por lo tanto a la de los grafos marcados (MG's). Un WTS es un DSSP en el que todas sus SM's, o procesos secuenciales, contienen una única transición y un lugar. Un MG es un DSSP ordinario en el que todas sus SM's están formadas por una única transición y un lugar. Por lo tanto, habrá cierta relación entre las técnicas de aproximación de throughput en DSSP's y en WTS's. No se incluirá en este capítulo las cuestiones relativas al funcionamiento, descomposición o reducción de WTS's porque ya se ha tratado en el capítulo anterior.

Todo lo expuesto en este capítulo son aportaciones de esta memoria. En concreto, se exponen dos técnicas que permiten reducir en distinto grado las distintas SM's que componen un DSSP. Estas técnicas se presentaron en [PJCS96c, PJCS96a].

El capítulo está organizado de la siguiente manera. La sección 5.1 describe a grandes rasgos las técnicas que se van a presentar en el capítulo y

las relaciona con otras existentes en la literatura, en la sección 5.2 se desarrolla una técnica de reducción parcial de SM's (la técnica presentada en [PJCS96c]) y en la sección 5.3 se estudia bajo qué condiciones se puede realizar una reducción total de una SM en un DSSP a una sola transición (trabajo presentado en [PJCS96a]). La sección 5.4 se dedica a la técnica de descomposición estructural de DSSP's y la sección 5.5 al método iterativo para la aproximación del throughput de sus transiciones. Por último, en la sección 5.6 se indican las conclusiones finales del capítulo.

5.1 Introducción

Los DSSP's son una subclase de P/T sistemas que se obtienen de aplicar un principio simple de diseño modular e incluyen de forma controlada primitivas para el modelado de concurrencia, decisiones, sincronizaciones, bloqueos y movimiento de trabajos en bloques. Varias unidades funcionales, en este caso procesos secuenciales modelados por medio de SM's, operan de forma concurrente y cooperan entre ellos por un mecanismo de paso de mensajes a través de canales (modelados por medio de lugares con arcos de entrada y salida que pueden tener pesos). Los canales son de destino privado, es decir, los mensajes que pasan por un canal van a un único proceso secuencial. De esta forma se evita la competencia entre las distintas unidades funcionales. Además, los canales no representan condiciones laterales en los conflictos internos de las unidades funcionales, es decir, las decisiones son libres. Por lo tanto, el único efecto de los canales en el funcionamiento de las SM's es la posible ralentización de las mismas debido a la espera de la llegada de marcas a los canales de entrada de las transiciones que interactúan con ellos. En la literatura han aparecido diversas clases de redes siguiendo este principio de diseño modular [Rei82, Sou93, TSCC95, RTS95]. Los DSSP's tienen mayor potencia de modelado que los MG's o WTS's tratados en los capítulos anteriores ya que permiten modelar decisiones. En la sección 2.4.4 se desarrollan las definición y propiedades de los DSSP's. Desde el punto de vista de redes de colas, los DSSP's suponen una ligera generalización de las redes de colas distribución-conjunción ("fork-join") con sincronizaciones. En DSSP's, los servidores son complejos (SM's 1-limitadas con conexiones a distintos canales) y se permite el movimiento de trabajos en bloques.

Los DSSP's pueden emplearse para el modelado de sistemas de ensamblado y desensamblado de piezas en los que el comportamiento de las distintas

máquinas que intervienen en el proceso es complejo (modelado con las SM's).

Para que tenga sentido aproximar el throughput de las transiciones de un DSSP es necesario exigir algunas condiciones. Por un lado el DSSP debe ser conexo porque en caso contrario se tienen varios DSSP's independientes que se pueden estudiar por separado (uno por cada componente conexas). Se restringirá el estudio a DSSP's limitados con el objeto de obtener cadenas de Markov en tiempo continuo (CTMC) finitas. Además, es necesario que el DSSP sea vivo porque si no el throughput de algunas transiciones es nulo y en ese caso no tiene sentido desarrollar una técnica de aproximación. Por último, el DSSP debe tener estados recurrentes, lo que asegura la existencia de distribución en estado estacionario única para su CTMC isomorfa (teorema 2.85). Aplicando el teorema 2.101, todo DSSP vivo y limitado tiene estados recurrentes. Como el objetivo es aproximar el throughput en estado estacionario de las transiciones de un DSSP, se puede suponer sin pérdida de generalidad que el marcado inicial es un estado recurrente (si este no es el caso basta disparar una secuencia de disparo que lleve a un estado recurrente y tomar éste como marcado inicial). Con ese nuevo marcado inicial el DSSP es reversible lo que asegura la ergodicidad de su CTMC isomorfa (teorema 2.85). En consecuencia, se desarrollará una técnica de aproximación del throughput de las transiciones de DSSP's vivos y limitados, ya que estos admiten un marcado inicial que los hace reversibles.

El objetivo de este capítulo consiste en desarrollar una técnica de aproximación del throughput de las transiciones de un DSSP. Esta técnica, al igual que todas las de esta memoria, se basa en una descomposición estructural del modelo original y un método iterativo de aproximación del throughput de las transiciones basado en la preservación del tiempo de respuesta de las distintas subredes. A partir de la descomposición estructural, se construyen varios sistemas de bajo nivel (en los que hay una subred producto de la descomposición del modelo original y un resumen del resto de las subredes) y un sistema de alto nivel o esqueleto básico (formado por los resúmenes de todas las subredes producto del corte).

Para efectuar la reducción de las distintas subredes producto del corte de un DSSP se presentan dos técnicas diferentes que pretenden reducir las SM's que componen un DSSP. La primera técnica, que se presenta en la sección 5.2, está diseñada principalmente para DSSP's con grandes SM's y su objetivo es reducirlas al máximo preservando las transiciones de conexión con los canales. Esta técnica se designará como técnica de reducción parcial de SM's.

La segunda técnica, que se presenta en la sección 5.3, tiene por objetivo explotar la técnica del capítulo anterior para WTS's. Para ello es necesario conocer bajo qué condiciones una SM de un DSSP puede reducirse a una única transición. Esta técnica se designará como técnica de reducción total de SM's, pero no se podrá aplicar a cualquier SM de cualquier DSSP, sino sólo a las SM's que cumplan una serie de condiciones.

Una vez aplicadas una o las dos técnicas anteriores a las diferentes SM's, pueden quedar subredes con estructura de WTS y en ese caso podría utilizarse la técnica de reducción del capítulo anterior para WTS's.

Como las dos técnicas que se presentan persiguen la reducción de SM's, en general no tiene sentido aplicarlas a WTS's ni a MG's por lo que no se pueden comparar respecto a las técnicas de los capítulos anteriores en cuanto a los efectos de reducción. Por otra parte, los MG's y WTS's son a su vez DSSP's. Desde este punto de vista, la técnica de aproximación de throughput para DSSP's es una generalización de las presentadas en los capítulos anteriores para MG's y WTS's, en el sentido de que la técnica de aproximación para DSSP's aplicada a un MG o a un WTS es exactamente la misma que las de los capítulos anteriores para cada clase de redes.

La técnica de reducción parcial de SM's preserva un conjunto de propiedades más fuertes que la técnica de reducción total de SM's. En concreto, la técnica de reducción parcial de SM's preserva las probabilidades de ramificación entre transiciones observables (en la sección 5.2 se definirá formalmente estas probabilidades), las proyecciones de las secuencias de disparo sobre las transiciones que se preservan en la reducción y las proyecciones de marcados alcanzables sobre los lugares que se preservan, cuando tengan sentido estas proyecciones. La técnica de reducción total preserva vivacidad y limitación (y por el teorema 2.101 la existencia de estados recurrentes). Por lo tanto, si se emplean las dos técnicas de reducción a la vez, sólo se puede asegurar que los sistemas agregados son vivos y limitados, por lo que tienen estados recurrentes (teorema 2.101) y sus CTMC's isomorfas tienen distribución en estado estacionario única (teorema 2.85).

Por lo que respecta a la parte numérica, no hay grandes cambios respecto a las expuestas en los capítulos anteriores para MG's y WTS's. En este capítulo se adapta el algoritmo numérico de aproximación del tiempo de respuesta para operar con DSSP's. Los experimentos numéricos realizados aseguran buen comportamiento del método, con convergencia en pocas iteraciones (entre 3 y 5) y error asumible en una técnica de aproximación (inferior al 5%).

5.2 Reducción parcial de SM's

En esta sección se va a presentar una técnica de reducción de SM's que va a ser necesaria para la descomposición y reducción de DSSP's. La técnica está diseñada especialmente para DSSP's compuestos por SM's grandes, ya que en otro caso el efecto de la reducción no es notable. Este trabajo fue desarrollado en [PJCS96c] y utiliza una técnica similar a la desarrollada en [BI82] para redes de colas.

En la sección 2.4.2 se da la definición de las SM's y sus principales propiedades. Son redes ordinarias en las que las transiciones tienen un único lugar de entrada y uno único de salida. Además, en los DSSP's se exige que las SM's que lo componen sean fuertemente conexas y 1-limitadas, es decir, que tengan un único lugar marcado y con una sola marca. Por lo tanto, la técnica de reducción que se va a desarrollar en esta sección es válida para cualquier SM fuertemente conexa y 1-limitada.

Como las SM's de un DSSP están conectadas entre sí por medio de canales, se preservarán en cada SM las transiciones que tienen canales como lugares de entrada o salida. Esto se traduce en que la técnica de reducción debe distinguir entre dos tipos de transiciones en una SM; las transiciones que hay que preservar, que se llamarán *observables* y las que se reducirán, que se llamarán *no observables*. El objetivo de la técnica consiste en calcular, a partir de una SM fuertemente conexa y 1-limitada, otra SM fuertemente conexa y 1-limitada preservando las transiciones observables y reduciendo al máximo el grafo tangible de alcanzabilidad. De esa manera, los sistemas agregados que se construyan a partir de la descomposición de un DSSP tendrán el menor número de estados y por lo tanto sus CTMC's isomorfas serán lo más pequeñas posible.

Pero no es suficiente reducir nodos en las SM's. Para que el método numérico de aproximación de throughput funcione correctamente es necesario que en el proceso de reducción se preserven una serie de propiedades respecto al funcionamiento interno de la SM. En concreto, con esta técnica de reducción se mantienen exactamente las proyecciones de estados alcanzables y de secuencias de disparo sobre los nodos preservados, cuando estas proyecciones tengan sentido. Más aun, se preservarán las probabilidades de ramificación entre transiciones observables (en la definición 5.3 se definirán estas probabilidades). De esa forma se preservan otras propiedades lógicas tales como vivacidad, limitación, pasos y existencia de estados recurrentes.

A lo largo de toda esta sección se supondrá que el marcado inicial de las

SM's es tal que sensibiliza una transición observable. Si no es éste el caso, en un tiempo $O(|T|)$ se puede calcular un marcado alcanzable de la SM que cumpla esta condición y considerarlo como inicial.

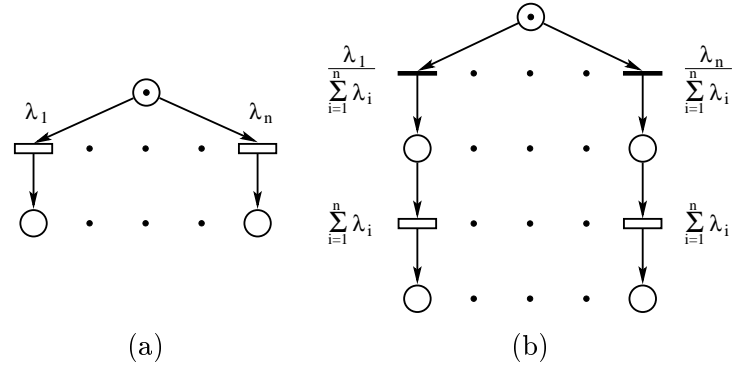


Figura 5.1: Cambio de estructura de un conflicto.

Se puede suponer, sin pérdida de generalidad, que los conflictos entre transiciones de una SM están modelados por medio de transiciones inmediatas a las que se les asocia unas probabilidades de disparo independientes del marcado y del tiempo. De esta forma se separa la resolución de conflictos de la duración de las actividades. Si éste no fuera el caso, es inmediato modificar la estructura y temporización de la SM para que cumpla la suposición. En la figura 5.1 se observa cómo se puede modificar un conflicto entre n transiciones temporizadas por su equivalente con transiciones inmediatas. Si se tienen n transiciones $\{t_i\}_{i=1}^n$ a las que se les asocia un tiempo de disparo distribuido exponencialmente con tasas $\{\lambda_i\}_{i=1}^n$ respectivamente, la probabilidad de que se dispare la transición t_i si se utiliza política de carrera es $\lambda_i / \sum_{j=1}^n \lambda_j$, de ahí las probabilidades asociadas a cada transición inmediata. Además, el tiempo de resolución del conflicto está distribuido exponencialmente con tasa $\sum_{i=1}^n \lambda_i$, de ahí las tasas asociadas a las nuevas transiciones temporizadas. Realizando esta transformación en los conflictos se puede suponer que la SM no tiene en su estructura ningún conflicto entre transiciones temporizadas. Esta transformación es válida también para SM's pertenecientes a un DSSP ya que, por la definición de DSSP, los conflictos internos de las SM son libres, es decir, no resultan modificados por las conexiones de la SM con los canales.

Para exponer la técnica de reducción de SM's es necesario definir formalmente algunos conceptos.

Definición 5.1 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle B \cup (\bigcup_{i=1}^K P_i), \bigcup_{i=1}^K T_i, F, W \rangle$ un DSSP compuesto por K SM's, donde B es su conjunto de canales. Sea $\langle \mathcal{N}_i, \mathbf{m}_0^i \rangle$ con $\mathcal{N}_i = \langle P_i, T_i, F_i \rangle$ una SM de \mathcal{S} .

- i) $T_v = T_i \cap (\bullet B \bullet)$ es el conjunto de transiciones observables de \mathcal{N}_i .
- ii) $T_i \setminus T_v$ es el conjunto de transiciones no observables de \mathcal{N}_i .

Las transiciones observables de una SM en un DSSP son las transiciones de la SM que tienen algún canal como lugar de entrada o salida. Las transiciones no observables de una SM en un DSSP son las transiciones de la SM que no tienen ningún canal como lugar de entrada o salida.

Definición 5.2 Sea $\langle \mathcal{S}, w \rangle$ una SM estocástica, 1-limitada y fuertemente conexa y $t_i, t_j \in T$ dos transiciones de \mathcal{N} . La probabilidad de encaminamiento de t_i a t_j , denotada por p_{ij} es la probabilidad de que t_j sea la primera transición de \mathcal{N} que se dispara después del disparo de t_i .

Las probabilidades de encaminamiento cumplen la misma función que las *routing probabilities* en redes de colas. Las probabilidades de encaminamiento en redes de colas indican la probabilidad p_{ij} de que un cliente que abandona la estación t_i vaya a la estación t_j . Las SM's se pueden ver como redes de colas en las que cada transición modela una estación y las transiciones inmediatas utilizadas para resolver los conflictos modelan las probabilidades de encaminamiento. El disparo de una transición temporizada en una SM produce la sensibilización de otra transición temporizada con una probabilidad dependiente de los pesos asociados a las transiciones inmediatas que resuelven los conflictos. Por ser la SM 1-limitada y resolver los conflictos por medio de transiciones inmediatas, no puede haber dos transiciones temporizadas a la vez, por lo que la definición anterior es correcta con el lenguaje de redes de Petri. De hecho, al resolver los conflictos de la SM con transiciones inmediatas, lo que se hace es introducir en la estructura de la SM estas probabilidades de encaminamiento, cosa que se hace de forma natural en redes de colas.

Definición 5.3 Sea $\langle \mathcal{S}, w \rangle$ una SM estocástica, 1-limitada y fuertemente conexa. Sea $T_v \subseteq T$ un subconjunto de transiciones observables de \mathcal{N} y $t_i, t_j \in T_v$ dos transiciones observables. La probabilidad de ramificación de t_i a t_j , denotada por q_{ij} es la probabilidad de que t_j sea la primera transición observable de \mathcal{N} que se dispara después del disparo de t_i .

El siguiente resultado relaciona las probabilidades de encaminamiento y de ramificación en una SM con una única transición no observable.

Proposición 5.4 *Sea $\langle \mathcal{S}, w \rangle$ una SM estocástica, 1-limitada y fuertemente conexa donde $T = \{t_i\}_{i=1}^n$ y $T_v = T \setminus \{t_n\}$ su conjunto de transiciones observables. Sean p_{ij} con $1 \leq i, j \leq n$ sus probabilidades de encaminamiento y q_{ij} con $1 \leq i, j < n$ sus probabilidades de ramificación. Entonces, para todo $1 \leq i, j < n$ se tiene:*

$$q_{ij} = p_{ij} + p_{in} \left(\sum_{u=0}^{\infty} (p_{nn})^u \right) p_{nj} = p_{ij} + \frac{p_{in}p_{nj}}{(1-p_{nn})}.$$

Demostración:

En este caso hay $n - 1$ transiciones observables $\{t_i\}_{i=1}^{n-1}$ y una transición no observable t_n . Por la definición 5.3, sólo tiene sentido hablar de probabilidades de ramificación q_{ij} para $1 \leq i, j < n$.

Una vez fijados i, j con $1 \leq i, j < n$, la probabilidad de ramificación q_{ij} es la probabilidad de que, después del disparo de t_i , la siguiente transición observable sensibilizada sea t_j . Sea $\mathbf{m} \in \text{RG}(\mathcal{S})$ tal que t_i está sensibilizada en \mathbf{m} . En $\langle \mathcal{N}, \mathbf{m} \rangle$ las únicas secuencias de disparo que cumplen las condiciones para el cálculo de q_{ij} son $\sigma = t_i(t_n)^u t_j$ con $u \geq 0$.

Si $u = 0$, entonces $\sigma = t_i t_j$, y la probabilidad de que ocurra esto es p_{ij} .

Si $u > 0$, entonces $\sigma = t_i(t_n)^u t_j$, y la probabilidad de que ocurra esto es $p_{in} p_{nn}^{u-1} p_{nj}$.

Resumiendo, $q_{ij} = p_{ij} + \sum_{u=1}^{\infty} (p_{in} p_{nn}^{u-1} p_{nj}) = p_{ij} + p_{in} (\sum_{u=0}^{\infty} (p_{nn})^u) p_{nj}$. Por ser p_{nn} una probabilidad $p_{nn} \leq 1$. Además si $p_{nn} = 1$ quiere decir que después del disparo de t_n la única transición sensibilizada vuelve a ser t_n , es decir que ya no se abandona t_n , lo que contradice la hipótesis de que la SM es fuertemente conexa. Por lo tanto, debe ser $p_{nn} < 1$, es decir, $q_{ij} = p_{ij} + \frac{p_{in}p_{nj}}{(1-p_{nn})}$. \diamond

Este resultado permite ir eliminando una por una las transiciones no observables de una SM y construir en cada paso una nueva SM con una transición menos tal que las probabilidades de encaminamiento de la nueva SM sean exactamente las probabilidades de ramificación de la SM original. Cuando todas las transiciones no observables se han eliminado, las probabilidades de encaminamiento de la SM resultante (que sólo tiene como transiciones temporizadas a las transiciones observables de la SM original) coinciden con las probabilidades de ramificación de la SM original.

El algoritmo de cálculo de las probabilidades de ramificación de una SM, dado su conjunto T_v de transiciones observables cualquiera es el siguiente:

Algoritmo 5.5 Cálculo de las probabilidades de ramificación.

input: $T = \{t_i\}_{i=1}^{r+s}$ conjunto de transiciones de una SM
 $T_v = \{t_i\}_{i=1}^r$ conjunto de transiciones observables
 \mathbf{P} matriz $(r+s) \times (r+s)$ de probabilidades de encaminamiento
 $\mathbf{Q} := \mathbf{P}$
for $k := r+s$ **down to** $r+1$ **do**
 for $i := 1$ **to** $k-1$ **do**
 for $j := 1$ **to** $k-1$ **do**
 $\mathbf{Q}(i, j) := \mathbf{Q}(i, j) + \frac{\mathbf{Q}(i, k)\mathbf{Q}(k, j)}{1 - \mathbf{Q}(k, k)}$
 end for
 end for
end for
output: \mathbf{Q} Matriz $r \times r$ de probabilidades de ramificación.

La explicación de este algoritmo se sigue de la proposición 5.4. La entrada del algoritmo es el conjunto de transiciones de una SM, sus probabilidades de encaminamiento y el conjunto de transiciones consideradas como observables. Para simplificar el algoritmo, se supone que la SM tiene $r+s$ transiciones y las r primeras son las observables. Al principio se inicializan las probabilidades de ramificación como las de encaminamiento. El bucle exterior recorre el conjunto de transiciones no observables. Al eliminar la transición no observable t_k se recalculan las probabilidades de encaminamiento de una nueva SM con una transición menos que la anterior de acuerdo con la fórmula de la proposición 5.4. Al finalizar el algoritmo se tienen calculadas las probabilidades de encaminamiento de la SM que resulta de eliminar de la original todas sus transiciones no observables. El orden de eliminación de las transiciones no observables no afecta a los resultados finales del algoritmo. Se utiliza el recorrido del algoritmo porque es el que simplifica los accesos a las estructuras de datos. Y esta nueva SM tiene como probabilidades de encaminamiento las probabilidades de ramificación de la original.

La complejidad de este algoritmo es $O((r+s)^3) = O(|T|^3)$ en tiempo y $O((r+s)^2) = O(|T|^2)$ en espacio, es decir, polinómico en el número de transiciones de la SM original.

El siguiente paso consiste en utilizar las probabilidades de ramificación calculadas con el algoritmo anterior para construir la SM reducida.

Algoritmo 5.6 Cálculo de la SM reducida.

input: $T_v = \{t_i\}_{i=1}^r$ conjunto de transiciones observables
Q matriz $r \times r$ de probabilidades de encaminamiento
 $P := \emptyset; T := T_v; F := \emptyset$
for $i := 1$ **to** r **do**
 $P := P \cup \{p_i^{in}, p_i^{out}\}$
 $F := F \cup \{(p_i^{in}, t_i), (t_i, p_i^{out})\}$
end for
for $i := 1$ **to** r **do**
 for $j := 1$ **to** r **do**
 if $q_{ij} = 1$ **then**
 fusionar p_i^{out} y p_j^{in}
 else if $q_{ij} > 0$ **then**
 $T := T \cup \{t_{ij} \text{ transición inmediata con probabilidad } q_{ij}\}$
 $F := F \cup \{(p_i^{out}, t_{ij}), (t_{ij}, p_j^{in})\}$
 end if
 end for
end for
output: $\langle P, T, F \rangle$ SM reducida.

Es evidente que la salida de este algoritmo es una SM ya que cada transición que se añade a la estructura, sea temporizada o inmediata, tiene un único lugar de entrada y un único lugar de salida. La SM reducida tiene tantas transiciones temporizadas como transiciones observables tiene la SM original. Por cada transición temporizada t_i se añade un lugar p_i^{in} de entrada y otro p_i^{out} de salida junto con los arcos correspondientes. Esta tarea se hace en el primer bucle que recorre las transiciones observables de la SM original. Posteriormente hay que añadir transiciones inmediatas según las probabilidades de ramificación q_{ij} que se han calculado en el algoritmo 5.5. Para simplificar la estructura de la SM reducida al máximo se distinguen tres casos.

Si $q_{ij} = 0$, quiere decir que en la SM original, después del disparo de la transición observable t_i , no se puede sensibilizar la transición observable t_j sin pasar antes por otra transición observable. En este caso no hay que añadir nuevos nodos a la SM reducida.

Si $q_{ij} = 1$, quiere decir que en la SM original, el disparo de la transición observable t_i sensibiliza únicamente a la transición observable t_j . En este caso hay que fusionar en la SM reducida el lugar p_i^{out} de salida de t_i con el

lugar p_j^{in} de entrada de t_j .

Si $0 < q_{ij} < 1$, quiere decir que en la SM original, después del disparo de la transición observable t_i se pueden sensibilizar al menos 2 transiciones observables distintas, es decir, hay un conflicto en la SM reducida a la salida de t_i . Por ello se añade una transición inmediata t_{ij} que conecte el lugar p_i^{out} de salida de t_i con el lugar p_j^{in} de entrada de t_j . La probabilidad de disparo de t_{ij} debe ser igual a la probabilidad de ramificación q_{ij} .

La complejidad de este algoritmo es $O(r^2) = O(|T_v|^2)$ en espacio y en tiempo, es decir, cuadrático en el número de transiciones observables de la SM original.

El número de transiciones inmediatas que tiene la SM reducida puede ser, en el peor caso, cuadrático en el número de transiciones observables. Pero en cualquier caso, el número de estados alcanzables tangibles (aquellos que sensibilizan transiciones temporizadas) es exactamente el número de transiciones observables, es decir, menor que el número de estados alcanzables tangibles de la SM original, suponiendo que T_v es un subconjunto propio de T . Si $T_v = T$ no hay reducción.

El algoritmo anterior permite calcular la estructura de la SM reducida. Para completar toda la información de la SM reducida falta hablar del marcado inicial y de las tasas asociadas a las transiciones temporizadas. Como se ha supuesto al comienzo de la sección que la SM original tenía un marcado inicial que sensibilizaba una transición observable t , el marcado inicial de la SM reducida será el único que sensibiliza a t , es decir, el marcado con una sola marca en t^{in} . Respecto a las tasas de las transiciones temporizadas, basta mantener las mismas tasas que tenían estas transiciones en la SM original.

Una vez expuesta la técnica de reducción parcial de SM's se van a demostrar sus principales propiedades.

Teorema 5.7 *Sea $\langle \mathcal{S}, w \rangle$ una SM estocástica, 1-limitada y fuertemente conexa y $T_v \subseteq T$ un subconjunto de transiciones observables de \mathcal{N} . Sea $\langle \mathcal{S}_v, w_v \rangle$ la SM que resulta de reducir $\langle \mathcal{S}, w \rangle$ respecto a T_v . Entonces las probabilidades de encaminamiento de $\langle \mathcal{S}_v, w_v \rangle$ son las probabilidades de ramificación de $\langle \mathcal{S}, w \rangle$ entre transiciones observables.*

Demostración:

Evidente por la construcción de $\langle \mathcal{S}_v, w_v \rangle$ (ver la proposición 5.4 y los algoritmos 5.5 y 5.6). \diamond

Teorema 5.8 Sea $\langle \mathcal{S}, w \rangle$ una SM estocástica, 1-limitada y fuertemente conexa y $T_v \subseteq T$ un subconjunto de transiciones observables de \mathcal{N} . Sea $\langle \mathcal{S}_v, w_v \rangle$ la SM que resulta de reducir $\langle \mathcal{S}, w \rangle$ respecto a T_v . Sean \mathbf{v} y \mathbf{v}_v los vectores de ratios de visita (ver definición 2.66) de $\langle \mathcal{S}, w \rangle$ y $\langle \mathcal{S}_v, w_v \rangle$ respectivamente. Entonces:

1. $\mathbf{v}|_{T_v} = \mathbf{v}_v$.
2. $L(\mathcal{S})|_{T_v} = L(\mathcal{S}_v)|_{T_v}$.
3. $\{\mathbf{m}|_{\bullet_{T_v}} \mid \mathbf{m} \in R(\mathcal{S})\} \setminus \mathbf{0} = \{\mathbf{m}|_{\bullet_{T_v}} \mid \mathbf{m} \in R(\mathcal{S}_v)\} \setminus \mathbf{0}$.

Demostración:

1) Basta demostrar que al eliminar una transición no observable de $\langle \mathcal{S}, w \rangle$, se mantienen las ratios de visita del resto de transiciones. En $\langle \mathcal{S}, w \rangle$ las ratios de visita de las distintas transiciones cumplen las ecuaciones de balance local [Kan92], esto es: $v_j = \sum_{i=1}^n v_i p_{ij}$ para todo $1 \leq j \leq n$ donde n es el número de transiciones de \mathcal{N} . Sea t_k una transición no observable que se va a eliminar. Se tiene que $v_k = \sum_{i=1}^n v_i p_{ik} = v_k p_{kk} + \sum_{i \neq k} v_i p_{ik}$, es decir, $v_k = \frac{1}{1-p_{kk}} \sum_{i \neq k} v_i p_{ik}$. Sustituyendo v_k en v_j con $j \neq k$ queda: $v_j = \sum_{i=1}^n v_i p_{ij} = v_k p_{kj} + \sum_{i \neq k} v_i p_{ij} = \frac{p_{kj}}{1-p_{kk}} \sum_{i \neq k} v_i p_{ik} + \sum_{i \neq k} v_i p_{ij} = \sum_{i \neq k} v_i (p_{ij} + \frac{p_{ik} p_{kj}}{1-p_{kk}}) = \sum_{i \neq k} v_i q_{ij}$ (la última igualdad es por la proposición 5.4). Estos cálculos demuestran que las ratios de visita de las transiciones t_j con $j \neq k$ en $\langle \mathcal{S}, w \rangle$ coinciden con las ratios de visita en la SM que resulta de eliminar t_k con la técnica de reducción. Por lo tanto, al eliminar todas las transiciones no observables por medio de la técnica de reducción, se preservan las ratios de visita de las transiciones observables.

2) Sea $\sigma \in L(\mathcal{S})$. Entonces σ puede ponerse como $\sigma = t_{i_1} \rho_1 \dots t_{i_n} \rho_n$ donde $t_{i_j} \in T_v$ para $1 \leq j \leq n$ y $\rho_j \subseteq T \setminus T_v$ (pueden ser vacías) para $1 \leq j \leq n$. Sean $\mathbf{m}_j \in R(\mathcal{S})$ y $\mathbf{m}_j^v \in R(\mathcal{S}_v)$ los marcados que sensibilizan la transición observable $t_{i_j} \in T_v$. En estas condiciones $\sigma \in L(\mathcal{S}) \Leftrightarrow t_{i_j} \rho_j t_{i_{j+1}} \in L(\mathcal{N}, \mathbf{m}_j)$ para todo $1 \leq j < n$. Como $\rho_j \subseteq T \setminus T_v$, $t_{i_j} \rho_j t_{i_{j+1}} \in L(\mathcal{N}, \mathbf{m}_j)$ si y sólo si $q_{i_j i_{j+1}} > 0$ en $\langle \mathcal{S}, w \rangle$ para todo $1 \leq j < n$. Por la proposición 5.4, esto ocurre si y sólo si $p_{i_j i_{j+1}} > 0$ en $\langle \mathcal{S}_v, w_v \rangle$ para todo $1 \leq j < n$. Para cada $1 \leq j < n$ sea ρ'_j la secuencia de disparo vacía si $p_{i_j i_{j+1}} = 1$ o la secuencia formada por la transición inmediata que conecta el lugar de salida de t_{i_j} con el de entrada de $t_{i_{j+1}}$ si $p_{i_j i_{j+1}} < 1$. Entonces $t_{i_j} \rho_j t_{i_{j+1}} \in L(\mathcal{N}, \mathbf{m}_j)$ si y sólo si $t_{i_j} \rho'_j t_{i_{j+1}} \in L(\mathcal{N}_v, \mathbf{m}_j^v)$ para todo $1 \leq j < n$. Sea ahora $\sigma' = t_{i_1} \rho'_1 \dots \rho'_{i_{n-1}} t_{i_n}$.

Por construcción, se tiene que $\sigma|_{T_v} = t_{i_1} \dots t_{i_n} = \sigma'|_{T_v}$ y $\sigma \in L(\mathcal{S})$ si y sólo si $\sigma' \in L(\mathcal{S}_v)$.

3) Por ser \mathcal{S} fuertemente conexa y 1-limitada, $\{\mathbf{m}|_{\bullet T_v} \mid \mathbf{m} \in R(\mathcal{S})\} \setminus \mathbf{0}$ es el conjunto de las proyecciones sobre $\bullet T_v$ de los marcados alcanzables de $R(\mathcal{S})$ que sensibilizan a una transición observable. Análogamente, por ser \mathcal{S}_v fuertemente conexa y 1-limitada, $\{\mathbf{m}|_{\bullet T_v} \mid \mathbf{m} \in R(\mathcal{S}_v)\} \setminus \mathbf{0}$ es el conjunto de las proyecciones sobre $\bullet T_v$ de los marcados alcanzables de $R(\mathcal{S}_v)$ que sensibilizan a una transición observable. Como el conjunto de transiciones observables de \mathcal{N} y \mathcal{N}_v son iguales, los dos conjuntos coinciden. \diamond

El primer punto de este teorema demuestra que las ratios de visita de las transiciones observables se preservan en el proceso de reducción de SM's. Ésta es una propiedad más debil que la preservación de las probabilidades de ramificación del teorema 5.7. El segundo punto establece que la SM reducida es una proyección exacta sobre el conjunto de transiciones observables de la SM original en términos de secuencias de disparo. Las transiciones observables son las que se preservan en el proceso de reducción. El tercer punto establece una propiedad muy parecida al mantenimiento exacto de las proyecciones de estados alcanzables sobre los lugares mantenidos en el proceso de reducción. La proyección de un marcado sólo tiene sentido si ésta tiene marcas en algún lugar. Como los únicos lugares que se preservan son los de entrada a una transición observable, es posible que la proyección de un marcado alcanzable en la SM original o en la reducida dé como resultado el marcado vacío $\mathbf{0}$. Este caso se elimina de los conjuntos proyectados.

Ahora es necesario ver cómo afecta la reducción de SM's de un DSSP al funcionamiento del mismo. Las transiciones observables de una SM en un DSSP son aquellas que tienen algún canal como lugar de entrada o de salida, por lo que el conjunto de transiciones observables de una SM en un DSSP queda determinado por la estructura del DSSP. Para realizar este estudio se tendrán en cuenta las mismas consideraciones iniciales sobre el marcado inicial que para SM's. Entonces, el marcado inicial del DSSP será tal que en cada SM que lo compone está sensibilizada una transición observable. Si éste no es el caso, con un algoritmo de complejidad lineal en el número de transiciones del DSSP se puede conseguir un marcado alcanzable que cumpla la condición. Basta considerar este marcado alcanzable como marcado inicial.

El siguiente resultado da un algoritmo de cálculo de las ratios de visita de las transiciones de un DSSP.

Teorema 5.9 [Cam90] Sea $\langle \mathcal{S}, w \rangle$ con $\mathcal{N} = \langle B \cup (\bigcup_{i=1}^K P_i), \bigcup_{i=1}^K T_i, F, W \rangle$ un DSSP estocástico, vivo y limitado compuesto por K SM's (denotadas por \mathcal{N}_i). Las ratios de visita de las transiciones de \mathcal{N} pueden calcularse por medio del siguiente algoritmo de complejidad polinomial en el tamaño de la red.

1. Para cada $1 \leq i \leq K$, calcular el vector \mathbf{u}_i de ratios de visita para las transiciones de \mathcal{N}_i .
2. Para cada $1 \leq i \leq K$, sustituir \mathcal{N}_i por una sólo transición t_i con $W(b, t_i) = \sum_{t \in T_i} \mathbf{u}_i(t) W(b, t)$ y $W(t_i, b) = \sum_{t \in T_i} \mathbf{u}_i(t) W(t, b)$ para todo $b \in B$.
3. Calcular el vector \mathbf{w} de ratios de visita de las transiciones del sistema resultante del apartado 2 resolviendo el sistema $\mathbf{C} \cdot \mathbf{w} = \mathbf{0}$; $\mathbf{w}(1) = 1$.
4. El vector \mathbf{v} de ratios de visita de las transiciones de \mathcal{N} viene dado por $\mathbf{v}(t) = \mathbf{w}(i) \mathbf{u}_i(t)$, para cada $t \in T_i$ y para todo $1 \leq i \leq K$.

Este resultado demuestra que la ratio de visita $\mathbf{v}(t)$ de una transición t del DSSP depende de la ratio de visita $\mathbf{u}_i(t)$ de t en la SM a la que pertenece tomada por separado y de un factor $\mathbf{w}(i)$ que depende de las conexiones entre las SM's del DSSP.

Se está en condiciones de demostrar las principales propiedades que se preservan al reducir una SM en un DSSP.

Teorema 5.10 Sea $\langle \mathcal{S}, w \rangle$ con $\mathcal{N} = \langle B \cup (\bigcup_{i=1}^K P_i), \bigcup_{i=1}^K T_i, F, W \rangle$ un DSSP estocástico, vivo y limitado compuesto por K SM's. Sea $\langle \mathcal{S}_i, w_i \rangle$ una SM de $\langle \mathcal{S}, w \rangle$, $T_v = T_i \cap (\bullet B \cup B \bullet)$ su conjunto de transiciones observables y $\langle \mathcal{S}'_i, w'_i \rangle$ la SM que resulta de reducir $\langle \mathcal{S}_i, w_i \rangle$ respecto a T_v . Sea $\langle \mathcal{S}', w' \rangle$ el DSSP que resulta de sustituir $\langle \mathcal{S}_i, w_i \rangle$ por $\langle \mathcal{S}'_i, w'_i \rangle$ en $\langle \mathcal{S}, w \rangle$. Si \mathbf{v} y \mathbf{v}' son los vectores de ratios de visita de $\langle \mathcal{S}, w \rangle$ y $\langle \mathcal{S}', w' \rangle$ respectivamente, entonces:

1. $\mathbf{v}|_{T''} = \mathbf{v}'|_{T''}$.
2. $L(\mathcal{S})|_{T''} = L(\mathcal{S}')|_{T''}$.
3. $\{\mathbf{m} \in R(\mathcal{S}) \mid \mathbf{m}|_{\bullet T_v} \neq \mathbf{0}\}|_{P''} = \{\mathbf{m} \in R(\mathcal{S}') \mid \mathbf{m}|_{\bullet T_v} \neq \mathbf{0}\}|_{P''}$.

Donde $T'' = (T \setminus T_i) \cup T_v$ y $P'' = (P \setminus P_i) \cup \bullet T_v$.

Demostración:

1) Por el teorema anterior, la ratio de visita de una transición t de \mathcal{N}_j es la ratio de visita $\mathbf{u}_j(t)$ de t en $\langle \mathcal{S}_j, w_j \rangle$, ponderada por el peso $\mathbf{w}(j)$ asociado a $\langle \mathcal{S}_j, w_j \rangle$ en $\langle \mathcal{S}, w \rangle$. Al reducir transiciones no observables de \mathcal{N}_i , los pesos $\mathbf{w}(j)$ con $1 \leq j \leq K$ no se modifican ya que no cambian las conexiones entre SM's del DSSP. Por otro lado, la reducción de la transición $t \in T_i$ no modifica los vectores \mathbf{u}_j de ratios de visita para $j \neq i$ ya que no se modifica $\langle \mathcal{S}_j, w_j \rangle$. Por último, por el apartado 1 del teorema 5.8, la reducción de $t \in T_i$ preserva las ratios de visita $\mathbf{u}_i(t)$ con $t \in T_v$. Por lo tanto, la reducción de todas las transiciones no observables de una SM en un DSSP preserva las ratios de visita de las transiciones $t \in (T \setminus T_i) \cup T_v = T''$.

2) \subseteq) Sea $\sigma \in L(\mathcal{S})$. Entonces $\sigma = \rho_0 \sigma_1 \rho_1 \dots \sigma_n \rho_n$ donde $\sigma_j \subseteq T_i$ para $1 \leq j \leq n$ y $\rho_j \in T \setminus T_i$ (pueden ser vacías) para $0 \leq j \leq n$. Si $\bar{\sigma} = \sigma_1 \dots \sigma_n$ entonces $\bar{\sigma} \in L(\mathcal{S}_i)$. Por el apartado 2 del teorema 5.8, existe $\bar{\sigma}' \in L(\mathcal{S}'_i)$ tal que $\bar{\sigma}|_{T_v} = \bar{\sigma}'|_{T_v}$. Ahora se puede descomponer $\bar{\sigma}' = \sigma'_1 \dots \sigma'_n$ de forma que $\sigma_j|_{T_v} = \sigma'_j|_{T_v}$ para $1 \leq j \leq n$. La descomposición no es única en general, pero no afecta a resto de la demostración. Sea $\sigma' = \rho_0 \sigma'_1 \rho_1 \dots \sigma'_n \rho_n$. Por construcción, es evidente que $\sigma|_{T''} = \sigma'|_{T''}$. Falta demostrar que $\sigma' \in L(\mathcal{S}')$. Se hará por inducción sobre n .

Para $n = 0$ se tiene $\sigma = \rho_0 = \sigma' \subseteq T \setminus T_i$, luego $\sigma' \in L(\mathcal{S}')$ ya que \mathcal{N} y \mathcal{N}' coinciden en $T \setminus T_i$ y $\sigma \in L(\mathcal{S})$.

Suponiendo el resultado cierto para $n - 1$, ver que también lo es para n . Sean $\tau = \rho_0 \sigma_1 \rho_1 \dots \sigma_{n-1} \rho_{n-1}$ y $\tau' = \rho_0 \sigma'_1 \rho_1 \dots \sigma'_{n-1} \rho_{n-1}$. Por la hipótesis inicial, $\tau \in L(\mathcal{S})$ y por hipótesis de inducción $\tau' \in L(\mathcal{S}')$. Sean $\mathbf{m}_0 \xrightarrow{\tau} \mathbf{m}$ en \mathcal{S} , $\mathbf{m}'_0 \xrightarrow{\tau'} \mathbf{m}'$ en \mathcal{S}' , $\mathbf{m}_i = \mathbf{m}|_{P_i}$ y $\mathbf{m}'_i = \mathbf{m}'|_{P'_i}$ (P'_i denota el conjunto de lugares de \mathcal{S}'_i). Es claro que \mathbf{m} y \mathbf{m}' coinciden en los canales y en los lugares de \mathcal{N}_j con $j \neq i$ ya que $\tau|_{T''} = \tau'|_{T''}$. Como $\sigma_n \in L(\mathcal{N}, \mathbf{m})$, el marcado de los canales en \mathbf{m} no interfiere el disparo local de σ_n en $\langle \mathcal{N}_i, \mathbf{m}_i \rangle$. Por ser $\mathbf{m}|_B = \mathbf{m}'|_B$ y $\sigma_n|_{T_v} = \sigma'_n|_{T_v}$, el marcado de los canales no interfiere el posible disparo local de σ'_n en $\langle \mathcal{N}'_i, \mathbf{m}'_i \rangle$. Además, localmente en \mathcal{N}'_i se tiene que $\bar{\sigma}' \in L(\mathcal{N}'_i, \mathbf{m}'_0)$, luego $\sigma'_n \in L(\mathcal{N}'_i, \mathbf{m}'_i)$, es decir, $\tau' \sigma'_n \in L(\mathcal{S}')$. Sean ahora $\mathbf{m} \xrightarrow{\sigma_n} \mathbf{m}_1$ en \mathcal{S} y $\mathbf{m}' \xrightarrow{\sigma'_n} \mathbf{m}'_1$ en \mathcal{S}' . Como $\tau \sigma_n|_{T''} = \tau' \sigma'_n|_{T''}$, entonces \mathbf{m}_1 y \mathbf{m}'_1 coinciden en los canales y en los lugares de \mathcal{N}_j con $j \neq i$. Además, $\rho_n \in L(\mathcal{N}, \mathbf{m}_1)$ luego $\rho_n \in L(\mathcal{N}', \mathbf{m}'_1)$, es decir, $\sigma' = \tau' \sigma'_n \rho_n \in L(\mathcal{S}')$.

\supseteq) Se demuestra con la misma técnica del apartado \subseteq), construyendo a partir de $\sigma' \in L(\mathcal{S}')$ una secuencia $\sigma \in L(\mathcal{S})$ cambiando la subsecuencia $\bar{\sigma}' = \sigma'|_{T_i} \in L(\mathcal{S}'_i)$ por otra $\bar{\sigma} \in L(\mathcal{S}_i)$ tal que $\bar{\sigma}'|_{T_v} = \bar{\sigma}|_{T_v}$ que se sabe que

existe por el apartado 2 del teorema 5.8.

3) \subseteq) Sea $\mathbf{m} \in R(\mathcal{S})$ tal que $\mathbf{m}|_{\bullet_{T_v}} \neq \mathbf{0}$ y $\sigma \in L(\mathcal{S})$ tal que $\mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$. Sean $\bar{\sigma} = \sigma|_{T_i}$ y $\mathbf{m}_i = \mathbf{m}|_{P_i}$. En \mathcal{S}_i se tiene $\mathbf{m}_0^i \xrightarrow{\bar{\sigma}} \mathbf{m}_i$ con alguna $t \in T_v$ sensibilizada en \mathbf{m}_i (por ser $\mathbf{m}|_{\bullet_{T_v}} \neq \mathbf{0}$). Sea $\bar{\sigma}_1 = \bar{\sigma}t \in L(\mathcal{S}_i)$. Por el apartado 2 del teorema 5.8, existe $\bar{\sigma}'_1 \in L(\mathcal{S}'_i)$ tal que $\bar{\sigma}_1|_{T_v} = \bar{\sigma}'_1|_{T_v}$, es decir, $\bar{\sigma}'_1 = \bar{\sigma}'t\bar{\sigma}'_2$ con $\bar{\sigma}' \in L(\mathcal{S}'_i)$ y $\bar{\sigma}'_2 \subseteq T_i \setminus T_v$. Por lo tanto, en \mathcal{S}'_i se tiene $\mathbf{m}_0^i \xrightarrow{\bar{\sigma}'_1} \mathbf{m}'_i$ tal que \mathbf{m}'_i sensibiliza a t . Sustituyendo en σ la subsecuencia $\bar{\sigma}$ por $\bar{\sigma}'$ como en el apartado 2, se obtiene $\sigma' \in L(\mathcal{S}')$ tal que $\mathbf{m}'_0 \xrightarrow{\sigma'} \mathbf{m}'$, es decir, $\mathbf{m}' \in R(\mathcal{S}')$ y t sensibilizada en \mathbf{m}' por lo que $\mathbf{m}'|_{\bullet_{T_v}} \neq \mathbf{0}$. Además, $\mathbf{m}|_{P''} = \mathbf{m}'|_{P''}$ por ser $\sigma|_{T''} = \sigma'|_{T''}$.

\supseteq) Se demuestra siguiendo la misma técnica que en el apartado \subseteq) pero intercambiando los papeles de \mathcal{S}_i y \mathcal{S}'_i . \diamond

Con este resultado queda desarrollada la técnica de reducción parcial de SM's en un DSSP y sus principales propiedades.

5.3 Reducción total de SM's

En la sección anterior se ha desarrollado una técnica de reducción de DSSP's basada en la reducción del grafo tangible de alcanzabilidad de sus SM's. Por lo tanto, la técnica estaba diseñada especialmente para DSSP's compuestos por SM's grandes, porque en ese caso se obtiene una reducción importante del espacio de estados global del DSSP. Pero si el DSSP está compuesto por un número grande de SM's pequeñas, la técnica de reducción anterior puede no ser suficiente para reducir de forma significativa el tamaño del grafo tangible de alcanzabilidad. En esta sección se va a desarrollar una técnica que puede permitir reducir en mayor grado ciertos DSSP's. Este trabajo fue desarrollado en [PJCS96a]. El objetivo de esta técnica consiste en estudiar bajo qué condiciones una SM de un DSSP puede reducirse totalmente a una única transición. Si se reducen ciertas SM's de un DSSP a una única transición y algunos canales son de origen y destino privado (es decir, conectan una única SM de origen con una única SM de destino) entonces se pueden obtener subredes con estructura de WTS. En ese caso se puede utilizar la técnica del capítulo 4 para reducir en mayor grado estas subredes.

Las propiedades que se van a preservar después de la agregación total de una SM en un DSSP son más débiles que las de la técnica de la sección anterior. En concreto, se va a preservar la vivacidad y limitación (y por

el teorema 2.101 la existencia de estados recurrentes) lo que asegura la obtención de CTMC's isomorfas con distribución en estado estacionario única (teorema 2.85).

Antes de exponer las condiciones necesarias para que una SM de un DSSP sea totalmente reducible a una transición hay que tener en cuenta algunas consideraciones sobre los canales del DSSP.

Propiedad 5.11 *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle B \cup (\bigcup_{i=1}^K P_i), \bigcup_{i=1}^K T_i, F, W \rangle$ un DSSP vivo y limitado, $b \in B$ un canal tal que $\bullet b \cup b \bullet \subseteq T_i$ para algún i tal que $1 \leq i \leq K$. Entonces b es un lugar implícito (ver definición 2.57).*

Demostración:

Sea $\mathcal{N}_i = \langle P_i, T_i, F_i \rangle$ la SM de \mathcal{N} a la que pertenecen las transiciones de entrada y salida de b . Sea \mathbf{y} el vector de dimensión el número de lugares de \mathcal{N} que toma valores 1 en sus componentes correspondientes a los lugares de $P_i \cup \{b\}$ y 0 en el resto. Es evidente que $\mathbf{y} \geq \mathbf{0}$. Además, $\mathbf{y} \cdot \mathbf{C} = \mathbf{l}_p$ ya que si \mathbf{l}_t es el vector de incidencia de una transición t cualquiera de \mathcal{N} se tienen los siguientes casos. Si $t \in T_j$ con $j \neq i$, entonces $(P_i \cup \{b\}) \cap \bullet t = \emptyset$ luego $\mathbf{y} \cdot \mathbf{l}_t = 0$. Si $t \in T_i$, entonces en P_i tiene un único lugar de entrada y uno solo de salida, por lo que \mathbf{l}_p tiene una componente 1 y otra -1 en lugares correspondientes a P_i . Por lo tanto, si $t \notin \bullet b \cup b \bullet$ entonces $\mathbf{y} \cdot \mathbf{l}_t = 0$, si $t \in \bullet b$ entonces $\mathbf{y} \cdot \mathbf{l}_t = W(t, b)$ y si $t \in b \bullet$ entonces $\mathbf{y} \cdot \mathbf{l}_t = W(b, t)$. Resumiendo, $\mathbf{y} \cdot \mathbf{C} = \mathbf{l}_p$. Por el teorema 2.59, b es un lugar estructuralmente implícito.

Si ahora se tiene en cuenta el marcado inicial \mathbf{m}_0 , se demostrará por reducción al absurdo que b es implícito. Si b no es implícito, existen $t \in b \bullet$ y $\mathbf{m} \in R(\mathcal{S})$ tales que $\mathbf{m}[b] < W(b, t)$ y $\mathbf{m}[p] \geq W(p, t)$ para cualquier $p \in \bullet t$ con $p \neq b$, en particular para el lugar $p = T_i \cap \bullet t$. Para que t pueda dispararse es necesario que $\mathbf{m}[b]$ aumente y para ello tiene que dispararse alguna de sus transiciones de entrada. Pero $\bullet b \subseteq T_i$, \mathcal{N}_i es 1-limitada y en \mathbf{m} la única marca de \mathcal{N}_i está en p , por lo que es necesario que se dispare t para que posteriormente $\mathbf{m}[b]$ aumente. Por lo tanto, \mathcal{S} no es vivo, lo que contradice la hipótesis inicial. \diamond

Por lo tanto, se puede considerar sin pérdida de generalidad que un DSSP vivo y limitado no tiene ningún canal con todas sus transiciones de entrada y salida en la misma SM del DSSP, ya que estos lugares son implícitos y por lo tanto se pueden eliminar sin modificar el funcionamiento del DSSP.

En general, una SM de un DSSP podrá reducirse totalmente a una única transición dependiendo de sus conexiones con los canales. Primero se explicará informalmente qué condiciones tiene que cumplir la SM para que pueda

ser reducida totalmente y luego se demostrará formalmente que una SM que cumpla el conjunto de reglas puede reducirse totalmente a una transición preservando en el DSSP las propiedades de vivacidad y limitación.

Al reducir una SM de un DSSP a una única transición t_s lo que se está haciendo es enmascarar el funcionamiento interno de la SM y observar únicamente su interfaz con el resto del DSSP, es decir, la variación que produce la SM en el marcado de los canales de entrada y salida. El disparo de la transición t_s en el DSSP reducido produce un cambio fijo en los canales de entrada y salida que dependen de los pesos de los arcos de entrada y salida de t_s . Como una SM puede tener internamente decisiones que la hagan evolucionar por distintos estados internos, es necesario que estas distintas evoluciones internas no se traduzcan en cambios diferentes en los canales. Esta propiedad tiene que ver con la posibilidad de calcular unos pesos para los arcos que conectan los canales de entrada y salida de SM con la transición t_s que se utilizará para resumirla. Si la SM puede evolucionar internamente de distintas formas y estas evoluciones producen distintos cambios de marcas en los canales de entrada y salida, no será posible calcular estos pesos y por lo tanto la SM no será reducible. Se tiene por lo tanto una condición necesaria que debe cumplir la SM en el DSSP y que depende de su estructura y de las conexiones de la SM con sus canales de entrada y salida.

Por otro lado, una SM de un DSSP puede tomar marcas de sus canales de entrada o depositar marcas en sus canales de salida sin importar el orden en que se hagan estas operaciones. Si se quiere resumir todo el comportamiento interno de la SM como el de una única transición, habrá que exigir que la SM tome todas las marcas de los canales de entrada antes de depositar las marcas en los canales de salida. Sólo en ese caso la evolución interna de la SM se asemeja al disparo de una transición en una red de Petri. Esta propiedad depende de la estructura de la SM, de sus conexiones con los canales y del marcado inicial de la SM en el DSSP.

Ahora se procederá a exponer formalmente estas propiedades.

Notación 5.12 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle B \cup (\bigcup_{i=1}^K P_i), \bigcup_{i=1}^K T_i, F, W \rangle$ un DSSP y $\mathcal{N}' = \langle P_i, T_i, F_i \rangle$ una SM de \mathcal{N} . Se denotará por:

$$i) B_{in} = \bullet T_i \cap B.$$

$$ii) B_{out} = T_i \bullet \cap B.$$

iii) $t_{\mathcal{SM}}$ la transición que reduce totalmente a \mathcal{N}' (cuando exista).

B_{in} es el conjunto de canales de entrada de la SM en cuestión y B_{out} el conjunto de canales de salida. Si la SM es totalmente reducible a una transición, se denotará por t_{SM} a esta transición.

Es evidente que, si \mathcal{N}' es totalmente reducible a una transición se debe cumplir que $\bullet t_{SM} = B_{in}$ y $t_{SM} \bullet = B_{out}$. El primer problema consiste en calcular los pesos de los arcos que unen los lugares de B_{in} con t_{SM} y t_{SM} con los lugares de B_{out} . Dependiendo de las conexiones de la SM original con sus canales de entrada y salida se podrán calcular o no estos pesos para los arcos que conectan la transición resumen t_{SM} con los canales de entrada y salida. Se puede caracterizar la existencia de estos pesos en función del rango de una submatriz de la matriz de incidencia del DSSP original.

Propiedad 5.13 *Sea \mathcal{S} un DSSP vivo y limitado con matriz de incidencia \mathbf{C} y $\mathcal{N}' = \langle P', T', F' \rangle$ una SM de \mathcal{N} con matriz de incidencia \mathbf{C}' . Sea \mathbf{C}^* la submatriz de \mathbf{C} formada por las filas de $P' \cup B_{in} \cup B_{out}$ y las columnas de T' . Sea $b_i \in B_{in}$ y $\mathbf{C}^*[b_i]$ su vector de incidencia restringido a T' . Entonces $\text{rank}(\mathbf{C}^*) = |P'|$ si y sólo si para todo $b_j \in B_{in} \cup B_{out}$ existe un único $k_{ij} \in \mathbb{Q}$ tal que para cualquier $\mathbf{x} \geq \mathbf{0}$ con $\mathbf{C}' \cdot \mathbf{x} = \mathbf{0}$ se cumple $\mathbf{C}^*[b_j] \cdot \mathbf{x} = k_{ij}(\mathbf{C}^*[b_i] \cdot \mathbf{x})$. Si $\text{rank}(\mathbf{C}^*) = |P'|$ se dirá que \mathcal{N}' cumple la propiedad **P1**.*

Demostración:

La condición $\forall \mathbf{x} \geq \mathbf{0}$ tal que $\mathbf{C}' \cdot \mathbf{x} = \mathbf{0}$ es equivalente en este caso a $\forall \mathbf{x}$ tal que $\mathbf{C}' \cdot \mathbf{x} = \mathbf{0}$ (sin exigir $\mathbf{x} \geq \mathbf{0}$), ya que por ser $\langle \mathcal{N}', \mathbf{m}'_0 \rangle$ vivo y limitado, también es consistente (teorema 2.50) y por lo tanto existe $\mathbf{y} \geq \mathbf{1}$ tal que $\mathbf{C}' \cdot \mathbf{y} = \mathbf{0}$. Ahora si \mathbf{x} tiene componentes negativas, existe $k > 0$ tal que $\mathbf{x} + k\mathbf{y} \geq \mathbf{0}$. Como $\mathbf{C}' \cdot (\mathbf{x} + k\mathbf{y}) = \mathbf{C}' \cdot \mathbf{x} + k\mathbf{C}' \cdot \mathbf{y} = \mathbf{C}' \cdot \mathbf{x}$ entonces $\mathbf{C}' \cdot \mathbf{x} = \mathbf{0}$ si y sólo si $\mathbf{C}' \cdot (\mathbf{x} + k\mathbf{y}) = \mathbf{0}$, es decir, $\mathbf{C}' \cdot \mathbf{x} = \mathbf{0}$ si y sólo si existe $\mathbf{x}' \geq \mathbf{0}$ tal que $\mathbf{C}' \cdot \mathbf{x}' = \mathbf{0}$. Se ha mantenido la condición $\mathbf{x} \geq \mathbf{0}$ por ser más débil aparentemente. Para la demostración no se tendrá en cuenta. Por otro lado, se considerarán en esta demostración todos los números como racionales. Es suficiente por ser $(\mathbb{Q}, +, \cdot)$ un cuerpo.

Por el apartado 1 del teorema 2.96, se tiene que $\text{rank}(\mathbf{C}') = |P'| - 1$. Sea \mathbf{C}_1 la matriz que resulta de añadir a \mathbf{C}' el vector fila $\mathbf{C}^*[b_i]$. Se demostrará que $\mathbf{C}^*[b_i]$ no es combinación lineal de las filas de \mathbf{C}' .

Si $\mathbf{C}^*[b_i]$ es combinación lineal de las filas de \mathbf{C}' , entonces existe un vector \mathbf{y} tal que $\mathbf{C}^*[b_i] = \mathbf{y} \cdot \mathbf{C}'$. Ahora, por ser \mathbf{C}' la matriz de incidencia de una SM, $\mathbf{1} \cdot \mathbf{C}' = \mathbf{0}$, luego se puede suponer que $\mathbf{y} \geq \mathbf{0}$ (basta con sumar $\mathbf{1}$ a \mathbf{y}

tantas veces como el valor absoluto de su menor componente). En estas condiciones, por el teorema 2.59, b_i es estructuralmente implícito en \mathcal{N}' y por lo tanto debe ser un canal de entrada y salida lo que es absurdo (por la propiedad 5.11 se eliminaban estos canales). Por lo tanto, $\mathbf{C}^*[b_i]$ no es combinación lineal de las filas de \mathbf{C}' y entonces $\text{rank}(\mathbf{C}_1) = |P'|$.

Para simplificar notación, se supondrá a partir de ahora que \mathcal{N}' tiene n lugares $\{p_j\}_{j=1}^n$ con $n > 1$ (si $n = 1$, \mathcal{N}' tiene sólo un lugar y una transición y por lo tanto ya está totalmente reducida).

\Rightarrow) Si $b_j = b_i$, el resultado es evidente tomando $k_{ij} = 1$, por lo que se supondrá que $b_j \neq b_i$.

Si $\text{rank}(\mathbf{C}^*) = |P'| = n$, entonces $\text{rank}(\mathbf{C}^*) = \text{rank}(\mathbf{C}_1)$. Sea $b_j \neq b_i$ un canal cualquiera de entrada o salida de \mathcal{N}' . Entonces el vector $\mathbf{C}^*[b_j]$ es combinación lineal de los vectores fila de \mathbf{C}_1 , es decir, existen racionales $\{\lambda_k\}_{k=1}^n, k_{ij}$ tales que $\mathbf{C}^*[b_j] = k_{ij}\mathbf{C}^*[b_i] + \sum_{k=1}^n \lambda_k \mathbf{C}^*[p_k]$. Multiplicando por la derecha por cualquier vector \mathbf{x} tal que $\mathbf{C}' \cdot \mathbf{x} = \mathbf{0}$ se obtiene $\mathbf{C}^*[b_j] \cdot \mathbf{x} = k_{ij}\mathbf{C}^*[b_i] \cdot \mathbf{x} + \sum_{k=1}^n \lambda_k \mathbf{C}^*[p_k] \cdot \mathbf{x} = k_{ij}\mathbf{C}^*[b_i] \cdot \mathbf{x}$.

Falta demostrar la unicidad de k_{ij} .

Suponer $\mathbf{C}^*[b_j] = k_{ij}\mathbf{C}^*[b_i] + \sum_{k=1}^n \lambda_k \mathbf{C}^*[p_k] = k'_{ij}\mathbf{C}^*[b_i] + \sum_{k=1}^n \lambda'_k \mathbf{C}^*[p_k]$. Como $\mathbf{1} \cdot \mathbf{C}' = \mathbf{0}$, $\sum_{k=1}^n \mathbf{C}^*[p_k] = \mathbf{0}$, luego $\mathbf{C}^*[p_1] = -\sum_{k=2}^n \mathbf{C}^*[p_k]$. Sustituyendo $\mathbf{C}^*[p_1]$ en la ecuación, $\mathbf{C}^*[b_j] = k_{ij}\mathbf{C}^*[b_i] + \sum_{k=2}^n (\lambda_k - \lambda_1)\mathbf{C}^*[p_k] = k'_{ij}\mathbf{C}^*[b_i] + \sum_{k=2}^n (\lambda'_k - \lambda'_1)\mathbf{C}^*[p_k]$. Ahora, la familia $\{\{\mathbf{C}^*[p_k]\}_{k=2}^n, \mathbf{C}^*[b_i]\}$ es linealmente independiente ($\{\mathbf{C}^*[p_k]\}_{k=2}^n$ son linealmente independientes por ser $\text{rank}(\mathbf{C}') = n - 1$ y $\mathbf{C}^*[b_i]$ es linealmente independiente de la subfamilia anterior como se ha demostrado), por lo que las dos combinaciones lineales de $\mathbf{C}^*[b_j]$ deben ser la misma, es decir, $k_{ij} = k'_{ij}$.

\Leftarrow) Se tiene demostrado en el comienzo que $\text{rank}(\mathbf{C}_1) = |P'|$. Sea ahora $b_j \in B_{in} \cup B_{out}$. Para demostrar el resultado basta probar que $\mathbf{C}^*[b_j]$ es combinación lineal de las filas de \mathbf{C}_1 . Por hipótesis, existe $k_{ij} \in \mathbb{Q}$ tal que $\mathbf{C}^*[b_j] \cdot \mathbf{x} = k_{ij}\mathbf{C}^*[b_i] \cdot \mathbf{x}$ para todo \mathbf{x} tal que $\mathbf{C}' \cdot \mathbf{x} = \mathbf{0}$. Sean p y q enteros con $q \neq 0$ tales que $k_{ij} = p/q$ y $\mathbf{v} = q\mathbf{C}^*[b_j] - p\mathbf{C}^*[b_i]$. Se tiene que $\mathbf{v} \cdot \mathbf{x} = 0$ para todo \mathbf{x} tal que $\mathbf{C}' \cdot \mathbf{x} = \mathbf{0}$. Considerando \mathbf{C}' como la matriz de una aplicación lineal $f : \mathbb{Q}^m \rightarrow \mathbb{Q}^n$ ($m = |T'|$) se tiene que $\text{rank}(f) = \dim(\text{Im}f) = \text{rank}(\mathbf{C}') = n - 1$, luego por el corolario 2.15 $\dim(\text{Ker}f) = m - (n - 1) = m - n + 1$. Por lo tanto, por el teorema 2.13 $\text{Im}f \cong \mathbb{Q}^{(n-1)}$ y $\text{Ker}f \cong \mathbb{Q}^{(m-n+1)}$. Además, las filas de \mathbf{C}' constituyen una familia generadora de $\text{Im}f$ (eliminando cualquier fila se obtiene una base). Como por el teorema 2.14 $\mathbb{Q}^m \cong \text{Ker}f \oplus \text{Im}f$ (aquí \oplus representa la suma directa de subespacios vectoriales) y $\mathbf{v} \in \mathbb{Q}^m$ es un vec-

tor tal que $\mathbf{v} \cdot \mathbf{x} = 0$ para todo $\mathbf{x} \in \text{Ker}f$ entonces $\mathbf{v} \in \text{Im}f$. Como las filas de \mathbf{C}' son una familia generadora de $\text{Im}f$, existen $\{\lambda_k\}_{k=1}^n \in \mathbb{Q}$ tales que $\mathbf{v} = \sum_{k=1}^n \lambda_k \mathbf{C}^*[p_k]$. Como también $\mathbf{v} = q\mathbf{C}^*[b_j] - p\mathbf{C}^*[b_i]$, entonces $\mathbf{C}^*[b_j] = p/q\mathbf{C}^*[b_i] + \sum_{k=1}^n \lambda_k/q\mathbf{C}^*[p_k] = k_{ij}\mathbf{C}^*[b_i] + \sum_{k=1}^n (\lambda_k/q)\mathbf{C}^*[p_k]$, es decir, $\mathbf{C}^*[b_j]$ es combinación lineal de las filas de \mathbf{C}_1 . \diamond

Esta propiedad da una caracterización algebraica en términos del rango de la matriz \mathbf{C}^* (una submatriz de la matriz \mathbf{C} de incidencia del DSSP) para la existencia de los pesos de los arcos que conectan la transición reducción t_{SM} de una SM con los canales de entrada y salida de la SM original. La condición que debe cumplir una SM para poderse reducir a una única transición es que todos sus t-semiflujos (ciclos en una SM) produzcan el mismo efecto (o efecto nulo) en el conjunto de canales de entrada y salida de la SM. Y esta propiedad puede comprobarse en tiempo polinomial calculando el rango de la matriz \mathbf{C}^* . Si se cumple la propiedad **P1**, es posible calcular, para cada canal b_j de entrada o salida de la SM, un único número racional k_{ij} que es el cociente entre el peso del arco correspondiente al canal b_j y el de b_i en t_{SM} . Como el canal de referencia b_i se ha tomado de entrada, si $k_{ij} > 0$ es porque b_j también es de entrada y si $k_{ij} < 0$ es porque b_j es de salida. De esta forma no se conocen exactamente el valor de todos los pesos de los arcos, pero si su valor relativo respecto al peso del arco del canal b_i , es decir, se tiene que:

- $W(b_i, t_{SM}) = \alpha$
- $W(b_j, t_{SM}) = \alpha k_{ij}$ para todo $b_j \in B_{in} \setminus \{b_i\}$.
- $W(t_{SM}, b_j) = -\alpha k_{ij}$ para todo $b_j \in B_{out}$.

Por lo tanto, se tiene una condición necesaria que debe cumplir toda SM en un DSSP para poderse reducir a una única transición. Pero esta condición no es suficiente y, en el caso de que la SM sea finalmente reducible, tampoco permite el cálculo exacto de los pesos de los arcos de entrada y salida de t_{SM} . Falta calcular el factor α que no es más que el peso del arco correspondiente al canal que se toma como referencia. Se procederá ahora a completar la técnica.

Para que el funcionamiento interno de una SM en un DSSP se asemeje al disparo de una única transición es necesario que todas las marcas de los canales de entrada a la SM entren en la misma antes de que se produzca la primera salida de marcas hacia los canales de salida. Esta condición afecta

no sólo a las conexiones de los canales con la SM sino también al orden en el que éstas ocurren. En los casos en los que la evolución interna de una SM en un DSSP alterne la entrada de marcas de los canales de entrada y la salida de marcas a los canales de salida no se podrá realizar la reducción ya que sería posible perder la vivacidad de todo el sistema. Para evitar esos potenciales problemas, en esos casos no se efectuará la reducción.

Para detectar estos casos se traducirá la estructura de una SM en un DSSP a un grafo dirigido etiquetado y se utilizarán algoritmos de recorrido de grafos para distinguir los casos agregables de los que no lo son.

Definición 5.14 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un DSSP de matriz de incidencia \mathbf{C} y $\mathcal{N}' = \langle P', T', F' \rangle$ una SM de \mathcal{N} . Se define el grafo dirigido etiquetado asociado a \mathcal{N}' como la terna $G = (V, L, A)$ donde $V = P'$ es el conjunto de vértices, $L = \{\text{columnas de } \mathbf{C}[B_{in} \cup B_{out}, T']\}$ es el conjunto de etiquetas y $A \subseteq V \times V \times L$ es el conjunto de aristas etiquetadas tales que $(p_1, p_2, l) \in A$ si y sólo si existe $t \in T'$ tal que $\bullet t = p_1$ y $t \bullet = p_2$. La etiqueta l asociada a la arista es el vector $\mathbf{C}[B_{in} \cup B_{out}, t] \in L$. Además, los lugares de salida de transiciones $t \in T'$ con canales de salida se dicen bloqueados.

Una SM puede verse como un grafo dirigido con un vértice por cada lugar de la SM y una arista (v_1, v_2) por cada transición que conecta el lugar correspondiente a v_1 con el lugar correspondiente a v_2 . En la definición anterior se han añadido etiquetas a las aristas para incluir en el grafo las conexiones de la SM con los canales del DSSP. De esta forma G contiene toda la información de la estructura de la SM y su conexión con el DSSP.

Propiedad 5.15 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un DSSP vivo y limitado, \mathcal{N}' una SM de \mathcal{N} inicialmente marcada en p_q y G su grafo dirigido etiquetado asociado. Se dirá que \mathcal{N}' tiene la propiedad **P2** si su grafo dirigido etiquetado asociado G cumple:

1. Partiendo de p_q es posible alcanzar todas las aristas cuyas etiquetas contienen elementos estrictamente negativos sin pasar por ningún lugar bloqueado.
2. La suma vectorial de las etiquetas de las aristas de cualquier camino simple (definición 2.41) de p_q a un lugar bloqueado es la misma.

En este caso el valor del parámetro α es la componente correspondiente a b_i de la suma vectorial de etiquetas de cualquier camino simple de p_q a un lugar bloqueado.

Es posible comprobar en tiempo polinomial si una SM en un DSSP cumple o no la propiedad **P2**. Para ver si su grafo dirigido etiquetado asociado cumple el apartado 1, basta con hacer un recorrido en anchura partiendo del vértice p_q y parando en los vértices bloqueados. Al acabar el recorrido no puede quedar sin recorrer ninguna arista cuya etiqueta tenga componentes negativas. Si ocurre esto, quiere decir que la SM toma marcas de algún canal de entrada después de poner marcas en algún canal de salida y por lo tanto no es reducible. La complejidad del algoritmo de recorrido en anchura es $O(\max\{|P'|, |T'|\})$.

Por lo que respecta al apartado 2, puede hacerse con una ligera modificación del algoritmo de Dijkstra [AHU83] para el cálculo en un grafo dirigido de los caminos de peso mínimo de un vértice a todos los demás. La modificación consiste en la función de coste. En el algoritmo de Dijkstra los pesos de los arcos del grafo son números naturales y las operaciones que se utilizan son la suma y el min de números naturales. En este caso, las etiquetas son vectores de enteros. La operación de suma de naturales se puede sustituir en este caso por la suma vectorial y la operación min por la de min componente a componente. Esta última sustitución podría dar problemas debido a los números negativos. Estos aparecen con los canales de entrada y como lo único que importa es el número de marcas que se toma de cada uno en un camino dado, a todos los efectos se comportan como los números naturales, es decir, la función de coste para ellos será el max (o el min de sus valores absolutos). La complejidad del algoritmo de Dijkstra es $O(|P'|^2)$.

Ya se tienen las propiedades que debe cumplir una SM en un DSSP para poderse reducir a una única transición.

Definición 5.16 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un DSSP vivo y limitado y \mathcal{N}' una SM de \mathcal{N} que cumple las propiedades **P1** y **P2** enunciadas en las propiedades 5.13 y 5.15. Entonces \mathcal{N}' puede reducirse a una única transición $t_{\mathcal{SM}}$ de forma que $\bullet t_{\mathcal{SM}} = B_{in}$, $t_{\mathcal{SM}} \bullet = B_{out}$, $W(b_j, t_{\mathcal{SM}}) = k_{ij}\alpha$ para todo $b_j \in B_{in}$ y $W(t_{\mathcal{SM}}, b_j) = k_{ij}\alpha$ para todo $b_j \in B_{out}$, donde α y k_{ij} son los valores calculados en las propiedades 5.15 y 5.13 respectivamente.

Queda por aclarar alguna cuestión relativa al marcado inicial de una SM en un DSSP. De acuerdo con el enunciado de la propiedad **P2**, una SM en un DSSP puede ser reducible o no dependiendo de su marcado inicial. Sin embargo, es posible que una SM tenga un comportamiento reducible a una transición (que tome todas las marcas de los canales de entrada antes de

poner las marcas en los canales de salida) y que no cumpla la propiedad **P2** simplemente porque el marcado inicial no es el adecuado. Por ello, si una SM cumple la propiedad **P1**, es conveniente observar su marcado inicial antes de comprobar si cumple **P2** o no. Hay dos casos particulares que se pueden arreglar.

- En el marcado inicial la SM ya ha tomado todas las marcas de todos los canales de entrada pero no ha efectuado todas las salidas de marcas a los canales de salida.
- En el marcado inicial la SM ha tomado alguna marca de algún canal de entrada, pero no todas.

En estas dos situaciones se tiene que la SM almacena en su estado interno marcas que evolucionan normalmente por los canales del DSSP.

En el primer caso hay que disparar transiciones de la SM hasta que realice todas las salidas de marcas. De esta forma quedan todas las marcas en los canales de salida y la SM en un estado interno en el que no ha comenzado de nuevo a tomar marcas de los canales de entrada. Si la SM es tal que la última transición de un ciclo con canales de salida también tiene canales de entrada, entonces la transición se puede desplegar en dos conectadas por un lugar tales que la primera tenga los canales de salida y la segunda los de entrada.

En el segundo caso es posible disparar hacia atrás las transiciones que han tomado las marcas de los canales de entrada, hasta que todas estas marcas estén en sus respectivos canales de entrada. De esta forma se deja igualmente el estado interno de la SM en un punto de sus ciclos en el que no se han tomado marcas de los canales de entrada. El hecho de disparar hacia atrás transiciones no es problema en un DSSP vivo y limitado, porque el nuevo sistema que se obtenga seguirá teniendo estados recurrentes y por lo tanto el mismo comportamiento en estado estacionario.

El último paso que queda consiste en demostrar las propiedades que se preservan al reducir totalmente una SM en un DSSP a una transición.

Teorema 5.17 *Sea \mathcal{S} un DSSP compuesto por K SM's. Sea \mathcal{N}_i una SM de \mathcal{N} reducible a una transición y \mathcal{S}' el DSSP que resulta de reducir \mathcal{N}_i a una única transición. Entonces \mathcal{S}' es conservativo, vivo y tiene estados recurrentes si y sólo si lo es/tiene \mathcal{S} . Además los lugares de \mathcal{S}' tienen las mismas cotas de marcado que los de \mathcal{S} .*

Demostración:

La reducción total de SM's se puede ver como un caso particular de la eliminación de transiciones con el algoritmo de cálculo de p-semiflujos de [CS91]. Por lo tanto se preserva la conservatividad y también la limitación. De acuerdo con las condiciones de reducción, una SM es reducible si cualquier t-semiflujo de la SM (un ciclo) con efecto externo sobre los canales toma todas las marcas de los canales de entrada (puede ser que en distinto orden dependiendo del ciclo) antes de depositar todas las marcas en los canales de salida (propiedad **P2**) y la cantidad de marcas que toma de cada canal de entrada y deposita en cada canal de salida es la misma independientemente del ciclo ejecutado (propiedad **P1**). El proceso de reducción puede hacerse en dos pasos. El primero consiste en reducir la SM a dos lugares y dos transiciones en ciclo. El primer lugar se corresponde con el inicialmente marcado, la primera transición t_1 resume todas las aportaciones de marcas a cualquier ciclo por parte de los canales de entrada y esta transición tiene un lugar de salida p_{12} que la conecta con otra transición t_2 que resume todas las aportaciones de marcas de cualquier ciclo a los canales de salida. Obviamente estas dos reducciones preservan la vivacidad, k-limitación (ver figura 2.1) y existencia de estados recurrentes. Posteriormente este ciclo de dos transiciones se reduce a una única transición t_{12} eliminando el lugar que conecta las transiciones t_1 y t_2 . Esta reducción también preserva la vivacidad y existencia de estados recurrentes.

Queda demostrar que se mantienen las cotas de marcado de los lugares preservados en la reducción. Esto es claro en los lugares internos de las SM's no reducidas, cuya cota de marcado es 1. Y con los canales también es claro que se preserva teniendo en cuenta la doble reducción del párrafo anterior. Antes de depositar marcas en los canales de salida es necesario realizar todas las aportaciones de los canales de entrada y estos canales tienen destino privado por lo que su cota de marcado no cambia. \diamond

Esta técnica de reducción total de SM's no mantiene exactamente ni las proyecciones de los marcados alcanzables ni las proyecciones de las secuencias de disparo del DSSP original sobre los nodos preservados. Se demostrará con un ejemplo.

En la figura 5.2.a se tiene un DSSP muy sencillo formado por dos SM, cada una en el interior de una zona sombreada. La primera SM es una única transición para simplificar. La segunda está formada por 6 transiciones y 5 lugares. El DSSP tiene 3 canales (B_1 , B_2 y B_3) conectando entre sí las

Aplicando la propiedad 5.13 y tomando como referencia el canal B_1 , los pesos de los arcos correspondientes a cada canal en la transición reducida serán α para B_1 y B_3 (B_3 es canal de salida por lo que se cambia el signo del coeficiente) y $3\alpha/2$ para B_2 .

Para calcular el valor de α hay que aplicar la propiedad 5.15. En este caso, el lugar P_1 es de salida de la transición T_6 que tiene canales de salida, por lo que está bloqueado. Desde el lugar P_3 (marcado inicial de la SM) no es posible alcanzar la transición T_3 sin pasar por el lugar bloqueado P_1 por lo que en principio la SM no cumple la propiedad **P2**. Sin embargo, éste es uno de los casos en los que es posible modificar el marcado inicial de la SM para que si cumpla la propiedad **P2**. En efecto, si se dispara hacia atrás la transición T_2 se obtiene otro marcado alcanzable del DSSP y considerando este marcado inicial la SM cumple la propiedad **P2**, por lo que si es reducible. Además, desde P_1 se puede alcanzar el único lugar bloqueado (P_1 de nuevo) por dos caminos diferentes y los dos toman 2 marcas del canal B_1 . Por lo tanto, el valor de α es 2 y los pesos de los arcos de la transición reducida serán 2 para B_1 y B_3 y 3 para B_2 .

El resultado de la reducción de esta SM puede verse en la figura 5.2.b. Queda por último ver que en este caso no se mantienen las proyecciones de los marcados alcanzables ni de las secuencias de disparo del DSSP original en los nodos preservados en la reducción. En efecto, si en el DSSP original se dispara la secuencia formada por T_4 y T_5 se quitan primero dos marcas al canal B_1 y después otras dos a B_2 . Estos dos estados no se pueden distinguir en el DSSP reducido, ya que al reducir la SM a una única transición se producen todos los cambios de marcas en los canales con un único disparo de una sola transición. Este hecho supone la agregación de ciertos estados del DSSP original y por lo tanto no se mantienen las proyecciones de estados alcanzables ni de secuencias de disparo.

Con este ejemplo queda desarrollada la técnica de reducción total de SM's en un DSSP y sus principales propiedades.

5.4 Descomposición estructural de DSSP's

Una vez desarrolladas las dos técnicas de reducción de SM's de las secciones anteriores, se va a desarrollar la técnica estructural de descomposición. La descomposición de DSSP's sigue la misma filosofía que las de MG's y WTS's de los capítulos anteriores. Un DSSP vivo y limitado se descompone

en K subredes por medio de un corte Q definido sobre los canales de forma que cada SM del DSSP esté totalmente contenida en una subred. A partir del corte y de las subredes se construyen K sistemas de bajo nivel \mathcal{LS}_i con $1 \leq i \leq K$ y un sistema de alto nivel o esqueleto básico \mathcal{BS} . Cada \mathcal{LS}_i contiene completamente una de las subredes producto del corte y agregaciones del resto de las subredes. El \mathcal{BS} se compone de las agregaciones de todas las subredes.

La forma de agregar subredes en DSSP's se puede hacer de diversas maneras. Una primera posibilidad consiste en reducir parcialmente las SM's de cada subred por medio de la técnica de reducción parcial de SM's de la sección 5.2. Otra posibilidad consiste en aplicar la técnica de reducción total de SM's de la sección 5.3 a las SM's del DSSP susceptibles de ser reducidas totalmente (la que cumplan las propiedades **P1** y **P2**). Por último, una vez reducidas total o parcialmente las SM's se puede aplicar la técnica de reducción de WTS's del capítulo anterior a ciertas partes de las subredes que tengan estructura de WTS. La forma de agregación de las subredes dependerá básicamente del tamaño del espacio de estados del DSSP original. Cuanto mayor sea este espacio de estados, mayor reducción se querrá alcanzar y habrá que intentar aplicar la máxima agregación posible. El mecanismo para obtener la máxima reducción posible consiste en detectar primero las SM's que pueden ser agregadas totalmente y reducirlas a una única transición, posteriormente reducir parcialmente las SM's que queden y por último agregar las subredes con estructura de WTS.

La técnica de reducción de DSSP's es una extensión de la de WTS's, y por lo tanto de la de MG's, en el sentido de que la técnica de agregación que se va a exponer en esta sección aplicada a un WTS o un MG da exactamente los mismos sistemas agregados que los que se conseguían con las técnicas de los capítulos 3 y 4 respectivamente.

Las propiedades del DSSP original que se preservan en los sistemas agregados dependen de las técnicas de reducción de subredes empleadas. Si sólo se reducen parcialmente SM's de acuerdo con la técnica de la sección 5.2, los sistemas agregados que se consiguen son, al igual que en el caso de MG's, proyecciones exactas en cuanto a marcados alcanzables y secuencias de disparo del DSSP original sobre los nodos preservados en cada sistema agregado (teorema 5.10). En cuanto se haga una agregación total de una SM o se reduzca alguna parte de una subred con estructura de WTS, ya no se puede asegurar que se mantengan las proyecciones de marcados alcanzables ni de secuencias de disparo.

Definición 5.18 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle P, T, F, W \rangle$ y $P = B \cup (\bigcup_{i=1}^q PS_i)$, $T = \bigcup_{i=1}^q TS_i$ un DSSP compuesto por q SM's. Un subconjunto de lugares $Q \subseteq B$ se dice K -corte ($K \geq 2$) de \mathcal{N} si existen K subredes \mathcal{N}_i con $\mathcal{N}_i = \langle P_i, T_i, F_i, W_i \rangle$ para $1 \leq i \leq K$ de \mathcal{N} tales que:

- i) $\bigcup_{i=1}^K T_i = T$ y $T_i \cap T_j = \emptyset$ para $1 \leq i, j \leq K, i \neq j$.
- ii) Para cada $1 \leq i \leq q$ existe $1 \leq j \leq K$ tal que $TS_i \subseteq T_j$.
- iii) $P_i = \bullet T_i \cup T_i \bullet$ para $1 \leq i \leq K$.
- iv) $\bigcup_{i=1}^K P_i = P$ y $\bigcup_{i \neq j} (P_i \cap P_j) = B$ para $1 \leq i, j \leq K$.
- v) $F_i = F \cap ((P_i \times T_i) \cup (T_i \times P_i))$ y $W_i = W|_{F_i}$ para $1 \leq i \leq K$.

Las transiciones del conjunto $TI = \bullet B \cup B \bullet$ se llaman transiciones de interfaz. El resto se llaman transiciones internas.

La única diferencia entre la definición de corte en DSSP's y las equivalentes para MG's y WTS's está en el punto *ii*, en el que se exige que cada SM del DSSP esté totalmente contenida en una única subred producto del corte. Este punto se añade para facilitar las operaciones de agregación.

Una vez definido un corte sobre un DSSP hay que calcular los \mathcal{LS}_i y el \mathcal{BS} . Para realizar esta tarea es necesario reducir cada subred producto del corte a un conjunto de nodos (lugares y transiciones) que resumen el comportamiento interno de la subred. El conjunto de nodos que resume el comportamiento de una determinada subred depende de las técnicas de reducción que se apliquen (reducción parcial o total de SM's y reducción de partes con estructura de WTS). En cada subred se pretende mantener las transiciones de interfaz, con el fin de que en la parte numérica de la técnica de aproximación sean éstas las transiciones que resuman la evolución temporal de la subred (en concreto el tiempo de respuesta).

Definición 5.19 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un DSSP vivo y limitado, Q un K -corte de \mathcal{N} y $\mathcal{N}_i = \langle P_i, T_i, F_i, W_i \rangle$ con $1 \leq i \leq K$ las subredes producto de Q . Se utilizarán las siguientes notaciones:

$$H_i = \{\text{lugares } p \text{ necesarios para reducir } \mathcal{N}_i \text{ con } \mathbf{m}_0[p] \text{ y } \mathbf{l}_p\}$$

$$I_i = \{\text{transiciones inmediatas } t \text{ necesarias para reducir } \mathcal{N}_i \text{ con } \mathbf{l}_t\}$$

Se denota por $H = \bigcup_{i=1}^K H_i$ y por $I = \bigcup_{i=1}^K I_i$.

El contenido de los conjuntos H e I depende de las técnicas de reducción empleadas, pudiendo ser I el conjunto vacío. Ahora se pueden construir los sistemas agregados. Como paso previo se construirá el sistema extendido.

Definición 5.20 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un DSSP vivo y limitado y $B \subseteq P$ un K -corte de \mathcal{N} . El sistema extendido $\mathcal{ES} = \langle \mathcal{EN}, \mathbf{m}_0^{\mathcal{EN}} \rangle$ de \mathcal{S} es el sistema resultante de añadir a \mathcal{S} los nodos de los conjuntos H e I con sus arcos de entrada y salida y marcado inicial.

Definición 5.21 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle P, T, F, W \rangle$ un DSSP vivo y limitado, $B \subseteq P$ un K -corte de \mathcal{N} y \mathcal{ES} su sistema extendido.

- i) El sistema de bajo nivel $\mathcal{LS}_i = \langle \mathcal{LN}_i, \mathbf{m}_0^i \rangle$ con $1 \leq i \leq K$ es el sistema que se obtiene eliminando de \mathcal{ES} los lugares $\bigcup_{j \neq i} P_j \setminus B$ y las transiciones $\bigcup_{j \neq i} T_j \setminus \text{TI}$ y sus arcos adyacentes.
- ii) El sistema de alto nivel (o esqueleto básico) $\mathcal{BS} = \langle \mathcal{BN}, \mathbf{m}_0^{\mathcal{BN}} \rangle$ es el sistema que se obtiene eliminando de \mathcal{ES} los lugares $\bigcup_{i=1}^K P_i \setminus B$ y las transiciones $\bigcup_{i=1}^K T_i \setminus \text{TI}$ y sus arcos adyacentes.

En cada \mathcal{LS}_i , todas las subredes \mathcal{N}_j con $j \neq i$ se reducen a sus transiciones de interfaz y a los nodos de los conjuntos H_j e I_j (ver definición 5.19), mientras que la subred \mathcal{N}_i se preserva completamente. En el \mathcal{BS} se reducen todas las subredes.

El siguiente teorema demuestra las propiedades que tienen los sistemas agregados recién construidos.

Teorema 5.22 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un DSSP vivo y limitado, $B \subseteq P$ un K -corte de \mathcal{N} . Entonces, \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} son DSSP's vivos y limitados.

Demostración:

Por el teorema 5.10, la reducción parcial de una SM en un DSSP mantiene exactamente las proyecciones de estados alcanzables y secuencias de disparo sobre los nodos preservados en la reducción. En particular, preserva la vivacidad y limitación. Por el teorema 5.17, la reducción total de SM's en un DSSP susceptibles de ser agregadas preserva la vivacidad y conservatividad, por lo tanto también limitación. Y por el teorema 4.37, la reducción de subredes con estructura de WTS de acuerdo con la técnica de reducción

desarrollada en el capítulo 4 preserva la vivacidad y limitación. Por lo tanto, cualquier técnica de reducción de subredes en DSSP's que utilice cualquiera de estas tres técnicas de reducción preserva vivacidad y limitación. \diamond

Notar que aunque las únicas propiedades que se preservan con cualquier técnica de reducción de DSSP's de las expuestas son vivacidad y limitación (y por el teorema 2.101 también la existencia de estados recurrentes), si en la reducción de subredes sólo se emplea la técnica de reducción parcial de SM's de la sección 5.2, entonces se puede asegurar que además se mantienen exactamente las proyecciones de los marcados alcanzables y de las secuencias de disparo del DSSP original sobre los nodos preservados en cada sistema agregado cuando tengan sentido estas proyecciones (teorema 5.10).

A modo de resumen, en esta sección se ha desarrollado una técnica estructural eficiente que permite descomponer un DSSP vivo y limitado en un número finito arbitrario K de subredes y a partir de ellas construir K sistemas de bajo nivel (en los que se mantiene una de las subredes y se reducen el resto) y un sistema de alto nivel o esqueleto básico (en el que se reducen todas las subredes). Estos $K + 1$ sistemas agregados son también DSSP's vivos y limitados (por lo tanto con estados reversibles), por lo que sus cadenas de Markov isomorfas poseen distribución en estado estacionario única (teorema 2.85).

En la siguiente sección se emplearán estos sistemas agregados para aproximar el throughput de las transiciones del DSSP original.

5.5 Aproximación iterativa del throughput

El contenido de esta sección es similar al de las secciones 3.4 y 4.4. Se incluye aquí con las adaptaciones necesarias para DSSP's con objeto de aumentar la legibilidad de la técnica completa de aproximación.

En la sección anterior se ha desarrollado una técnica para descomponer un DSSP vivo y limitado en K subredes, a partir de las cuales se construyen $K + 1$ sistemas agregados (\mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS}). Cada \mathcal{LS}_i contiene una de las subredes producto del corte, los lugares del corte, las transiciones de interfaz y un conjunto de nodos que resumen el comportamiento del resto de subredes. El \mathcal{BS} contiene los lugares del corte, las transiciones de interfaz y los nodos que resumen el comportamiento de todas las subredes. Ahora se van a emplear estos $K + 1$ sistemas agregados para aproximar el throughput de las transiciones del DSSP original.

Por el teorema 5.9, el vector \mathbf{v} de ratios de visita de un DSSP viene determinado por su estructura. Entonces, basta aproximar el throughput de una transición del DSSP y por medio de \mathbf{v} obtener aproximaciones para el throughput de las demás con el mismo porcentaje de error.

La técnica que se va a exponer en esta sección es básicamente la misma que la del capítulo 3 para MG's. Esta técnica es un método de aproximación del tiempo de respuesta. Las transiciones de interfaz de cada subred reducida en cada sistema agregado aproximarán el tiempo de respuesta de toda la subred. La técnica es iterativa y utiliza un sistema de alto nivel o esqueleto básico para no tener problemas de convergencia y de malas aproximaciones (como en MG's). El algoritmo utilizado es el siguiente:

Algoritmo 5.23

input: $\langle \mathcal{S}, w \rangle$ DSSP estocástico vivo y limitado
seleccionar un K -corte $B \subseteq P$ de \mathcal{N}
construir \mathcal{LS}_i y \mathcal{BS} para $1 \leq i \leq K$
seleccionar tasas iniciales $\mu_t^{(0)}$ para cada $t \in T_j \cap \text{TI}$ con $2 \leq j \leq K$
 $n := 0$ (contador de iteraciones)
repeat
 $n := n + 1$
 for $i := 1$ **to** K **do**
 solve \mathcal{LS}_i **con:**
 In: tasas $\mu_l^{(n)}$ para las transiciones de $T_l \cap \text{TI}$ con $1 \leq l < i$
 tasas $\mu_l^{(n-1)}$ para las transiciones de $T_l \cap \text{TI}$ con $i < l \leq K$
 Out: proporciones μ_i entre tasas de las transiciones de $T_i \cap \text{TI}$ y
 throughput $\chi_i^{(n)}$ de las transiciones de $T_i \cap \text{TI}$
 solve \mathcal{BS} (encontrar factor de escala λ) **con:**
 In: tasas $\mu_l^{(n)}$ para las transiciones de $T_l \cap \text{TI}$ con $1 \leq l < i$
 tasas $\mu_l^{(n-1)}$ para las transiciones de $T_l \cap \text{TI}$ con $i < l \leq K$
 proporciones entre tasas μ_i para las transiciones de $T_i \cap \text{TI}$
 throughput $\chi_i^{(n)}$ de las transiciones de $T_i \cap \text{TI}$
 Out: tasas reales $\mu_i^{(n)} = \lambda \cdot \mu_i$ de las transiciones de $T_i \cap \text{TI}$
 tales que $\chi_{\mathcal{BS}}^{(n)} = \chi_i^{(n)}$
 end for
until convergencia de $\{\chi_i^{(n)}\}_{i=1}^K$
output: throughput $\{\chi_i^{(n)}\}_{i=1}^K$ de las transiciones de $\langle \mathcal{S}, w \rangle$

En este algoritmo iterativo se van resolviendo uno por uno todos los sistemas de bajo nivel \mathcal{LS}_i (con $1 \leq i \leq K$). Cuando en la iteración n del esquema iterativo se resuelve el \mathcal{LS}_i , se obtienen para las transiciones de $T_i \cap \text{TI}$ los vectores $\chi_i^{(n)}$ de throughput y μ_i de proporciones entre tasas de servicio. Con esta información se busca, por medio del \mathcal{BS} , un factor de escala λ para las proporciones entre las tasas de disparo, de forma que los throughput de las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i coincidan. El cálculo de este factor de escala puede realizarse por medio de una búsqueda lineal en \mathcal{BS} . Se va resolviendo el \mathcal{BS} cambiando todas las tasas de las transiciones de $T_i \cap \text{TI}$ de forma que se mantengan las proporciones μ_i hasta que se consigan los mismos throughput en las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i . Notar que \mathcal{BS} tiene muchos menos estados que el sistema original e incluso que los \mathcal{LS}_i , por lo que el coste de esta búsqueda lineal es despreciable comparado con la complejidad del estudio de los otros sistemas.

Después de este doble paso se obtiene una nueva aproximación $\mu_i^{(n)}$ para las tasas de disparo de las transiciones de $T_i \cap \text{TI}$ que se mantienen en el estudio posterior de otros \mathcal{LS}_j y en \mathcal{BS} .

Quedan por aclarar algunos puntos de este algoritmo iterativo. Por un lado, cuestiones respecto a las restricciones del corte. Por otro lado, la selección de las tasas de disparo iniciales $\mu_j^{(0)}$ de las transiciones de $T_j \cap \text{TI}$ con $2 \leq j \leq K$, que son necesarias para resolver por primera vez el sistema \mathcal{LS}_1 e iniciar el algoritmo iterativo. Otra cuestión a tener en cuenta es la forma de calcular, al resolver el \mathcal{LS}_i , las proporciones μ_i entre las tasas de disparo de las transiciones de $T_i \cap \text{TI}$. Por último, consideraciones acerca de la búsqueda lineal del factor de escala λ en \mathcal{BS} para el cálculo de las tasas de disparo reales de las transiciones de interfaz en cada iteración y el test de convergencia para decidir el final del algoritmo iterativo.

Respecto a las restricciones sobre el corte, en la sección anterior se podía definir sobre cualquier conjunto de lugares que induzca una partición en el conjunto de transiciones y de lugares (excluidos los del corte) del DSSP original. Si se tiene en cuenta ahora las cuestiones numéricas, no es deseable que aparezca ninguna transición de interfaz que sea inmediata. Si éste es el caso, podría haber errores de ejecución al poder obtener como medida de utilización de alguna transición de interfaz 0 y por lo tanto divisiones por 0 al calcular su tasa relativa de disparo.

Respecto a la selección de las tasas iniciales para las transiciones de interfaz, el algoritmo iterativo no se ve afectado, en lo que respecta a las

aproximaciones de los throughput obtenidas, por la selección de estos valores. Por lo tanto, en principio se pueden seleccionar valores arbitrarios. Pero es razonable poner unos valores iniciales que tengan relación con el modelo. Una selección adecuada consiste en poner a cada transición de interfaz la tasa que tiene en el modelo original. De esta forma se reduce el número de iteraciones necesarias para alcanzar la convergencia del algoritmo. Esta elección dará sistemas agregados más rápidos en la primera iteración del método pero se corrige en las siguientes.

Por lo que respecta a la estimación de las proporciones μ_i entre las tasas de disparo de las transiciones de interfaz de la subred \mathcal{N}_i , se utilizan los nodos calculados en la fase de descomposición estructural. Como se explicó en la sección anterior, para cada subred \mathcal{N}_i producto del corte del DSSP original se calculaba un conjunto de nodos que resumen el comportamiento interno de la subred. Son estos nodos los que se emplean para realizar la estimación. El sistema de bajo nivel \mathcal{LS}_i contiene completamente la subred \mathcal{N}_i y los nodos del conjunto $H_i \cup I_i$. Por lo tanto, al resolver su CTMC isomorfa, se puede calcular la utilización de cada transición de interfaz si sólo se tienen en cuenta los lugares del conjunto $H_i \cup B$ (es decir, se calcula la probabilidad de que la transición esté sensibilizada una vez se eliminan los nodos internos de la subred \mathcal{N}_i). De esta manera, se está relacionando el funcionamiento completo de la subred \mathcal{N}_i con el de su resumen para poder exportar esta información al resto de sistemas de bajo nivel. Posteriormente, basta emplear la fórmula del cálculo del throughput del teorema 2.87 para encontrar las proporciones entre las tasas de disparo de las transiciones de interfaz. En la iteración n del algoritmo, la tasa proporcional de disparo de la transición $t \in T_i \cap \text{TI}$ es:

$$\mu_i[t] = \frac{\chi_i^{(n)}[t]}{P\{\mathbf{m}[p] \geq W(p, t) \quad \forall p \in \bullet t \cap (H_i \cup B)\}}$$

Esta expresión es válida si se emplea semántica de un solo servidor. Si se emplea semántica de infinitos servidores, hay que sustituir la utilización de las transiciones por su grado medio de sensibilización, es decir, la tasa proporcional de disparo es en este caso:

$$\mu_i[t] = \frac{\chi_i^{(n)}[t]}{E[\max\{k \mid \mathbf{m}[p] \geq kW(p, t) \quad \forall p \in \bullet t \cap (H_i \cup B)\}]}$$

En cuanto al algoritmo de búsqueda lineal del factor de escala en \mathcal{BS}

para calcular las tasas reales de disparo de las transiciones de interfaz, hay que tener en cuenta lo siguiente. Se quiere conseguir que coincidan los throughput de las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i , es decir, que coincidan los throughput de las transiciones de interfaz. A diferencia de MG's, en DSSP's los throughput de las transiciones pueden ser diferentes. Por lo tanto, no se puede hablar de throughput de un DSSP sino de throughput de una transición de un DSSP. Entonces, conseguir que los throughput de todas las transiciones de interfaz coincidan en los dos sistemas podría suponer un problema. Como en DSSP's las ratios de visita o throughput relativos de las transiciones vienen determinados por la estructura de la red (teorema 5.9) y los sistemas agregados construidos preservan estas ratios de visita, entonces desaparece el problema y basta que el throughput de una transición t de interfaz cualquiera coincida en \mathcal{BS} y \mathcal{LS}_i para poder asegurar que coinciden todos los throughput de todas las transiciones de interfaz. A lo largo del algoritmo iterativo se va modificando el valor de las tasas de las transiciones de interfaz de los \mathcal{LS}_i y de \mathcal{BS} . El throughput de las transiciones de estos sistemas es una función racional (cociente de polinomios) con respecto a estas variables. Como se han calculado unas proporciones entre las tasas de disparo que se quieren mantener en \mathcal{BS} , si se quiere calcular un factor de escala λ de forma que coincidan los throughput de las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i , lo que se está haciendo en el fondo es plantear una ecuación racional de la forma:

$$\chi_{\mathcal{BS}}^{(n)}[t](\lambda) = \chi_{\mathcal{LS}_i}^{(n)}[t]$$

En la ecuación anterior, t es una transición de interfaz cualquiera que se fija como referencia en \mathcal{BS} y \mathcal{LS}_i . Entonces, $\chi_{\mathcal{LS}_i}^{(n)}[t]$ es el throughput en \mathcal{LS}_i de la transición t calculado justo antes de empezar el estudio en \mathcal{BS} . Por su parte, $\chi_{\mathcal{BS}}^{(n)}[t](\lambda)$ es una función de λ , que devuelve el throughput en \mathcal{BS} de la misma transición t de referencia. Esta función es racional en λ y no se conoce de forma explícita. Calculando una solución λ de la ecuación anterior, se puede asegurar que el throughput de todas las transiciones de interfaz coincide en \mathcal{BS} y \mathcal{LS}_i . El problema numérico de resolución de esta ecuación puede resolverse con algoritmos como el de la secante.

Por último, queda por aclarar el test de convergencia. Este algoritmo numérico calcula unas aproximaciones del throughput aceptables para una técnica de aproximación (por debajo del 5%). Teniendo en cuenta la aproximación que se va a obtener al final, no es necesario que el test de convergencia

sea muy exigente ya que esto provocaría el cálculo de un mayor número de iteraciones sin que por ello se aumente la precisión de los resultados obtenidos. Si se pone un test de convergencia poco exigente, entonces el algoritmo finalizará antes de lo debido, aumentando en general el error de la aproximación. El test de convergencia que se utiliza consiste en iterar hasta que la diferencia entre los throughput de las transiciones de interfaz de todos los \mathcal{LS}_i en dos iteraciones consecutivas del método sea menor que el 0.1%.

En este momento, sería necesario estudiar las propiedades numéricas del algoritmo iterativo para demostrar propiedades de convergencia, unicidad de solución e incluso estimar el error cometido en la aproximación, pero se ha dejado como trabajo futuro. A falta de un mejor soporte teórico, la experimentación numérica con diferentes ejemplos sugiere que el esquema converge en muy pocos pasos (entre 3 y 6 iteraciones), converge a la misma solución independientemente de las tasas iniciales elegidas para iniciar el algoritmo, y el error cometido al final del mismo (inferior al 5%) es admisible para técnicas de aproximación en evaluación de prestaciones de sistemas concurrentes basada en modelos formales.

5.6 Conclusiones

En este capítulo se han desarrollado en detalle los trabajos realizados en [PJCS96c] y [PJCS96a] sobre aproximación de throughput en DSSP's. Los DSSP's están formados por varias SM's conectadas por canales, por lo que la técnica de descomposición estructural de DSSP's se basa en la reducción de las SM's.

Todo lo expuesto en este capítulo son aportaciones de esta memoria. En concreto estas aportaciones son:

- Técnica de reducción parcial de SM's.
- Técnica de reducción total de SM's.
- Adaptación del algoritmo iterativo de aproximación a DSSP's.

La técnica de reducción parcial de SM's es básicamente el trabajo realizado en [PJCS96c]. El objetivo es reducir al máximo el grafo tangible de alcanzabilidad de una SM en un DSSP preservando las transiciones que tienen canales como lugares de entrada o salida. De esta forma se logra

mantener exactamente las proyecciones de marcados alcanzables y secuencias de disparo del DSSP original sobre los nodos preservados en los sistemas agregados.

Esta técnica de reducción parcial de SM's consigue una reducción importante del DSSP original si éste está compuesto por SM's de cierto tamaño. Si éste no es el caso, es necesario realizar agregaciones más importantes. Este es el objetivo de la reducción total de SM's, trabajo realizado en [PJCS96a]. Aquí se estudian las condiciones necesarias que debe cumplir una SM en un DSSP para que se pueda agregar completamente a una única transición. No todas las SM's de un DSSP se pueden agregar totalmente, por lo que esta es una técnica que no se puede aplicar a cualquier DSSP. Después de reducir las SM's de un DSSP con una o las dos técnicas anteriores es posible reducir aun más ciertas partes que eventualmente pueden tener estructura de WTS por medio de la técnica de reducción del capítulo 4.

La clase de los DSSP's incluye a las de WTS's y MG's. Desde este punto de vista, la técnica de descomposición estructural de DSSP's desarrollada en este capítulo es una extensión de las de los capítulos 3 y 4 para MG's y WTS's, en el sentido de la técnica de descomposición aplicada a un WTS o un MG producen los mismos sistemas agregados que las de los capítulos 3 y 4 respectivamente.

El algoritmo iterativo de aproximación es, salvo ligeras adaptaciones a la nueva clase de redes, el mismo de [CCJS94] para MG's que se ha explicado en detalle en el capítulo 3.

Dependiendo de la técnica de reducción de subredes que se emplee al generar los sistemas agregados de un DSSP, estos pueden preservar distintas propiedades del DSSP original. Como mínimo se asegura que los sistemas agregados son vivos y limitados, por lo que tienen estados recurrentes, lo que asegura la existencia de distribución en estado estacionario única para sus CTMC's isomorfas. Pero si sólo se emplea la técnica de reducción parcial de SM's de la sección 5.2, entonces los sistemas agregados mantienen exactamente las proyecciones de estados alcanzables y de secuencias de disparo del DSSP original sobre los nodos preservados en cada sistema agregado.

En el capítulo 6 se expone una técnica basada en el estudio del espacio de estados que permite preservar al menos las proyecciones de los estados alcanzables en los sistemas agregados de cualquier red de Petri, en particular de los DSSP's.

Queda como trabajo futuro demostrar las propiedades de convergencia del algoritmo numérico de aproximación del tiempo de respuesta.

Capítulo 6

Descomposición del grafo de alcanzabilidad para el análisis numérico de redes de Petri estocásticas

Una vez expuestas las técnicas de aproximación del throughput de las transiciones de grafos marcados (MG's), grafos marcados con pesos (WTS's) y sistemas deterministas de procesos secuenciales (DSSP's) de los tres capítulos anteriores, el siguiente paso consiste en intentar desarrollar una técnica general que permita la aproximación del throughput de las transiciones de cualquier SPN. En este capítulo se van a desarrollar los resultados obtenidos en este intento de generalización. Estos resultados se exponen en un marco más general que el de aproximación de throughput de transiciones ya que pueden ser utilizados para otros propósitos.

Entre los resultados obtenidos cabe destacar un algoritmo que permite reducir la complejidad tanto en espacio como en tiempo (respecto al algoritmo clásico) del cálculo y representación descompuesta del grafo de alcanzabilidad (RG) de cualquier SPN. Esta representación puede ser utilizada para diversos fines como pueden ser el cálculo exacto de la distribución en estado estacionario o de otros índices de prestaciones.

En este capítulo se utiliza esta representación del RG para el desarrollo de los sistemas agregados asociados al corte de una SPN cualquiera. En concreto, se desarrollan tres técnicas diferentes para generar los sistemas

agregados dependiendo de las características que se requieran en los mismos. En estas condiciones es posible utilizar el algoritmo numérico de los capítulos anteriores para aproximar el throughput de las transiciones de cualquier SPN. Pero los sistemas agregados podrían utilizarse para aproximar otros índices de prestaciones.

Todo lo expuesto en este capítulo son aportaciones de esta memoria exceptuando los de la sección 6.2. Los resultados fueron presentados en [PJC98, PJC99a, PJC99b].

El capítulo está organizado de la siguiente manera. En la sección 6.1 se exponen los trabajos realizados por otros autores que han permitido el desarrollo de estas técnicas, en la sección 6.2 desarrolla una técnica de descomposición a nivel de red válida para SPN's desarrollada en [CDS99] que fue aplicada para el cálculo exacto de la distribución en estado estacionario de la cadena de Markov en tiempo continuo (CTMC) del modelo original. La sección 6.3 desarrolla la primera técnica de generación de sistemas agregados, en la que se asegura la ergodicidad de los mismos (en este capítulo se entenderá que un sistema agregado es ergódico si lo es su CTMC isomorfa) y por lo tanto su posible utilización para cálculos numéricos. En la sección 6.4 se expone una visión estructurada del RG de cualquier SPN lo que permite desarrollar un algoritmo eficiente para su cálculo y almacenamiento. En las secciones 6.5 y 6.6 se emplea la representación del RG de la sección 6.4 para la generación de los sistemas agregados. Los sistemas agregados de la técnica de la sección 6.5 mantienen las proyecciones de los estados alcanzables de la red original pero pueden contener secuencias de disparo espurias. En la técnica de la sección 6.6 se reducen algunas de estas secuencias de disparo espurias. En la sección 6.7 se emplean los sistemas agregados generados con cualquiera de las tres técnicas anteriores para la aproximación del throughput de las transiciones de la red original. Por último, en la sección 6.8 se indican las conclusiones finales del capítulo.

6.1 Introducción

En el capítulo 3 se ha desarrollado en detalle el primer trabajo sobre aproximación de throughput de transiciones expuesto en [CCJS94] aplicable a cualquier MG. Este trabajo se basa en una técnica de descomposición a nivel de red que permite el desarrollo de unos sistemas agregados que son proyecciones exactas (en términos de marcados alcanzables y secuencias de

disparo) del modelo original. Posteriormente se logra extender la técnica de descomposición a WTS's [PJCS96b] y DSSP's [PJCS96c, PJCS96a], como se ha visto en los capítulos 4 y 5. Para estas clases de redes ya no es posible construir, a nivel de red, sistemas agregados de menor tamaño que mantengan las proyecciones de los marcados alcanzables y secuencias de disparo del sistema original. Afortunadamente, empleando técnicas heurísticas ad-hoc (los conceptos de resistencia y marcado ponderado en WTS's y la reducción de máquinas de estados en DSSP's) es posible construir sistemas agregados que permiten obtener buenas aproximaciones del throughput de las transiciones en estas clases de redes. Posteriormente, en [CT93, CDS99] se introducen técnicas basadas en *álgebra tensorial*. Trabajos recientes basados en técnicas de álgebra tensorial son [Cia01, BK01, BCDK00]. En este tipo de técnicas, por medio de una visión estructurada a nivel de red del modelo original, se construyen varios sistemas agregados cuyas CTMC's se pueden combinar para obtener la CTMC del modelo original. De esta manera es posible calcular de forma exacta la distribución en estado estacionario del modelo original sin calcular ni almacenar su CTMC completa, sino únicamente las de sus sistemas agregados.

El intento de adaptar estas técnicas basadas en álgebra tensorial para el cálculo aproximado del throughput de transiciones conduce al desarrollo de las aportaciones de este capítulo. Desde el punto de vista de cálculo aproximado, los sistemas agregados de [CDS99] tienen un problema fundamental y es que en general no son ergódicos, por lo que la solución de su CTMC puede no tener sentido. En [PJC98, PJC99a] se resuelve este problema con una técnica muy sencilla (expuesta en la sección 6.3) que permite generar, a partir de los sistemas agregados de [CDS99], CTMC's ergódicas asociadas a los mismos. Aplicando algoritmos como el de aproximación del tiempo de respuesta de [CCJS94] (que se viene utilizando en toda esta memoria) a estas CTMC's se tiene una primera técnica de aproximación del throughput de transiciones que puede aplicarse a cualquier SPN.

Aunque una técnica como la que se expone en la sección 6.3 permite calcular aproximaciones con sentido para cualquier SPN, en algunas situaciones el resultado obtenido puede no ser suficientemente preciso (error inferior al 5% para una técnica de aproximación). El problema de falta de precisión es debido, desde el punto de vista del autor de esta memoria, a dos problemas. El primero es la posible inclusión en las CTMC's asociadas a los sistemas agregados de *estados espurios*, esto es, marcados alcanzables en los sistemas agregados que no se corresponden con la proyección de ningún

estado alcanzable de la red original sobre los nodos que se preserven en cada sistema agregado. El segundo problema es la posible *agregación inadecuada* de estados diferentes del modelo original en el mismo estado de la CTMC asociada a un sistema agregado. Este segundo punto requiere una explicación más detallada, ya que es evidente que para reducir el espacio de estados del modelo original hay que agregar estados diferentes de la red original en el mismo estado de los sistemas agregados. La idea de las técnicas de descomposición desarrolladas en los capítulos anteriores para clases particulares de redes consiste en enmascarar en un determinado sistema agregado el comportamiento interno de todas las subredes excepto una. Esto se traduce a nivel del RG en la agregación de estados del sistema original que están conectados por caminos no orientados de transiciones internas de los módulos que se quieren reducir. El efecto lateral que se produce reduciendo a nivel de red es la posible agregación de estados que no están conectados de esta manera en el RG del sistema original. Este efecto se produce por la pérdida de información sobre el estado interno de distintos módulos reducidos y se traduce en la aparición de estados espurios y secuencias de disparo espurias.

Para resolver estos problemas, en [PJC99a] se presenta una visión estructurada del RG de SPN's (esta visión se desarrolla en la sección 6.4) que permite calcular y almacenar de forma eficiente tanto en espacio como en tiempo una representación del mismo de forma descompuesta. Aunque esta representación del RG se desarrolla con el objetivo de resolver los problemas de precisión de la técnica de descomposición anterior, al contener toda la información del RG del modelo original, también puede utilizarse para otro tipo de análisis, como por ejemplo análisis exacto o aproximado de otros índices de prestaciones.

En [PJC99a], la representación del RG se emplea para desarrollar otras dos técnicas de descomposición del modelo original, que constituyen las técnicas segunda y tercera de este capítulo. La segunda técnica (que se presenta en la sección 6.5) genera sistemas agregados ergódicos que mantienen las proyecciones de los estados alcanzables del modelo original sobre los nodos preservados en los sistemas agregados. De esta manera, los sistemas agregados tienen un comportamiento más cercano al del modelo original que los de la primera técnica. Sin embargo, los sistemas agregados construidos con esta técnica incluyen secuencias de disparo espurias, esto es, secuencias disparables en los sistemas agregados que no se corresponden con la proyección de ninguna secuencia de disparo en el modelo original. La tercera técnica (que se presenta en la sección 6.6) genera sistemas agregados

ergódicos que preservan las proyecciones de los estados alcanzables y reducen (respecto a la segunda técnica) algunas de las secuencias de disparo espurias. Aun así no se consiguen eliminar en general todas las secuencias de disparo espurias.

En la sección 6.7 se aplican las tres técnicas de descomposición anteriores al cálculo aproximado del throughput de las transiciones del modelo original por medio del algoritmo iterativo de aproximación del tiempo de respuesta de [CCJS94] que se está empleando a lo largo de toda la memoria.

Las técnicas que se presentan en este capítulo son más eficientes tanto en tiempo como en espacio que el algoritmo exacto clásico de resolución de la CTMC del modelo original ya que ninguna de estas técnicas requiere el almacenamiento ni la resolución de la CTMC del modelo original, sino los de las CTMC's asociadas a los sistemas agregados que son considerablemente más reducidas. El precio a pagar por la reducción de la complejidad es la precisión de los resultados obtenidos, ya que no se obtienen los resultados exactos sino simplemente aproximados.

6.2 Descomposición estructural de redes de Petri

En esta sección se va a desarrollar la técnica de descomposición de redes de Petri generales propuesta en [CDS99].

6.2.1 Visión estructurada de redes de Petri

Cualquier red de Petri puede verse como un conjunto de *módulos* (redes de Petri disjuntas más sencillas) que se comunican por medio de un conjunto de *canales* (lugares).

Definición 6.1 [CDS99] (SAM).

Un P/T sistema $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ con $\mathcal{N} = \langle B \cup (\bigcup_{i=1}^K P_i), \bigcup_{i=1}^K T_i, F, W \rangle$ es un sistema de módulos comunicados asincrónicamente (SAM), si:

- i) $P_i \cap B = \emptyset$ y $P_i \cap P_j = \emptyset$ para $1 \leq i, j \leq K, i \neq j$.
- ii) $T_i = P_i^\bullet \cup {}^\bullet P_i$ y $T_i \cap T_j = \emptyset$ para $1 \leq i, j \leq K, i \neq j$.

$\mathcal{S}_i = \langle \mathcal{N}_i, \mathbf{m}_0^i \rangle = \langle P_i, T_i, F_i, W_i, \mathbf{m}_0^i \rangle$ con $F_i = F \cap ((P_i \times T_i) \cup (T_i \times P_i))$, $W_i = W|_{F_i}$ y $\mathbf{m}_0^i = \mathbf{m}_0|_{P_i}$ para $1 \leq i \leq K$ se llaman módulos de \mathcal{S} . Los lugares de B se llaman canales, las transiciones de $T_i = {}^\bullet B \cup B^\bullet$ se llaman transiciones de interfaz y el resto transiciones internas.

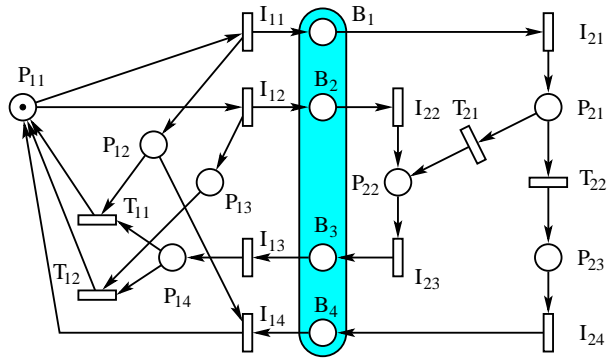


Figura 6.1: Un SAM.

En la figura 6.1 puede verse un SAM compuesto por dos módulos y 4 canales. El módulo \mathcal{S}_i con $i = 1, 2$ es el sistema generado por los nodos cuyas etiquetas comienzan por P_i , T_i o I_i . Los canales son los lugares dentro de la zona sombreada y se les denotará con etiquetas que comienzan por B .

Todas las redes de Petri fuertemente conexas pueden verse como un SAM con solo añadir una *visión estructurada* al modelo. Esta visión estructurada puede obtenerse por construcción del modelo o bien por observación del mismo. Una misma red de Petri fuertemente conexa admite muchas visiones estructuradas que van desde considerar cada transición como un módulo diferente (y por lo tanto cada lugar es un canal) hasta considerar que toda la red es un único módulo (y por lo tanto no hay canales).

Con respecto a la interpretación temporal, se supondrá que el modelo inicial tiene un comportamiento en estado estacionario único para que admita el cálculo de índices de prestaciones en estado estacionario. Por ello se supondrá que el modelo original es estructuralmente vivo y limitado, por lo tanto consistente y conservativo (teorema 2.49) y reversible, por lo tanto ergódico (teorema 2.85). En realidad es suficiente que el sistema tenga estados recurrentes, porque tomando uno de ellos como marcado inicial se obtiene un sistema reversible.

6.2.2 Regla de reducción y vistas abstractas

En esta sección se va a desarrollar la regla de reducción expuesta en [CDS99], para el comportamiento interno de los módulos de un SAM. Cada módulo

se descompone en varias subredes de forma que cada subred se sustituye por un conjunto de lugares MSIP's (ver definición 2.61). Posteriormente, por medio de la regla de reducción de los módulos, a partir del modelo original se pueden construir K sistemas de bajo nivel \mathcal{LS}_i con $1 \leq i \leq K$ y un sistema de alto nivel o esqueleto básico \mathcal{BS} . En cada sistema de bajo nivel se mantiene un solo módulo mientras que el resto se reducen. En el \mathcal{BS} se reducen todos los módulos. En [CDS99], los \mathcal{LS}_i y el \mathcal{BS} se utilizan para la resolución exacta de la CTMC del modelo original por medio de una técnica basada en álgebra tensorial. En [PJC99a] se adapta esta descomposición para desarrollar una técnica de aproximación más eficiente.

Definición 6.2 [CDS99] Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM con $\mathcal{N} = \langle P, T, F, W \rangle$. Se define la relación R en $P = B \cup (\bigcup_{i=1}^K P_i)$ como: $(p, p') \in R$ para $p, p' \in P$ si y sólo si existe un camino no dirigido π en \mathcal{N} de p a p' tal que $\pi \cap \text{TI} = \emptyset$, es decir, compuesto únicamente por lugares y transiciones internas.

Proposición 6.3 [CDS99] Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM. la relación R de la definición anterior es una relación de equivalencia.

Demostración:

Reflexiva: $(p, p) \in R$ para todo $p \in P$ ya que p está conectado consigo mismo por medio del camino vacío que no contiene transiciones de interfaz.

Simétrica: Si $(p_1, p_2) \in R$, sea π_1 un camino no dirigido en \mathcal{N} de p_1 a p_2 . Entonces $\pi_1 = n_1 \dots n_m$ donde los n_i son nodos de \mathcal{N} que no pertenecen a TI. Sea $\pi_2 = n_m \dots n_1$ el camino π_1 recorrido en sentido inverso. Entonces π_2 es un camino no dirigido en \mathcal{N} de p_2 a p_1 y que está formado por nodos que no pertenecen a TI, es decir, $(p_2, p_1) \in R$.

Transitiva: Si $(p_1, p_2), (p_2, p_3) \in R$, sean π_1, π_2 caminos no dirigidos en \mathcal{N} sin transiciones de interfaz de p_1 a p_2 y de p_2 a p_3 respectivamente. Entonces $\pi_1 \circ \pi_2$ es un camino no dirigido en \mathcal{N} sin transiciones de interfaz de p_1 a p_3 por lo que $(p_1, p_3) \in R$. \diamond

Propiedad 6.4 [CDS99] Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM con $\mathcal{N} = \langle P, T, F, W \rangle$ y R la relación sobre el conjunto $P = B \cup (\bigcup_{i=1}^K P_i)$ de la definición 6.2. Se cumplen las siguientes propiedades:

1. $(b, p) \notin R$ para todo $b \in B$ y $p \neq b$.
2. Si $p \in P_i$ y $(p, p') \in R$ con $p \neq p'$ entonces $p' \in P_i$.

Demostración:

1) Por la definición 6.1 todas las transiciones de entrada y salida de cualquier canal son transiciones de interfaz. Entonces, por la definición de la relación de equivalencia R un canal sólo puede estar relacionado consigo mismo.

2) Si $(p, p') \in R$, entonces existe un camino π no dirigido en \mathcal{N} sin transiciones de interfaz de p a p' . Por la definición 6.1, los únicos nodos de \mathcal{N} que pueden conectar nodos de dos módulos distintos de \mathcal{N} son los canales que pueden conectar transiciones de interfaz pertenecientes a distintos módulos. Como π no tiene transiciones de interfaz, tampoco puede tener canales y por lo tanto no puede abandonar el módulo \mathcal{N}_i , es decir, $p' \in P_i$. \diamond

Esta propiedad da una idea de cuáles son las clases de equivalencia que la relación R induce sobre P . Cada canal forma individualmente una clase de equivalencia y las clases de equivalencia que no son canales deben estar totalmente contenidas en un módulo del SAM, es decir, la relación de equivalencia R induce relaciones de equivalencia en cada P_i del SAM.

Notación 6.5 [CDS99] Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM con $P = B \cup (\bigcup_{i=1}^K P_i)$ y R la relación de la definición 6.2. Se denotan por P_i^j con $1 \leq j \leq r(i)$ a las $r(i)$ clases de equivalencia que R induce en P_i para $1 \leq i \leq K$.

El siguiente paso consiste en añadir por cada clase de equivalencia P_i^j con $1 \leq i \leq K$ y $1 \leq j \leq r(i)$ un conjunto H_i^j (normalmente de menor cardinal) de lugares MSIP's (ver definición 2.61). La razón de utilizar lugares MSIP's es que su cómputo puede realizarse estructuralmente de forma eficiente. Serán los lugares de los conjuntos H_i^j con $1 \leq j \leq r(i)$ los que resuman el comportamiento del módulo \mathcal{N}_i para $1 \leq i \leq K$. El algoritmo de cálculo de los conjuntos H_i^j es el siguiente.

Algoritmo 6.6 [CDS99] **Cálculo de MSIP's.**

input: P_i^j clase de equivalencia de R en P_i .

Calcular $Y_i^j := \{\mathbf{y} \mid \mathbf{y}[P_i^j] \cdot \mathbf{C}[P_i^j, T_i^j] = \mathbf{0}, \mathbf{y}[P \setminus P_i^j] = \mathbf{0} \text{ con } \mathbf{y} \text{ de soporte mínimo}\}$.

Calcular $H_i^j := \{p_{\mathbf{y}} \mid p_{\mathbf{y}} \text{ es lugar con vector de incidencia } \mathbf{l}_{p_{\mathbf{y}}} = \mathbf{y} \cdot \mathbf{C}\}$.

for each $p \in H_i^j$ **do**

 calcular su marcado inicial por medio del algoritmo 2.60

end for

output: Conjunto H_i^j de lugares MSIP's con sus marcados iniciales.

La idea de este algoritmo consiste en considerar todos los MSIP's $p_{\mathbf{y}}$ derivados a partir de los p-semiflujos mínimos \mathbf{y} de la subred generada por P_i^j , es decir, $p_{\mathbf{y}}$ es un lugar con vector de incidencia $\mathbf{l}_{p_{\mathbf{y}}} = \mathbf{y} \cdot \mathbf{C}$ donde \mathbf{y} es un p-semiflujo mínimo de la red generada por P_i^j . El cardinal del conjunto H_i^j puede crecer exponencialmente en el tamaño de la subred generada por P_i^j , por lo que la complejidad teórica de este algoritmo (tanto en tiempo como en espacio) es exponencial en el tamaño de la subred generada por P_i^j . De todas formas, hay que tener en cuenta que las clases de equivalencia P_i^j son una parte de un módulo de la red original por lo que el algoritmo no tiene en la práctica una complejidad elevada.

Se está en condiciones de construir el sistema extendido y los sistemas agregados.

Notación 6.7 Sea $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM con $P = B \cup (\bigcup_{i=1}^K P_i)$. Sean H_i^j con $1 \leq i \leq K$ y $1 \leq j \leq n(i)$ los conjuntos de lugares MSIP's calculados con el algoritmo 6.6. Se denota por H_i con $1 \leq i \leq K$ al conjunto $H_i = \bigcup_{j=1}^{n(i)} H_i^j$ y por H al conjunto $H = \bigcup_{i=1}^K H_i$.

Definición 6.8 [CDS99] Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM. El sistema extendido $\mathcal{ES} = \langle \mathcal{EN}, \mathbf{m}_0^{\mathcal{EN}} \rangle$ de \mathcal{S} es el sistema resultante de añadir a \mathcal{S} los lugares del conjunto H con sus vectores de incidencia y los marcados iniciales necesarios para hacerlos implícitos.

El sistema extendido se forma a partir del modelo original añadiendo un conjunto de lugares implícitos, por lo que \mathcal{ES} tiene el mismo número de estados que el modelo original (sólo difieren en la codificación de los mismos) y el mismo lenguaje de secuencias de disparo.

A partir del sistema extendido es inmediato construir los sistemas agregados.

Definición 6.9 [CDS99] Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM con $\mathcal{N} = \langle P, T, F, W \rangle$, $P = B \cup (\bigcup_{i=1}^K P_i)$ y $T = \bigcup_{i=1}^K T_i$. Sea \mathcal{ES} su sistema extendido.

- i) El sistema de bajo nivel $\mathcal{LS}_i = \langle \mathcal{LN}_i, \mathbf{m}_0^i \rangle$ con $1 \leq i \leq K$ es el sistema que se obtiene eliminando de \mathcal{ES} los lugares $\bigcup_{j \neq i} P_j$ y las transiciones $\bigcup_{j \neq i} T_j \setminus \text{TI}$ y sus arcos adyacentes.
- ii) El sistema de alto nivel (o esqueleto básico) $\mathcal{BS} = \langle \mathcal{BN}, \mathbf{m}_0^{\mathcal{BN}} \rangle$ es el sistema que se obtiene eliminando de \mathcal{ES} los lugares $\bigcup_{i=1}^K P_i$ y las transiciones $\bigcup_{i=1}^K T_i \setminus \text{TI}$ y sus arcos adyacentes.

En cada \mathcal{LS}_i , todas las subredes \mathcal{N}_j con $j \neq i$ se reducen a sus transiciones de interfaz y a los lugares del conjunto H_j , mientras que la subred \mathcal{N}_i se preserva completamente. En el \mathcal{BS} se reducen todas las subredes. Los \mathcal{LS}_i representan diferentes vistas de bajo nivel del modelo original. El \mathcal{BS} representa una vista de alto nivel del sistema original.

Los siguientes resultados establecen las propiedades que cumplen los sistemas agregados que se acaban de construir.

Teorema 6.10 [CDS99] *Sea \mathcal{S} un SAM con K módulos conservativo. Entonces, \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} son conservativos.*

Teorema 6.11 [CDS99] *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM con $\mathcal{N} = \langle P, T, F, W \rangle$, $P = B \cup (\bigcup_{i=1}^K P_i)$ y $T = \bigcup_{i=1}^K T_i$. Entonces, \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} cumplen:*

1. $L(\mathcal{S})|_{T_i \cup T_I} \subseteq L(\mathcal{LS}_i)$ para $1 \leq i \leq K$ y $L(\mathcal{S})|_{T_I} \subseteq L(\mathcal{BS})$.
2. $R(\mathcal{S})|_{P_i \cup B \cup H} \subseteq R(\mathcal{LS}_i)$ para $1 \leq i \leq K$ y $R(\mathcal{S})|_{B \cup H} \subseteq R(\mathcal{BS})$.

Este teorema asegura que los sistemas agregados incluyen las proyecciones de los marcado alcanzables y las secuencias de disparo de la red original sobre los nodos preservados en cada sistema. Pero como no se tiene una igualdad, estos sistemas agregados pueden añadir nuevos marcados alcanzables o secuencias de disparo que no se corresponden con proyecciones de marcados alcanzables o secuencias de disparo del sistema original.

La técnica de descomposición que se acaba de presentar aplicada a un MG produce exactamente los mismos sistemas agregados que la técnica del capítulo 3, aunque el conjunto de lugares implícitos necesarios para resumir el comportamiento de cada subred se calculen por métodos diferentes. En cambio, para WTS's esta técnica de descomposición no es equivalente la técnica del capítulo 3, ya que no se preservan las resistencias entre transiciones de interfaz.

En [CDS99] se emplea esta técnica de descomposición para calcular la distribución en estado estacionario del modelo original a partir de las CTMC's de los sistemas agregados. La principal aportación de este trabajo desde el punto de vista de las técnicas de aproximación que se están desarrollando en esta memoria consiste en la posibilidad de descomponer a nivel de red cualquier red de Petri y obtener a partir de ella unos sistemas agregados con los que se puede hacer cálculos. En la siguiente sección se estudiarán los principales problemas que genera la utilización de estos sistemas agregados y cómo resolverlos.

6.3 Garantizando la ergodicidad de los sistemas agregados

A partir del teorema 6.11, se tiene que los RG's de los sistemas agregados incluyen al menos todas las proyecciones (en los nodos preservados en cada sistema) de los marcados alcanzables del sistema original. Pero como las inclusiones de este teorema no son estrictas, los RG's de los sistemas agregados pueden incluir nuevos marcados alcanzables o secuencias de disparo espurios que no se corresponden con marcados alcanzables o secuencias de disparo del sistema original. En algunos casos, este comportamiento no deseado puede conducir a sistemas agregados no ergódicos. En esta sección se presenta una técnica para asegurar la ergodicidad de los sistemas agregados, evitando que estos comportamientos no deseados afecten a la técnica numérica de aproximación. Es necesario que los sistemas agregados sean ergódicos puesto que la técnica numérica de aproximación debe resolver las CTMC's de estos sistemas.

En la figura 6.1 se tenía un SAM compuesto por dos módulos y 4 canales. El módulo i con $i = 1, 2$ es la subred generada por los nodos cuyas etiquetas comienzan por P_i , T_i o I_i . Aplicando la técnica de descomposición de la sección anterior se obtienen \mathcal{ES} y \mathcal{LS}_1 de las figuras 6.2.a y 6.2.b respectivamente. En el sistema original (se puede observar también en \mathcal{ES}), después de cada disparo de la transición de interfaz I_{22} sólo puede dispararse la transición de interfaz I_{23} . En cambio, en \mathcal{LS}_1 también es posible disparar la transición de interfaz I_{24} después de cada disparo de I_{22} . Esta nueva secuencia de disparo hace que \mathcal{LS}_1 no sea vivo. En efecto, la secuencia de disparo $\sigma = I_{12}I_{22}I_{24}$ es disparable en \mathcal{LS}_1 y el marcado que se alcanza es B_4P_{13} que es un bloqueo total (no hay ninguna transición sensibilizada). En otras palabras, los sistemas agregados pueden no ser ergódicos.

Afortunadamente, hay una forma directa de asegurar la ergodicidad de los sistemas agregados. En general, los RG's de los sistemas agregados contruidos con la técnica de la sección anterior pueden tener varias componentes fuertemente conexas. Para que las CTMC's de estos sistemas sean ergódicas, deben tener una única componente fuertemente conexa (definición 2.80). Por lo tanto, es necesario eliminar del RG de cada sistema agregado todas sus componentes fuertemente conexas excepto la que contiene al marcado inicial. Se demostrará que los RG's reducidos de esta forma incluyen al menos todas las proyecciones (en los nodos preservados en cada sistema agregado) de los marcados alcanzables y secuencias de disparo del sistema original.

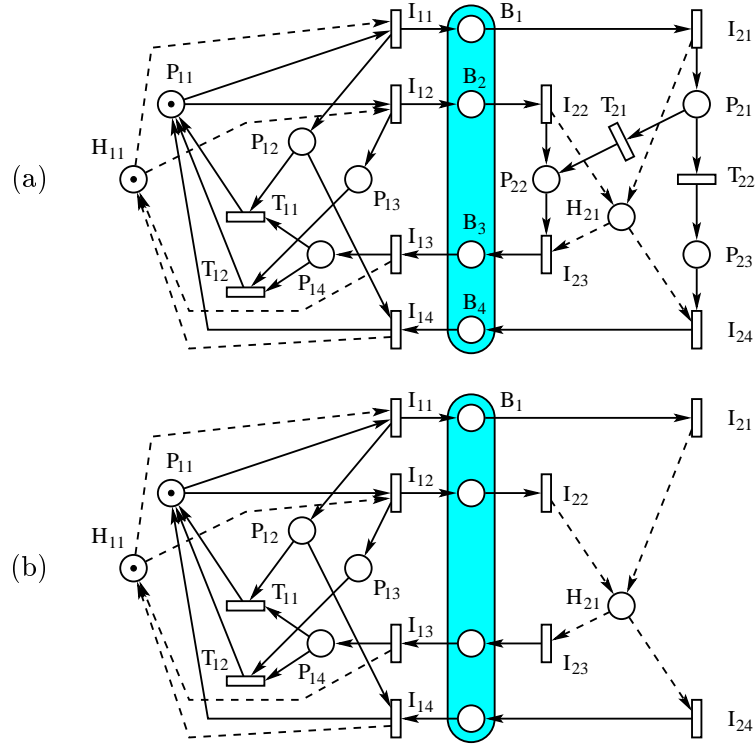


Figura 6.2: (a) \mathcal{ES} y (b) \mathcal{LS}_1 del SAM de la figura 6.1.

Teorema 6.12 Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM, \mathcal{ES} su sistema extendido, \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} sus sistemas agregados. Sean $\text{RG}^*(\mathcal{LS}_i)$ con $1 \leq i \leq K$ y $\text{RG}^*(\mathcal{BS})$ las componentes fuertemente conexas de $\text{RG}(\mathcal{LS}_i)$ y $\text{RG}(\mathcal{BS})$ respectivamente que contienen a sus marcados iniciales. Denotando por $\text{R}^*(\mathcal{LS}_i)$, $\text{R}^*(\mathcal{BS})$, $\text{L}^*(\mathcal{LS}_i)$ y $\text{L}^*(\mathcal{BS})$ a los estados alcanzables y secuencias de disparo de $\text{RG}^*(\mathcal{LS}_i)$ y $\text{RG}^*(\mathcal{BS})$ respectivamente, se tiene:

1. $\text{L}(\mathcal{S})|_{T_i \cup T_I} \subseteq \text{L}^*(\mathcal{LS}_i)$ para $1 \leq i \leq K$ y $\text{L}(\mathcal{S})|_{T_I} \subseteq \text{L}^*(\mathcal{BS})$.
2. $\text{R}(\mathcal{ES})|_{P_i \cup B \cup H} \subseteq \text{R}^*(\mathcal{LS}_i)$ para $1 \leq i \leq K$ y $\text{R}(\mathcal{ES})|_{B \cup H} \subseteq \text{R}^*(\mathcal{BS})$.

Demostración:

Considerar el sistema extendido \mathcal{ES} de \mathcal{S} . Por construcción, todos los lugares del conjunto $H = \bigcup_{i=1}^K H_i$ son implícitos en \mathcal{ES} . Entonces $\text{L}(\mathcal{S}) = \text{L}(\mathcal{ES})$.

Como se asume que \mathcal{S} es reversible, \mathbf{m}_0 es un estado recurrente (si \mathcal{S} no es reversible, pero tiene algún estado recurrente, el resultado también es cierto tomando cualquiera de ellos como \mathbf{m}_0). Sea $\mathbf{m}^{\mathcal{E}\mathcal{N}} \in \mathbf{R}(\mathcal{E}\mathcal{S})$. Por ser $\mathcal{E}\mathcal{S}$ reversible, existen secuencias de disparo σ, τ tales que en $\mathcal{E}\mathcal{S}$ $\mathbf{m}_0^{\mathcal{E}\mathcal{N}} \xrightarrow{\sigma} \mathbf{m}^{\mathcal{E}\mathcal{N}} \xrightarrow{\tau} \mathbf{m}_0^{\mathcal{E}\mathcal{N}}$. Sea \mathbf{m}_0^i el marcado inicial de $\mathcal{L}\mathcal{S}_i$. Por construcción, es evidente que $\mathbf{m}_0^i = \mathbf{m}_0^{\mathcal{E}\mathcal{N}}|_{P_i \cup B \cup H}$. Ahora sean $\mathbf{m}^i = \mathbf{m}^{\mathcal{E}\mathcal{N}}|_{P_i \cup H \cup B}$ (la proyección de $\mathbf{m}^{\mathcal{E}\mathcal{N}}$ sobre los lugares de $\mathcal{L}\mathcal{S}_i$), $\sigma^i = \sigma|_{T_i \cup \text{TI}}$ y $\tau^i = \tau|_{T_i \cup \text{TI}}$ (las proyecciones de σ y τ sobre las transiciones de $\mathcal{L}\mathcal{S}_i$ respectivamente). Se demostrará que $\mathbf{m}^i \in \mathbf{R}^*(\mathcal{L}\mathcal{S}_i)$ y $\sigma^i \in \mathbf{L}^*(\mathcal{L}\mathcal{S}_i)$. Por construcción de $\mathcal{E}\mathcal{S}$ es evidente que el marcado de los lugares del conjunto $B \cup H$ sólo cambia por el disparo de transiciones del conjunto TI (transiciones de interfaz), y el marcado de los lugares del conjunto P_i sólo cambia por el disparo de transiciones del conjunto T_i . Por lo tanto, en $\mathcal{L}\mathcal{S}_i$ se tiene que $\mathbf{m}_0^i \xrightarrow{\sigma^i} \mathbf{m}^i \xrightarrow{\tau^i} \mathbf{m}_0^i$, es decir, $\mathbf{m}^i \in \mathbf{R}^*(\mathcal{L}\mathcal{S}_i)$ y $\sigma^i \in \mathbf{L}^*(\mathcal{L}\mathcal{S}_i)$. Por lo tanto, queda demostrado el teorema. Los apartados referentes al esqueleto básico se demuestran de la misma forma. \diamond

Este teorema demuestra que tomando en cada sistema agregado la componente fuertemente conexa de su RG que contiene al marcado inicial se obtiene un \mathbf{RG}^* reducido cuya CTMC isomorfa es ergódica y por lo tanto tiene distribución única en es estado estacionario. Queda resuelto el problema inicial que tenían los sistemas agregados calculados con la técnica de reducción de la sección anterior.

Las componentes fuertemente conexas de un grafo dirigido se pueden calcular por medio de un algoritmo [Gib85] cuya complejidad en tiempo es de $O(\max(n, |A|))$, donde n es el número de nodos del grafo (en este caso el número de estados alcanzables de cada sistema agregado) y $|A|$ el número de aristas del grafo (en este caso el número de transiciones entre estados del RG de cada sistema agregado).

En el caso de la figura 6.1, se tenía que $\mathcal{L}\mathcal{S}_1$ tenía un bloqueo total en el marcado alcanzable B_4P_{13} . $\mathbf{RG}(\mathcal{L}\mathcal{S}_1)$ tiene 11 estados alcanzables, distribuidos en dos componentes fuertemente conexas. Una de ellas está formada por el estado B_4P_{13} que al constituir un bloqueo total es un estado absorbente de la CTMC isomorfa. La otra componente fuertemente conexa está formada por los otros 10 estados alcanzables, incluido el marcado inicial. Por lo tanto, $\mathbf{RG}^*(\mathcal{L}\mathcal{S}_1)$ consta de 10 estados alcanzables y no contiene el estado B_4P_{13} , por lo que su CTMC isomorfa será ergódica (definición 2.80). Se ha conseguido eliminar en el proceso el estado que daba problemas.

Con este resultado se tiene una técnica de descomposición general, aplicable a cualquier red de Petri estructuralmente viva, estructuralmente limitada y reversible, que permite obtener, a partir del sistema original, varias CTMC's ergódicas asociadas a sus sistemas agregados y por lo tanto susceptibles de ser analizados en su comportamiento en estado estacionario. Pero el hecho de tener CTMC's analizables no quiere decir que por ello se obtengan buenas aproximaciones.

En el caso de MG's, los sistemas agregados que se obtienen con la técnica de descomposición de la sección anterior son exactamente los mismos que los que se obtenían con la técnica expuesta en el capítulo 3 y por lo tanto se obtendrán buenas aproximaciones con la técnica numérica.

En el caso de WTS's, se puede observar que los sistemas agregados que se obtienen con la técnica de la sección anterior no preservan el concepto de resistencia de los caminos entre transiciones de interfaz que se desarrolló en el capítulo 4. Por lo tanto las aproximaciones que se obtengan con estos sistemas agregados pueden no ser buenas.

El problema reside en que $RG^*(\mathcal{LS}_i)$ y $RG^*(\mathcal{BS})$ pueden incluir todavía marcados o secuencias de disparo espurias que no se corresponden con proyecciones de marcados o secuencias de disparo del sistema original sobre los nodos preservados en cada sistema agregado. Esto es debido a que la técnica desarrollada en esta sección únicamente es capaz de eliminar los estados que pertenezcan a componentes fuertemente conexas del RG que no contengan al marcado inicial. Pero si la componente fuertemente conexa del marcado inicial contiene marcados o secuencias de disparo espurias, la técnica no puede eliminarlos.

Por ejemplo, en la figura 6.3.a se tiene un WTS. Tomando los lugares B_1 y B_2 como canales, se obtiene un SAM con dos módulos (uno con los nodos de la mitad izquierda de la red y otro con los de la mitad derecha). Si se aplica la técnica de reducción de la sección anterior, se obtiene el \mathcal{LS}_1 de la figura 6.3.b y el \mathcal{BS} de la figura 6.3.c. La CTMC isomorfa a \mathcal{LS}_1 es ergódica, por lo que $RG^*(\mathcal{LS}_1) = RG(\mathcal{LS}_1)$ (sólo tiene una componente fuertemente conexa e incluye al marcado inicial). Pero $RG^*(\mathcal{LS}_1)$ contiene marcados alcanzables y secuencias de disparo espurias. Por ejemplo, en el sistema original todos los marcados alcanzables cumplen que $\mathbf{m}[B_1] \cdot \mathbf{m}[B_2] = 0$, es decir, los dos canales no pueden tener marcas a la vez. Sin embargo, en \mathcal{LS}_1 el marcado B_1B_2 es alcanzable y no cumple la condición anterior, por lo que es espurio. Ya se vio en el capítulo 4 que este tipo de sistemas agregados en WTS's lleva a aproximaciones pobres en la técnica numérica ya que no se

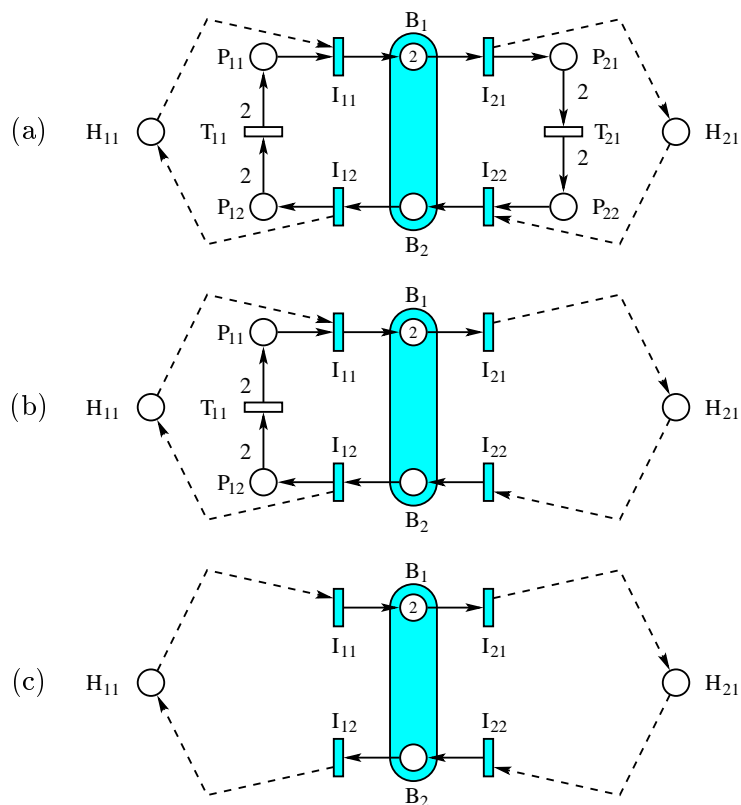


Figura 6.3: (a) Un SAM y sus (b) \mathcal{LS}_1 y (c) \mathcal{BS} .

preserva la resistencia de los caminos en la reducción.

Por lo tanto es necesario profundizar en el estudio de los RG's de los sistemas agregados para intentar en la medida de lo posible eliminar estos marcados y secuencias espurias. Para ello es necesario comprender la estructura de estos RG's, trabajo que se hará en la siguiente sección.

6.4 Visión estructurada del grafo de alcanzabilidad

En algunos casos, la técnica de descomposición desarrollada en la sección 6.2 junto con la eliminación de bloques totales de la sección 6.3 es suficiente

para obtener buenas aproximaciones con algoritmos iterativos del estilo del que se está utilizando a lo largo de esta memoria. Pero hay también casos en los que estas aproximaciones pueden ser muy pobres, como en el caso de la figura 6.3. En opinión del autor de esta memoria, los problemas se deben a dos causas.

La primera es la inclusión en las CTMC's asociadas a los sistemas agregados de marcados espurios, es decir, marcados alcanzables en los sistemas agregados que no se corresponden con proyecciones de marcados alcanzables del sistema original sobre los nodos preservados en cada sistema agregado. Estos marcados espurios aparecen porque al eliminar ciertos nodos en la construcción de los sistemas agregados, se eliminan restricciones de disparo de algunas transiciones. Al tener ciertas transiciones en los sistemas agregados con menos precondiciones para la sensibilización, aparecen secuencias de disparo y marcados espurios. En ocasiones estos marcados y secuencias de disparo espurias modifican de forma considerable el comportamiento de los sistemas agregados con respecto al del sistema original. Si luego se pretende aproximar el funcionamiento del sistema original por medio de sistemas agregados con un funcionamiento diferente, es lógico que se obtengan aproximaciones muy pobres. En la figura 6.3.a el estado B_1B_2 no es alcanzable en el sistema original pero si lo es en los sistemas agregados.

La segunda causa es la agregación inadecuada de ciertos estados diferentes del sistema original en el mismo estado de un sistema agregado. Cuando se construyen sistemas agregados lo que se busca es precisamente eso, la agregación de ciertos estados del sistema original en el mismo estado de un sistema agregado. Este tipo de agregaciones pretenden enmascarar el funcionamiento interno de un determinado módulo o subred. De esta forma, se logra reducir drásticamente el número de estados alcanzables de los sistemas agregados con respecto al del sistema original. Al construir los sistemas agregados con la técnica de la sección 6.2, aparecen agregaciones de estados que no enmascaran el funcionamiento interno de ningún módulo o subred y son estas agregaciones las que pueden provocar un funcionamiento muy diferente de los sistemas agregados con respecto al del sistema original.

Se verá mejor con un ejemplo de cada tipo de agregación. En la figura 6.3.a los estados alcanzables $H_{21}^2P_{21}^2$ y $H_{21}^2P_{22}^2$ se agregan en el \mathcal{LS}_1 en el mismo estado, el H_{21}^2 . Con esta agregación se pretende enmascarar el funcionamiento interno del módulo formado por los nodos $P_{21}T_{21}P_{22}$. De hecho en el sistema original se tiene $H_{21}^2P_{21}^2 \xrightarrow{T_{21}} H_{21}^2P_{22}^2$. Este sería un caso de agregación correcta, caracterizada por el hecho de que los dos estados que

resultarán agregados ($H_{21}^2 P_{21}^2$ y $H_{21}^2 P_{22}^2$) están conectados por una transición interna (T_{21}) que se quiere enmascarar en determinados sistemas agregados (\mathcal{LS}_1 y \mathcal{BS}).

Un ejemplo de agregación inadecuada en la figura 6.3.a es la de los estados $H_{11}H_{21}P_{11}P_{21}$ y $H_{11}H_{21}P_{12}P_{22}$. Estos se agregan en el mismo estado $H_{11}H_{21}$ en el \mathcal{BS} . En esta agregación se pierde información relevante. En el \mathcal{BS} , a partir de $H_{11}H_{21}$ pueden dispararse I_{11} o I_{22} indistintamente. En cambio, en el sistema original, desde $H_{11}H_{21}P_{11}P_{21}$ sólo puede dispararse I_{11} y desde $H_{11}H_{21}P_{12}P_{22}$ sólo puede dispararse I_{22} . Si se pretende que \mathcal{BS} resuma el comportamiento de la red original enmascarando únicamente el comportamiento interno de los módulos, desde el estado $H_{11}H_{21}$ sólo se debería poder disparar una de las dos transiciones, I_{11} si se corresponde con el estado $H_{11}H_{21}P_{11}P_{21}$ del sistema original o I_{22} si se corresponde con el estado $H_{11}H_{21}P_{12}P_{22}$. Pero resulta imposible a nivel de red distinguir estos dos estados ya que el \mathcal{BS} no tiene los lugares P_{11} , P_{21} , P_{12} y P_{22} . En este caso se produce una agregación en el \mathcal{BS} de dos estados diferentes del sistema original ($H_{11}H_{21}P_{11}P_{21}$ y $H_{11}H_{21}P_{12}P_{22}$) que no están conectados por transiciones internas. Si para pasar de un estado a otro en el sistema original es necesario disparar alguna transición de interfaz es porque no se pretende que estos estados resulten agregados en los sistemas agregados.

Los ejemplos que se han expuesto aquí con una red sencilla se pueden generalizar a redes más complejas. En general, el tipo de agregaciones de estados que se pretende realizar al construir un sistema agregado es de estados del sistema original que están conectados por transiciones internas de ciertos módulos, enmascarando de esta forma el comportamiento interno de esos módulos. Se intentará evitar otro tipo de agregaciones. Como estas agregaciones aparecen por modificar la estructura de la red original (eliminación de lugares o transiciones) la resolución de estos problemas exigirá operar a un nivel inferior, como es el nivel del RG del sistema.

En esta sección se desarrollan las bases necesarias para evitar las dos causas de problemas que se han identificado en la construcción de los sistemas agregados. En concreto lo que se hace es estudiar cómo se traduce la visión estructurada de un SAM a nivel de red en su correspondiente visión estructurada a nivel del RG, lo que permitirá evitar estas agregaciones incorrectas.

Aunque el objetivo de este estudio de la estructura del RG de un SAM era la construcción de sistemas agregados con comportamiento similar al del sistema original para su posterior utilización en técnicas de aproximación,

se han conseguido otro tipo de aplicaciones. En concreto, utilizando otro tipo de estructura de datos es posible calcular y almacenar de una forma más eficiente que la clásica todo el RG de la red original. Por lo tanto, es posible emplear esta nueva codificación del RG para otro tipo de técnicas de análisis, incluido el análisis exacto.

En secciones posteriores se empleará esta codificación para el desarrollo de sistemas agregados adecuados para técnicas de aproximación. En la sección 6.5 se generan sistemas agregados sin marcados espurios, es decir, evitando la primera causa de problemas que se ha indicado y en la sección 6.6 se generan sistemas agregados sin agregaciones inadecuadas, es decir, evitando la segunda causa de problemas.

Para entender la estructura del RG de un SAM es necesario conocer antes algunas propiedades de los grafos dirigidos etiquetados.

Definición 6.13 *Un grafo dirigido etiquetado es una terna $G = (V, L, A)$ donde V es el conjunto de vértices, L un conjunto cualquiera de etiquetas y $A \subseteq V \times V \times L$ el conjunto de aristas etiquetadas.*

Un grafo dirigido etiquetado no es más que un grafo dirigido al que se le añade etiquetas a las aristas. Para ello, formalmente hace falta añadir un conjunto de etiquetas y asociar una etiqueta a cada arista.

El RG de un SAM \mathcal{S} puede verse como un grafo dirigido etiquetado cuyo conjunto de vértices es el conjunto $R(\mathcal{S})$ de estados alcanzables, el conjunto de etiquetas es el conjunto de transiciones del SAM y el conjunto de aristas son las secuencias de disparo de longitud 1 entre vértices, de forma que $\mathbf{m}_1 \xrightarrow{t} \mathbf{m}_2$ si y sólo si $(\mathbf{m}_1, \mathbf{m}_2, t) \in A$.

Notación 6.14 *Sea $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM con K módulos y $RG(\mathcal{S})$ su RG interpretado como grafo dirigido etiquetado. Si $s \in R(\mathcal{S})$ es un vértice de $RG(\mathcal{S})$ que representa a un marcado alcanzable \mathbf{m} , se denotará por $M(s)$ al marcado alcanzable \mathbf{m} .*

Definición 6.15 *Sean $G = (V, L, A)$ y $G' = (V', L', A')$ dos grafos dirigidos etiquetados. G' se dice subgrafo dirigido etiquetado de G (denotado por $G' \subseteq G$) si $V' \subseteq V$, $L' \subseteq L$ y $A' \subseteq A$.*

Definición 6.16 *Sea $G = (V, L, A)$ un grafo dirigido etiquetado y $V' \subseteq V$ un subconjunto de vértices. El grafo dirigido etiquetado generado por V' es $G(V') = (V', L', A')$ donde $L' = \{l \in L \mid \exists (v_1, v_2, l) \in A \text{ tal que } v_1, v_2 \in V'\}$ y $A' = A \cap (V' \times V' \times L')$.*

La siguiente proposición es evidente a partir de las definiciones anteriores.

Proposición 6.17 *Sea $G = (V, L, A)$ un grafo dirigido etiquetado, $V' \subseteq V$ un subconjunto de vértices suyo y $G' = G(V')$ el grafo dirigido etiquetado generado por V' . Entonces G' es subgrafo de G .*

Demostración:

Evidente por la definiciones 6.15 y 6.16. \diamond

Si G es un grafo dirigido etiquetado y V' un subconjunto de vértices de G , $G(V')$ es el subgrafo dirigido etiquetado de G formado por el conjunto V' de vértices y las aristas etiquetadas que los conectan.

Definición 6.18 *Sea $G = (V, A)$ un grafo dirigido y $v_1, v_2 \in V$ dos vértices de G . Se dice que v_2 es vértice sucesor de v_1 si existen aristas $\{a_i\}_{i=1}^n \in A$ tales que si $a_i = (v_i^1, v_i^2) \in V \times V$ para $1 \leq i \leq n$, entonces $v_1^1 = v_1$, $v_n^2 = v_2$ y $v_i^2 = v_{i+1}^1$ para $1 \leq i < n$.*

Un vértice v_2 es sucesor de otro vértice v_1 si existe un camino dirigido de v_1 a v_2 formado por aristas del grafo dirigido. El concepto de vértice sucesor se puede aplicar también a grafos dirigidos etiquetados con solo añadir las etiquetas a las aristas.

Definición 6.19 *Sea $G = (V, A)$ un grafo dirigido y $v \in V$ un vértice de G . Se dice que v es un vértice cabecera de G si no es sucesor de ningún otro vértice del grafo dirigido.*

El concepto de vértice cabecera se puede aplicar también a grafos dirigidos etiquetados. En el caso de un SAM reversible, no hay ningún vértice cabecera en su RG ya que éste es fuertemente conexo.

Definición 6.20 *Sean $G_1 = (V_1, L_1, A_1)$ y $G_2 = (V_2, L_2, A_2)$ dos grafos dirigidos etiquetados. El grafo producto de G_1 y G_2 es el grafo dirigido etiquetado $G_1 \times G_2 = (V, L, A)$ tal que:*

- i) $V = V_1 \times V_2$.
- ii) $L = L_1 \cup L_2$.
- iii) $A = \{((s_1, s_2), (s'_1, s'_2), l) \in V \times V \times L \mid (s_2 = s'_2 \text{ y } (s_1, s'_1, l) \in A_1) \text{ o } (s_1 = s'_1 \text{ y } (s_2, s'_2, l) \in A_2))\}$.

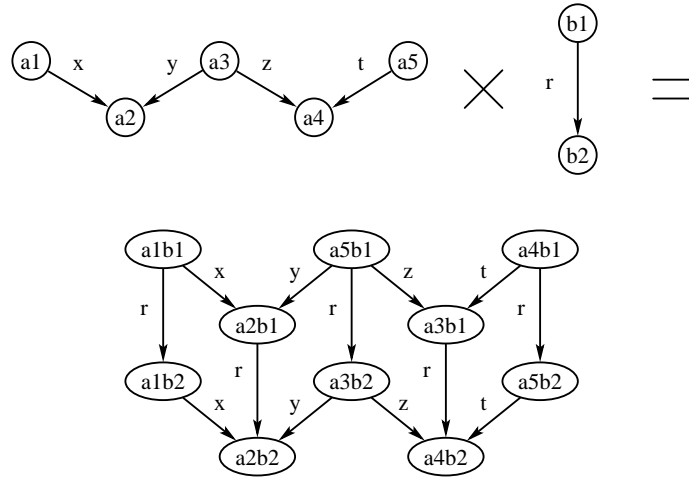


Figura 6.4: Producto de grafos dirigidos etiquetados.

La figura 6.4 muestra el producto de dos grafos dirigidos etiquetados.

Definición 6.21 Sean $G_1 = (V_1, L_1, A_1)$ y $G_2 = (V_2, L_2, A_2)$ dos grafos dirigidos etiquetados. El grafo unión de G_1 y G_2 es el grafo dirigido etiquetado $G_1 \cup G_2 = (V, L, A)$ con $V = V_1 \cup V_2$, $L = L_1 \cup L_2$ y $A = A_1 \cup A_2$.

Definición 6.22 Sean $G_1 = (V_1, L_1, A_1)$ y $G_2 = (V_2, L_2, A_2)$ dos grafos dirigidos etiquetados. Se dice que el grafo G_1 es isomorfo a G_2 (denotado por $G_1 \cong G_2$) si existen $f : V_1 \rightarrow V_2$, $g : L_1 \rightarrow L_2$ biyecciones tales que $(v, v', t) \in A_1$ si y sólo si $(f(v), f(v'), g(t)) \in A_2$.

Definición 6.23 Sea \mathcal{S} un SAM con $T = \bigcup_{i=1}^K T_i$. Sea R_i con $1 \leq i \leq K$ la siguiente relación definida sobre $R(\mathcal{S})$: Dados s_1, s_2 vértices de $RG(\mathcal{S})$, $(s_1, s_2) \in R_i$ si y sólo si existe un camino no dirigido π en $RG(\mathcal{S})$ que conecta s_1 con s_2 formado por aristas de etiquetas en $T \setminus (T_i \cup TI)$. Sea R_0 la siguiente relación definida sobre $R(\mathcal{S})$: Dados s_1, s_2 vértices de $RG(\mathcal{S})$, $(s_1, s_2) \in R_0$ si y sólo si existe un camino no dirigido π en $RG(\mathcal{S})$ que conecta s_1 con s_2 formado por aristas de etiquetas en $T \setminus TI$.

Proposición 6.24 Sea \mathcal{S} un SAM con K módulos y R_i con $0 \leq i \leq K$ las relaciones de la definición 6.23 sobre $R(\mathcal{S})$. Entonces R_i es una relación de equivalencia para $0 \leq i \leq K$.

Demostración:

Se hará la demostración para R_i con $1 \leq i \leq K$. Para R_0 la demostración es igual.

Reflexiva: $(s, s) \in R_i$ para todo $s \in R(\mathcal{S})$ ya que s está conectado consigo mismo por medio del camino vacío que no contiene aristas de etiquetas en $T_i \cup \text{TI}$.

Simétrica: Si $(s_1, s_2) \in R_i$, sea π_1 un camino no dirigido en $\text{RG}(\mathcal{S})$ que conecta s_1 y s_2 formado por aristas de etiquetas en $T \setminus (T_i \cup \text{TI})$. Entonces π_1 recorrido en sentido contrario es un camino no dirigido en $\text{RG}(\mathcal{S})$ que conecta s_2 y s_1 formado por aristas de etiquetas en $T \setminus (T_i \cup \text{TI})$, es decir, $(s_2, s_1) \in R_i$.

Transitiva: Si $(s_1, s_2), (s_2, s_3) \in R_i$, sean π_1, π_2 caminos no dirigidos en $\text{RG}(\mathcal{S})$ que conectan s_1 con s_2 y s_2 con s_3 respectivamente, formados por aristas de etiquetas en $T \setminus (T_i \cup \text{TI})$. Entonces $\pi_1 \circ \pi_2$ es un camino no dirigido en $\text{RG}(\mathcal{S})$ que conecta s_1 con s_3 formado por aristas de etiquetas en $T \setminus (T_i \cup \text{TI})$, es decir, $(s_1, s_3) \in R_i$. \diamond

La siguiente propiedad es inmediata a partir de la definición 6.23.

Propiedad 6.25 *Sea \mathcal{S} un SAM con K módulos y R_i con $0 \leq i \leq K$ las relaciones de equivalencia de la definición 6.23. Entonces, para cada $1 \leq i \leq K$ se tiene que $R_i \subseteq R_0$. Por lo tanto, si D_i es una clase de equivalencia de R_i entonces existe D_0 clase de equivalencia de R_0 tal que $D_i \subseteq D_0$.*

Las relaciones de equivalencia que se acaban de definir inducen particiones en $R(\mathcal{S})$. En la siguiente proposición se demostrarán las principales características de estas clases de equivalencia.

Proposición 6.26 *Sea \mathcal{S} un SAM con K módulos y R_i con $0 \leq i \leq K$ las relaciones de equivalencia de la definición 6.23. Sea D_i una clase de equivalencia de R_i . Entonces se cumple:*

1. Si $s_1, s_2 \in D_0$ entonces $M(s_1)|_B = M(s_2)|_B$.
2. Si $s_1, s_2 \in D_i$ entonces $M(s_1)|_{P_i \cup B} = M(s_2)|_{P_i \cup B}$ para $1 \leq i \leq K$.

Demostración:

1) Por definición de R_0 existe un camino no dirigido $\pi = a_1 \cdots a_n$ con a_j de etiqueta $t_j \in T \setminus \text{TI}$. Sean $\{q_i\}_{i=0}^n$ los vértices que atraviesa el camino π de tal forma que $q_0 = s_1$ y $q_n = s_2$. Si se recorre el camino no dirigido π desde

el vértice s_1 al vértice s_2 se induce un determinado sentido al recorrido de sus aristas (la arista a_j va del vértice q_{j-1} al q_j para $1 \leq j \leq n$). El sentido inducido en las aristas de π puede coincidir o no con la dirección de la arista en el grafo dirigido etiquetado $\text{RG}(\mathcal{S})$. Si el sentido inducido en a_j coincide con el de $\text{RG}(\mathcal{S})$ es porque $M(q_{j-1}) \xrightarrow{t_j} M(q_j)$ y si el sentido inducido en a_j es el contrario al de $\text{RG}(\mathcal{S})$ es porque $M(q_j) \xrightarrow{t_j} M(q_{j-1})$. Las transiciones del conjunto $T \setminus \text{TI}$ tienen todos sus lugares de entrada y de salida en $P \setminus B$ por lo que su disparo no modifica el marcado de lugares del conjunto B . Si $M(q_{j-1}) \xrightarrow{t_j} M(q_j)$ o $M(q_j) \xrightarrow{t_j} M(q_{j-1})$ entonces se tiene que $M(q_{j-1})|_B = M(q_j)|_B$ para todo $1 \leq j \leq n$, en particular $M(s_1)|_B = M(s_2)|_B$.

2) Por definición de R_i existe un camino no dirigido $\pi = a_1 \cdots a_n$ con a_j de etiqueta $t_j \in T \setminus (T_i \cup \text{TI})$. Sean $\{q_i\}_{i=0}^n$ los vértices que atraviesa el camino π de tal forma que $q_0 = s_1$ y $q_n = s_2$. Como en el apartado 1, esto quiere decir que $M(q_{j-1}) \xrightarrow{t_j} M(q_j)$ o $M(q_j) \xrightarrow{t_j} M(q_{j-1})$ para todo $1 \leq j \leq n$. Ahora las transiciones del conjunto $T \setminus (T_i \cup \text{TI})$ tienen todos sus lugares de entrada y de salida en $P \setminus (P_i \cup B)$ por lo que su disparo no modifica el marcado de lugares del conjunto $P_i \cup B$. Si $M(q_{j-1}) \xrightarrow{t_j} M(q_j)$ o $M(q_j) \xrightarrow{t_j} M(q_{j-1})$ entonces se tiene que $M(q_{j-1})|_{P_i \cup B} = M(q_j)|_{P_i \cup B}$ para todo $1 \leq j \leq n$, en particular $M(s_1)|_{P_i \cup B} = M(s_2)|_{P_i \cup B}$. \diamond

Esta proposición demuestra que la relación de equivalencia R_0 induce una partición en $\text{R}(\mathcal{S})$ en la que todos los estados de cada clase de equivalencia tienen el mismo marcado en el conjunto B de canales, enmascarando de esta forma el estado interno de todos los módulos del SAM. Esta es la tarea que se pretende realizar con el \mathcal{BS} en las técnicas de descomposición desarrolladas en esta memoria.

Por su parte, la relación de equivalencia R_i con $1 \leq i \leq K$ induce una partición en $\text{R}(\mathcal{S})$ en la que todos los estados de cada clase de equivalencia tienen el mismo marcado en los lugares del conjunto $P_i \cup B$ (canales y lugares internos al módulo \mathcal{N}_i) enmascarando de esta forma el estado interno de todos los módulos del SAM, excepto el \mathcal{N}_i . Esta es la tarea que se pretende realizar con el \mathcal{LS}_i en las técnicas de descomposición desarrolladas en esta memoria.

Por lo tanto, las relaciones de equivalencia R_i con $1 \leq i \leq K$ representan el tipo de agregación de estados que se pretende realizar con la construcción de los distintos sistemas agregados en las técnicas de descomposición desarrolladas en esta memoria. Esta agregación sólo es posible realizarla

de forma exacta a nivel de red en el caso de MG's. Para redes de Petri en general es necesario bajar al nivel del RG para obtener la agregación que se pretende. Ahora se van a utilizar estas relaciones de equivalencia para definir nuevos grafos dirigidos etiquetados que se asociarán con los sistemas agregados de un SAM.

Definición 6.27 Sea \mathcal{S} un SAM con $T = \bigcup_{i=1}^K T_i$, $G = (V, L, A)$ su RG interpretado como grafo dirigido etiquetado y R_i con $0 \leq i \leq K$ las relaciones de equivalencia de la definición 6.23. Para $0 \leq i \leq K$ se define el siguiente grafo dirigido etiquetado agregado $G_i = (V_i, L_i, A_i)$ (se toma $T_0 = \emptyset$):

- i) $V_i = V/R_i$ (conjunto cociente).
- ii) $L_i = T_i \cup TI$.
- iii) $A_i = \{(s_1, s_2, t) \in V_i \times V_i \times L_i \mid \exists v_1 \in s_1, v_2 \in s_2 \text{ tal que } (v_1, v_2, t) \in A\}$.

El grafo dirigido etiquetado G_i tiene como conjunto de vértices el conjunto de clases de equivalencia de $R(\mathcal{S})$ respecto a la relación de equivalencia R_i . Las aristas de A_i están formadas por un par ordenado de elementos de V_i para los que existe una arista en G con un representante de cada clase. En ese caso se mantiene la etiqueta de la arista original en el grafo agregado.

El grafo dirigido etiquetado G_i con $1 \leq i \leq K$ de la definición anterior constituye la interpretación como grafo dirigido etiquetado del RG que se pretende conseguir con el sistema agregado \mathcal{LS}_i del SAM. Por su parte, el grafo dirigido etiquetado G_0 constituye la interpretación como grafo dirigido etiquetado del RG que se pretende conseguir con el \mathcal{BS} del SAM. No es posible conseguir en general ningún RG cuyo comportamiento en términos de marcados alcanzables y secuencias de disparo se parezca más al del modelo original a la vez que se reduce su tamaño.

Por lo tanto, el siguiente objetivo será calcular los grafos dirigidos etiquetados G_i con $0 \leq i \leq K$ de la definición anterior sin construir a nivel de red ningún sistema agregado, es decir, directamente a partir del modelo original. Esto es posible si se estudia la relación que hay entre las aristas del RG del SAM original y las de los grafos G_i con $0 \leq i \leq K$.

Notación 6.28 Sea \mathcal{S} un SAM con K módulos y R_i, G_i con $0 \leq i \leq K$ las relaciones de equivalencia y los grafos dirigidos etiquetados agregados de las definiciones 6.23 y 6.27. Sea $s \in V_i = R(\mathcal{S})/R_i$ y $v \in R(\mathcal{S})$ tal que $v \in s$. Se denotará por $M(s)$ a $M(v)|_{P_i \cup B}$ (se toma $P_0 = \emptyset$).

Demostración:

Esta notación es correcta, ya que por la proposición 6.26 todos los estados que pertenecen a una misma clase de equivalencia de R_i con $0 \leq i \leq K$ tienen el mismo marcado local en $P_i \cup B$. Se asociará como marcado de la clase s de R_i al marcado local en $P_i \cup B$ de cualquier representante suyo. \diamond

Para facilitar la comprensión de los teoremas de estructura del RG se demostrará siguiente resultado.

Proposición 6.29 *Sea \mathcal{S} un SAM con K módulos y $\mathbf{m} \in R(\mathcal{S})$ tal que $t \in T_i \setminus \text{TI}$ y $t' \in T_j \setminus \text{TI}$ con $1 \leq i, j \leq K$ y $j \neq i$ están sensibilizadas en \mathbf{m} . Sean $\mathbf{m} \xrightarrow{t} \mathbf{m}_i$ y $\mathbf{m} \xrightarrow{t'} \mathbf{m}_j$. Entonces se cumple:*

1. t' está sensibilizada en \mathbf{m}_i .
2. t está sensibilizada en \mathbf{m}_j .
3. Si $\mathbf{m}_i \xrightarrow{t'} \mathbf{m}_{ij}$ y $\mathbf{m}_j \xrightarrow{t} \mathbf{m}_{ji}$ entonces $\mathbf{m}_{ij} = \mathbf{m}_{ji}$.

Demostración:

Por ser $t \in T_i \setminus \text{TI}$, $\bullet t \cup t \bullet \subseteq P_i$. Análogamente, por ser $t' \in T_j \setminus \text{TI}$, $\bullet t' \cup t' \bullet \subseteq P_j$. Como $i \neq j$ entonces $P_i \cap P_j = \emptyset$, luego t y t' son concurrentes, es decir, el disparo de una no desensibiliza la otra (lo que demuestra los apartados 1 y 2) y pueden dispararse en cualquier orden (lo que demuestra el apartado 3). \diamond

Esta proposición no hace más que explotar la concurrencia existente entre las transiciones internas de los distintos módulos de un SAM. Cuando a una red de Petri se le da una visión estructurada de SAM automáticamente se tienen conjuntos de transiciones que son concurrentes entre sí. Dos transiciones internas cualesquiera que pertenezcan a módulos diferentes de un SAM son concurrentes entre sí. Este resultado puede extenderse sin problemas a secuencias de disparo internas a cada módulo del SAM. Por lo tanto, esta proposición demuestra que en un SAM es posible reordenar dos transiciones cualesquiera consecutivas de una secuencia de disparo siempre que estas dos transiciones pertenezcan a dos módulos diferentes del SAM (ya que en ese caso son concurrentes). Esta concurrencia entre transiciones a nivel de red se traduce, a nivel del RG, en una estructura de producto de grafos dirigidos etiquetados. Esta estructura producto se demuestra en los dos siguientes resultados.

Proposición 6.30 *Sea \mathcal{S} un SAM con K módulos y R_i, G_i con $0 \leq i \leq K$ las relaciones de equivalencia y los grafos dirigidos etiquetados agregados de las definiciones 6.23 y 6.27. Sea D el grafo dirigido etiquetado generado por una clase de equivalencia de R_0 , h un vértice cabecera de D y D_h el grafo dirigido etiquetado generado por los sucesores de h en D . Sea D_h^i con $1 \leq i \leq K$ el grafo dirigido etiquetado generado en G_i por los vértices de G_i que pertenecen a D_h . Entonces $D_h \cong D_h^1 \times \cdots \times D_h^K$.*

Demostración:

Se denotará por $D = (V_D, L_D, A_D)$ al grafo dirigido etiquetado D . Sea $M(h)$ el marcado asociado al vértice $h \in V_D$. Se denotará por h^0 a $M(h)|_B$ (el marcado de h en los canales) y por h^i con $1 \leq i \leq K$ a $M(h)|_{P_i}$ (el marcado local de h en cada módulo del SAM). Entonces $M(h) = (h^0, h^1, \dots, h^K)$.

D_h es el grafo dirigido etiquetado $D_h = (V_h, L_h, A_h)$ con:

$$V_h = \{s \in V_D \mid \exists \sigma \subseteq T \setminus \text{TI} \text{ secuencia tal que } M(h) \xrightarrow{\sigma} M(s) \text{ en } \text{RG}(\mathcal{S})\}.$$

$$L_h = T \setminus \text{TI}.$$

$$A_h = \{(s_1, s_2, t) \mid s_1, s_2 \in V_h, t \in T \setminus \text{TI} \text{ tal que } M(s_1) \xrightarrow{t} M(s_2) \text{ en } \text{RG}(\mathcal{S})\}.$$

Para cada $1 \leq i \leq K$, sea \mathcal{S}'_i el subsistema de \mathcal{S} generado por $P_i \cup B \cup T_i$.

Entonces $\mathcal{S}'_i = \langle \mathcal{N}'_i, h^{i'} \rangle$ donde $h^{i'} = (h^0, h^i)$ y $\mathcal{N}'_i = \langle P_i \cup B, T_i, F_i, W_i \rangle$ con $F_i = F \cap (((P_i \cup B) \times T_i) \cup (T_i \times (P_i \cup B)))$ y $W_i = W|_{F_i}$. El propósito de

añadir los canales es para incluir en sus estados alcanzables el marcado de los mismos. Se denotará a G_i por $G_i = (V_{G_i}, L_{G_i}, A_{G_i})$. En estas condiciones,

D_h^i con $1 \leq i \leq K$ es el grafo dirigido etiquetado $D_h^i = (V_h^i, L_h^i, A_h^i)$ con:

$$V_h^i = \{s \in V_{G_i} \mid \exists \sigma \subseteq T_i \setminus \text{TI} \text{ secuencia tal que } h^{i'} \xrightarrow{\sigma} M(s) \text{ en } \text{RG}(\mathcal{S}'_i)\}.$$

$$L_h^i = T_i \setminus \text{TI}.$$

$$A_h^i = \{(s_1, s_2, t) \mid s_1, s_2 \in V_h^i, t \in T_i \setminus \text{TI} \text{ tal que } M(s_1) \xrightarrow{t} M(s_2) \text{ en } \text{RG}(\mathcal{S}'_i)\}.$$

Dado $s \in V_h$ se denotará por s^i a $M(s)|_{P_i}$ para $0 \leq i \leq K$ (tomando $P_0 = B$). Entonces $M(s) = (s^0, s^1, \dots, s^K)$ donde $s^0 = h^0$ (ya que todos los vértices de V_h están en la misma clase de equivalencia de R_0). A partir de s se pueden construir s_i con $1 \leq i \leq K$ tales que $M(s_i) = (h^0, s^i)$.

Sea $f : V_h \rightarrow V_h^1 \times \cdots \times V_h^K$ con $f(s) = (s_1, \dots, s_K)$ y $g : L_h \rightarrow \bigcup_{i=1}^K L_h^i$ la aplicación identidad. Ver que $G_h \cong G_1 \times \cdots \times G_K$ por medio de f y g (ver definición 6.22).

Es evidente que g es biyectiva ya que $L_h = T \setminus \text{TI} = (\bigcup_{i=1}^K T_i) \setminus \text{TI} = \bigcup_{i=1}^K (T_i \setminus \text{TI}) = \bigcup_{i=1}^K L_h^i$.

Ver que f está bien definida, es decir, $s \in V_h \Rightarrow s_i \in V_h^i$ para $1 \leq i \leq K$.

Si $s \in V_h$ existe $\sigma \subseteq T \setminus \text{TI}$ tal que $M(h) \xrightarrow{\sigma} M(s)$ en $\text{RG}(\mathcal{S})$, es decir, $(h^0, \dots, h^K) \xrightarrow{\sigma} (s^0, \dots, s^K)$ en $\text{RG}(\mathcal{S})$. Sea $\sigma_i = \sigma|_{T_i}$ para $1 \leq i \leq K$.

Por la proposición 6.29, se puede considerar sin pérdida de generalidad que $\sigma = \sigma_1 \cdots \sigma_K$. Ahora, $\sigma_i \subseteq T_i \setminus \text{TI}$ y $(h^0, h^i) \xrightarrow{\sigma_i} (s^0, s^i)$ en $\text{RG}(\mathcal{S}'_i)$, es decir, $h^{i'} \xrightarrow{\sigma_i} M(s_i)$ en $\text{RG}(\mathcal{S}'_i)$, luego $s_i \in V_h^i$ para $1 \leq i \leq K$. Por lo tanto, f está bien definida.

Ver que f es biyectiva.

Sean $u, v \in V_h$ tales que $f(u) = f(v)$. Entonces $u_i = v_i$ para $1 \leq i \leq K$. Además $u_0 = v_0 = h_0$, luego $M(u_i) = M(v_i)$ para $0 \leq i \leq K$, por lo que $u^i = v^i$ para $0 \leq i \leq K$, es decir, $M(u) = M(v)$ y por lo tanto $u = v$. Entonces f es inyectiva.

Para ver que f es sobreyectiva, sea $(s_1, \dots, s_K) \in V_h^1 \times \cdots \times V_h^K$. Entonces $s_i \in V_h^i$, es decir, existe $\sigma_i \subseteq T_i \setminus \text{TI}$ tal que $h^{i'} \xrightarrow{\sigma_i} M(s_i)$ en $\text{RG}(\mathcal{S}'_i)$, luego $M(s_i)|_B = h^0$ para $1 \leq i \leq K$. Sea $s^i = M(s_i)|_{P_i}$ para $1 \leq i \leq K$ y s tal que $M(s) = (h^0, s^1, \dots, s^K)$. Por construcción de s es claro que $f(s) = (s_1, \dots, s_k)$. Falta ver que $s \in V_h$. Sea $\sigma = \sigma_1 \cdots \sigma_K$. Es claro que $\sigma \in T \setminus \text{TI}$ y como $h^{i'} \xrightarrow{\sigma_i} M(s_i)$ en $\text{RG}(\mathcal{S}'_i)$ para $1 \leq i \leq K$, entonces $M(h) \xrightarrow{\sigma} M(s)$ en $\text{RG}(\mathcal{S})$ (las σ_i son concurrentes entre sí) luego $s \in V_h$.

Ver que $(u, v, t) \in A_h \Leftrightarrow (f(u), f(v), g(t))$ es arista de $D_h^1 \times \cdots \times D_h^K$.

Siguiendo la misma notación, $f(u) = (u_1, \dots, u_K)$ y $f(v) = (v_1, \dots, v_K)$. Por lo tanto, $(f(u), f(v), g(t))$ es arista de $D_h^1 \times \cdots \times D_h^K$ si y sólo si existe $1 \leq i \leq K$ tal que $u_j = v_j$ si $i \neq j$ y $(u_i, v_i, t) \in A_h^i$ (ya que $g(t) = t$). Entonces hay que demostrar que $(u, v, t) \in A_h$ si y sólo si existe $1 \leq i \leq K$ tal que $u_j = v_j$ si $i \neq j$ y $(u_i, v_i, t) \in A_h^i$. Por ser $u, v \in V_h$ entonces $M(u) = (h^0, u^1, \dots, u^K)$ y $M(v) = (h^0, v^1, \dots, v^K)$. Como $(u, v, t) \in A_h$ entonces $t \in T \setminus \text{TI}$. Sea i tal que $t \in T_i \setminus \text{TI}$. En estas condiciones, $(u, v, t) \in A_h$ si y sólo si $M(u) \xrightarrow{t} M(v)$ en $\text{RG}(\mathcal{S})$. Por ser $t \in T_i \setminus \text{TI}$, esto ocurre si y sólo si $u^j = v^j$ para $i \neq j$ y $u^i \xrightarrow{t} v^i$ en $\text{RG}(\mathcal{S}'_i)$ si y sólo si $M(u_j) = M(v_j)$ para $i \neq j$ y $M(u_i) \xrightarrow{t} M(v_i)$ en $\text{RG}(\mathcal{S}'_i)$ si y sólo si $u_j = v_j$ para $i \neq j$ y $(u_i, v_i, t) \in A_h^i$. \diamond

Se va a dar una explicación en términos de redes de Petri del resultado que demuestra esta proposición. Se sabe que R_0 induce una partición en el conjunto $R(\mathcal{S})$. Una clase de equivalencia de R_0 está formada por un conjunto de estados alcanzables conectados entre sí por transiciones internas del SAM (sin importar la orientación de las aristas). D es el grafo dirigido etiquetado generado por esta clase de equivalencia, es decir, es un subgrafo dirigido etiquetado de $\text{RG}(\mathcal{S})$ formado por todos los estados de la clase de equivalencia junto con las transiciones entre ellos (que son transiciones internas del SAM por definición de R_0). La estructura completa de D se deja

para el siguiente resultado. En éste se toma un vértice cabecera h cualquiera de D . Los vértices cabecera de D no son más que estados de $\text{RG}(\mathcal{S})$ que sólo son alcanzables por el disparo de transiciones de interfaz. Los sucesores de h en D son estados alcanzables a partir de h debido únicamente al disparo de transiciones internas del SAM. Todos estos estados generan un grafo dirigido etiquetado que en la proposición se denota por D_h . Si se agrega el grafo dirigido etiquetado D_h por medio de la relación de equivalencia R_i para $1 \leq i \leq K$ se obtiene el grafo dirigido etiquetado D_h^i , que es un subgrafo del grafo agregado G_i . La proposición relaciona la estructura de D_h con la de los grafos D_h^i con $1 \leq i \leq K$. La ventaja de tener esta relación estriba en que ya no es necesario calcular todo el D_h como se haría en el algoritmo clásico, sino que es suficiente generar los D_h^i . La generación del grafo dirigido etiquetado D_h^i se puede hacer localmente en el módulo \mathcal{N}_i del SAM (para ello basta conocer el marcado inicial local $M(h)|_{P_i}$ y la estructura del módulo \mathcal{N}_i). Este hecho supone una rebaja de la complejidad tanto en tiempo como en espacio del problema del cálculo de D_h , ya que no es necesario calcular todas las posibles combinaciones de estados locales de cada módulo entre sí. Además, en distintas clases de equivalencia de R_0 se puede tener localmente en alguno de los módulos una clase local repetida, en cuyo caso tampoco será necesario repetir esos cálculos.

Esta proposición explota la concurrencia entre transiciones internas de un SAM permitiendo calcular el grafo dirigido etiquetado D_h a partir de los grafos dirigidos etiquetados locales en cada módulo del SAM. De esta forma sólo es necesario calcular y almacenar estos sucesores locales en cada módulo, rebajando de forma considerable la complejidad tanto en espacio como en tiempo de la generación de D_h .

El siguiente teorema demuestra la estructura completa de una clase de equivalencia de R_0 , que se reduce a repetir los cálculos sobre cada vértice cabecera de la clase de equivalencia.

Teorema 6.31 *Sea \mathcal{S} un SAM con K módulos y R_i, G_i con $0 \leq i \leq K$ las relaciones de equivalencia y los grafos dirigidos etiquetados agregados de las definiciones 6.23 y 6.27. Sea D el grafo dirigido etiquetado generado por una clase de equivalencia de R_0 , $\{h_i\}_{i=1}^n$ el conjunto de vértices cabecera de D y D_{h_i} con $1 \leq i \leq n$ el grafo dirigido etiquetado generado por los sucesores de h_i en D . Sea $D_{h_i}^j$ con $1 \leq i \leq n$ y $1 \leq j \leq K$ el grafo dirigido etiquetado generado en G_j por los vértices de G_j que pertenecen a D_{h_i} . Entonces $D \cong \bigcup_{i=1}^n D_{h_i}^1 \times \cdots \times D_{h_i}^K$.*

La demostración es evidente a partir de la proposición 6.30. Notar que distintos vértices cabecera de D pueden compartir localmente en algún módulo parte de los grafos dirigidos etiquetados generados por sus sucesores. En estos casos sólo es necesario calcular y almacenar una sola vez los vértices, etiquetas y aristas locales.

Con este teorema queda caracterizado el conjunto de transiciones internas entre estados alcanzables de un SAM pudiéndose calcular localmente en cada módulo del mismo. A partir de estos cálculos locales (mucho más eficientes tanto en espacio como en tiempo) es posible reconstruir todas las transiciones internas del sistema original por medio de productos y uniones de grafos dirigidos etiquetados. Además de esta forma también se calculan todas las transiciones internas en cada grafo agregado.

Para completar la relación existente entre el $\text{RG}(\mathcal{S})$ del SAM interpretado como grafo dirigido etiquetado y sus grafos agregados (que se pretende asociar a sus sistemas agregados), falta conocer la relación entre las transiciones de interfaz que aparecen en los grafos agregados. Este es el propósito del siguiente teorema.

Teorema 6.32 *Sea \mathcal{S} un SAM con K módulos y R_i, G_i con $0 \leq i \leq K$ las relaciones de equivalencia y los grafos dirigidos etiquetados agregados de las definiciones 6.23 y 6.27. Sea D el grafo dirigido etiquetado generado por una clase de equivalencia de R_0 , h un vértice cabecera de D y D_h el grafo dirigido etiquetado generado por los sucesores de h en D . Sea $D_h^i = (V_h^i, L_h^i, A_h^i)$ con $1 \leq i \leq K$ el grafo dirigido etiquetado generado en $G_i = (V_i, L_i, A_i)$ por los vértices de G_i que pertenecen a D_h . Sea $s \in V_h^i$. En estas condiciones, existe una arista de la forma $(s, \cdot, t) \in A_i$ con $t \in T_j \cap \text{TI}$, $i \neq j$ si y sólo si existe una arista de la forma $(s', \cdot, t) \in A_j$ con $s' \in V_h^j$.*

Demostración:

Se denotará a D_h por $D_h = (V_h, L_h, A_h)$ y por $G = (V, L, A)$ a $\text{RG}(\mathcal{S})$ interpretado como grafo dirigido etiquetado. Como $t \in \text{TI}$, entonces $t \in L_i$ para $1 \leq i \leq K$.

Por la definición 6.27, como $t \in L_i$, $(s, \cdot, t) \in A_i$ si y sólo si existe $(m, \cdot, t) \in A$ con $m \in s$.

Sea s' la clase de equivalencia de m en R_j . Se tiene que $s \in V_h^i$ si y sólo si $s \subseteq V_h$ (por ser $V_h^i = V_h/R_i$) si y sólo si $m \in V_h$ (por ser $m \in s$) si y sólo si $s' \subseteq V_h$ (por ser $m \in s'$) si y sólo si $s' \in V_h^j$ (por ser $V_h^j = V_h/R_j$ y s' clase de R_j). Por lo tanto, se ha demostrado que $s \in V_h^i$ si y sólo si $s' \in V_h^j$.

Ahora, por la definición 6.27, como $m \in s'$ y $t \in L_j$, $(m, \cdot, t) \in A$ si y sólo si $(s', \cdot, t) \in A_j$ con $s' \in V_h^j$. \diamond

Se va a dar una explicación en términos de redes de Petri del resultado que demuestra este teorema. La relación de equivalencia R_0 induce una partición en $R(\mathcal{S})$. Una clase de equivalencia de R_0 está formada por un conjunto de estados alcanzables conectados entre sí por transiciones internas del SAM. D es el grafo dirigido etiquetado generado por esta clase de equivalencia, es decir, es un subgrafo de $RG(\mathcal{S})$ formado por todos los estados de la clase de equivalencia junto con las transiciones internas entre ellos (por definición de R_0). En D se toma un vértice cabecera h cualquiera. Los vértices cabecera de D no son más que estados de $RG(\mathcal{S})$ que sólo son alcanzables por el disparo de transiciones de interfaz. Los sucesores de h en D son estados alcanzables a partir de h debido al disparo de transiciones internas del SAM. Todos estos estados y las transiciones entre ellos generan D_h . Si se agrega el grafo dirigido etiquetado D_h mediante R_i para $1 \leq i \leq K$ se obtiene D_h^i , que es un subgrafo del grafo agregado G_i (definición 6.27). El teorema relaciona las transiciones de interfaz sensibilizadas en los estados de D_h^i con las transiciones de interfaz sensibilizadas en los estados de D_h^j donde $1 \leq i, j \leq K$ y $i \neq j$. Con la relación obtenida, ya no es necesario calcular en cada estado de D_h^i todas las transiciones de interfaz sensibilizadas como se haría en el algoritmo clásico, sino que es suficiente con calcular las transiciones de interfaz de $T_i \cap TI$ sensibilizadas. Esas transiciones también estarán sensibilizadas en todos los estados de D_h^j para $i \neq j$. Este cálculo puede hacerse localmente en el módulo \mathcal{N}_i del SAM extendido con sus canales de entrada y salida (para ello basta conocer el marcado en $P_i \cup B$ y la estructura del módulo extendido). Este hecho supone una rebaja de la complejidad tanto en espacio como en tiempo del problema del cálculo de las transiciones de interfaz en $RG(\mathcal{S})$ o en los grafos agregados G_i con $1 \leq i \leq K$. Además, en distintas clases de equivalencia de R_0 se puede tener localmente en alguno de los módulos una clase local repetida, en cuyo caso tampoco será necesario repetir esos cálculos.

Una vez desarrollados los resultados teóricos que exponen la visión estructurada del RG de un SAM, se va a desarrollar el algoritmo de cálculo y almacenamiento eficiente de una descripción descompuesta del RG. Esta descripción descompuesta contiene toda la información necesaria para generar el RG completo y emplea únicamente los módulos del SAM para los cálculos.

Algoritmo 6.33 Descripción del RG de un SAM.

input: $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ un SAM

for $i := 1$ **to** K **do**

Generar módulo $\mathcal{S}_i = \langle \mathcal{N}_i, H_i \rangle$ $H_i = \mathbf{m}_0|_{P_i}$

$\text{bfs}(\mathcal{N}_i, H_i) \rightarrow \text{RG}(\mathcal{S})|_{P_i}$ búsqueda en anchura estados locales

end for

$H := (H_1, \dots, H_K)$

$Q := \emptyset; S := \emptyset; \text{Rel}(S) = \emptyset$ estructuras sobre vértices cabecera

$b_0 := \mathbf{m}_0|_B$; Inserta (b_0, H) en Q y en S

while $Q \neq \emptyset$ **do**

Extrae $Q \rightarrow (b, H)$ primer vértice cabecera de Q

$L := \emptyset$ transiciones de interfaz sensibilizadas

for $j := 1$ **to** K

$L := L \cup \{I_j \in T_j \cap \text{TI} \mid H_j \xrightarrow{\sigma} h_j, \sigma \subseteq T_j \setminus \text{TI}, I_j \text{ sensibil. en } (b, h_j)\}$

end for

for each $I_j \in L$ **do**

for each h_j tal que $H_j \xrightarrow{\sigma} h_j, \sigma \subseteq T_j \setminus \text{TI}, I_j$ sensibil. en (b, h_j) **do**

$(b, h_j) \xrightarrow{I_j} (b', h'_j)$

$H' := (H_1, \dots, h'_j, \dots, H_K)$

if $h'_j \notin \text{RG}(\mathcal{S})|_{P_j}$ **then** $\text{bfs}(\mathcal{N}_j, h'_j) \rightarrow \text{RG}(\mathcal{S})|_{P_j}$

nuevo:=true

if $b' \in S|_B$ **then**

for each $(b', H'') \in S$ **do**

if (b', H') sucesor de (b', H'') por $T \setminus \text{TI}$ **then** nuevo:= false

else if (b', H'') sucesor de (b', H') por $T \setminus \text{TI}$ **then**

Sustituir (b', H'') por (b', H') en S

Actualizar $\text{RG}(S)$ y $\text{Rel}(S)$; nuevo:=false

else if $(b', H'), (b', H'')$ sucesor común por $T \setminus \text{TI}$ **then**

Inserta $((b', H'), (b', H''))$ en $\text{Rel}(S)$

end if

end for

end if

if nuevo = true **then** Inserta (b', H') en Q y en S

end for

Inserta $(b, H) \xrightarrow{I_j} (b', H')$ en $\text{RG}(S)$

end for

end while

output: S conjunto de vértices cabecera

$$\text{RG}(S) = \{\text{transiciones entre vértices cabecera}\}$$

$$\text{Rel}(S) = \{\text{vértices cabecera relacionados}\}$$

$$\text{RG}(\mathcal{S})|_{P_i} \text{ para } 1 \leq i \leq K$$

Se va a dar una explicación del funcionamiento del algoritmo. El objetivo del algoritmo consiste en recorrer ciertos nodos (el menor posible) del RG de un SAM con el propósito de calcular unas estructuras de datos que contengan toda la información necesaria para generar el RG completo. Estas estructuras son por un lado las proyecciones del RG del SAM sobre cada uno de los módulos (los grafos dirigidos etiquetados $\text{RG}(\mathcal{S})|_{P_i}$ con $1 \leq i \leq K$ del algoritmo) y por otro lado información sobre los vértices cabecera del SAM (estructuras S , $\text{RG}(S)$ y $\text{Rel}(S)$ en el algoritmo). La proyección de $\text{RG}(\mathcal{S})$ sobre el módulo \mathcal{S}_i consta de las proyecciones de $\text{R}(\mathcal{S})$ sobre el conjunto de lugares P_i junto con las transiciones entre ellos (que obviamente serán transiciones de $T_i \setminus \text{TI}$). Estas proyecciones se almacenan como grafos dirigidos etiquetados y contienen toda la información relativa a las transiciones internas entre estados alcanzables del SAM. Los vértices cabecera del SAM (conjunto S en el algoritmo) son los estados alcanzables del mismo a los que sólo se puede llegar por el disparo de transiciones de interfaz. Para almacenar toda la información relativa a las transiciones de interfaz entre estados alcanzables del SAM, se calcula el grafo dirigido etiquetado de transiciones de interfaz entre vértices cabecera ($\text{RG}(S)$ en el algoritmo). El significado de este grafo dirigido etiquetado es el siguiente. Cuando se dispara una transición de interfaz t se produce un cambio de estado tal que los estados implicados son sucesores de vértices cabecera diferentes. Si $(b, H) \xrightarrow{t} (b', H')$ es una arista de $\text{RG}(S)$, quiere decir que hay un sucesor del vértice cabecera (b, H) por medio de transiciones internas del SAM que sensibiliza a la transición t , y tal que su disparo produce un nuevo estado del SAM que es sucesor del vértice cabecera (b', H') por transiciones internas. En particular $\text{RG}(S)$ contiene, para cada vértice cabecera, la lista de transiciones de interfaz que se pueden sensibilizar a partir de sus sucesores por medio de transiciones internas del SAM. Por último, el conjunto $\text{Rel}(S)$ incluye los pares de vértices cabecera que pertenecen a la misma clase de equivalencia de R_0 (en el fondo, esta relación es una relación de equivalencia y se puede implementar como un conjunto cociente, para reducir espacio en memoria).

A partir de la estructura del SAM original se generan los módulos \mathcal{S}_i con $1 \leq i \leq K$. El marcado inicial se proyecta sobre los canales $(b_0$ en el

algoritmo) y sobre los lugares de cada módulo (H_i con $i \leq i \leq K$ en el algoritmo), obteniendo el primer vértice para explorar.

Se mantiene una estructura de tipo cola (Q en el algoritmo) para ir explorando distintos vértices de $\text{RG}(\mathcal{S})$. El primer ocupante de esta cola es el marcado inicial. El hecho de que un determinado marcado esté en la cola no quiere decir que al final sea un vértice cabecera, sino únicamente que será tratado como tal. Si durante la ejecución del algoritmo se llega a observar que es sucesor por transiciones internas de otro vértice, entonces será sustituido por ese vértice en las estructuras \mathcal{S} , $\text{RG}(\mathcal{S})$ y $\text{Rel}(\mathcal{S})$. El conjunto \mathcal{S} contiene a lo largo de todo el algoritmo un conjunto de vértices que se cree que son vértices cabecera. Al final del algoritmo contendrá todos los vértices cabecera del SAM.

Antes de insertar cualquier vértice (b, H) en la cola se asegura que estén calculados todos sus sucesores en la clase de equivalencia de R_0 , es decir, todos los estados alcanzables a partir de (b, H) por medio de transiciones internas del SAM. De acuerdo con el teorema 6.31, si $(b, H) = (b, H_1, \dots, H_K)$ es suficiente con explorar en cada módulo \mathcal{S}_i los sucesores de H_i por transiciones de $T_i \setminus \text{TI}$ para $1 \leq i \leq K$. Con estos sucesores se tiene una descripción completa de todos los sucesores de (b, H) en su clase de R_0 . Esta operación puede hacerse por medio de un recorrido en anchura (bfs en el algoritmo). Este recorrido en anchura permite calcular estados locales en cada módulo que es necesario almacenar en $\text{RG}(\mathcal{S})|_{P_i}$ para no repetir cálculos con posterioridad (estos estados locales son parte de la proyección del RG del SAM sobre cada módulo). Por lo tanto, el primer bucle del algoritmo genera estos vértices sucesores del marcado inicial por transiciones internas de cada módulo. Una vez hecha esta operación puede insertarse en la cola el marcado inicial y entrar en el bucle *while* del algoritmo que es el que realiza el resto de cálculos.

En cada iteración del bucle *while* se extrae el primer vértice (b, H) de la cola. Cualquier vértice que ha sido insertado en la cola, tiene ya calculados en cada módulo sus sucesores por transiciones internas del SAM y por lo tanto es posible calcular las transiciones de interfaz que se pueden sensibilizar a partir del estado (b, H) o de sus sucesores por transiciones internas. Por el teorema 6.32, basta explorar localmente en cada módulo \mathcal{S}_j las transiciones de interfaz del módulo \mathcal{S}_j que se pueden sensibilizar a partir de (b, H_j) o de sus sucesores por transiciones internas del módulo \mathcal{S}_j . Con esta exploración se construye el conjunto L de transiciones de interfaz que se pueden sensibilizar a partir de (b, H) o sucesores por transiciones internas.

El siguiente paso consiste en disparar cada transición de interfaz del conjunto L lo que permitirá generar nuevos vértices del SAM. Se utiliza la notación I_j para transiciones de $T_j \cap \text{TI}$. El disparo de la transición I_j modifica el marcado de los canales y de los lugares internos del módulo \mathcal{N}_j por lo que se puede restringir el estudio a (b, H_j) . En general, puede haber varios sucesores (b, h_j) de (b, H_j) por transiciones de $T_j \setminus \text{TI}$ con la transición I_j sensibilizada. Para cada (b, h_j) de estos se dispara la transición I_j produciendo un nuevo marcado (b', h'_j) . En este caso se sabe que en el SAM global, el disparo de I_j en (b, H) produce el estado (b', H') con la notación del algoritmo ($H' = (H_1, \dots, h'_j, \dots, H_k)$).

Ahora hay que actualizar las estructuras de datos. Lo primero que hay que hacer es ver si h'_j es nuevo en $\text{RG}(\mathcal{S})|_{P_j}$. Si es así, un recorrido en anchura calcula y almacena en $\text{RG}(\mathcal{S})|_{P_j}$ nuevos estados locales del módulo \mathcal{S}_j y con ellos ya se tienen todos los sucesores por transiciones internas del estado global (b', H') ya que en el resto de módulos no ha cambiado el marcado. Sólo falta ver si el estado (b', H') modifica de alguna manera la información sobre los vértices cabecera. Si el marcado de los canales b' aparece por primera vez en $S|_B$, se tiene un nuevo vértice que hay que insertar en la cola, en S y una nueva arista en $\text{RG}(S)$. En caso contrario, es posible que el marcado (b', H') modifique la información sobre los vértices cabecera previamente calculados. Para ello hay que revisar todos los vértices cabecera con el mismo marcado en los canales, es decir, estados del conjunto S del tipo (b', H'') . Si H' es sucesor por transiciones internas de algún H'' (test que puede hacerse localmente en cada módulo por tenerse almacenados todos los marcados locales) entonces el estado (b', H') no es vértice cabecera y no es necesario insertarlo en la cola. En otro caso, se inserta (b', H') en la cola puesto que de momento no es sucesor por transiciones internas de otro estado y hay que tratarlo como vértice cabecera. Además, si algún $(b', H'') \in S$ es sucesor de (b', H') entonces el vértice (b', H'') que hasta ahora era tratado como vértice cabecera, se sabe ya que no lo es y por lo tanto es necesario actualizar S , $\text{RG}(S)$ y $\text{Rel}(S)$ de acuerdo con esta nueva información. El último caso que queda es si (b', H') y (b', H'') tienen un sucesor por transiciones internas en común (test que puede hacerse localmente en cada módulo). En ese caso se marcan como relacionados en $\text{Rel}(S)$, porque pertenecen a la misma clase de R_0 . Esta estructura de datos con información sobre los vértices cabecera relacionados se necesita en la sección 6.6.

Las inserciones en S , $\text{RG}(S)$ y $\text{Rel}(S)$ se toman a lo largo de todo el algoritmo en el sentido conjuntista, es decir, si en algún caso se intenta in-

sertar un elemento por segunda vez, el resultado de esa inserción no modifica el conjunto. Esta interpretación se hace para simplificar la exposición del algoritmo.

La salida del algoritmo consta de las estructuras de datos necesarias para reconstruir el $\text{RG}(\mathcal{S})$ y de $\text{Rel}(S)$, necesaria para un algoritmo posterior. Por un lado calcula los marcados locales de cada módulo junto con las transiciones internas entre ellos ($\text{RG}(\mathcal{S})|_{P_i}$ para $1 \leq i \leq K$) y por otro lado información relativa a las transiciones de interfaz y los vértices cabecera. En concreto, se calcula el conjunto S de vértices cabecera del SAM junto con las transiciones de interfaz que conectan sucesores por transiciones internas de un vértice cabecera con los de otro (grafo dirigido etiquetado $\text{RG}(S)$ en el algoritmo). Por último, se calcula el conjunto $\text{Rel}(S)$ de vértices cabecera relacionados entre sí, es decir, vértices cabecera que pertenecen a la misma clase de equivalencia de R_0 .

Estas estructuras constituyen una descripción de $\text{RG}(\mathcal{S})$ descompuesta por módulos ya que a partir de los vértices cabecera y por medio de los grafos dirigidos etiquetados locales de los módulos ($\text{RG}(\mathcal{S})|_{P_i}$ para $1 \leq i \leq K$) es posible calcular todas las transiciones internas entre estados alcanzables del SAM (por el teorema 6.31). Para calcular las transiciones de interfaz entre estados alcanzables del SAM se utilizan el conjunto S y el grafo dirigido etiquetado $\text{RG}(s)$ (por el teorema 6.32).

No se ha realizado un estudio detallado de la complejidad de este algoritmo por su elevada dificultad. En el peor caso no se rebaja la complejidad ni en espacio ni en tiempo respecto al algoritmo clásico ya que puede ser necesario recorrer todos los estados alcanzables del SAM (esto ocurre por ejemplo si todas las transiciones del SAM se consideran de interfaz). Si el SAM tiene varios módulos y hay concurrencia entre el funcionamiento de los mismos entonces la rebaja, tanto en espacio como en tiempo respecto al algoritmo clásico, es exponencial debido a que no es necesario calcular todas las posibles combinaciones de estados locales de los módulos.

Esta descripción descompuesta del RG de un SAM puede utilizarse para varios propósitos, como por ejemplo la generación eficiente de todo el RG del SAM o, en el marco de esta memoria, la generación directa de los RG's asociados a los sistemas agregados del SAM sin tener que realizar ningún tipo de reducción a nivel de red.

Como ejemplo de aplicación, se va a desarrollar a continuación un algoritmo que permite la generación del RG del SAM original a partir de las estructuras de datos generadas en el algoritmo anterior.

Algoritmo 6.34 Generación del RG de un SAM.

input: S conjunto de vértices cabecera de un SAM (algoritmo 6.33)
 $\text{RG}(\mathcal{S})|_{P_i}$ para $1 \leq i \leq K$ (algoritmo 6.33)
 $G := \emptyset$
for each $H = (H_0, \dots, H_K) \in S$ **do**
 $g(H) := H_0$ marcado en los canales.
for $i := 1$ **to** K
 $G_i := \text{subgrafo de } \text{RG}(\mathcal{S})|_{P_i} \text{ de sucesores de } H_i \text{ por } T_i \setminus \text{TI}$
 $g(H) := g(H) \times G_i$
end for
 $G := G \cup g(H)$
end for
for each $H \in S$ **do**
for $i := 1$ **to** K
for each sucesor h_i de H_i por $T_i \setminus \text{TI}$
 $L := \{t \in T_i \cap \text{TI} \mid t \text{ sensibilizada en } (H_0, h_i)\}$
for each $t \in L$ **do**
 $(H_0, h_i) \xrightarrow{t} (H'_0, h'_i)$
for each $s \in g(H)$ tal que $(s_0, s_i) = (H_0, h_i)$ **do**
Insertar $s \xrightarrow{t} s'$ en G
end for
end for
end for
end for
end for
output: $G \cong \text{RG}(\mathcal{N}, \mathbf{m}_0)$

Se va a dar una breve explicación de este algoritmo. A partir de la descripción descompuesta del RG de un SAM (algoritmo 6.33), todos los estados alcanzables se pueden calcular a partir de los vértices cabecera y secuencias de disparo formadas únicamente por transiciones internas a los módulos del SAM. Esta es la tarea que se hace en el primer conjunto de bucles encajados del algoritmo. Se van recorriendo todos los vértices cabecera (H en el algoritmo) y para cada uno de estos se visitan todos los módulos del SAM y se extraen los subgrafos dirigidos etiquetados de sucesores locales al vértice cabecera en cada módulo (G_i en la notación del algoritmo). El grafo G_i con $1 \leq i \leq K$ contiene los sucesores por transiciones internas del módulo \mathcal{S}_i de cualquier estado del SAM que tenga a H_i como marcado local.

Estos grafos dirigidos etiquetados ya se tienen calculados a partir del algoritmo 6.33, por lo que no es necesario repetir cálculos. Por el teorema 6.31, el grafo producto de todos los G_i está formado por estados alcanzables del SAM. En esta fase sólo hay que combinar estados locales entre sí de forma automática. Como un determinado estado alcanzable del SAM puede ser sucesor de varios vértices cabecera, es necesario realizar una union conjuntista de los grafos producto que se calculan para cada vértice cabecera. De esta forma no se repiten estados alcanzables. Por lo tanto, con el primer conjunto de bucles encajados se consigue calcular todos los estados alcanzables del SAM y todas las transiciones internas que hay entre ellos.

El segundo conjunto de bucles encajados se encarga de añadir las transiciones de interfaz entre los estados alcanzables del SAM. Para ello se vuelven a recorrer todos los vértices cabecera y para cada uno de ellos se calcula localmente en cada módulo sus sucesores (h_i en el algoritmo). A cada sucesor se le añade el marcado de los canales (H_0 en el algoritmo) y se calcula el conjunto de transiciones de interfaz correspondientes al módulo \mathcal{S}_i que están sensibilizadas en (H_0, h_i) (L en el algoritmo). Por el teorema 6.32, todas estas transiciones de interfaz también están sensibilizadas en todos los estados del SAM que resultan de la combinación del estado local (H_0, h_i) con los del resto de módulos, de ahí el recorrido del grafo $g(H)$ de los sucesores del vértice cabecera. Este recorrido se puede hacer visitando localmente los módulos \mathcal{S}_j con $j \neq i$ sin necesidad de tener almacenado $g(H)$. Es posible gestionar el cálculo de las transiciones de interfaz y las internas a la vez, pero se ha elegido esta forma por simplicidad de exposición, aunque quede un algoritmo de mayor complejidad. La forma de tratar a la vez los dos tipos de transiciones consiste en realizar los cálculos de transiciones internas y de interfaz localmente en cada módulo y después hacer una operación similar a la del producto de grafos dirigidos etiquetados que trate de forma diferente los dos tipos de transiciones.

En general, la combinación de los algoritmos 6.33 y 6.34 consiguen calcular el RG de un SAM con menor complejidad en tiempo que en el algoritmo clásico ya que no es necesario calcular uno por uno todos los estados alcanzables del SAM ni todas las transiciones entre ellos. Ciertos conjuntos de transiciones que se sabe que son concurrentes a partir de la visión estructurada a nivel de red, lo que permite la generación de estados localmente en cada módulo y combinarlos de forma automática sin hacer un test previo sobre el conjunto de transiciones sensibilizadas. Este hecho acelera los cálculos.

El siguiente paso consiste en emplear esta descripción descompuesta del RG de un SAM para la generación de RG's asociados a los sistemas agregados del SAM. Estos sistemas agregados se emplean posteriormente para aproximar el throughput de las transiciones del modelo original. Esta tarea se realizará en las dos secciones siguientes. En la sección 6.5 se construirán RG's asociados a los sistemas agregados del SAM sin marcados espurios (eliminando la primera fuente de problemas identificada en la sección 6.4) y en la sección 6.6 se reducirán algunas secuencias de disparo espurias que todavía pueden quedar en los sistemas agregados de la sección 6.5 (eliminando la segunda fuente de problemas identificada en la sección 6.4).

6.5 Eliminación de marcados espurios

En esta sección se va a utilizar la descripción descompuesta del RG de un SAM calculada en la sección anterior con el algoritmo 6.33 para generar directamente RG's asociados a los sistemas agregados del SAM (\mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS}). El objetivo en esta sección consiste en conseguir que los RG's asociados a los sistemas agregados no contengan ningún estado espurio, es decir, que sean proyecciones exactas en términos de marcados alcanzables del RG del SAM original sobre los nodos preservados en cada sistema agregado ($P_i \cup B$ en \mathcal{LS}_i con $1 \leq i \leq K$ y B en el \mathcal{BS}). Por lo tanto, la mejora de esta segunda técnica de descomposición con respecto a la primera presentada en la sección 6.3 consiste en la eliminación de todos los estados espurios que pueden aparecer si se opera a nivel de red. Con respecto a la tercera técnica de descomposición que se presentará en la sección 6.6, todavía persiste el segundo problema identificado en la sección 6.4 respecto a la agregación inadecuada de estados no conectados por transiciones internas en el SAM original.

Para formalizar los cálculos que se pretenden realizar es preciso definir con precisión el objeto de cálculo de esta sección, es decir, los RG's que se quieren asociar a los sistemas agregados del SAM. Esta es la tarea que se realiza en las siguientes definiciones.

Definición 6.35 *Sea \mathcal{S} un SAM con K módulos y $P = B \cup (\bigcup_{i=1}^K P_i)$. Sea $R_i^!$ con $1 \leq i \leq K$ la siguiente relación definida sobre $R(\mathcal{S})$: Dados s_1, s_2 vértices de $RG(\mathcal{S})$, $(s_1, s_2) \in R_i^!$ si y sólo si $s_1|_{P_i \cup B} = s_2|_{P_i \cup B}$. Sea R_0 la siguiente relación definida sobre $R(\mathcal{S})$: Dados s_1, s_2 vértices de $RG(\mathcal{S})$, $(s_1, s_2) \in R_0$ si y sólo si $s_1|_B = s_2|_B$.*

Se denotan las relaciones por R'_i con $0 \leq i \leq K$ para distinguirlas de las relaciones R_i de la definición 6.23.

Proposición 6.36 *Sea \mathcal{S} un SAM con K módulos y R'_i con $0 \leq i \leq K$ las relaciones de la definición 6.35 sobre $R(\mathcal{S})$. Entonces R'_i es una relación de equivalencia para $0 \leq i \leq K$.*

Demostración:

Se hará la demostración tomando $P_0 = \emptyset$.

Reflexiva: $(s, s) \in R'_i$ para todo $s \in R(\mathcal{S})$ ya que $s|_{P_i \cup B} = s|_{P_i \cup B}$.

Simétrica: Si $(s_1, s_2) \in R'_i$ entonces $s_1|_{P_i \cup B} = s_2|_{P_i \cup B}$, es decir, $s_2|_{P_i \cup B} = s_1|_{P_i \cup B}$ y por lo tanto $(s_2, s_1) \in R'_i$.

Transitiva: Si $(s_1, s_2), (s_2, s_3) \in R'_i$, entonces $s_1|_{P_i \cup B} = s_2|_{P_i \cup B}$ y $s_2|_{P_i \cup B} = s_3|_{P_i \cup B}$, es decir, $s_1|_{P_i \cup B} = s_3|_{P_i \cup B}$ y por lo tanto $(s_1, s_3) \in R'_i$.

Por lo tanto, R_i es una relación de equivalencia para $0 \leq i \leq K$. \diamond

Definición 6.37 *Sea \mathcal{S} un SAM con K módulos, $T = \bigcup_{i=1}^K T_i$ su conjunto de transiciones y $G = (V, L, A)$ su RG interpretado como grafo dirigido etiquetado. Sean R'_i con $0 \leq i \leq K$ las relaciones de equivalencia de la definición 6.35 sobre $R(\mathcal{S})$. Para $0 \leq i \leq K$ se define el grafo dirigido etiquetado agregado $RG'_i = (V'_i, L'_i, A'_i)$ con:*

- i) $V'_i = V/R'_i$ (conjunto cociente).
- ii) $L'_i = T_i \cup TI$ (se toma $T_0 = \emptyset$).
- iii) $A'_i = \{(S_1, S_2, t) \in V'_i \times V'_i \times L'_i \mid \exists (s_1, s_2, t) \in A \text{ con } s_1 \in S_1, s_2 \in S_2\}$.

A partir de esta definición, el grafo dirigido etiquetado RG'_0 será el RG que se quiere asociar a \mathcal{BS} y el grafo dirigido etiquetado RG'_i con $1 \leq i \leq K$ será el RG que se quiere asociar a \mathcal{LS}_i . El grafo dirigido etiquetado RG'_0 es el resultado de agregar en un único estado todos los estados del RG del SAM que tienen el mismo marcado en los canales y el grafo dirigido etiquetado RG'_i con $1 \leq i \leq K$ es el resultado de agregar en un único estado todos los estados del RG del SAM que tienen el mismo marcado en $P_i \cup B$ (canales y lugares internos del módulo \mathcal{S}_i).

Por definición, los grafos dirigidos etiquetados RG'_i con $0 \leq i \leq K$ constituyen proyecciones exactas en términos de marcados alcanzables del RG del SAM sobre el conjunto de nodos preservados en cada sistema agregado.

En este caso no contienen ningún marcado espurio, es decir, todos los estados alcanzables de \mathcal{LS}_i y \mathcal{BS} son proyecciones de algún estado alcanzable del SAM original sobre los nodos preservados en cada sistema agregado.

Para ayudar a comprender el algoritmo de cálculo de los grafos dirigidos etiquetados RG'_i con $0 \leq i \leq K$, se demostrará el siguiente resultado relativo a las transiciones entre estados de los RG'_i con $0 \leq i \leq K$.

Proposición 6.38 *Sea \mathcal{S} un SAM con K módulos y R'_i con $0 \leq i \leq K$ las relaciones de la definición 6.35 sobre $\text{R}(\mathcal{S})$. Sean s_1 y s_2 dos estados de $\text{RG}(\mathcal{S})$ tales que $(s_1, s_2) \in R'_i$ y $t \in T$ una transición sensibilizada en los dos estados. Si $s_1 \xrightarrow{t} s'_1$ y $s_2 \xrightarrow{t} s'_2$ entonces $(s'_1, s'_2) \in R'_i$.*

Demostración:

Sea $s_1 = (s_1^0, \dots, s_1^K)$ donde $s_1^i = s_1|_{P_i}$ para $0 \leq i \leq K$ (se toma $P_0 = B$). Análogamente $s_2 = (s_2^0, \dots, s_2^K)$. Por definición de R'_i , se cumple que $s_1^0 = s_2^0$ y $s_1^i = s_2^i$.

Caso 1) $t \in T_i$ con $i > 0$. Entonces el disparo de t modifica el marcado de $P_i \cup B$, es decir, $s'_1 = (s_1^0, s_1^1, \dots, s_1^{i-1}, s_1^i, s_1^{i+1}, \dots, s_1^K)$. Análogamente $s'_2 = (s_2^0, s_2^1, \dots, s_2^{i-1}, s_2^i, s_2^{i+1}, \dots, s_2^K)$. Como $s_1^0 = s_2^0$, $s_1^0 \xrightarrow{t} s_1^0$ y $s_2^0 \xrightarrow{t} s_2^0$ entonces $s_1^0 = s_2^0$. Análogamente, como $s_1^i = s_2^i$, $s_1^i \xrightarrow{t} s_1^i$ y $s_2^i \xrightarrow{t} s_2^i$ entonces $s_1^i = s_2^i$. Por lo tanto $(s'_1, s'_2) \in R'_i$.

Caso 2) $t \in T_j$ con $j \neq i$. Entonces el disparo de t modifica el marcado de $P_j \cup B$, es decir, $s'_1 = (s_1^0, s_1^1, \dots, s_1^{j-1}, s_1^j, s_1^{j+1}, \dots, s_1^K)$. Análogamente $s'_2 = (s_2^0, s_2^1, \dots, s_2^{j-1}, s_2^j, s_2^{j+1}, \dots, s_2^K)$. Como $s_1^0 = s_2^0$, $s_1^0 \xrightarrow{t} s_1^0$ y $s_2^0 \xrightarrow{t} s_2^0$ entonces $s_1^0 = s_2^0$. Por otro lado, si $i > 0$ entonces $s_1^i = s_1^i = s_2^i = s_2^i$, luego $(s'_1, s'_2) \in R'_i$. \diamond

Este resultado demuestra que las agregaciones de estados mediante las relaciones de equivalencia R'_i de la definición 6.35 preservan las transiciones entre clases, es decir, si s es una clase de equivalencia de R'_i con $1 \leq i \leq K$ (por lo tanto un estado del sistema agregado correspondiente) y $t \in T$ es una transición cualquiera del SAM que está sensibilizada en dos representantes s_1 y s_2 de s , al disparar t desde cualquiera de ellos se alcanzan estados que pertenecen a la misma clase de equivalencia de R'_i . Esto quiere decir que al construir los grafos dirigidos etiquetados agregados RG'_i con $1 \leq i \leq K$, el disparo de una transición sensibilizada en un estado agregado produce un único estado agregado.

El siguiente paso consiste en desarrollar el algoritmo de cálculo de los grafos dirigidos etiquetados agregados RG'_i con $0 \leq i \leq K$.

Algoritmo 6.39 Proyección RG'_l del RG de un SAM.

input: S conjunto de vértices cabecera de un SAM (algoritmo 6.33)
 $RG(S) = \{\text{transiciones entre vértices cabecera}\}$ (algoritmo 6.33)
 $RG(S)|_{P_i}$ para $1 \leq i \leq K$ (algoritmo 6.33)
 $0 \leq l \leq K$ RG'_l a calcular ($P_0 = T_0 = \emptyset$)
 $\bar{S} := \{(s_0, s_l) \mid (s_0, \dots, s_K) \in S\}$ si $l = 0$, s_l se elimina
for each $\bar{s} \in \bar{S}$
 $L(\bar{s}) := \{t \in TI \setminus T_l \mid \exists s \in S \text{ t.q. } \bar{s} = (s_0, s_l) \text{ y } t \text{ sensibilizada en } s\}$
end for
 $p := \mathbf{m}_0|_{B \cup P_l}$ marcado inicial de RG'_l .
 $Q := \emptyset$ cola de vértices por explorar
 Inserta p en Q y RG'_l
while $Q \neq \emptyset$ **do**
 Extrae $Q \rightarrow p := (b, m_l)$ primer par de la cola
 $A := \{(b, s_l) \in \bar{S} \mid s_l \xrightarrow{\sigma} m_l, \sigma \subseteq T_l \setminus TI\}$
 $L := \emptyset$
for each $\bar{s} \in A$ **do** $L := L \cup L(\bar{s})$
 $L := L \cup \{t \in T_l \setminus TI \mid t \text{ sensibilizada en } m_l\}$
 $L := L \cup \{t \in T_l \cap TI \mid t \text{ sensibilizada en } p\}$
for each $t \in L$ **do**
 $p \xrightarrow{t} p'$
if p' es nuevo en RG'_l **then** Inserta p' en Q y en RG'_l
 Inserta $p \xrightarrow{t} p'$ en RG'_l
end for
end while
output: RG'_l

Se va a dar ahora una explicación de este algoritmo para el caso $l > 0$, es decir, el cálculo del grafo dirigido etiquetado RG'_l (el caso del grafo dirigido etiquetado RG'_0 es parecido). Lo primero que hace el algoritmo es una proyección de los vértices cabecera sobre el módulo preservado (si $l > 0$). El objeto de esta proyección es adaptar las estructuras de datos de salida del algoritmo 6.33 al cálculo que se pretende realizar. Como se quieren agregar estados del SAM que coinciden en el marcado de $P_l \cup B$ (tomando $P_0 = \emptyset$), el conjunto de transiciones de interfaz sensibilizadas a partir de un vértice cabecera proyectado puede verse alterado. Puede ocurrir que diferentes vértices cabecera del SAM se proyecten en el mismo estado. Por lo tanto, el conjunto $L(\bar{s})$ de transiciones de $TI \setminus T_l$ disparables a partir de

un vértice cabecera proyectado será la unión de los conjuntos asociados a los vértices cabecera que tienen la misma proyección \bar{s} (por el teorema 6.32). Esta tarea se realiza en el primer bucle que recorre el conjunto S de vértices de cabecera. Después de la proyección de todos estos vértices se obtiene un conjunto \bar{S} con los vértices cabecera proyectados y para cada uno de ellos una lista de transiciones de $TI \setminus T_l$ que pueden dispararse en sus sucesores por medio de transiciones internas.

Posteriormente, el algoritmo realiza básicamente un recorrido en anchura del grafo RG'_l . Se mantiene una estructura de datos de tipo cola (Q en el algoritmo) con estados por explorar de RG'_l . El principal problema consiste en calcular el conjunto de transiciones que se pueden disparar a partir de un estado $p = (b, m_l)$ de RG'_l . Como p tiene información sobre el marcado de $P_l \cup B$ no es problema calcular las transiciones del conjunto T_l sensibilizadas en p . En particular, las transiciones internas de $T_l \setminus TI$ sensibilizadas en m_l ya se tienen calculadas en el algoritmo 6.33 (en $RG(\mathcal{S})|_{P_l}$) por lo que no es necesario repetir el cálculo. Pero falta conocer qué transiciones del conjunto $TI \setminus T_l$ se pueden disparar a partir de p . Para ello se utiliza la información de los vértices cabecera (los conjuntos $L(\bar{s})$ en el algoritmo). La gestión de los vértices cabecera hay que hacerla con cuidado. Es evidente que p puede ser sucesor por medio de transiciones de $T_l \setminus TI$ de varios vértices cabecera proyectados del conjunto \bar{S} . Por ello se calcula el conjunto A con todos los elementos de \bar{S} que tienen a p como sucesor por transiciones de $T_l \setminus TI$. Entonces, cualquier transición de los conjuntos $L(\bar{s})$ con $\bar{s} \in A$ puede dispararse a partir de p (teorema 6.32).

Una vez calculadas las transiciones sensibilizadas en el estado p se procede al disparo de las mismas. El disparo de una transición $t \in L$ produce un nuevo estado p' de RG'_l . Si este estado es nuevo, se inserta en la cola Q y en RG'_l para su posterior estudio. Si no es nuevo, no es necesario su inserción porque ya ha sido estudiado. Simplemente se añade la nueva transición $p \xrightarrow{t} p'$ en RG'_l .

Por construcción, todos los estados alcanzables de RG'_i con $0 \leq i \leq K$ son proyecciones de estados del SAM original en los nodos preservados en cada sistema agregado, por lo que las inclusiones del apartado 2 del teorema 6.11 se vuelven ahora igualdades.

La complejidad tanto en espacio como en tiempo del algoritmo necesario para generar los RG'_i con $0 \leq i \leq K$ es menor que la de la primera técnica general de descomposición (sección 6.3) ya que no se genera en ningún momento ningún estado espurio.

6.6 Reducción de secuencias de disparo espurias

En esta sección se va a utilizar la descripción descompuesta del RG de un SAM para generar directamente otros RG's asociados a los sistemas agregados del SAM (\mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS}) diferentes a los de la sección anterior. El objetivo en esta sección consiste en eliminar las agregaciones inadecuadas de estados que se pueden producir con las técnicas anteriores al tener en cuenta únicamente el marcado de ciertos subconjuntos de lugares de la red original (problema descrito en detalle en la sección 6.4).

En este caso se calcularán sistemas agregados cuyos RG's asociados sean las agregaciones del RG del SAM original por medio de las relaciones R_i con $0 \leq i \leq K$ de la definición 6.23. De esta forma se consiguen sistemas agregados sin estados espurios (como con la técnica de la sección anterior) y se eliminan las agregaciones de estados no conectados por caminos formados por transiciones internas. De esta forma se resuelve el segundo problema sobre agregación inadecuada de estados mencionado en la sección 6.4.

Por lo tanto, la mejora de esta tercera técnica de descomposición con respecto a la primera (sección 6.3) es la eliminación de todos los estados espurios que podían aparecer y con respecto a la segunda (sección 6.5) es la eliminación de las agregaciones inadecuadas de estados. En general, los sistemas agregados desarrollados con esta tercera técnica tendrán más estados que los que se generaban con la técnica de la sección anterior, pero al reducirse todas las agregaciones inadecuadas, estos sistemas agregados tienen un comportamiento más parecido al sistema original lo que permite mejorar las aproximaciones al aplicarles los algoritmos numéricos.

Para formalizar los cálculos que se pretenden realizar es preciso definir con precisión el objeto de cálculo de esta sección, es decir, los RG's que se quieren asociar a los sistemas agregados del SAM.

Definición 6.40 Sea \mathcal{S} un SAM con K módulos, $T = \bigcup_{i=1}^K T_i$ su conjunto de transiciones y $G = (V, L, A)$ su RG interpretado como grafo dirigido etiquetado. Sean R_i con $0 \leq i \leq K$ las relaciones de equivalencia de la definición 6.23 sobre $R(\mathcal{S})$. Para $0 \leq i \leq K$ se define el grafo dirigido etiquetado agregado $\text{RG}_i = (V_i, L_i, A_i)$ con:

- i) $V_i = V/R_i$ (conjunto cociente).
- ii) $L_i = T_i \cup \text{TI}$ (se toma $T_0 = \emptyset$).
- iii) $A_i = \{(S_1, S_2, t) \in V_i \times V_i \times L_i \mid \exists (s_1, s_2, t) \in A \text{ con } s_1 \in S_1, s_2 \in S_2\}$.

A partir de la esta definición, el grafo dirigido etiquetado RG_0 será el RG que se quiere asociar a \mathcal{BS} y el grafo dirigido etiquetado RG_i con $1 \leq i \leq K$ será el RG que se quiere asociar a \mathcal{LS}_i . El grafo dirigido etiquetado RG_0 es el resultado de agregar en un único estado todos los estados del RG del SAM que están conectados por caminos no dirigidos de transiciones internas del SAM y el grafo dirigido etiquetado RG_i con $1 \leq i \leq K$ es el resultado de agregar en un único estado todos los estados del RG del SAM que están conectados por caminos no dirigidos de transiciones del conjunto $T \setminus (T_i \cup TI)$, es decir, transiciones internas a los módulos que no se preservan en \mathcal{LS}_i .

Por definición, los RG_i con $0 \leq i \leq K$ constituyen proyecciones exactas en términos de marcados alcanzables del RG del SAM sobre el conjunto de lugares que se preservan en cada sistema agregado. Por lo tanto, al asociar estos grafos dirigidos etiquetados a los sistemas agregados se tiene que \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} no contienen ningún marcado espurio, es decir, todos los estados de \mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS} son proyecciones de algún estado alcanzable del SAM original sobre los lugares preservados en cada sistema agregado.

Se va a demostrar una proposición que permitirá ilustrar algunas características sobre los nuevos sistemas agregados que se van a calcular.

Proposición 6.41 *Sea \mathcal{S} un SAM con K módulos y $G = (V, L, A)$ su RG interpretado como grafo dirigido etiquetado. Sean R_i, R'_i con $0 \leq i \leq K$ las relaciones de equivalencia de las definiciones 6.23 y 6.35 sobre $R(\mathcal{S})$. Para $0 \leq i \leq K$ se cumple que $R_i \subseteq R'_i$.*

Demostración:

Evidente por la proposición 6.26 ◇

Esta proposición indica que dos estados distintos del SAM original que pertenezcan a la misma clase de equivalencia de R_i con $0 \leq i \leq K$ también pertenecen a la misma clase de equivalencia de R'_i , es decir, su marcado coincide en el conjunto $P_i \cup B$ (con $P_0 = \emptyset$). Por lo tanto se pueden identificar todos los estados de una clase de equivalencia de R_i por el marcado de cualquiera de sus representantes en $P_i \cup B$. El otro contenido no es cierto en general, es decir, puede haber estados diferentes del SAM original con el mismo marcado en $P_i \cup B$ que no pertenezcan a la misma clase de equivalencia de R_i . Este hecho se verá con un ejemplo.

En la figura 6.5 se tiene un SAM compuesto por dos módulos si se toman los lugares B_1, B_2 y B_3 como canales. Las transiciones I_{ij} con $i = 1, 2$ y

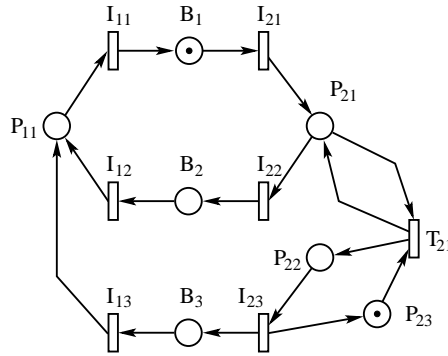


Figura 6.5: Un SAM.

$j = 1, 2, 3$ son las transiciones de interfaz. El módulo \mathcal{S}_1 está generado por el lugar P_{11} y el módulo \mathcal{S}_2 por los lugares P_{21} , P_{22} y P_{23} . En la figura 6.6 se tiene su RG. En él se puede observar que los estados B_2P_{22} y B_2P_{23} son vértices cabecera ya que sólo se pueden alcanzar tras el disparo de transiciones de interfaz (en este caso I_{22} en ambos). Atendiendo a su marcado, los dos estados coinciden en los canales y en los lugares del módulo \mathcal{S}_1 , por lo que pertenecen a la misma clase de equivalencia de R'_1 de la definición 6.35. Sin embargo, estos dos estados no pertenecen a la misma clase de equivalencia de R_1 porque no están conectados por un camino no dirigido de transiciones internas del módulo \mathcal{S}_2 . Si se utiliza la técnica de agregación de la sección anterior, los estados B_2P_{22} y B_2P_{23} resultan agregados en \mathcal{LS}_1 y \mathcal{BS} , lo que supone una agregación inadecuada de estados según se ha comentado en la sección 6.4.

Al evitar este tipo de agregaciones, en los RG's de los sistemas agregados pueden aparecer estados distintos con el mismo marcado en los lugares del sistema agregado. Esta situación no ocurre en redes de Petri, pero es fundamental para obtener mejores aproximaciones. No constituye ningún problema ya que este hecho simplemente indica que esos estados provienen de evoluciones diferentes del SAM original y que estas diferencias se quieren mantener en los sistemas agregados. Esta diferenciación entre estados agregados aparecerá cuando provengan de estados del SAM original que provoquen diferentes secuencias de disparo de transiciones de interfaz. Como estas transiciones si se preservan al generar los sistemas agregados, es lógico diferenciar también los estados agregados que provoquen distintas

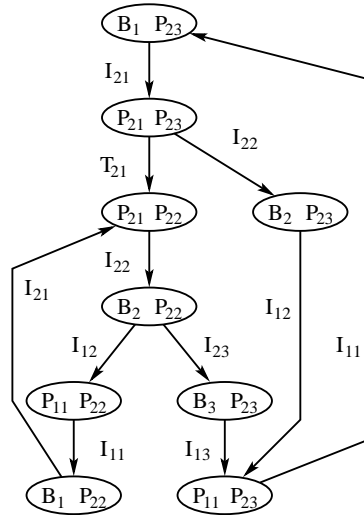


Figura 6.6: RG del SAM de la figura 6.5.

secuencias de disparo de transiciones de interfaz.

Por lo que respecta a las transiciones sensibilizadas en varios representantes de una clase de equivalencia de R_i , ahora no existe un resultado equivalente a la proposición 6.38 de la sección anterior, es decir, es posible que existan estados diferentes s_1 y s_2 de un SAM pertenecientes a la misma clase de equivalencia de R_i , con una misma transición sensibilizada t tales que si $s_1 \xrightarrow{t} s'_1$ y $s_2 \xrightarrow{t} s'_2$ entonces s'_1 y s'_2 no pertenezcan a la misma clase de equivalencia de R_i . Un ejemplo de esta situación se puede observar en el RG de la figura 6.6. Los estados $P_{21}P_{23}$ y $P_{21}P_{22}$ pertenecen a la misma clase de equivalencia de R_1 porque $P_{21}P_{23} \xrightarrow{T_{21}} P_{21}P_{22}$. Los dos estados tienen la transición de interfaz I_{22} sensibilizada y su disparo produce B_2P_{23} y B_2P_{22} respectivamente. Pero estos últimos estados no pertenecen a la misma clase de equivalencia de R_1 ya que no están conectados por un camino no dirigido de transiciones internas del módulo \mathcal{S}_2 . Este hecho se traduce en que al generar los RG_i , es posible que a partir de un estado agregado haya dos transiciones a otros dos estados diferentes disparando la misma transición, cosa que tampoco ocurre en redes de Petri.

El siguiente paso consiste en desarrollar el algoritmo que calcula los RG_i con $0 \leq i \leq K$.

Algoritmo 6.42 Proyección RG_l del RG de un SAM.

input: S conjunto de vértices cabecera de un SAM (algoritmo 6.33)
 $RG(S) = \{\text{transiciones entre vértices cabecera}\}$ (algoritmo 6.33)
 $RG(S)|_{P_i}$ para $1 \leq i \leq K$ (algoritmo 6.33)
 $Rel(S)$ (algoritmo 6.33)
 $0 \leq l \leq K$ RG_l a calcular

for each $s \in S$ **do**
 $A(s) := \{(s, s', t) \in RG(S) \mid t \in TI \setminus T_l\}$ ($T_0 = \emptyset$)
end for

$p := \mathbf{m}_0|_{P_l}$ si $l = 0$, p desaparece
 $H := (H_0, \dots, H_K)$ vértice cabecera cualquiera de \mathbf{m}_0
 $Q := \emptyset$ cola de vértices por explorar
 Inserta (H, p) en Q y RG_l

while $Q \neq \emptyset$ **do**
 Extrae $Q \rightarrow (H, p)$ primer par de la cola
 $L := \emptyset$
for each $t \in T_l \setminus TI$ sensibilizada en p **do** $L := L \cup (H, t)$
for each $t \in TI \cap T_l$ sensibilizada en (H_0, p) **do**
 $(H_0, p) \xrightarrow{t} (H'_0, p')$
 $H^* = (H'_0, H_1, \dots, H_{l-1}, p', H_{l+1}, \dots, H_K)$
 $H' =$ vértice cabecera cualquiera de H^*
 $L := L \cup (H', t)$
end for
 $C := \{s \in S \mid (H, s) \in Rel(S) \text{ y } p \text{ sucesor de } s|_{P_l} \text{ en } RG(S)|_{P_l}\}$
for each $s \in C$ **do**
 $L := L \cup \{(s', t) \mid \exists (s, s', t) \in A(s)\}$ ¹
end for
for each $(H', t) \in L$ **do**
 $(H, p) \xrightarrow{t} (H', p')$
if existe $(H'', p') \in RG_l$ con $(H', H'') \in Rel(S)$ **then** $H' := H''$
if (H', p') es nuevo en RG_l **then** Inserta (H', p') en Q y en RG_l
 Inserta $(H, p) \xrightarrow{t} (H', p')$ en RG_l
end for
end while
 proyectar RG_l sobre $B \cup P_l$ sin agregar estados con el mismo marcado

output: RG_l

¹Unión especial. Ver explicación del algoritmo.

Se va a dar ahora una explicación de este algoritmo para el caso $l > 0$, es decir, el cálculo de RG_l (el caso del grafo dirigido etiquetado RG_0 es parecido). La primera tarea del algoritmo consiste en generar, para cada vértice cabecera s del SAM, una lista $A(s)$ de aristas (s, s', t) de $RG(S)$ donde s' es otro vértice cabecera y t es una transición de interfaz de un módulo reducido en el RG_l . Esto quiere decir que hay un sucesor por medio de transiciones internas del SAM del vértice cabecera s que tiene sensibilizada la transición t y que al dispararla produce un estado sucesor por medio de transiciones internas del SAM del vértice cabecera s' . Esta información puede extraerse directamente del grafo $RG(S)$ y se utilizará posteriormente. El objeto de esta proyección es adaptar las estructuras de datos de salida del algoritmo 6.33 al cálculo que se pretende realizar.

Posteriormente, el algoritmo realiza básicamente un recorrido en anchura del RG_l . Se mantiene una estructura de datos de tipo cola (Q en el algoritmo) con pares formados por un vértice cabecera y un estado local del módulo preservado en el sistema agregado (si $l > 0$). Como se ha dicho antes, pueden aparecer estados diferentes en RG_l que tienen el mismo marcado en los lugares preservados del sistema agregado, por lo que el marcado de un determinado estado no es suficiente para las operaciones de comparación o inserción. Por ello, en el desarrollo del algoritmo se mantienen ese tipo de pares. El vértice cabecera H fija la clase de equivalencia de R_0 a la que pertenece el estado que se está explorando y el marcado local p fija el estado local del módulo preservado en el sistema agregado (si $l > 0$). Entonces hay que entender que durante la generación de RG_l se insertarán los pares completos (H, p) pero que una vez finalizados los cálculos sólo es necesario almacenar las proyecciones de estos pares sobre los nodos preservados, es decir, (H_0, p) . Al realizar la proyección no se agregará ningún estado, aunque coincida su marcado con el de otro.

El principal problema consiste en calcular el conjunto de transiciones que se pueden disparar a partir del estado de RG_l definido por el par (H, p) . Además, cada vez que se dispare una transición será necesario conocer la clase de equivalencia de R_0 del estado destino. Por el teorema 6.31, cada clase de equivalencia de R_0 queda determinada por una serie de vértices cabecera, y todos estos vértices cabecera están relacionados entre sí en el sentido de que todos tienen sucesores comunes por medio de transiciones internas del SAM. Como en el algoritmo 6.33 se han calculado los vértices cabecera que están relacionados entre sí (la estructura de datos $Rel(S)$), es suficiente conocer un vértice cabecera cualquiera del estado destino para conocer la clase

de equivalencia de R_0 a la que pertenece el nuevo estado. En consecuencia, dado un par (H, p) se calculará un conjunto (L en el algoritmo) de pares (H', t) donde t es una transición sensibilizada en (H, p) y H' es un vértice cabecera de la clase de equivalencia de destino. De esta forma es posible tratar también el caso observado de que a partir de un estado agregado haya transiciones a diferentes estados agregados disparando la misma transición.

El conjunto de transiciones sensibilizadas en el estado definido por un par (H, p) puede dividirse en tres subconjuntos; transiciones internas al módulo preservado (si $l > 0$), transiciones de interfaz del módulo preservado (si $l > 0$) y transiciones de interfaz de los módulos reducidos. Cada subconjunto debe tratarse de forma diferente.

Las transiciones internas al módulo preservado (si $l > 0$) pueden calcularse a partir del estado local p y de la estructura de datos $\text{RG}(\mathcal{S})|_{P_l}$ ya calculada en el algoritmo 6.33. Además, el disparo de una transición t de éstas no modifica la clase de equivalencia de R_0 del estado destino, por lo que hay que insertar en el conjunto L un par de la forma (H, t) con $t \in T_i \setminus \text{TI}$.

Las transiciones de interfaz del módulo preservado (si $l > 0$) pueden calcularse a partir del estado local p y del marcado en los canales del vértice cabecera H_0 . El disparo de una transición t de éstas producirá un cambio de clase de equivalencia de la cuál es necesario calcular un vértice cabecera. Para ello se toma el estado del SAM $(H_0, \dots, H_{l-1}, p, H_{l+1}, \dots, H_K)$ y se dispara la transición produciendo el estado H^* del algoritmo (es un estado alcanzable del SAM). En general, H^* puede no ser un vértice cabecera, por lo que habrá que calcular un estado predecesor suyo por transiciones internas del SAM que si lo sea (H' en el algoritmo). Entonces se inserta el par (H', t) en el conjunto L .

Por lo que respecta a las transiciones de interfaz de los módulos reducidos, se calcularán a partir de los vértices cabecera. De todos los vértices cabecera de la clase de R_0 a la que pertenece el estado que se está explorando puede haber varios que tengan un sucesor por medio de transiciones internas con el marcado local p . Estos vértices cabecera forman el conjunto C del algoritmo. Para cada vértice s del conjunto C se dispone del conjunto $A(s)$ de aristas de $\text{RG}(S)$ con las transiciones de interfaz de los módulos reducidos que se pueden disparar desde sucesores de s por transiciones internas del SAM y a qué clase de equivalencia se llega. Por lo tanto, hay que añadir al conjunto L las proyecciones de estas ternas sobre sus dos últimas componentes, es decir, eliminando el vértice cabecera inicial, que ya es conocido. Notar que la línea del algoritmo en la que se realiza esta ope-

ración está marcada con un ¹. Esto quiere decir que la operación de unión es especial. Los pares que se van añadiendo al conjunto L hay que tomarlos exceptuando vértices cabecera relacionados, es decir, en el conjunto L no habrá dos pares (s_1, t) y (s_2, t) con $(s_1, s_2) \in \text{Rel}(S)$. Esto es evidente ya que lo que se quiere denotar con los vértices cabecera destino de los disparos de las transiciones es la clase de equivalencia de R_0 destino. Como estas clases pueden tener varios vértices cabecera, hacer una unión clásica en el sentido conjuntista puede provocar la existencia de pares repetidos para el propósito del algoritmo.

Una vez calculadas todas transiciones sensibilizadas en el estado de RG_l definido por el par (H, p) junto con las clases de equivalencia de destino se procede al disparo de las mismas. El disparo de un par $(H', t) \in L$ indica que en el RG_l , desde el estado local p de la clase de equivalencia H de R_0 se puede disparar la transición t y se obtiene el estado local p' de la clase de equivalencia H' de R_0 . Lo siguiente que hay que hacer es mirar si existe un par (H'', p') en RG_l tal que $(H', H'') \in \text{Rel}(S)$. Si es así, los dos pares definen el mismo estado de RG_l , por lo que se cambia H' por H'' . Si el par (H', p') es nuevo en RG_l , se inserta en Q y en RG_l para su futura inspección. En cualquier caso se inserta la transición $(H, p) \xrightarrow{t} (H', p')$ en RG_l .

Al finalizar el recorrido en anchura se tiene un grafo que, interpretado como grafo dirigido etiquetado, es isomorfo al RG_l que se quiere calcular. No son exactamente lo mismo debido a que cada vértice calculado es un par formado por un vértice cabecera del SAM y un estado local del módulo \mathcal{S}_l . Para obtener el grafo dirigido etiquetado RG_l sólo falta proyectar estos pares sobre los lugares preservados en el sistema agregado, es decir, sobre $P_l \cup B$ (tomando $P_0 = \emptyset$). Recordar que la proyección hay que realizarla sin agregaciones de estados debido al marcado de los mismos, es decir, sólo se cambia la codificación de los estados para obtener RG_l .

Por construcción, todos los estados alcanzables de RG_i con $0 \leq i \leq K$ son proyecciones de estados del SAM original en los lugares preservados en cada sistema agregado, por lo que las inclusiones del apartado 2 del teorema 6.11 son también ahora igualdades. En cuanto a las proyección de las secuencias de disparo, en los RG 's calculados con la técnica de esta sección pueden seguir apareciendo secuencias de disparo espurias (secuencias de disparo en los sistemas agregados que no se corresponden con proyecciones de secuencias de disparo del SAM original sobre el conjunto de transiciones preservadas en el sistema agregado). En cualquier caso, esta técnica elimina las secuencias de disparo espurias que aparecían por la agregación inadecuada de estados

que se ha resuelto. Por lo tanto, los sistemas agregados que se consiguen con la técnica de esta sección tienen menos secuencias de disparo espurias que los de la técnica de la sección anterior.

La complejidad en espacio y tiempo del algoritmo 6.42 es menor que la de la primera técnica general de descomposición (sección 6.3) ya que no se genera en ningún momento ningún estado espurio. Respecto al algoritmo 6.39, es más costoso en espacio y en tiempo debido a que no se agregan tantos estados, pero tiene la ventaja de calcular sistemas agregados que permiten obtener mejores aproximaciones.

En la siguiente sección se emplearán los sistemas agregados generados con las tres técnicas de descomposición que se han desarrollado en este capítulo para aproximar el throughput de las transiciones del SAM original.

6.7 Aplicación a la aproximación de throughput

El contenido de esta sección es similar al de las secciones 3.4, 4.4 y 5.5. Se incluye aquí con las adaptaciones necesarias para SAM's con objeto de aumentar la legibilidad de la técnica completa de aproximación.

En las secciones anteriores se han desarrollado tres técnicas de descomposición diferentes para generar, a partir de un SAM con K módulos, $K + 1$ RG's que se asocian los sistemas agregados (\mathcal{LS}_i con $1 \leq i \leq K$ y \mathcal{BS}). En las tres técnicas, una vez que se adopta una visión estructurada a nivel de red se calculan los RG's asociados a los sistemas agregados resolviendo los distintos problemas que se han identificado. Aunque la forma de calcular los RG's es diferente según la técnica empleada, en las tres técnicas se pretende que cada \mathcal{LS}_i con $1 \leq i \leq K$ contenga de forma completa el funcionamiento de uno de los módulos de SAM y agregaciones del resto. El \mathcal{BS} supone una agregación del funcionamiento de todos los módulos del SAM. Ahora se van a emplear estos $K + 1$ sistemas agregados para aproximar el throughput de las transiciones del SAM original.

La técnica que se va a exponer en esta sección es básicamente la misma que la del capítulo 3 para MG's. Esta técnica es un método de aproximación del tiempo de respuesta. Las transiciones de interfaz de cada módulo reducido en cada sistema agregado aproximarán el tiempo de respuesta de todo el módulo. La técnica es iterativa y utiliza un sistema de alto nivel o esqueleto básico para no tener problemas de convergencia y de malas aproximaciones (como en MG's). El algoritmo utilizado es el siguiente:

Algoritmo 6.43

input: $\langle \mathcal{S}, w \rangle$ SAM estocástico vivo, limitado y reversible
construir $\text{RG}(\mathcal{LS}_i)$ y $\text{RG}(\mathcal{BS})$ para $1 \leq i \leq K$
seleccionar tasas iniciales $\mu_t^{(0)}$ para cada $t \in T_j \cap \text{TI}$ con $2 \leq j \leq K$
 $n := 0$ (contador de iteraciones)
repeat
 $n := n + 1$
for $i := 1$ **to** K **do**
solve \mathcal{LS}_i con:
 In: tasas $\mu_l^{(n)}$ para las transiciones de $T_l \cap \text{TI}$ con $1 \leq l < i$
 tasas $\mu_l^{(n-1)}$ para las transiciones de $T_l \cap \text{TI}$ con $i < l \leq K$
 Out: proporciones μ_i entre tasas de las transiciones de $T_i \cap \text{TI}$ y
 throughput $\chi_i^{(n)}$ de las transiciones de $T_i \cap \text{TI}$
solve \mathcal{BS} (encontrar factor de escala λ) con:
 In: tasas $\mu_l^{(n)}$ para las transiciones de $T_l \cap \text{TI}$ con $1 \leq l < i$
 tasas $\mu_l^{(n-1)}$ para las transiciones de $T_l \cap \text{TI}$ con $i < l \leq K$
 proporciones entre tasas μ_i para las transiciones de $T_i \cap \text{TI}$
 throughput $\chi_i^{(n)}$ de las transiciones de $T_i \cap \text{TI}$
 Out: tasas reales $\mu_i^{(n)} = \lambda \cdot \mu_i$ de las transiciones de $T_i \cap \text{TI}$
 tales que $\chi_{\mathcal{BS}}^{(n)} = \chi_i^{(n)}$
end for
until convergencia de $\{\chi_i^{(n)}\}_{i=1}^K$
output: throughput $\{\chi_i^{(n)}\}_{i=1}^K$ de las transiciones de $\langle \mathcal{S}, w \rangle$

En este algoritmo iterativo se van resolviendo uno por uno todos los sistemas de bajo nivel \mathcal{LS}_i (con $1 \leq i \leq K$). Cuando en la iteración n del esquema iterativo se resuelve el \mathcal{LS}_i , se obtienen para las transiciones de $T_i \cap \text{TI}$ los vectores $\chi_i^{(n)}$ de throughput y μ_i de proporciones entre tasas de servicio. Con esta información se busca, por medio del \mathcal{BS} , un factor de escala λ para las proporciones entre las tasas de disparo, de forma que los throughput de las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i coincidan. El cálculo de este factor de escala puede realizarse por medio de una búsqueda lineal en \mathcal{BS} . Se va resolviendo el \mathcal{BS} cambiando todas las tasas de las transiciones de $T_i \cap \text{TI}$ de forma que se mantengan las proporciones μ_i hasta que se consigan los mismos throughput en las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i . Notar que \mathcal{BS} tiene muchos menos estados que el sistema original e incluso

que los \mathcal{LS}_i , por lo que el coste de esta búsqueda lineal es despreciable comparado con la complejidad del estudio de los otros sistemas.

Después de este doble paso se obtiene una nueva aproximación $\mu_i^{(n)}$ para las tasas de disparo de las transiciones de $T_i \cap \text{TI}$ que se mantienen en el estudio posterior de otros \mathcal{LS}_j y en \mathcal{BS} .

Quedan por aclarar algunos puntos de este algoritmo iterativo. Por un lado, cuestiones respecto a las restricciones de selección de módulos. Por otro lado, la selección de las tasas de disparo iniciales $\mu_j^{(0)}$ de las transiciones de $T_j \cap \text{TI}$ con $2 \leq j \leq K$, que son necesarias para resolver por primera vez el sistema \mathcal{LS}_1 e iniciar el algoritmo iterativo. Otra cuestión a tener en cuenta es la forma de calcular, al resolver el \mathcal{LS}_i , las proporciones μ_i entre las tasas de disparo de las transiciones de $T_i \cap \text{TI}$. Por último, consideraciones acerca de la búsqueda lineal del factor de escala λ en \mathcal{BS} para el cálculo de las tasas de disparo reales de las transiciones de interfaz en cada iteración y el test de convergencia para decidir el final del algoritmo iterativo.

Respecto a las restricciones sobre la selección de módulos, por la definición 6.1, es posible definir los módulos de cualquier forma que induzca una partición en el conjunto de transiciones y de lugares (excluidos los canales) del SAM original. Teniendo ahora en cuenta las cuestiones numéricas, hay que añadir alguna restricción más. Si se emplean transiciones inmediatas para resolver los conflictos, no es deseable que aparezca ninguna transición de interfaz que sea inmediata. Si éste es el caso, podría haber errores de ejecución al poder obtener como medida de utilización de alguna transición de interfaz 0 y por lo tanto divisiones por 0 al calcular su tasa proporcional de disparo. Además, a diferencia de las clases de redes tratadas en los capítulos anteriores, en redes generales no se puede asegurar que la relación entre los throughput de las transiciones venga determinada por la estructura de la red. En general, los ratios de visita de las transiciones de un SAM es una función de la estructura, marcado inicial y tasas de todas las transiciones. Este hecho puede crear problemas a la hora de aplicar algoritmos como el 6.43, ya que es un algoritmo iterativo en el que se van cambiando las tasas de las transiciones de interfaz de los sistemas agregados. Si dos transiciones de interfaz del mismo módulo están en conflicto, el cambio de estas tasas en las distintas iteraciones puede provocar el cambio de sus ratios de visita y por lo tanto tener problemas de convergencia en las búsquedas del factor de escala en el \mathcal{BS} . Hay varias formas sencillas de resolver este problema. Una forma es imponer restricciones a la hora de seleccionar los módulos del SAM, de forma que no se dé esta situación. Esta solución añade

restricciones al trabajo del analista, que no necesita conocer la técnica a este nivel de detalle. Por ello se dan ahora otras soluciones más simples. Otra forma consiste en resolver los conflictos en el modelo original por medio de transiciones inmediatas. De esta forma se separa la resolución de conflictos de la duración de las actividades y desaparece el problema. La última solución consiste en refinar las transiciones de interfaz en un trío formado por una transición interna, un lugar interno y la transición de interfaz. Esta es una solución del mismo estilo de la anterior con transiciones inmediatas. El objetivo consiste en asegurar que los ratios de visita de las transiciones de interfaz no dependan de las tasas de las mismas. Así se eliminan problemas a la convergencia del método numérico.

Respecto a la selección de las tasas iniciales para las transiciones de interfaz, el algoritmo iterativo no se ve afectado, en lo que respecta a las aproximaciones de los throughput obtenidas, por la selección de estos valores. Por lo tanto, en principio se pueden seleccionar valores arbitrarios. Pero es razonable poner unos valores iniciales que tengan relación con el modelo. Una selección adecuada consiste en poner a cada transición de interfaz la tasa que tiene en el modelo original. De esta forma se reduce el número de iteraciones necesarias para alcanzar la convergencia del algoritmo. Esta elección dará en la mayor parte de los casos sistemas agregados más rápidos en la primera iteración del método pero se corrige en las siguientes.

Por lo que respecta a la estimación de las proporciones μ_i entre las tasas de disparo de las transiciones de interfaz del módulo \mathcal{S}_i , la forma de hacerlo depende de la técnica de descomposición utilizada.

En el caso de la primera técnica (sección 6.3) se utilizan los nodos calculados en la fase de descomposición estructural. Como se explicó en la sección 6.2, para cada módulo \mathcal{S}_i del SAM original se calculaba un conjunto H_i de lugares que resumen el comportamiento interno del módulo. Son estos lugares los que se emplean para realizar la estimación. El sistema de bajo nivel $\mathcal{L}\mathcal{S}_i$ contiene completamente el módulo \mathcal{S}_i y los lugares del conjunto H_i . Por lo tanto, al resolver su CTMC isomorfa, se puede calcular la utilización de cada transición de interfaz si sólo se tienen en cuenta los lugares del conjunto $H_i \cup B$ (es decir, se calcula la probabilidad de que la transición esté sensibilizada una vez se eliminan los nodos internos del módulo \mathcal{S}_i). De esta manera, se está relacionando el funcionamiento completo del módulo \mathcal{S}_i con el de su resumen para poder exportar esta información al resto de sistemas de bajo nivel. Posteriormente, basta emplear la fórmula del cálculo del throughput del teorema 2.87 para encontrar las proporciones entre las tasas

de disparo de las transiciones de interfaz. En la iteración n del algoritmo, la tasa proporcional de disparo de la transición $t \in T_i \cap \text{TI}$ es:

$$\mu_i[t] = \frac{\chi_i^{(n)}[t]}{P\{\mathbf{m}[p] \geq W(p, t) \quad \forall p \in \bullet t \cap (H_i \cup B)\}}$$

Esta expresión es válida si se emplea semántica de un solo servidor. Si se emplea semántica de infinitos servidores, hay que sustituir la utilización de las transiciones por su grado medio de sensibilización, es decir, la tasa proporcional de disparo es en este caso:

$$\mu_i[t] = \frac{\chi_i^{(n)}[t]}{E[\max\{k \mid \mathbf{m}[p] \geq kW(p, t) \quad \forall p \in \bullet t \cap (H_i \cup B)\}]}$$

En el caso de las técnicas de descomposición segunda (sección 6.5) y tercera (sección 6.6), la estimación de utilidades o grados medios de sensibilización es un poco más complicado, ya que no se tienen los conjuntos de lugares H_i con $1 \leq i \leq K$. En estos casos hay que modificar ligeramente los algoritmos 6.39 y 6.42 añadiendo nuevas estructuras de datos que calculen, para cada transición de interfaz en cada RG_i con $1 \leq i \leq K$ (los RG's asociados a los \mathcal{LS}_i) su utilización o grado medio de sensibilización si no se tiene en cuenta el módulo \mathcal{S}_i preservado. Este cálculo es posible realizarlo a la vez que se construyen los RG's ya que, a partir del estado local del módulo \mathcal{S}_i y el estado de sus canales de entrada y salida, es posible calcular qué transiciones de $T_i \cap \text{TI}$ se pueden disparar a partir de los sucesores del estado local por medio de transiciones de $T_i \setminus \text{TI}$. No se han incluido estos cálculos en los algoritmos 6.39 y 6.42 por simplificar su exposición. Una vez estimada la utilización o grado medio de sensibilización de una transición de interfaz, empleando alguna de las fórmulas anteriores se obtienen sus tasas proporcionales de disparo.

En cuanto al algoritmo de búsqueda lineal del factor de escala en \mathcal{BS} para calcular las tasas reales de disparo de las transiciones de interfaz, hay que tener en cuenta lo siguiente. Se quiere conseguir que coincidan los throughput de las transiciones comunes a \mathcal{BS} y \mathcal{LS}_i , es decir, que coincidan los throughput de las transiciones de interfaz. El throughput de las transiciones de estos sistemas es una función racional (cociente de polinomios) con respecto a estas variables. Como se han calculado unas proporciones entre las tasas de disparo que se quieren mantener en \mathcal{BS} , lo que se quiere calcular es un factor de escala λ de forma que coincidan los throughput de las

transiciones de interfaz en \mathcal{BS} y \mathcal{LS}_i . En las clases de redes de los capítulos anteriores (MG's, WTS's y DSSP's) era posible calcular el throughput de cualquier transición si se conocía el de una de ellas y la estructura de la red. Esta propiedad se traducía en este punto del algoritmo en que bastaba fijarse en una única transición de referencia para buscar el factor de escala, es decir, se tenía una ecuación racional. Este hecho no es cierto en general para SAM's, por lo que en este caso se tiene un sistema de ecuaciones racionales en λ , con una ecuación por cada transición de interfaz del módulo que se esté estudiando:

$$\chi_{\mathcal{BS}}^{(n)}(\lambda) = \chi_{\mathcal{LS}_i}^{(n)}$$

Resolver un sistema de ecuaciones racionales tiene mayor complejidad que resolver una única ecuación, por lo que es importante poder obtener una única ecuación a partir de este sistema. Afortunadamente esto es posible tomando normas, por ejemplo la norma $\|\cdot\|_\infty$ (ver definición 2.7). Cualquier otra norma también es válida por ser todas equivalentes en \mathbb{R}^n . Esto no supone ningún problema, ya que dos vectores $\mathbf{u} = (u_1, \dots, u_n)$ y $\mathbf{v} = (v_1, \dots, v_n)$ son iguales si y sólo si $\|\mathbf{u} - \mathbf{v}\|_\infty = \max_{1 \leq i \leq n} \{|u_i - v_i|\} = 0$. Por lo tanto, la ecuación racional que se va a resolver es:

$$\|\chi_{\mathcal{BS}}^{(n)}(\lambda) - \chi_{\mathcal{LS}_i}^{(n)}\|_\infty = 0$$

Esta es una ecuación racional en λ cuya expresión explícita no se conoce por su complejidad. El término $\chi_{\mathcal{LS}_i}^{(n)}$ es constante respecto a λ y se obtiene del estudio del \mathcal{LS}_i realizado justo antes de empezar el estudio de \mathcal{BS} . Por su parte, $\chi_{\mathcal{BS}}^{(n)}(\lambda)$ es una función de λ , que devuelve el vector de throughput en \mathcal{BS} . Calculando una solución λ de la ecuación anterior, se puede asegurar que el throughput de todas las transiciones de interfaz coincide en \mathcal{BS} y \mathcal{LS}_i . El problema numérico de resolución de esta ecuación puede resolverse con algoritmos como el de la secante. En este punto, mientras no se tenga una demostración formal de existencia de solución podría ocurrir que la ecuación no tenga solución y que el método numérico no finalice. En ninguna prueba realizada hasta el momento se han tenido problemas en este punto.

Por último, queda por aclarar el test de convergencia. Este algoritmo numérico calcula unas aproximaciones de throughput aceptables para una técnica de aproximación (por debajo del 5%). Teniendo en cuenta la aproximación que se va a obtener al final, no es necesario que el test de convergencia sea muy exigente ya que esto provocaría el cálculo de un mayor número de

iteraciones sin que por ello se aumente la precisión de los resultados obtenidos. Si se pone un test de convergencia poco exigente, entonces el algoritmo finalizará antes de lo debido, aumentando en general el error de la aproximación. El test de convergencia que se utiliza consiste en iterar hasta que la diferencia entre los throughput de las transiciones de interfaz de todos los \mathcal{LS}_i en dos iteraciones consecutivas del método sea menor que el 0.1%.

En este momento, sería necesario estudiar las propiedades numéricas del algoritmo iterativo para demostrar propiedades de convergencia, unicidad de solución e incluso estimar el error cometido en la aproximación, pero se ha dejado como trabajo futuro. A falta de un mejor soporte teórico, la experimentación numérica con diferentes ejemplos sugiere que, en el caso de la tercera técnica de descomposición (sección 6.6), el esquema converge en muy pocos pasos (entre 3 y 6 iteraciones), converge a la misma solución independientemente de las tasas iniciales elegidas para iniciar el algoritmo, y el error cometido al final del mismo (inferior al 5%) es admisible para técnicas de aproximación en evaluación de prestaciones de sistemas concurrentes basada en modelos formales. Respecto a las técnicas de descomposición de las secciones 6.3 y 6.5 se han encontrado casos en los que la aproximación obtenida es muy pobre, pero pueden funcionar en ejemplos que no tengan comportamientos anómalos de los que se han identificado a lo largo del capítulo.

6.8 Conclusiones

En este capítulo se han desarrollado los trabajos de [PJC98, PJC99a] sobre técnicas de descomposición de redes de Petri para su análisis numérico. La descomposición viene inducida por la estructura de la red, a la que se le da una visión estructurada.

Se puede observar que una visión estructurada a nivel de red induce una visión estructurada en el RG. Esta visión estructurada del RG permite observar ciertas estructuras regulares sobre los estados alcanzables de una red de Petri y las transiciones entre ellos. El conocimiento de estas estructuras regulares permite por ejemplo generar una descripción descompuesta del RG consumiendo menos recursos, tanto de espacio como de tiempo que el algoritmo clásico. Como ejemplo se expone un algoritmo más eficiente en tiempo que el clásico para la generación del RG de una red a partir de su descripción descompuesta.

Se ha aprovechado la ocasión para introducir algunas mejoras técnicas

en el desarrollo teórico y en los algoritmos de descomposición para mejorar su eficiencia sin disminuir su legibilidad.

Todo lo expuesto en este capítulo son aportaciones de esta memoria, excepto los contenidos de la sección 6.2. En concreto estas aportaciones son:

- Exposición de la visión estructurada del RG de una red de Petri inducida por una visión estructurada a nivel de red.
- Algoritmo de cálculo del RG de una red de Petri más eficiente en tiempo que el clásico.
- Tres técnicas generales de descomposición de redes de Petri para su análisis numérico.
- Adaptación del método numérico de aproximación del throughput de transiciones para cualquier red de Petri.

Como aplicación de la descripción descompuesta del RG de un SAM a la aproximación de throughput de sus transiciones, se exponen tres técnicas de descomposición de la red original. Las principales características de las tres técnicas de descomposición son las siguientes.

La primera técnica, desarrollada en la sección 6.3, simplemente genera sistemas agregados ergódicos y por lo tanto con distribución única en estado estacionario. Estos sistemas agregados pueden incluir numerosos estados espurios, es decir, estados alcanzables en los sistemas agregados que no se corresponden con proyecciones de estados de la red original sobre los lugares preservados en el sistema agregado. Esta técnica sirve por ejemplo para MG's, pero en general daría errores superiores al 5% para WTS's o clases más generales.

La segunda técnica, desarrollada en la sección 6.5, genera sistemas agregados sin estados espurios, es decir, sistemas que son proyecciones exactas, (desde el punto de vista de marcados alcanzables) del sistema original en los lugares preservados en el sistema agregado. Pero los sistemas agregados pueden tener secuencias de disparo espurias.

La tercera técnica, desarrollada en la sección 6.6, genera sistemas agregados que no contienen estados espurios (como la segunda) pero que además elimina ciertas agregaciones de estados que llevan a comportamientos de los sistemas agregados muy diferentes del de la red original. Desde el punto de

vista de secuencias de disparo, esta técnica reduce algunas de las secuencias de disparo espurias que podían tener los sistemas agregados generados con la segunda técnica, pero no todas. Lo que se hace es definir un criterio de agregación por medio de relaciones de equivalencia que no se fija únicamente en el marcado de los estados. Es por lo tanto un criterio de agregación más fuerte que permite diferenciar estados que en la segunda técnica resultarían agregados. Con esta tercera técnica se esperan los mejores resultados.

Las tres técnicas preservan ciertas propiedades de la red original, como son la vivacidad, limitación y existencia de estados recurrentes. Todas las técnicas tienen una complejidad, tanto en espacio como en tiempo inferior a la solución clásica, ya que nunca se calcula ni se resuelve la CTMC de la red original, sino varias más pequeñas correspondientes a los sistemas agregados.

El algoritmo iterativo de aproximación es, salvo ligeras adaptaciones a la nueva clase de redes, el mismo de [CCJS94] para MG's que se ha explicado en detalle en el capítulo 3.

Las tres técnicas de descomposición desarrolladas en este capítulo, aplicadas a un MG dan exactamente los mismos sistemas agregados que los del capítulo 3, por lo que los resultados numéricos serán exactamente los mismos. Esto ya no es cierto para WTS's ni DSSP's, ya que las técnicas de descomposición desarrolladas en los capítulos anteriores para estas clases de redes no mantenían las proyecciones de estados del sistema original sobre los nodos preservados en cada subsistema, por lo que los resultados que se obtengan con las técnicas de este capítulo (sobre todo la tercera) serán normalmente mejores.

Queda como trabajo futuro estudiar las propiedades numéricas (convergencia, unicidad de solución, etc.) del algoritmo iterativo de aproximación del tiempo de respuesta. Se ha intentado demostrar sin éxito completo hasta el momento.

Capítulo 7

Comparación numérica de técnicas

En los capítulos 3, 4 y 5 anteriores se han desarrollado técnicas para el cálculo aproximado del throughput de las transiciones de grafos marcados (MG's), grafos marcados con pesos (WTS'S) y sistemas deterministas de procesos secuenciales (DSSP's) respectivamente. Finalmente, en el capítulo anterior se han desarrollado tres técnicas de descomposición para el análisis numérico de redes de Petri generales. Posteriormente se adaptó el método numérico de aproximación del throughput de transiciones. En los desarrollos teóricos no se han incluido ejemplos de aplicación de estas técnicas, porque se ha preferido ponerlas en este capítulo.

Aprovechando por una parte que se tienen varias técnicas para diversas subclases de redes y por otra parte que las subclases que han sido objeto de estudio en esta memoria son a su vez subclases unas de otras (todo MG es un WTS, todo WTS es un DSSP y todo DSSP es una red de Petri) se ha decidido dedicar este capítulo no sólo para exponer una serie de ejemplos de aplicación de las técnicas, sino también para comparar los resultados obtenidas con todas las técnicas que sean de aplicación para cada ejemplo. De esta forma, el capítulo está organizado en cuatro secciones, cada una dedicada a una subclase de redes. Dentro de cada sección se desarrollarán diversos ejemplos y para cada ejemplo se aplicarán todas las técnicas desarrolladas en la memoria que sean posibles. De esta forma se obtiene una extensa batería de ejemplos que permiten observar y comparar el funcionamiento de las técnicas desarrolladas en la memoria.

Los ejemplos se han tomado de los distintos artículos publicados en el desarrollo de esta memoria y no tienen por objeto modelar sistemas reales sino exponer las características de las técnicas expuestas en los capítulos anteriores. En general, las redes de ejemplo son suficientemente complejas como para que el problema de descomposición y reducción del modelo no sea en absoluto trivial y sea necesario emplear todas las técnicas anteriores.

En cuanto a la interpretación temporal de las redes, por uniformidad se ha decidido emplear en todos los ejemplos y para todas las transiciones semántica de un solo servidor. Esta elección se debe a los programas que ha sido necesario implementar para realizar las pruebas. En una primera versión, con el propósito de experimentación, los programas debían ser lo más sencillos posible y es más fácil implementar el cálculo de los distintos índices de prestaciones si se utiliza semántica de un solo servidor que si se utiliza semántica de infinitos servidores (es más fácil comprobar si una transición está sensibilizada que calcular su grado de sensibilización). Para efectuar pruebas con semántica de infinitos servidores es necesario realizar cambios en los programas y esta tarea se ha dejado como trabajo futuro ya que el número aproximado de líneas de código que ha habido que codificar ha sido de 7000.

En las tablas que se presentan en todo este capítulo aparecen los números con una cantidad variable de dígitos simplemente con el objetivo de que las tablas quepan en la página con un tamaño de letra razonable, pero todos los cálculos se han realizado con 7 cifras en base 10 de precisión. Entonces es posible que los errores indicados en las distintas aproximaciones difieran ligeramente de los que se pueden calcular a partir de los números que aparecen en las tablas. Los errores reales cometidos son los que aparecen en las tablas y han sido calculados a partir de los números con 7 cifras en base 10 de precisión.

7.1 Grafos marcados

En esta sección se van a desarrollar dos ejemplos de MG's tomados de [CCJS94]. En principio, dado que los MG's son la clase de redes más restringida de las que se han tratado en la memoria, se les puede aplicar todas las técnicas anteriores. Sin embargo, el hecho de que la descomposición y reducción presentada en el capítulo 3 consiga sistemas agregados que son proyecciones exactas funcionales del sistema original (en términos

de marcados alcanzables y de secuencias de disparo) hace que las distintas técnicas de esta memoria sean equivalentes en el caso de los MG's. Por ejemplo, la técnica de descomposición y reducción de WTS's es completamente equivalente a la técnica de MG's como se ha visto en el capítulo 4. Las técnicas desarrolladas para DSSP's consisten en la reducción de SM's por lo que salvo en casos excepcionales (un MG que tenga a su vez subredes que también sean SM's) no tiene sentido su aplicación. Por último, las técnicas desarrolladas en el capítulo anterior tenían por objeto eliminar a nivel de grafo de alcanzabilidad comportamientos anómalos de los sistemas agregados (bloqueos totales, marcados alcanzables o secuencias de disparo espurias). Como se ha demostrado en el capítulo 3 los sistemas agregados generados con la técnica de descomposición no tienen ningún comportamiento anómalo respecto al del sistema original, lo que convierte las tres técnicas del capítulo anterior en equivalentes cuando se aplican a un MG. Es posible que el conjunto de lugares implícitos que se calculan con la primera técnica general sea un superconjunto del que se calcula con la técnica del capítulo 3, pero en ese caso lo que ocurre es que el superconjunto contiene lugares que son implícitos respecto a otros. La diferencia entre las dos técnicas consiste en que en la técnica para MG's se ha incluido un mecanismo de reducción del número de lugares implícitos para evitar redundancias, pero los grafos de alcanzabilidad de los sistemas agregados que se obtienen son isomorfos, por lo que la aproximación que se obtiene es la misma.

Por lo tanto, en esta sección se va a desarrollar completamente la técnica de aproximación de throughput desarrollada en el capítulo 3 con dos ejemplos. Con el resto de técnicas los resultados serán exactamente los mismos, al producir sistemas agregados con grafos de alcanzabilidad isomorfos.

El primer ejemplo es el de la figura 7.1. La red es un MG ya que es ordinaria y todos sus lugares tienen una sola transición de entrada y una sola transición de salida (definición 2.89). Además, es fuertemente conexo y vivo, por lo que tiene distribución única en estado estacionario. El MG tiene una estructura de red asimétrica y su grafo de alcanzabilidad tiene 89358 estados. El objetivo de la técnica de aproximación consiste en cortar la red en varios trozos (2 en este caso) y a partir de ellos construir los sistemas agregados con los que se realizarán los cálculos. Estos sistemas agregados tendrán menos estados alcanzables y por ello su resolución será menos costosa. En este caso se ha seleccionado un corte formado por los lugares B_1 , B_2 y B_3 , que divide la red original en dos subredes (\mathcal{N}_1 y \mathcal{N}_2). La subred \mathcal{N}_1 está formada por las transiciones internas $\{T_{1i}\}_{i=1}^7$, las transiciones de interfaz $\{I_{1i}\}_{i=1}^3$

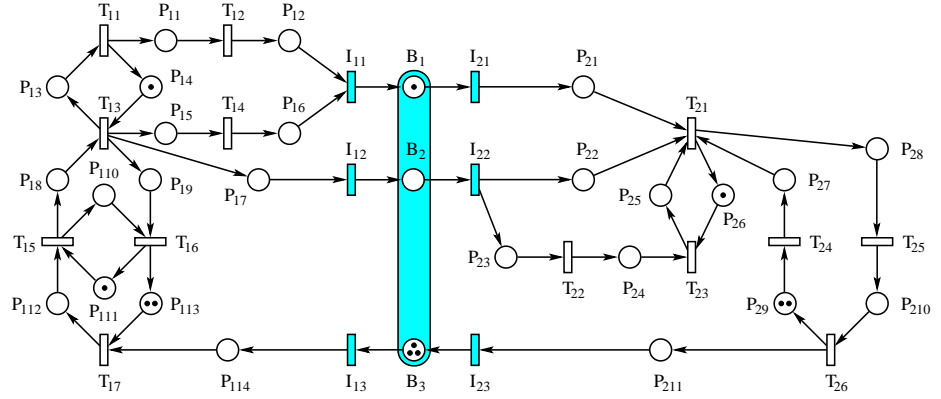


Figura 7.1: Ejemplo 1 de MG's.

y los lugares $\{P_{1i}\}_{i=1}^{14}$. La subred \mathcal{N}_2 está formada por las transiciones internas $\{T_{2i}\}_{i=1}^6$, las transiciones de interfaz $\{I_{2i}\}_{i=1}^3$ y los lugares $\{P_{2i}\}_{i=1}^{11}$. Las transiciones de interfaz son las que tienen lugares de entrada o salida pertenecientes al corte. La notación de las etiquetas será en la misma para todos los ejemplos de este capítulo.

El siguiente paso consiste en calcular los sistemas agregados. En este caso, como el corte de la red original se ha hecho en dos subredes, habrá que generar dos sistemas de bajo nivel (\mathcal{LS}_1 y \mathcal{LS}_2) y un sistema de alto nivel o esqueleto básico (\mathcal{BS}). Cada \mathcal{LS}_i está formado por la subred \mathcal{N}_i y una agregación de la otra. El \mathcal{BS} está formado por las agregaciones de las dos subredes. Por lo tanto, es necesario calcular previamente las agregaciones de cada subred. Como se ha explicado en el capítulo 3, las agregaciones de las subredes se realiza por medio de lugares estructuralmente implícitos marcados de transición a transición (TT-MSIP's) junto con un marcado que los hace implícitos. El cálculo de estos lugares y de sus marcados iniciales se realiza por medio del algoritmo 3.13. Este algoritmo aseguraba el cálculo un conjunto reducido de TT-MSIP's (en el sentido de que ningún lugar del conjunto calculado es implícito respecto a otros) necesarios para reducir una subred. Aquí se va a desarrollar la ejecución del algoritmo para ver la interpretación de las estructuras de datos que gestiona.

El algoritmo 3.13 toma como entrada el grafo dirigido asociado al MG que se quiere descomponer junto con la información del número de marcas de cada arista del grafo. El grafo dirigido asociado a un MG tiene un vértice

Por lo que respecta a los elementos de \mathbf{L} que no aparecen en blanco, por ejemplo $(T_{11}, T_{13}) = 1$, su interpretación es que hay un lugar en el MG que conecta T_{11} con T_{13} (el lugar P_{14}) y que este lugar tiene 1 marca en el marcado inicial. En consecuencia, la matriz \mathbf{L} tiene inicialmente tantos elementos distintos de infinito como lugares tiene el MG. De esta manera, la matriz \mathbf{L} contiene toda la información sobre la estructura y marcado inicial del MG. A partir de ella es posible calcular los caminos de menor número de marcas entre cualesquiera dos transiciones del MG.

Al aplicar el algoritmo 3.13 a la matriz \mathbf{L} , se producen como salida dos matrices \mathbf{L} y \mathbf{B} (\mathbf{L} puede ser la misma matriz en la implementación real, aunque la interpretación de sus datos es diferente a la que se ha hecho antes). Aunque estas matrices son de tamaño 19×19 , para efectuar la reducción de las subredes producto del corte del MG sólo interesan los datos correspondientes a las transiciones de interfaz, es decir, son de tamaño 6×6 en este caso.

$$\mathbf{L} = \begin{array}{c} I_{11} \\ I_{12} \\ I_{13} \\ I_{21} \\ I_{22} \\ I_{23} \end{array} \begin{array}{c} I_{11} \\ I_{12} \\ I_{13} \\ I_{21} \\ I_{22} \\ I_{23} \end{array} \begin{bmatrix} 0 & 4 & 4 & 1 & 4 & 1 \\ 3 & 0 & 3 & 4 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 3 & 0 \\ 3 & 3 & 3 & 4 & 0 & 0 \\ 3 & 3 & 3 & 4 & 3 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{array}{c} I_{11} \\ I_{12} \\ I_{13} \\ I_{21} \\ I_{22} \\ I_{23} \end{array} \begin{array}{c} I_{11} \\ I_{12} \\ I_{13} \\ I_{21} \\ I_{22} \\ I_{23} \end{array} \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

La interpretación de estas matrices es la siguiente. La matriz \mathbf{L} contiene el número de marcas de los caminos de menor número de marcas que conectan dos transiciones de interfaz cualesquiera. Por ejemplo, $\mathbf{L}(I_{13}, I_{11}) = 0$ indica que el camino de menor número de marcas en el MG que conecta I_{13} con I_{11} tiene 0 marcas en total. En este caso, un camino mínimo entre I_{13} e I_{11} es el formado por los lugares $P_{114}P_{112}P_{18}P_{15}P_{16}$ y que no tiene marcas.

Por lo que respecta a la matriz \mathbf{B} , sus elementos son variables lógicas, es decir, sólo toman valores 0 y 1. Un elemento de valor 0, por ejemplo $\mathbf{B}(I_{13}, I_{11})$, indica que ningún camino de menor número de marcas en el MG que conecta I_{13} con I_{11} pasa por otras transiciones de interfaz, es decir, por la definición 3.14, en la reducción de las subredes hace falta poner un lugar que conecte estas dos transiciones para resumir este conjunto de caminos que será eliminado en algunos sistemas agregados. Un elemento de valor 1, por ejemplo el $\mathbf{B}(I_{11}, I_{12})$, indica que algún camino de menor número de

marcas en el MG que conecta I_{11} con I_{12} pasa por otras transiciones de interfaz y por el teorema 3.5 no hace falta poner un lugar implícito que resuma este camino porque será implícito respecto a otros. En este caso, la matriz \mathbf{L} indica que el menor número de marcas entre I_{11} e I_{12} es 4. Un camino de menor número de marcas entre I_{11} e I_{12} es el que forman los lugares $B_1P_{21}P_{28}P_{210}P_{211}B_3P_{114}P_{112}P_{18}P_{17}$ que pasa por las transiciones de interfaz I_{21} , I_{23} e I_{13} . Entonces, su reducción puede lograrse por medio de la reducción de los caminos de menor número de marcas de I_{11} a I_{21} , de I_{21} a I_{23} , de I_{23} a I_{13} y de I_{13} a I_{12} .

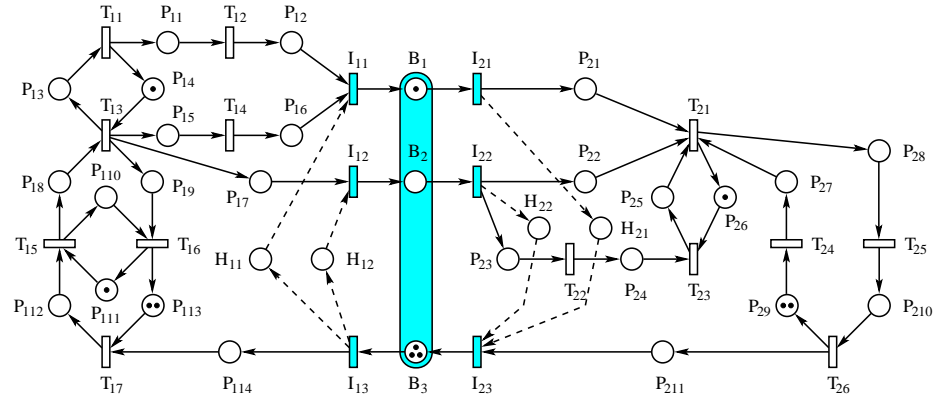
En consecuencia, hay que poner tantos lugares implícitos como elementos nulos tenga la matriz \mathbf{B} . En este punto hay que notar que algunos de los elementos nulos de \mathbf{B} se corresponden con los lugares del corte o lugares ya presentes en la red inicial. Obviamente no será necesario duplicar lugares. En este caso, de los 7 elementos nulos de \mathbf{B} , tres se corresponden con lugares del corte, en concreto $\mathbf{B}(I_{11}, I_{21})$, $\mathbf{B}(I_{12}, I_{22})$ y $\mathbf{B}(I_{23}, I_{13})$.

Por lo tanto, la interpretación de las matrices \mathbf{L} y \mathbf{B} de este ejemplo señala que para reducir las subredes producto del corte son necesarios 4 lugares implícitos que conectan los pares de transiciones (I_{13}, I_{11}) , (I_{13}, I_{12}) , (I_{21}, I_{23}) e (I_{22}, I_{23}) . Ninguno de estos lugares necesita marcas inicialmente para ser implícitos si se observan las entradas correspondientes de la matriz \mathbf{L} . Los lugares implícitos que resumen el comportamiento de la subred \mathcal{N}_i se denotarán con etiquetas H_{ij} con $1 \leq j \leq m$.

Si se añaden a la red original los lugares implícitos calculados anteriormente se obtiene el sistema extendido \mathcal{ES} de la figura 7.2. La razón de construir el \mathcal{ES} , es que a partir de él es inmediato generar los sistemas agregados.

En efecto, para construir los sistemas de bajo nivel a partir del \mathcal{ES} basta con eliminar los nodos internos a las subredes que se quieren reducir. Por ejemplo, para construir el \mathcal{LS}_1 basta con eliminar los nodos internos de la subred \mathcal{N}_2 , es decir, los lugares P_{2i} con $1 \leq i \leq 11$ y las transiciones internas T_{2i} con $1 \leq i \leq 6$. La construcción del \mathcal{LS}_2 es similar a la anterior, pero eliminando los nodos internos de la subred \mathcal{N}_1 . Por último, para construir el \mathcal{BS} a partir del \mathcal{ES} hay que eliminar todos los nodos internos de todas las subredes. En las figuras 7.3.a, 7.3.b y 7.3.c se pueden observar los sistemas agregados correspondientes al corte realizado en la red original. El \mathcal{LS}_1 tiene 8288 estados alcanzables, el \mathcal{LS}_2 3440 y el \mathcal{BS} 231.

Para aproximar el throughput de las transiciones del MG original basta aplicar el algoritmo iterativo 3.18 a los sistemas agregados. Se han realizado

Figura 7.2: \mathcal{ES} del MG de la figura 7.1

dos experimentos, el primero poniendo las tasas de todas las transiciones a 1 (semántica de un solo servidor) y el segundo cambiando al azar el valor de algunas tasas.

\mathcal{LS}_1					\mathcal{LS}_2				
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	λ
0.26842	0.98452	0.93938	0.26842	1.21037	0.20744	0.20744	0.20744	0.81557	1.12386
0.20628	0.86585	0.75096	0.20628	1.29283	0.20737	0.20737	0.20737	0.80851	1.12733
0.20741	0.86870	0.75464	0.20741	1.29166	0.20739	0.20739	0.20739	0.80871	1.12726
0.20739	0.86865	0.75458	0.20739	1.29166	0.20739	0.20739	0.20739	0.80870	1.12726
$\mathcal{X}(I_{11})$ exacto: 0.202947									Error: +2.19%

Tabla 7.1: Aproximación de \mathcal{X} para Fig. 7.1. Tasas 1.0

En la tabla 7.1 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas 1. En este caso, el throughput exacto de las transiciones de la red original es 0.202947. La tabla está dividida en dos partes encabezadas con \mathcal{LS}_1 y \mathcal{LS}_2 en las que se recogen los resultados del estudio de cada sistema de bajo nivel y la correspondiente visita al \mathcal{BS} . La información que se calcula en cada \mathcal{LS}_i es el throughput de una transición del mismo (normalmente será la transición de interfaz I_{i1}) y las utilizaciones de las transiciones de interfaz de la subred \mathcal{N}_i presente en el \mathcal{LS}_i (columnas encabezadas con μ_{ij}) teniendo en cuenta únicamente los lugares implícitos y los del corte. Esta información se utiliza en el \mathcal{BS} para encontrar, por medio de un factor de escala λ , nuevas aproximaciones de las tasas de disparo de

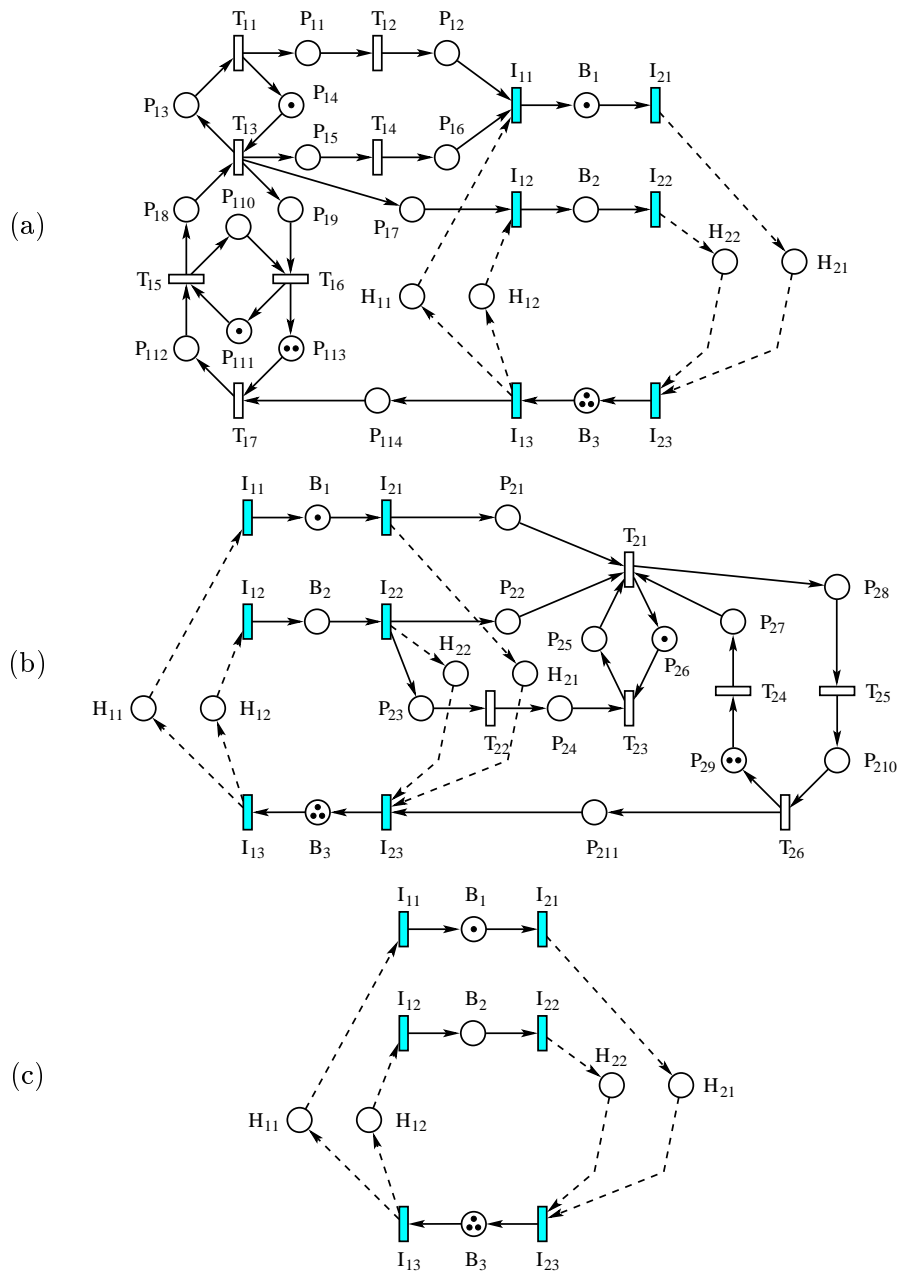


Figura 7.3: (a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del MG de la figura 7.1.

las transiciones de interfaz de \mathcal{LS}_i . En estas condiciones, la nueva tasa de la transición de interfaz I_{ij} será $\lambda \mathcal{X}(I_{ij})/\mu_{ij}$. Estas nuevas tasas no se incluyen en la tabla para no aumentar su tamaño. En cada fila de la tabla aparecen los resultados de cada iteración del algoritmo, hasta alcanzar la convergencia. En la última fila de la tabla aparece a la izquierda el throughput exacto de una transición cualquiera de la red original (por el corolario 2.94, en MG's todas las transiciones tienen el mismo throughput, pero en otras clases de redes es necesario fijar la transición de referencia) y a la derecha el error de la aproximación. En este caso se observa que se alcanza la convergencia en 4 iteraciones, obteniendo una aproximación de throughput de 0.207389 por lo que el error en la aproximación final es del +2.19%.

\mathcal{LS}_1					\mathcal{LS}_2				
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	λ
0.16498	0.98624	0.63513	0.16498	1.08627	0.13865	0.13865	0.46216	0.70623	1.09751
0.13632	0.91761	0.52990	0.13632	1.14830	0.13723	0.13723	0.45742	0.68602	1.09920
0.13729	0.92087	0.53352	0.13729	1.14612	0.13726	0.13726	0.45754	0.68660	1.09915
0.13726	0.92077	0.53341	0.13726	1.14618	0.13726	0.13726	0.45754	0.68657	1.09915
0.13726	0.92077	0.53342	0.13726	1.14618	0.13726	0.13726	0.45754	0.68658	1.09915
$\mathcal{X}(I_{11})$ exacto: 0.138341									Error: -0.78%

Tabla 7.2: Aproximación de \mathcal{X} para Fig. 7.1. Tasas variables.

En la tabla 7.2 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas variables. En concreto se han elegido al azar las siguientes tasas; $I_{11} = 0.2$, $I_{22} = 0.3$, $T_{22} = 0.5$, $T_{23} = 0.6$, $T_{12} = 0.7$, $T_{11} = 0.8$ y el resto tasa 1.0. En este caso, el throughput exacto de las transiciones de la red original es 0.138341. La leyenda de la tabla es la misma que la anterior y se puede observar que el algoritmo converge en 5 iteraciones, obteniendo una aproximación de throughput de 0.137261 lo que supone un error en la aproximación del -0.78% .

La diferencia entre los resultados de la tabla 7.2 y los de [CCJS94] se deben a ligeras modificaciones introducidas en el algoritmo de aproximación. En concreto, en el experimento de [CCJS94] se utilizaba semántica de infinitos servidores en las transiciones de interfaz mientras que aquí se ha utilizado semántica de un solo servidor. Al modificar la semántica de disparo de las transiciones de interfaz, hay que modificar la estimación de las tasas proporcionales de disparo e las transiciones de interfaz en los \mathcal{LS}_i . En [CCJS94] se calculaba el grado medio de sensibilización de las mismas, y aquí se ha calculado la utilización (ver definición 2.68). Estas diferencias hacen

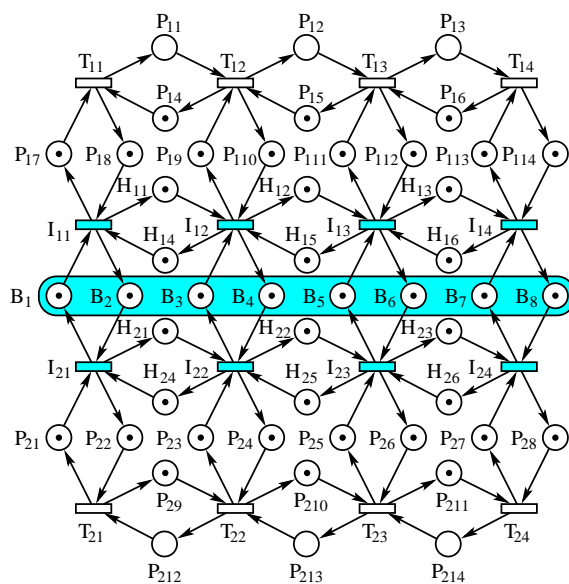


Figura 7.4: Ejemplo 2 de MG's.

que las tasas de las distintas transiciones de interfaz que se van calculando en el algoritmo iterativo sean ligeramente diferentes en ambos experimentos y que la aproximación final también lo sea. En cualquier caso, la diferencia en la calidad de la aproximación es despreciable. Otro factor que influye en las diferencias numéricas es el cambio de plataforma de ejecución y de las versiones de la aplicación utilizada, en este caso el GreatSPN [Chi87]. Estos cambios motivan ligeras modificaciones en el tratamiento y redondeo interno de los datos en coma flotante (por ejemplo todos los números reales tanto en precisión simple como en doble precisión). Estas diferencias pueden acumularse y como consecuencia hay ligeras diferencias en los resultados obtenidos incluso en el cálculo del throughput de una transición en una red sencilla.

La estructura de la red que se ha estudiado hasta ahora era asimétrica, es decir, las subredes producto del corte eran diferentes entre sí. El segundo ejemplo que se va a estudiar es el de un MG con estructura muy simétrica, de forma que cualquier corte produzca subredes con la misma estructura y muy conectadas entre sí. Éste es el caso del MG de la figura 7.4. La red es un MG ya que es ordinaria y todos sus lugares tienen una sola transición de entrada y una sola transición de salida (definición 2.89). Además,

es fuertemente conexo y vivo, por lo que tiene distribución única en estado estacionario. Su grafo de alcanzabilidad tiene 49398 estados. Se ha seleccionado un corte formado por los lugares $\{B_i\}_{i=1}^8$ que divide la red original en dos subredes (\mathcal{N}_1 y \mathcal{N}_2). La subred \mathcal{N}_1 (la de la parte superior) está generada por las transiciones internas $\{T_{1i}\}_{i=1}^4$ y las transiciones de interfaz $\{I_{1i}\}_{i=1}^4$. La subred \mathcal{N}_2 está generada por las transiciones internas $\{T_{2i}\}_{i=1}^4$ y las transiciones de interfaz $\{I_{2i}\}_{i=1}^4$. Se ha cambiado ligeramente la notación de algunos lugares internos (etiquetas H_{ij} en vez de P_{ij}) porque luego resultarán ser lugares implícitos necesarios para resumir el comportamiento de las subredes, es decir, se ha tomado una notación tal que mantenga las etiquetas de los lugares implícitos.

Como en el ejemplo anterior, ahora hay que calcular los sistemas agregados, que serán dos sistemas de bajo nivel (\mathcal{LS}_1 y \mathcal{LS}_2) y el esqueleto básico (\mathcal{BS}). Para ello hay que calcular los lugares implícitos que resumen las subredes, por medio del algoritmo 3.13. En este caso, la matriz \mathbf{L} que contiene la información de la estructura y marcado inicial del MG es de tamaño $n \times n$ con $n = 16$ porque el MG tiene 16 transiciones.

$$\mathbf{L} = \begin{matrix} & T_{11} & T_{12} & T_{13} & T_{14} & T_{21} & T_{22} & T_{23} & T_{24} & I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \\ \begin{matrix} T_{11} \\ T_{12} \\ T_{13} \\ T_{14} \\ T_{21} \\ T_{22} \\ T_{23} \\ T_{24} \\ I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{matrix} & \left[\begin{array}{cccc|cccc|cccc} 0 & & & & & & & & 1 & & & & & & & & \\ 1 & 0 & & & & & & & & 1 & & & & & & & & \\ & 1 & 0 & & & & & & & & 1 & & & & & & & \\ & & 1 & & & & & & & & & 1 & & & & & & \\ \hline & & & & & 1 & & & & & & & & 1 & & & & \\ & & & & & 0 & 1 & & & & & & & & 1 & & & \\ & & & & & & 0 & 1 & & & & & & & & 1 & & \\ & & & & & & & 0 & & & & & & & & & 1 & \\ \hline 1 & & & & & & & & & 1 & & & & 1 & & & & \\ & 1 & & & & & & & & 1 & 1 & & & 1 & & & & \\ & & 1 & & & & & & & & 1 & 1 & & & 1 & & & \\ & & & 1 & & & & & & 1 & & & & & 1 & & & 1 \\ & & & & 1 & & & & & 1 & & & & 1 & & & & \\ & & & & & 1 & & & & & 1 & & & 1 & & 1 & & \\ & & & & & & 1 & & & & & 1 & & 1 & & 1 & & \\ & & & & & & & 1 & & & & & 1 & & 1 & & 1 & \\ & & & & & & & & 1 & & & & 1 & & 1 & & 1 & \end{array} \right] \end{matrix}$$

Al igual que en el ejemplo anterior, los elementos que aparecen en blanco son los pares de vértices que no son aristas del grafo, es decir, transiciones

que no están conectadas por un lugar en el MG. Los elementos de \mathbf{L} que no aparecen en blanco se corresponden con los lugares del MG que tienen como transición de origen la indicada por su fila y como transición de salida la indicada por su columna. El valor numérico es el número de marcas del lugar en cuestión. A partir de \mathbf{L} , aplicando el algoritmo 3.13, se calculan los caminos de menor número de marcas entre cualesquiera dos transiciones del MG. La salida de este algoritmo se compone de dos matrices \mathbf{L} y \mathbf{B} de tamaño 8×8 en este caso, que es el número de transiciones de interfaz que hay en el MG.

$$\mathbf{L} = \begin{matrix} & I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \\ \begin{matrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{matrix} & \begin{bmatrix} 0 & 1 & 2 & \underline{2} & 1 & 2 & 3 & 3 \\ 1 & 0 & 1 & 2 & 2 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 3 & 2 & 1 & 2 \\ 3 & 2 & 1 & 0 & 3 & 3 & 2 & 1 \\ 1 & 2 & 3 & 3 & 0 & 1 & 2 & 3 \\ 2 & 1 & 2 & 3 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 2 & 2 & 1 & 0 & 1 \\ 3 & 3 & 2 & 1 & \underline{2} & 2 & 1 & 0 \end{bmatrix} \end{matrix} \quad \mathbf{B} = \begin{matrix} & I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \\ \begin{matrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & \underline{0} & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & \underline{0} & 1 & 0 & 1 \end{bmatrix} \end{matrix}$$

En el ejemplo anterior ya se ha explicado la interpretación de los datos de estas dos matrices. La matriz \mathbf{L} contiene el marcado inicial de los lugares implícitos necesarios para la reducción de las subredes producto del corte y la matriz \mathbf{B} indica cuántos lugares implícitos son necesarios (en principio tantos como elementos nulos). En este caso aparecen 22 elementos nulos en la matriz \mathbf{B} , pero 20 de ellos se corresponden exactamente con lugares ya presentes en el MG original, por lo que no es necesario duplicarlos (los 8 lugares del corte y los lugares internos H_{1i} y H_{2i} con $1 \leq i \leq 6$). Esta es la razón del cambio de notación en algunos lugares internos del MG original. A estos lugares se les ha puesto etiquetas H_{ij} en vez de P_{ij} porque en el proceso de reducción resultan ser necesarios para la reducción de las subredes y por lo tanto implícitos. De esta forma se preserva la notación de los nodos de los sistemas agregados sin duplicar lugares del MG inicial.

En consecuencia, la interpretación de las matrices \mathbf{L} y \mathbf{B} señala que sólo es necesario añadir dos lugares implícitos para construir el sistema extendido \mathcal{ES} . Estos lugares son el H_{17} y el H_{27} , ambos con marcado inicial 2 y que conectan I_{11} con I_{14} e I_{24} con I_{21} respectivamente. Estos lugares son los que indican las matrices \mathbf{L} y \mathbf{B} en sus elementos subrayados. Si se añaden a la red original estos lugares implícitos se obtiene el \mathcal{ES} de la figura 7.5.

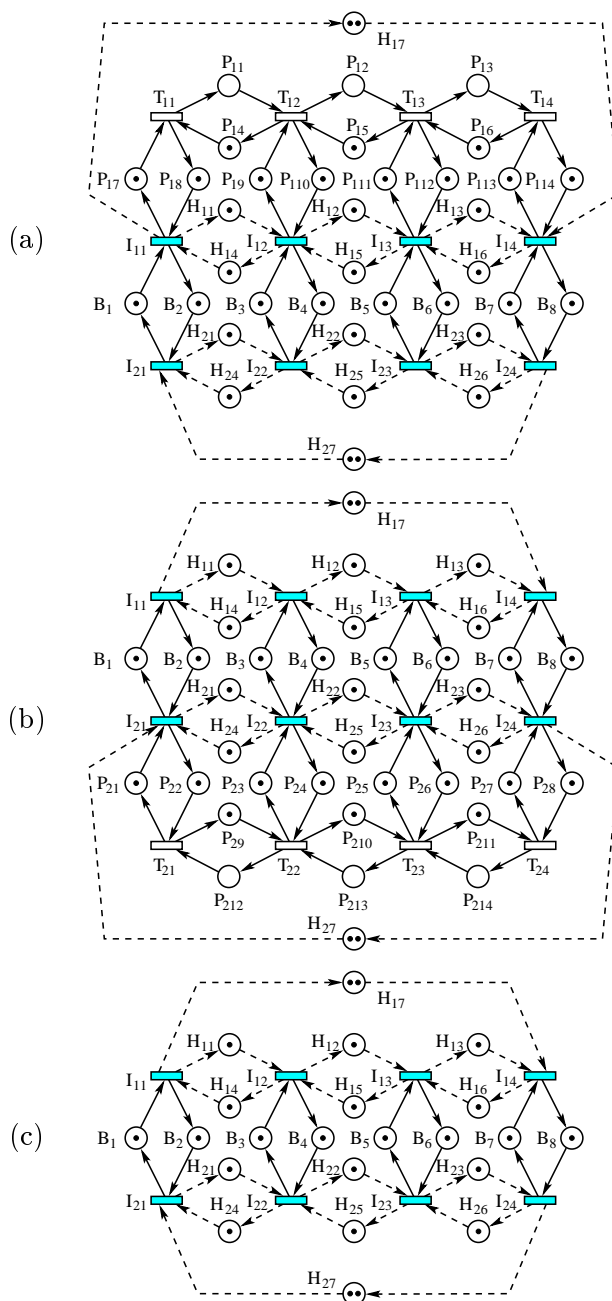


Figura 7.6: (a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del MG de la figura 7.4.

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.3284	0.4039	0.4708	0.5687	0.8184	1.1925	0.2954	0.7779	0.5114	0.3955	0.3355	1.2278
0.2962	0.3371	0.4004	0.5234	0.7973	1.2351	0.2943	0.7953	0.5203	0.3966	0.3341	1.2368
0.2942	0.3339	0.3965	0.5206	0.7959	1.2372	0.2942	0.7958	0.5205	0.3964	0.3338	1.2373
0.2941	0.3338	0.3964	0.5205	0.7959	1.2373	0.2941	0.7959	0.5205	0.3963	0.3338	1.2373
0.2941	0.3338	0.3963	0.5205	0.7959	1.2373	0.2941	0.7959	0.5205	0.3963	0.3338	1.2374
$\mathcal{X}(I_{11})$ exacto: 0.295945										Error: -0.61%	

Tabla 7.3: Aproximación de \mathcal{X} para Fig. 7.4. Tasas 1.0

En la tabla 7.3 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas 1. En este caso, el throughput exacto de las transiciones de la red original es 0.295945. La leyenda de la tabla es la misma que la del ejemplo anterior e incluye en cada iteración y para cada \mathcal{LS}_i el throughput de la transición I_{i1} obtenida en el estudio del \mathcal{LS}_i , las utilidades de las transiciones de interfaz de la subred \mathcal{N}_i y el factor de escala calculado en el \mathcal{BS} para que tenga el mismo throughput que \mathcal{LS}_i . En este caso se observa que se alcanza la convergencia en 5 iteraciones, obteniendo una aproximación de throughput de 0.294141 lo que supone un error final en la aproximación del -0.61% .

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.3340	0.4256	0.4987	0.6037	0.8746	1.1979	0.3331	0.4124	0.2290	0.1805	0.1701	1.1989
0.3330	0.4180	0.4960	0.6043	0.8773	1.2016	0.3329	0.4158	0.2296	0.1804	0.1700	1.1992
0.3329	0.4178	0.4959	0.6043	0.8773	1.2017	0.3329	0.4159	0.2296	0.1804	0.1700	1.1990
0.3329	0.4178	0.4959	0.6043	0.8773	1.2017	0.3329	0.4159	0.2296	0.1804	0.1700	1.1990
$\mathcal{X}(I_{11})$ exacto: 0.333356										Error: -0.13%	

Tabla 7.4: Aproximación de \mathcal{X} para Fig. 7.4. Tasas 1 en \mathcal{N}_1 y 2 en \mathcal{N}_2 .

En la tabla 7.4 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas variables entre las dos subredes producto del corte. En concreto se han puesto tasas 1 a todas las transiciones de la subred \mathcal{N}_1 y 2 a todas las transiciones de la subred \mathcal{N}_2 . En este caso, el throughput exacto de las transiciones de la red original es 0.333356. El algoritmo converge en 4 iteraciones, obteniendo una aproximación de throughput de 0.332936 lo que supone un error final en la aproximación del -0.13% .

En la tabla 7.5 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas diferentes simétricas en las dos subredes producto

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.3964	0.2228	0.2308	0.5302	0.7545	1.0999	0.3567	0.8124	0.5316	0.5569	0.4223	1.2961
0.3580	0.1883	0.1996	0.4790	0.7267	1.1528	0.3571	0.8186	0.5380	0.5617	0.4232	1.2998
0.3572	0.1878	0.1990	0.4777	0.7256	1.1535	0.3572	0.8188	0.5382	0.5619	0.4233	1.2999
0.3572	0.1878	0.1990	0.4776	0.7256	1.1535	0.3572	0.8188	0.5382	0.5619	0.4233	1.2999
$\mathcal{X}(I_{11})$ exacto: 0.362585						Error: -1.49%					

Tabla 7.5: Aproximación de \mathcal{X} para Fig. 7.4. Tasas 2 en T_{11} , T_{12} , T_{21} , T_{22} , I_{11} , I_{12} , I_{21} e I_{22} . Resto tasas 1.

del corte, pero diferentes entre sí. En concreto se han puesto tasas 2 a las transiciones T_{11} , T_{12} , T_{21} , T_{22} , I_{11} , I_{12} , I_{21} e I_{22} , dejando el resto de las transiciones con tasa 1. En este caso, el throughput exacto de las transiciones de la red original es 0.362585. El algoritmo converge en 4 iteraciones, obteniendo una aproximación de throughput de 0.357174 lo que supone un error final en la aproximación del -1.49% .

Todas las pruebas que se han realizado con distintos MG's han dado resultados similares en cuanto a velocidad de convergencia (4 ó 5 iteraciones) y a error final en la aproximación (inferior al 3%).

En la siguiente sección se van a dar unos ejemplos con otra clase de redes, los grafos marcados con pesos.

7.2 Grafos marcados con pesos

En esta sección se va a desarrollar en detalle el ejemplo de aproximación de throughput de las transiciones del WTS de [PJCS96b]. A un WTS se le pueden aplicar varias técnicas de descomposición y reducción de las desarrolladas en esta memoria. Obviamente se le puede aplicar la técnica del capítulo 4 que estaba desarrollada precisamente para WTS's y se basaba en los conceptos de resistencia y marcado ponderado (era una extensión de la técnica de MG's). Y también es posible aplicar las tres técnicas del capítulo 6 que estaban desarrolladas para poderse aplicar a cualquier red de Petri. Se aprovechará esta sección para comparar los resultados que se obtienen con las 4 técnicas de descomposición y reducción aplicadas a la misma red original, en este caso un WTS. No tiene sentido, en general, aplicarle a un WTS las técnicas de reducción de SM's desarrolladas para DSSP's (salvo que el WTS que se estudie tenga subredes con estructura de SM).

El ejemplo que se va a desarrollar en detalle es el de la figura 7.7. La red

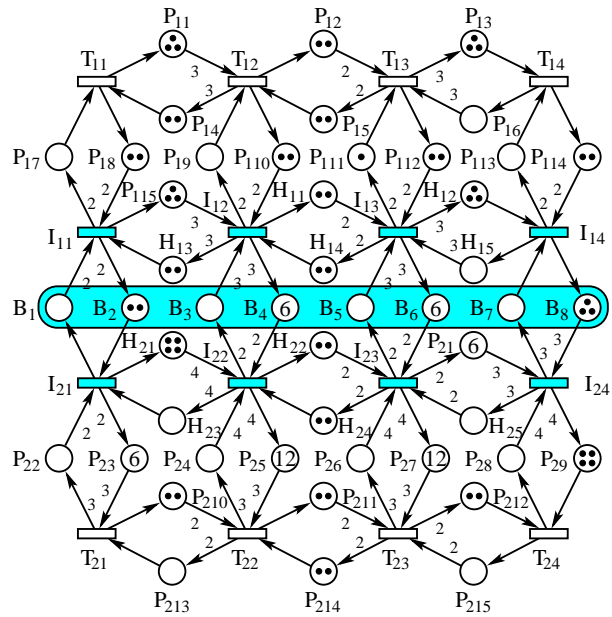


Figura 7.7: Ejemplo de WTS's.

es un WTS ya que todos sus lugares tienen una sola transición de entrada y una sola transición de salida (definición 2.97). Además es vivo y limitado, por lo que tiene distribución única en estado estacionario. Se ha elegido esta red por tener una estructura parecida a la del ejemplo 2 de MG's. La diferencia es que ahora hay pesos múltiples en los arcos y marcados diferentes en los lugares. La red tiene una estructura muy simétrica de forma que cualquier corte que se defina de lugar a subredes con la misma estructura y muy conectadas entre sí. De esta forma, el cálculo de las resistencias máximas y marcados ponderados mínimos entre transiciones de interfaz no es en absoluto evidente y es un buen test para todos los algoritmos de cálculo desarrollados en el capítulo 4. El grafo de alcanzabilidad de esta red tiene 90171 estados. Se ha elegido como corte el mismo que en el caso de MG's, es decir, el corte en formado por los lugares $\{B_i\}_{i=1}^8$. Este corte divide la red original en dos subredes (\mathcal{N}_1 y \mathcal{N}_2). La subred \mathcal{N}_1 (la de la parte superior) está generada por las transiciones internas $\{T_{1i}\}_{i=1}^4$ y las transiciones de interfaz $\{I_{1i}\}_{i=1}^4$. La subred \mathcal{N}_2 (la de la parte inferior) está generada por las transiciones internas $\{T_{2i}\}_{i=1}^4$ y las transiciones de interfaz $\{I_{2i}\}_{i=1}^4$. Se

ha cambiado de nuevo la notación de algunos lugares internos (etiquetas H_{ij} en vez de $P_{i,j}$) porque luego resultarán ser lugares implícitos necesarios para resumir el comportamiento de las subredes, es decir, se ha tomado una notación tal que mantenga las etiquetas de los lugares implícitos.

El siguiente paso consiste en calcular los sistemas agregados. En este caso, como el corte de la red original se ha hecho en dos subredes, habrá que generar dos sistemas de bajo nivel (\mathcal{LS}_1 y \mathcal{LS}_2) y un sistema de alto nivel o esqueleto básico (\mathcal{BS}). Cada \mathcal{LS}_i está formado por la subred \mathcal{N}_i y una agregación de la otra. El \mathcal{BS} está formado por las agregaciones de las dos subredes. Por lo tanto, es necesario calcular previamente las agregaciones de cada subred. Como se ha explicado en el capítulo 4, las agregaciones de las subredes se realiza por medio de transiciones inmediatas y lugares estructuralmente implícitos marcados de transición a transición (TT-MSIP's) junto con un marcado que los hace implícitos. Los lugares implícitos se utilizan para preservar los marcados ponderados mínimos entre transiciones de interfaz (extensión a WTS del concepto de marcado mínimo en MG's). Las transiciones inmediatas se utilizan para preservar la máxima resistencia entre transiciones de interfaz (concepto introducido para la reducción de WTS's y necesario para preservar el grado de sincronización interna de los caminos debido a los pesos de los arcos y así obtener mejores aproximaciones).

El cálculo de estos nodos se realiza por medio de los algoritmos 4.23, 4.28, 4.31 y 4.32. Estos algoritmos aseguran el cálculo un conjunto reducido de nodos (en el sentido de que se eliminan en el cálculo algunos nodos que se sabe que van a ser implícitos respecto a otros) necesarios para reducir una subred. En una implementación real es posible realizar todos los cálculos en un único algoritmo que, a partir de la estructura, marcado inicial y corte elegido en el WTS, genere el conjunto de nodos necesarios para la reducción. Pero, para que sea más comprensible el funcionamiento de los distintos algoritmos se ha decidido desarrollar cada uno por separado dando la interpretación de las estructuras de datos que utiliza.

El primer algoritmo de cálculo es el 4.23, que calcular, a partir de la estructura del WTS un vector \mathbf{g} de ganancias relativas entre dos transiciones cualesquiera del WTS. Con este vector es inmediato calcular la ganancia entre dos transiciones del WTS y es necesario para efectuar cálculos posteriores. Al igual que en MG's, todo WTS puede interpretarse como un grafo dirigido con un vértice por cada transición y una arista por cada lugar, ya que los lugares de todo WTS tienen una única transición de entrada y una sola transición de salida. Para el cálculo de ganancias no interviene el mar-

cado inicial de los lugares, sino únicamente lo pesos de los arcos del WTS. Y para el cálculo de las ganancias ni siquiera es necesario conocer los pesos, sino únicamente las ganancias de los caminos elementales (los formados por dos transiciones conectadas por un único lugar). Por lo tanto, para el cálculo de ganancias sólo hace falta una matriz \mathbf{L} de ganancias elementales (se sigue la notación del algoritmo 4.23) de tamaño $n \times n$ donde n es el número de vértices del grafo ($n = 16$ en este caso), es decir, el número de transiciones del WTS. En este caso, la matriz \mathbf{L} es:

$$\mathbf{L} = \begin{matrix} & \begin{matrix} T_{11} & T_{12} & T_{13} & T_{14} & T_{21} & T_{22} & T_{23} & T_{24} & I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \end{matrix} \\ \begin{matrix} T_{11} \\ T_{12} \\ T_{13} \\ T_{14} \\ T_{21} \\ T_{22} \\ T_{23} \\ T_{24} \\ I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{matrix} & \left[\begin{array}{cccccccccccccccc} & & \frac{1}{3} & & & & & & \frac{1}{2} & & & & & & & & \\ 3 & & & & & & & & \frac{1}{2} & & & & & & & & \\ & 2 & & \frac{1}{2} & & & & & \frac{1}{2} & & & & & & & & \\ & & & \frac{1}{3} & & & & & \frac{1}{2} & & & & & & & & \\ \hline & & & & & \frac{1}{2} & & & & & & & & \frac{3}{2} & & & \\ & & & & 2 & & \frac{1}{2} & & & & & & & \frac{3}{4} & & \\ & & & & 2 & & 2 & & 2 & & & & & & \frac{3}{4} & \\ & & & & & & \frac{1}{2} & & & & & & & & & \frac{1}{4} \\ \hline 2 & & & & & & & & \frac{1}{3} & & & & 2 & & & & \\ & 2 & & & & & & & 3 & & \frac{1}{2} & & & \frac{3}{2} & & & \\ & & 2 & & & & & & 2 & & 3 & & & & \frac{3}{2} & & \\ & & & 2 & & & & & & & \frac{1}{3} & & & & & \frac{1}{3} & \\ & & & & 2 & & & & \frac{1}{2} & & & & & \frac{1}{4} & & & \\ & & & & \frac{2}{3} & & & & \frac{2}{3} & & & & & 4 & & \frac{1}{2} & \\ & & & & & \frac{4}{3} & & & \frac{2}{3} & & & & & & 2 & & \\ & & & & & & \frac{4}{3} & & \frac{2}{3} & & & & & & & & \frac{2}{3} \\ & & & & & & & 4 & & & & & & & 3 & & \frac{3}{2} \end{array} \right] \end{matrix}$$

La interpretación de los datos de esta matriz es la siguiente. Los elementos que aparecen en blanco son las combinaciones de vértices que no son aristas del grafo, es decir, transiciones que no están conectadas por un lugar en el WTS. Por ejemplo, el elemento $\mathbf{L}(T_{11}, T_{13})$ aparece en blanco porque en el WTS original no hay ningún lugar que conecte la transición T_{11} con la T_{13} . En la implementación real de esta matriz, estos elementos tendrían valor 0 codificando la no existencia del lugar correspondiente (ningún camino puede tener ganancia 0 por la definición 4.1 de ganancia). Estos elementos se han dejado en blanco para hacer más legible la matriz.

Por lo que respecta a los elementos de \mathbf{L} que no aparecen en blanco, por ejemplo $\mathbf{L}(T_{11}, T_{12}) = 1/3$, su interpretación es que hay un lugar en el WTS

que conecta T_{11} con T_{13} (el lugar P_{11}) y que el camino elemental $T_{11}P_{11}T_{13}$ tiene ganancia $1/3$ (ya que $W(T_{11}, P_{11}) = 1$ y $W(P_{11}, T_{12}) = 3$).

En consecuencia, la matriz \mathbf{L} tiene tantos elementos distintos de 0 como lugares tiene el WTS. De esta manera, la matriz \mathbf{L} contiene toda la información sobre la ganancia de los caminos elementales del WTS. Notar que como el WTS de partida es vivo y limitado se cumple que $\mathbf{L}(i, j) \cdot \mathbf{L}(j, i) = 1$ (teorema 4.5) siempre que existan caminos elementales de i a j y de j a i . En este ejemplo, por la simetría del WTS, siempre que existe un camino elemental de i a j también existe otro de j a i . A partir de \mathbf{L} es posible calcular las ganancias entre cualesquiera dos transiciones del WTS visitando todos los vértices del grafo asociado al WTS.

Al aplicar el algoritmo 4.23 a la matriz \mathbf{L} , se produce como salida un vector \mathbf{g} de tamaño 16 (el número de transiciones del WTS) con las ganancias relativas entre transiciones, tomando como referencia la primera, en este caso $\mathbf{g}(T_{11}) = 1$.

$$\mathbf{g} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} & T_{21} & T_{22} & T_{23} & T_{24} & I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \\ 1 & \frac{1}{3} & \frac{1}{6} & \frac{1}{2} & \frac{2}{3} & \frac{1}{3} & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} & \frac{1}{6} & \frac{1}{12} & \frac{1}{4} & 1 & \frac{1}{4} & \frac{1}{8} & \frac{1}{12} \end{bmatrix}$$

La interpretación de este vector es la siguiente (ver algoritmo 4.23). Si se quiere calcular la ganancia entre dos transiciones t_i, t_j del WTS, entonces $G(t_i, t_j) = \mathbf{g}(t_j)/\mathbf{g}(t_i)$. Por ejemplo, la ganancia entre la transición I_{11} y la transición I_{14} será $G(I_{11}, I_{14}) = \mathbf{g}(I_{14})/\mathbf{g}(I_{11}) = \frac{1/4}{1/2} = \frac{1}{2}$. Efectivamente, cualquier camino que conecta la transición I_{11} con la I_{14} , por ejemplo el camino $\pi = I_{11}P_{15}I_{12}H_{11}I_{13}H_{12}I_{14}$, tiene ganancia $G(\pi) = \frac{1 \cdot 1 \cdot 3}{3 \cdot 2 \cdot 1} = \frac{1}{2}$.

Este vector \mathbf{g} de ganancias relativas entre transiciones es fundamental para cálculos posteriores ya que la ganancia entre transiciones interviene en el cálculo de resistencias y de marcados ponderados. Además, este vector es la forma más eficiente en espacio de almacenar todas las ganancias entre dos transiciones cualesquiera del WTS vivo y limitado.

El siguiente paso consiste en calcular los marcados ponderados mínimos entre transiciones de interfaz del WTS. Estos marcados ponderados mínimos permiten el cálculo de los lugares implícitos necesarios para la reducción. Este cálculo se realiza en por medio del algoritmo 4.28 que no es más que la extensión natural del que se ha utilizado para MG's pero introduciendo pesos en los arcos. En este algoritmo se ha empleado una técnica similar a la de MG's para reducir el conjunto de lugares calculado en el sentido de que ningún lugar calculado sea implícito respecto a otros del conjunto.

que el camino elemental $\pi = T_{12}P_{14}T_{11}$ tiene marcado ponderado $2/3$ (ya que $M(\pi, \mathbf{m}_0) = \mathbf{m}_0[P_{14}]/W(T_{12}, P_{14}) = 2/3$). En consecuencia, la matriz \mathbf{L} tiene inicialmente tantos elementos finitos como lugares tiene el WTS. De esta manera, la matriz \mathbf{L} contiene toda la información sobre la estructura y el marcado ponderado de los caminos elementales del WTS en el marcado inicial. A partir de ella y del vector \mathbf{g} es posible calcular, por medio del algoritmo 4.28, el marcado ponderado de los caminos de menor marcado ponderado entre cualesquiera dos transiciones del WTS.

Al aplicar el algoritmo 4.28 a la matriz \mathbf{L} y el vector \mathbf{g} , se producen como salida dos matrices \mathbf{L} y \mathbf{B} (\mathbf{L} puede ser la misma matriz en la implementación real, aunque la interpretación de sus datos es diferente a la que se ha hecho antes). Aunque estas matrices son de tamaño 16×16 , para efectuar la reducción de las subredes producto del corte del WTS sólo interesan los datos correspondientes a las transiciones de interfaz, es decir, son de tamaño 8×8 en este caso.

$$\mathbf{L} = \begin{matrix} & \begin{matrix} I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \end{matrix} \\ \begin{matrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{matrix} & \begin{bmatrix} 0 & 3 & 7 & \frac{19}{2} & 1 & 3 & 7 & 10 \\ \frac{2}{3} & 0 & 2 & \frac{8}{3} & 1 & \frac{5}{3} & 3 & 4 \\ \frac{4}{3} & 1 & 0 & 1 & \frac{3}{2} & \frac{11}{6} & 2 & 2 \\ \frac{5}{2} & 3 & 0 & 0 & 3 & 4 & 3 & 3 \\ 0 & 4 & 12 & 18 & 0 & 4 & 12 & 18 \\ 0 & 0 & 2 & 4 & 0 & 0 & 2 & \frac{9}{2} \\ 1 & 1 & 0 & \frac{3}{2} & 1 & 1 & 0 & 3 \\ \frac{2}{3} & \frac{2}{3} & 0 & 0 & \frac{2}{3} & \frac{2}{3} & 0 & 0 \end{bmatrix} \end{matrix} \quad \mathbf{B} = \begin{matrix} & \begin{matrix} I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \end{matrix} \\ \begin{matrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & \underline{0} & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & \underline{0} & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ \underline{0} & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & \underline{0} \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \end{matrix}$$

La interpretación de estas matrices es la siguiente. La matriz \mathbf{L} contiene el marcado ponderado de los caminos de marcado ponderado mínimo que conectan dos transiciones de interfaz cualesquiera. Por ejemplo, el elemento $\mathbf{L}(I_{11}, I_{14}) = 19/2$ indica que el camino de marcado ponderado mínimo en el WTS que conecta la transición I_{11} con la transición I_{14} tiene un marcado ponderado de $19/2$. En este caso, un camino de marcado ponderado mínimo que conecta la transición I_{11} con la transición I_{14} es el formado por los lugares $P_{17}P_{11}P_{12}P_{13}P_{114}$.

Por lo que respecta a la matriz \mathbf{B} , sus elementos son variables lógicas, es decir, sólo toman valores 0 y 1. Un elemento de valor 0, por ejemplo $\mathbf{B}(I_{11}, I_{14})$, indica que ningún camino de marcado ponderado mínimo en el WTS que conecta I_{11} con I_{14} pasa por otras transiciones de interfaz, es decir, en la reducción de las subredes hace falta poner un lugar que conecte estas

dos transiciones para resumir este conjunto de caminos que será eliminado en algunos sistemas agregados (ver definición 4.33). Un elemento de valor 1, por ejemplo $\mathbf{B}(I_{11}, I_{13})$, indica que algún camino de menor marcado ponderado en el WTS que conecta I_{11} con I_{13} pasa por otras transiciones de interfaz y por el teorema 4.21 no hace falta poner un lugar implícito que resuma este camino porque será implícito respecto a otros. En este caso, la matriz \mathbf{L} indica que el marcado ponderado mínimo entre I_{11} e I_{13} es 7. Un camino de marcado ponderado mínimo entre I_{11} e I_{13} es el $\pi = I_{11}B_2I_{21}H_{21}I_{22}H_{22}I_{23}B_5I_{13}$ que pasa por otras transiciones de interfaz, por lo que la reducción de π puede lograrse por medio de la reducción de los caminos de menor marcado ponderado de I_{11} a I_{21} , de I_{21} a I_{22} , de I_{22} a I_{23} y de I_{23} a I_{13} .

En consecuencia, hay que poner tantos lugares implícitos como elementos nulos tenga la matriz \mathbf{B} . En este punto hay que notar que algunos de los elementos nulos de \mathbf{B} se corresponden con los lugares del corte o lugares ya presentes en la red inicial. Obviamente no será necesario duplicar estos lugares. En este caso, de los 20 elementos nulos de \mathbf{B} , 16 se corresponden con lugares de la red inicial. Únicamente los elementos nulos de \mathbf{B} subrayados se corresponden con lugares que habría que añadir.

También es interesante en este caso notar que los elementos $\mathbf{B}(I_{12}, I_{22})$ y $\mathbf{B}(I_{13}, I_{23})$ aparecen marcados con 1, es decir, se pueden eliminar. Por otro lado estos elementos se corresponden con los canales B_4 y B_6 de la red inicial. Esto quiere decir que estos canales ya son lugares implícitos en la red inicial y que se podían haber eliminado desde el principio sin alterar el grafo de alcanzabilidad ni las prestaciones del sistema. Pese a ello se mantendrán en los sistemas agregados para no confundir al lector y generar los sistemas agregados preservando los lugares del corte.

Por lo tanto, la interpretación de las matrices \mathbf{L} y \mathbf{B} de este ejemplo señala que para reducir las subredes producto del corte son necesarios 4 lugares implícitos que conectan los pares de transiciones (I_{11}, I_{14}) , (I_{12}, I_{14}) , (I_{14}, I_{11}) y (I_{21}, I_{24}) . Posteriormente se verá que al reducir los caminos de máxima resistencia entre transiciones de interfaz sobra el lugar implícito que conecta el par de transiciones (I_{12}, I_{14}) ya que resulta ser implícito respecto a los nodos que resumen los caminos de máxima resistencia. Por lo tanto, sólo será necesario añadir 3 lugares implícitos. El teorema 4.19 permite calcular, a partir de la matriz \mathbf{L} y el vector \mathbf{g} , los pesos de los arcos de entrada y salida de estos lugares y el marcado inicial necesario para hacerlos implícitos (ver figura 7.8).

El siguiente paso consiste en calcular los caminos de máxima resistencia

entre transiciones de interfaz del WTS. Estos caminos de resistencia máxima permiten preservar en la reducción de las subredes el grado de sincronización interna de los caminos que se van a reducir debido a los pesos de los arcos y eventualmente las marcas que puedan estar atrapadas debido a las sincronizaciones en las transiciones. Los caminos de máxima resistencia se reducen por medio de un camino que une las transiciones de interfaz formado por una transición inmediata y dos lugares implícitos. En el capítulo 4 esta parte se ha desarrollado por medio de dos algoritmos, uno que calcula las resistencias máximas (el algoritmo 4.31) entre transiciones de interfaz y otro que calcula los nodos necesarios para la reducción de estos caminos (el algoritmo 4.32).

El algoritmo 4.31 toma como entrada el grafo dirigido asociado al WTS que se quiere descomponer junto con la información del marcado ponderado de los caminos elementales y el vector de ganancias relativas de las transiciones. Toda esta información puede recogerse entre el vector \mathbf{g} anterior y la información de la estructura del WTS en sus matrices de preincidencia y postincidencia.

Al aplicar el algoritmo 4.31 al WTS en estudio, se producen como salida tres matrices \mathbf{R} , \mathbf{K} y \mathbf{L} . La matriz \mathbf{R} contiene el valor de la máxima resistencia entre transiciones cualesquiera del WTS, la matriz \mathbf{K} contiene la información necesaria para conocer exactamente cuáles son los nodos por los que pasan los caminos de máxima resistencia y la matriz \mathbf{L} es exactamente la misma que la del algoritmo anterior aunque cambia su función a lo largo del algoritmo. En este caso los elementos de \mathbf{L} indican el marcado ponderado que puede circular libremente por los caminos de máxima resistencia entre transiciones cualesquiera del WTS. Como se ha explicado en el capítulo 4, el marcado ponderado no atrapado en un camino entre dos transiciones cualesquiera de un WTS coincide con el mínimo marcado ponderado de todos los caminos que unen las dos transiciones. Por ello la matriz \mathbf{L} contiene exactamente la misma información que la del algoritmo anterior. Este hecho hace posible la implementación conjunta en la práctica de los algoritmos 4.28 y 4.31. En esta memoria se ha preferido tratarlos por separado para que sean más fáciles de exponer y a la vez más legibles y comprensibles para el lector.

Se procederá ahora a explicar en detalle el contenido de las matrices \mathbf{R} y \mathbf{K} . En el WTS del ejemplo, la matriz \mathbf{R} es de tamaño 16×16 en principio, pero para efectuar la reducción de las subredes producto del corte sólo interesan los datos correspondientes a las transiciones de interfaz, es

decir, es de tamaño 8×8 en este caso:

$$\mathbf{R} = \begin{matrix} & I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \\ \begin{matrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{matrix} & \begin{bmatrix} 0 & 3 & 6 & \frac{11}{2} & \frac{1}{2} & 2 & 4 & 6 \\ \frac{1}{2} & 0 & 2 & 1 & \frac{1}{2} & \frac{2}{3} & \frac{4}{3} & 2 \\ \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{2} & \frac{1}{2} & 1 & 1 \\ \frac{3}{2} & 3 & 3 & 0 & \frac{3}{2} & 2 & 3 & 3 \\ 2 & 6 & 12 & 12 & 0 & 4 & 8 & 12 \\ \frac{1}{2} & \frac{3}{2} & 3 & \frac{5}{2} & \frac{1}{4} & 0 & 2 & 3 \\ \frac{1}{2} & 1 & \frac{3}{2} & \frac{3}{2} & \frac{1}{2} & \frac{1}{2} & 0 & \frac{3}{2} \\ \frac{2}{3} & \frac{2}{3} & 1 & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} & \frac{2}{3} & 0 \end{bmatrix} \end{matrix}$$

Los elementos de \mathbf{R} permiten ilustrar la diferencia entre los conceptos de resistencia y de resistencia estructural (ver la definición 4.11 de resistencia estructural). El elemento $\mathbf{R}(I_{21}, I_{11}) = 2$ indica que la máxima resistencia entre la transición I_{21} e I_{11} es 2. Si se observa la matriz \mathbf{K} posterior se puede ver que un camino de resistencia máxima entre I_{21} e I_{11} es el camino elemental $I_{21}B_1I_{11}$ que tiene resistencia estructural 2 y marcado ponderado 0, es decir, su resistencia y resistencia estructural coinciden (no puede tener marcas atrapadas que disminuyan la resistencia estructural).

Un ejemplo en el que la resistencia de un camino no es igual a su resistencia estructural es el caso del elemento $\mathbf{R}(I_{11}, I_{14}) = 11/2$. Si se observa la matriz \mathbf{K} posterior se puede ver que un camino de resistencia máxima entre I_{11} e I_{14} es el que pasa por las transiciones $\pi = I_{11}I_{21}T_{21}T_{22}T_{23}T_{24}I_{24}I_{14}$. La resistencia estructural de π es 6 y su marcado ponderado 10. Ahora, el marcado ponderado mínimo entre I_{11} e I_{14} es $\mathbf{L}(I_{11}, I_{14}) = 19/2$ (un camino de marcado ponderado mínimo entre I_{11} e I_{14} es $I_{11}T_{11}T_{12}T_{13}T_{14}I_{14}$). Como π tiene mayor marcado ponderado que algún otro camino entre I_{11} e I_{14} , por el teorema 4.6, tiene parte de su marcado ponderado atrapado que disminuye la resistencia del camino respecto a su resistencia estructural. En este caso, π tiene $10 - 19/2 = 1/2$ de su marcado ponderado atrapado, que disminuye su resistencia respecto a su resistencia estructural hasta $6 - 1/2 = 11/2 = \mathbf{R}(I_{11}, I_{14})$.

Por lo que respecta a la matriz \mathbf{K} , contiene la información necesaria para saber exactamente cuáles son los caminos de máxima resistencia entre dos transiciones cualesquiera del WTS. Esta información es necesaria para que el algoritmo 4.32 pueda calcular los nodos necesarios para reducir los caminos de máxima resistencia entre transiciones de interfaz del WTS. Aunque sólo se necesita reducir los caminos de máxima resistencia entre transiciones de

interfaz, estos caminos pueden pasar por cualquier transición del WTS, por lo que es necesario tener en cuenta toda la matriz. En el ejemplo, la matriz \mathbf{K} es de tamaño 16×16 .

$$\mathbf{K} = \begin{array}{c} T_{11} \\ T_{12} \\ T_{13} \\ T_{14} \\ T_{21} \\ T_{22} \\ T_{23} \\ T_{24} \\ I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{array} \begin{array}{c} T_{11} \\ T_{12} \\ T_{13} \\ T_{14} \\ T_{21} \\ T_{22} \\ T_{23} \\ T_{24} \\ I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{array} \begin{bmatrix} 0 & 0 & 2 & 3 & 13 & 13 & 13 & 13 & 0 & 9 & 15 & 16 & 9 & 13 & 13 & 13 \\ 0 & 0 & 0 & 3 & 13 & 13 & 13 & 13 & 1 & 0 & 12 & 4 & 9 & 13 & 13 & 13 \\ 2 & 0 & 0 & 0 & 13 & 13 & 13 & 16 & 2 & 2 & 12 & 4 & 9 & 13 & 16 & 12 \\ 3 & 3 & 0 & 0 & 13 & 13 & 16 & 16 & 3 & 12 & 12 & 0 & 9 & 13 & 16 & 12 \\ \hline 13 & 14 & 13 & 16 & 0 & 0 & 6 & 7 & 13 & 14 & 15 & 16 & 14 & 6 & 7 & 8 \\ 14 & 14 & 14 & 16 & 0 & 0 & 0 & 7 & 14 & 14 & 15 & 16 & 14 & 0 & 7 & 8 \\ 15 & 14 & 15 & 16 & 15 & 0 & 0 & 0 & 15 & 14 & 15 & 16 & 15 & 15 & 0 & 8 \\ 16 & 16 & 16 & 16 & 16 & 7 & 0 & 0 & 16 & 16 & 15 & 16 & 16 & 16 & 16 & 0 \\ \hline 0 & 10 & 2 & 16 & 13 & 13 & 13 & 13 & 0 & 14 & 15 & 16 & 0 & 13 & 13 & 13 \\ 2 & 0 & 2 & 3 & 13 & 13 & 13 & 13 & 2 & 0 & 0 & 4 & 9 & 13 & 13 & 13 \\ 10 & 10 & 0 & 3 & 13 & 13 & 13 & 16 & 10 & 0 & 0 & 0 & 9 & 13 & 16 & 12 \\ 11 & 11 & 11 & 0 & 13 & 13 & 16 & 16 & 11 & 11 & 0 & 0 & 9 & 16 & 16 & 0 \\ 9 & 14 & 9 & 16 & 0 & 5 & 6 & 7 & 0 & 14 & 15 & 16 & 0 & 0 & 14 & 8 \\ 10 & 10 & 10 & 10 & 13 & 13 & 13 & 13 & 10 & 0 & 15 & 16 & 0 & 0 & 0 & 13 \\ 11 & 11 & 11 & 11 & 14 & 14 & 14 & 14 & 11 & 11 & 0 & 11 & 14 & 0 & 0 & 12 \\ 15 & 12 & 12 & 12 & 15 & 15 & 8 & 0 & 15 & 12 & 12 & 0 & 15 & 15 & 0 & 0 \end{bmatrix}$$

La interpretación de los elementos de la matriz \mathbf{K} es la misma que se puede observar en el algoritmo clásico de Floyd (ver [AHU83]) para el cálculo del camino de coste mínimo entre cualesquiera dos vértices de un grafo dirigido con pesos. Para conocer exactamente cuáles son los caminos de coste mínimo que ha calculado el algoritmo es suficiente una matriz cuadrada de tamaño el número de vértices del grafo en estudio (16 en este caso) cuyos elementos son 0 o el índice de un determinado vértice. La numeración de los vértices en el WTS es el número de fila o columna que tiene la etiqueta del vértice en cuestión. Un elemento 0, por ejemplo $\mathbf{K}(I_{21}, I_{11})$ indica que el camino de máxima resistencia de I_{21} a I_{11} es elemental, es decir, que hay un lugar que conecta I_{21} con I_{11} (B_1) y que el camino $I_{21}B_1I_{11}$ es el camino de máxima resistencia de I_{21} a I_{11} que ha calculado el algoritmo (puede haber varios caminos de máxima resistencia). Un elemento distinto de 0, por ejemplo $\mathbf{K}(I_{11}, I_{13}) = 15$, indica que el camino de máxima resistencia de I_{11} a I_{13} no es elemental y que pasa por el vértice número 15, es decir, por I_{23} . En este caso, el camino de máxima resistencia de I_{11} a I_{13} se descompone

en dos subcaminos; el camino de máxima resistencia de I_{11} a I_{23} y el de I_{23} a I_{13} . Recursivamente se van recorriendo todos los nodos de un camino de máxima resistencia de I_{11} a I_{13} y se obtiene el camino $I_{11}I_{21}I_{22}I_{23}I_{13}$.

El último cálculo que hay que realizar es el de los nodos necesarios para reducir los caminos de máxima resistencia. Esta tarea la realiza el algoritmo 4.32 que toma como entrada la matriz \mathbf{K} de caminos de máxima resistencia y el vector \mathbf{g} de ganancias relativas entre transiciones. El algoritmo recorre todos los nodos de todos los caminos de máxima resistencia entre transiciones de interfaz y calcula los nodos necesarios para su reducción. Se ha intentado reducir al máximo el conjunto de nodos resultante, y por ello, si el algoritmo detecta que un camino de máxima resistencia pasa por otra transición de interfaz, o si la reducción de un determinado camino puede efectuarse con un único lugar implícito, el algoritmo elimina estos nodos ya que otros nodos previamente calculados o por calcular resumen el camino en cuestión.

La salida del algoritmo 4.32 produce tres matrices \mathbf{E} , \mathbf{D}_1 y \mathbf{D}_2 de tamaño 16×16 en principio, pero para efectuar la reducción de las subredes producto del corte sólo interesan los datos correspondientes a las transiciones de interfaz, es decir, son de tamaño 8×8 en este caso.

$$\mathbf{E} = \begin{matrix} & I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \\ \begin{matrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

Los elementos de la matriz \mathbf{E} pueden ser 0 o el índice de una transición del WTS. Un elemento no nulo de \mathbf{E} , por ejemplo $\mathbf{E}(I_{12}, I_{11}) = 2$, es el índice de la transición del WTS (siguiendo la numeración de la matriz \mathbf{K}) en la que el camino de máxima resistencia alcanza su resistencia estructural. En este caso, la transición de índice 2 es T_{12} , indicando que el camino de máxima resistencia de I_{12} a I_{11} , que es el camino $I_{12}T_{12}T_{11}I_{11}$, alcanza su resistencia estructural en la transición T_{12} . Para reducir este camino es necesario preservar la transición T_{12} como inmediata. Los elementos no nulos de \mathbf{E} se corresponderán siempre con transiciones internas de la misma

subred a la que pertenezcan las transiciones de interfaz en estudio.

Los elementos nulos de \mathbf{E} indican que no es necesario poner nodos para la reducción de los correspondientes caminos de máxima resistencia. Esta situación se puede deber a dos causas. La primera es que el camino en cuestión pase por otra transición de interfaz, de forma que su reducción se puede obtener por medio de la reducción de subcaminos suyos (de esta forma se reduce el número de nodos necesarios para la reducción de las subredes). Éste es el caso de $\mathbf{E}(I_{11}, I_{12}) = 0$, ya que un camino de máxima resistencia de I_{11} a I_{12} es el $I_{11}I_{21}I_{22}I_{12}$, que pasa por otras transiciones de interfaz. Este camino se reduce por medio de las reducciones de sus subcaminos $I_{11}I_{21}$, $I_{21}I_{22}$ e $I_{22}I_{12}$. La segunda situación en la que se obtienen elemento nulos de \mathbf{E} ocurre cuando un camino alcanza su máxima resistencia estructural en la última transición. En ese caso debe ser también un camino de marcado ponderado mínimo y por lo tanto ya se ha calculado su resumen anteriormente. Es el caso de $\mathbf{E}(I_{21}, I_{24}) = 0$, ya que un camino de máxima resistencia de I_{21} a I_{24} es el $I_{21}T_{21}T_{22}T_{23}T_{24}I_{24}$, que alcanza su máxima resistencia estructural en I_{24} . El lugar que resume este camino se ha calculado en el algoritmo 4.28 entre la transición I_{21} e I_{24} (el lugar H_{26} de la figura 7.8).

Resumiendo, la matriz \mathbf{E} del ejemplo indica que sólo es necesario poner nodos para la reducción de dos caminos de máxima resistencia. El resto de caminos se puede reducir con estos o con los lugares implícitos calculados con el algoritmo 4.28. Serán necesarias dos transiciones inmediatas, que se corresponden con las transiciones T_{12} y T_{13} de la red original.

$$\mathbf{D}_1 = \begin{matrix} & I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \\ \begin{matrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{matrix} & \begin{bmatrix} - & - & - & - & - & - & - & - \\ 0 & - & - & 1 & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{bmatrix} \end{matrix} \quad \mathbf{D}_2 = \begin{matrix} & I_{11} & I_{12} & I_{13} & I_{14} & I_{21} & I_{22} & I_{23} & I_{24} \\ \begin{matrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{21} \\ I_{22} \\ I_{23} \\ I_{24} \end{matrix} & \begin{bmatrix} - & - & - & - & - & - & - & - \\ \frac{4}{3} & - & - & \frac{5}{3} & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{bmatrix} \end{matrix}$$

Las matrices \mathbf{D}_1 y \mathbf{D}_2 dan la información necesaria para conocer el marcado inicial de los lugares que resumen los caminos de máxima resistencia. Por lo tanto, sólo tienen sentido los elementos de \mathbf{D}_1 y \mathbf{D}_2 para los que el correspondiente elemento de \mathbf{E} es no nulo. En el ejemplo, sólo tienen sentido

los elementos de coordenadas (I_{12}, I_{11}) y (I_{12}, I_{14}) . El resto se han marcado con guiones indicando que no es útil su información. Como el resumen de un camino de máxima resistencia se compone de dos lugares y una transición inmediata, este camino reducido se puede dividir en dos subcaminos elementales. Los elementos de \mathbf{D}_1 indican el marcado ponderado del primer subcamino y los de \mathbf{D}_2 el marcado ponderado del segundo subcamino. Con ayuda del vector \mathbf{g} de ganancias relativas entre transiciones, aplicando el teorema 4.19 es inmediato calcular los pesos de los arcos que conectan los distintos nodos de un camino reducido. Una vez conocidos estos pesos, por medio de los marcados ponderados de \mathbf{D}_1 y \mathbf{D}_2 se obtienen los marcados iniciales de los lugares del camino reducido.

Una vez desarrollados con el ejemplo en cuestión los cuatro algoritmos de cálculo de los nodos necesarios para reducir las subredes, es el momento de reunir toda la información y saber cuántos nodos son necesarios para reducir las subredes:

Según la matriz \mathbf{B} son necesarios 20 lugares para reducir los caminos de mínimo marcado ponderado. Estos se pueden dividir en:

- 6 canales B_1, B_2, B_3, B_5, B_7 y B_8 . Los canales B_4 y B_6 son implícitos en la red original según la matriz \mathbf{B} , pero se pondrán por coherencia con los desarrollos teóricos.
- 10 lugares que ya están presentes en la red original y que no es necesario duplicar; H_{ij} con $i = 1, 2$ y $1 \leq j \leq 5$.
- 4 lugares nuevos que tendrán las siguientes etiquetas (ver figura 7.8).
 - H_{16} de I_{11} a I_{14} . Como $G(I_{11}, I_{14}) = \frac{\mathbf{g}(I_{14})}{\mathbf{g}(I_{11})} = \frac{1}{2}$, por el teorema 4.19, $W(I_{11}, H_{16}) = 1$ y $W(H_{16}, I_{14}) = 2$. Además, el marcado inicial es $\mathbf{m}_0[H_{16}] = \lfloor W(I_{11}, H_{16}) \cdot \mathbf{L}(I_{11}, I_{14}) \rfloor = \lfloor 1 \cdot 19/2 \rfloor = 9$.
 - H_{17} de I_{14} a I_{11} . $W(I_{14}, H_{17}) = 2$, $W(H_{17}, I_{11}) = 1$, $\mathbf{m}_0[H_{17}] = 5$.
 - H'_1 de I_{12} a I_{14} . $W(I_{12}, H'_1) = 3$, $W(H'_1, I_{14}) = 2$, $\mathbf{m}_0[H'_1] = 8$.
 - H_{26} de I_{21} a I_{24} . $W(I_{21}, H_{26}) = 1$, $W(H_{26}, I_{24}) = 12$, $\mathbf{m}_0[H_{26}] = 18$.

Por lo que respecta a los nodos que resumen los caminos de máxima resistencia, según la matriz \mathbf{E} hay 2 caminos que es necesario reducir, lo que suponen 2 transiciones inmediatas y 4 lugares nuevos:

- Camino de I_{12} a I_{11} . Transición inmediata nueva t_{12} (réplica inmediata de T_{12}), lugares nuevos H_{18}, H_{19} . Por el teorema 4.19, como $G(I_{12}, T_{12}) = \frac{\mathbf{g}(T_{12})}{\mathbf{g}(I_{12})} = 2$ entonces $W(I_{12}, H_{18}) = 2$ y $W(H_{18}, t_{12}) = 1$, y por ser $\mathbf{D}_1(I_{12}, I_{11}) = 0$ entonces $\mathbf{m}_0[H_{18}] = \lfloor W(I_{12}, H_{18}) \cdot 0 \rfloor = 0$. Como $G(T_{12}, I_{11}) = \frac{\mathbf{g}(I_{11})}{\mathbf{g}(T_{12})} = \frac{3}{2}$, $W(t_{12}, H_{19}) = 3$ y $W(H_{19}, I_{11}) = 2$, y por ser $\mathbf{D}_2(I_{12}, I_{11}) = \frac{4}{3}$, $\mathbf{m}_0[H_{19}] = \lfloor W(t_{12}, H_{19}) \cdot \frac{4}{3} \rfloor = \lfloor 3 \cdot \frac{4}{3} \rfloor = 4$.
- Camino de I_{12} a I_{14} . Transición inmediata nueva t_{13} (réplica inmediata de T_{13}), lugares nuevos H_{110}, H_{111} . $W(I_{12}, H_{110}) = 1$, $W(H_{110}, t_{13}) = 1$ y $\mathbf{m}_0[H_{110}] = 1$; $W(t_{13}, H_{111}) = 3$, $W(H_{111}, I_{14}) = 2$ y $\mathbf{m}_0[H_{111}] = 5$.

En este punto hay que notar que también es posible que un camino reducido de máxima resistencia entre dos transiciones de interfaz sea a la vez de mínimo marcado ponderado. Esto ocurre en el caso del camino de I_{12} a I_{14} . El mínimo marcado ponderado entre estas dos transiciones es $\mathbf{L}(I_{12}, I_{14}) = \frac{8}{3}$ y el camino reducido de máxima resistencia tiene marcado ponderado $\frac{\mathbf{m}_0[H_{111}]}{3} + \mathbf{m}_0[H_{110}] = \frac{5}{3} + 1 = \frac{8}{3}$. En los desarrollos teóricos no se ha tenido en cuenta esta posibilidad por no extender demasiado el capítulo 4, pero es evidente que en estos casos se puede eliminar el lugar implícito que resume el camino de menor marcado ponderado, ya que esta información está recogida también en el camino de máxima resistencia. Por ello en el ejemplo actual, se puede eliminar el lugar H'_1 anterior (de ahí la notación temporal elegida).

Si se añaden a la red original los nodos de la lista anterior se obtiene el sistema extendido \mathcal{ES} de la figura 7.8.

A partir del \mathcal{ES} es inmediato construir los sistemas agregados. En las figuras 7.9.a, 7.9.b y 7.9.c se pueden observar el \mathcal{LS}_1 , \mathcal{LS}_2 y \mathcal{BS} correspondientes al corte realizado en la red original. El \mathcal{LS}_1 tiene 10751 estados alcanzables, el \mathcal{LS}_2 9256 y el \mathcal{BS} 989.

La diferencia entre los tamaños de los grafos de alcanzabilidad de los sistemas agregados en este experimento y los encontrados en el mismo experimento de [PJCS96b] se deben a una errata en este último cometida al calcular los marcados iniciales de algunos lugares implícitos. Ahora se mejorarán las aproximaciones finales de throughput.

Para aproximar el throughput de las transiciones del WTS original basta aplicar el algoritmo iterativo 4.38 a los sistemas agregados. Este algoritmo es básicamente el mismo que el de MG's. Se han realizado tres experimentos; el primero poniendo las tasas de todas las transiciones a 1 (semántica de un

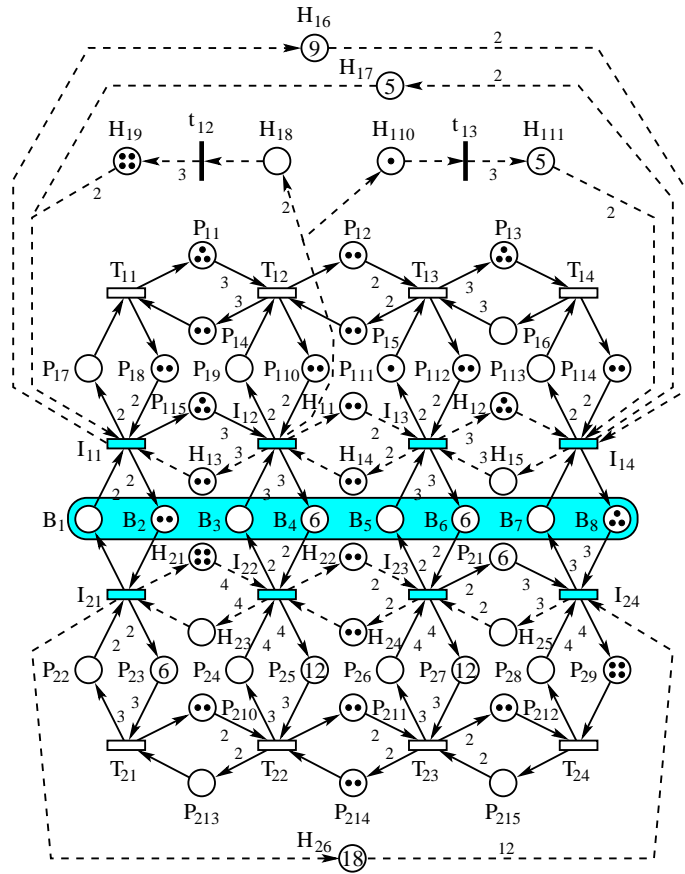


Figura 7.8: \mathcal{ES} del WTS de la figura 7.7 (técnica de WTS's).

solo servidor), el segundo poniendo tasa 10 a las transiciones de la primera subred (la de la parte superior) y tasa 0.1 a las de la segunda subred (la de la parte inferior) y el tercero poniendo tasas arbitrarias a todas las transiciones (las mismas que en [PJCS96b]).

En el caso de MG's todas las transiciones tienen el mismo throughput pero en WTS's no es así. Sin embargo, por el teorema 4.9, el cociente entre el throughput de dos transiciones cualesquiera de un WTS es igual al cociente entre las ganancias relativas de las dos transiciones y en la técnica de reducción se preservan estas ganancias entre transiciones, por lo que es suficiente con fijarse en el throughput de una transición cualquiera. El error

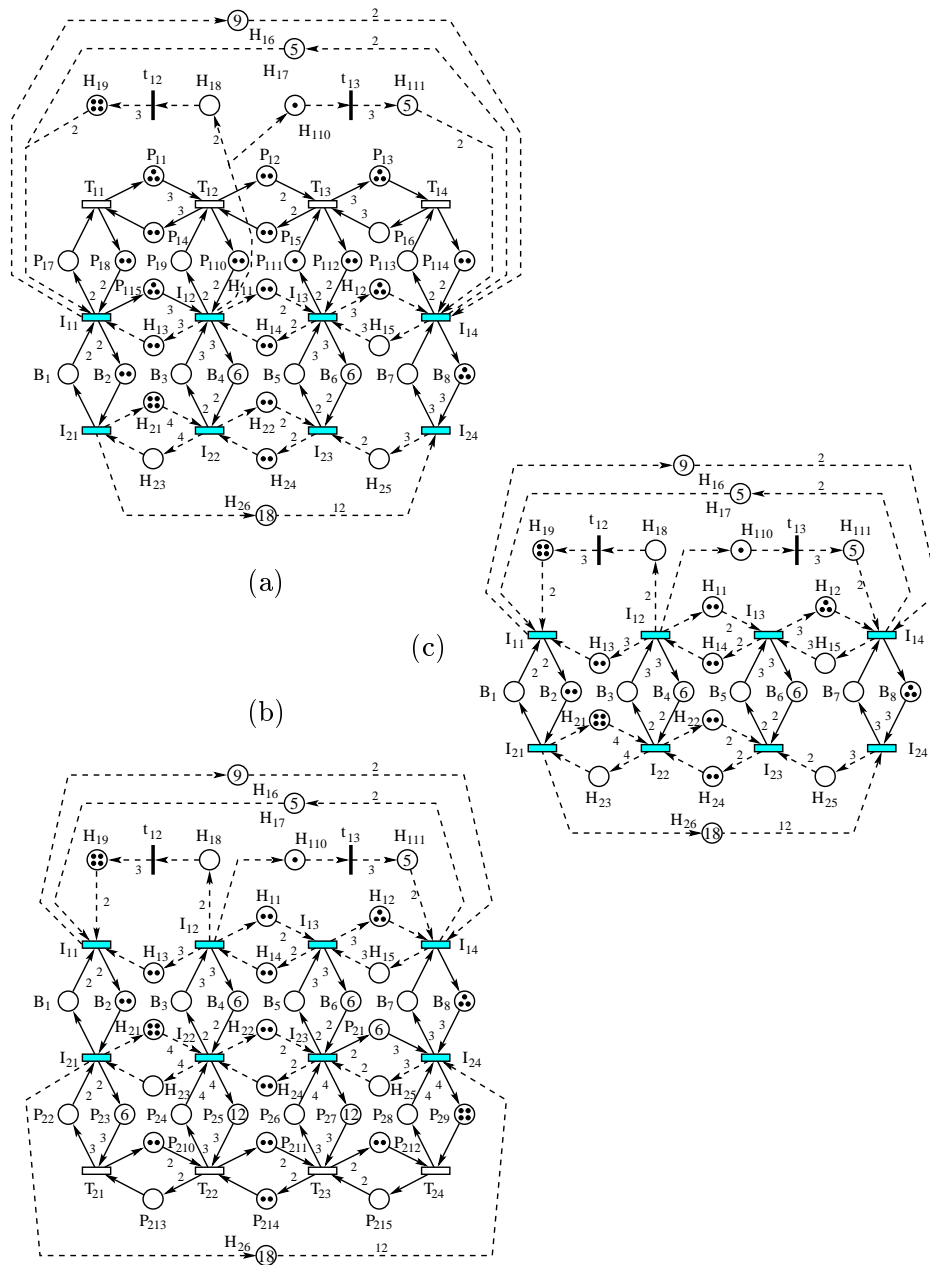


Figura 7.9: (a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del WTS de la figura 7.7 (técnica de WTS's).

en tanto por ciento cometido en la aproximación del throughput de cualquier otra transición será el mismo.

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.2448	0.4630	0.0826	0.0408	0.5676	1.0145	0.4274	0.4575	0.2525	0.2140	0.1958	1.0523
0.2187	0.4078	0.0735	0.0365	0.4187	1.0323	0.4324	0.4636	0.2559	0.2216	0.2150	1.0423
0.2168	0.4028	0.0728	0.0361	0.4086	1.0354	0.4333	0.4647	0.2565	0.2220	0.2165	1.0418
0.2167	0.4025	0.0728	0.0361	0.4081	1.0355	0.4334	0.4647	0.2565	0.2220	0.2166	1.0417
0.2167	0.4024	0.0728	0.0361	0.4081	1.0355	0.4334	0.4647	0.2565	0.2220	0.2166	1.0417
$\mathcal{X}(I_{11})$ exacto: 0.217662										Error: -0.45%	

Tabla 7.6: Aproximación de \mathcal{X} para Fig. 7.7. Tasas 1.0 (técnica WTS's).

En la tabla 7.6 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas 1. En este caso, el throughput exacto de la transición I_{11} de la red original es 0.217662. La tabla está dividida en dos partes encabezadas con \mathcal{LS}_1 y \mathcal{LS}_2 en las que se recogen los resultados del estudio de cada sistema de bajo nivel y la correspondiente visita al \mathcal{BS} . En cada \mathcal{LS}_i se calcula el throughput de una transición de referencia del mismo (normalmente I_{i1}) y las utilizaciones de las transiciones de interfaz de la subred \mathcal{N}_i presente en el \mathcal{LS}_i teniendo en cuenta sólo nodos de \mathcal{BS} (columnas encabezadas con μ_{ij}). Esta información se utiliza en el \mathcal{BS} para encontrar, por medio de un factor de escala λ , nuevas aproximaciones de las tasas de disparo de las transiciones de interfaz de \mathcal{LS}_i . En estas condiciones, la nueva tasa de la transición de interfaz I_{ij} será $\lambda \mathcal{X}(I_{ij})/\mu_{ij}$. Estas nuevas tasas no se incluyen en la tabla para no aumentar su tamaño. En cada fila de la tabla aparecen los resultados de cada iteración del algoritmo, hasta alcanzar la convergencia. En la última fila de la tabla aparece a la izquierda el throughput exacto de la transición de referencia en la red original y a la derecha el error de la aproximación. En este caso se observa que se alcanza la convergencia en 5 iteraciones, obteniendo una aproximación de throughput de 0.216682 por lo que el error en la aproximación final es del -0.45% .

El siguiente experimento se ha realizado poniendo tasas de diferentes órdenes de magnitud entre las transiciones de una subred y las de la otra. En este caso se ha puesto tasa 10 a las 8 transiciones de la parte superior y 0.1 a las 8 transiciones de la parte inferior. Como se vio en MG's, en estas condiciones el algoritmo converge en menos iteraciones y el error de la aproximación es despreciable. En la tabla 7.7 aparecen los resultados de la ejecución del algoritmo iterativo en este experimento. En este caso, el

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.0399	0.0040	0.0013	0.0007	0.2429	1.0118	0.0554	0.6568	0.3415	0.2232	0.2990	1.0409
0.0277	0.0028	0.0009	0.0005	0.1095	1.5438	0.0554	0.6571	0.3416	0.2233	0.3002	1.0396
0.0277	0.0028	0.0009	0.0005	0.1093	1.5438	0.0554	0.6571	0.3416	0.2233	0.3002	1.0396
0.0277	0.0028	0.0009	0.0005	0.1093	1.5746	0.0554	0.6572	0.3416	0.2233	0.3002	1.0396
$\mathcal{X}(I_{11})$ exacto: 0.027686						Error: +0.02%					

Tabla 7.7: Aproximación de \mathcal{X} para Fig. 7.7. Tasas 10 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 (técnica de WTS's).

throughput exacto de la transición I_{11} de la red original es 0.027686. La leyenda de la tabla es la misma que la de la anterior. El algoritmo converge en 4 iteraciones, obteniendo una aproximación de throughput de 0.027692 por lo que el error en la aproximación final es del +0.02%.

El ultimo experimento se ha realizado poniendo a todas las transiciones tasas elegidas al azar. En concreto las tasas de las transiciones siguiendo el orden de izquierda a derecha primero y de arriba a abajo después son: 2, 3, 2, 3, 6, 3, 2, 4, 10, 8, 2, 4, 4, 6, 8 y 5. En la tabla 7.8 aparecen los resultados de la ejecución del algoritmo iterativo en este experimento. En este caso, el throughput exacto de la transición I_{11} de la red original es 0.778787. La leyenda de la tabla es la misma que la de las anteriores. El algoritmo converge en 5 iteraciones, obteniendo una aproximación de throughput de 0.777055 por lo que el error en la aproximación final es del -0.22%.

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.8091	0.8276	0.0949	0.0674	0.6525	1.0388	1.5125	0.1741	0.1690	0.2379	0.1205	1.1279
0.7803	0.7939	0.0897	0.0650	0.5583	1.0523	1.5461	0.1785	0.1735	0.2516	0.1355	1.1067
0.7775	0.7881	0.0893	0.0648	0.5459	1.0541	1.5531	0.1795	0.1744	0.2534	0.1377	1.1042
0.7771	0.7873	0.0893	0.0648	0.5445	1.0544	1.5540	0.1796	0.1745	0.2536	0.1379	1.1040
0.7771	0.7872	0.0893	0.0648	0.5443	1.0544	1.5541	0.1796	0.1745	0.2536	0.1379	1.1039
$\mathcal{X}(I_{11})$ exacto: 0.778787						Error: -0.22%					

Tabla 7.8: Aproximación de \mathcal{X} para Fig. 7.7. Tasas variables (técnica de WTS's).

Hasta ahora, en esta sección se han desarrollado tres experimentos de aproximación de throughput de las transiciones de un WTS empleando la técnica de descomposición y reducción desarrollada en el capítulo 4 y que opera a nivel de red. Como en el capítulo 6 se han desarrollado tres técnicas

de descomposición y reducción aplicables a cualquier red de Petri, se van a realizar ahora los mismos experimentos con estas tres técnicas para poder comparar los resultados.

La red de partida es la misma de la figura 7.7 así como el corte elegido. La diferencia de la técnica anterior con las que se van a aplicar ahora reside en los sistemas agregados generados en la descomposición y reducción. Para aplicar la primera técnica general es necesario calcular un conjunto de lugares implícitos de acuerdo con la técnica desarrollada en [CDS99] (ver sección 6.2.2). Estos lugares implícitos resumen el comportamiento interno de las subredes producto del corte. Sin entrar en detalles sobre su cálculo, el algoritmo 6.6 da como resultado que son necesarios 24 lugares estructuralmente implícitos, que conectan cada transición de interfaz con el resto de las transiciones de interfaz de la misma subred. Por medio del algoritmo 2.60 se calculan los marcados iniciales de cada lugar para hacerlo implícito. Posteriormente se ha reducido la cantidad de lugares implícitos eliminando los que resultan ser implícitos respecto a otros (cálculo realizado para este ejemplo en concreto y no con una técnica general). El resultado es que para reducir la primera subred (la de la parte superior) son necesarios 7 lugares implícitos y 6 para la segunda subred (la de la parte inferior). Si se añaden a la red original los nodos de la lista anterior se obtiene el sistema extendido \mathcal{ES} de la figura 7.10.

A partir del \mathcal{ES} es inmediato construir los sistemas agregados. En las figuras 7.9.a, 7.9.b y 7.9.c se pueden observar el \mathcal{LS}_1 , \mathcal{LS}_2 y \mathcal{BS} correspondientes al corte realizado en la red original. Los tamaños del espacio de estados de cada sistema agregado dependen de la técnica general empleada. Si se aplica la primera técnica general (sección 6.3), que calcula en cada sistema agregado la componente fuertemente conexa del grafo de alcanzabilidad que contiene al marcado inicial, el \mathcal{LS}_1 tiene 10751 estados alcanzables, el \mathcal{LS}_2 9256 y el \mathcal{BS} 989. En este ejemplo estos grafos de alcanzabilidad son los mismos que los que se obtenían con la técnica de WTS's (en general puede no ser así). Pese a esta coincidencia, los resultados numéricos pueden no coincidir, ya que al utilizar distintos conjuntos de nodos en la reducción de las subredes, en los \mathcal{LS}_i hay diferencias en la estimación de las utilizaciones de las transiciones de interfaz, lo que motiva cambios al aplicar el algoritmo numérico de aproximación de throughput.

Si se aplica la segunda técnica general (sección 6.5), que consigue proyecciones exactas del sistema original sobre los nodos preservados en cada sistema agregado (algoritmos 6.33 y 6.39), entonces se obtienen sistemas

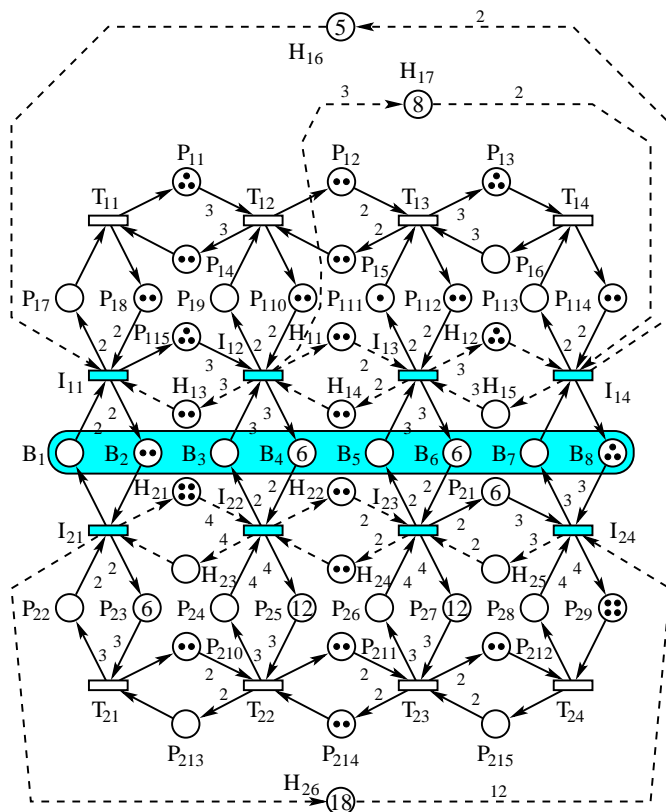


Figura 7.10: \mathcal{ES} del WTS de la figura 7.7 (técnicas generales).

agregados cuyos grafos de alcanzabilidad tienen 10093 estados el de \mathcal{LS}_1 , 8966 el de \mathcal{LS}_2 y 911 el de \mathcal{BS} . Estos sistemas agregados no tienen ningún marcado alcanzable espurio, lo que demuestra que los sistemas agregados obtenidos con las técnicas anteriores si los tienen, incluso preservando las resistencias máximas entre transiciones de interfaz (cosa que ya se había adelantado en el capítulo 4).

Por último, si se aplica la tercera técnica general (sección 6.6), que elimina agregaciones inadecuadas de estados no conectados por transiciones internas de subredes (algoritmos 6.33 y 6.42), entonces se obtienen los mismos sistemas agregados que con la segunda (en este ejemplo). Esto quiere decir que en este ejemplo no hay agregaciones inadecuadas de estados. Además se puede anticipar que los resultados numéricos con las técnicas generales

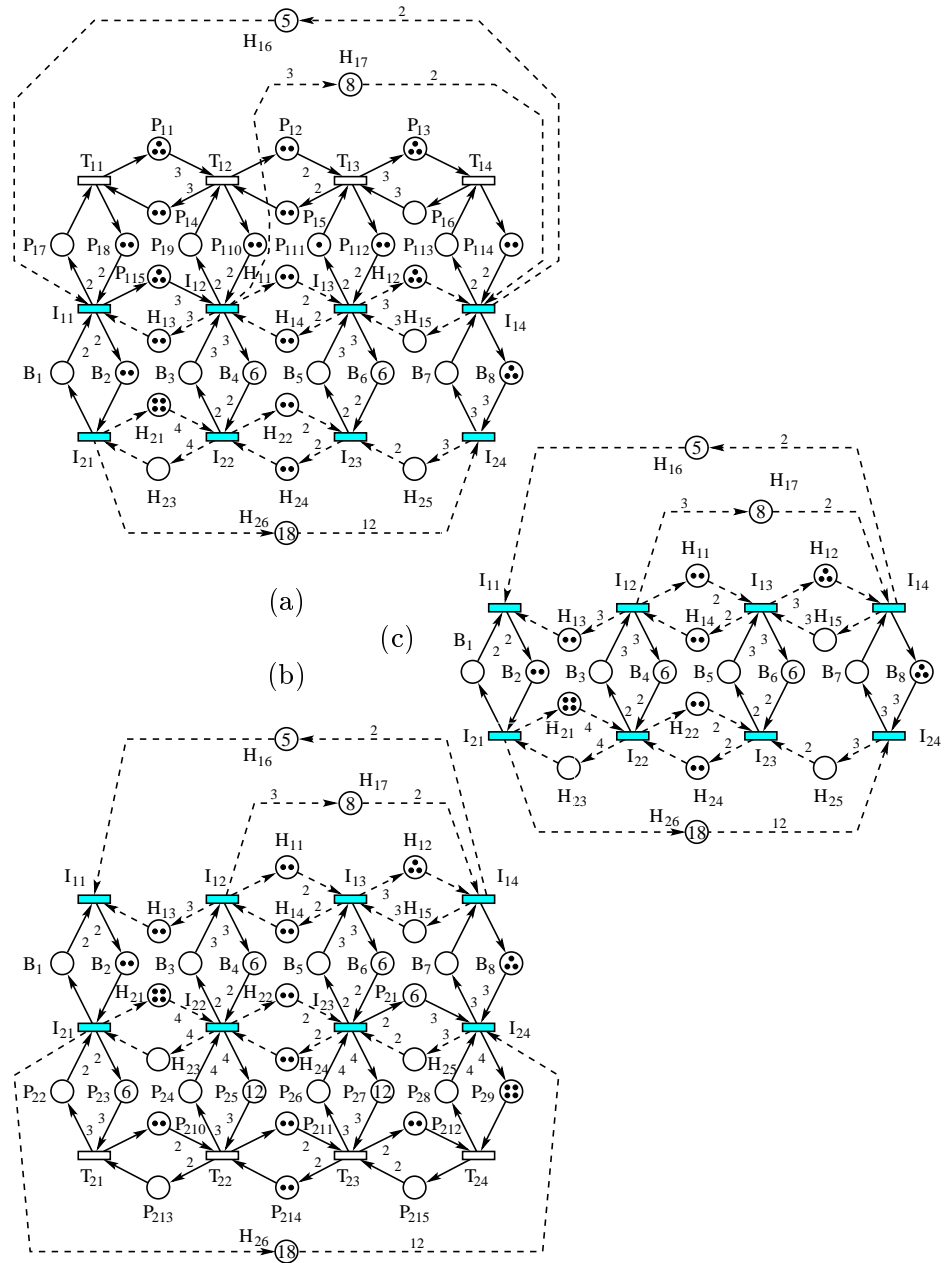


Figura 7.11: (a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del WTS de la figura 7.7 (técnicas generales).

2 y 3 serán exactamente los mismos, al no haber ninguna variación en el algoritmo numérico. Por ello, sólo se aplicará una de las dos técnicas, por ejemplo la segunda.

Para aproximar el throughput de las transiciones del WTS original basta aplicar el algoritmo iterativo 6.43 a los sistemas agregados. Se han realizado los mismos tres experimentos que con la técnica de WTS's, para comparar resultados. El primero poniendo las tasas de todas las transiciones a 1 (semántica de un solo servidor), el segundo poniendo tasa 10 a las transiciones de la primera subred (la de la parte superior) y tasa 0.1 a las de la segunda subred (la de la parte inferior) y el tercero poniendo tasas arbitrarias a todas las transiciones (las mismas que en [PJCS96b]). Estos tres experimentos se realizarán con las técnicas generales 1 y 2 (la técnica 3 da lo mismos resultados que la 2).

$\mathcal{L}S_1$						$\mathcal{L}S_2$					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.2448	0.4630	0.0826	0.0408	0.5204	1.0089	0.4271	0.4571	0.2524	0.2179	0.2037	1.0470
0.2178	0.4055	0.0732	0.0363	0.3821	1.0256	0.4319	0.4629	0.2556	0.2233	0.2195	1.0397
0.2163	0.4016	0.0726	0.0361	0.3759	1.0278	0.4325	0.4636	0.2560	0.2235	0.2203	1.0395
0.2163	0.4014	0.0726	0.0360	0.3757	1.0278	0.4325	0.4637	0.2560	0.2235	0.2204	1.0395
0.2163	0.4014	0.0726	0.0360	0.3757	1.0278	0.4325	0.4637	0.2560	0.2235	0.2204	1.0395
$\mathcal{X}(I_{11})$ exacto: 0.217662						Error: -0.64%					

Tabla 7.9: Aproximación de \mathcal{X} para Fig. 7.7. Tasas 1.0 (técnica general 1).

En la tabla 7.9 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas 1 con la primera técnica general (la de la sección 6.3). El throughput exacto de la transición I_{11} de la red original es 0.217662 y la leyenda de la tabla es la misma que en anteriores experimentos. En este caso se observa que se alcanza la convergencia en 5 iteraciones, obteniendo una aproximación de throughput de 0.216261 por lo que el error en la aproximación final es del -0.64% .

En la tabla 7.10 aparecen los resultados en el caso de tasas 1 con la segunda o tercera técnica general (las de las secciones 6.5 y 6.6). En este caso se observa que se alcanza la convergencia en 4 iteraciones, obteniendo una aproximación de throughput de 0.215806 por lo que el error en la aproximación final es del -0.85% .

En la tabla 7.11 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas de diferentes órdenes de magnitud (10 para las transiciones de la primera subred y 0.1 para las de la segunda) con la

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.2448	0.4630	0.0826	0.0408	0.4146	1.0045	0.4269	0.4568	0.2524	0.1248	0.2182	1.0271
0.2162	0.3994	0.0726	0.0360	0.3351	1.0151	0.4315	0.4624	0.2554	0.1264	0.2241	1.0262
0.2158	0.3985	0.0725	0.0360	0.3342	1.0155	0.4316	0.4626	0.2554	0.1265	0.2243	1.0262
0.2158	0.3984	0.0724	0.0360	0.3342	1.0154	0.4316	0.4626	0.2554	0.1265	0.2243	1.0262
$\mathcal{X}(I_{11})$ exacto: 0.217662						Error: -0.85%					

Tabla 7.10: Aproximación de \mathcal{X} para Fig. 7.7. Tasas 1.0 (técnicas generales 2 y 3).

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.0399	0.0040	0.0013	0.0007	0.1258	1.0000	0.0554	0.6567	0.3415	0.2233	0.3001	1.0397
0.0277	0.0028	0.0009	0.0005	0.0550	1.1999	0.0554	0.6569	0.3416	0.2233	0.3003	1.0394
0.0277	0.0028	0.0009	0.0005	0.0550	1.2256	0.0554	0.6570	0.3416	0.2233	0.3003	1.0394
0.0277	0.0028	0.0009	0.0005	0.0550	1.2256	0.0554	0.6570	0.3416	0.2233	0.3003	1.0394
$\mathcal{X}(I_{11})$ exacto: 0.027686						Error: +0.01%					

Tabla 7.11: Aproximación de \mathcal{X} para Fig. 7.7. Tasas 10 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 (técnica general 1).

primera técnica general (la de la sección 6.3). El throughput exacto de la transición I_{11} de la red original es 0.027686. En este caso se observa que se alcanza la convergencia en 4 iteraciones, obteniendo una aproximación de throughput de 0.027689 por lo que el error en la aproximación final es del +0.01%.

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.0399	0.0040	0.0013	0.0007	0.0061	1.0000	0.0554	0.6567	0.3415	0.1605	0.3003	1.0312
0.0277	0.0028	0.0009	0.0005	0.0041	1.0000	0.0554	0.6567	0.3415	0.1605	0.3003	1.0312
0.0277	0.0028	0.0009	0.0005	0.0041	1.0000	0.0554	0.6567	0.3415	0.1605	0.3003	1.0312
$\mathcal{X}(I_{11})$ exacto: 0.027686						Error: 0.00%					

Tabla 7.12: Aproximación de \mathcal{X} para Fig. 7.7. Tasas 10 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 (técnicas generales 2 y 3).

En la tabla 7.12 aparecen los resultados del experimento análogo con la segunda o tercera técnica general (las de las secciones 6.5 y 6.6). En este caso se alcanza la convergencia en 3 iteraciones, obteniendo una aproximación de throughput de 0.027686, alcanzando el throughput exacto (error 0.00%).

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.8091	0.8276	0.0949	0.0674	0.6370	1.0348	1.5116	0.1740	0.1691	0.2404	0.1229	1.1235
0.7798	0.7928	0.0896	0.0650	0.5377	1.0478	1.5445	0.1783	0.1735	0.2544	0.1386	1.1014
0.7767	0.7867	0.0892	0.0647	0.5256	1.0498	1.5517	0.1793	0.1744	0.2560	0.1408	1.0991
0.7763	0.7860	0.0892	0.0647	0.5243	1.0500	1.5525	0.1794	0.1745	0.2562	0.1410	1.0989
0.7763	0.7859	0.0892	0.0647	0.5242	1.0500	1.5525	0.1794	0.1745	0.2562	0.1410	1.0989
$\mathcal{X}(I_{11})$ exacto: 0.778787										Error: -0.32%	

Tabla 7.13: Aproximación de \mathcal{X} para Fig. 7.7. Tasas variables (técnica general 1).

El último experimento se ha realizado poniendo a todas las transiciones tasas elegidas al azar (las mismas que en el experimento análogo con la técnica de WTS's). Las tasas de las transiciones siguiendo el orden de izquierda a derecha primero y de arriba a abajo después son: 2, 3, 2, 3, 6, 3, 2, 4, 10, 8, 2, 4, 4, 6, 8 y 5. En la tabla 7.13 aparecen los resultados de la ejecución del algoritmo iterativo en este experimento por medio de la primera técnica general (la de la sección 6.3). El throughput exacto de la transición I_{11} de la red original es 0.778787. En este caso se observa que se alcanza la convergencia en 5 iteraciones, obteniendo una aproximación de throughput de 0.776283 por lo que el error en la aproximación final es del -0.32% .

\mathcal{LS}_1						\mathcal{LS}_2					
$\mathcal{X}(I_{11})$	μ_{11}	μ_{12}	μ_{13}	μ_{14}	λ	$\mathcal{X}(I_{21})$	μ_{21}	μ_{22}	μ_{23}	μ_{24}	λ
0.8091	0.8275	0.0949	0.0674	0.5066	1.0164	1.5068	0.1733	0.1694	0.1412	0.1422	1.0457
0.7749	0.7770	0.0889	0.0646	0.4493	1.0308	1.5464	0.1786	0.1741	0.1456	0.1506	1.0424
0.7742	0.7753	0.0888	0.0645	0.4474	1.0312	1.5483	0.1788	0.1743	0.1458	0.1509	1.0423
0.7742	0.7753	0.0888	0.0645	0.4473	1.0313	1.5484	0.1788	0.1743	0.1458	0.1509	1.0423
$\mathcal{X}(I_{11})$ exacto: 0.778787										Error: -0.59%	

Tabla 7.14: Aproximación de \mathcal{X} para Fig. 7.7. Tasas variables (técnicas generales 2 y 3).

En la tabla 7.14 aparecen los resultados del experimento análogo con la segunda o tercera técnica general (las de las secciones 6.3 y 6.3). En este caso se observa que se alcanza la convergencia en 4 iteraciones, obteniendo una aproximación de throughput de 0.774189 por lo que el error en la aproximación final es del -0.59% .

A modo de resumen, la tabla 7.15 presenta los resultados finales obteni-

Técnica	Tasas 1			Tasas 10, 0.1			Tasas variables		
	Aprox	It	Error	Aprox	It	Error	Aprox	It	Error
WTS's	0.216682	5	-0.45%	0.027692	4	+0.02%	0.777055	5	-0.22%
General 1	0.216261	5	-0.64%	0.027689	4	+0.01%	0.776283	5	-0.32%
General 2, 3	0.215806	4	-0.85%	0.027686	3	0.00%	0.774189	4	-0.59%
$\mathcal{X}(I_{11})$ exacto	0.217662			0.027686			0.778787		

Tabla 7.15: Comparación de técnicas para el WTS de la Fig. 7.7.

dos en los 9 experimentos realizados con el WTS de la figura 7.7. En cada fila aparecen los resultados de cada técnica empleada con sus tres experimentos variando las tasas. Para cada experimento se muestra el valor final de la aproximación del throughput de la transición I_{11} , el número de iteraciones que ha necesitado el algoritmo numérico para converger y el error de la aproximación. En la última fila se indican los valores exactos del throughput dependiendo de las tasas que se han utilizado en cada caso.

Se puede observar que en este ejemplo no hay grandes diferencias en las aproximaciones obtenidas con cualquiera de las técnicas empleadas. Esto no quiere decir que de igual emplear una u otra en WTS's. Se ha decidido seguir con este ejemplo, porque fue desarrollado para aplicar los conceptos de ganancia, resistencia y marcado ponderado. En ese momento no se habían desarrollado las técnicas generales del capítulo 6.

En este ejemplo no se obtienen diferencias importantes entre las distintas técnicas debido a una serie de factores. Por un lado, dentro de las subredes producto del corte no hay caminos con elevada resistencia interna respecto a su ganancia. Esto motiva que la técnica general 1 de resultados aceptables. En general, esta técnica no preserva la resistencia de los caminos, lo que puede provocar errores muy grandes en WTS's incluso muy sencillos (ver caso de la figura 4.3). Por otro lado, en este caso no hay agregaciones inadecuadas de estados no conectados por transiciones internas del WTS, lo que provoca que los resultados obtenidos con las técnicas generales 2 y 3 sean los mismos, cuando hay ejemplos incluso sencillos de que esto no es así en general (ver caso de la figura 6.2).

Las diferentes aproximaciones obtenidas entre la técnica de WTS's y la primera técnica general, pese a que se parte de los mismos sistemas agregados, se deben a que en la técnica de WTS's la estimación de las utilidades de las transiciones de interfaz es más realista que la utilizada en la primera técnica general.

Cuando se ponen tasas de diferentes órdenes de magnitud entre las dis-

tintas subredes la convergencia del algoritmo numérico es muy rápida y el error muy pequeño. Este hecho tiene su sentido ya que la subred de tasas más bajas es la que ralentiza todo el sistema y el \mathcal{LS}_i que la contenga tiene unas prestaciones muy parecidas a la red original. No importa si la reducción de la otra subred es buena o no, porque la que domina la velocidad total de la red es la subred lenta.

En cuanto al número de iteraciones, las técnicas generales 2 y 3 convergen aproximadamente en una iteración menos debido a que los sistemas agregados con los que se opera tienen el funcionamiento más parecido al sistema original que puede conseguirse a nivel de espacio de estados.

7.3 DSSP's

Esta sección se va a dedicar al estudio de dos ejemplos de aproximación de throughput de las transiciones de DSSP's tomados de [PJCS96c, PJCS96a]. A un DSSP se le pueden aplicar varias técnicas de descomposición y reducción de las desarrolladas en esta memoria. Obviamente se le pueden aplicar las técnicas del capítulo 5 que estaban desarrolladas precisamente para DSSP's, basadas en las reducciones total o parcial de algunas de las SM's que componen el DSSP. Y también es posible aplicar las tres técnicas del capítulo 6 que estaban desarrolladas para poderse aplicar a cualquier red de Petri. Se aprovechará esta sección para comparar los resultados que se obtienen con las 5 técnicas de descomposición y reducción aplicadas a la misma red original.

El primer ejemplo, tomado de [PJCS96c] es un modelo de un sistema de fabricación cuyo esquema aparece en la figura 7.12. Este sistema puede fabricar dos tipos de productos (A y B). Dependiendo del tipo de producto a fabricar, la materia prima tiene que seguir distintos caminos en el sistema, pasando por distintas máquinas hasta su elaboración final y salida del sistema. La salida del sistema se produce también por distintos puntos dependiendo del tipo de producto. De esta manera, el producto de tipo A es manufacturado por las máquinas 1 y 2 y el de tipo B por las máquinas 1, 3 y 4. Cada máquina necesita una cantidad mínima de los subproductos de entrada para poder operar. Estas cantidades mínimas se modelan con el número de soportes necesarios para transportarlas. Por ejemplo, la máquina 1 necesita dos soportes de materia prima para poder trabajar, la máquina 2 necesita 3 soportes, etc. La materia prima entra en el sistema

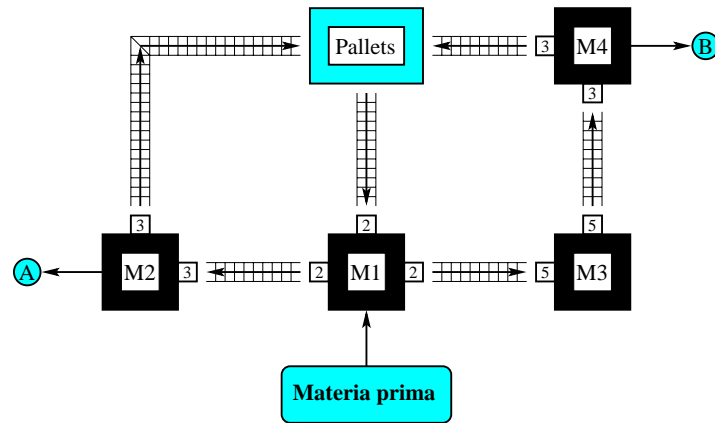


Figura 7.12: Sistema de fabricación.

por la máquina 1 y los productos finales dejan el sistema por las máquinas 2 (producto de tipo A) y 4 (producto de tipo B).

Se va a aproximar el funcionamiento temporal del sistema en términos de la velocidad de producción de cada tipo de producto, en función del número de soportes disponibles para el transporte interno de los distintos subproductos.

El funcionamiento del sistema de la figura 7.12 puede modelarse por medio de la red de la figura 7.13. En esta red, el comportamiento interno de cada máquina del sistema se modela con una SM. Por ejemplo, las máquinas 1 y 2 están modeladas por las SM's cuyos nodos tienen etiquetas P_{ij} , I_{ij} , T_{ij} y t_{ij} con $i = 1, 2$. Las máquinas 3 y 4 están modeladas por las dos SM's cuyos nodos tienen etiquetas del tipo P_{3j} , I_{3j} , T_{3j} y t_{3j} (la máquina 3 del sistema se corresponde con la SM de la parte inferior derecha de la red de la figura 7.13 y la máquina 4 con la de la parte superior derecha).

Los lugares con etiquetas $\{B_i\}_{i=1}^4$ modelan las cintas de transporte entre máquinas del sistema. Estos lugares serán los canales del sistema. En particular, el canal B_2 contiene n marcas, que es el número total de soportes disponibles en el sistema. Este será el parámetro del estudio. Para que el sistema sea vivo son necesarios por lo menos 10 soportes, es decir, $n \geq 10$. Los pesos de los arcos que conectan transiciones de las SM's con los canales modelan el número de soportes que necesita una máquina del sistema para trabajar.

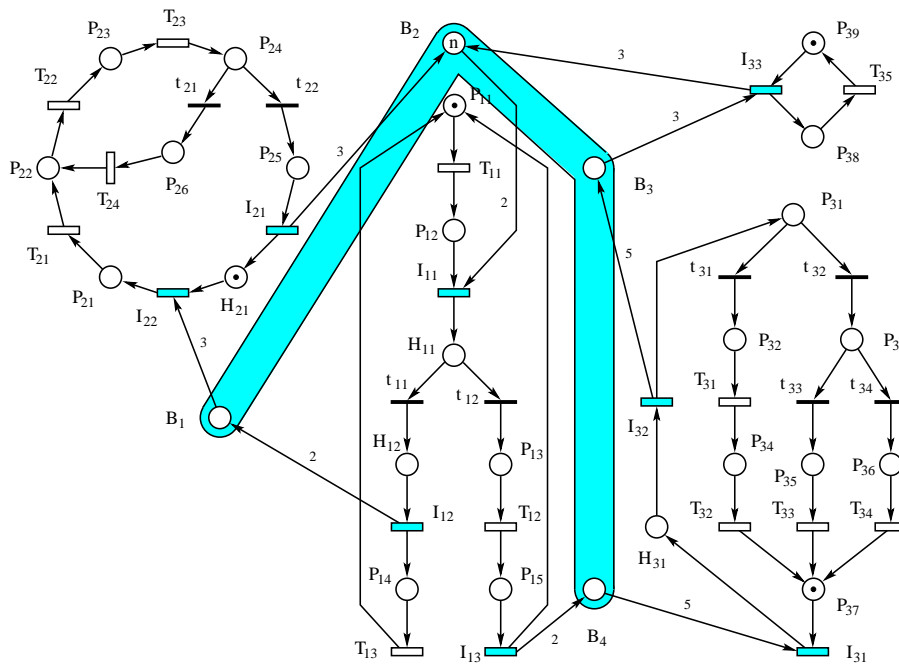


Figura 7.13: DSSP que modela el sistema de la figura 7.12.

El resultado de modelar el sistema de fabricación produce en este caso un DSSP formado por 4 SM's que están conectadas entre sí por medio de 4 canales. Además, el objetivo del estudio es calcular la velocidad a la que el sistema es capaz de fabricar los productos de tipo A y B. Estas velocidades se pueden medir en términos de los throughput de las transiciones I_{21} para el producto de tipo A e I_{33} para el producto de tipo B. Por lo tanto, es posible emplear las técnicas desarrolladas en los capítulos 5 y 6 para aproximar el throughput de las transiciones de un DSSP.

Por lo que respecta a la evolución temporal de la red, las tasas de disparo de las transiciones aparecen en la tabla 7.16. En ella aparecen tanto las transiciones temporizadas como las inmediatas. En el caso de las transiciones inmediatas, el valor hay que interpretarlo como el peso asociado al disparo de la misma.

Como el estudio se va a realizar en función del número n de soportes disponibles en el sistema, el tamaño de los grafos de alcanzabilidad de la red original y de los distintos sistemas agregados es una función de n . Por

Tasa	Transiciones
1.0	$I_{13}, I_{22}, T_{24}, T_{31}, t_{12}, t_{21}, t_{31}, t_{33}$
2.0	T_{11}, T_{13}, T_{23}
3.0	$I_{12}, I_{21}, T_{22}, T_{32}, t_{11}$
4.0	T_{12}, T_{33}
5.0	$I_{11}, I_{32}, I_{33}, T_{21}, T_{35}, t_{22}, t_{32}, t_{34}$
7.0	T_{34}
10.0	I_{31}

Tabla 7.16: Tasas de las transiciones del DSSP de la Fig. 7.13.

ello, se desarrollará en detalle el proceso de descomposición y reducción para $n = 17$ y después se pondrán los resultados obtenidos para otros valores del parámetro.

El grafo de alcanzabilidad de esta red tiene (para $n = 17$) 242448 estados. Se ha elegido como corte el formado por los 4 canales, efectuando una descomposición de la red original en 3 subredes ($\mathcal{N}_1, \mathcal{N}_2$ y \mathcal{N}_3). La subred \mathcal{N}_1 (la de la parte central) está formada por las transiciones internas temporizadas $\{T_{1i}\}_{i=1}^3$, las transiciones internas inmediatas t_{11} y t_{12} , las transiciones de interfaz $\{I_{1i}\}_{i=1}^3$ y los lugares internos $\{P_{1i}\}_{i=1}^5$ y $\{H_{1i}\}_{i=1}^2$. La subred \mathcal{N}_2 (la de la parte superior izquierda) está formada por las transiciones internas temporizadas $\{T_{2i}\}_{i=1}^4$, las transiciones internas inmediatas t_{21} y t_{22} , las transiciones de interfaz I_{21} e I_{22} y los lugares internos $\{P_{2i}\}_{i=1}^6$ y H_{21} . La subred \mathcal{N}_3 (la de la parte derecha) está formada por las transiciones internas temporizadas $\{T_{3i}\}_{i=1}^5$, las transiciones internas inmediatas $\{t_{3i}\}_{i=1}^4$, las transiciones de interfaz $\{I_{3i}\}_{i=1}^3$ y los lugares internos $\{P_{3i}\}_{i=1}^9$ y H_{31} . Se ha cambiado de nuevo la notación de algunos lugares internos (etiquetas H_{ij} en vez de P_{ij}) porque luego resultarán ser lugares implícitos necesarios para resumir el comportamiento de las subredes, es decir, se ha tomado una notación tal que mantenga las etiquetas de los lugares implícitos.

La primera técnica que se va a aplicar es la de reducción parcial de SM's (técnica desarrollada en detalle en la sección 5.2). En esta técnica los sistemas agregados se componen de las reducciones parciales de ciertas SM's y se mantienen los canales y las transiciones de entrada y salida de los mismos. Por lo tanto, es necesario calcular las reducciones parciales de cada SM. Estas reducciones pueden realizarse localmente, así que se detallará la reducción de una SM cualquiera, por ejemplo la correspondiente a la

máquina 1 del sistema de fabricación. Para el resto se dará el resultado final de la reducción. Para efectuar las reducciones es necesario conocer los pesos asignados a cada transición inmediata de la red original, ya que estos pesos son los que resuelven los conflictos internos de las SM's y permiten conocer las probabilidades de encaminamiento entre sus transiciones (definición 5.2). Estos pesos también aparecen en la tabla 7.16.

La reducción de SM's se realiza por medio de los algoritmos 5.5 y 5.6. El primero calcula las probabilidades de ramificación entre transiciones observables (las transiciones de interfaz) y el segundo calcula la estructura de la SM reducida.

El algoritmo 5.5 toma como entrada la matriz de probabilidades de encaminamiento entre transiciones temporizadas de la SM en cuestión. La SM correspondiente a la máquina 1 del sistema de fabricación tiene 6 transiciones temporizadas, de las cuales 3 son observables. Por lo tanto, teniendo en cuenta los pesos de las transiciones inmediatas t_{11} y t_{12} , la matriz \mathbf{P} de probabilidades de encaminamiento de la SM es:

$$\mathbf{P} = \begin{array}{l} I_{11} \\ I_{12} \\ I_{13} \\ T_{11} \\ T_{12} \\ T_{13} \end{array} \begin{array}{c} I_{11} \quad I_{12} \quad I_{13} \quad T_{11} \quad T_{12} \quad T_{13} \\ \left[\begin{array}{ccc|ccc} 0 & \frac{3}{4} & 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right] \end{array}$$

La interpretación de esta matriz es la siguiente. El elemento $\mathbf{P}(i, j)$ es la probabilidad de que en la SM después del disparo de la transición de índice i sea la transición de índice j la que se sensibiliza (con la resolución de conflictos por medio de transiciones inmediatas este hecho equivale a que la siguiente transición en dispararse será la de índice j). Por ejemplo, $\mathbf{P}(I_{11}, I_{12}) = 3/4$ indica que después del disparo de I_{11} hay una probabilidad $3/4$ de que se sensibilice I_{12} . Se han ordenado las transiciones temporizadas de la SM de forma que aparezcan primero las observables, para facilitar la implementación del algoritmo 5.5.

Al aplicar el algoritmo 5.5 a la matriz \mathbf{P} , se produce como salida una matriz \mathbf{Q} con las probabilidades de ramificación entre transiciones observables (\mathbf{Q} puede ser la misma matriz que \mathbf{P} en la implementación real, aunque la interpretación de sus datos es diferente a la que se ha hecho antes). Esta

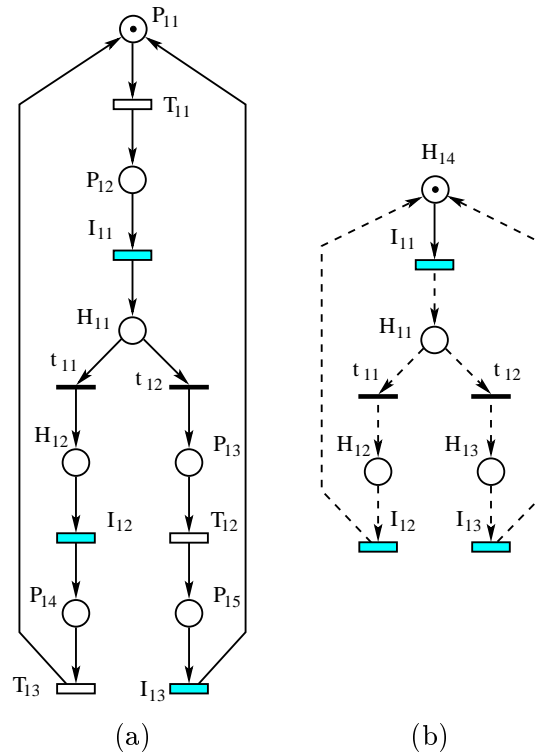


Figura 7.14: (a) Una SM y (b) su reducción parcial.

matriz \mathbf{Q} es de tamaño 3×3 en este caso ya que la SM tiene 3 transiciones observables.

$$\mathbf{Q} = \begin{matrix} & I_{11} & I_{12} & I_{13} \\ \begin{matrix} I_{11} \\ I_{12} \\ I_{13} \end{matrix} & \begin{bmatrix} 0 & \frac{3}{4} & \frac{1}{4} \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{matrix}$$

La interpretación de los elementos de esta matriz es la siguiente. El elemento $\mathbf{Q}(i, j)$ es la probabilidad de que en la SM, después del disparo de la transición observable de índice i , sea la transición observable de índice j la siguiente transición observable que se sensibiliza. Por ejemplo, $\mathbf{Q}(I_{11}, I_{13}) = 1/4$ indica que después del disparo de la transición observable I_{11} hay una probabilidad $1/4$ de que la siguiente transición observable

que se sensibilice sea I_{12} (aunque para ello es necesario que se dispare previamente la transición interna T_{12}). La matriz \mathbf{Q} contiene lo que se ha denominado en la memoria como probabilidades de ramificación entre transiciones observables, para distinguirlas de las anteriores probabilidades de encaminamiento.

El siguiente paso consiste en aplicar el algoritmo 5.6 a la matriz \mathbf{Q} , dando como resultado la SM reducida parcialmente. En la figura 7.14 puede verse la SM original y la reducida por medio de los algoritmos anteriores. Notar que los pesos asociados a las transiciones inmediatas de la SM reducida vienen dados por la matriz \mathbf{Q} (en este caso coinciden con las de igual etiqueta en la SM original).

Una vez expuesta la reducción parcial de una SM, se procede de la misma manera con las otras SM's del DSSP, obteniéndose las reducciones parciales de todas las SM's. Con las SM's reducidas es posible generar directamente los sistemas agregados. Sin embargo, por similitud con el resto de la memoria, se va a construir un sistema extendido \mathcal{ES} y a partir de él los sistemas agregados. El sistema extendido es el formado a partir de la red original, añadiendo el conjunto de nodos necesario para la reducción de cada SM. Como algunos de los lugares que resultan del cálculo de las SM's reducidas ya están en la red original, no se duplican. En la figura 7.15 puede observarse el sistema extendido correspondiente a la red de la figura 7.13.

A partir del \mathcal{ES} es inmediato construir los sistemas agregados. En las figuras 7.16.a, 7.16.b, 7.17.a y 7.17.b se pueden observar el \mathcal{LS}_1 , \mathcal{LS}_2 , \mathcal{LS}_3 y \mathcal{BS} correspondientes al corte realizado en la red original. El \mathcal{LS}_1 tiene (para $n = 17$) 12348 estados alcanzables, el \mathcal{LS}_2 14049, el \mathcal{LS}_3 46594 y el \mathcal{BS} 5849.

Para aproximar el throughput de las transiciones del DSSP original basta aplicar el algoritmo iterativo 5.23 a los sistemas agregados. Se han realizado múltiples experimentos variando el valor del parámetro n . Se va a dar el resultado detallado de las iteraciones del algoritmo iterativo en el caso $n = 17$ y para el resto de casos simplemente el valor final de la aproximación y el error cometido.

En el caso de DSSP's, a diferencia del caso de MG's, no todas las transiciones tienen el mismo throughput. Pero por el teorema 5.9, las ratios de visita de las transiciones de un DSSP (throughput relativos) quedan determinadas por las conexiones entre SM's y la resolución de los conflictos internos de las SM's. En la técnica de reducción se preservan estas ratios de visita por lo que, al igual que en WTS's, es suficiente conocer el throughput

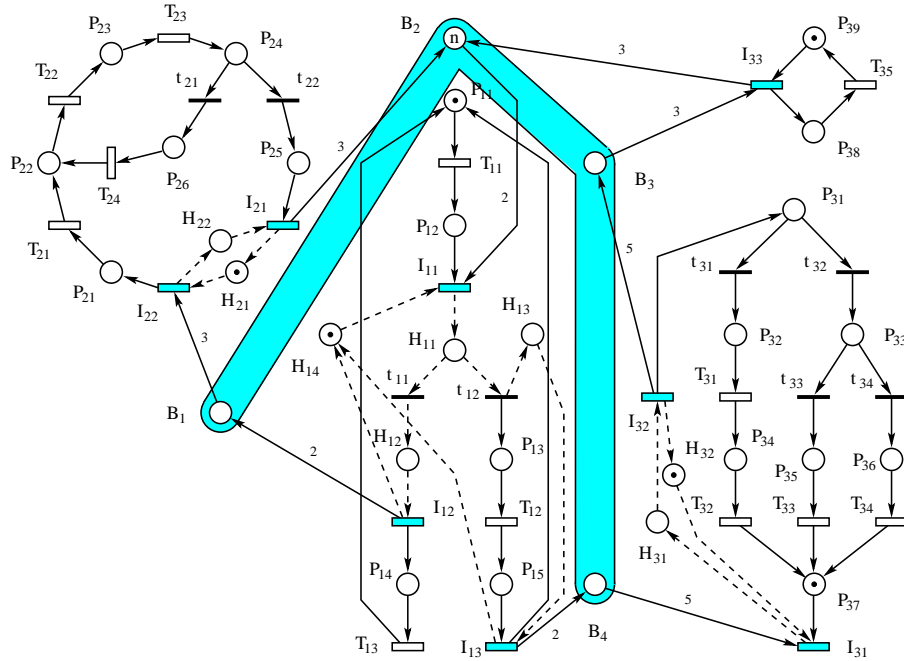


Figura 7.15: \mathcal{ES} del DSSP de la figura 7.13 (reducción parcial).

de una transición cualquiera para conocer los de las demás. El error en tanto por ciento cometido en la aproximación del throughput de cualquier otra transición será el mismo.

\mathcal{LS}_1		\mathcal{LS}_2		\mathcal{LS}_3	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ	$\mathcal{X}(I_{31})$	λ
0.610332	1.002251	0.286059	1.053280	0.057211	1.158418
0.587708	1.017473	0.293559	1.049884	0.058712	1.158418
0.587289	1.017553	0.293643	1.049856	0.058728	1.154018
0.587284	1.017553	0.293642	1.049856	0.058729	1.154018
$\mathcal{X}(I_{21})$ exacto: 0.294443				Error: -0.27%	

Tabla 7.17: Aproximación de \mathcal{X} para Fig. 7.13 ($n = 17$, reducción parcial).

En la tabla 7.17 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de $n = 17$. En este caso, el throughput exacto de la transición I_{21} de la red original (que modela la tasa de producción del producto de tipo A en el sistema de fabricación) es 0.294443. La tabla

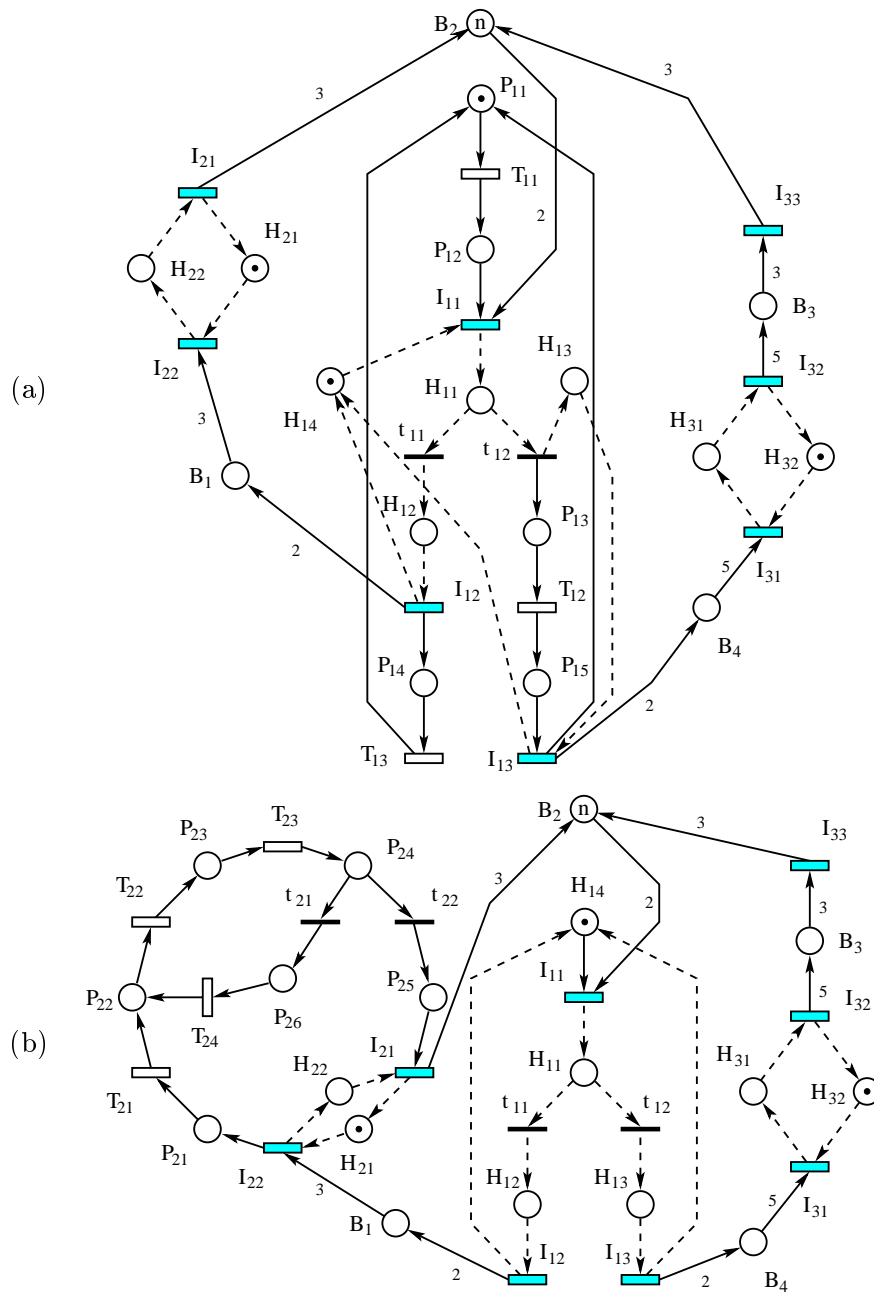


Figura 7.16: (a) \mathcal{LS}_1 y (b) \mathcal{LS}_2 de la figura 7.13 (reducción parcial).

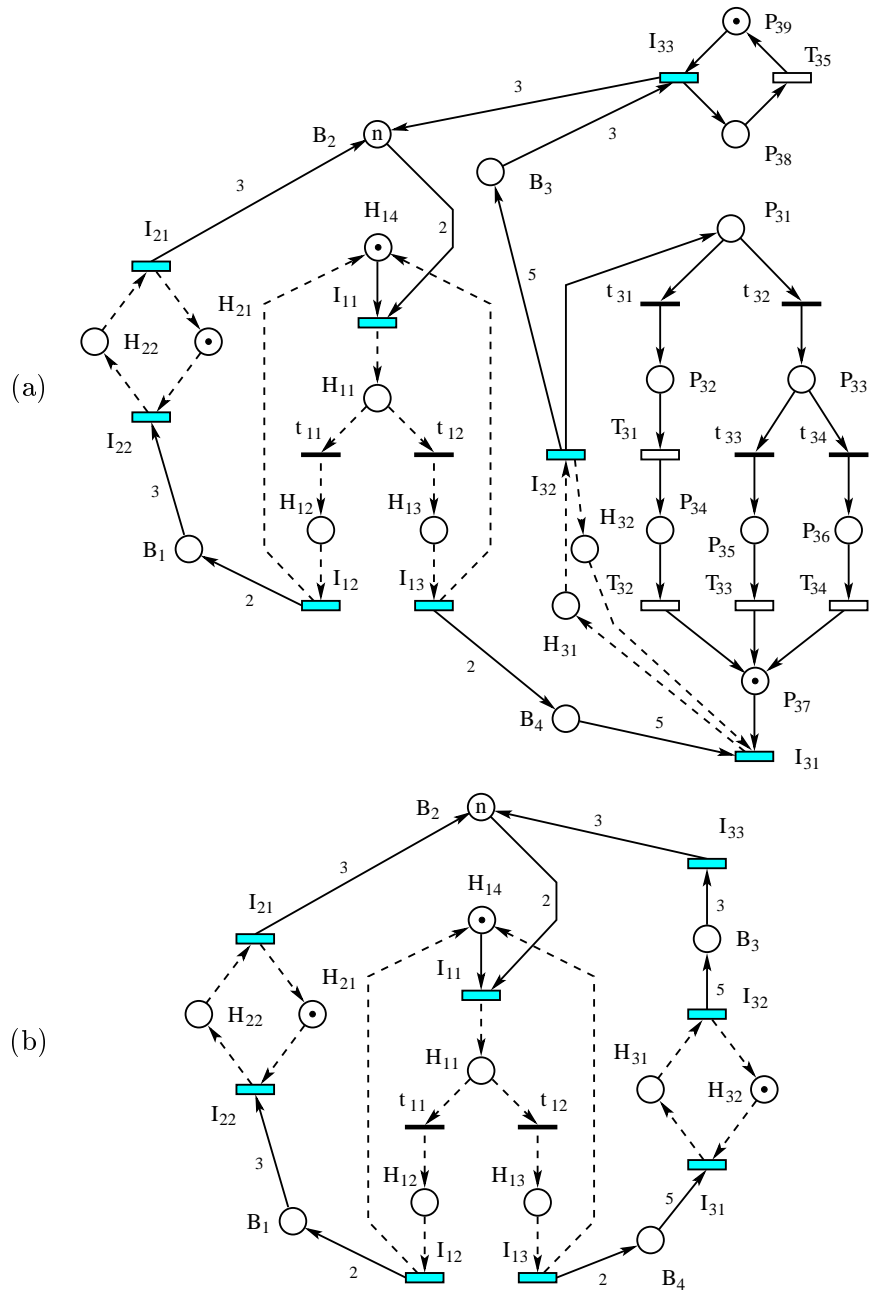


Figura 7.17: (a) \mathcal{LS}_3 y (b) \mathcal{BS} de la figura 7.13. (reducción parcial).

está dividida en tres partes encabezadas con \mathcal{LS}_1 , \mathcal{LS}_2 y \mathcal{LS}_3 en las que se recogen los resultados del estudio de cada sistema de bajo nivel y la correspondiente visita al \mathcal{BS} . En cada \mathcal{LS}_i se calcula el throughput de la transición de interfaz I_{i1} . En este caso se han eliminado de la tabla las utilizaciones de las transiciones de interfaz de la subred \mathcal{N}_i presente en el \mathcal{LS}_i para reducir el tamaño de la tabla, pero el algoritmo sigue haciendo cálculos con ellas. Por lo que respecta al estudio del \mathcal{BS} , se indica el factor de escala λ que permite la actualización de las tasas de disparo de las transiciones de interfaz de \mathcal{LS}_i . Estas nuevas tasas no se incluyen en la tabla para no aumentar su tamaño. En cada fila de la tabla aparecen los resultados de cada iteración del algoritmo, hasta alcanzar la convergencia. En la última fila de la tabla aparece a la izquierda el throughput exacto de la transición de referencia en la red original y a la derecha el error de la aproximación. En este caso se observa que se alcanza la convergencia en 4 iteraciones, obteniendo una aproximación de throughput de 0.293642 por lo que el error en la aproximación final es del -0.27% .

Se han realizado otros experimentos con diferentes valores del parámetro n (número de soportes disponibles en el sistema). En vez de poner una tabla similar a la anterior para cada uno, se van a reunir las principales características de cada experimento en la siguiente tabla.

n	Marcados alcanzables					$\mathcal{X}(T_{21})$	$\mathcal{X}(T_{21})$	% Error
	Original	\mathcal{LS}_1	\mathcal{LS}_2	\mathcal{LS}_3	\mathcal{BS}	Exacto	Aprox.	
10	40092	2058	1932	8696	956	0.247962	0.245527	-0.982
11	55608	2832	2763	11734	1319	0.262604	0.260446	-0.822
12	74868	3798	3834	15444	1774	0.272949	0.271174	-0.650
13	98304	4980	5181	19898	2333	0.280018	0.278563	-0.520
14	126348	6402	6840	25168	3008	0.285181	0.283951	-0.431
15	159432	8088	8847	31326	3811	0.289048	0.287991	-0.366
16	197988	10062	11238	38444	4754	0.292068	0.291153	-0.313
17	242448	12348	14049	46594	5849	0.294443	0.293642	-0.272
18	293244	14970	17316	55848	7108	0.296345	0.295641	-0.238
19	350808	17952	21075	66278	8543	0.297880	0.297258	-0.209
20	415572	21318	25362	77956	10166	0.299131	0.298580	-0.184
21	487968	25092	30213	90954	11989	0.300156	0.299665	-0.164
22	568428	29298	35664	105344	14024	0.301001	0.300564	-0.145
23	657384	33960	41751	121198	16283	0.301700	0.301314	-0.128
24	755268	39102	48510	138588	18778	0.302282	0.301942	-0.112
25	862512	44748	55977	157586	21521	0.302767	0.302463	-0.100

Tabla 7.18: Aproximaciones para diversos valores de n (reducción parcial).

Esta tabla contiene, para cada experimento, el valor del parámetro n , el

número de marcados alcanzables de la red original y de los sistemas agregados, los valores exacto y aproximado del throughput de la transición I_{21} y el error en tanto por ciento cometido en la aproximación. De esta forma puede compararse el tamaño del espacio de estados de la red original con los de los sistemas agregados. Por otro lado puede compararse el valor exacto del throughput de la transición de referencia con el valor de su aproximación y el error cometido. Puede verse que en este DSSP, al aumentar el valor del parámetro, la calidad de la aproximación mejora. Además, en todos los casos se obtienen errores inferiores al 1% sobre el valor exacto, lo que es bastante aceptable para una técnica de aproximación.

Ahora se va a realizar el estudio de la misma red aplicando otra técnica de descomposición y agregación. De esta forma se podrán comparar los resultados obtenidos con las distintas técnicas. La segunda técnica que se va a aplicar es la de reducción total de SM's (desarrollada en detalle en la sección 5.3). Como esta técnica no puede aplicarse a cualquier SM de cualquier DSSP, se utilizará en los casos en los que las SM's sean totalmente reducibles. Para el resto de SM's, se seguirá utilizando la técnica de reducción parcial. Por lo tanto, es necesario estudiar qué SM's del DSSP son reducibles totalmente a una única transición y cuáles no.

Para que una SM en un DSSP sea totalmente reducible a una única transición (ver definición 5.16) debe cumplir las propiedades **P1** y **P2** que están enunciadas en las propiedades 5.13 y 5.15 respectivamente.

La primera propiedad da una caracterización algebraica en términos del rango de una matriz para que la SM tenga un único comportamiento exterior observable en los canales de entrada y salida de la SM, independientemente de su evolución interna. Esta propiedad es necesaria para poder calcular los pesos de los arcos que conectarán la transición resumen con los canales, si la SM resulta ser totalmente reducible.

La segunda propiedad asegura que en cualquier evolución interna de la SM, ésta toma todas las marcas de los canales de entrada antes de depositar marcas en los canales de salida. De esta forma, el funcionamiento exterior de la SM se asemeja al disparo de una transición en una red de Petri y se podrá realizar la reducción.

Un ejemplo de SM totalmente reducible es el de la SM que modela a la máquina 2 del sistema de fabricación (la que tiene nodos de etiquetas P_{2i} , H_{2i} , I_{2i} , T_{2i} y t_{2i}). La matriz \mathbf{C}^* que se define en la propiedad 5.13 es la restricción de la matriz de incidencia del DSSP a los lugares y transiciones de la SM, a la que se le añaden las filas correspondientes a los canales de

entrada y salida de la misma (separadas por líneas horizontales). En este caso se tiene:

$$\mathbf{C}^* = \begin{array}{l} P_{21} \\ P_{22} \\ P_{23} \\ P_{24} \\ P_{25} \\ P_{26} \\ H_{21} \\ B_1 \\ B_2 \end{array} \left[\begin{array}{cccccccc} I_{21} & I_{22} & T_{21} & T_{22} & T_{23} & T_{24} & t_{21} & t_{22} \\ & 1 & -1 & & & & & \\ & & 1 & -1 & & 1 & & \\ & & & 1 & -1 & & & \\ & & & & 1 & & -1 & -1 \\ -1 & & & & & & & 1 \\ & & & & & -1 & 1 & \\ \hline 1 & -1 & & & & & & \\ \hline -3 & & & & & & & \\ \hline 3 & & & & & & & \end{array} \right]$$

Los elementos que aparecen en blanco son 0. Se dejan en blanco para aumentar la legibilidad de la matriz. Como $\text{rank}(\mathbf{C}^*) = 7 = |P_{\mathcal{SM}}|$, la SM cumple la propiedad **P1**. Además, es evidente que la SM toma las marcas del canal de entrada B_1 antes de depositar las marcas en el canal de salida B_2 , por lo que cumple la propiedad **P2**. En consecuencia, la SM correspondiente a la máquina 2 del sistema de fabricación es totalmente reducible a una única transición.

Puede observarse, que para cualquier t-semiflujo \mathbf{x} de la SM, se cumple que $\mathbf{C}^*[B_2] \cdot \mathbf{x} = -\mathbf{C}^*[B_1] \cdot \mathbf{x}$ lo que indica que en la transición $t_{\mathcal{SM}}$ que reduce a la SM los pesos de los arcos que conectan B_1 y B_2 con $t_{\mathcal{SM}}$ son iguales. Además, de acuerdo con la propiedad **P2**, el lugar inicialmente marcado es H_{21} y también es el único lugar bloqueado (por ser lugar de salida de una transición con canales de salida). Cualquier camino en la SM de H_{21} que recorra todas las transiciones con canales de entrada sin atravesar de nuevo H_{21} toma 3 marcas de B_1 por lo que el peso del arco $W(B_1, t_{\mathcal{SM}}) = 3$, es decir, $W(t_{\mathcal{SM}}, B_2) = 3$. Por lo tanto, se tiene calculado el vector de incidencia de la transición resumen $t_{\mathcal{SM}}$.

En la figura 7.18 puede verse la SM original y la reducida totalmente. Como la reducción es posible o no dependiendo de las conexiones de la SM con los canales, se incluyen también en las figuras estos canales de entrada y salida.

Un ejemplo de SM que no es totalmente reducible a una transición, es el de la SM que modela a la máquina 1 del sistema de fabricación (la que tiene nodos de etiquetas P_{1i} , H_{1i} , I_{1i} , T_{1i} y t_{1i}). La matriz \mathbf{C}^* es en este caso:

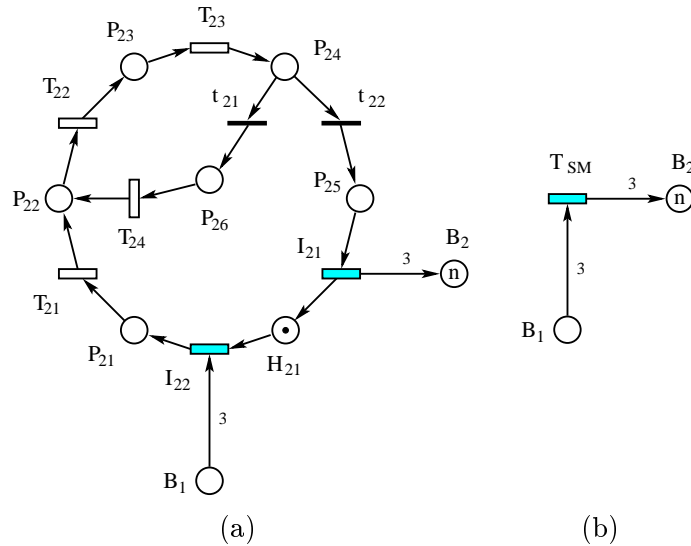


Figura 7.18: (a) Una SM en un DSSP y (b) su reducción total.

$$\mathbf{C}^* = \begin{array}{l} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ P_{15} \\ H_{11} \\ H_{12} \\ B_2 \\ B_1 \\ B_3 \end{array} \begin{array}{c} I_{11} \quad I_{12} \quad I_{13} \quad T_{11} \quad T_{12} \quad T_{13} \quad t_{11} \quad t_{12} \\ \left[\begin{array}{cccccccc} & & & 1 & -1 & & 1 & \\ -1 & & & & 1 & & & \\ & & & & & -1 & & 1 \\ & & 1 & & & & -1 & \\ & & -1 & & 1 & & & \\ 1 & & & & & & -1 & -1 \\ -1 & & & & & & & 1 \\ \hline -2 & & & & & & & \\ \hline & 2 & & & & & & \\ & & 2 & & & & & \end{array} \right] \end{array}$$

Los elementos que aparecen en blanco son 0. Se dejan en blanco para aumentar la legibilidad de la matriz. Como $\text{rank}(\mathbf{C}^*) = 8$ y $|P_{SM}| = 7$, la SM no cumple la propiedad **P1**. Puede observarse que la SM tiene dos t-semiflujos distintos con diferente efecto en los canales. Los posibles efectos son $(-2B_2, +2B_1)$ o $(-2B_2, +2B_4)$ dependiendo de que se dispare t_{11} o t_{12} . En consecuencia, la SM no es totalmente reducible a una única transición. Para esta SM se seguirá empleando la técnica de reducción parcial de SM's.

Completando el estudio de las 4 SM's del DSSP, se tiene que las SM's que modelan el comportamiento interno de las máquinas 2, 3 y 4 del sistema de fabricación son totalmente reducibles a una transición, mientras que la SM que modela el de la máquina 1 no lo es.

Una vez expuesta la técnica de reducción total de SM's en un DSSP, se van a construir los sistemas agregados. En este caso no tiene sentido hablar de sistema extendido como en el resto de técnicas de esta memoria, ya que hay ciertas transiciones de interfaz que desaparecen en el proceso (ver figura 7.18). Por lo tanto se construirán directamente los sistemas agregados. Esto no afecta al algoritmo numérico de aproximación ya que éste opera con los sistemas agregados y no con el sistema extendido.

En las figuras 7.19.a, 7.19.b, 7.20.a y 7.20.b se pueden observar el \mathcal{LS}_1 , \mathcal{LS}_2 , \mathcal{LS}_3 y \mathcal{BS} correspondientes al corte realizado en la red original empleando la técnica de reducción total de SM's. El \mathcal{LS}_1 tiene (para $n = 17$) 5769 estados alcanzables, el \mathcal{LS}_2 10722, el \mathcal{LS}_3 29774 y el \mathcal{BS} 2772. Se puede observar (ver tabla 7.18) que para el mismo valor de $n = 17$ se obtienen sistemas agregados con menos estados alcanzables, debido a la mayor agregación de algunas de las SM's.

Para aproximar el throughput de las transiciones del DSSP original basta aplicar el algoritmo iterativo 5.23 a los nuevos sistemas agregados. Se han realizado los mismos experimentos que con la técnica anterior.

\mathcal{LS}_1		\mathcal{LS}_2		\mathcal{LS}_3	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ	$\mathcal{X}(I_{31})$	λ
0.610510	1.001182	0.285909	0.937957	0.057159	1.033333
0.585532	1.013457	0.292107	0.944990	0.058421	1.033333
0.584544	1.013601	0.292257	0.945191	0.058452	1.029733
0.584521	1.013601	0.292259	0.945191	0.058452	1.029733
$\mathcal{X}(I_{21})$ exacto: 0.294443			Error: -0.74%		

Tabla 7.19: Aproximación de \mathcal{X} para Fig. 7.13 ($n = 17$, reducción total).

En la tabla 7.19 aparecen los resultados de la ejecución del algoritmo iterativo para $n = 17$. En este caso, el throughput exacto de la transición I_{21} de la red original (tasa de producción del producto de tipo A) es 0.294443. La leyenda de la tabla es la misma que con la técnica anterior. Para cada \mathcal{LS}_i se da el throughput de la transición de interfaz de referencia (I_{i1}) y el factor de escala encontrado en el estudio del \mathcal{BS} para actualizar las tasas de disparo de las transiciones de interfaz de \mathcal{LS}_i . En cada fila de la tabla aparecen los resultados de cada iteración del algoritmo, hasta alcanzar la convergencia.

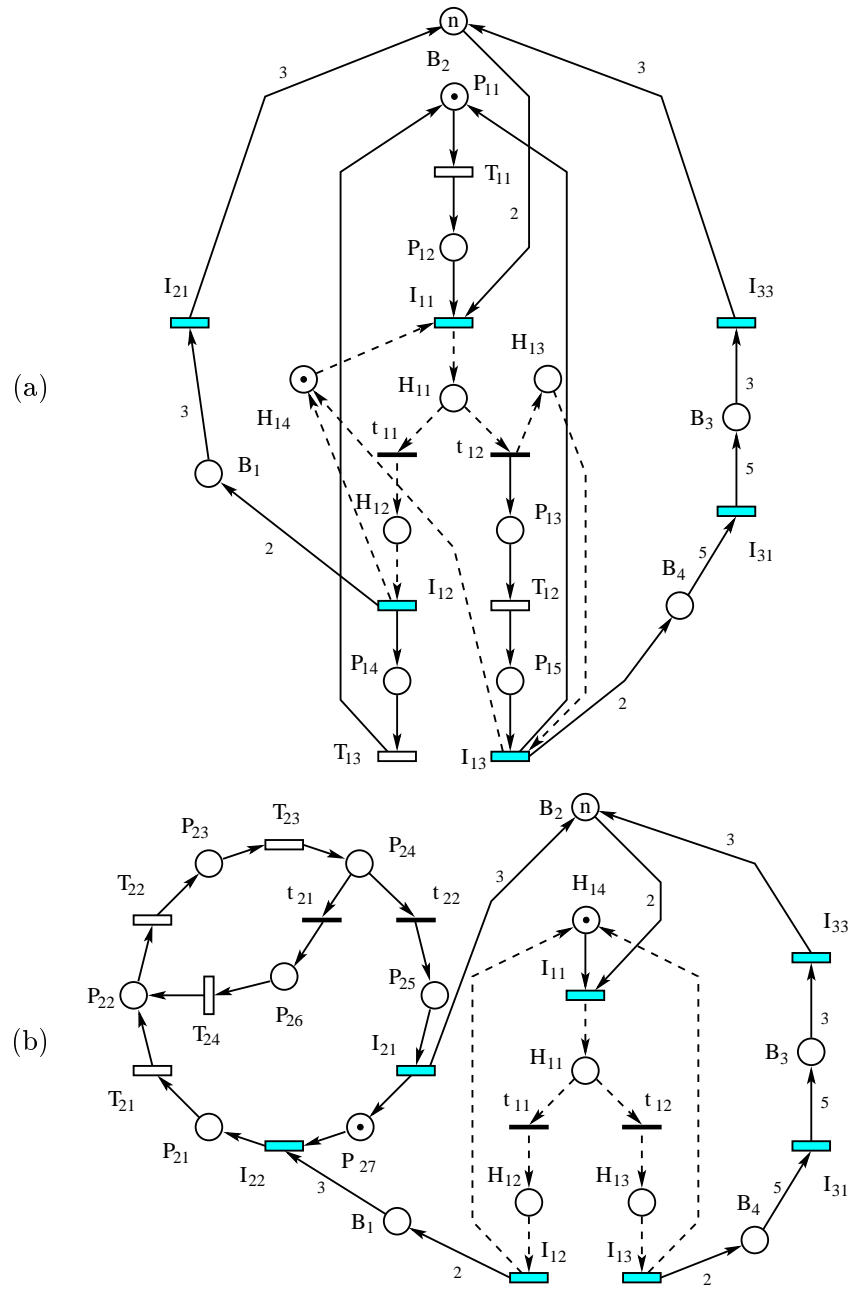


Figura 7.19: (a) \mathcal{LS}_1 y (b) \mathcal{LS}_2 de la figura 7.13. (reducción total).

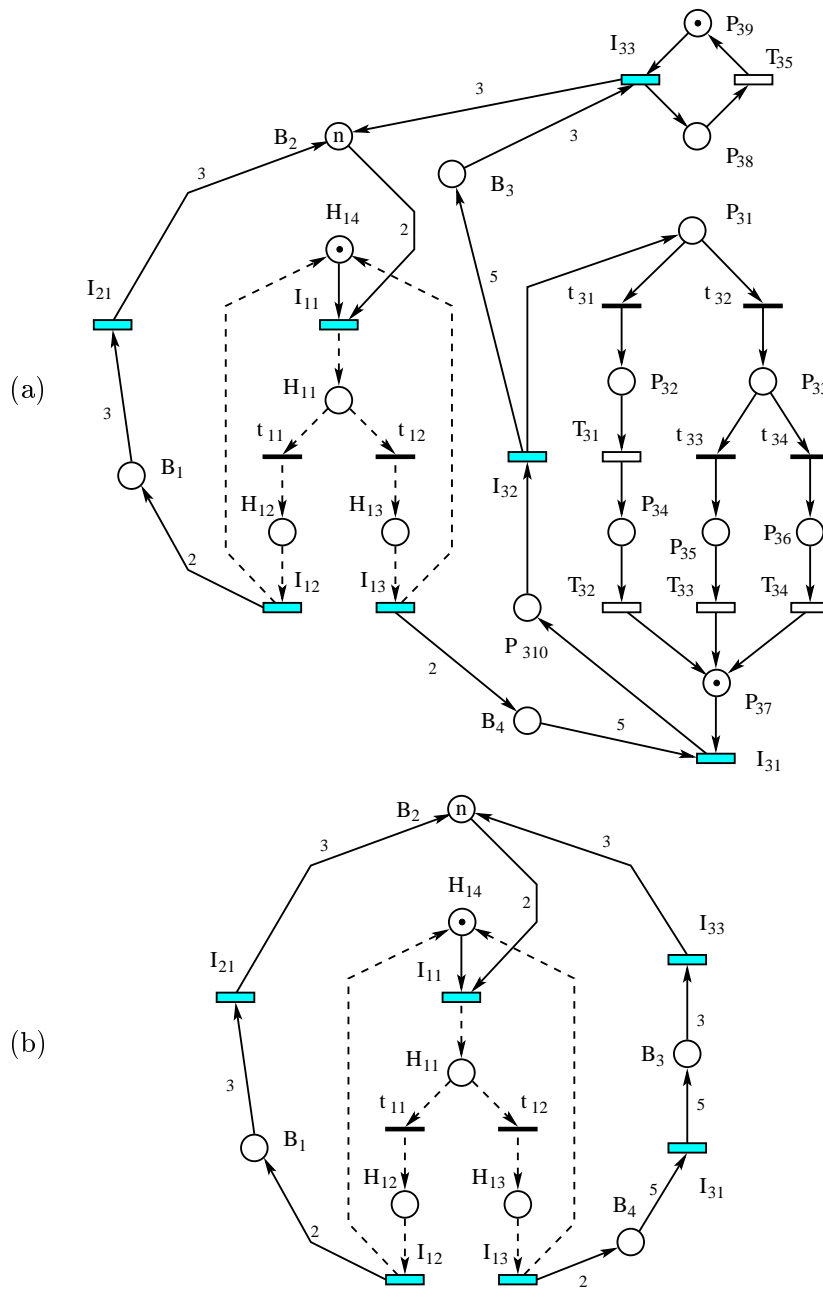


Figura 7.20: (a) \mathcal{LS}_3 y (b) \mathcal{BS} de la figura 7.13. (reducción total).

En la última fila de la tabla aparece a la izquierda el throughput exacto de la transición de referencia en la red original y a la derecha el error de la aproximación. En este caso se observa que se alcanza la convergencia en 4 iteraciones, obteniendo una aproximación de throughput de 0.292259 por lo que el error en la aproximación final es del -0.74% . Por lo tanto ha aumentado el error de la aproximación respecto al mismo caso de la técnica anterior. Este hecho se debe a la mayor reducción de las subredes en los sistemas agregados. La ventaja es que al operar con espacios de estados más pequeños, el algoritmo iterativo acaba antes. El precio a pagar es la ligera pérdida de precisión.

El resto de experimentos con otros valores de n se recogen en la siguiente tabla:

n	Marcados alcanzables					$\mathcal{X}(T_{21})$	$\mathcal{X}(T_{21})$	% Error
	Original	\mathcal{LS}_1	\mathcal{LS}_2	\mathcal{LS}_3	\mathcal{BS}	Exacto	Aprox.	
10	40092	1296	1776	6352	616	0.247962	0.241717	-2.519
11	55608	1689	2469	8348	804	0.262604	0.257078	-2.104
12	74868	2154	3327	10734	1027	0.272949	0.268345	-1.687
13	98304	2697	4368	13546	1288	0.280018	0.276172	-1.373
14	126348	3324	5610	16820	1590	0.285181	0.281889	-1.154
15	159432	4041	7071	20592	1936	0.289048	0.286198	-0.986
16	197988	4854	8769	24898	2329	0.292068	0.289585	-0.850
17	242448	5769	10722	29774	2772	0.294443	0.292259	-0.742
18	293244	6792	12948	35256	3268	0.296345	0.294418	-0.650
19	350808	7929	15465	41380	3820	0.297880	0.296173	-0.573
20	415572	9186	18291	48182	4431	0.299131	0.297615	-0.507
21	487968	10569	21444	55698	5104	0.300156	0.298807	-0.449
22	568428	12084	24942	63964	5842	0.301001	0.299799	-0.399
23	657384	13737	28803	73016	6648	0.301700	0.300630	-0.355
24	755268	15534	33045	82890	7525	0.302282	0.301328	-0.316
25	862512	17481	37686	93622	8476	0.302767	0.301917	-0.281

Tabla 7.20: Aproximaciones para diversos valores de n (reducción total).

De la misma forma que con la técnica de reducción parcial, al aumentar el valor del parámetro n , la calidad de la aproximación mejora. En general, puede observarse que los sistemas agregados obtenidos con la técnica de reducción total de SM's tienen espacios de estados más pequeños que los obtenidos con la técnica de reducción parcial de SM's. En cuanto a las aproximaciones obtenidas son ligeramente peores con esta última técnica. En cualquier caso, los errores obtenidos con las dos técnicas son inferiores al 3% en esta red, lo que es aceptable para una técnica de aproximación.

El único paso que falta para acabar el estudio de este ejemplo consiste

en aplicar las técnicas de descomposición del capítulo 6 y comparar los resultados que se obtengan con los de las técnicas anteriores. Por lo tanto, se van a realizar ahora los mismos experimentos anteriores con las tres técnicas generales para poder comparar los resultados. En este punto conviene adelantar que los sistemas agregados que se obtienen con cualquiera de las tres técnicas generales en este ejemplo son los mismos, por lo que es suficiente aplicar una de ellas, por ejemplo la primera (sección 6.3). Las aproximaciones que se obtienen con las otras dos técnicas generales son exactamente las mismas ya que operan con los mismos sistemas agregados y con el mismo algoritmo numérico.

La red de partida es la misma de la figura 7.13, así como el corte elegido. Para aplicar la primera técnica general es necesario calcular un conjunto de lugares implícitos de acuerdo con la técnica desarrollada en [CDS99] (desarrollada en la sección 6.2.2). Estos lugares implícitos resumen el comportamiento interno de las subredes producto del corte. Sin entrar en detalles sobre su cálculo, el algoritmo 6.6 da como resultado que son necesarios 6 lugares estructuralmente implícitos marcados (MSIP's), dos de ellos ya presentes en la red original. Por lo tanto sólo son necesarios 4 lugares MSIP para resumir el comportamiento de las subredes. Por medio del algoritmo 2.60 se calculan los marcados iniciales de cada lugar para hacerlo implícito. Si se añaden a la red original los lugares implícitos calculados se obtiene el sistema extendido \mathcal{ES} de la figura 7.21. Notar que es muy parecido al sistema extendido obtenido con la técnica de reducción parcial (figura 7.15). El único cambio consiste en la reducción de la SM que modela la primera máquina del sistema de fabricación. El resto de SM's resultan reducidas de la misma forma.

A partir del \mathcal{ES} es inmediato construir los sistemas agregados. En las figuras 7.22.a, 7.22.b, 7.23.a y 7.23.b se pueden observar el \mathcal{LS}_1 , \mathcal{LS}_2 , \mathcal{LS}_3 y \mathcal{BS} correspondientes al corte realizado en la red original empleando la primera técnica general de reducción. Para $n = 17$, el \mathcal{LS}_1 tiene 12348 estados alcanzables, el \mathcal{LS}_2 10072, el \mathcal{LS}_3 33072 y el \mathcal{BS} 4172. Se puede observar que para el mismo valor de $n = 17$ se obtienen sistemas agregados con tamaños de espacios de estados menores o iguales que los obtenidos con la técnica de reducción parcial y mayores o iguales (excepto \mathcal{LS}_2) que los de la técnica de reducción total.

El hecho de que las tres técnicas generales produzcan los mismos sistemas agregados quiere decir que estos sistemas agregados no contienen ningún marcado alcanzable espurio (ver sección 6.5 con el desarrollo de la segunda

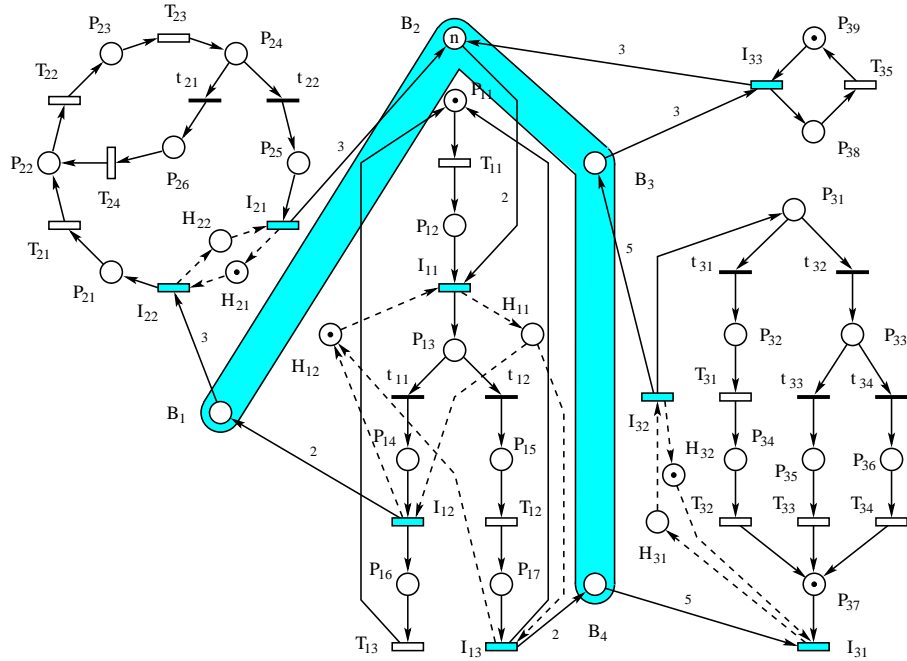


Figura 7.21: \mathcal{ES} del DSSP de la figura 7.13 (técnicas generales).

técnica general) y que tampoco hay agregaciones inadecuadas de estados no conectados por transiciones internas (ver sección 6.6 con el desarrollo de la tercera técnica general).

Para aproximar el throughput de las transiciones del DSSP original basta aplicar el algoritmo iterativo 6.43 a los nuevos sistemas agregados. Se han realizado los mismos experimentos que con la técnica anterior.

\mathcal{LS}_1		\mathcal{LS}_2		\mathcal{LS}_3	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ	$\mathcal{X}(I_{31})$	λ
0.610332	1.000769	0.290211	1.060939	0.058042	1.157040
0.588627	0.996831	0.294181	1.059102	0.058836	1.157040
0.588410	0.996754	0.294204	1.059102	0.058841	1.157040
0.588410	0.996754	0.294204	1.059102	0.058841	1.157040
$\mathcal{X}(I_{21})$ exacto: 0.294443				Error: -0.08%	

Tabla 7.21: Aproximación de \mathcal{X} para Fig. 7.13 ($n = 17$, técnicas generales).

En la tabla 7.21 aparecen los resultados de la ejecución del algoritmo

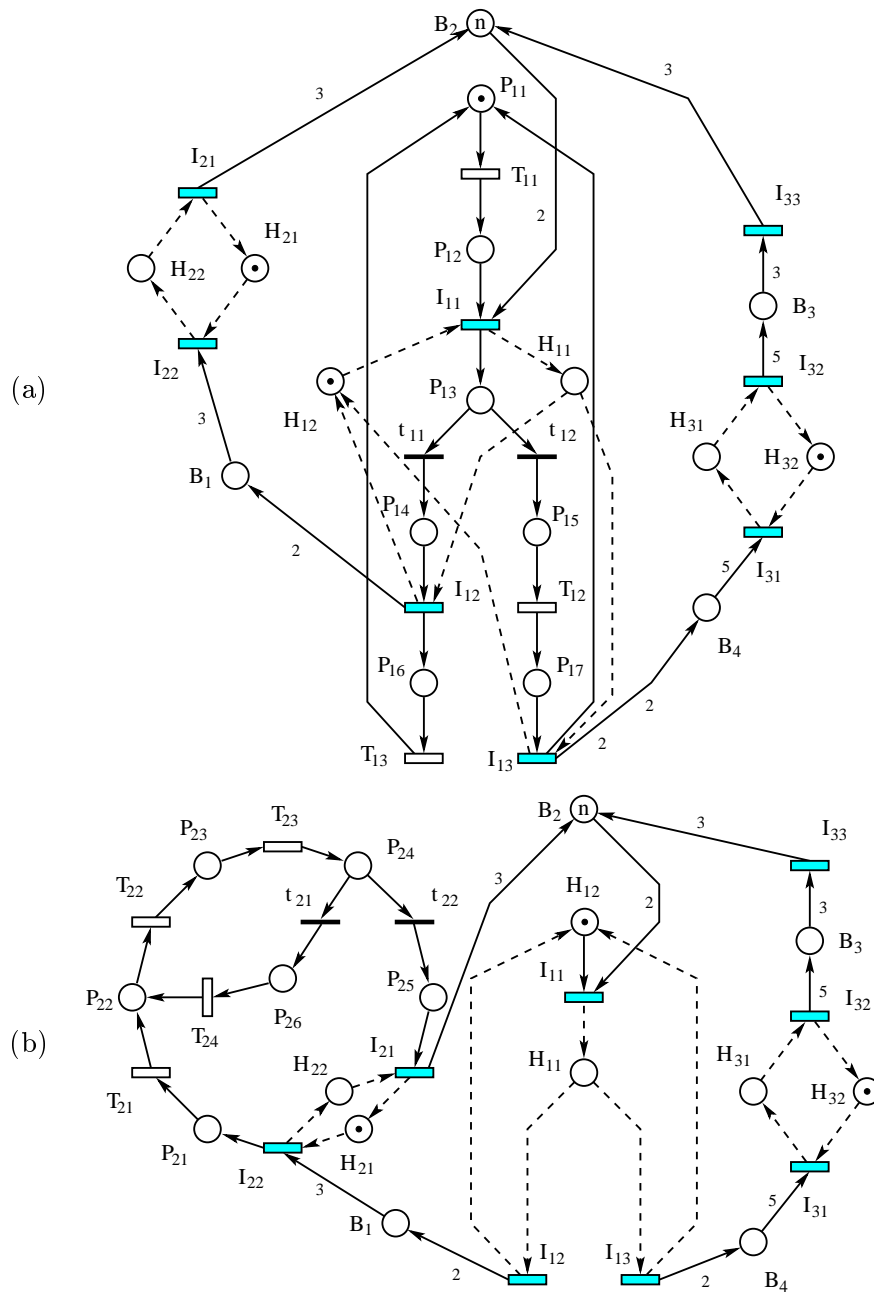


Figura 7.22: (a) \mathcal{LS}_1 y (b) \mathcal{LS}_2 de la figura 7.13. (técnicas generales).

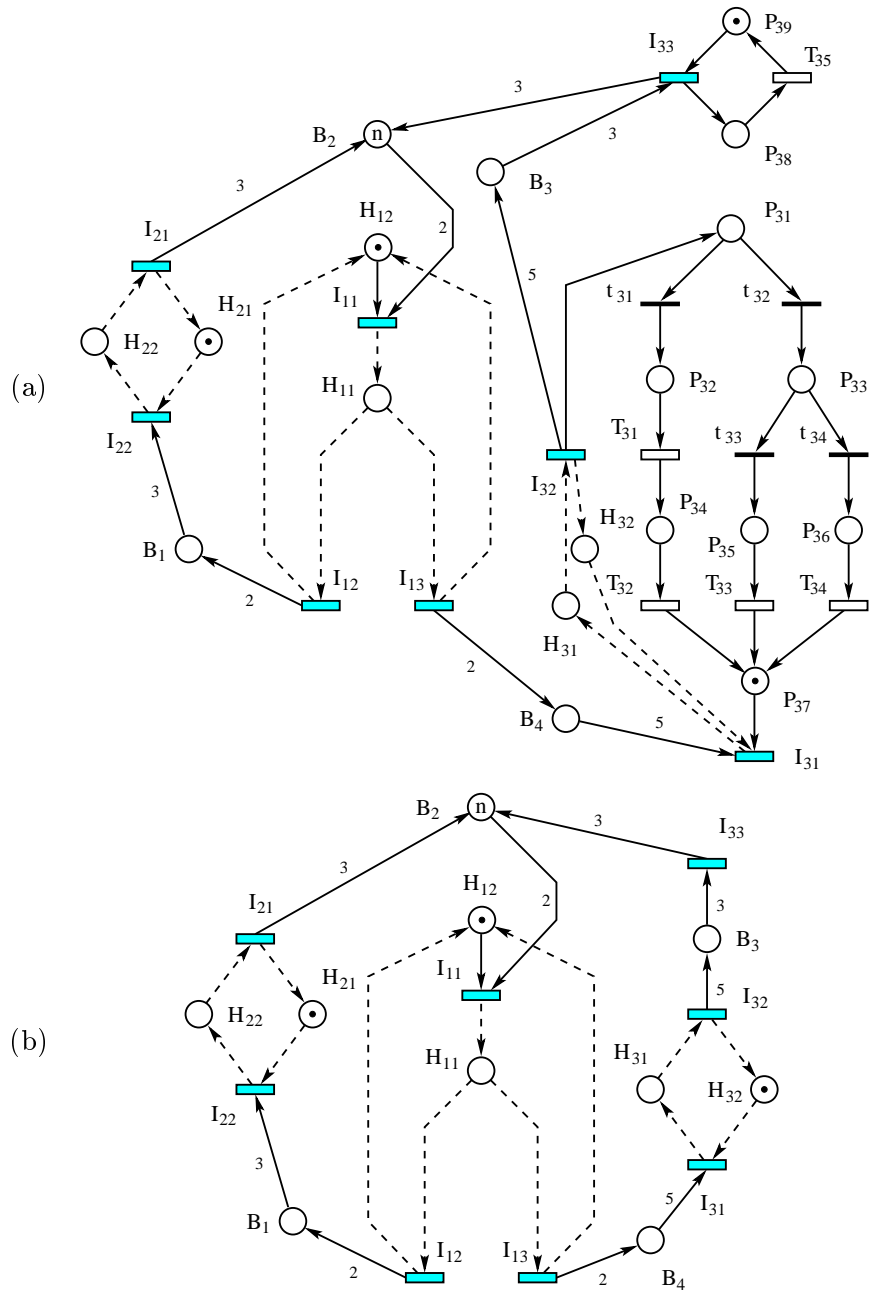


Figura 7.23: (a) \mathcal{LS}_3 y (b) \mathcal{BS} de la figura 7.13. (técnicas generales).

iterativo en el caso de $n = 17$. En este caso, el throughput exacto de la transición I_{21} de la red original (tasa de producción del producto de tipo A) es 0.294443. La leyenda de la tabla es la misma que con las técnicas anteriores. Para cada \mathcal{LS}_i se da el throughput de la transición de interfaz de referencia (I_{i1}) y el factor de escala encontrado en el estudio del \mathcal{BS} para actualizar las tasas de disparo de las transiciones de interfaz de \mathcal{LS}_i . En cada fila de la tabla aparecen los resultados de cada iteración del algoritmo, hasta alcanzar la convergencia. En la última fila de la tabla aparece a la izquierda el throughput exacto de la transición de referencia en la red original y a la derecha el error de la aproximación. En este caso se observa que se alcanza la convergencia en 4 iteraciones, obteniendo una aproximación de throughput de 0.294204 por lo que el error en la aproximación final es del -0.08% . Por lo tanto, las técnicas generales ofrecen en este ejemplo menor error en la aproximación que las técnicas anteriores de reducción parcial o total de SM's. Las diferencias numéricas se deben a las diferencias en la reducción de las subredes, que se traducen en diferencias en las estimaciones de las utilidades de las transiciones de interfaz de cada subred. Por otro lado, la mejora en la aproximación no se ha hecho a costa de aumentar significativamente el tamaño de los espacios de estados de los sistemas agregados, por lo que se puede concluir que, en este caso, los mejores resultados se obtienen con las técnicas generales.

El resto de experimentos con otros valores de n se recogen en la tabla 7.22

n	Marcados alcanzables					$\mathcal{X}(T_{21})$	$\mathcal{X}(T_{21})$	% Error
	Original	\mathcal{LS}_1	\mathcal{LS}_2	\mathcal{LS}_3	\mathcal{BS}	Exacto	Aprox.	
10	40092	2058	1462	6444	714	0.247962	0.245970	-0.803
11	55608	2832	2068	8616	976	0.262604	0.261075	-0.582
12	74868	3798	2842	11252	1302	0.272949	0.271908	-0.381
13	98304	4980	3808	14400	1700	0.280018	0.279323	-0.248
14	126348	6402	4990	18108	2178	0.285181	0.284692	-0.171
15	159432	8088	6412	22424	2744	0.289048	0.288678	-0.128
16	197988	10062	8098	27396	3406	0.292068	0.291777	-0.100
17	242448	12348	10072	33072	4172	0.294443	0.294204	-0.081
18	293244	14970	12358	39500	5050	0.296345	0.296143	-0.068
19	350808	17952	14980	46728	6048	0.297880	0.297707	-0.058
20	415572	21318	17962	54804	7174	0.299131	0.298978	-0.051
21	487968	25092	21328	63776	8436	0.300156	0.300024	-0.044
22	568428	29298	25102	73692	9842	0.301001	0.300882	-0.040
23	657384	33960	29308	84600	11400	0.301700	0.301595	-0.035
24	755268	39102	33970	96548	13118	0.302282	0.302190	-0.030
25	862512	44748	39112	109584	15004	0.302767	0.302683	-0.028

Tabla 7.22: Aproximaciones para diversos valores de n (técnicas generales).

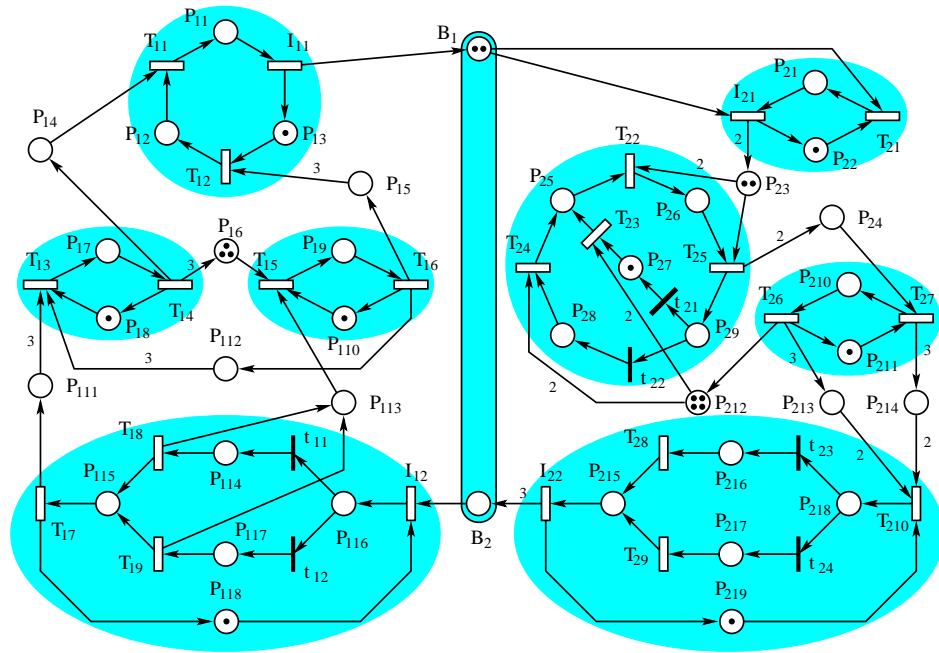


Figura 7.24: DSSP con SM's totalmente reducibles.

De la misma forma que con las técnicas anteriores, al aumentar el valor del parámetro n , la calidad de la aproximación mejora. En general puede observarse que los sistemas agregados obtenidos con las técnicas generales de reducción tienen espacios de estados de tamaño intermedio entre los obtenidos con las técnicas anteriores. En cuanto a las aproximaciones obtenidas son ligeramente mejores que las obtenidas con las técnicas anteriores.

El ejemplo que se ha desarrollado hasta ahora en esta sección correspondía a un DSSP en el que las SM's eran lo suficientemente grandes como para obtener una reducción significativa del espacio de estados por medio de la técnica de reducción parcial de SM's. El siguiente ejemplo que se va a desarrollar en lo que queda de sección es el de un DSSP compuesto por pequeñas SM's muy conectadas entre sí. El ejemplo está tomado de [PJCS96a] y es el de la figura 7.24.

Es un DSSP compuesto por 8 SM's (cada una incluida en una elipse gris) y muchos canales conectando entre sí las SM's. La mayor parte de las transiciones temporizadas tienen algún canal de entrada o salida, de forma

Tasa	Transiciones
1.0	$t_{11}, t_{12}, t_{21}, t_{22}, t_{23}, t_{24}$
2.0	$I_{12}, I_{22}, T_{14}, T_{21}, T_{25}$
3.0	$T_{15}, T_{17}, T_{23}, T_{24}, T_{29}$
4.0	$I_{21}, T_{13}, T_{18}, T_{210}$
5.0	$T_{19}, T_{22}, T_{26}, T_{27}$
8.0	T_{16}, T_{28}
10.0	I_{11}, T_{11}, T_{12}

Tabla 7.23: Tasas de las transiciones del DSSP de la Fig. 7.24.

que la técnica de reducción parcial de SM's (sección 5.2), al mantener este tipo de transiciones, no conseguiría una reducción significativa del espacio de estados. Por lo tanto, en un ejemplo como éste no tiene sentido aplicar la técnica de reducción parcial de SM's. Si que se pueden aplicar el resto de técnicas disponibles para DSSP's, esto es, la técnica de reducción total (sección 5.3) y las técnicas generales de reducción (capítulo 6). Por lo tanto se aplicarán estas técnicas y se obtendrá una nueva comparación entre ellas.

Con cada técnica se realizarán tres experimentos; uno con todas las tasas de disparo de todas las transiciones a 1, otro con tasas de diferentes órdenes de magnitud entre las transiciones de las distintas subredes y otro con tasas elegidas al azar. Estas tasas están en la tabla 7.23.

El grafo de alcanzabilidad de esta red tiene 57288 estados. Se ha elegido como corte el formado por los canales B_1 y B_2 , efectuando una descomposición de la red original en 2 subredes (\mathcal{N}_1 y \mathcal{N}_2). La subred \mathcal{N}_1 (la de la parte izquierda) está formada por las transiciones internas temporizadas $\{T_{1i}\}_{i=1}^9$, las transiciones internas inmediatas t_{11} y t_{12} , las transiciones de interfaz I_{11} e I_{12} y los lugares internos $\{P_{1i}\}_{i=1}^8$. La subred \mathcal{N}_2 (la de la parte derecha) está formada por el resto de los nodos (descontando también los canales de corte).

La primera técnica que se va a aplicar es la de reducción total de SM's. Como en el ejemplo anterior ya se ha explicado en detalle cómo funciona la técnica, aquí se va a dar directamente el resultado de la agregación. En este caso todas las SM's son totalmente reducibles a una sola transición. Si se testean las propiedades **P1** y **P2** de las propiedades 5.13 y 5.15 se puede observar que la SM generada por las transiciones $T_{22}, T_{23}, T_{24}, T_{25}, t_{21}$ y t_{22} en principio no cumple la propiedad **P2** debido al marcado inicial que tiene.

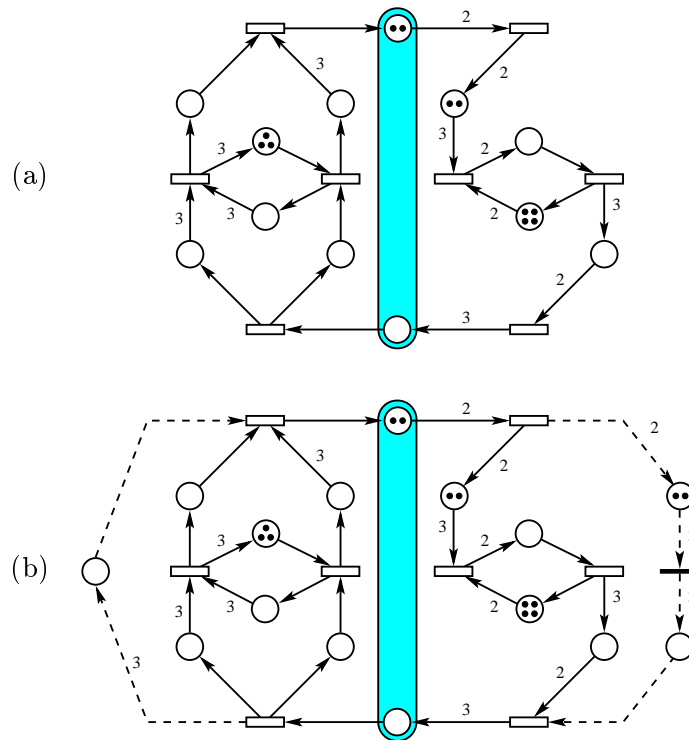


Figura 7.25: (a) Reducción total de SM's de la figura 7.24 y (b) \mathcal{ES} del WTS de (a) (técnica de WTS's).

Sin embargo, si el marcado inicial estuviera en P_{29} si que la cumpliría. Como este marcado también es alcanzable en el DSSP (sin modificar el marcado inicial del resto de lugares) se puede concluir que esta SM también es totalmente reducible.

El resultado de agregar las 8 SM's a una única transición produce la red de la figura 7.25.a. Este esqueleto tiene estructura de WTS por lo que se puede aplicar ahora la técnica de reducción de WTS's del capítulo 4 para reducir todavía en mayor grado las subredes del DSSP. Se ha mantenido el corte inducido por el DSSP en el WTS para saber qué subredes del WTS es necesario reducir. Tampoco se detallará el cómputo de los nodos necesarios para reducir estas subredes del WTS porque ya se ha explicado en el ejemplo de la sección 7.2. Estos nodos se pueden ver en la figura 7.25.b. (son los que

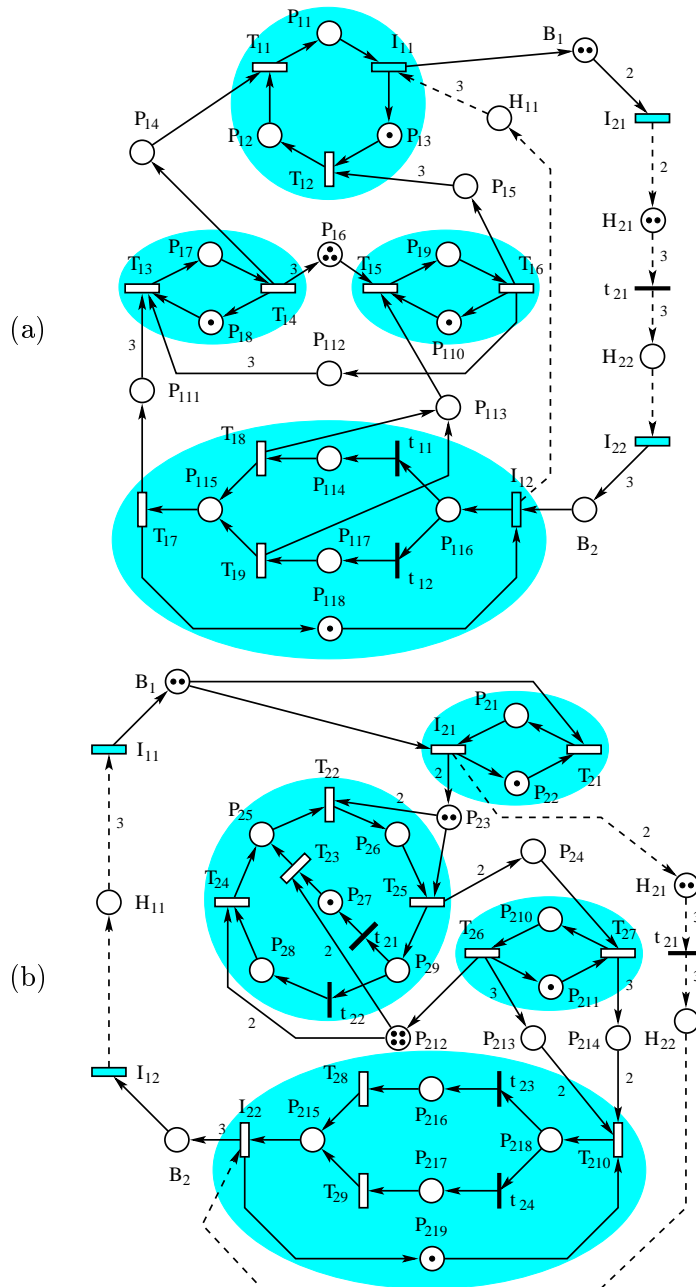


Figura 7.26: (a) \mathcal{LS}_1 y (b) \mathcal{LS}_2 de la figura 7.24. (reducción total).

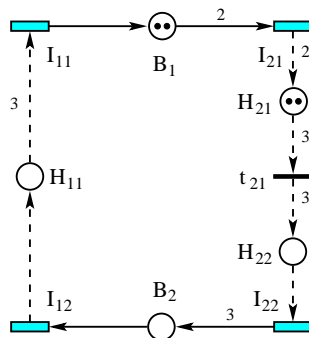


Figura 7.27: \mathcal{BS} de la figura 7.24 (reducción total).

tienen arcos de entrada y salida discontinuos).

Por lo tanto se está en condiciones ya de generar los sistemas agregados correspondientes a la descomposición efectuada en el DSSP de partida. En este caso no tiene sentido hablar de sistema extendido ya que hay ciertas transiciones de interfaz que desaparecen en el proceso (ver figura 7.25). Por lo tanto se construirán directamente los sistemas agregados. Esto no afecta al algoritmo numérico de aproximación ya que éste opera con los sistemas agregados y no con el sistema extendido.

En las figuras 7.26.a, 7.26.b y 7.27. se pueden observar el \mathcal{LS}_1 , \mathcal{LS}_2 y \mathcal{BS} correspondientes al corte realizado en la red original empleando la técnica de reducción total de SM's junto con la de reducción de WTS's. El \mathcal{LS}_1 tiene 6854 estados alcanzables, el \mathcal{LS}_2 1885, y el \mathcal{BS} 133.

Para aproximar el throughput de las transiciones del DSSP original basta aplicar el algoritmo iterativo 5.23 a los nuevos sistemas agregados. Ahora se van a dar los resultados que se han obtenido para los tres experimentos que se querían realizar.

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.090540	1.139683	0.039450	1.401732
0.079954	1.142390	0.040013	1.396328
0.080022	1.142410	0.040011	1.396328
0.080022	1.142410	0.040011	1.396328
$\mathcal{X}(I_{11})$ exac: 0.079497		Err: +0.66%	

Tabla 7.24: Aproximación de \mathcal{X} para Fig. 7.24. Tasas 1.0.

En la tabla 7.24 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas 1.0 para todas las transiciones de la red original. En este caso, el throughput exacto de la transición I_{11} de la red original es 0.079497. La leyenda de la tabla es la misma que en el ejemplo anterior. Se observa que en este caso el algoritmo converge en 4 iteraciones, obteniendo una aproximación de throughput de 0.080022 por lo que el error en la aproximación final es del +0.66%.

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.077193	1.004778	0.011771	1.001127
0.023542	0.984712	0.011771	1.001170
0.023542	0.984712	0.011771	1.001170
$\mathcal{X}(I_{11})$ exac: 0.023542		Err: 0.0%	

Tabla 7.25: Aproximación de \mathcal{X} para Fig. 7.24. Tasas 100.0 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 .

En la tabla 7.25 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas de diferentes órdenes de magnitud entre las distintas subredes producto del corte. En concreto se ha puesto tasa 100 a las transiciones de la subred \mathcal{N}_1 (la de la parte izquierda) y 0.1 a las de la subred \mathcal{N}_2 (la de la parte derecha). En este caso, el throughput exacto de la transición I_{11} de la red original es 0.023542. Se observa que el algoritmo converge en 3 iteraciones, obteniendo una aproximación de throughput de 0.023542 por lo que se ha alcanzado la solución exacta. Como en anteriores ejemplos con otras clases de redes, cuando el corte separa subredes de diferentes velocidades, el algoritmo converge en menos iteraciones y la aproximación es prácticamente exacta.

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.274443	1.079778	0.124975	1.399735
0.253229	1.083789	0.126695	1.393013
0.253374	1.083789	0.126687	1.393013
0.253373	1.083789	0.126687	1.393013
$\mathcal{X}(I_{11})$ exac: 0.253419		Err: -0.02%	

Tabla 7.26: Aproximación de \mathcal{X} para Fig. 7.24. Tasas de la tabla 7.23.

En la tabla 7.26 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas elegidas al azar (las de la tabla 7.23). En este

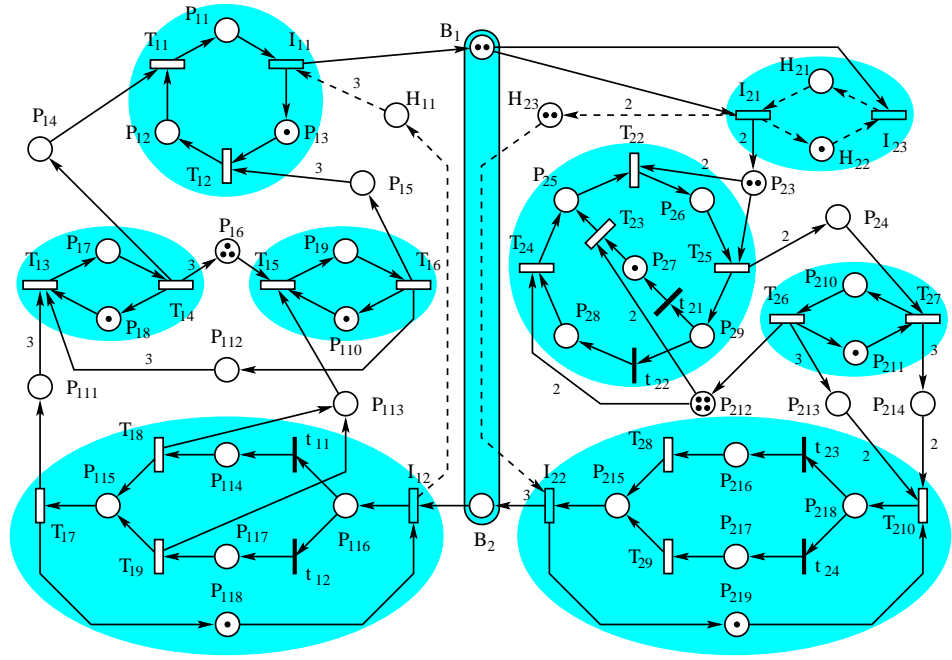


Figura 7.28: \mathcal{ES} del DSSP de la figura 7.24 (técnicas generales).

caso, el throughput exacto de la transición I_{11} de la red original es 0.253419. Se observa que el algoritmo converge en 4 iteraciones, obteniendo una aproximación de throughput de 0.253373 por lo que el error en la aproximación final es del -0.02% .

Ahora se van a comparar los resultados de estos tres experimentos con los que se obtienen empleando las técnicas generales de reducción. La red de partida es la misma de la figura 7.24, así como el corte elegido. Para aplicar la primera técnica general es necesario calcular un conjunto de lugares implícitos de acuerdo con la técnica desarrollada en [CDS99] (ver sección 6.2.2). Sin entrar en detalles sobre su cálculo, el algoritmo 6.6 da como resultado que son necesarios 4 lugares estructuralmente implícitos marcados (MSIP's), dos de ellos ya presentes en la red original. Por lo tanto, sólo son necesarios 2 lugares MSIP para resumir el comportamiento de las subredes. Por medio del algoritmo 2.60 se calculan los marcados iniciales de cada lugar para hacerlo implícito. Si se añaden a la red original los lugares implícitos calculados se obtiene el sistema extendido \mathcal{ES} de la figura 7.28.

A partir del sistema extendido es inmediato construir los sistemas agregados. En las figuras 7.29.a, 7.29.b y 7.30 se pueden observar el \mathcal{LS}_1 , \mathcal{LS}_2 y \mathcal{BS} correspondientes al corte realizado en la red original empleando las técnicas generales de reducción del capítulo 6. Ahora, dependiendo de qué técnica de general de reducción se emplee, se obtienen grafos de alcanzabilidad de distintos tamaños para los sistemas agregados. Si se emplean la primera (sección 6.3) o segunda técnica general (sección 6.5), se obtienen los mismos sistemas agregados. El \mathcal{LS}_1 tiene 6413 estados alcanzables, el \mathcal{LS}_2 1885, y el \mathcal{BS} 135. Este hecho indica que los sistemas agregados que se obtienen con estas dos técnicas son proyecciones exactas en términos de marcados alcanzables de la red original en los nodos preservados en cada sistema agregado (por las características de la segunda técnica general de descomposición). Por lo tanto, basta con aplicar una de las dos primeras técnicas generales, porque con ambas se obtendrán los mismos resultados. En cambio, si se emplea la tercera técnica general (sección 6.6), el \mathcal{LS}_1 tiene 9061 estados alcanzables, el \mathcal{LS}_2 1885, y el \mathcal{BS} 197. Este hecho indica que los sistemas agregados calculados con las otras dos técnicas generales tienen agregaciones inadecuadas de estados del sistema original no conectados por transiciones internas (por las características de la tercera técnica general de descomposición). Esto ocurre en el \mathcal{LS}_1 y \mathcal{BS} . Este hecho puede hacer que los resultados numéricos que se obtengan con las dos primeras técnicas generales sean peores que con la tercera.

Para aproximar el throughput de las transiciones del DSSP original basta aplicar el algoritmo iterativo 6.43 a los nuevos sistemas agregados. Ahora se van a dar los resultados que se han obtenido para los tres experimentos que se querían realizar con las técnicas generales 1 y 3.

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.108222	1.156320	0.039256	1.122805
0.081220	1.209201	0.041719	1.121706
0.083315	1.207406	0.041609	1.121706
0.083223	1.207486	0.041614	1.121706
0.083227	1.207458	0.041613	1.121706
$\mathcal{X}(I_{11})$ exac: 0.079497		Err: +4.69%	

Tabla 7.27: Aproximación de \mathcal{X} para Fig. 7.24. Tasas 1.0 (técnicas generales 1 y 2).

En la tabla 7.27 aparecen los resultados de la ejecución del algoritmo

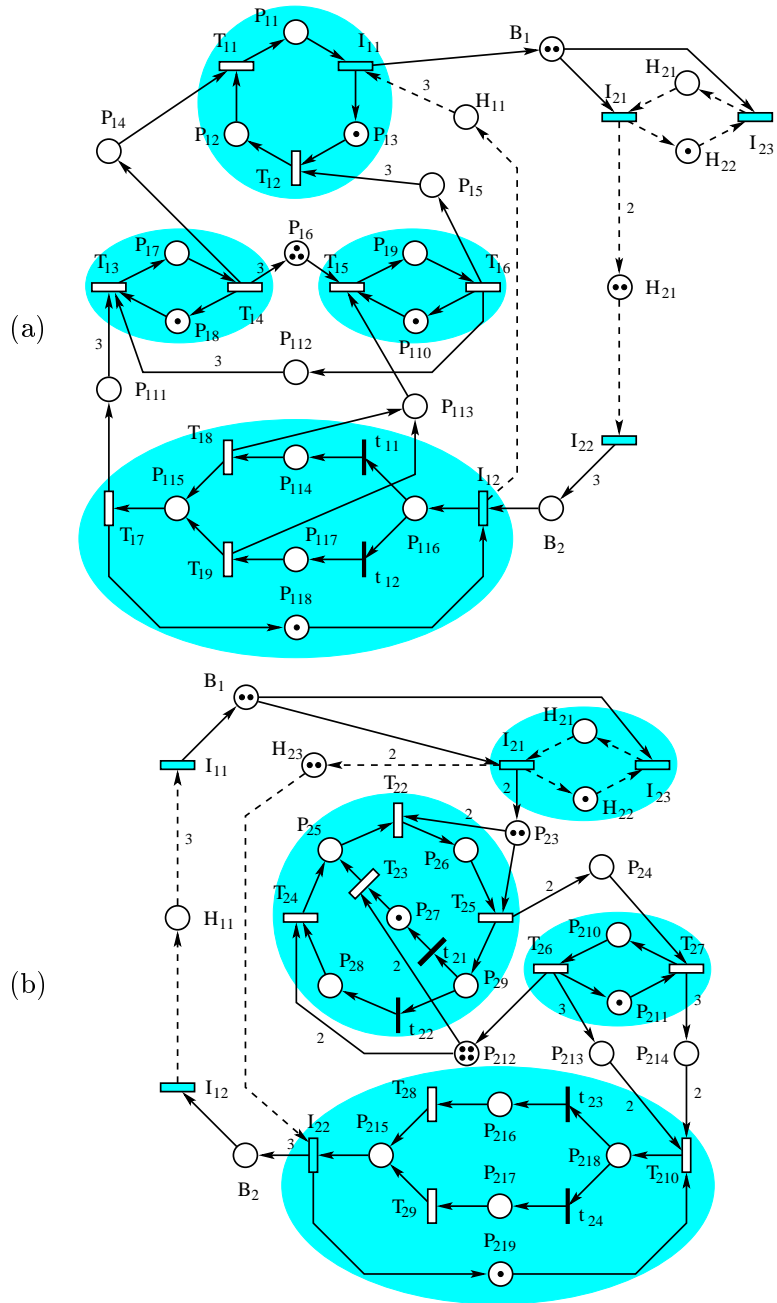


Figura 7.29: (a) \mathcal{LS}_1 y (b) \mathcal{LS}_2 de la figura 7.24. (técnicas generales).

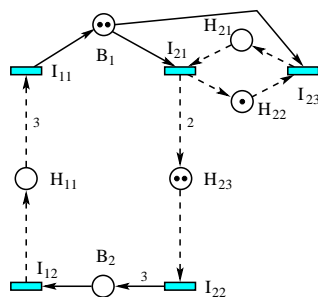


Figura 7.30: \mathcal{BS} de la figura 7.24 (técnicas generales).

iterativo en el caso de tasas 1.0 con las técnicas generales 1 ó 2. En este caso, el throughput exacto de la transición I_{11} de la red original es 0.079497. La leyenda de la tabla es la misma que en ejemplos anteriores. Se observa que en este caso el algoritmo alcanza la convergencia en 5 iteraciones, obteniendo una aproximación de throughput de 0.083227 por lo que el error en la aproximación final es del +4.69%. Este error es mucho mayor que el obtenido para el mismo ejemplo con la técnica de reducción total de SM's y es debido al gran número de agregaciones inadecuadas de estados que se producen en \mathcal{LS}_1 y \mathcal{BS} .

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.090529	1.141588	0.039503	0.890452
0.080003	1.144247	0.040028	0.892926
0.080052	1.144247	0.040026	0.892926
0.080053	1.144247	0.040026	0.892884
$\mathcal{X}(I_{11})$ exac: 0.079497		Err: +0.70%	

Tabla 7.28: Aproximación de \mathcal{X} para Fig. 7.24. Tasas 1.0 (técnica general 3).

En la tabla 7.28 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas 1.0 con la técnica general 3. En este caso se observa que se alcanza la convergencia en 4 iteraciones, obteniendo una aproximación de throughput de 0.080053 por lo que el error en la aproximación final es del +0.70%. Este error es ligeramente superior al obtenido con la técnica de reducción total de SM's y sensiblemente inferior al de las técnicas generales 1 y 2. Se debe a que se han eliminado esas agregaciones inadecuadas de estados, lo que provoca una mejora en las aproximaciones.

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.076630	1.000000	0.011771	1.006368
0.023542	1.000000	0.011771	1.006368
0.023542	1.000000	0.011771	1.006368
$\mathcal{X}(I_{11})$ exac: 0.023542		Err: 0.0%	

Tabla 7.29: Aproximación de \mathcal{X} para Fig. 7.24. Tasas 100 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 (técnicas generales 1 y 2).

En la tabla 7.29 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas de diferentes órdenes de magnitud entre las distintas subredes producto del corte con las técnicas generales 1 ó 2. En este caso se observa que se alcanza la convergencia en 3 iteraciones, obteniendo una aproximación de throughput de 0.023542 por lo que se ha alcanzado la solución exacta. Se obtiene la misma aproximación que con la técnica de reducción total de SM's.

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.066049	1.004539	0.011771	1.009952
0.023543	1.004539	0.011771	1.009974
0.023543	1.004539	0.011771	1.009974
$\mathcal{X}(I_{11})$ exac: 0.023542		Err: 0.0%	

Tabla 7.30: Aproximación de \mathcal{X} para Fig. 7.24. Tasas 100 en \mathcal{N}_1 y 0.1 en \mathcal{N}_2 (técnica general 3).

En la tabla 7.30 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas de diferentes órdenes de magnitud entre las distintas subredes producto del corte con la técnica general 3. En este caso se observa que se alcanza la convergencia en 3 iteraciones, obteniendo una aproximación de throughput de 0.023543 por lo que prácticamente se ha alcanzado la solución exacta.

En la tabla 7.31 aparecen los resultados de la ejecución del algoritmo iterativo en el caso de tasas elegidas al azar (las de la tabla 7.23) con las técnicas generales 1 ó 2. En este caso se observa que se alcanza la convergencia en 5 iteraciones, obteniendo una aproximación de throughput de 0.257688 por lo que el error en la aproximación final es del +1.68%.

En la tabla 7.32 aparecen los resultados de la ejecución del algoritmo

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.314276	1.062045	0.120225	1.125632
0.251056	1.120448	0.129329	1.121567
0.258031	1.118324	0.128818	1.121844
0.257670	1.118440	0.128845	1.121813
0.257688	1.118441	0.128844	1.121813
$\mathcal{X}(I_{11})$ exac: 0.253419		Err: +1.68%	

Tabla 7.31: Aproximación de \mathcal{X} para Fig. 7.24. Tasas de la tabla 7.23 (técnicas generales 1 y 2).

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.278646	1.082155	0.125044	0.887630
0.253964	1.088814	0.127060	0.890057
0.254101	1.088789	0.127051	0.890057
0.254100	1.088789	0.127051	0.890057
$\mathcal{X}(I_{11})$ exac: 0.253419		Err: +0.27%	

Tabla 7.32: Aproximación de \mathcal{X} para Fig. 7.24. Tasas de la tabla 7.23 (técnica general 3).

iterativo en el caso de tasas elegidas al azar (las de la tabla 7.23) con la técnica general 3. En este caso se observa que se alcanza la convergencia en 4 iteraciones, obteniendo una aproximación de throughput de 0.254100 por lo que el error en la aproximación final es del +0.27%.

Resumiendo los resultados obtenidos para este ejemplo con las distintas técnicas se puede observar lo siguiente. El tamaño del espacio de estados de los sistemas agregados es menor en las técnicas generales 1 y 2, y mayor en la técnica general 3. En cuanto a las aproximaciones obtenidas, las mejores se consiguen con la técnica de reducción total de SM's, y después por la técnica general 3 (las dos muy parecidas). Las técnicas generales 1 y 2 fallan ligeramente en este ejemplo con tasas 1 aunque se mantiene un error inferior al 5%. Las tres técnicas funcionan perfectamente en el caso de tasas de diferentes órdenes de magnitud, debido a que hay una subred (la lenta) que es la que ralentiza el funcionamiento de toda la red. Por lo tanto, el sistema de bajo nivel que contiene a la subred lenta tiene una velocidad muy parecida a la de la red original sin importar demasiado cómo se realice la reducción de la otra subred.

En la siguiente sección se van a desarrollar unos ejemplos con redes

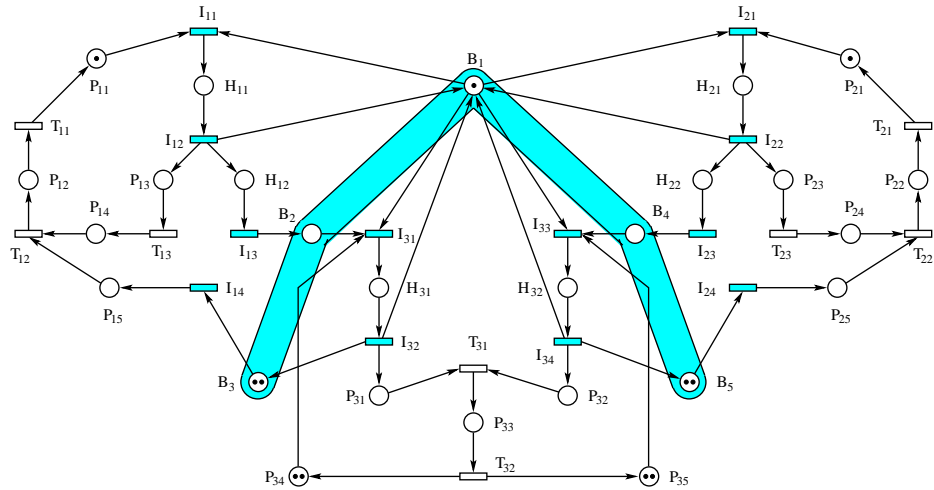


Figura 7.31: Ejemplo 1 de SAM's.

generales, no pertenecientes a las clases particulares tratadas en esta sección y en las anteriores.

7.4 Redes generales

Esta sección se dedica al desarrollo de dos ejemplos de aproximación de throughput de las transiciones de un SAM, esto es, una red de Petri general a la que se le añade una visión estructurada en módulos y canales que los conectan (ver definición 6.1). Estos ejemplos están tomados de [PJC99a] y a cada uno se le aplicarán las tres técnicas generales de descomposición y reducción del capítulo 6. Estos dos ejemplos están desarrollados para ilustrar las diferencias entre las tres técnicas generales. En el primer ejemplo se observará que las tres técnicas calculan los mismos sistemas agregados, por lo que las tres conseguirán los mismos resultados numéricos. En cambio, el segundo ejemplo produce sistemas agregados diferentes para cada una de las técnicas, por lo que se podrán ilustrar sus diferencias.

El primer ejemplo es el de la figura 7.31. Es un SAM compuesto por tres módulos conectados entre sí por medio de 5 canales. Cada módulo \mathcal{S}_i con $i = 1, 2, 3$ está compuesto por los lugares de etiquetas P_{ij} y H_{ij} y las transiciones de etiquetas I_{ij} y T_{ij} . Se sigue utilizando la misma notación de

Tasa	Transiciones
1.0	$I_{32}, I_{34}, T_{11}, T_{12}, T_{21}, T_{22}$
2.0	$I_{11}, I_{14}, I_{21}, I_{24}, I_{31}, I_{33}, T_{13}, T_{23}$
3.0	I_{12}, I_{22}
4.0	T_{31}, T_{32}
5.0	I_{13}, I_{23}

Tabla 7.33: Tasas de las transiciones del SAM de la Fig. 7.31.

todos los ejemplos anteriores; etiquetas B_j para los canales, etiquetas P_{ij} para lugares internos de los módulos, etiquetas H_{ij} para lugares internos de los módulos que luego resultan ser necesarios para la reducción de las mismas, etiquetas I_{ij} para las transiciones de interfaz y etiquetas T_{ij} para las transiciones internas de los módulos.

Por lo que respecta a la evolución temporal de la red, las tasas de disparo de las transiciones aparecen en la tabla 7.33.

El grafo de alcanzabilidad de esta red tiene 94080 estados. En un SAM la descomposición viene inducida por su visión estructurada, por lo que en este caso se ha efectuado una descomposición del sistema original en sus 3 módulos (\mathcal{S}_1 , \mathcal{S}_2 y \mathcal{S}_3).

El siguiente paso consiste en calcular el conjunto de lugares implícitos que resumen el comportamiento de cada módulo. Para ello primero hay que aplicar la relación de equivalencia de la definición 6.2. Esta relación está definida sobre el conjunto de lugares internos de un módulo. Dos lugares internos de un módulo están relacionados si y sólo si existe un camino no dirigido de arcos que los une sin pasar por ninguna transición de interfaz. De esta forma, para el módulo \mathcal{S}_1 quedan tres clases de equivalencia. Una está formada por el lugar H_{11} , otra por el lugar H_{12} y la última formada por los lugares $\{P_{1i}\}_{i=1}^5$. Por simetría, en el módulo \mathcal{S}_2 se obtienen otras tres clases de equivalencia, una formada por el lugar H_{21} , otra por el lugar H_{22} y otra por los lugares $\{P_{2i}\}_{i=1}^5$. Por último, para el módulo \mathcal{S}_3 se obtienen otras tres clases de equivalencia, una formada por el lugar H_{31} , otra por el lugar H_{32} y otra por los lugares $\{P_{3i}\}_{i=1}^5$.

Una vez obtenidas las clases de equivalencia de lugares internos de los módulos, se calcula un conjunto de lugares que resumen el comportamiento de la subred generada por los lugares de cada clase de equivalencia. Este cálculo se realiza por medio del algoritmo 6.6. Como es un algoritmo ya

utilizado por otros autores (ver [CDS99]) no se va a detallar el cálculo, sino que simplemente se darán los resultados. En las clases de equivalencia formadas por un único lugar, el lugar debe conectar dos transiciones de interfaz, por lo que es un canal y el mismo constituye el resumen. Para clases de equivalencia con más de un lugar pueden necesitarse varios lugares. Por ejemplo, para resumir la subred generada por $\{P_{1i}\}_{i=1}^5$ son necesarios dos lugares, uno de I_{12} a I_{11} y otro de I_{14} a I_{11} . Por simetría, para resumir la subred generada por $\{P_{2i}\}_{i=1}^5$ son necesarios dos lugares, uno de I_{22} a I_{21} y otro de I_{24} a I_{21} . Por último, para resumir la subred generada por $\{P_{3i}\}_{i=1}^5$ son necesarios cuatro lugares que conectan I_{32} con I_{31} , I_{34} con I_{33} , I_{32} con I_{33} e I_{34} con I_{31} . En total son necesarios 14 lugares, seis de ellos ya presentes en la red inicial, por lo que es necesario añadir 8 lugares.

Todos estos lugares son estructuralmente implícitos marcados (MSIP's) por el algoritmo utilizado para su cálculo. Lo que falta es calcular el marcado inicial que deben tener para que sean implícitos al añadirlos a la red original. Este cálculo se realiza con el algoritmo 2.60.

Si se añaden a la red original los lugares implícitos calculados se obtiene el sistema extendido \mathcal{ES} de la figura 7.32. Los lugares cuyos arcos de entrada y salida son discontinuos son los lugares implícitos que resumen el comportamiento de los distintos módulos, junto con el marcado inicial que los hace implícitos.

A partir del \mathcal{ES} es inmediato construir los sistemas agregados. En las figuras 7.33.a, 7.33.b, 7.33.c y 7.34 se pueden observar el \mathcal{LS}_1 , \mathcal{LS}_2 , \mathcal{LS}_3 y \mathcal{BS} correspondientes al SAM original.

Como se ha observado en el capítulo 6, los sistemas agregados así contruidos pueden tener grafos de alcanzabilidad cuyo funcionamiento no se corresponde con el del sistema inicial. Pueden aparecer marcados alcanzables y secuencias de disparo espurias que eventualmente pueden hacer no ergódicos a los sistemas agregados, o si no ocurre esto, hacer que el funcionamiento de los sistemas agregados difiera considerablemente del del sistema inicial. Por ello se han desarrollado tres técnicas diferentes de reducción, para minimizar estos efectos. La primera técnica general (desarrollada en la sección 6.3) asegura la ergodicidad de los sistemas agregados, eliminando las secuencias de disparo espurias que lleven componentes fuertemente conexas de su grafo de alcanzabilidad distintas de la del marcado inicial. La segunda técnica general (desarrollada en la sección 6.5) elimina de los grafos de alcanzabilidad de los sistemas agregados todos los marcados alcanzables que no se corresponden con proyecciones de estados alcanzables del sistema original

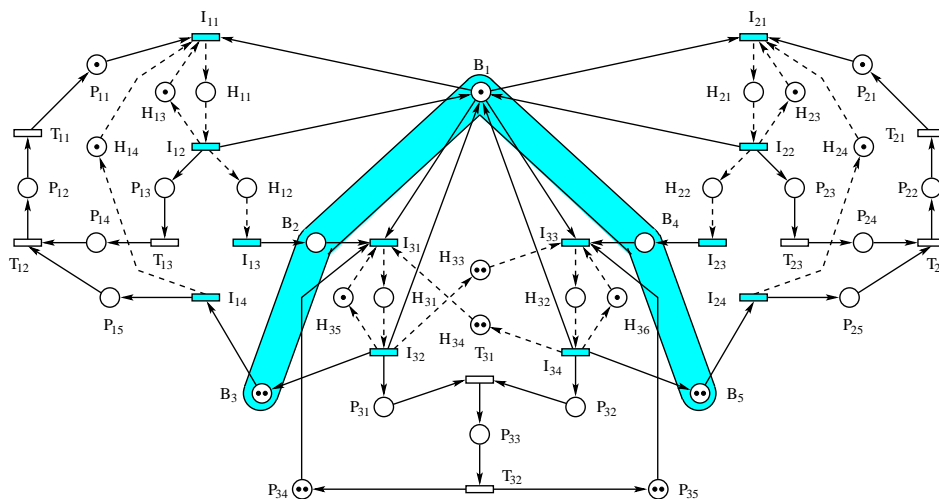


Figura 7.32: \mathcal{ES} del SAM de la figura 7.31.

sobre los nodos preservados en el sistema agregado. Por lo tanto, se puede asegurar que los sistemas agregados calculados por esta segunda técnica son proyecciones exactas en términos de marcados alcanzables del sistema original sobre los nodos preservados en cada sistema agregado. Por último, la tercera técnica general (desarrollada en la sección 6.6) elimina agregaciones inadecuadas de estados de la red original que no están conectados por transiciones internas, ya que estas agregaciones pueden eventualmente llevar a aproximaciones muy pobres.

En el ejemplo que se está desarrollando afortunadamente no aparece ninguno de estos efectos no deseados. Los sistemas agregados son vivos, limitados, reversibles y no tienen ningún estado alcanzable espurio. Esto se puede demostrar aplicando las tres técnicas de reducción y observando que las tres producen los mismos grafos de alcanzabilidad asociados a los sistemas agregados. En concreto, los \mathcal{LS}_1 y \mathcal{LS}_2 tienen 14640 estados alcanzables, el \mathcal{LS}_3 14400, y el \mathcal{BS} 5600. Si las tres técnicas de reducción producen los mismos sistemas agregados, se puede concluir que estos no tienen ningún marcado alcanzable espurio (por la segunda técnica) ni agregaciones inadecuadas de estados del sistema inicial no conectados por transiciones internas (por la tercera técnica).

Para aproximar el throughput de las transiciones del SAM original basta

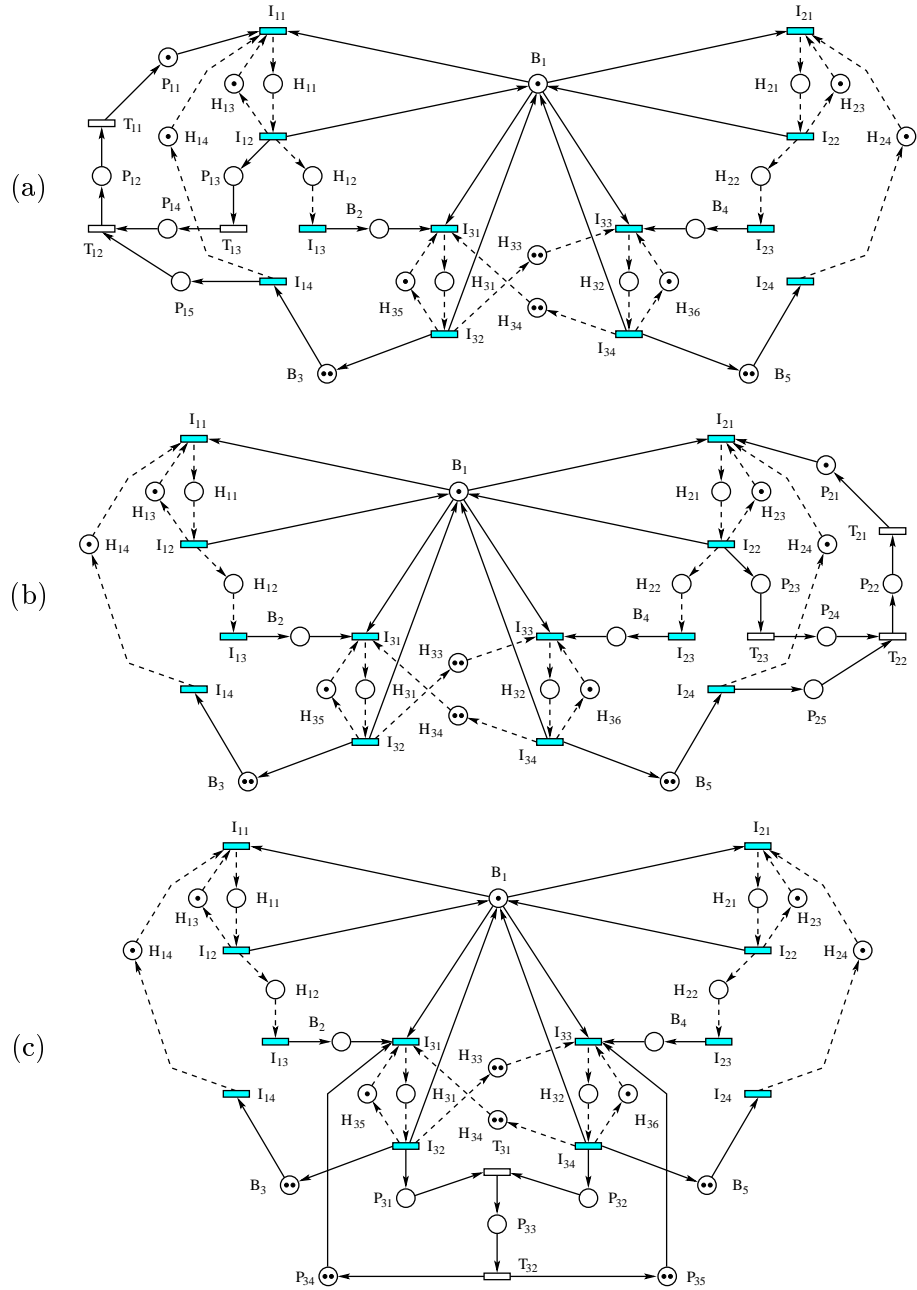


Figura 7.33: (a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{LS}_3 del SAM de la figura 7.31.

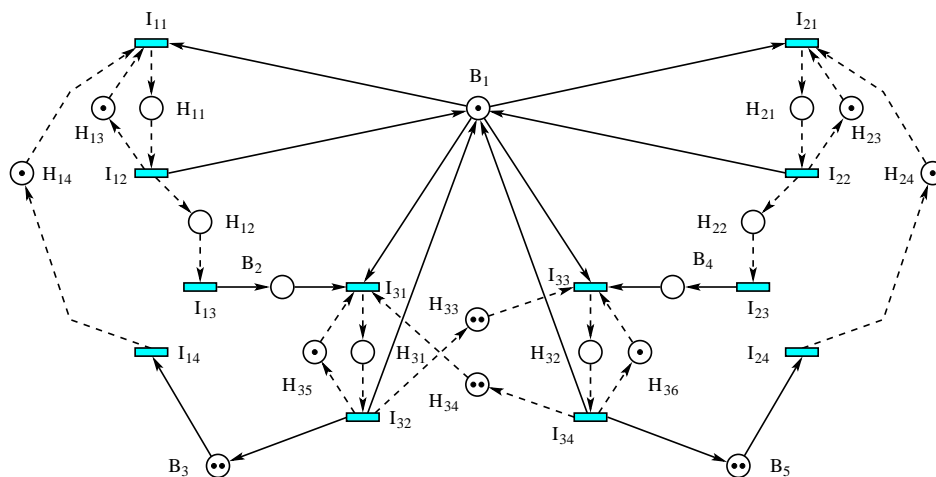


Figura 7.34: \mathcal{BS} del SAM de la figura 7.31.

aplicar el algoritmo iterativo 6.43 a los sistemas agregados.

\mathcal{LS}_1		\mathcal{LS}_2		\mathcal{LS}_3	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ	$\mathcal{X}(I_{31})$	λ
0.230496	1.025007	0.223043	1.089388	0.222493	1.005590
0.223777	1.090060	0.223460	1.091597	0.223461	1.005590
0.223471	1.091623	0.223472	1.091597	0.223472	1.005590
0.223470	1.091623	0.223471	1.091597	0.223472	1.005590
$\mathcal{X}(I_{11})$ exacto: 0.224607				Error: -0.51%	

Tabla 7.34: Aproximación de \mathcal{X} para Fig. 7.31 (técnicas generales).

En la tabla 7.34 aparecen los resultados de la ejecución del algoritmo iterativo en el ejemplo que se está desarrollando. En este caso, el throughput exacto de la transición I_{11} de la red original es 0.224607. La leyenda de la tabla es la misma que en ejemplos anteriores. Para cada sistema de bajo nivel se muestra el throughput de la transición de referencia (I_{i1}) y el valor del factor de escala λ calculado en el \mathcal{BS} para actualizar las tasas de las transiciones de interfaz en los siguientes sistemas agregados. En este caso se observa que se alcanza la convergencia en 4 iteraciones, obteniendo una aproximación de throughput de 0.223470 por lo que el error en la aproximación final es del -0.51% , bastante aceptable para una técnica de aproximación.

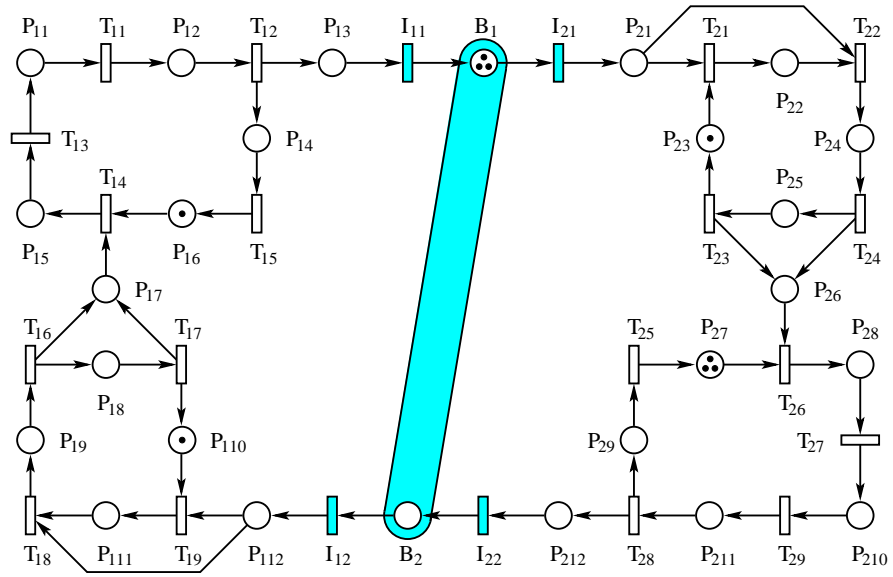


Figura 7.35: Ejemplo 2 de SAM's.

El ejemplo que se ha desarrollado hasta ahora no ilustra de forma adecuada la diferencia entre las tres técnicas desarrolladas en el capítulo 6 ya que los sistemas agregados que se obtienen por medio del conjunto de lugares implícitos calculado no introducen comportamientos no deseados en los grafos de alcanzabilidad asociados a los sistemas agregados. Por ello se va a estudiar a continuación un ejemplo en el que si ocurren estos efectos no deseados y por lo tanto las tres técnicas dan resultados distintos.

Este segundo ejemplo es el de la figura 7.35. Es un SAM compuesto por dos módulos conectados entre sí por medio de 2 canales. Cada módulo \mathcal{S}_i con $i = 1, 2$ está compuesto por los lugares de etiquetas P_{ij} y las transiciones de etiquetas I_{ij} y T_{ij} . Se sigue utilizando la misma notación de todos los ejemplos anteriores.

Por lo que respecta a la evolución temporal de la red, todas las tasas de disparo de las transiciones se han puesto a 1.0.

El grafo de alcanzabilidad de esta red tiene 2698 estados. En un SAM la descomposición viene inducida por su visión estructurada, por lo que en este caso se ha efectuado una descomposición del sistema original en sus 2 módulos (\mathcal{S}_1 y \mathcal{S}_2).

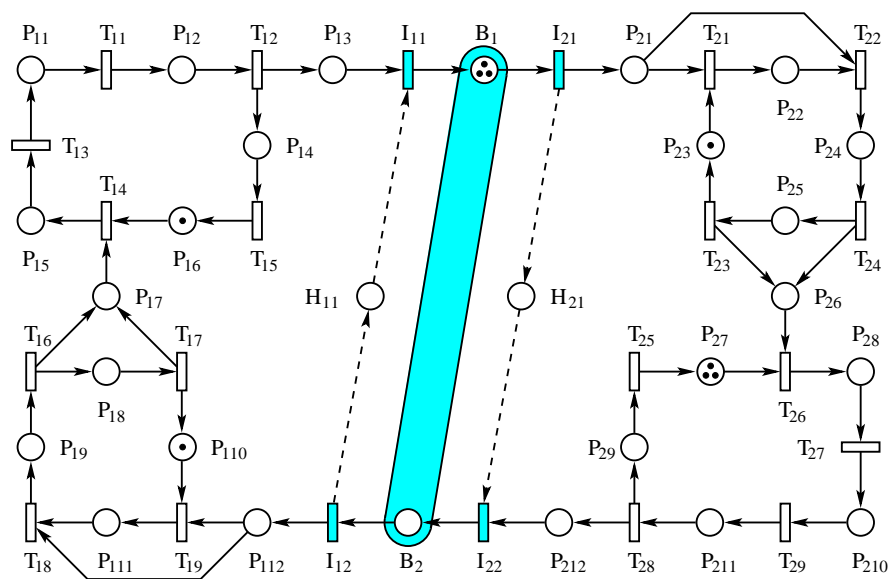


Figura 7.36: \mathcal{ES} del SAM de la figura 7.35.

El siguiente paso consiste en calcular el conjunto de lugares implícitos que resumen el comportamiento de cada módulo. Para ello hay que tomar cada módulo y aplicar la relación de equivalencia de la definición 6.2. En este caso la relación de equivalencia induce una única clase de equivalencia por módulo a la que pertenecen todos sus lugares internos.

El siguiente paso consiste en calcular, por medio del algoritmo 6.6, el conjunto de lugares que resumen el comportamiento de cada módulo. En este caso hace falta un único lugar por módulo. Por ejemplo, para resumir el módulo \mathcal{S}_1 hace falta un único lugar que conecta I_{12} con I_{11} y para resumir el módulo \mathcal{S}_2 hace falta un único lugar que conecta I_{21} con I_{22} . En total son necesarios 2 lugares y ninguno está presente en la red inicial, por lo que es necesario añadir los 2.

Estos lugares son estructuralmente implícitos marcados (MSIP's), por lo que es necesario calcular el marcado inicial necesario para hacerlos implícitos al añadirlos a la red inicial. Aplicando el algoritmo 2.60 se calculan estos marcados iniciales, que resultan ser nulos en los dos casos.

Si se añaden a la red original los lugares implícitos calculados se obtiene el sistema extendido \mathcal{ES} de la figura 7.36. Los lugares cuyos arcos de entrada y

salida son discontinuos son los lugares MSIP que resumen el comportamiento de los dos módulos.

A partir del \mathcal{ES} es inmediato construir los sistemas agregados. En las figuras 7.37.a, 7.37.b y 7.37.c se pueden observar el \mathcal{LS}_1 , \mathcal{LS}_2 y \mathcal{BS} correspondientes al SAM original.

En este caso, a diferencia del ejemplo anterior si aparecen efectos no deseados en los sistemas agregados dependiendo de la técnica general de reducción que se aplique.

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.174202	1.030663	0.124153	1.209259
0.114948	1.229055	0.113760	1.255223
0.113605	1.236202	0.113590	1.256088
0.113587	1.236291	0.113587	1.256088
0.113586	1.236291	0.113587	1.256088
$\mathcal{X}(I_{11})$ exac: 0.104304		Err: +8.90%	

Tabla 7.35: Aproximación de \mathcal{X} para Fig. 7.35 (técnica general 1).

Si se aplica la primera técnica general de reducción (desarrollada en la sección 6.3) el \mathcal{LS}_1 tiene 310 estados alcanzables, el \mathcal{LS}_2 828 y el \mathcal{BS} 20. Aplicando el algoritmo numérico 6.43 a estos sistemas agregados se obtiene los resultados de la tabla 7.35. El throughput exacto de la transición I_{11} de la red original es 0.104304 y la aproximación obtenida después de 5 iteraciones de 0.113586, lo que supone un error del +8.90%, excesivo ya para una técnica de aproximación.

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.158793	0.998916	0.105566	1.139154
0.082152	0.949469	0.079293	1.253284
0.077368	0.950757	0.077125	1.265766
0.076956	0.950939	0.076935	1.266893
0.076920	0.950960	0.076918	1.266987
0.076916	0.950960	0.076917	1.267034
$\mathcal{X}(I_{11})$ exac: 0.104304		Err: -26.26%	

Tabla 7.36: Aproximación de \mathcal{X} para Fig. 7.35 (técnica general 2).

Si se aplica la segunda técnica general de reducción (desarrollada en la sección 6.5) el \mathcal{LS}_1 tiene 211 estados alcanzables, el \mathcal{LS}_2 487 y el \mathcal{BS} 18.

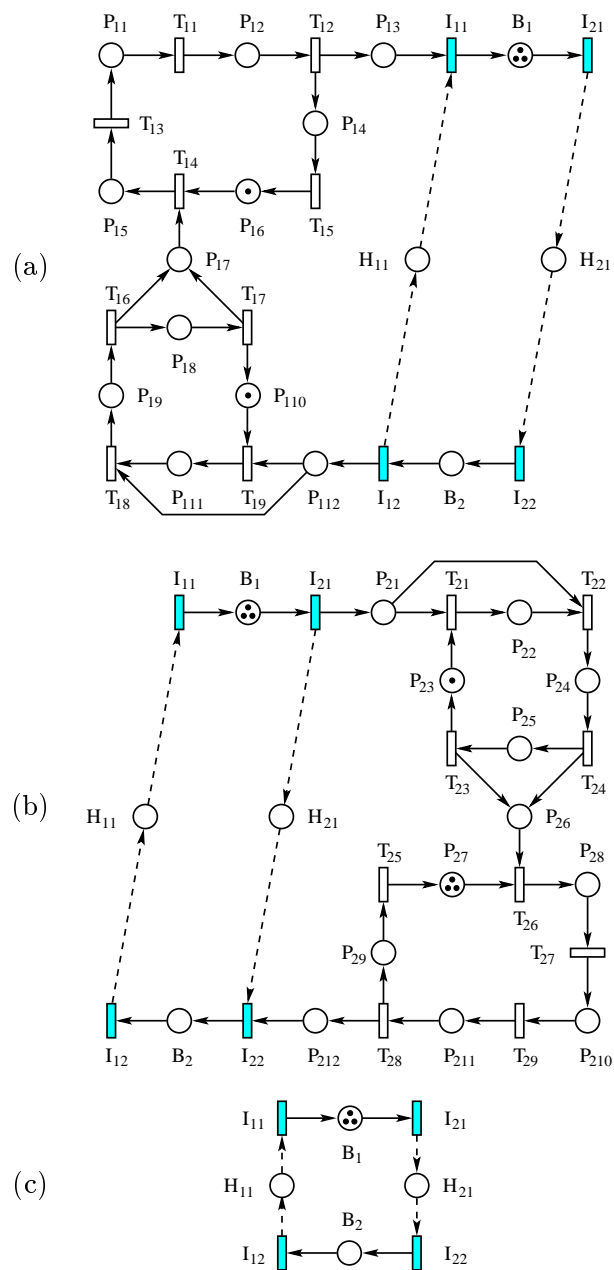


Figura 7.37: (a) \mathcal{LS}_1 , (b) \mathcal{LS}_2 y (c) \mathcal{BS} del SAM de la figura 7.35.

Aplicando el algoritmo numérico 6.43 a estos sistemas agregados se obtiene los resultados de la tabla 7.36. Como el throughput exacto de la transición I_{11} era 0.104304 y la aproximación obtenida después de 6 iteraciones es de 0.076916, se ha cometido un error del -26.26% , todavía peor que en el caso anterior.

\mathcal{LS}_1		\mathcal{LS}_2	
$\mathcal{X}(I_{11})$	λ	$\mathcal{X}(I_{21})$	λ
0.158793	1.048865	0.110149	1.031971
0.105799	1.003225	0.105927	1.034708
0.105940	1.003435	0.105941	1.034708
0.105939	1.003435	0.105941	1.034708
$\mathcal{X}(I_{11})$ exac: 0.104304		Err: +1.57%	

Tabla 7.37: Aproximación de \mathcal{X} para Fig. 7.35 (técnica general 3).

Por último, si se aplica la tercera técnica general de reducción (desarrollada en la sección 6.6) el \mathcal{LS}_1 tiene 211 estados alcanzables, el \mathcal{LS}_2 487 y el \mathcal{BS} 22. Aplicando el algoritmo numérico 6.43 a estos sistemas agregados se obtiene los resultados de la tabla 7.37. Ahora el algoritmo converge en 4 iteraciones y se obtiene una aproximación para el throughput de la transición I_{11} de 0.105939, que respecto al valor exacto de 0.104304 supone un error de $+1.57\%$, ya aceptable.

Para analizar estos resultados hay que fijarse en los tamaños de los espacios de estados de los sistemas agregados obtenidos con cada técnica de descomposición y reducción. Utilizando como referencia la segunda técnica de reducción, en la que los sistemas agregados no tienen marcados espurios, se observa que los obtenidos con la primera técnica tienen muchos marcados espurios. El \mathcal{LS}_1 de la primera técnica tiene 310 estados frente a los 211 que se obtienen con las otras dos técnicas, lo que supone un 47% de estados alcanzables añadidos respecto a los que debería tener. En el \mathcal{LS}_2 la situación es peor porque tiene 828 estados alcanzables frente a los 487 que debería tener (un 70% más). Obviamente es significativo el número de estados espurios respecto a los que no lo son y de ahí que falle la aproximación. Para saber de dónde vienen estos marcados espurios es necesario profundizar en la estructura y funcionamiento de la red inicial. Si se observa la SM generada por las transiciones T_{21} , T_{22} , T_{23} y T_{24} se ve que cada ciclo de esta SM consume dos marcas de P_{21} y deposita dos marcas en P_{26} . Por lo tanto en el módulo \mathcal{S}_2 se tiene una sincronización interna entre el lugar P_{21} y el P_{26} debido a la SM. Son necesarias dos marcas en P_{21} para que la SM pueda

dar un ciclo completo y pasar las marcas a P_{26} . Este tipo de sincronización es del mismo tipo estudiado en el capítulo 4 para WTS con el concepto de resistencia. Lo mismo le ocurre al módulo \mathcal{S}_1 . En los lugares implícitos calculados no se preservan estas sincronizaciones internas, por lo que al reducir los módulos en \mathcal{LS}_1 y \mathcal{LS}_2 estas sincronizaciones internas desaparecen y el número de marcados alcanzables aumenta.

Para entender la razón del fallo de la segunda técnica hay que fijarse en el \mathcal{BS} . Los \mathcal{LS}_1 y \mathcal{LS}_2 obtenidos con las técnicas segunda y tercera son exactamente los mismos y son proyecciones exactas en términos de marcados alcanzables de la red original sobre los lugares preservados en cada sistema agregado. La única diferencia estriba entonces en el \mathcal{BS} . El \mathcal{BS} de la segunda técnica tiene 4 estados menos que el de la tercera. Esta diferencia sólo puede venir de la agregación inadecuada de estados que se puede producir en los sistemas agregados calculados con la segunda técnica. En este ejemplo, especialmente diseñado para que falle la segunda técnica, la simple agregación de 4 estados en \mathcal{BS} hace fallar la aproximación.

7.5 Conclusiones

En este capítulo se han aplicado todas las técnicas desarrolladas en la memoria para la descomposición y reducción de distintas clases de redes con objeto de aproximar por medio de un algoritmo numérico el throughput de sus transiciones. Aprovechando la exposición conjunta de las técnicas y el hecho de que dado un ejemplo es posible aplicarle más de una técnica, se ha utilizado el capítulo para hacer todas las comparaciones posibles. En la mayor parte de los ejemplos expuestos el algoritmo numérico opera con sistemas agregados que tienen un tamaño de espacio de estados un orden de magnitud inferior al de la red inicial, converge en 3, 4 ó 5 iteraciones y consigue aproximaciones con un error inferior al 3%, lo que es aceptable para una técnica de aproximación.

Se han comparado en diversos ejemplos las técnicas desarrolladas para clases particulares de redes y las desarrolladas para aplicarse a cualquier red de Petri. La comparación entre ellas no permite concluir que son mejores unas u otras. En algunos ejemplos las técnicas particulares funcionan ligeramente mejor y en otros ligeramente peor. La ventaja de las técnicas generales estriba precisamente en el campo más amplio de aplicación, aunque pueden fallar en algunos casos, como se ha podido observar en el último

ejemplo.

Queda como trabajo futuro estudiar si la tercera técnica general, la más elaborada, falla en algún caso y en ese caso bajo qué condiciones.

Capítulo 8

Conclusiones

En esta memoria se han desarrollado diversas técnicas para la aproximación del throughput de las transiciones de redes de Petri estocásticas. Estas técnicas se basan en una descomposición estructural del modelo original en varias componentes, a partir de las cuales se construyen varios sistemas agregados. Posteriormente, por medio de un algoritmo numérico iterativo basado en la aproximación del tiempo de respuesta se aproxima el throughput de las transiciones del modelo original a partir de los sistemas agregados. Para realizar esta tarea, se han estudiado clases de redes cada vez más complejas hasta que se ha estado en disposición de atacar el problema general.

En el capítulo 3 se han añadido algunas mejoras a la técnica de [CCJS94] de aproximación de throughput en grafos marcados (MG's). En concreto, se ha ampliado el tipo de corte para permitir descomponer un MG en varias componentes, no sólo en dos. También se ha reducido la estructura de los sistemas agregados obtenidos y se ha demostrado la existencia de punto de convergencia para el algoritmo numérico.

En el capítulo 4 se ha desarrollado en detalle el trabajo de [PJCS96b] sobre aproximación de throughput en grafos marcados con pesos (WTS's). Se ha aprovechado la ocasión para introducir algunas mejoras técnicas. En concreto, se ha desarrollado la descomposición estructural de WTS's, se ha reducido la estructura de los sistemas agregados obtenidos, se ha adaptado el algoritmo numérico (es básicamente el mismo de [CCJS94]) a la nueva clase de redes y se ha demostrado formalmente la existencia de punto de convergencia para el mismo. Esta técnica es una extensión de la de MG's, en el sentido de que las dos técnicas aplicadas a un MG producen los mismos resultados.

En el capítulo 5 se han desarrollado en detalle los trabajos realizados en [PJCS96c, PJCS96a] sobre aproximación de throughput en sistemas deterministas de procesos secuenciales (DSSP's). Los DSSP's están formados por un conjunto de máquinas de estados (SM's) conectadas entre sí por una serie de lugares llamados canales. Por lo tanto, es necesario reducir la estructura de estas SM's. En concreto, se desarrollan dos técnicas de reducción para SM's, una de reducción parcial y otra de reducción total. Una vez reducidas las SM's de un DSSP es posible que queden subredes con estructura de WTS, en cuyo caso se puede aplicar la técnica del capítulo 4 para obtener una mayor reducción de la estructura del DSSP. Por último, se ha adaptado el algoritmo numérico de aproximación (sigue siendo básicamente el mismo que el de los capítulos anteriores). La técnica de aproximación para DSSP's es una extensión de las anteriores en el sentido de que al aplicarla a un MG o WTS produce exactamente resultados que las técnicas de los capítulos anteriores.

En el capítulo 6 se han desarrollado en detalle los trabajos realizados en [PJC98, PJC99a] sobre técnicas de descomposición de redes de Petri para su análisis numérico. La descomposición viene inducida por la estructura de la red, a la que se le da una visión estructurada. Las aportaciones de este capítulo son la visión estructurada del grafo de alcanzabilidad de una red de Petri, el desarrollo de tres técnicas generales de descomposición de redes de Petri para su análisis numérico y la adaptación del método numérico de aproximación del throughput de transiciones para cualquier red de Petri. La visión estructurada del grafo de alcanzabilidad permite, por ejemplo, generar una descripción descompuesta de todo el grafo de alcanzabilidad de la red original con un algoritmo más eficiente que el clásico tanto en espacio como en tiempo. Esta descripción descompuesta se emplea posteriormente para generar el grafo de alcanzabilidad de una red con un algoritmo más eficiente que el clásico en tiempo.

En el capítulo 7 se desarrolla una batería de ejemplos con distintas redes en las que se comparan los resultados obtenidos con las técnicas que le son aplicables.

Queda como trabajo futuro demostrar las propiedades del algoritmo numérico tales como convergencia y unicidad de solución. Se ha intentado demostrar sin éxito completo hasta el momento.

Bibliografía

- [ABS84] S. C. Agrawal, J. P. Buzen, and A. W. Shum. Response time preservation: A general technique for developing approximate algorithms for queueing networks. In *Proceedings of the 1984 ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pages 63–77, Cambridge, MA, August 1984.
- [AHU83] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Data Structures and Algorithms*. Addison-Wesley, 1983.
- [AI89] H. H. Ammar and S. M. R. Islam. Time scale decomposition of a class of generalized stochastic Petri net models. *IEEE Transactions on Software Engineering*, 15(6):809–820, June 1989.
- [AMBB⁺89] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Transactions on Software Engineering*, 15(7):832–846, July 1989.
- [AMBC84] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(2):93–122, May 1984.
- [AMBC⁺95] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons, 1995.
- [AMBCC87] M. Ajmone Marsan, G. Balbo, G. Chiola, and G. Conte. Generalized stochastic Petri nets revisited: Random switches and

- priorities. In *Proceedings of the International Workshop on Petri Nets And Performance Models*, pages 44–53, Madison, WI, USA, August 1987. IEEE Computer Society Press.
- [BCDK00] P. Buchholz, G. Ciardo, S. Donatelli, and P. Kemper. Kronecker operations and sparse matrices with applications to the solution of Markov models. *INFORMS Journal on Computing*, 12(3):203–222, 2000.
- [BD90a] B. Baynat and Y. Dallery. Approximate techniques for general closed queueing networks with subnetworks having population constraints. Rapport MASI 90.49, Laboratoire MASI, Université Paris VI, Paris, France, October 1990.
- [BD90b] B. Baynat and Y. Dallery. A unified view of product-form approximation techniques for general closed queueing networks. Rapport MASI 90.48, Laboratoire MASI, Université Paris VI, Paris, France, October 1990.
- [BD96] B. Baynat and Y. Dallery. A product-form approximation method for general closed queueing networks with several classes of customers. *Performance Evaluation*, pages 165–188, 1996.
- [Ber86] G. Berthelot. Checking properties of nets using transformations. In G. Rozenberg, editor, *Advances in Petri Nets 1985*, volume 222 of *Lecture Notes in Computer Science*, pages 19–40. Springer-Verlag, Berlin, 1986.
- [BI82] S. Balsamo and G. Iazeolla. An extension of Norton’s theorem for queueing networks. *IEEE Transactions on Software Engineering*, 8(4):298–305, July 1982.
- [BK01] P. Buchholz and P. Kemper. Compact representations of probability distributions in the analysis of superposed GSPN’s. In *Proceedings of the 9th International Workshop on Petri Nets and Performance Models*, pages 81–90, Aachen, Germany, September 2001.
- [BL92] F. Baccelli and Z. Liu. Comparison properties of stochastic decision free Petri nets. *IEEE Transactions on Automatic Control*, 37(12):1905–1920, December 1992.

- [BLS96] F. Baccelli, Z. Liu, and M. Silva. Local and global monotonicities of stochastic Petri nets. Private talk, June 1996.
- [BT81] A. Bertoni and M. Torelli. Probabilistic Petri nets and semi-Markov systems. In *Proceedings of the 2nd European Workshop on Petri Nets*, pages 59–78, Bad Honnef, Germany, September 1981.
- [Cam90] J. Campos. *Performance Bounds for Synchronized Queueing Networks*. PhD thesis, Departamento de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Spain, October 1990. Research Report GISI-RR-90-20.
- [CCCS92] J. Campos, G. Chiola, J. M. Colom, and M. Silva. Properties and performance bounds for timed marked graphs. *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 39(5):386–401, May 1992.
- [CCJS94] J. Campos, J. M. Colom, H. Jungnitz, and M. Silva. Approximate throughput computation of stochastic marked graphs. *IEEE Transactions on Software Engineering*, 20(7):526–535, July 1994.
- [CCS91] J. Campos, G. Chiola, and M. Silva. Properties and performance bounds for closed free choice synchronized monoclase queueing networks. *IEEE Transactions on Automatic Control*, 36(12):1368–1382, December 1991.
- [CDS99] J. Campos, S. Donatelli, and M. Silva. Structured solution of asynchronously communicating stochastic modules. *IEEE Transactions on Software Engineering*, 25(2):147–165, March 1999.
- [CHEP71] F. Commoner, A. Holt, S. Even, and A. Pnueli. Marked directed graphs. *Journal of Computer and System Sciences*, 5(5):511–523, October 1971.
- [Chi87] G. Chiola. A graphical Petri net tool for performance analysis. In *Proceedings of the 3rd International Workshop on Modeling Techniques and Performance Evaluation*, Paris, France, March 1987. AFCET.

- [Cia01] G. Ciardo. What a structural world. In *Proceedings of the 9th International Workshop on Petri Nets and Performance Models*, pages 3–16, Aachen, Germany, September 2001.
- [CLR90] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 1990.
- [CS91] J. M. Colom and M. Silva. Improving the linearly based characterization of P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 113–145. Springer-Verlag, Berlin, 1991.
- [CS93] J. Campos and M. Silva. Embedded product-form queueing networks and the improvement of performance bounds for Petri net systems. *Performance Evaluation*, 18(1):3–19, July 1993.
- [CT91] G. Ciardo and K. S. Trivedi. A decomposition approach for stochastic Petri nets models. In *Proceedings of the 4th International Workshop on Petri Nets and Performance Models*, pages 74–83, Melbourne, Australia, December 1991. IEEE Computer Society Press.
- [CT93] G. Ciardo and K. Trivedi. A decomposition approach for stochastic reward net models. *Performance Evaluation*, 18(1):37–59, September 1993.
- [DHP⁺93] F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva, and F.B. Vernadat. *Practice of Petri Nets in Manufacturing*. Chapman & Hall, London, 1993.
- [DLT90] Y. Dallery, Z. Liu, and D. Towsley. Equivalence, reversibility and symmetry properties in fork/join queueing networks with blocking. Technical Report MASI-90-32, Laboratoire MASI, Université Paris VI, Paris, France, June 1990.
- [Don94] S. Donatelli. Superposed generalized stochastic Petri nets: Definition and efficient solution. In R. Valette, editor, *Application and Theory of Petri Nets 1994*, volume 815 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1994.

- [DS92] S. Donatelli and M. Sereno. On the product form solution for stochastic Petri nets. In K. Jensen, editor, *Application and Theory of Petri Nets 1992*, volume 616 of *Lecture Notes in Computer Science*, pages 154–172. Springer-Verlag, Berlin, 1992.
- [Fin93] A. Finkel. The minimal coverability graph for Petri nets. In G. Rozenberg, editor, *Advances in Petri Nets 1993*, volume 674 of *Lecture Notes in Computer Science*, pages 210–243. Springer-Verlag, Berlin, 1993.
- [FN85a] G. Florin and S. Natkin. Les réseaux de Petri stochastiques. *Technique et Science Informatiques*, 4(1):143–160, February 1985. In French.
- [FN85b] G. Florin and S. Natkin. Les réseaux de Petri stochastiques, 1985. Thesis de Doctorat d’Etat, Université Pierre et Marie Curie, Paris. In French.
- [FZ98] J. Freiheit and A. Zimmermann. Extending a response time approximation technique to colored stochastic Petri nets. In *Proceedings of the 4th International Workshop on Performance Modeling of Computer and Communication Systems (PMCCS 4)*, pages 67–71, Williamsburg, VA, USA, September 1998.
- [FZ01] J. Freiheit and A. Zimmermann. A divide and conquer approach for the performance evaluation of large stochastic Petri nets. In *Proceedings of the 9th International Workshop on Petri Nets and Performance Models*, pages 91–100, Aachen, Germany, September 2001.
- [Gib85] A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, London, 1985.
- [Gre76] W. Greub. *Linear Algebra*. Springer-Verlag, 1976.
- [HHM95] H. Hermans, U. Herzog, and V. Mertsiotakis. Stochastic process algebras as a tool for performance and dependability modelling. In *Proceedings of the IEEE International Computer Performance and Dependability Symposium*, pages 102–113, Erlangen, Germany, April 1995. IEEE-Computer Society Press.

- [Ho89] Y. Ho. Scanning the issue, dynamics of discrete event systems. *Proceedings of the IEEE*, 77(1), 1989.
- [HV85] M. A. Holliday and M. K. Vernon. A generalized timed Petri net model for performance analysis. In *Proceedings of the International Workshop on Timed Petri Nets*, pages 181–190, Torino, Italy, July 1985. IEEE-Computer Society Press.
- [JD91] H. Jungnitz and A. Desrochers. Flow equivalent nets for the performance analysis of flexible manufacturing systems. In *Proceedings of the IEEE Robotics and Automation Conference 1991*, pages 122–127, 1991.
- [JSS92] H. Jungnitz, B. Sánchez, and M. Silva. Approximate throughput computation of stochastic marked graphs. *Journal of Parallel and Distributed Computing*, 15:282–295, 1992.
- [Jun92] H. J. Jungnitz. *Approximation Methods for Stochastic Petri Nets*. PhD thesis, Dept. of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA, May 1992.
- [Kan92] K. Kant. *Introduction to Computer System Performance Evaluation*. Mc Graw-Hill, 1992.
- [Kla86] G. Klambauer. *Aspects of Calculus*. Springer-Verlag, Berlin, 1986.
- [KS76] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Springer-Verlag, 1976.
- [LW91a] Y. Li and C. M. Woodside. Iterative decomposition and aggregation of stochastic marked graphs Petri nets. In *Proceedings of the 12th International Conference on Applications and Theory of Petri Nets*, pages 257–275, Gjern, Denmark, June 1991.
- [LW91b] Y. Li and C. M. Woodside. Performance Petri net analysis of communications protocol software by delay-equivalent aggregation. In *Proceedings of the 4th International Workshop on Petri Nets and Performance Models*, pages 64–73, Melbourne, Australia, December 1991. IEEE-Computer Society Press.

- [LW95] Y. Li and C. M. Woodside. Complete decomposition of stochastic Petri nets representing generalized service networks. *IEEE Transactions on Computers*, 44(8):1031–1046, August 1995.
- [Mar79] R. A. Marie. An approximate analytical method for general queueing networks. *IEEE Transactions on Software Engineering*, 5(5):530–538, September 1979.
- [Mol81] M. K. Molloy. *On the Integration of Delay and Throughput Measures in Distributed Processing Models*. PhD thesis, UCLA, Los Angeles, CA, USA, 1981.
- [Mol82] M. K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Transactions on Computers*, 31(9):913–917, September 1982.
- [MS82] J. Martínez and M. Silva. A simple and fast algorithm to obtain all invariants of a generalized Petri net. In C. Girault and W. Reisig, editors, *Application and Theory of Petri Nets*, volume 52 of *Informatik-Fachberichte*, pages 301–310, Berlin, Germany, 1982. Springer-Verlag.
- [MT95] V. Mainkar and K. Trivedi. Fixed point iteration using stochastic reward nets. In *Proceedings of the 6th International Workshop on Petri Nets and Performance Models*, pages 21–30, Durham, North Carolina, USA, October 1995. IEEE-Computer Society Press.
- [Mun93] A. Munier. Régime asymptotique optimal d'un graphe d'événements temporisé généralisé: application à un problème d'assemblage. *APII*, 27(5):487–513, 1993.
- [Mur89] T. Murata. Petri nets: Properties, analysis, and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.
- [Pet66] C. A. Petri. Communication with automata. Technical report RADC-TR-65-377, Rome Air Dev. Center, New York, NY, USA, 1966.
- [Pet81] J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1981.

- [PJC98] C. J. Pérez-Jiménez and J. Campos. A response time approximation technique for stochastic general P/T systems. In *Proceedings of the IMACS/IEEE-SMC Multiconference on Computational Engineering in Systems Applications (CESA'98)*, Nabeul-Hammamet, Tunisia, April 1998.
- [PJC99a] C. J. Pérez-Jiménez and J. Campos. On state space decomposition for the numerical analysis of stochastic Petri nets. In *Proceedings of the 8th International Workshop on Petri Nets and Performance Models*, pages 32–41, Zaragoza, Spain, September 1999.
- [PJC99b] C. J. Pérez-Jiménez and J. Campos. On state space decomposition for the numerical analysis of stochastic Petri nets. In *Actas de las VII Jornadas de Concurrencia*, pages 249–268, Gandía (Valencia), Spain, June 1999.
- [PJCS95] C. J. Pérez-Jiménez, J. Campos, and M. Silva. On approximate throughput computation of deterministic systems of sequential processes. In *Actas de las IV Jornadas Nacionales sobre Concurrencia*, pages 156–171, El Escorial, Spain, July 1995.
- [PJCS96a] C. J. Pérez-Jiménez, J. Campos, and M. Silva. Approximate throughput computation of a class of cooperating sequential processes. In *Proceedings of the Rensselaer's Fifth International Conference on Computer Integrated Manufacturing and Automation Technology (CIMAT'96)*, pages 382–389, Grenoble, France, May 1996.
- [PJCS96b] C. J. Pérez-Jiménez, J. Campos, and M. Silva. On approximate performance evaluation of manufacturing systems modelled with weighted T-systems. In *Proceedings of the IMACS/IEEE-SMC Multiconference on Computational Engineering in Systems Applications (CESA'96)*, pages 201–207, Lille, France, July 1996.
- [PJCS96c] C. J. Pérez-Jiménez, J. Campos, and M. Silva. State machine reduction for the approximate performance evaluation of manufacturing systems modelled with cooperating sequential processes. In *Proceedings of the 1996 IEEE International Con-*

- ference on Robotics and Automation*, pages 1159–1165, Minneapolis, Minnesota, USA, April 1996.
- [PS00] B. Plateau and W. Stewart. *Advances in Computational Probability, Chapter Stochastic Automata Networks*. Kluwer Academic Publishers, 2000.
- [Ram74] C. Ramchandani. *Analysis of Asynchronous Concurrent Systems by Petri Nets*. PhD thesis, MIT, Cambridge, MA, USA, February 1974.
- [Rei82] W. Reisig. Deterministic buffer synchronization of sequential processes. *Acta Informatica*, 18:117–134, 1982.
- [RP84] R. R. Razouk and C. V. Phelps. Performance analysis using timed Petri nets. In *Proceedings of the International Conference on Parallel Processing*, pages 126–129, August 1984.
- [RTS95] L. Recalde, E. Teruel, and M. Silva. On well-formedness analysis: The case of deterministic systems of sequential processes. In J. Desel, editor, *Proceedings of the International Workshop on Structures in Concurrency Theory (STRICT)*, pages 279–293. Springer Verlag, 1995.
- [RTS98] L. Recalde, E. Teruel, and M. Silva. Modeling and analysis of sequential processes that cooperate through buffers. *IEEE Transactions on Robotics and Automation*, 14(2): 267–277, 1998.
- [SC98] M. Silva and J. Campos. Performance evaluation of DEDS with conflicts and synchronizations: Net-driven decomposition techniques. In *Proceedings of the 4th International Workshop on Discrete Event Systems*, pages 398–413, Cagliari, Italy, August 1998.
- [Shi87] M. W. Shields. *An Introduction to Automata Theory*. Blackwell Scientific Publications, 1987.
- [Sif78] J. Sifakis. Use of Petri nets for performance evaluation. *Acta Cybernetica*, 4(2):185–202, 1978.

- [Sil85] M. Silva. *Las Redes de Petri en la Automática y la Informática*. Editorial AC, Madrid, 1985. In Spanish.
- [Sma74] D. R. Smart. *Fixed Point Theorems*. Cambridge University Press, 1974.
- [Sou93] Y. Souissi. Deterministic systems of sequential processes: A class of structured Petri nets. In G. Rozenberg, editor, *Advances in Petri Nets 1993*, volume 674 of *Lecture Notes in Computer Science*, pages 406–426. Springer-Verlag, Berlin, 1993.
- [ST96] M. Silva and E. Teruel. A systems theory perspective of discrete event dynamic systems: The Petri net paradigm. In P. Borne, J. C. Gentina, E. Craye, and S. El Khattabi, editors, *Symposium on Discrete Events and Manufacturing Systems. Proceedings of the IMACS/IEEE-SMC Multiconference on Computational Engineering in Systems Applications (CESA'96)*, pages 1–12, Lille, France, July 1996.
- [TCWCS92] E. Teruel, P. Chrzastowski-Wachtel, J. M. Colom, and M. Silva. On weighted T-systems. In K. Jensen, editor, *Application and Theory of Petri Nets 1992*, volume 616 of *Lecture Notes in Computer Science*, pages 348–367. Springer-Verlag, Berlin, 1992.
- [Ter94] E. Teruel. *Structure Theory of Weighted Place/Transition Net Systems: The Equal Conflict Hiatus*. PhD thesis, Departamento de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Spain, June 1994.
- [TSCC95] E. Teruel, M. Silva, J. M. Colom, and J. Campos. Functional and performance analysis of cooperating sequential processes. In F. Baccelli, A. Jean-Marie, and I. Mitrani, editors, *Quantitative Methods in Parallel Systems*, Esprit Basic Research Series, pages 52–65. Springer, 1995.
- [Zub85] W. M. Zuberek. Performance evaluation using timed Petri nets. In *Proceedings of the International Workshop on Timed Petri Nets*, pages 272–278, Torino, Italy, July 1985. IEEE-Computer Society Press.