Chapter 6

Logical properties of P/T systems and their analysis

6.1 Basic logical properties

Only a few qualitative properties will be considered in this introductory chapter. They are general in the sense that they are meaningful for any concurrent system, not only for those modeled with Petri nets. Nevertheless, their statements using Petri net concepts and objects make them specially "easy to understand" in many cases. The properties to be considered are:

1) *boundedness*, characterising finiteness of the state space.

2) *liveness*, related to potential fireability in all reachable markings. *Deadlock-freeness* is a weaker condition in which only global infinite activity (i.e. fireability) of the net system model is guaranted, even if some parts of it do not work at all.

3) *reversibility*, characterizing recoverability of the initial marking from any reachable marking.

4) *mutual exclusion*, dealing with the impossibility of simultaneous *submarkings* (p-mutex) or *firing concurrency* (t-mutex).

Consider the net in Figure 6.1.a. Firing t_2 leads to $\mathbf{m} = p_3 + p_4$. Firing now t_4, $\mathbf{m}_1 = p_1 + p_3$ is reached. Repeating ω times the sequence $t_2 t_4$ the marking $\mathbf{m}_\omega = p_1 + \omega\, p_3$ is reached. So the marking of p_3 can be arbitrarily large, place p_3 is said to be *unbounded*. In practice, the capacity of the physical element represented by p_3 should be finite, so an *overflow* can appear, which is a pathological situation.

The maximum number of tokens a place may contain is its (marking) *bound*. A place is bounded if its bound is finite. A net system is bounded if each place is

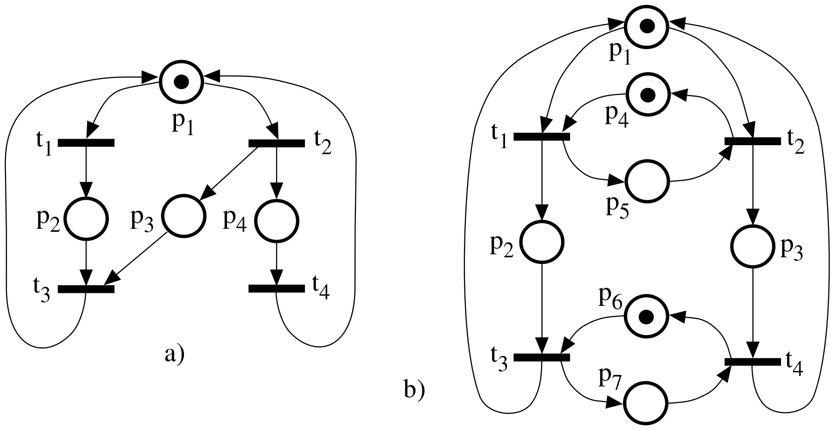


Figure 6.1: On qualitative pathological behaviors: (a) an unbounded, deadlockable (non-live), non-reversible net system; (b) increasing the initial marking (e.g. $\mathbf{m_0}[p_5] = 1$) the live net system can reach a deadlock state!

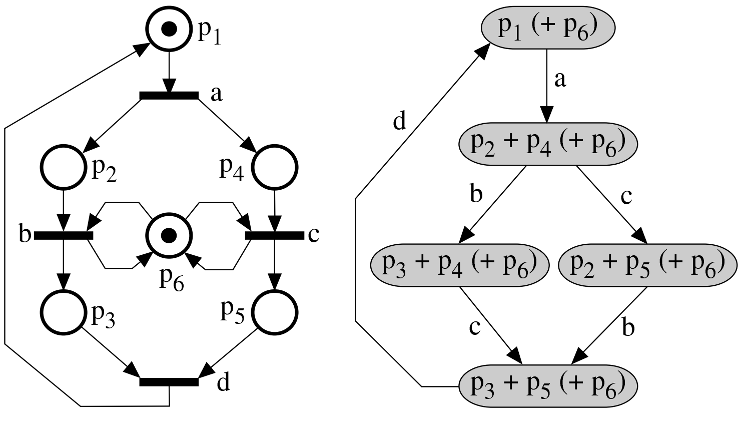


Figure 6.2: Bounded, live and reversible system and its reachability graph

bounded. System boundedness (i.e. all places bounded) is a generally required behavioural property.

For any initial marking we can define on the net structure of Figure 6.2a the following token conservation laws:

$$\begin{aligned}
\mathbf{m}[p_1] + \mathbf{m}[p_2] + \mathbf{m}[p_3] = \mathbf{m_0}[p_1] + \mathbf{m_0}[p_2] + \mathbf{m_0}[p_3] = k_1(\mathbf{m_0}) \\
\mathbf{m}[p_1] + \mathbf{m}[p_4] + \mathbf{m}[p_5] = \mathbf{m_0}[p_1] + \mathbf{m_0}[p_4] + \mathbf{m_0}[p_5] = k_2(\mathbf{m_0}) \\
\mathbf{m}[p_6] = \mathbf{m_0}[p_6] = k_3(\mathbf{m_0})
\end{aligned}$$

where $\mathbf{m_0}$ is the initial marking and $\mathbf{m}$ any reachable marking. Therefore:

$$\begin{aligned}
\mathbf{m}[p_1] \leq \min(k_1(\mathbf{m_0}), k_2(\mathbf{m_0})) \\
\mathbf{m}[p_i] \leq k_1(\mathbf{m_0}); i = 2, 3 \\
\mathbf{m}[p_j] \leq k_2(\mathbf{m_0}); j = 4, 5 \\
\mathbf{m}[p_6] = k_3(\mathbf{m_0})
\end{aligned}$$

The above inequalities mean that *for any* $\mathbf{m_0}$ the net system is bounded. This property, stronger than boundedness, is called *structural boundedness* because it holds independently of the initial marking (only finiteness of $\mathbf{m_0}$ is assumed).

Let us consider now a different scenario where we fire t_1 from the marking in Figure 6.1a. After that, no transition can be fired: a *total deadlock* situation has been reached. A net system is said to be *deadlock-free* if always (i.e. from any reachable marking) at least one transition can occur. A stronger condition than deadlock-freeness is *liveness*. A transition t is potentially fireable at a given marking $\mathbf{m}$ if there exists a transition firing sequence σ leading to a marking $\mathbf{m}'$ in which t is enabled (i.e. $\mathbf{m}\xrightarrow{\sigma}\mathbf{m}' \geq \mathbf{Pre}[P, t]$). A transition is *live* if it is potentially fireable in all reachable markings. In other words, a transition is live if it never loses the possibility of firing (i.e. of performing some activity). A net system is live if all the transitions are live.

For any initial marking we can define on the net structure in Figure 6.1a, non liveness holds (in fact, a total deadlock can always be reached). Non-liveness for arbitrary initial markings reflect a pathology of the net structure: *structural non-liveness*. A net is structurally live if there exists at least one live initial marking.

At first glance it may be accepted as intuitive that increasing the initial marking (e.g. increasing the number of resources) of a net system "helps" in making it live. A paradoxical behaviour of concurrent systems is the following: The net system in Figure 6.1b shows that increasing the number of resources can lead to deadlock situations (Adding a token to p_5, t_2 can be fired and a deadlock is reached!). In other words, in general, liveness is not *monotonic* with respect to the initial marking. Nevertheless, it can be pointed out that liveness can be marking-monotonic on certain net subclasses.

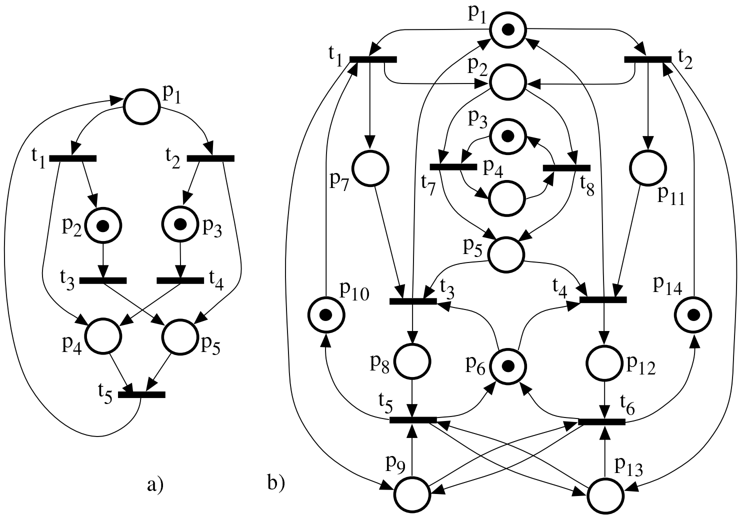


Figure 6.3: On home states: (a) The initial marking is not a home state, but all successor markings are home states; (b) Net system that presents two livelocks, so there are no home states.

A marking is a *home state* if it is reachable from any other reachable marking. The initial marking of the net system in Figure 6.3a is not a home state: after the firing of transition t_3 or t_4 it is not possible to recover this initial marking. Nevertheless, each one of the reachable markings from the initial one is a home state. For some subclasses of net systems the existence of home states is guaranteed [45, 40], but in general the existence of home states does not hold. The net system in Figure 6.3.b [6] is live and bounded but there are no home states. In fact, there exist two different terminal live behaviours that are mutually unreachable. Each one of these terminal live behaviours is called a *livelock*. The set of home states of a net system is called the *home space*. The existence of a home space for a net system is a desirable property because it is strongly related to properties such as ergodicity, of crucial importance in the context of performance evaluation or system simulation.

In the particular case that the initial marking is a home state, the net system is reversible, so it is always possible to return to the initial marking.

Liveness, boundedness, and reversibility are just three different "good" (often required) behavioural properties that may be interesting to study in a net system. Figure 6.4 shows that they are independent of each other, giving examples of the eight cases we may have.

The last basic property we introduce in this section is *mutual exclusion*. This property captures constraints like the impossibility of a simultaneous access by two robots to a single store. Two places (transitions) are in mutual exclusion if can they never be simultaneously marked (fired). For instance, in the net system in Figure 6.2 we can write: $\mathbf{m}[p_1] + \mathbf{m}[p_2] + \mathbf{m}[p_3] = 1$, so p_1, p_2, and

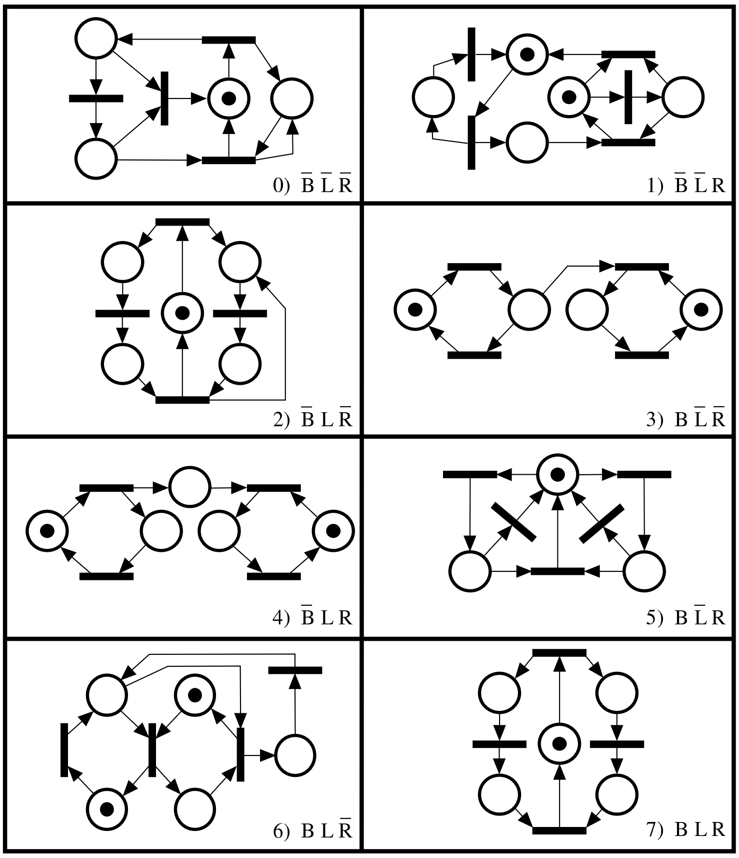

0) $\overline{B}\,\overline{L}\,\overline{R}$
1) $\overline{B}\,\overline{L}\,R$
2) $\overline{B}\,L\,\overline{R}$
3) $B\,\overline{L}\,\overline{R}$
4) $\overline{B}\,L\,R$
5) $B\,\overline{L}\,R$
6) $B\,L\,\overline{R}$
7) $B\,L\,R$

Figure 6.4: Boundedness (B), liveness (L) and reversibility (R) are independent properties

(1)	Bound of place p in $\langle \mathcal{N}, \mathbf{m_0} \rangle$ $\mathbf{b}(p) = \sup\{\mathbf{m}[p] \mid \mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})\}$
(2)	p is bounded in $\langle \mathcal{N}, \mathbf{m_0} \rangle$ iff $\mathbf{b}(p) < \infty$
(3)	$\langle \mathcal{N}, \mathbf{m_0} \rangle$ is bounded if all places are bounded
(4)	$\langle \mathcal{N}, \mathbf{m_0} \rangle$ is deadlock-free iff $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ $\exists t \in T$ such that t is fireable at $\mathbf{m}$
(5)	t is live in $\langle \mathcal{N}, \mathbf{m_0} \rangle$ iff $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ $\exists \sigma$ such that $\mathbf{m} \xrightarrow{\sigma t} \mathbf{m}'$
(6)	$\langle \mathcal{N}, \mathbf{m_0} \rangle$ is live if all transitions are live
(7)	$\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ is a home state iff $\forall \mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ $\exists \sigma$ such that $\mathbf{m}' \xrightarrow{\sigma} \mathbf{m}$
(8)	$\langle \mathcal{N}, \mathbf{m_0} \rangle$ is reversible iff $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ $\exists \sigma$ such that $\mathbf{m} \xrightarrow{\sigma} \mathbf{m_0}$
(9)	Mutual exclusion in $\langle \mathcal{N}, \mathbf{m_0} \rangle$: p_i and p_j are in marking mutual exclusion iff $\nexists$ $\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ such that $(\mathbf{m}[p_i] > 0)$ and $(\mathbf{m}[p_j] > 0)$ t_i and t_j are in firing mutual exclusion iff $\nexists$ $\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ such that $\mathbf{m} \geq \mathbf{Pre}[P, t_i] + \mathbf{Pre}[P, t_j]$
(10)	Structural properties (abstractions of behavioural properties): $\mathcal{N}$ is structurally bounded iff $\forall \mathbf{m_0}$ (finite) $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is bounded $\mathcal{N}$ is structurally live iff $\exists \mathbf{m_0}$(finite) making $\langle \mathcal{N}, \mathbf{m_0} \rangle$ a live system

Table 6.1: Summarising some basic logical properties

p_3 are in mutual exclusion.

Table 6.1 summarises the definitions of the different properties we introduced in this section.

6.2 Basic analysis techniques for P/T net systems

Conventionally, analysis techniques for Petri nets, are classified as: (1) Enumeration; (2) Transformation; and (3) Structural analysis. Simulation methods have also been applied to study systems modeled with P/T nets. They proceed playing the token game (firing enabled transitions) on the net system model under certain strategies. In general, simulation methods do not allow to prove properties, but they might be of great help for understanding the modeled system or to fix the manifested problems during simulation. Simulation methods are extremely useful when time is associated with the net evolution (timed systems), or when we wish to know the response of the system described with a net in an environment which is also defined by simulation (see Part VII of this book). In this section we do not consider simulation methods and we will only overview the three previously mentioned analysis techniques on P/T nets without interpretation.

Enumeration methods are based on the construction of a *reachability graph* (RG) which represents, individually, the net markings and single transition firings between them. If the net system is bounded, the reachability graph is finite and the different qualitative properties can be easily verified. If the net system is unbounded, the RG is infinite and its construction is not possible. In this case, finite graphs known as *coverability graphs* can be constructed (see, for example, [31, 33, 18]). In spite of its power, enumeration is often difficult to apply, even in small nets, due to its computational complexity (it is strongly combinatory).

Analysis by transformation proceeds transforming a net system $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ into a net system $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m_0}' \rangle$ preserving the set of properties Π to be verified (i.e. $\mathcal{S}'$ satisfies the properties Π iff $\mathcal{S}$ satisfies them). The final goal is to verify the properties Π in $\mathcal{S}'$ in a more easy way than in $\mathcal{S}$. The state space of $\mathcal{S}'$ may be bigger than that of $\mathcal{S}$, but $\mathcal{S}'$ may belong to a subclass for which state enumeration can be avoided.

Reduction methods are a special class of transformation methods in which a sequence of net systems preserving the properties to be studied is constructed. The construction is done in such a way that the net system $\langle \mathcal{N}_{i+1}, \mathbf{m_0}_{i+1} \rangle$ is "smaller" (less markings or maintaining the reachability set it has less places or transitions) than the previous in the sequence, $\langle \mathcal{N}_i, \mathbf{m_0}_i \rangle$.

The applicability of reduction methods is limited by the existence of irreducible net systems. Practically speaking, the reductions obtained are normally considerable, and can allow the desired properties to be verified directly. Because of the existence of irreducible systems, this method must be complemented by others.

Finally, *structural analysis techniques* investigate the relationships between the behaviour of a net system and its structure (hence their name), while the initial marking acts, basically, as a parameter. In this last class of analysis techniques, we can distinguish two subgroups:

1) *Linear algebra / Linear programming based techniques*, which are based on the net state equation. In certain analysis they permit a fast diagnosis without the necessity of enumeration.

2) *Graph based techniques*, in which the net is seen as a bipartite graph and some "ad hoc" reasonings (frequently derived from the firing rule) are applied. These methods are especially effective in analysing restricted subclasses of ordinary nets.

The three groups of analysis techniques outlined above are by no means exclusive, but complementary. Normally the designer can use them according to the needs of the ongoing analysis process. Obviously, although we have distinguished between transformation/reduction and structural analysis methods, it must be pointed out that most popular reduction techniques act basically on the net structure level and thus can be considered also as structural techniques.

For what concerns the qualitative analysis of interpreted systems, it should be pointed out that conclusions about the properties of the underlying autonomous model can be only sufficient (e.g. for boundedness), necessary (e.g. for reachability) or neither sufficient nor necessary (e.g. for liveness). For particular net subclasses under "reasonable assumptions" on the behaviour of the environment necessary and sufficient conditions exist.

6.3 Analysis based on the reachability graph

Given a net system, $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$, its reachability graph (recall definition 2.5.4) is a directed graph, $\mathrm{RG}(\mathcal{S}) = (V, E)$, where $V = \mathrm{RS}(\mathcal{S})$ and $E = \{\langle \mathbf{m}, t, \mathbf{m}' \rangle | \mathbf{m}, \mathbf{m}' \in \mathrm{RS}(\mathcal{S})$ and $\mathbf{m} \xrightarrow{t} \mathbf{m}'\}$ are the sets of nodes and edges, respectively.

If the net system $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ is bounded, the $\mathrm{RG}(\mathcal{S})$ is finite and it can be constructed, for example, by the algorithm 6.1. It finishes when all the possible firings from the reachable markings have been explored. The tagging scheme in step 2.1 ensures that no marking is visited more than once. Each marking visited is tagged (step 2.1), and step 2.2.3 ensures that the only markings added to V are ones that have not previously added. When a marking is visited, only those edges representing the firing of an enabled transition are added to the set E in 2.2.4.

Let us consider, for example, the net system in Figure 6.2 without the place p_6 and its reachability graph, obtained by applying the algorithm 6.1,. The net system has five markings, thus it is bounded. It is also easy to conclude that all places are 1-bounded. A closer look allows to state that p_1, p_2 and p_3 (p_1, p_4 and p_5) are in mutual exclusion. Moreover, considering RS and the net structure (the pre-function), firing concurrency between transitions b and c can be decided. Observe at this point that introducing p_6 in our net system, the reachability graph does not change, but transitions b and c become in firing mutual exclusion. This example shows that the obtained RG is a *sequentialised observation* of the net system behaviour.

For unbounded net systems, $\mathcal{S}$, $\mathrm{RS}(\mathcal{S})$ is not a finite set and therefore the construction of $\mathrm{RG}(\mathcal{S})$ never ends. Karp and Miller [23] showed how to detect unboundedness of a net system by means of the following condition (incorporated in step 2.2.2 of algorithm 6.1 as a break condition): the system $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ is unbounded iff there exists $\mathbf{m}'$ reachable from $\mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $\mathbf{m} \xrightarrow{\sigma} \mathbf{m}'$, such that $\mathbf{m} \lneq \mathbf{m}'$ (the repetition of σ allows to conclude on unboundedness because the occurrence of σ strictly increases the content of tokens of the starting marking $\mathbf{m}$).

Coverability graphs allow to obtain finite representations of the RG of unbounded net systems [23, 31, 33, 18]. Roughly speaking, in a coverability graph the set of nodes is a finite set of marking vectors (called the coverability set) that covers all the markings of the reachability set. There is an edge, representing the firing of a transition t, between two nodes, $\mathbf{m}$ and $\mathbf{m}'$ if and only if t is fireable from $\mathbf{m}$ and a marking covered by $\mathbf{m}'$ is reached. The loss of information in the computation of a coverability graph makes that many important properties (e.g. marking reachability or deadlock freeness) cannot be decided on it.

In order to analyse a given property in a bounded net system, the reachability graph is used as the basis for the corresponding *decision procedure*. It allows to

Algorithm 6.1 (Computation of the Reachability Graph)

```
    Input - The net system $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$
    Output - The directed graph $\mathrm{RG}(\mathcal{S}) = (V, E)$ for bounded net systems

1.  Initialize $\mathrm{RG}(\mathcal{S}) = (\{\mathbf{m_0}\}, \emptyset)$; $\mathbf{m_0}$ is untagged;
2.  while there are untagged nodes in $V$ do
    2.1 Select an untagged node $\mathbf{m} \in V$ and tag it
    2.2 for each enabled transition, $t$, at $\mathbf{m}$ do
        2.2.1 Compute $\mathbf{m}'$ such that $\mathbf{m} \xrightarrow{t} \mathbf{m}'$;
        2.2.2 if there exists $\mathbf{m}'' \in V$ such that $\mathbf{m}'' \xrightarrow{\sigma} \mathbf{m}'$ and $\mathbf{m}'' \lneq \mathbf{m}'$
              then the algorithm fails and exits;
                   (the unboundedness condition of $\mathcal{S}$ has been detected)
        2.2.3 if there is no $\mathbf{m}'' \in V$ such that $\mathbf{m}'' = \mathbf{m}'$
              then $V := V \cup \{\mathbf{m}'\}$; ($\mathbf{m}'$ is an untagged node)
        2.2.4 $E := E \cup \{\langle \mathbf{m}, t, \mathbf{m}' \rangle\}$
3.  The algorithm succeds and $\mathrm{RG}(\mathcal{S})$ is the reachability graph
```

decide whether the net system satisfies a given property. All procedures are, in general, of exponential complexity in the size of the net (measured, for example, by the number of places) but they are of polynomial complexity on the size of the reachability graph (measured, for example, by the number of nodes and arcs). The focus of the rest of this section is in two general decision procedures.

In the sequel we will call *marking predicate* to a propositional formula whose atoms are inequalities of the form: $\sum_{p \in A} k_p \mathbf{m}[p] \leq k$, where k_p and k are rational constants and A is a subset of places. Let us consider a net system $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$.

The first group of properties are the so called *marking invariance properties*. A given marking predicate, Π, must be satisfied for all reachable markings (hence the name of marking invariance property): $\forall \mathbf{m} \in \mathrm{RS}(\langle \mathcal{N}, \mathbf{m_0} \rangle)$, $\mathbf{m}$ satisfies Π. Examples of this are:

1) *k-boundedness of place p*: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $\mathbf{m}[p] \leq k$.

2) *Marking mutual exclusion between p and p'*: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $(\mathbf{m}[p] = 0) \vee (\mathbf{m}[p'] = 0)$.

3) *Deadlock-freeness*: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $\bigvee_{t \in T} (\mathbf{m} \geq \mathbf{Pre}[P, t])$.

Marking invariance properties can be decided through Algorithm 6.2, which is linear in the size of $\mathrm{RS}(\mathcal{S})$: each node is visited no more than once. If the algorithm succeeds, then all reachable markings from $\mathbf{m_0}$ satisfy Π. If the algorithm fails at step 2.2, there is a path in the $\mathrm{RG}(\mathcal{S})$ from $\mathbf{m_0}$, containing at least a marking that does not satisfy Π.

Example 6.1 (Analysis of marking invariances) Let us consider the net system in figure 6.2 for which $\mathrm{RS}(\mathcal{S}) = \{p_1 + p_6,\ p_2 + p_4 + p_6,\ p_3 + p_4 + p_6,$

Algorithm 6.2 (Decision procedure for marking invariances)

```
    Input -  The reachability set $\mathrm{RS}(\mathcal{S})$. The property $\Pi$.
    Output -  TRUE if the property is verified.

1.  Initialise all elements of $\mathrm{RS}(\mathcal{S})$ as untagged.
2.  while there is an untagged node $\mathbf{m} \in \mathrm{RS}(\mathcal{S})$ do
    2.1 Select an untagged node $\mathbf{m} \in \mathrm{RS}(\mathcal{S})$ and tag it
    2.2 if $\mathbf{m}$ does not satisfy $\Pi$
        then return FALSE (the property is not verified).
3.  Return TRUE
```

Algorithm 6.3 (Decision procedure for liveness invariances)

```
    Input -  The reachability graph $\mathrm{RG}(\mathcal{N}, \mathbf{m_0})$. The property $\Pi$
    Output -  TRUE if the property is verified.

1.  Decompose $\mathrm{RG}(\mathcal{N}, \mathbf{m_0})$ into its strongly connected components $C_1, \ldots, C_r$
2.  Obtain the graph $\mathrm{RG}^c(\mathcal{S}) = (V_c, E_c)$ by shrinking $C_1, \ldots, C_r$ to a single
    node, i.e. $V_c = \{C_1, \ldots, C_r\}$. $\langle C_i, t, C_j \rangle \in E_c$ iff there exists $\langle \mathbf{m}, t, \mathbf{m}' \rangle \in E$,
    such that $\mathbf{m}$ is in the SCC $C_i$, $\mathbf{m}'$ is in the SCC $C_j$, and $i \neq j$.
3.  Compute the set F of terminal strongly connected components from $\mathrm{RG}^c(\mathcal{S})$
4.  while there is a $C_i \in F$ do
    3.1 if $C_i$ it does not contain a $\mathbf{m}'$ satisfying $\Pi$
        then return FALSE
    3.2 Remove $C_i$ from $F$
5.  Return TRUE
```

$p_2 + p_5 + p_6$, $p_3 + p_5 + p_6\}$. The execution of Algorithm 6.2 to verify the mutual exclusion property between places p_5 and p_6 ($\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $(\mathbf{m}[p_5] = 0) \vee (\mathbf{m}[p_6] = 0)$) starts initialising all elements of $\mathrm{RS}(\mathcal{S})$ as untagged (step 1). Then the markings are visited one by one (e.g. in the previous order) until $p_2 + p_5 + p_6$ is visited, where the predicate Π is false, hence the algorithm stops and return FALSE.

The second group of properties are the so called *liveness invariance properties*. For each reachable marking of a net system, $\mathbf{m}$, there exists at least a reachable marking from it satisfying the property Π: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $\exists \mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m})$, $\mathbf{m}'$ satisfies Π. Examples of this are:

1) *Liveness of t*: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $\exists \mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m})$ such that $\mathbf{m}' \geq \mathbf{Pre}[P, t]$.

2) $\mathbf{m}_H$ *is home state*: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $\exists \mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m})$ such that $\mathbf{m}' = \mathbf{m}_H$.

3) *Reversibility*: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $\exists \mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m})$ such that $\mathbf{m}' = \mathbf{m_0}$.

These properties cannot be verified by an exclusive linear inspection of the reachability set (as in algorithm 6.2). The verification requires to find a reachable marking, satisfying Π, from each one of the markings in $\mathrm{RS}(\mathcal{S})$. In order to verify the property we will classify the markings of $\mathrm{RS}(\mathcal{S})$ into subsets of mutually reachable markings through the concept of strongly connected component of a directed graph. Therefore, the property will be easily verified checking that each terminal strongly connected component contains at least a marking satisfying Π. We recall now some basic concepts.

A *path* in a reachability graph $\mathrm{RG}(\mathcal{S})$ is any sequence $\mathbf{m}_1 \ldots \mathbf{m}_i \mathbf{m}_{i+1} \ldots \mathbf{m}_k$ of nodes of $\mathrm{RG}(\mathcal{S}) = (V, E)$ where all succesive nodes $\mathbf{m}_i$ and $\mathbf{m}_{i+1}$ in the path satisfy that $\langle \mathbf{m}_i, t, \mathbf{m}_{i+1} \rangle \in E$ for some t. The reachability graph, $\mathrm{RG}(\mathcal{S})$, is *strongly connected* iff there is a path from each node in V to any other node in V. A *strongly connected component* (SCC) of a reachability graph is a maximal strongly connected subgraph. A strongly connected component of a graph will be called *terminal* if no node in the component has an edge leaving the component. The strongly connected components of a digraph (V, E) can be found in order $(|V| + |E|)$ steps (e.g. [26]).

When computing the SCCs $C_1, \ldots, C_r$ of a reachability graph $\mathrm{RG}(\mathcal{S}) = (V, E)$, a new graph $\mathrm{RG}^c(\mathcal{S}) = (V_c, E_c)$ is induced by shrinking the strongly connected components to a single node, i.e. $V_c = \{C_1, \ldots, C_r\}$. For each edge $\langle \mathbf{m}, t, \mathbf{m}' \rangle \in E$, such that $\mathbf{m}$ is in a SCC C_i, and $\mathbf{m}'$ is in a different SCC C_j, there is an induced edge $\langle C_i, t, C_j \rangle \in E_c$. The graph $\mathrm{RG}^c(\mathcal{S})$ is an acyclic digraph. Therefore the terminal SCCs of $\mathrm{RG}(\mathcal{S})$ can be identified with linear complexity in the size of $\mathrm{RG}^c(\mathcal{S})$. This fact will be exploited in the algorithm 6.3 for liveness invariance checking.

Algorithm 6.3 allows to decide liveness invariance properties. The algorithm is of linear complexity in the size of the $\mathrm{RG}(\mathcal{S})$. If the algorithm succeeds, all terminal SCCs contain at least one marking satisfying the property Π, and therefore for all reachable marking there exists at least a successor marking satisfyig the property Π. If the algorithm fails, there exists at least one terminal SCC that does not contain markings satisfying the property Π, and therefore the reachable markings belonging to this SCC (at least) do not satisfy the liveness invariance property.

Remark It is possible to design more specific (efficient) decision procedures for the analysis of a property if we know, a priori, some characteristics of the property to be verified or we know some other properties of the net system to be analysed. For the first case we can consider as an example the reversibility property. It is easy to see that if a net system is reversible then all terminal SCCs must contain the initial marking, i.e. the reachability graph must be strongly connected. For the second case, for example, we may know that the net system is reversible; then liveness of a transition t can be decided checking the existence of an edge in the reachability graph labeled t (since the reachability graph is SC and therefore always is possible to reach the marking from which t can be fired).

Example 6.2 (Analysis of liveness invariances) Let us consider the net system in Figure 6.3.b for which the reachability graph is depicted in Figure

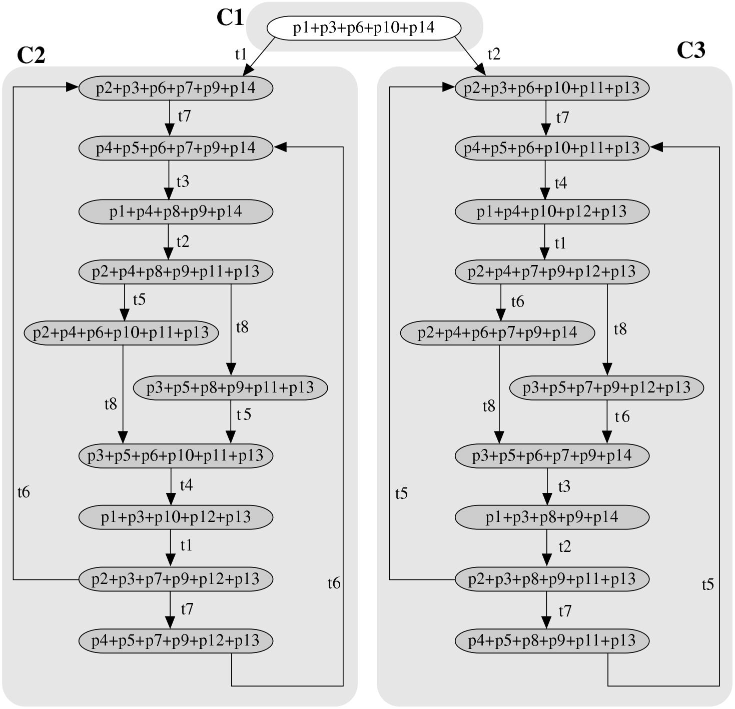


Figure 6.5: Reachability graph of the net system in Figure 6.3.b

6.5. The execution of the algorithm 6.3 to verify the liveness property of this net system ($\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $\bigwedge_{t \in T} [\exists \mathbf{m}^t \in \mathrm{RS}(\langle \mathcal{N}, \mathbf{m} \rangle), \mathbf{m}^t \geq \mathbf{Pre}[P,t]]$) requires the computation of the strongly connected components of the $\mathrm{RG}(\mathcal{S})$ (step 1). In this case, there are three SCCs depicted in Figure 6.5 and named C_1, C_2 and C_3. The SCCs C_2 and C_3 are the terminal ones. The step 4 of the algorithm will verify that each one of these two SCCs contains for each transition t a marking $\mathbf{m}^t$ satisfying $\mathbf{m}^t \geq \mathbf{Pre}[P,t]$ (equivalently, contains edges labelled with all transitions of the net). The reader can observe by inspection of the figure that all transitions appear in some edge of C_2 and C_3, therefore the answer of the algorithm will be TRUE.

The execution of the algorithm 6.3 to verify that the marking $\mathbf{m}_H = p_2 + p_3 + p_6 + p_7 + p_9 + p_{14}$ is a home state ($\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $\exists \mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m})$ such that $\mathbf{m}' = \mathbf{m}_H$) gives as result FALSE, because the terminal SCC C_2 contains the marking $\mathbf{m}_H$, but the terminal SCC C_3 does not. Therefore, step 3.1 returns FALSE.

From a practical point of view, it is commonly accepted today that systems are too complex to be verified by hand. As a result, analysis increasingly is becoming synonymous with *computer-aided verification* [1]. Computer-aided verification means using a computer, for increased speed and reliability, to perform the analysis steps. For instance, the following example considers the analysis of a property belonging to the group of the so called *synchronic properties* [37], pointing out that an analysis by hand can be very hard.

Example 6.3 Figure 6.6 shows a very simple net system: Parts are sent from *STORE 1* to *STORE 2* and *STORE 3*. The subnet generated by places $\{B, C, E, F\}$ imposes some restrictions on the way parts are distributed to the destination stores (i.e. partially schedule the distribution). The reachability graph is, even if it has been "structured" for more clear presentation, difficult to understand and manage. The reader can try to check on the reachability graph (!) that the imposed distribution strategy is: parts are sent in a 1:1 relation to the destination stores, but allowing sometimes until four consecutive sendings to a given store (i.e. locally adjusting the possible demand, but maintaining the overall fair distribution).

Summarising, analysis techniques based on the reachability graph are only theoretically possible for bounded systems. They are very simple from a conceptual point of view. The problem that makes this approach not practical (impossible) in many cases is its computational complexity: *the state space explosion problem.*

On the other hand, it must be pointed out that the reachability/coverability graphs are computed for a given initial marking. This means that a parametric analysis of a net system (needed in earlier phases of the system design) where the initial marking of some places (e.g. representing the number of resources in the system) is a parameter, is not possible since for each value of the parameter a (completely different) new reachability graph must be computed. Moreover, the reachability graph presents some difficulties in order to analyze properties

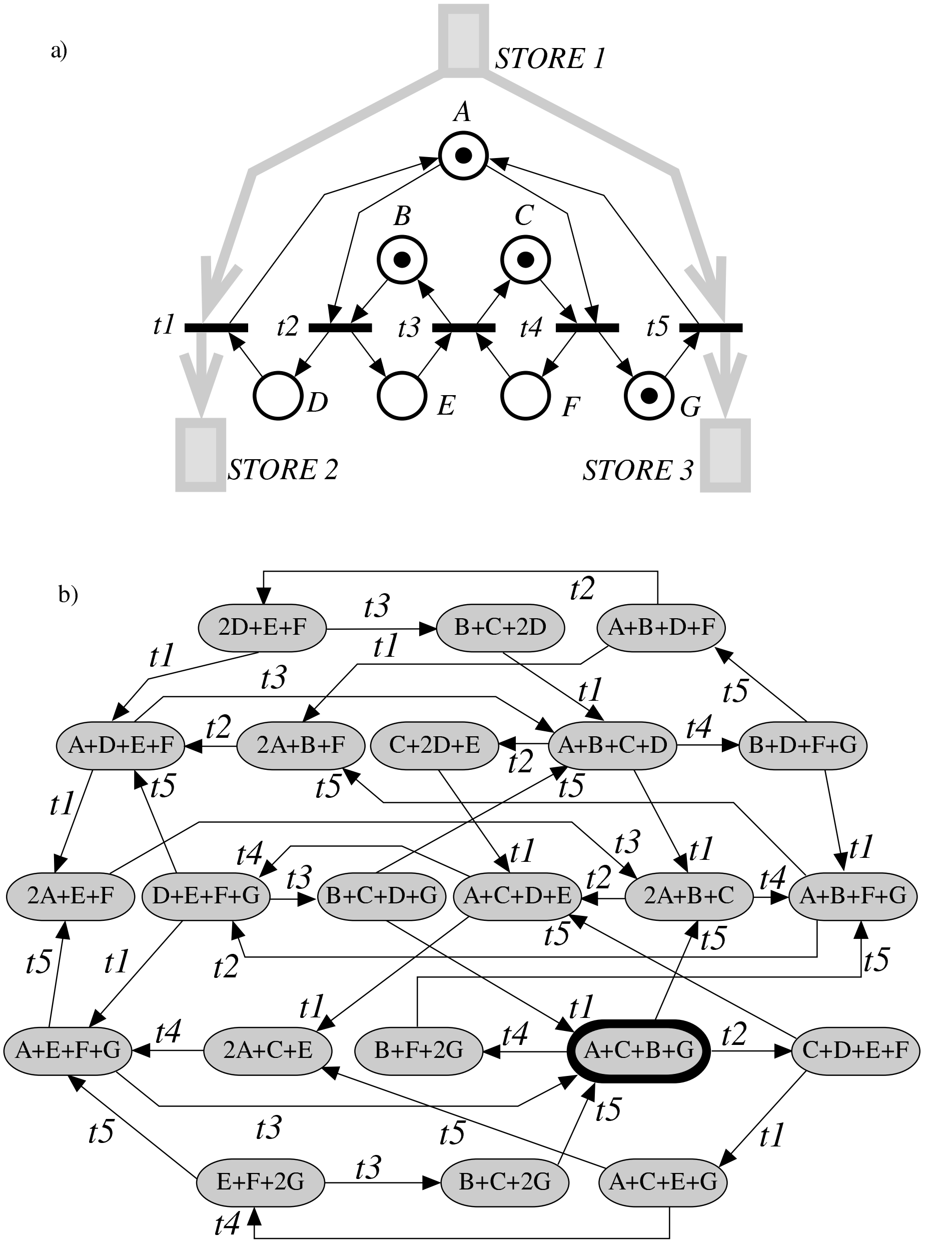


Figure 6.6: Parts of *STORE 1* are sent to *STORE 2* and *STORE 3* according to the strategy defined by the subnet generated by $\{B, C, E, F\}$: (a) the net system; (b) the reachability graph.

where the distinction between conflict and concurrency plays a fundamental role (recall the net in figure 6.2, the reachability graph is the same with place p_6 and without it!). This is because the reachability graph gives a sequentialized view of the behaviour of the net system.

Although these analysis techniques present the drawbacks above mentioned, for bounded net systems they are the more general ones and, in some cases, the only way to verify a given property.

6.4 Net system reductions

In order to paliate the state space explosion problem several techniques has been introduced to obtain *reduced state spaces*. As an example we can cite the stubborn set method [43, 44]. These techniques work directly in the construction phase of the reachability graph maintaining the original net model. In this section we review a different kind of reduction techniques named *net system reductions*. These reductions proceed transforming the net structure and, sometimes, the initial marking.

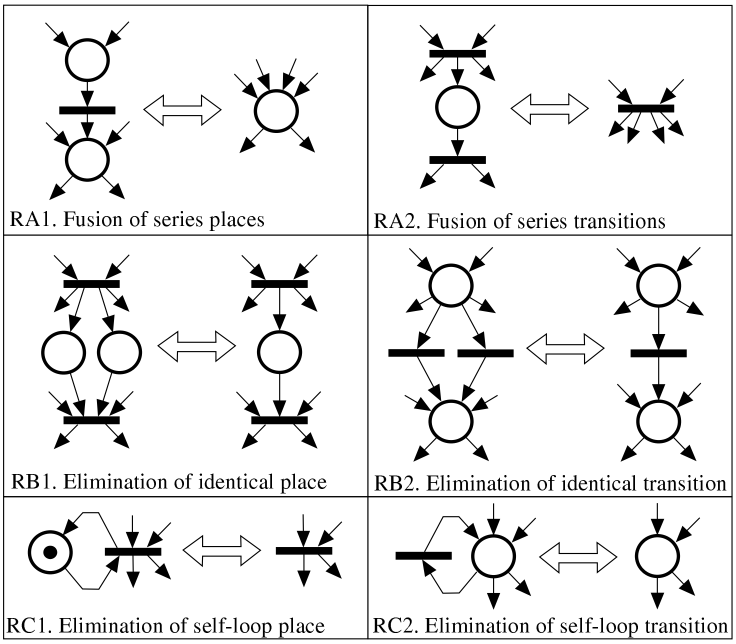
From an operational point of view, the approach is based on the definition of a kit or catalog of *reduction rules*, each one preserving a subset of properties (liveness, boundedness, reversibility, etc) to be analysed. A reduction rule characterises a type of subnet system (*locality principle*) to be substituted by another (simpler) subnet system.

The preconditions to be fulfilled have a *behavioural* and/or *structural* formulation. Behavioural preconditions can be more powerful for a given initial marking, but their verification is usually much more complex. So preconditions presented here are based on structural considerations and properties of the initial marking (i.e. the initial marking is considered as a parameter).

The design of a catalog of reduction rules is based on a tradeoff between completeness (i.e. transformation capabilities) and usefulness (i.e. applicability).

Given a catalog of reduction rules, analysis by reduction (the transformation procedure) is iterative by nature: Given the property (or properties) to be analyzed, the subset of rules that preserve it (them) is applied until the reduced system becomes irreducible. The irreducible system may be so simple that the property under study is trivially checked (see Figure 6.9.d). In other cases, the irreducible net is just "simpler" to analyse using another analysis technique (e.g. we can obtain a reduced state space on which it is possible to analyse the property that has been preserved in the reduction process). In other words, techniques to analyse net system models are complementary, not exclusive.

Reduction rules are transformation rules interesting for net analysis. When considered in the reverse sense they become expansion rules, interesting for net synthesis: stepwise refinements (or top-down) approach. Examples of this approach can be found in the context of synthesis of live and bounded Free Choice systems [17] or in the definition of subclasses of nets by the recursive application of classical expansion rules as the case of Macroplace/Macrotransition systems



RA1. Fusion of series places

RA2. Fusion of series transitions

RB1. Elimination of identical place

RB2. Elimination of identical transition

RC1. Elimination of self-loop place

RC2. Elimination of self-loop transition

Figure 6.7: A basic reduction kit.

[15]. Using this approach, with adequate expansion rules, the model will verify by construction the specification. This is interesting when compared with the more classical approach based on the iteration of the design and analysis phases until the specification is satisfied. The iterative process has two basic disadvantages:

1) the lack of general criteria for modifying (correcting) a model which does not meet the requirements.

2) the operational difficulty inherent to the validation phase.

Nevertheless, since no kit of reduction rules is complete (i.e. able to fully reduce any system), it is not possible to synthesize an arbitrary system by such stepwise refinements.

A very basic kit of reduction rules is presented. Additional details are given only for the rule of implicit places, which are redundancies in the net system model: if an implicit place is removed, then illusory synchronizations disappear and other reduction rules can be applied.

6.4.1 A basic kit of reduction rules

Figure 6.7 presents graphically the structural and marking conditions for a kit of very particular cases of reduction rules. It is not difficult to observe that they

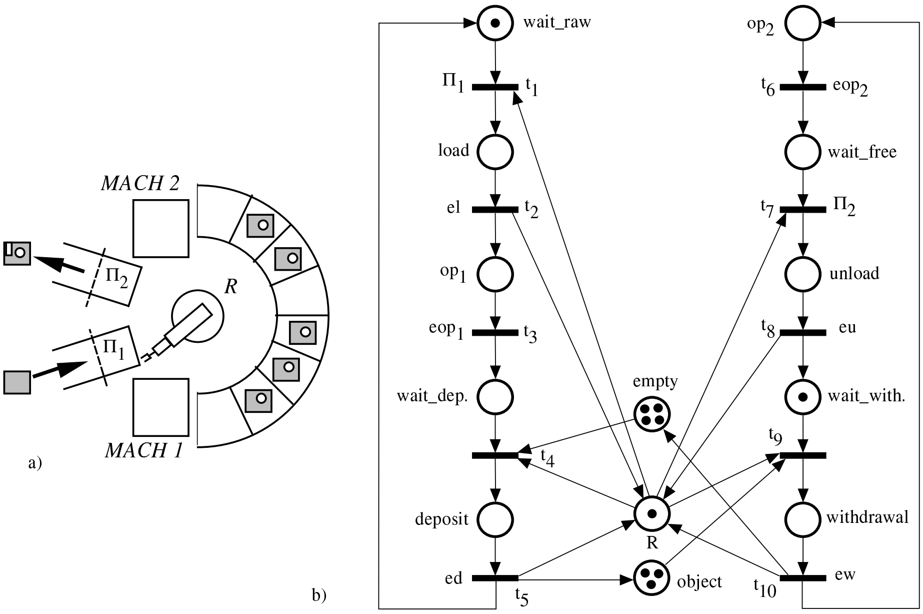


Figure 6.8: a) A production cell with two machines, one robot and a store. b) Net system specifying its behavior.

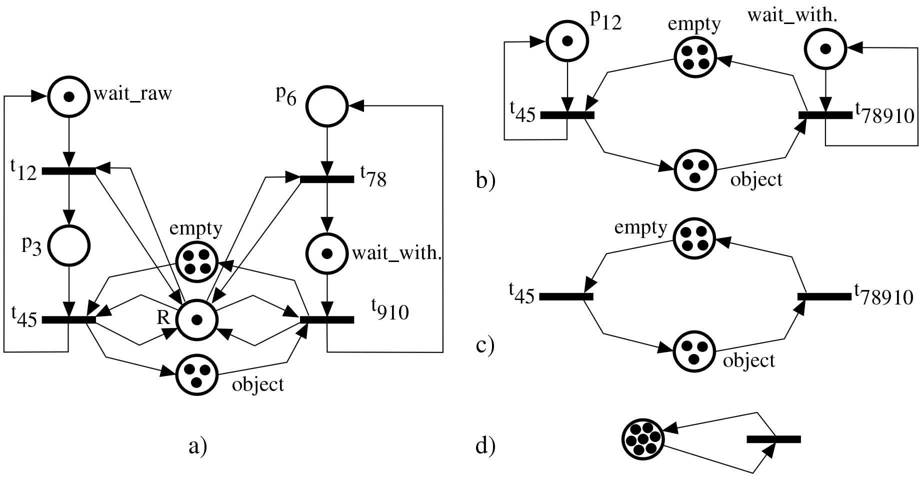


Figure 6.9: The reduction process shows (see (d)) that the net system in figure 6.8 is live, 7-bounded and reversible.

preserve properties such as liveness, the bound of places (thus boundedness), and the existence of home states (but they do not preserve reversibility because the rule *RA1*)

- *RA1* is a particular case of the *macroplace rule* [34]. If the output place of the transition has only this transition as input transition, then the entire kit preserves the reversibility property.
- *RA2* is a particular case of the *transition fusion rule* [3]
- *RB1* and *RC1* are particular cases of the *implicit place rule* [35, 37] (later considered in more detail). Observe that *RC1* can be trivially generalized creating several self-loops in which the place always appears. Liveness, the bound of places, and reversibility are preserved. Moreover if the place contains several tokens, liveness, boundedness (in general not the bound of the net system) and reversibility are preserved.
- *RB2* and *RC2* are particular cases of *identical* and *identity transition rules* [3], respectively.

An interesting remark is the analogy between rules at the same row in Figure 6.7: Basically rules *RX2* are obtained from rules *RX1* by changing the role of places and transitions (*duality*) and reversing the arrows (important only for rules *RA*).

Example 6.4 The local controller attached to the production cell depicted in Figure 6.8.a can be described by the given PN model. The places *wait_load*, *load*, *op*$_1$, *wait_dep.*, and *deposit* represent the possible states of *MACH 1*; The place R is marked when the robot is available; The places *empty* and *parts* contain as many tokens as empty slots or parts are available in the temporary buffer, etc. In this model actions are associated to places, e.g., *MACH 2* performs its operations while place *op*$_2$ is marked, and transitions represent atomic instantaneous changes of state. External inputs (from plant sensors) condition these possible changes of state, e.g., a load operation is initiated (transition t_1 is fired) when *MACH 1* is waiting for a raw part (*wait_load* marked), the robot is available (R marked), and a raw part is detected by the sensor Π_1 (Π_1 is true). As an example of constraint that is reflected in the model, a deposit operation cannot be initiated unless an empty slot is available. If the self-loops between *empty* and t_4 and between *object* and t_9 were deleted the system could reach a deadlock situation, e.g., *MACH 2* is "withdrawing" a part when there are none available but *MACH 1* cannot deposit any because the robot is busy.

Let us consider now the net system in Figure 6.8.b. The subnet defined by op$_1 - t_3 -$ wait_dep. verifies the precondition of rule *RA1* (but in the constrained form that preserves reversibility). Thus it can be reduced to a place, p_3 (Fig. 6.9.a). The same holds for op$_2$ $-$ t_6 $-$ wait_free that is reduced to p_6 (Fig. 6.9.a). The subnets t_1 $-$ load $-$ t_2, t_4 $-$ deposit $-$ t_5, t_7 $-$ unload $-$ t_8, and t_9 $-$ withdraw $-$ t_{10} can be reduced according to *RA2* (see t_{12}, t_{45}, t_{78} and t_{910} in Fig. 6.9.a). Place R in Fig. 6.9.a is implicit (one of the trivial generalizations mentioned for *RC1*). Thus it can be removed, and wait_draw $- t_{12} - p_3$ and $t_{910} - p_6 - t_{78}$ can be reduced to p_{12} and t_{78910}, respectively (see Figure 6.9.b). Places p_{12} and wait_with. are implicit (*RC1*) in Figure 6.9.b, thus the net system in Figure 6.9.c is obtained. Playing the token game, a place (e.g. object) can became empty in Figure 6.9.c and t_{45} − object $- t_{78910}$ can be reduced (*RA2*) to a single transition (Fig. 6.9.d). Therefore, the original net system is live, 7-bounded and reversible.

6.4.2 Implicit places

A place in a net system is a constraint to the firing of its output transitions. If the removal of a place does not change the behaviour of the original net system, it represents a redundancy in the system and it can be removed. A place whose removal preserves the behaviour of the system is called an *implicit place*. Two notions of behaviour equivalence are used to define implicit places. The first one considers that the two net systems have the same behaviour if they present the same fireable sequences. That is, this place can be removed without changing the *sequential observation* of the behaviour of the net system (i.e. the set of fireable sequences: interleaving semantics). Implicit places under this equivalence notion are called *sequential implicit places* (SIP). The second notion of equivalence imposes that the two net systems must have the same sequences of steps. In this case the implicit places are called *concurrent implicit places* (CIP) and their removal does not change the possibilities of simultaneous occurrences of transitions in the original net system. Implicit places model illusory synchronisations on their output transitions.

Definition 6.1 *Let $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ be a net system and $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m_0}' \rangle$ the net system resulting from removing place p from $\mathcal{S}$. The place p is a*

1. *Sequential Implicit Place (SIP) iff* $\mathrm{L}(\mathcal{N}, \mathbf{m_0}) = \mathrm{L}(\mathcal{N}', \mathbf{m_0}')$, *i.e., the removing of place p preserves all firing sequences of the original net.*
2. *Concurrent Implicit Place(CIP) iff* $\mathrm{LS}(\mathcal{N}, \mathbf{m_0}) = \mathrm{LS}(\mathcal{N}', \mathbf{m_0}')$, *i.e., the removing of place p preserves all sequences of steps of the original net.*

It is easy to see that if a place p is a CIP then it is also a SIP (since the preservation of the sequences of steps implies the preservation of the firing sequences). Nevertheless, the contrary is not true in general. Let us consider, for example, the net in Figure 6.2. The place p_6 is a SIP since its removal does not change the set of firing sequences (the reachability graphs of the original net system and the net system without place p_6 are the same), but the place p_6 is not a CIP because after its removal transitions b and c can occur simultaneously and in the original net system they are sequentialised (i.e. the steps are not preserved). A SIP with self-loops, in order to be a CIP may require more tokens in its initial marking than those making it a SIP (in our example p_6 to be CIP requires two tokens in the initial marking). In [9] it is proven that a self-loop free SIP is also a CIP.

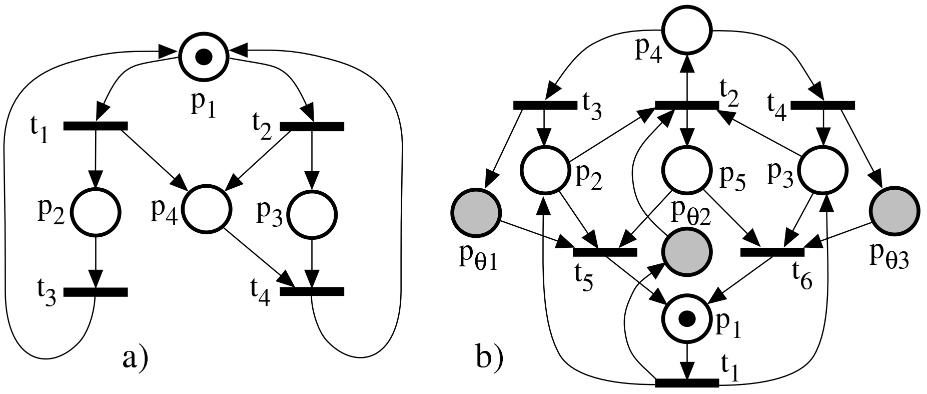


Figure 6.10: a) Place p_4 is firing implicit but not marking implicit. Removing p_4 the "false" synchronisation in t_4 disappears. b) The places in the set $\{p_{\theta 1}, p_{\theta 2}, p_{\theta 3}\}$ (or $\{p_2, p_3, p_5\}$) are CIPs.

Let p be an CIP of the net system $\mathcal{S}$ and $\mathcal{S}'$ the net system $\mathcal{S}$ without p. Let σ_s be a fireable sequence of steps in $\mathcal{S}$, such that $\mathbf{m_0} \xrightarrow{\sigma_s} \mathbf{m}$. The sequence σ_s is also fireable in the net system $\mathcal{S}'$, i.e., $\mathbf{m_0}' \xrightarrow{\sigma_s} \mathbf{m}'$. A trivial consequence of this is that the reached markings in $\mathcal{S}$ and $\mathcal{S}'$, firing the same sequence σ_s, are strongly related: $\forall q \in P \setminus \{p\}$, $\mathbf{m}[q] = \mathbf{m}'[q]$. Moreover, if $\mathbf{s}$ is a step enabled at $\mathbf{m}'$ the following holds: $\mathbf{m}' \geq \mathbf{Pre}' \cdot \mathbf{s} \Longrightarrow \mathbf{m}[p] \geq \sum_{t \in (p^\bullet \cap ||\mathbf{s}||)} \mathbf{s}[t] \cdot \mathbf{Pre}[p, t]$. If p is a SIP the previous property can be writen in the following way: $\forall t \in p^\bullet$, $\mathbf{m}' \geq \mathbf{Pre}'[P', t] \Longrightarrow \mathbf{m}[p] \geq \mathbf{Pre}[p, t]$.

The elimination of a CIP or a SIP preserves: deadlock-freeness, liveness and marking mutual exclusion properties; but it does not preserve: boundedness or reversibility. Moreover, the elimination of a CIP preserves the firing mutual exclusion property, but this is not true for SIPs.

Example 6.5 The net system in Fig. 6.10.a is unbounded (p_4 is the unique unbounded place) and non-reversible (also because of p_4). Place p_4 is a CIP. Removing p_4 the system becomes bounded and reversible! On the other hand, place p_6 in Figure 6.2 imposes firing mutual exclusion between b and c. Being p_6 a SIP, the reduction rule does not preserve firing mutual exclusion. According to the definition, fireable sequences are preserved.

Sometimes it is practical to impose an additional condition to the definition of implicit places, asking their marking to be redundant (computable) with respect to (from) the marking of the other places in the net (i.e. a marking redundancy property). Let us consider the CIP $p_{\theta 1}$ of the net system, $\mathcal{S}$, depicted in Figure 6.10.b. This place is CIP and its marking can be computed from the marking of places p_1, p_2 and p_5: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, $\mathbf{m}[p_{\theta 1}] = \mathbf{m}[p_1] + \mathbf{m}[p_2] + \mathbf{m}[p_5] - 1$. This class of places will be called *marking implicit places*. Nevertheless, the marking of some implicit places cannot be exclusively computed from the marking of the other places in the net. These places will be called *firing implicit* *places*. As an example consider the CIP p_4 in Figure 6.10.a: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$, such that $\mathbf{m_0} \xrightarrow{\sigma} \mathbf{m}$, $\mathbf{m}[p_4] = \mathbf{m}[p_3] + \boldsymbol{\sigma}[t_1]$). The classification of the implicit places into marking and firing implicit places can be applied to the two previously defined classes: CIP and SIP. Because of the additional condition, marking implicit places preserve the state space (i.e., the reachability graph of the net system with and without p are isomorphous), therefore they preserve boundedness and reversibility, too.

Implicit places presented until now are in a behavioural setting. In order to do the verification we must resort to algorithms based on the reachability graph with the inherent limitations and the high associated computational complexity. *Structurally implicit places* is a class of places that become implicit provided they are marked with enough tokens. The characterization of these places and a good bound of the minimum initial marking needed to be implicit can be done efficiently by means of Linear Programming techniques, avoiding the construction of the reachability graph.

Definition 6.2 *Let $\mathcal{N}$ be a net. A place p of $\mathcal{N}$ is a structurally implicit place iff there exists a subset $I_p \subseteq P \setminus \{p\}$ such that $\mathbf{C}[p,T] \geq \sum_{q \in I_p} y_q \cdot \mathbf{C}[q,T]$, where y_q is a nonnegative rational number (i.e. $\exists \mathbf{y} \geq 0$, $\mathbf{y}[p] = 0$ such that $\mathbf{y} \cdot \mathbf{C} \leq \mathbf{C}[p,T]$ and $I_p = ||\mathbf{y}||$.*

Obviously, the above structural condition can be checked in polynomial time. The next property gives the initial marking conditions to be satisfied by a structurally implicit place to become a SIP or a CIP. This condition is based on the solution of a Linear Programming Problem (the LPP in 6.1) that computes an upper bound of the minimal initial marking of a structurally implicit place to be SIP or CIP in the net system $\langle \mathcal{N}, \mathbf{m_0} \rangle$. Because, LPPs are of polynomial time complexity [29], the evaluation of this condition has this complexity.

Property 6.3 *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a net system. A structurally implicit place p of $\mathcal{N}$, with initial marking $\mathbf{m_0}[p]$, is a SIP (CIP) if $\mathbf{m_0}[p] \geq z$, where z is the optimal value of the LPP 6.1 with $\alpha = 1$ ($\alpha = \max\{\sum_{t \in p^\bullet} \mathrm{s}[t] | \mathrm{s} \in \mathrm{LS}(\mathcal{N}, \mathbf{m_0})\}$).*

$$\begin{array}{rll} z = & \mathit{min.} & \mathbf{y} \cdot \mathbf{m_0} + \alpha \cdot \mu \\ & \mathit{s.t.} & \mathbf{y} \cdot \mathbf{C} \leq \mathbf{C}[p,T] \\ & & \mathbf{y} \cdot \mathbf{Pre}[P,t] + \mu \geq \mathbf{Pre}[p,t] \quad \forall t \in p^\bullet \\ & & \mathbf{y} \geq 0, \mathbf{y}[p] = 0 \end{array} \tag{6.1}$$

If the optimal solution of the LPP 6.1, for a structurally implicit place p, verifies that $\mathbf{y} \cdot \mathbf{C} = \mathbf{C}[p,T]$, then p is a marking implicit place and the following holds: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$, $\mathbf{m}[p] = \mathbf{y} \cdot \mathbf{m} + \alpha \cdot \mu$.

Observe that a structurally implicit place, p, can become implicit for any initial marking of places $P \setminus \{p\}$, if we have the freedom to select an adequate initial marking for it. This property is not true for CIPs (or SIPs) that are not structurally implicit places. For example, the place p_{10} in Figure 2.4.a is a CIP but it is not a structurally implicit place. Moreover, the place p_{10} is not implicit if we change the initial marking of place p_4 from 0 to 1.

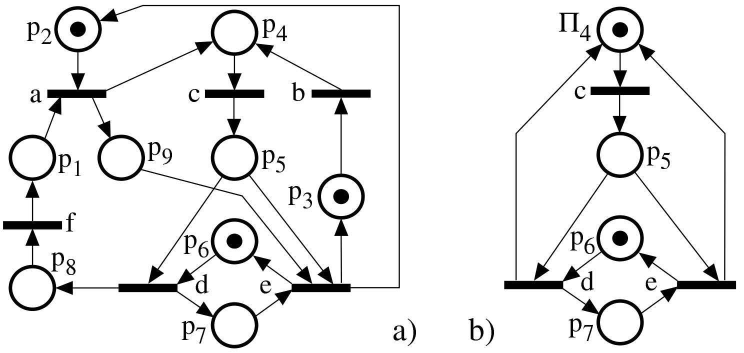


Figure 6.11: Places p_9 and p_2 (or p_2 and p_7) are implicits

Example 6.6 Solving the LPP in 6.1 for the place p_9 in Fig. 6.11.a with $\alpha = 1$ we obtain $z = 0$, for the optimal solution: $\mathbf{y} = [0,\ 0,\ 1,\ 1,\ 1,\ 0,\ 1,\ 0,\ 0]$ and $\mu = -1$. Moreover, $\mathbf{C}[p_9, T] = \mathbf{C}[p_3, T]+\ \mathbf{C}[p_4, T]+\ \mathbf{C}[p_5, T]+\ \mathbf{C}[p_7, T]$. Because $\mathbf{m_0}[p] \geq z = 0$, p_9 is a SIP (since p_9 is self-loop free place it is also a CIP) and can be removed. Being p_9 a marking implicit place we can write: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$, $\mathbf{m}[p_9] = \mathbf{m}[p_3] + \mathbf{m}[p_4] + \mathbf{m}[p_5] + \mathbf{m}[p_7] - 1$.

Once p_9 is removed, a similar computation for p_2 can be done and p_2 is also shown to be a CIP. Figure 6.11.b shows a reduced net system. It can be obtained reducing $p_3 - b - p_4$ into a place (say p_{34}) (*RA1*) and finally $p_8 - f - p_1 - a - p_{34}$ into Π_4. The rule *RA1* allows to fuse Π_4 and p_5. The new place is implicit, so it can be removed. Then a cycle with $p_6 - d - p_7 - e - p_6$ remains. Finally it can be reduced to a basic net, $p_6 - t_{de} - p_6$, with one token. Therefore the original net system is live, bounded. It is also reversible, but we cannot guarantee this because of the fusion of $p_3 - b - p_4$ into p_{34}.

6.5 Linear algebraic techniques

Analysis techniques based on linear algebra allow the verification of properties of a general net system. The key idea is simple, and it has been already commented previously: Let $\mathcal{S}$ be a net system with incidence matrix $\mathbf{C}$. If $\mathbf{m}$ is reachable from $\mathbf{m_0}$ by firing sequence σ, then $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$. Therefore the set of natural solutions, $(\mathbf{m}, \boldsymbol{\sigma})$, of this state equation defines a linearisation of the reachability set $\mathrm{RS}(\mathcal{S})$ denoted $\mathrm{LRS}^{SE}(\mathcal{S})$. This set can be used to analyse properties like marking and submarking reachability and coverability, firing concurrency, conflict situations, deadlock-freeness, mutual exclusion, k-boundedness, existence of frozen tokens (they never leaves a place), synchronic relations, etc. To do so, the properties are expressed as formulas of a first order logic having linear inequalities as atoms, where the reachability or fireability conditions are relaxed by satisfiability of the state equation. These formulas are verified checking existence of solutions to systems of linear equations that are automatically obtained from them [9]. For instance, if $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) : \mathbf{m}[p] = 0 \vee \mathbf{m}[p'] = 0$; then places p and p' are in mutual exclusion. This is verified checking absence of (natural) solutions to $\{\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \wedge \mathbf{m}[p] > 0 \wedge \mathbf{m}[p'] > 0\}$. *Integer Linear Programming Problems* [30] where the state equation is included in the set of constraints can be posed to express optimization problems, like the computation of marking bounds, synchronic measures, etc. [9, 37]. This approach is a generalization of the classical reasoning using linear invariants [25, 27], and it deeply bridges the domains of net theory and convex geometry resulting in a unified framework to understand and enhance structural techniques [9] (see subsection 6.5.1).

Unfortunately, it usually leads to only semidecision algorithms (i.e., only necessary or only sufficient conditions) because, in general, $\mathrm{RS}(\mathcal{S}) \subset \mathrm{LRS}^{SE}(\mathcal{S})$. The undesirable solutions are named *spurious*.

Example 6.7 (Existence of spurious solutions and their consequences in the analysis) Let us consider the net system depicted in Figure 2.3. The corresponding net state equation has the following marking spurious solutions: $\mathbf{m}_1 = 2 \cdot p_4$, $\mathbf{m}_2 = 2 \cdot p_2$, $\mathbf{m}_3 = 2 \cdot p_3$, $\mathbf{m}_4 = 2 \cdot p_5$, $\mathbf{m}_5 = p_2 + p_4$, $\mathbf{m}_6 = p_3 + p_4$. The first four solutions allow to conclude that p_2, p_3, p_4 and p_5 are 2-bounded, while they are really 1-bounded (check it). The solutions $\mathbf{m}_2$, $\mathbf{m}_3$ and $\mathbf{m}_4$ are total deadlocks. Then using the state equation we cannot conclude that the system in Fig. 2.3 is deadlock-free.

Spurious solutions can be removed using certain structural techniques, consequently improving the quality of the linear description of the system [11]. For example, it is clear that adding implicit places, a new system model with identical behaviour is obtained. For some net systems, if the implicit places are chosen carefully, the state equation of the new system may have no integer spurious solution preventing to conclude on the bound of a place or the deadlock freeness of the system.

Example 6.8 (Elimination of spurious solutions) The net system in Figure 6.10.b has been obtained adding the implicit places $p_{\theta 1}$, $p_{\theta 2}$ and $p_{\theta 3}$ to that in Figure 2.3. The above mentioned spurious solutions, $\mathbf{m}_i, i = 1 \ldots 6$; are not solutions of the new state equation. Moreover, we can conclude now that the new (and original) net system(s) were 1-bounded and deadlock-free!

Anyway the algorithms based on linear algebra do decide in many situations, and they are relatively efficient, specially if the integrality of variables is disregarded. (This further relaxation may spoil the quality, although in many cases it does not [13, 37].) Moreover, these techniques allow in an easy way an initial marking *parametric* analysis (e.g. changing the number of customers, size of resources, initial distribution of customers and/or resources, etc). The application of these techniques to the analysis of boundedness and deadlock-freeness properties is illustrated in subsections 6.5.2 and 6.5.3, repectively.

In temporal logic terms, the above outlined approach is well suited for *safety* properties ("some bad thing never happens"), but not so much for *liveness* properties ("some good thing will eventually happen"). For instance, the formula expressing reversibility would be $\forall \mathbf{m} \in \mathrm{LRS}^{SE}(\mathcal{S}) : \exists \boldsymbol{\sigma}' \gneqq 0 : \mathbf{m_0} = \mathbf{m} + \mathbf{C} \cdot \boldsymbol{\sigma}'$, but this is neither necessary nor sufficient for reversibility. The general approach to linearly verify these liveness properties is based on the verification of safety properties that are necessary for them to hold, together with some inductive reasoning [20]. For instance, deadlock-freeness is necessary for transition liveness, and the existence of some *decreasing potential function* proves reversibility [36] (see subsection 6.5.5).

Another important contribution of linear techniques to liveness analysis has been the derivation of *ad hoc* simple and efficient semidecision conditions. In subsection 6.5.4, we present one of these conditions based on a rank upper bound of the incidence matrix, which was originally conceived when computing the *visit ratios* in certain subclasses of net models [8].

The following subsections study linear invariants, marking bounds and boundedness, deadlock-freeness, structural liveness and liveness, and reversibility.

6.5.1 Linear invariants

A *p-flow* (*t-flow*) is a vector $\mathbf{y} : P \to \mathbb{Q}$ such that $\mathbf{y} \cdot \mathbf{C} = 0$ ($\mathbf{x} : T \to \mathbb{Q}$ such that $\mathbf{C} \cdot \mathbf{x} = 0$), where $\mathbf{C}$ is the incidence matrix of the net. The set of p-flows (t-flows) is a vector space, orthogonal to the space of rows (columns) of $\mathbf{C}$. Therefore, the flows can be generated from a *basis* of the space. Natural and non-negative flows are called *semiflows*: vectors $\mathbf{y} : P \to \mathbb{N}$ such that $\mathbf{y} \cdot \mathbf{C} = 0$ ($\mathbf{x} : T \to \mathbb{N}$ such that $\mathbf{C} \cdot \mathbf{x} = 0$). The following terminology is used with semiflows [27]: The *support of a p-semiflow (t-semiflow)*, $\mathbf{y}$ ($\mathbf{x}$): $||\mathbf{y}|| = \{p \in P | \mathbf{y}[p] > 0\}$ ($||\mathbf{x}|| = \{t \in T | \mathbf{x}[t] > 0\}$). A semiflow is *cannonical* iff the g.c.d. of its non-null elements is equal to one. A net is *conservative* (*consistent*) iff there exists a p-semiflow (t-semiflow) such that $||\mathbf{y}|| = P$ ($||\mathbf{x}|| = T$).

The set of cannonical semiflows of a given net can be infinite, since the weighted sum of any two semiflows is also a semiflow. Consider now the case of p-semiflows. A *generator set* of p-semiflows, $\Psi = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_q\}$, is made up of the least number of them which will generate any p-semiflow as follows:$\mathbf{y} = \sum_{\mathbf{y}_j \in \Psi} k_j \cdot \mathbf{y}_j$, $k_j \in \mathbb{Q}$ and $\mathbf{y}_j \in \Psi$. The p-semiflows of Ψ are said to be *minimal.* The following result characterizes the generator set of the semiflows of a net.

Proposition 6.4 *A semiflow is minimal iff it is cannonical and its support does not contain strictly the support of any other p-semiflow. Moreover, the set of minimal semiflows of a net is finite and unique.*

From the above result, the number of minimal semiflows is less than or equal to the number of incomparable vectors of dimension k ($k = |P|$ or $k = |T|$): Number of minimal semiflows $\leq \begin{pmatrix} k \\ \lceil k/2 \rceil \end{pmatrix}$. Where $\begin{pmatrix} \star \\ \star \end{pmatrix}$ denotes a combinatory number and $\lceil \star \rceil$ denotes rounding up to an integer. In practice this number is still too gross a bound for the number of minimal semiflows.

Algorithm 6.4 presents a simple version allowing the computation of the set of minimal p-semiflows, Ψ, from the incidence matrix of the net. Each row of matrix $\mathbf{\Psi}$ memorizes the coefficients of the positive linear combination of rows

Algorithm 6.4 (Computation of the minimal p-semiflows)

Input - The incidence matrix $\mathbf{C}$. A fixed but arbitrary order in P is supposed.
Output - The p-semiflows' matrix, $\mathbf{\Psi}$, where each row is a minimal p-semiflow.

```
1.   $\mathbf{A} = \mathbf{C}$; $\mathbf{\Psi} = \mathbf{I_n}$; { $\mathbf{I_n}$ is an identity matrix of dimension $n$ }
2.   for $i = 1$ to $m$ do { $m = |T|$ }
     2.1 Add to the matrix $[\mathbf{\Psi}|\mathbf{A}]$ all rows which are natural linear combinations
         of pairs of rows of $[\mathbf{\Psi}|\mathbf{A}]$ and which annul the i-th column of $\mathbf{A}$
     2.2 Eliminate from $[\mathbf{\Psi}|\mathbf{A}]$ the rows in which the i-th column of $\mathbf{A}$ is non-null.
3.   Transform the rows of $\mathbf{\Psi}$ into canonical p-semiflows and to remove all
     non-minimal p-semiflows from $\mathbf{\Psi}$ using the characterization of proposition 6.4.
```

of matrix $\mathbf{C}$ which generated the row of $\mathbf{A}$ with the same index. In step 3 of the algorithm, the rows of $\mathbf{A}$ are null and therefore each row $\mathbf{\Psi}[i]$ is a p-semiflow: $\mathbf{\Psi}[i] \cdot \mathbf{C} = 0$. The same algorithm can be used to compute the set of minimal t-semiflows if the input of the algorithm is the transpose of the incidence matrix.

The computation of minimal p-semiflows ($\mathbf{y}$) and minimal t-semiflows ($\mathbf{x}$) has been extensively studied [10]. Anyhow an *exponential number* of minimal semiflows may appear. Therefore the time complexity of this computation cannot be polynomial.

P- and T- semiflows are dual structural objects leading to linear invariant laws on the possible behaviours. These invariant laws arise from the structure of the net, and the initial marking plays the role of a parameter specifying a particular behaviour for the net. The two following classes of linear invariants can be obtained,

1) From p-semiflows: $\mathbf{y} \in \mathbb{N}^n$, $\mathbf{y} \cdot \mathbf{C} = 0 \Longrightarrow \forall \mathbf{m_0}$, $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$, $\mathbf{y} \cdot \mathbf{m} = \mathbf{y} \cdot \mathbf{m_0}$ (token conservation law)

2) From t-semiflows: $\mathbf{x} \in \mathbb{N}^m$, $\mathbf{C} \cdot \mathbf{x} = 0 \Longrightarrow \exists \mathbf{m_0}$, $\exists \sigma \in \mathrm{L}(\mathcal{N}, \mathbf{m_0})$ such that $\mathbf{m_0} \xrightarrow{\sigma} \mathbf{m_0}$ and $\boldsymbol{\sigma} = \mathbf{x}$ (cyclic behaviour law)

Classical reasoning to prove logical properties uses these *linear invariants* on the behaviour of a net system [25, 27]. The key idea is similar to that presented for the analysis of properties from the net state equation: Let $\mathcal{S}$ be a net system and $\mathbf{\Psi}$ a matrix where each row is a p-semiflow: $\mathbf{\Psi}[i] \cdot \mathbf{C} = 0$. If $\mathbf{m}$ is reachable from $\mathbf{m_0}$, then $\mathbf{\Psi} \cdot \mathbf{m} = \mathbf{\Psi} \cdot \mathbf{m_0}$. Therefore the set of natural solutions, $\mathbf{m}$, of this equation defines a linearisation of the reachability set $\mathrm{RS}(\mathcal{S})$ denoted $\mathrm{LRS}^{\Psi}(\mathcal{S})$. This set can be used to analyze properties in a similar way to the method based on the state equation. Moreover, as in the case of the net state equation, it usually leads to only semidecision algorithms because, in general, $\mathrm{RS}(\mathcal{S}) \subset \mathrm{LRS}^{SE}(\mathcal{S}) \subset \mathrm{LRS}^{\Psi}(\mathcal{S})$.

Example 6.9 (Analysis based on linear invariants) The marking linear invariants induced by the minimal p-semiflows of the net system in Figure 6.8

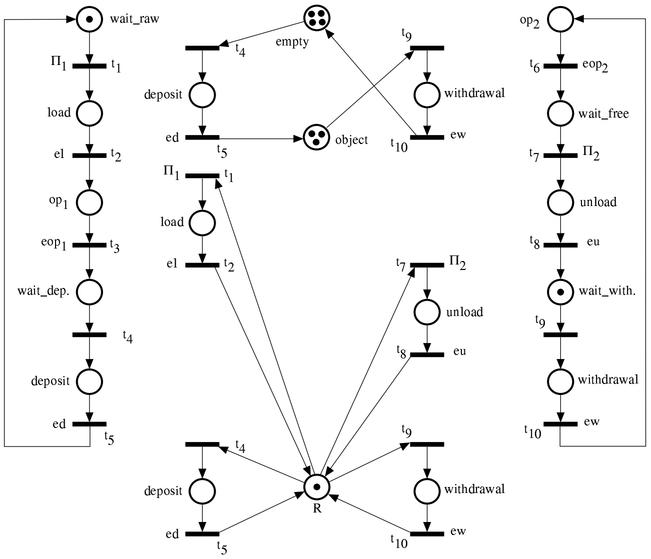


Figure 6.12: A decomposed view of the net system in Figure 6.8.

are the following:

$$\mathbf{m}[\text{wait\_raw}]+\mathbf{m}[\text{load}]+\mathbf{m}[\text{op}_1]+\mathbf{m}[\text{wait\_dep.}]+\mathbf{m}[\text{deposit}]=1 \qquad (6.2)$$

$$\mathbf{m}[\text{op}_2]+\mathbf{m}[\text{wait\_free}]+\mathbf{m}[\text{unload}]+\mathbf{m}[\text{wait\_with.}]+\mathbf{m}[\text{withdrawal}]=1 \qquad (6.3)$$

$$\mathbf{m}[\text{empty}]+\mathbf{m}[\text{deposit}]+\mathbf{m}[\text{object}]+\mathbf{m}[\text{withdrawal}]=7 \qquad (6.4)$$

$$\mathbf{m}[\text{R}]+\mathbf{m}[\text{load}]+\mathbf{m}[\text{unload}]+\mathbf{m}[\text{deposit}]+\mathbf{m}[\text{withdrawal}]=1 \qquad (6.5)$$

Because markings are non negative integers (i.e. $\forall p \in P$, $\mathbf{m}[p] \geq 0$), the following can be easily stated from the previous equalities:

1. Bounds: $\forall p_i \in P \backslash \{\text{empty}, \text{object}\}$, $\mathbf{m}[p_i] \leq 1$; $\mathbf{m}[\text{empty}] \leq 7$; and $\mathbf{m}[\text{object}] \leq 7$.
2. The places in each one of the following sets are in marking mutual exclusion:

   a) {wait_raw, load, op$_1$, wait_dep., deposit}

   b) {op$_2$, wait_free, unload, wait_with., withdrawal}

   c) {R, load, unload, deposit, withdrawal}

Finally, from a conceptual point of view, the consideration of semiflows provides *decomposed views* of the structure of the net model. In Figure 6.12 the decomposition induced by the minimal p-semiflows of the system in Figure 6.8

is graphically presented. The decomposed view of a net system is even useful to derive an *implementation.* For example, the net system in Figure 6.8 can be implemented using two sequential processes (for *Machine1* and *Machine2*) and three semaphores (*object*, *empty* and R), where R is a mutual exclusion semaphore.

Remark Other structural objects generalizing P- or T- semiflows have been defined [27] leading to other kind of linear invariants. A first type to consider are vectors $\mathbf{y} \in \mathbb{N}^n$ such that $\mathbf{y} \cdot \mathbf{C} \lneq 0$. A vector, $\mathbf{y}$, of this kind leads to the following marking law: $\forall \mathbf{m_0}$, $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$, $\mathbf{y} \cdot \mathbf{m} \leq \mathbf{y} \cdot \mathbf{m_0}$. A second type are vectors $\mathbf{x} \in \mathbb{N}^m$ such that $\mathbf{C} \cdot \mathbf{x} \gneq 0$. In this case, a vector $\mathbf{x}$ of this kind leads to: $\exists \mathbf{m_0}$, $\exists \sigma \in \mathrm{L}(\mathcal{N}, \mathbf{m_0})$ such that $\mathbf{m_0} \xrightarrow{\sigma} \mathbf{m} \geq \mathbf{m_0}$ and $\boldsymbol{\sigma} = \mathbf{x}$. This linear invariants (expressed as inequalities) can be used for analysis purposes in the same way that presented previously for linear invariants obtained from semiflows.

6.5.2 Bounds and boundedness

The study of the bound of a place p, $\mathbf{b}(p)$, through linear algebraic techniques, requires the linearisation of the reachability set in the definition of $\mathbf{b}(p)$ by means of the state equation of the net. In this subsection we assume that $\mathbf{m} \in \mathbb{R}^n$ and $\boldsymbol{\sigma} \in \mathbb{R}^m$. This linearisation of the definition of $\mathbf{b}(p)$ leads to a new quantity called the *structural bound* of p, $\mathbf{sb}(p)$:

$$\mathbf{sb}(p) = \sup\{\mathbf{m}(p) | \mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0, \boldsymbol{\sigma} \geq 0\} \tag{6.6}$$

Let $\mathbf{e_p}$ be the *characteristic vector* of p: $\mathbf{e_p}[q] :=$ if $q = p$ then 1 else 0. The structural bound of p, $\mathbf{sb}(p)$, can be obtained as the optimal solution of the following Linear Programming Problem (LPP):

$$\begin{array}{rll} \mathbf{sb}(p) = & \text{max.} & \mathbf{e_p} \cdot \mathbf{m} \\ & \text{s.t.} & \mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0 \\ & & \boldsymbol{\sigma} \geq 0 \end{array} \tag{6.7}$$

Therefore $\mathbf{sb}(p)$ can be computed in polynomial time. In sparse-matrix problems (matrix $\mathbf{C}$ is usually sparse), good implementations of the classical *simplex method* leads to quasi-linear time complexities.

Because $\mathrm{RS}(\mathcal{S}) \subset \mathrm{LRS}^{SE}(\mathcal{S})$, in general, we have that $\mathbf{sb}(p) \geq \mathbf{b}(p)$ (recall example 6.7). Therefore, if we are investigating the k-boundedness of a place (i.e. $\mathbf{m}[p] \leq k$), we have a sufficient condition in polynomial time: if $\mathbf{sb}(p) \leq k$ then $\mathbf{b}(p) \leq k$ (i.e. p is k-bounded).

In the sequel we argue on classical results from linear programming and convex geometry theories. We assume the reader is aware of these theories (see, for example, [28, 29]); otherwise all the needed arguments are compiled and adapted in [37]. The important point here is to convey the idea that other theories are helpful to understand in a deep and general framework many sparse results on net systems' behaviours. The dual linear programming problem of 6.7 is the following (see any text on linear programming to check it):

$$\begin{array}{rll} \mathbf{sb}(p)' = & \text{min.} & \mathbf{y} \cdot \mathbf{m_0} \\ & \text{s.t.} & \mathbf{y} \cdot \mathbf{C} \leq 0 \\ & & \mathbf{y} \geq \mathbf{e_p} \end{array} \tag{6.8}$$

The LPP in 6.7 has always a feasible solution ($\mathbf{m} = \mathbf{m_0}$, $\boldsymbol{\sigma} = 0$). Using duality and boundedness theorems from linear programming theory, both LPPs presented in 6.7 and 6.8 are bounded (thus p is structurally bounded) and $\mathbf{sb}(p) = \mathbf{sb}(p)'$ iff there exists a *feasible solution* for the LPP 6.8: $\mathbf{y} \geq \mathbf{e_p}$ such that $\mathbf{y} \cdot \mathbf{C} \leq 0$.

The reader can easily check that the LPP in 6.8 makes in polynomial time an "implicit search" for the structural bound of p on a set of structural objects including all the p-semiflows. In this sense, we can say that analysis methods based on the state equation are more general than those based on linear invariants. That is, the dual LPPs of those based on the state equation consider not only the p-semiflows but other structural objects as $\mathbf{y} \geq 0$ such that $\mathbf{y} \cdot \mathbf{C} \lneq 0$. On the other hand, we must say that the computational effort using the linear invariants is greater than using the state equation, since the computation of the minimal p-semiflows (in some cases, an exponential number!) must be done previously to the study of the property.

From the above discussion and using the *alternatives theorem* (an algebraic form of the *Minkowski-Farkas lemma*) the following properties can be proved:

Property 6.5 *The following three statements are equivalent:*

1. *p is structurally bounded, i.e. p is bounded for any $\mathbf{m_0}$.*
2. *There exists $\mathbf{y} \geq \mathbf{e_p}$ such that $\mathbf{y} \cdot \mathbf{C} \leq 0$. (place-based characterization)*
3. *For all $\mathbf{x} \geq 0$ such that $\mathbf{C} \cdot \mathbf{x} \geq 0$, $\mathbf{C}[p, T] \cdot \mathbf{x} = 0$. (transition-based characterization)*

Property 6.6 *The following three statements are equivalent:*

1. *$\mathcal{N}$ is structurally bounded, i.e. $\mathcal{N}$ is bounded for any $\mathbf{m_0}$.*
2. *There exists $\mathbf{y} \geq \mathbf{1}$ such that $\mathbf{y} \cdot \mathbf{C} \leq 0$. (place-based characterization)*
3. *For all $\mathbf{x} \geq 0$ such that $\mathbf{C} \cdot \mathbf{x} \geq 0$, $\mathbf{C} \cdot \mathbf{x} = 0$; i.e. $\nexists\, \mathbf{x} \geq 0$ s.t. $\mathbf{C} \cdot \mathbf{x} \gneq 0$. (transition-based characterization)*

6.5.3 Deadlock-freeness (and liveness)

Deadlock-freeness concerns the existence of some activity from any reachable state of the system. It is a necessary condition for liveness, although in general not sufficient. When no part of the system can evolve, it is said that the system has reached a state of total deadlock (or *deadlock* for short). In net system terms, a deadlock corresponds to a marking from which no transition is fireable. In order to study deadlock-freenes by means of linear algebraic techniques, the property must be expressed as a formula of a first order logic having linear inequalities as atoms, where the reachability or fireability conditions are relaxed by satisfiability of the state equation. The formula to express that a marking is a deadlock consists of a condition for every transition expressing that it is disabled at such marking. This condition consists of several inequalities, one per input place of the transition (expressing that the marking of such place is less than the corresponding weight) linked by the "$\vee$" connective (because lack of tokens in a single input place disables the transition). We give below a basic general sufficient condition for deadlock-freeness based on the absence of solutions satisfying simultaneously the net state equation and the formula expressing the total deadlock condition commented above.

Proposition 6.7 *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a net system. If there doesn't exist any solution $(\mathbf{m}, \boldsymbol{\sigma})$, for the system*

$$\begin{aligned} &\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \\ &\mathbf{m} \geq 0, \boldsymbol{\sigma} \geq 0 \\ &\textstyle\bigvee_{p \in {}^\bullet t} \mathbf{m}[p] < \mathbf{Pre}[p,t]; \forall t \in T \end{aligned} \tag{6.9}$$

then $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is deadlock-free.

Obviously, the deadlock conditions are non linear, because they are expressed using the "$\vee$" connective. Anyway we can express the above condition by means of a set of linear systems as follows. Let $\alpha : T \rightarrow P$ be a mapping that assigns to each transition one of its input places. If there doesn't exist α such that the system

$$\begin{aligned} &\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \\ &\mathbf{m} \geq 0, \boldsymbol{\sigma} \geq 0 \\ &\mathbf{m}[\alpha(t)] < \mathbf{Pre}[\alpha(t), t]; \forall t \in T \end{aligned} \tag{6.10}$$

has a solution, then $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is deadlock-free. The problem is that we have to check it for *every* mapping α of input places to transitions so we have to check $\prod_{t \in T} |{}^\bullet t|$ systems of linear inequalities. If every transition has exactly one input place (e.g. State Machines) then only one system needs to be checked, but in general the number might be large. Nevertheless it is possible to reduce the number of systems to be checked, preserving the set of *integer solutions*. For this purpose, in [39] four simplification rules of the deadlock condition are presented using information obtained from the net system, and a simple net transformation obtaining an equivalent one wrt. the deadlock-freeness property where the enabling conditions of transitions can be expressed linearly. As a result, deadlock-freeness of a wide variety of net systems can be proven by verifying absence of solutions to a *single system of linear inequalities*. Even more, in some subclasses it is known that there are no spurious solutions being deadlocks, so the method decides on deadlock-freeness [40]. The following example presents the deadlock-freenes analysis of the net system in figure 6.8 applying this technique.

Example 6.10 (Deadlock-freeness analysis and simplification rules) Let us consider the net system in Figure 6.8. The direct application of the method described in proposition 6.7 requires to check $\prod_{t\in T} |{}^{\bullet}t| = 36$ linear systems as that presented in 6.10. Nevertheless, below we show that we can reduce the deadlock-freeness analysis on this net to check a unique linear system applying the simplification rules presented in [39]. Solving the LPP 6.7 for the places of the net system we obtain the following: $\mathbf{sb}(\text{p}) = 1$, for all $p \in P\setminus\{\text{empty}, \text{object}\}$; and $\mathbf{sb}(\text{empty}) = \mathbf{sb}(\text{object}) = 7$ (the same can be obtained from the linear invariants in Eqs 6.2-6.5). The transitions t_1, t_4, t_7 and t_9 are those presenting complex conditions giving rise to the large number of linear systems. The simplification of these consitions is as follows:

a) The non-fireability condition of t_1 is $(\mathbf{m}[\text{wait\_raw}] = 0) \vee (\mathbf{m}[\text{R}] = 0)$. Taking into account that $\mathbf{sb}(\text{wait\_raw}) = \mathbf{sb}(\text{R}) = 1$, we can apply a particularization of rule 3 in [39] to replace the previous complex condition by a unique linear inequality: Let t be a transition such that each input place verifies that its structural bound is equal to the weight of its output arc joining it to t. The non fireability condition for transition t at a marking $\mathbf{m}$ is $\sum_{p\in{}^{\bullet}t} \mathbf{m}[p] \leq \sum_{p\in P} \mathbf{Pre}[p,t] - 1$. That is, the amount of tokens in the input places of t is less than the needed. Therefore, for the transition t_1 this linear condition is: $\mathbf{m}[\text{wait\_raw}] + \mathbf{m}[\text{R}] \leq 1$.

b) The non-fireability condition of t_7 is $(\mathbf{m}[\text{wait\_free}] = 0) \vee (\mathbf{m}[\text{R}] = 0)$. In a similar way to the case of transition t_1 we replace this condition by $\mathbf{m}[\text{wait\_free}] + \mathbf{m}[\text{R}] \leq 1$, since $\mathbf{sb}(\text{wait\_free}) = \mathbf{sb}(\text{R}) = 1$ and rule 3 in [39] can be applied.

c) The non-fireability condition of t_4 is $(\mathbf{m}[\text{wait\_dep.}] = 0) \vee (\mathbf{m}[\text{R}] = 0) \vee (\mathbf{m}[\text{empty}] = 0)$. Since $\mathbf{sb}(\text{wait\_dep.}) = \mathbf{sb}(\text{R}) = 1$ and $\mathbf{sb}(\text{empty}) = 7$ (i.e. only one input place of t_7 has a $\mathbf{sb}$ greater than the weight of the arc) rule 4 of [39] can be applied. Then, the previous complex condition is replaced by the following linear condition:

$$\begin{aligned}&\mathbf{sb}(\text{empty}) \cdot (\mathbf{m}[\text{wait\_dep.}] + \mathbf{m}[\text{R}]) + \mathbf{m}[\text{empty}] \leq \\ &\qquad \mathbf{sb}(\text{empty}) \cdot (\mathbf{Pre}[\text{wait\_dep.}, T] + \mathbf{Pre}[\text{R}, T]) + \mathbf{Pre}[\text{empty}, T] - 1\end{aligned}$$

i.e. $7(\mathbf{m}[\text{wait\_dep.}] + \mathbf{m}[\text{R}]) + \mathbf{m}[\text{empty}] \leq 14$.

d) The non-fireability condition of t_9 can be reduced to the following linear condition by similar reasons to the case of transition t_4: $7(\mathbf{m}[\text{wait\_with.}] + \mathbf{m}[\text{R}]) + \mathbf{m}[\text{object}] \leq 14$.

Applying the previously stated simplifications, the deadlock-freeness analysis for the net system in Figure 6.8 is reduced to verify that there doesn't exist any solution $(\mathbf{m}, \boldsymbol{\sigma})$, for the following single linear system (the reader can check that the system has no solutions).

$$\begin{array}{ll}
\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \boldsymbol{\sigma} & \\
\mathbf{m} \geq 0, \boldsymbol{\sigma} \geq 0 & \\
\mathbf{m}[\text{wait\_raw}] + \mathbf{m}[\text{R}] \leq 1; & \text{for } t_1 \\
\mathbf{m}[\text{load}] = 0; & \text{for } t_2 \\
\mathbf{m}[\text{op}_1] = 0; & \text{for } t_3 \\
7(\mathbf{m}[\text{wait\_dep.}] + \mathbf{m}[\text{R}]) + \mathbf{m}[\text{empty}] \leq 14; & \text{for } t_4 \\
\mathbf{m}[\text{deposit}] = 0; & \text{for } t_5 \\
\mathbf{m}[\text{op}_2] = 0; & \text{for } t_6 \\
\mathbf{m}[\text{wait\_free}] + \mathbf{m}[\text{R}] \leq 1; & \text{for } t_7 \\
\mathbf{m}[\text{unload}] = 0; & \text{for } t_8 \\
7(\mathbf{m}[\text{wait\_with.}] + \mathbf{m}[\text{R}]) + \mathbf{m}[\text{object}] \leq 14; & \text{for } t_9 \\
\mathbf{m}[\text{withdrawal}] = 0; & \text{for } t_{10}
\end{array} \tag{6.11}$$

Linear invariants may also be used to prove *deadlock-freeness*. Using the linear invariants in Eqs. (6.2-6.5), we shall prove that our net system in Figure 6.8 is deadlock-free.

If there exists a deadlock, no transition can be fired. Let us try to construct a marking in which no transition is fireable. When a unique input place of a transition exists, that place must be unmarked. So $\mathbf{m}[\text{load}] = \mathbf{m}[\text{op}_1] = \mathbf{m}[\text{deposit}] = \mathbf{m}[\text{op}_2] = \mathbf{m}[\text{unload}] = \mathbf{m}[\text{withdrawal}] = 0$, and the linear invariants in Eqs (6.2-6.5) reduce to:

$$\mathbf{m}[\text{wait\_raw}] + \mathbf{m}[\text{wait\_dep.}] = 1 \tag{6.12}$$

$$\mathbf{m}[\text{wait\_free}] + \mathbf{m}[\text{wait\_with.}] = 1 \tag{6.13}$$

$$\mathbf{m}[\text{empty}] + \mathbf{m}[\text{object}] = 7 \tag{6.14}$$

$$\mathbf{m}[\text{R}] = 1 \tag{6.15}$$

Since R should always be marked at the present stage, to prevent the firing of t_1 and t_7, places wait_raw and wait_free should be unmarked. The linear invariants are reduced once more, leading to:

$$\mathbf{m}[\text{wait\_dep.}] = 1 \tag{6.16}$$

$$\mathbf{m}[\text{wait\_with.}] = 1 \tag{6.17}$$

$$\mathbf{m}[\text{empty}] + \mathbf{m}[\text{object}] = 7 \tag{6.18}$$

$$\mathbf{m}[\text{R}] = 1 \tag{6.19}$$

Since $\mathbf{m}[\text{wait\_dep.}] = \mathbf{m}[\text{wait\_with.}] = 1$, to avoid the firing of t_4 and t_9, $\mathbf{m}[\text{empty}] + \mathbf{m}[\text{object}] = 0$ is needed. This contradicts Eq (6.18), so the net system is deadlock-free. A more compact, algorithmic presentation of the above deadlock-freeness proof is:

```
if $\mathbf{m}[\text{load}] + \mathbf{m}[\text{op}_1] + \mathbf{m}[\text{deposit}] + \mathbf{m}[\text{op}_2] + \mathbf{m}[\text{unload}] + \mathbf{m}[\text{withdrawal}] \geq 1$
  then one of $t_2$,$t_3$,$t_5$,$t_6$,$t_8$ or $t_{10}$ is fireable
  else if $\mathbf{m}[\text{wait\_raw}] + \mathbf{m}[\text{wait\_free}] \geq 1$
        then one of $t_1$or $t_7$ is fireable
        else one of $t_4$ or $t_9$ is fireable
```

As a final remark, we want to point out that liveness can be proved for the net system in Figure 6.8. Liveness implies deadlock-freeness, but the reverse is not true in general. Nevertheless, if the net is consistent and it has only one minimal t-semiflow, as it happens in the example, where the unique minimal t-semiflow is **1**; then any infinite behaviour must contain all transitions with relative firings given by such t-semiflow. Thus deadlock-freeness implies, in this case, liveness.

6.5.4 Structural liveness and liveness

A necessary condition for a transition t to be live in a system $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is its eventual infinite fireability, i.e. the existence of a firing repetitive sequence σ_R containing t: $\exists \sigma_R \in \mathrm{L}(\mathcal{N}, \mathbf{m_0})$ such that $\mathbf{m_0} \xrightarrow{\sigma_R} \mathbf{m} \geq \mathbf{m_0}$ and $\boldsymbol{\sigma}_R[t] > 0$.

Using the state equation as a linearisation of the reachability set, an *upper bound* of the number of times t can be fired in $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is given by the following LPP ($\mathbf{e_t}[u] :=$ if $u = t$ then 1 else 0):

$$\begin{aligned} \mathbf{sr}(t) = \ & \text{max.} \quad \mathbf{e_t} \cdot \boldsymbol{\sigma} \\ & \text{s.t.} \quad\ \ \mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq 0 \\ & \qquad\quad\ \ \boldsymbol{\sigma} \geq 0 \end{aligned} \tag{6.20}$$

The dual of (LPP 6.20) is:

$$\begin{aligned} \mathbf{sr}(t)' = \ & \text{min.} \quad \mathbf{y} \cdot \mathbf{m_0} \\ & \text{s.t.} \quad\ \ \mathbf{y} \cdot \mathbf{C} \leq -\mathbf{e_t} \\ & \qquad\quad\ \ \mathbf{y} \geq 0 \end{aligned} \tag{6.21}$$

We are interested on characterizing when $\mathbf{sr}(t)$ goes to infinity. The LPP 6.20 has $\mathbf{m} = \mathbf{m_0}$ and $\boldsymbol{\sigma} = 0$ as a feasible solution. Using first duality and unboundedness theorems from linear programming and later the alternatives theorem, the following properties can be stated:

Property 6.8 *The following three statements are equivalent:*

1. *t is structurally repetitive (i.e. there exists a "large enough" $\mathbf{m_0}$ such that t can be fired infinitely often).*
2. *There does not exist $\mathbf{y} \geq 0$ such that $\mathbf{y} \cdot \mathbf{C} \leq -\mathbf{e_t}$ (place-based perspective }*
3. *There exists $\mathbf{x} \geq \mathbf{e_t}$ such that $\mathbf{C} \cdot \mathbf{x} \geq 0$ { transition-based perspective }*

Property 6.9 *The following three statements are equivalent:*

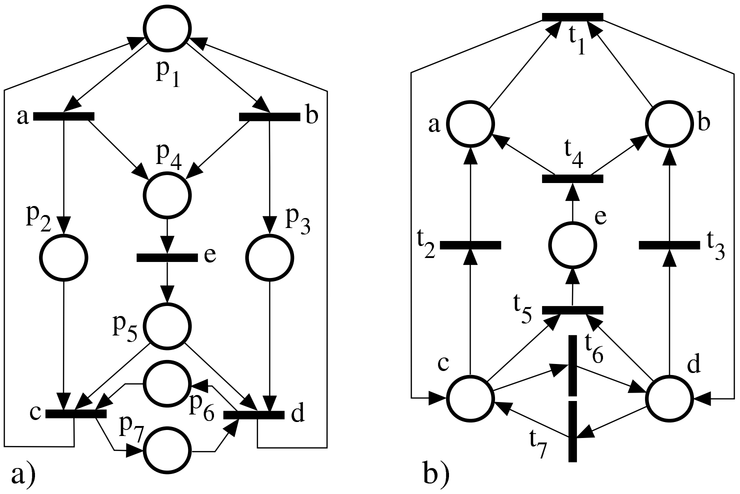


Figure 6.13: Two conservative and consistent, struturally non-live nets: (a) rank($\mathbf{C}$) = 4, |EQS| = 3, thus $\mathcal{N}$ is not structurally live; (b) rank($\mathbf{C}$) = 4, |EQS| = 4, |CCS| = 3, thus no answer.

1. *$\mathcal{N}$ is structurally repetitive (i.e. all transitions are structurally repetitive).*
2. *There does not exist* $\mathbf{y} \geq 0$ *such that* $\mathbf{y} \cdot \mathbf{C} \lneq 0$
3. *There exists* $\mathbf{x} \geq \mathbf{1}$ *such that* $\mathbf{C} \cdot \mathbf{x} \geq 0$

Aditionally, the following classical results can be stated [27, 7, 35]:

Property 6.10 *Let $\mathcal{N}$ be a net and* $\mathbf{C}$ *its incidence matrix.*

1. **if** *$\mathcal{N}$ is structurally live* **then** *$\mathcal{N}$ is structurally repetitive.*
2. **if** *$\mathcal{N}$ is structurally live and structurally bounded* **then** *$\mathcal{N}$ is conservative* ($\exists \mathbf{y} \geq \mathbf{1}$ *such that* $\mathbf{y} \cdot \mathbf{C} = 0$) *and consistent* ($\exists \mathbf{x} \geq \mathbf{1}$ *such that* $\mathbf{C} \cdot \mathbf{x} = 0$).
3. **if** *$\mathcal{N}$ is connected, consistent and conservative* **then** *it is strongly connected.*
4. **if** *$\mathcal{N}$ is live and bounded* **then** *$\mathcal{N}$ is strongly connected and consistent.*

Net structures in Figure 6.13 are consistent and conservative, but there does not exist a live marking for them. A more careful analysis allows to improve the above result with a *rank condition* on the incidence matrix of $\mathcal{N}$, $\mathbf{C}$. This and other results are summarized in the next property. Recall, from section I.2.2.4, that SEQS and SCCS denote the sets of Equal Conflict Sets and Coupled Conflict Sets, respectively.

Property 6.11 *Let $\mathcal{N}$ be a net and* $\mathbf{C}$ *its incidence matrix.*

1. **if** *$\mathcal{N}$ is live and bounded* **then** *$\mathcal{N}$ is strongly connected, consistent, and* $\text{rank}(\mathbf{C}) \leq |\text{SEQS}| - 1$.

2. **if** *$\mathcal{N}$ is conservative, consistent, and* $\text{rank}(\mathbf{C}) = |\text{SCCS}| - 1$ **then** *$\mathcal{N}$ is structurally live and structurally bounded.*

The condition in property 6.11.1 has been proven to be also sufficient for some subclasses of nets [12, 40]. Observe that, even for structurally bounded ordinary nets, we do not have a complete characterization of structural liveness. Since $|\text{SCCS}| \leq |\text{SEQS}|$, there is still a range of nets which satisfy neither the necessary nor the sufficient condition to be structurally live and structurally bounded! The added rank condition allows to state that the net in Figure 6.13.a is structurally non-live. Nevertheless, nothing can be said about structural liveness of the net in Figure 6.13.b.

Property 6.11 is purely structural (i.e., the initial marking is not considered at all). Nevertheless, it is clear that a too small initial marking (e.g. the empty marking) make non live any net structure. A less trivial lower bound for the initial marking based on marking linear invariants is based on fireability of every transition. If $t \in T$ is fireable at least once, for any p-semiflow $\mathbf{y}$, $\mathbf{y} \cdot \mathbf{m_0} \geq \mathbf{y} \cdot \mathbf{Pre}[P, t]$. Therefore:

Property 6.12 *If $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is a live system, then $\forall \mathbf{y} \geq 0$ such that $\mathbf{y} \cdot \mathbf{C} = 0$,* $\mathbf{y} \cdot \mathbf{m_0} \geq \max_{t \in T}(\mathbf{y} \cdot \mathbf{Pre}[P, t]) \geq 1$

Unfortunately no characterization of liveness exists in linear algebraic terms for general nets. The net system in Figure 6.1.b adding a token to p_5 is consistent, conservative, fulfills the rank condition and all p-semiflows are marked, but it is non live.

6.5.5 Reversibility (and liveness)

Let us use now a *Liapunov-stability-like* technique to prove that the net system in Figure 6.8 is reversible. It serves to illustrate the use of marking linear invariants and some inductive reasonings to analyze liveness properties.

As a preliminary consideration that makes easier the rest of the proof, the following simple property will be used: Let $\langle \mathcal{N}, \mathbf{m_1} \rangle$ be a reversible system and $\mathbf{m_0}$ reachable from $\mathbf{m_1}$ (i.e., $\exists \sigma \in \text{L}(\mathcal{N}, \mathbf{m_1})$ such that $\mathbf{m_1} \xrightarrow{\sigma} \mathbf{m_0}$). Then $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is reversible.

Assume $\mathbf{m_1}$ is like $\mathbf{m_0}$ (Figure 6.8), but making: $\mathbf{m_1}[\text{wait\_raw}] = \mathbf{m_1}[\text{empty}] = 0$, $\mathbf{m_1}[\text{wait\_dep.}] = 1$ and $\mathbf{m_1}[\text{object}] = 7$.

Let us prove first that $\langle \mathcal{N}, \mathbf{m_1} \rangle$ is reversible. Let $\mathbf{w}$ be a non-negative place weighting such that $\mathbf{w}[p_i] = 0$ iff p_i is marked in $\mathbf{m_1}$. Therefore, $\mathbf{w}[\text{wait\_dep.}] = \mathbf{w}[\text{R}] = \mathbf{w}[\text{object}] = \mathbf{w}[\text{wait\_with.}] = 0$ and $\mathbf{w}[p_j] > 0$ for all the other places. The function $\mathbf{v}(\mathbf{m}) = \mathbf{w} \cdot \mathbf{m}$ has the following properties: $\mathbf{v}(\mathbf{m}) \geq 0$ and $\mathbf{v}(\mathbf{m_1}) = 0$

For the system in Figure 6.8 a stronger property holds: $\mathbf{v}(\mathbf{m}) = 0 \Longleftrightarrow \mathbf{m} = \mathbf{m_1}$. This can be clearly seen because $\mathbf{w} \cdot \mathbf{m} = 0 \Longleftrightarrow \mathbf{m}[\text{wait\_raw}] = \mathbf{m}[\text{load}] =$

$\mathbf{m}[\mathrm{op}_1] = \mathbf{m}[\mathrm{deposit}] = \mathbf{m}[\mathrm{empty}] = \mathbf{m}[\mathrm{op}_2] = \mathbf{m}[\mathrm{wait\_free}] = \mathbf{m}[\mathrm{unload}] = \mathbf{m}[\mathrm{withdrawal}] = 0$. Even more, it is easy to check the following: $\mathbf{m}_1$ is the present marking $\Longleftrightarrow$ t_9 is the unique fireable transition.

If there exists (warning: in Liapunov-stability criteria the universal quantifier is used!) a finite firing sequence (i.e., a finite trajectory) per reachable marking $\mathbf{m}_i$ such that $\mathbf{m}_i \xrightarrow{\sigma_k} \mathbf{m}_{i+1}$ and $\mathbf{v}(\mathbf{m}_i) > \mathbf{v}(\mathbf{m}_{i+1})$, in a finite number of transition firings $\mathbf{v}(\mathbf{m}) = 0$ is reached. Because $\mathbf{v}(\mathbf{m}) = 0 \Longleftrightarrow \mathbf{m} = \mathbf{m}_1$, a proof that $\mathbf{m}_1$ is reachable from any marking has been obtained (i.e, $\langle \mathcal{N}, \mathbf{m}_1 \rangle$ is reversible).

Premultiplying the net state equation by $\mathbf{w}$ we obtain the following condition:**if** $\sigma_k = t_j$ **then** $[\mathbf{w} \cdot \mathbf{m}_{i+1} < \mathbf{w} \cdot \mathbf{m}_i] \Longleftrightarrow \mathbf{w} \cdot \mathbf{C}[P, t_j] < 0$

Now, removing in Figure 6.8 the places marked at $\mathbf{m}_1$ (i.e., wait_dep., R, object, wait_with.) and fireable transitions (i.e., t_9) an acyclic net is obtained, so there exists an $\mathbf{w}$ such that $\mathbf{w} \cdot \mathbf{C}[P, t_j] < 0$, $\forall j \neq 9$.

For example, taking as weights the levels in the acyclic graph we have:

$$\mathbf{w}[\mathrm{op}_1] = \mathbf{w}[\mathrm{unload}] = 1 \tag{6.22}$$

$$\mathbf{w}[\mathrm{load}] = \mathbf{w}[\mathrm{wait\_free}] = 2 \tag{6.23}$$

$$\mathbf{w}[\mathrm{wait\_raw}] = \mathbf{w}[\mathrm{op}_2] = 3 \tag{6.24}$$

$$\mathbf{w}[\mathrm{deposit}] = \mathbf{w}[\mathrm{withdrawal}] = 4 \tag{6.25}$$

$$\mathbf{w}[\mathrm{empty}] = 5 \tag{6.26}$$

and $\mathbf{w} \cdot \mathbf{C} = [-1, -1, -1, -1, -1, -1, -1, -1, +4, -1]$. In other words, the firing of any transition, except t_9, decreases $\mathbf{v}(\mathbf{m}) = \mathbf{w} \cdot \mathbf{m}$.

Using the algorithmic deadlock-freedom explanation in previous sections, the reversibility of $\langle \mathcal{N}, \mathbf{m}_1 \rangle$ is proven (observe that the p-invariants in Eqs (6.2-6.3-6.4-6.5) remain for $\mathbf{m}_1$):

```
if $\mathbf{m}[\mathrm{load}] + \mathbf{m}[\mathrm{op}_1] + \mathbf{m}[\mathrm{deposit}] + \mathbf{m}[\mathrm{op}_2] + \mathbf{m}[\mathrm{unload}] + \mathbf{m}[\mathrm{withdrawal}] \geq 1$
  then $\mathbf{v}(\mathbf{m})$ can decrease firing $t_2$,$t_3$,$t_5$,$t_6$,$t_8$ or $t_{10}$
  else if $\mathbf{m}[\mathrm{wait\_raw}] + \mathbf{m}[\mathrm{wait\_free}] \geq 1$
        then $\mathbf{v}(\mathbf{m})$ can decrease firing $t_1$or $t_7$
        else $\mathbf{v}(\mathbf{m})$ can decrease firing $t_4$ or $t_9$ is the unique fireable transition
            (iff $\mathbf{m}_1$ is the present marking)
```

Because $\mathbf{m}_0$ is reachable from $\mathbf{m}_1$ (e.g. firing $\sigma = (t_9 t_{10} t_6 t_7 t_8)^5 t_4 t_5$), $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ is a reversible system.

Once again liveness of the system in Figure 6.8 can be proved, because the complete sequence (i.e. containing all transitions) $\sigma = t_1 t_2 t_3 t_4 t_5 t_9 t_{10} t_6 t_7 t_8$ can be fired. Since the system is reversible, no transition loses the possibility of firing (i.e., all transitions are live).

6.6 Siphons and traps

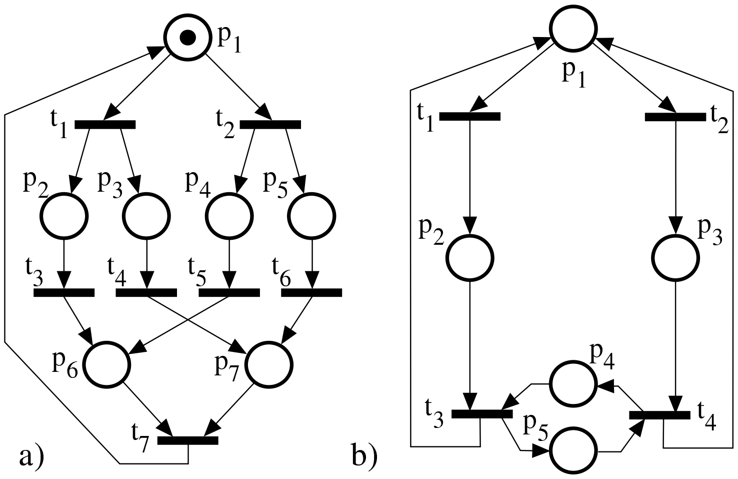


Figure 6.14: Two consistent and conservative free choice nets: (a) Structurally live rank(**C**) = 5, |EQS| = 5; (b) Structurally non-live rank(**C**) = 3, |EQS| = 2.

By means of graph theory based reasoning it is possible to characterize many properties of net subclasses. *Siphons* (also called *structural deadlocks*, or more simply - but ambiguously - *deadlocks*) and *traps* are easily recognizable subsets of places that generate very particular subnets.

Definition 6.13 *Let $\mathcal{N} = \langle P, T, F\rangle$ be an ordinary net.*

1. *A* siphon *is a subset of places, Σ, such that the set of its input transitions is contained in the set of its output transitions: $\Sigma \subseteq P$ is a siphon $\Longleftrightarrow$ ${}^\bullet\Sigma \subseteq \Sigma^\bullet$.*

2. *A* trap *is a subset of places, θ, such that the set of its output transitions is contained in the set of its input transitions: $\theta \subseteq P$ is a trap $\Longleftrightarrow$ $\theta^\bullet \subseteq {}^\bullet\theta$.*

$\Sigma = \{p_1, p_2, p_4, p_5, p_6\}$ is a siphon for the net in Figure 6.14.a: ${}^\bullet\Sigma = \{t_7, t_1, t_2, t_3, t_5\}$, while $\Sigma^\bullet = {}^\bullet\Sigma \cup \{t_6\}$. Σ contains a trap, $\theta = \Sigma \setminus \{p_5\}$. In fact θ is also a siphon (it is minimal: removing any number of places no siphon can be obtained).

Siphons and traps are reverse concepts: A subset of places of a net $\mathcal{N}$ is a siphon iff it is a trap on the reverse net, $\mathcal{N}^{-1}$ (i.e. that obtained reversing the arcs, its flow relation, F).

The following property "explains" why structural deadlocks or siphons (think on "soda siphons") and traps are the names of the above concepts.

Property 6.14 *Let $\langle \mathcal{N}, \mathbf{m_0}\rangle$ be an ordinary net system.*

1. *If $\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ is a deadlock state, then $\Sigma = \{p|\mathbf{m}[p] = 0\}$ is an unmarked (empty) siphon.*

2. *If a siphon is (or becomes) unmarked, it will remain unmarked for any possible net system evolution. Therefore all its input and output transitions are dead. So the system is not-live (but can be deadlock-free).*

3. *If a trap is (or becomes) marked, it will remain marked for any possible net system evolution (i.e. at least one token is "trapped").*

If a trap is not marked at $\mathbf{m_0}$, and the system is live, $\mathbf{m_0}$ will not be recoverable from those markings in which the trap is marked. Thus:

Corollary 6.15 *If a live net system is reversible, then* $\mathbf{m_0}$ *marks all traps.*

Remark For live and bounded free choice systems a stronger property holds: Marking all traps is a necessary and sufficient condition for reversibility [5]. The net system in Figure 6.14.a is reversible. Nevertheless, if $\mathbf{m_0} = [0, 1, 0, 0, 1, 0, 0]$, the new system is live and bounded but non reversible: The trap $\theta = \{p_1, p_3, p_4, p_6, p_7\}$ is not marked at $\mathbf{m_0}$.

A siphon which contains a marked trap will never become unmarked. So this more elaborate property can be of helpful for some liveness characterizations.

Definition 6.16 *Let $\mathcal{N}$ be an ordinary net. The system $\langle \mathcal{N}, \mathbf{m_0} \rangle$ has the* Marked-Siphon-Trap property, *MST-property, if each siphon contains a marked trap at* $\mathbf{m_0}$.

A siphon (trap) is *minimal* if it does not contain another siphon (trap). Thus, siphons in the above statement can be constrained to be minimal without any loss of generality.

The MST-property guarantees that all siphons will be marked. Thus no dead marking can be reached, according with property 6.14.1. Therefore:

Property 6.17 *If $\langle \mathcal{N}, \mathbf{m_0} \rangle$ has the MST-property, the system is deadlock-free.*

Figure 6.15 presents some limitations of the MST-property for liveness characterization.

Remark The MST-property is sufficient for liveness in simple net systems and necessary and sufficient for free-choice net systems. As a corollary, the *liveness monotonicity* result is true for the case of live free-choice systems: If $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is a live free-choice system, then for all $\mathbf{m_0}' \geq \mathbf{m_0}$, $\langle \mathcal{N}, \mathbf{m_0}' \rangle$ is also live. The previous result does not apply to Simple Net systems. The system in Figure 6.1.b is simple, $\Sigma = \{p_1, p_2, p_7\}$ is a siphon (${}^\bullet\Sigma = \{t_3, t_4, t_1\}$, $\Sigma^\bullet = {}^\bullet\Sigma \cup \{t_2\}$) that does not contain any trap. If we assume $\mathbf{m_0}[p_5] = 1$, t_2 can be fired and Σ becomes empty, leading to non-liveness.

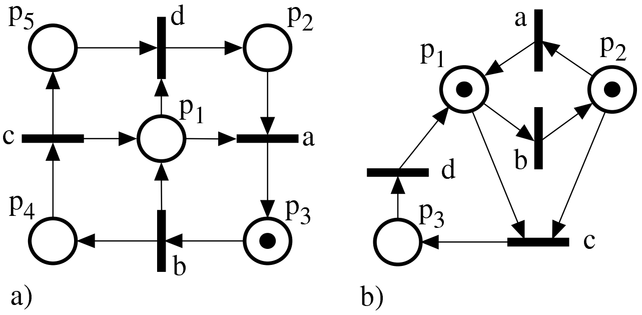


Figure 6.15: For the two nets, the MST-property does not hold, but: (a) The simple net is live and bounded; (b) The non-simple net is non-live (although deadlock-free) and bounded.

6.7 Analysis of net subclasses

In this section we quickly overview some of the analytical results for the subclasses defined in Chapter 2. We organise the material around properties instead of describing the results for each subclass, what would lead to abundant redundancies. (Of course, properties of large subclasses such as EQ systems, are inherited by their subclasses such as FC or DF systems.)

Our intention is to show how the restrictions imposed by subclasses' definitions, at the price of losing some modelling capabilities, facilitate the analysis. The designer must find a compromise between modelling power and availability of powerful analysis tools, while one of the theoretician's goals is obtaining better results for increasingly larger subclasses.

The general idea behind the structure theory of net subclasses is to investigate properties that every net system in the subclass enjoys, instead of analysing each particular system. These general properties are useful in two ways:

- The designer knows that her/his system (if it belongs to an appropriate subclass) behaves "well" (e.g., liveness monotonicity, existence of home states).
- General analysis methods become more applicable or more conclusive (e.g., model checking for FC, liveness analysis for all the subclasses considered).

The technical development of the presented results, and many other details that are out of the scope of this very succint presentation, can be found in [17], [14], [32], [38], [40].

6.7.1 Fairness and monopolies

In some systems, *impartiality* (or *global fairness*, that is, every transition appears infinitely often in infinite sequences) can be achieved *locally* (every solution of a — local — conflict that is effective infinitely often is taken infinitely often):

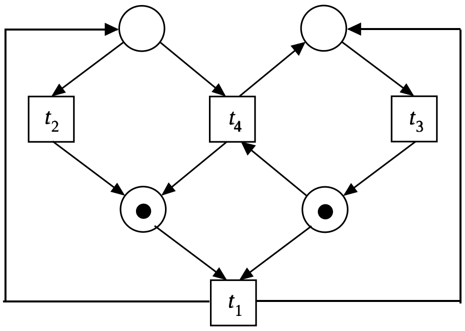


Figure 6.16: A net system where local fairness does not guarantee impartiality, and which can exhibit monopoly situations.

Theorem 6.18 *Let $\mathcal{S}$ be a bounded strongly connected EQ system or DSSP. A sequence $\sigma \in \mathrm{L}(\mathcal{S})$ is globally fair iff it is locally fair.*

This property is not true in general. Take for instance the net system in Figure 6.16. The sequence $\sigma = \{t_1 t_2 t_3\}^\omega$ is locally fair (actually, during the occurrence of σ no conflict is effective at all), but it is not globally fair since t_4 never occurs. Conversely, the sequence $\sigma = \{t_1 t_3 t_4 t_3 t_1 t_2 t_3\}^\omega$ is globally fair but not locally since whenever t_2 and t_4 are in conflict t_4 wins.

The equivalence of local and global fairness has two important consequences. The first one is equivalence of liveness and deadlock-freeness, what facilitates the analysis of liveness because it suffices to check the weaker property of deadlock-freeness:

Theorem 6.19 *Let $\mathcal{S}$ be a bounded strongly connected EQ system or DSSP. Then $\mathcal{S}$ is live iff it is deadlock-free.*

The second consequence is relevant for the eventual interpretation of the model. Assume, for instance, that the system in Figure 6.16 is interpreted so that transitions occur after a deterministic delay equal to their index. Then, the system behaves repeating the occurrence of $t_1 t_2 t_3$, never giving a chance to t_4, despite it was perfectly live in the autonomous model: the interpretation has destroyed liveness leading to a *monopoly* situation (the "resources" needed by t_4 are "monopolized" by t_2).

This can never happen to a bounded strongly connected EQ system or DSSP, assuming the interpretation allows progress (i.e., a transition that is continuously enabled eventually occurs): by imposing a fair conflict resolution policy, which can be done in a distributed fashion provided structurally conflicting transitions are allocated together, it is guaranteed that no action in the system becomes permanently disabled if the autonomous model was live.

6.7.2 Confluence and directedness

Persistent systems, which include structurally persistent ones (DF) enjoy a strong *confluence* property: whenever from a given marking we reach two different markings by firing two distinct sequences, then we can complete both such sequences, each with the firings left with respect to the other, reaching in any case the same marking [24]. Confluence is closely related to determinacy [22]: interpreting sequences as executions and transition occurrences as operations, when from a given point two different executions may occur, depending on operation times or other external matters, each operation in one execution will eventually occur in the other (assuming progress), possibly in a different order and with a different timing.

Moreover, confluence facilitates checking liveness (non-termination) of persistent systems: it suffices to find a repeatable sequence that contains every transition. This is because such a repeatable sequence allows to construct a sequence greater than any given sequence σ fireable from the initial marking, and this proves that σ can be continued to enable the repeatable sequence.

Stepping out from persistent systems, the presence of effective conflicts may destroy confluence. *Directedness* is a weaker property that states that a common successor of arbitrary reachable markings always exist, and which holds for some subclasses:

Theorem 6.20 *Let $\mathcal{S}$ be a live EQ system or DSSP. Let $\mathbf{m_a}, \mathbf{m_b} \in \mathrm{RS}(\mathcal{S})$. Then $\mathrm{RS}(\mathcal{N}, \mathbf{m_a}) \cap \mathrm{RS}(\mathcal{N}, \mathbf{m_b}) \neq \emptyset$.*

Informally, directedness means that the effect of a particular resolution of a conflict is not "irreversible": there is a point where the evolution joints with that which would have been if the decision had been other. The existence of *home states*, i.e., states that can be ultimately reached after whichever evolution, follows from directedness and boundedness:

Theorem 6.21 *Live and bounded EQ systems or DSSP have home states.*

The system in Figure 6.3.b is an example of a live and 1-bounded system without home states.

The existence of home state is an important property for many reasons:

- The system is known to have states to return to, which is often required in reactive systems. Chosing one such state as the initial one makes the system *reversible*, i.e., $\mathbf{m_0}$ can always be recovered.
- Model checking is largely simplified, since there is only one terminal strongly connected component in the reachability graph.
- Under a Markovian interpretation (e.g., as in *generalized stochastic Petri nets* [2]), *ergodicity* of the marking process is guaranteed; otherwise, simulation or computation of steady state performance indices could be meaningless.

6.7.3 Reachability and the state equation

As it was discussed in Section 6.5, reachable markings are solutions to the state equation but, in general, not conversely: some solutions of the state equation may be "spurious". This limits the use of the state equation as a convenient algebraic representation of the state space.

Fortunately stronger relations between reachable markings and solutions to the state equation are available for some subclasses:

Theorem 6.22 *Let $\mathcal{S}$ be a P/T system with reachability set* RS *and linearised reachability set wrt. the state equation* $\mathrm{LRS}^{\mathrm{SE}}$.

1. *If $\mathcal{S}$ is a live weighted T-system, or a live and consistent source private DSSP, then* $\mathrm{RS} = \mathrm{LRS}^{\mathrm{SE}}$ *(i.e. no spurious solutions). Moreover, if it is a live MG, then the integrality constraints can be disregarded (because in this case* $\mathbf{C}$ *is unimodular)*

2. *If $\mathcal{S}$ is a bounded, live, and reversible DF system, then* $\mathbf{m} \in \mathrm{RS}$ *iff* $\mathbf{m} \in \mathrm{LRS}^{\mathrm{SE}}$ *and the unique minimal T-semiflow of the net is fireable at* $\mathbf{m}$.

3. *If $\mathcal{S}$ is a live, bounded, and reversible FC system, then* $\mathbf{m} \in \mathrm{RS}(\mathcal{S})$ *iff* $\mathbf{m} \in \mathrm{LRS}^{\mathrm{SE}}(\mathcal{S})$ *(integrality constraints on* $\boldsymbol{\sigma}$ *can be disregarded) and every trap is marked at* $\mathbf{m}$.

4. *If $\mathcal{S}$ is a live EQ system or a live and consistent DSSP, and* $\mathbf{m_a}, \mathbf{m_b} \in \mathrm{LRS}^{\mathrm{SE}}$, *then* $\mathrm{RS}(\mathcal{N}, \mathbf{m_a}) \cap \mathrm{RS}(\mathcal{N}, \mathbf{m_b}) \neq \emptyset$.

We can take advantage of the above statements in a diversity of situations. For instance, the reachability characterisation for live MG allows to analyse some of their properties through linear programming. Even the last, and weakest, statement in the above theorem — a directedness result at the level of the linearised reachability graph — can be very helpful. In particular, it implies that there are no spurious deadlocks in live EQ systems, or live and consistent DSSP. Therefore, the deadlock-freeness analysis technique presented in Subsection 6.5.3 — which in these cases requires a single equation system — allows to decide liveness.

Figure I.2.3 shows an example of a live and 1-bounded system with spurious deadlocks.

6.7.4 Analysis of liveneness and boundedness

One of the properties that supports the claim that "good" behavior should be easier to achieve in some subclasses than in general systems is liveness *monotonicity* wrt. the initial marking. This means that liveness, provided that the net is "syntactically" correct as we shall precise later, is a matter of having enough tokens in the buffers (customers, resources, initial data, etc.), differently to what happens in general systems where the addition of tokens may well cause deadlocks due to poorly managed competition. For instance, in the net system of Figure 6.1.b adding a token (in p_5) to the initial marking destroys liveness.

Theorem 6.23 *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a live EQ system or DSSP. The EQ system or DSSP $\langle \mathcal{N}, \mathbf{m_0} + \Delta\mathbf{m_0} \rangle$, where $\Delta\mathbf{m_0} \geq \mathbf{0}$ is live too.*

Very often, a net system is required to be live and bounded. As we saw in Section 6.3 the verification of liveness can be very hard. In some cases we are able to decide using structural methods alone; in other cases we can characterise the nets that can be lively and boundedly marked, so the costful enumeration analysis needs to be used only when there is a chance of success.

Theorem 6.24 *Let $\mathcal{N}$ be an EQ or DSSP net. A marking $\mathbf{m_0}$ exists such that $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is a live and bounded EQ system or DSSP iff $\mathcal{N}$ is strongly connected, conservative (or consistent), and* $\mathrm{rank}(\mathbf{C}) = |\mathrm{SEQS}| - 1$. *Moreover, in EQ systems, liveness of the whole system is equivalent to liveness of each P-component (the P-subnets generated by the minimal P-semiflows).*

Particular cases of the above result are well-known in net theory. For instance, in the ordinary case, the P-components of a FC net are strongly connected SM, which are live iff they are marked, so the liveness criterion can be stated as "there are no unmarked P-semiflows". In the case of MG, which are always consistent and $\mathrm{rank}(\mathbf{C}) = |\mathrm{SEQS}| - 1 = |T| - 1$, the existence of a live and bounded marking is equivalent to strong connectedness. Since their P-components are their circuits, liveness can be checked removing the marked places and verifying that the remaining net is acyclic.

6.8 Logical properties in time constrained models

It must be noticed that the interpretations concerning timing or synchronization with external events restrict the behavior of the underlying autonomous model, so they should be taken into account for the analysis. On the one hand this may become extremely complicated in some cases because the notion of state must be enlarged, e.g., time PNs [4]. On the other hand performing analysis of the autonomous system only may not be conclusive except for some particular properties and subclasses of systems. For instance, the autonomous PN in Figure 6.17 (a) is not bounded unless the interpretation ensures that t' fires as often as t; the autonomous PN in (b) is not live (fire t twice) unless the interpretation precises that the conflict is resolved in alternating fashion; in (c), if t takes always more time than t' to fire then the system will not return to the initial marking and t'' will die, although the autonomous model is live and reversible. In general it can be said that *safety* properties of the autonomous system are preserved under any interpretation while *liveness* properties are neither necessary nor sufficient [35].

Fortunately, in some subclasses the the interpretation is not so disturbing. For instance, we shaw that in strongly connected and bounded EQ or DSSP liveness of the autonomous model is preserved under any "reasonable" interpretation (i.e., allowing progress and fairly resolving local conflicts). Similarly,

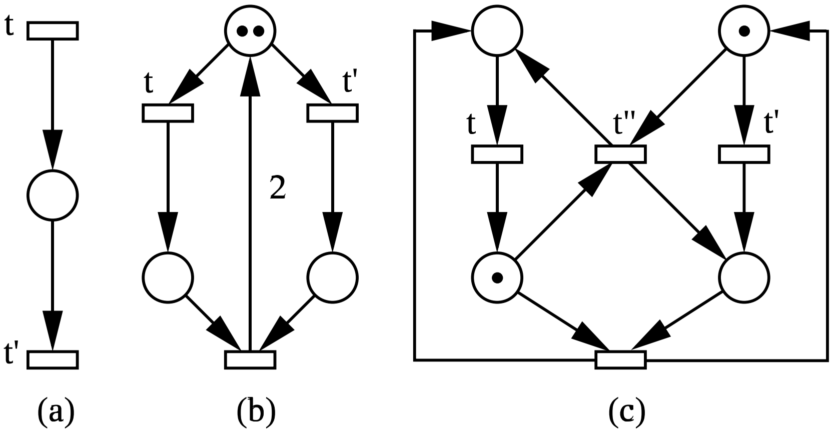


Figure 6.17: The interpretation affects qualitative properties.

some interpretations preserve the properties for every net system. For instance, under interpretations where the firing delay of transitions may range from zero to infinity (e.g., stochastic PN), the interpreted model has the same logical properties as the underlying autonomous model.

Bibliography

[1] *International Conference on Computer-Aided Verification.* The proceedings to date have been published as *Lecture Notes in Computer Science (LNCS)* **407** (1989), **531** (1990), **575** (1991), **663** (1992), **697** (1993), **818** (1994).

[2] M. Ajmone-Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets.* Wiley, 1995.

[3] G. Berthelot. Transformations and decompositions of nets. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Central Models and Their Properties. Advances in Petri Nets 1986. Part I*, volume 254 of *Lecture Notes in Computer Science*, pages 359–376. Springer Verlag, Berlin, 1987.

[4] B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time Petri nets. *IEEE Trans. on Software Engineering*, 17(3):259–273, 1991.

[5] E. Best, L. Cherkasova, J. Desel, and J. Esparza. Characterization of home states in free choice systems. Berichte 7/90, Hildesheimer Informatik, Hildesheim, Germany, July 1990.

[6] E. Best and K. Voss. Free choice systems have home states. *Acta Informatica*, 21:89–100, 1984.

[7] G.W. Brams. *Réseaux de Petri: théorie et pratique (2 vols.).* Masson, Paris, 1983.

[8] J. Campos, G. Chiola, and M. Silva. Properties and performance bounds for closed free choice synchronized monoclass queueing networks. *IEEE Transactions on Automatic Control*, 36(12):x–y, December 1991. [Special issue on *Multidimensional Queueing Networks*].

[9] J.M. Colom. *Análisis estructural de Redes de Petri, programación lineal y geometría convexa.* PhD thesis, Departamento de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Zaragoza, España, June 1989.

[10] J.M. Colom and M. Silva. Convex geometry and semiflows in P/T nets. A comparative study of algorithms for computation of minimal P-semiflows.

In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 79–112. Springer Verlag, Berlin, 1991.

[11] 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.

[12] J. Desel. A proof of the rank theorem for Extended Free Choice nets. In K. Jensen, editor, *Application and Theory of Petri Nets 1992*, volume 616 of *Lecture Notes in Computer Science*, pages 134–153. Springer Verlag, Berlin, 1992.

[13] J. Desel and J. Esparza. Reachability in cyclic Extended Free Choice nets. *Theoretical Computer Science*, 114:93–118, 1993.

[14] J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1995.

[15] A. Desrochers, H. Jungnitz, and M. Silva. An approximation method for the performance analysis of manufacturing systems based on GSPNs. In *Procs of the Third International Conference on Computer Integrated Manufacturing and Automation Technology (CIMAT'92)*, pages 46–55. IEEE Computer Society Press, 1992.

[16] J. Esparza and M. Nielsen. Decidability issues for Petri nets - a survey. *J. Inform. Process. Cybernet*, 30(3):143–160, 1994.

[17] J. Esparza and M. Silva. On the analysis and synthesis of free choice systems. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 243–286. Springer Verlag, Berlin, 1991.

[18] 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.

[19] M. Jantzen. Complexity of Place/Transition nets. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Central Models and Their Properties. Advances in Petri Nets 1986. Part I*, volume 254 of *Lecture Notes in Computer Science*, pages 413–434. Springer Verlag, Berlin, 1987.

[20] C. Johnen. Algorithmic verification of home spaces in P/T systems. In *Procs IMACS 1988. 12th World Congress on Scientific Computation*, pages 491–493, 1988.

[21] N.D. Jones, L.H. Landweber, and Y.E. Lien. Complexity of some problems in Petri nets. *Theoretical Computer Science*, 4:277–299, 1977.

[22] R. M. Karp and R. E. Miller. Properties of a model for parallel computations: Determinacy, termination, queueing. *SIAM Journal on Applied Mathematics*, 14(6):1390–1411, 1966.

[23] R.M. Karp and R.E. Miller. Parallel program schemata. *Journal of Computer Sciences*, 3:147–195, 1969.

[24] L. H. Landweber and E. L. Robertson. Properties of conflict-free and persistent Petri nets. *Journal of the ACM*, 25(3):352–364, 1978.

[25] K. Lautenbach. Linear algebraic techniques for Place/transition nets. In W. Brauer et al., editor, *Petri Nets: Central Models and their Properties. Advances in Petri Nets 1986*, volume 254 of *Lecture Notes in Computer Science*, pages 142–167. Springer Verlag, Berlin, 1987.

[26] K. Mehlhorn. *Graph algorithms and NP-completeness.* Springer-Verlag, Berlin, 1984.

[27] G. Memmi and G. Roucairol. Linear algebra in net theory. In W. Brauer, editor, *Net Theory and Applications*, volume 84 of *Lecture Notes in Computer Science*, pages 213–223. Springer Verlag, Berlin, 1980.

[28] K.G. Murty. *Linear Programming.* John Wiley and Sons, New York, 1983.

[29] G.L. Nemhauser, A.H. Rinnoy Kan, and M.J. Todd. *Optimization*, volume 1 of *Handbook in Operations Research and Management Science.* North Holland, Amsterdam, 1989.

[30] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley and Sons, 1988.

[31] J.L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice-Hall, New York, 1981.

[32] L. Recalde, E. Teruel, and M. Silva. Modeling and Analysis of Sequential Processes that Cooperate Through Buffers. *IEEE Trans. on Robotics and Automation*, 14(2):267–277, 1998.

[33] W. Reisig. *Petri Nets - An Introduction*, volume 4 of *Springer EATCS Monographs in Theoretical Computer Science.* Springer-Verlag, Berlin Heidelberg, 1985.

[34] M. Silva. Sur le concept de macroplace et son utilisation pour l'analyse des réseaux de Petri. *R.A.I.R.O. Automatique/Systems Analysis and Control*, 15(4):335–345, Avril 1981.

[35] M. Silva. *Las redes de Petri en la Automática y la Informática.* Editorial AC, Madrid, 1985.

[36] M. Silva. Logical controllers. In *Proceedings of IFAC Symposium on Low Cost Automation*, pages 157–166, 1989.

[37] M. Silva and J.M. Colom. On the computation of structural synchronic invariants in P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1988*, volume 340 of *Lecture Notes in Computer Science*, pages 386–417. Springer Verlag, Berlin, 1988.

[38] E. Teruel, J. M. Colom and M. Silva. Choice-free Petri Nets: A Model for Deterministic Concurrent Systems with Bulk Services and Arrivals. *IEEE Trans. on Systems, Man, and Cybernetics*, 27(1):73–83, 1997.

[39] E. Teruel, J.M. Colom, and M. Silva. Linear analysis of deadlock-freeness of Petri net models. In *Procs. of the European Control Conference, ECC'93*, pages 513–518, Groningen, The Netherlands, June 1993.

[40] E. Teruel and M. Silva. Structure theory of equal conflict systems. *Theoretical Computer Science*, 153(1-2):271–300, 1996.

[41] V. Valero, D. Frutos, and F. Cuartero. Simulation of timed Petri nets by ordinary Petri nets and applications to decidability of the timed reachability problem and other related problems. In *Procs of the Fourth IEEE International Workshop on Petri Nets and Performance Models (PNPM'91)*, pages 154–163. IEEE Computer Society Press, 1991.

[42] V. Valero, D. Frutos, and F. Cuartero. Decidability of the strict reachability problem for timed Petri nets with rational and real durations. In *Procs of the Fifth International Workshop on Petri Nets and Performance Models (PNPM'93)*, pages 138–147, Toulouse, France, October 1993. IEEE Computer Society Press.

[43] A. Valmari. Stunnorn sets for reduced state space generation. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 491–515. Springer Verlag, Berlin, 1991.

[44] A. Valmari. A stubborn attack on state explosion. *Formal Methods in System Design*, 1(4):297–322, 1992.

[45] W. Vogler. Live and bounded free choice nets have home states. *Petri Net Newsletter*, 32:18–21, April 1989.