

Programación Concurrente. *Model Checking* con Maria.

María José Ibáñez

1 Objetivos

1. Adquirir conocimiento empírico del funcionamiento del analizador para sistemas concurrentes María.
2. Edición y Análisis de sistemas concurrentes utilizando María.

2 Contenidos

La práctica consta de cuatro partes distintas, que deberán ser resueltas.

Parte 1

- Descarga la aplicación María desde <http://www.tcs.hut.fi/Software/maria/index.en.html>. El fichero descargado contiene un manual que puede ser útil a lo largo de la práctica.
- Descomprime el archivo .zip en un directorio con nombre *Maria*.
- Ejecuta en María el modelo `dining1.pn` (Figura 1) comprobando si tiene bloqueos. Desde el directorio *Maria* utiliza la orden `maria -b dining1.pn`.
- María responde que el modelo tiene un bloqueo en el estado 78. Muestra este estado con la orden `show @78`.
- **Cuestión 1.** ¿De qué estado se trata?
- Verifica si el modelo `dining1.pn` cumple las siguientes propiedades de lógica temporal.

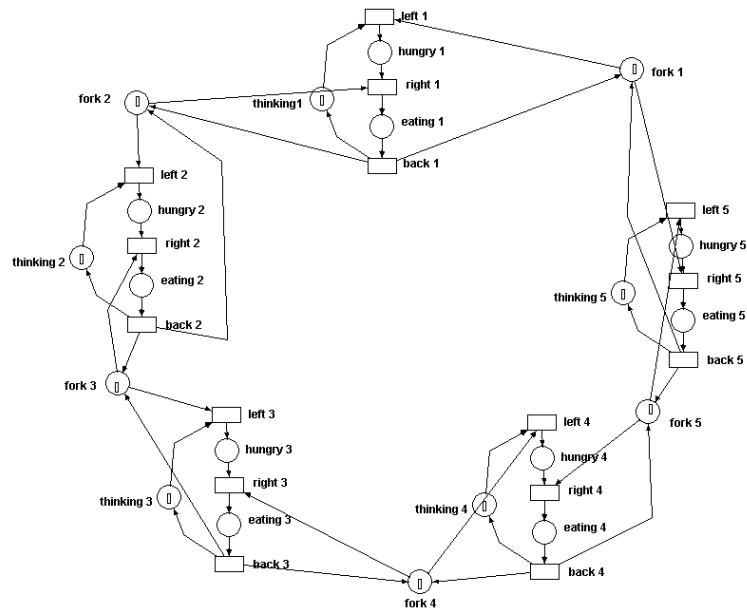


Figure 1: Red de Petri coloreada para representar el problema de los filósofos. dining1.pn

- $F\neg(\text{thinking1}=3)$:
Utiliza la orden $\langle \rangle !(\text{place thinking1 equals } 3)$. Esta pregunta significa si en el futuro no se alcanzará un estado en el cual el lugar thinking 1 tenga 3 marcas.
- $G\neg(\text{thinking1}=3)$:
Utiliza la orden $[\] !(\text{place thinking1 equals } 3)$.
- $FG\neg(\text{thinking1}=3)$:
Utiliza la orden $\langle \rangle [\] !(\text{place thinking1 equals } 3)$.
- $GF\neg(\text{thinking1}=3)$:
Utiliza la orden $[\] \langle \rangle !(\text{place thinking1 equals } 3)$.
- $F(\text{hungry1}=0)$:
Utiliza la orden $\langle \rangle (\text{place hungry1 equals empty})$.
- $G(\text{hungry1}=0)$:
Utiliza la orden $[\] (\text{place hungry1 equals empty})$.
- $GF(\text{hungry1} = 1) \text{ or } (\neg GF (\text{place hungry1}=1))$:
Utiliza la orden $[\] \langle \rangle (\text{place hungry1 equals } 1) \text{ or } ![\] \langle \rangle (\text{place thinking1 equals } 1)$.

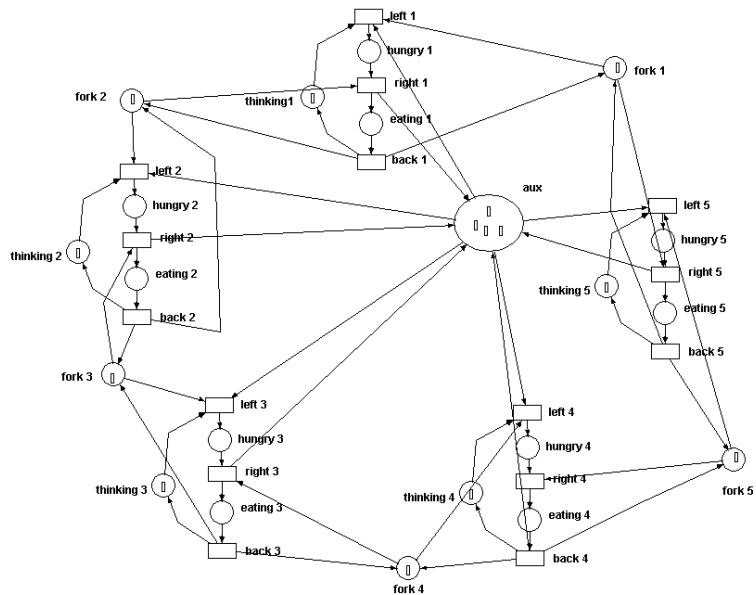


Figure 2: Red de Petri coloreada para representar el problema de los filósofos. `dining1NB.pn`

- **Cuestión 2.** Explica el significado y resultado de cada una de las preguntas.
- Ejecuta en María el modelo `dining1NB.pn` (Figura 2) comprobando si tiene bloqueos.
- Verifica si el modelo `dining1NB.pn` cumple las siguientes propiedades de lógica temporal.
 - $F\neg(\text{thinking1}=3)$:
Utiliza la orden `<> !(place thinking1 equals 3)`.
 - $G\neg(\text{thinking1}=3)$:
Utiliza la orden `[] !(place thinking1 equals 3)`.
 - $FG\neg(\text{thinking1}=3)$:
Utiliza la orden `<>[] !(place thinking1 equals 3)`.
 - $GF\neg(\text{thinking1}=3)$:
Utiliza la orden `[]<> !(place thinking1 equals 3)`.
 - $F(\text{hungry1}=0)$:
Utiliza la orden `<> (place hungry1 equals empty)`.

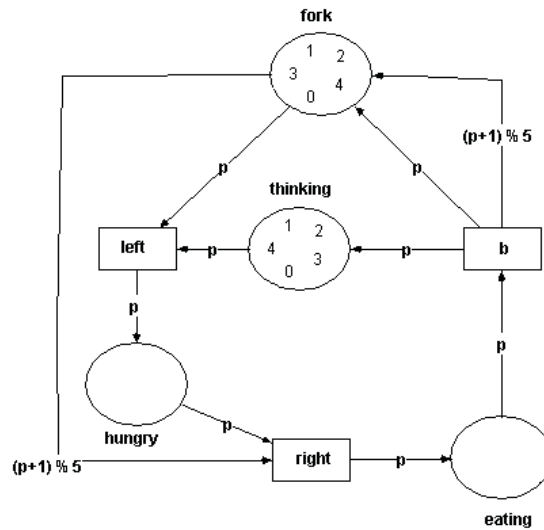


Figure 3: Red de Petri coloreada para representar el problema de los filósofos. dining2.pn

- $G(\text{hungry1}=0)$:
Utiliza la orden [] (place hungry1 equals empty).
- $GF(\text{hungry1} = 1)$ or $(\neg GF(\text{place hungry1}=1))$:
Utiliza la orden []<> (place hungry1 equals 1) or ! []<> (place thinking1 equals 1).
- **Cuestión 3.** ¿Qué preguntas devuelven un resultado distinto al caso anterior? ¿Por qué?
- Ejecuta en María el modelo dining2.pn (Figura 3) comprobando si tiene bloqueos.
- María responde que el modelo tiene un bloqueo en el estado 71. Muestra este estado.
- **Cuestión 4.** ¿De qué estado se trata?
- Verifica si el modelo dining2.pn cumple las siguientes propiedades de lógica temporal.
 - $F\neg(\text{thinking}=3)$:
Utiliza la orden <> !(place thinking equals 3).

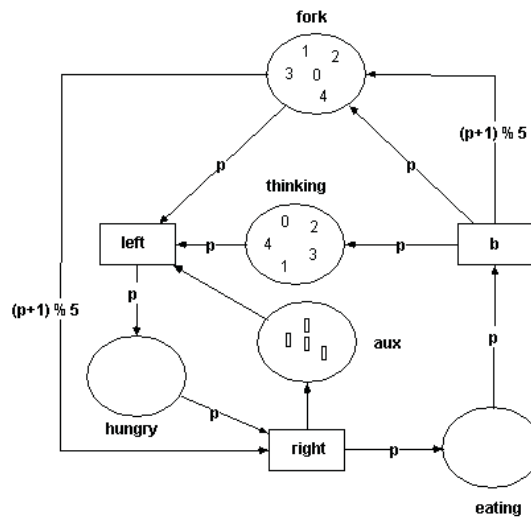


Figure 4: Red de Petri coloreada para representar el problema de los filósofos. dining2NB.pn

- $G\neg(\text{thinking}=3)$:
Utiliza la orden `[] !(place thinking equals 3)`.
 - $F\neg(\text{hungry}=0)$:
Utiliza la orden `<> (place hungry equals empty)`.
 - $G\neg(\text{hungry}=0)$:
Utiliza la orden `[] (place hungry equals empty)`.
 - $GF(\text{hungry} = 1)$ or $(\neg GF(\text{place hungry}=1))$:
Utiliza la orden `[]<> (place hungry equals 1)` or `![]<> (place thinking equals 1)`.
- **Cuestión 5.** Explica el significado y resultado de dos de las siguientes preguntas.
 - Ejecuta en María el modelo dining2NB.pn (Figura 4) comprobando si tiene bloqueos.
 - Verifica si el modelo dining2NB.pn cumple las siguientes propiedades de lógica temporal.
 - $F\neg(\text{thinking}=3)$:
Utiliza la orden `<> !(place thinking equals 3)`.

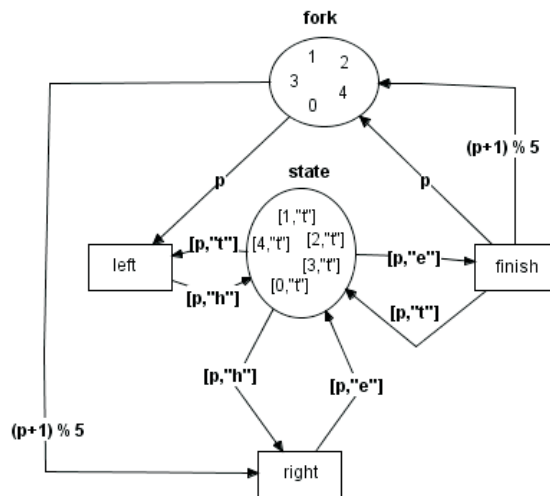


Figure 5: Red de Petri coloreada para representar el problema de los filósofos.

- $G\neg(\text{thinking}=3)$:
Utiliza la orden [] !(place thinking equals 3).
- $F\neg(\text{hungry}=0)$:
Utiliza la orden <> (place hungry equals empty).
- $G\neg(\text{hungry}=0)$:
Utiliza la orden [] (place hungry equals empty).
- $GF\neg(\text{hungry} = 1) \text{ or } (\neg GF (\text{place hungry}=1))$:
Utiliza la orden []<> (place hungry equals 1) or !
[]<> (place thinking equals 1).

- **Cuestión 6.** ¿Qué preguntas devuelven un resultado distinto al caso anterior?
¿Por qué?

Parte 2

- Modela en María la red de Petri coloreada representada en la Figura 5. Esta red corresponde a un modelado del problema de los filósofos utilizando redes de Petri de alto nivel. Guarda el fichero con el nombre dining3.pn.

Ayuda. Utiliza el siguiente código para generar la estructura de datos utilizada en el lugar state y el lugar state.

```
typedef struct {
    phil_t p,
    enum { thinking, hungry, eating } s
} status;

place state (#phil_t) status: phil_t p: { p, thinking };
```

- **Cuestión 7.** Da una explicación sobre el comportamiento del modelo.

Parte 3

Modifica el modelo obtenido en la parte 1 de la práctica para evitar el bloqueo del sistema. Guarda el fichero con el nombre `dining3NB.pn`.

Parte 4

Modifica el modelo obtenido en la parte 2 de la práctica para representar el mismo problema pero ampliándolo a 2, 4, 6, 8, 10, 12 filósofos.

- **Cuestión 8.** ¿Cómo crece el número de estados generado en María en relación con el crecimiento del número de filósofos? ¿Qué ocurre con el tiempo de ejecución de los nuevos modelos? Da una explicación sobre el comportamiento.

3 Entrega de material

- Como resultado de esta práctica hay que entregar el fichero `dining3NB.pn` y el fichero `practicaMC.txt` contestando las cuestiones. Se entregará a la dirección de correo `mjibanez@unizar.es`.
- Fecha límite de entrega será el 15 de Mayo.