Chapter 10
Introduction to Petri Nets

Maria Paola Cabasino, Alessandro Giua, and Carla Seatzu

10.1 Introduction

Petri nets (PNs) are a discrete event system model first introduced in the early 1960s by Carl Adam Petri in his Ph.D dissertation [14]. In this chapter we focus on the most common class of PNs, called *place/transition* (or *P/T*) *net*. It is a purely *logic model* that does not aim to represent the occurrence time of events, but only the order in which events occur.

Petri nets have been specifically designed to model systems with interacting components and as such are able to capture many characteristics of an event driven system, namely concurrency, asynchronous operations, deadlocks, conflicts, etc. Furthermore, the PN formalism may be used to describe several classes of logical models (e.g., P/T nets, Colored PNs, nets with inhibitor arcs), performance models (e.g., Timed PNs, Time PNs, Stochastic PNs), continuous and hybrid models (continuous PNs, hybrid PNs). Some of these models are considered in this book: timed PNs are studied in Chapters 16 and 17 while continuous PNs are the object of Chapters 18, 19 and 20.

The main features of PNs can be summarized in the following items.

- PNs are both a *graphical* and *mathematical* formalism. Being a graphical formalism, they are easy to interpret and provide a useful visual tool both in the design and analysis phase.
- They provide a *compact representation* of systems with a very large state space. Indeed they do not require to explicitly represent all states of a dynamical system but only an initial one: the rest of the state space can be determined from the rules that govern the system evolution. Thus a finite structure may be used to describe systems with an infinite number of states.

Maria Paola Cabasino · Alessandro Giua · Carla Seatzu
Department of Electrical and Electronic Engineering, University of Cagliari, Italy
e-mail: {cabasino,giua,seatzu}@diee.unica.it

C. Seatzu et al. (Eds.): Control of Discrete-Event Systems, LNCIS 433, pp. 191–211.
springerlink.com

- They permit a *modular representation*, i.e., if a system is composed by several subsystems that interact among them, it is usually possible to represent each subsystem with a simple subnet and then, through appropriate net operators, combine the subnets to obtain a model of the whole system.

Several PN analysis techniques have been presented in the literature. In this chapter we focus on *analysis by enumeration* that requires the construction of the *reachability graph* of the net representing the set of reachable markings and transition firings. If this set is not finite, a finite *coverability graph* may be constructed. Techniques based on *structural analysis*, on the contrary, permit the analysis of several properties based on the net structure, e.g., focusing on the state equation of the net or on the net graph; they are described in the next chapter.

The chapter is structured as follows. P/T nets and the rules that govern their evolution are introduced in Section 10.2. In Section 10.3 elementary PN structures are described and a physical modeling example is presented. In Section 10.4 the reachability and coverability graphs are presented. Behavioral properties of interest are also defined and characterized. Finally, Section 10.5 points out some further interesting reading.

10.2 Petri Nets and Net Systems

We will first define the algebraic and graphical structure of P/T nets. Adding a *marking* to such a structure, a *marked net* (or *net system*), i.e., a discrete event system, is obtained. The laws that govern its dynamical evolution are also studied.

10.2.1 Place/Transition Net Structure

A P/T net is a bipartite weighted directed graph. The two types of vertices are called *places* (represented by circles) and *transitions* (represented by bars or rectangles).

Definition 10.1. *A* place/transition *(or* P/T*)* net *is a structure* $N = (P, T, \boldsymbol{Pre}, \boldsymbol{Post})$ *where:*

- $P = \{p_1, p_2, \cdots p_m\}$ *is the set of* m places.
- $T = \{t_1, t_2, \cdots t_n\}$ *is the set of* n transitions.
- $\boldsymbol{Pre} : P \times T \longrightarrow \mathbb{N}$ *is the* pre-incidence function *that specifies the number of arcs directed from places to transitions (called "pre" arcs) and is represented as* $m \times n$ *matrix.*
- $\boldsymbol{Post} : P \times T \longrightarrow \mathbb{N}$ *is the* post-incidence function *that specifies the number of arcs directed from transitions to places (called "post" arcs) and is represented as* $m \times n$ *matrix.*

Example 10.1. In Fig. 10.1 it is represented the net $N = (P,T,\boldsymbol{Pre},\boldsymbol{Post})$ with set of places $P = \{p_1, p_2, p_3, p_4\}$ and set of transitions $T = \{t_1,t_2,t_3,t_4,t_5\}$. Here:

$$\boldsymbol{Pre} = \begin{array}{c} \left[\begin{array}{ccccc} 1&1&0&0&0\\ 0&0&1&1&0\\ 0&0&0&0&1\\ 0&0&0&0&1 \end{array}\right] \begin{array}{c} p_1\\ p_2\\ p_3\\ p_4 \end{array} \\ \begin{array}{ccccc} t_1&t_2&t_3&t_4&t_5 \end{array} \end{array} \qquad \boldsymbol{Post} = \begin{array}{c} \left[\begin{array}{ccccc} 1&0&0&0&1\\ 0&2&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0 \end{array}\right] \begin{array}{c} p_1\\ p_2\\ p_3\\ p_4 \end{array} \\ \begin{array}{ccccc} t_1&t_2&t_3&t_4&t_5 \end{array} \end{array}$$

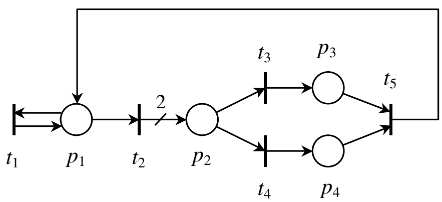

Fig. 10.1 A place/transition net

The element $\text{Post}[p_2,t_2] = 2$ denotes that there are two arcs from transition t_2 to place p_2. This is represented in the figure by means of a single barred arc with weight (or multiplicity) 2. ■

We denote by $\boldsymbol{Pre}[\cdot,t]$ the column of $\boldsymbol{Pre}$ relative to t, and by $\boldsymbol{Pre}[p,\cdot]$ the row of $\boldsymbol{Pre}$ relative to p. The same notation is used for matrix $\boldsymbol{Post}$.

The *incidence matrix* of a net defined as

$$\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}, \tag{10.1}$$

is represented by an $m \times n$ matrix of integers where a negative element is associated with a "pre" arc (from place to transition), while a positive element is associated with a "post" arc (from transition to place).

Note that the incidence matrix does not contain, in general, sufficient information to reconstruct the net structure. As an example, in the net in Fig. 10.1 it holds:

$$\boldsymbol{C} = \begin{bmatrix} 0&-1&0&0&1\\ 0&2&-1&-1&0\\ 0&0&1&0&-1\\ 0&0&0&1&-1 \end{bmatrix}.$$

In this net there exist both a "pre" and a "post" arc between place p_1 and transition t_1; we say that p_1 and t_1 form a *self-loop*, i.e., a directed cycle in the graph of the net only involving one place and one transition. In such a case, the algebraic sum of $\boldsymbol{Pre}$ and $\boldsymbol{Post}$ determines an element $C[p_1,t_1] = 0$, hiding the existence of arcs between these two vertices. A net without self-loops is called *pure*.

Finally, given a transition $t \in T$ we denote its set of *input* and *output* places as:

$$^\bullet t = \{p \in P \mid \text{Pre}[p,t] > 0\} \qquad \text{and} \qquad t^\bullet = \{p \in P \mid \text{Post}[p,t] > 0\},$$

while given a place $p \in P$ we denote its set of *input* and *output* transitions as:

$$^\bullet p = \{t \in T \mid \text{Post}[p,t] > 0\} \qquad \text{and} \qquad p^\bullet = \{t \in T \mid \text{Pre}[p,t] > 0\}.$$

As an example, in the net in Fig. 10.1 it holds $^\bullet t_2 = \{p_1\}$, $t_2^\bullet = \{p_2\}$, $^\bullet p_2 = \{t_2\}$ and $p_2^\bullet = \{t_3,t_4\}$.

10.2.2 Marking and Net System

The *marking* of a P/T net defines its state.

Definition 10.2. *A* marking *is a function* $\boldsymbol{m} : P \to \mathbb{N}$ *that assigns to each place a nonnegative integer number of* tokens.

As an example, in the net in Fig. 10.1, a possible marking $\boldsymbol{m}$ is $m[p_1] = 1$, $m[p_2] = m[p_3] = m[p_4] = 0$. Other possible markings are: $\boldsymbol{m}'$ with $m'[p_2] = 2$, $m'[p_1] = m'[p_3] = m'[p_4] = 0$; $\boldsymbol{m}''$ with $m''[p_2] = m''[p_4] = 1$, $m''[p_1] = m''[p_3] = 0$; etc. A marking is usually denoted as a column vector with as many entries as the number m of places. Thus $\boldsymbol{m} = [1\ 0\ 0\ 0]^T$, $\boldsymbol{m}' = [0\ 2\ 0\ 0]^T$, $\boldsymbol{m}'' = [0\ 1\ 0\ 1]^T$.

Graphically, tokens are represented as black bullets inside places. See as an example Fig. 10.4.

Definition 10.3. *A net N with an initial marking $\boldsymbol{m}_0$ is called* marked net *or* net system*, and is denoted $\langle N, \boldsymbol{m}_0 \rangle$.*

A marked net is a discrete event system with a dynamical behavior as discussed in the following section.

10.2.3 Enabling and Firing

Definition 10.4. *A transition t is* enabled *at a marking $\boldsymbol{m}$ if*

$$\boldsymbol{m} \geq \boldsymbol{Pre}[\cdot,t] \tag{10.2}$$

i.e., if each place $p \in P$ contains a number of tokens greater than or equal to $Pre[p,t]$. To denote that t is enabled at $\boldsymbol{m}$ we write $\boldsymbol{m}[t\rangle$. To denote that t' is not enabled at $\boldsymbol{m}$ we write $\neg\boldsymbol{m}[t'\rangle$.

In the net in Fig. 10.1 the set of enabled transitions at $\boldsymbol{m} = [1\ 0\ 0\ 0]^T$ is $\{t_1, t_2\}$; the set of enabled transitions at $\boldsymbol{m}' = [0\ 2\ 0\ 0]^T$ is $\{t_3, t_4\}$; $\{t_3, t_4\}$ is also the set of transitions enabled at $\boldsymbol{m}'' = [0\ 1\ 0\ 1]^T$ since t_5 is not enabled, even if p_4 is marked, because p_3 is not marked.

A transition with no input arcs, such as t in Fig. 10.2, is called a *source transition*. A source transition t is always enabled, since, being in such a case $\boldsymbol{Pre}[\cdot, t] = \mathbf{0}$, the condition in equation (10.2) is satisfied for all markings $\boldsymbol{m}$.

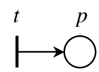

Fig. 10.2 A transition with no input arcs

Definition 10.5. *A transition t enabled at a marking $\boldsymbol{m}$ can* fire. *The* firing *of t removes $Pre[p,t]$ tokens from each place $p \in P$ and adds $Post[p,t]$ tokens in each place $p \in P$, yielding a new marking*

$$\boldsymbol{m}' = \boldsymbol{m} - \boldsymbol{Pre}[\cdot, t] + \boldsymbol{Post}[\cdot, t] = \boldsymbol{m} + \boldsymbol{C}[\cdot, t]. \tag{10.3}$$

To denote that the firing of t from $\boldsymbol{m}$ leads to $\boldsymbol{m}'$ we write $\boldsymbol{m}[t\rangle\boldsymbol{m}'$.

Note that the firing of a transition is an *atomic* operation since the removal of tokens from input places and their addition in output places occurs in an indivisible way. Consider the net in Fig. 10.1 at marking $\boldsymbol{m} = [1\ 0\ 0\ 0]^T$. If t_2 fires, $\boldsymbol{m}' = [0\ 2\ 0\ 0]^T$ is reached. Note that at marking $\boldsymbol{m} = [1\ 0\ 0\ 0]^T$, t_1 may also fire; the firing of such a transition does not modify the marking being $\boldsymbol{C}[\cdot, t_1] = \mathbf{0}$, thus it holds $\boldsymbol{m}[t_1\rangle\boldsymbol{m}$. If the marking of the net in Fig. 10.1 is equal to $\boldsymbol{m}' = [0\ 2\ 0\ 0]^T$, t_4 may fire leading to $\boldsymbol{m}'' = [0\ 1\ 0\ 1]^T$; note that t_3 is also enabled at $\boldsymbol{m}' = [0\ 2\ 0\ 0]^T$ and may fire instead of t_4.

Finally, in the marked net in Fig. 10.2 t is always enabled and can repeatedly fire, leading the initial marking $\boldsymbol{m}_0 = [0]$ to markings [1], [2] etc.

Definition 10.6. *A* firing sequence *at marking $\boldsymbol{m}_0$ is a string of transitions $\sigma = t_{j_1} t_{j_2} \cdots t_{j_r} \in T^*$, where T^* denotes the Kleene closure of T, such that*

$$\boldsymbol{m}_0[t_{j_1}\rangle\boldsymbol{m}_1[t_{j_2}\rangle\boldsymbol{m}_2 \cdots [t_{j_r}\rangle\boldsymbol{m}_r,$$

i.e., for all $k \in \{1, \ldots, r\}$ transition t_{j_k} is enabled at $\boldsymbol{m}_{k-1}$ and its firing leads to $\boldsymbol{m}_k = \boldsymbol{m}_{k-1} + \boldsymbol{C}[\cdot, t_{j_k}]$. To denote that the sequence σ is enabled at $\boldsymbol{m}$ we write $\boldsymbol{m}[\sigma\rangle$. To denote that the firing of σ at $\boldsymbol{m}$ leads to the marking $\boldsymbol{m}'$ we write $\boldsymbol{m}[\sigma\rangle\boldsymbol{m}'$.

The empty sequence *ε (i.e., the sequence of zero length) is enabled at all markings $\boldsymbol{m}$ and is such that $\boldsymbol{m}[\varepsilon\rangle\boldsymbol{m}$.*

In the net in Fig. 10.1 a possible sequence of transitions enabled at marking $\boldsymbol{m} = [1\ 0\ 0\ 0]^T$ is $\sigma = t_1 t_1 t_2 t_3$, whose firing leads to $\boldsymbol{m}''' = [0\ 1\ 1\ 0]^T$.

Let us now introduce the notion of conflict.

Definition 10.7. *Two transitions t and t' are in* structural conflict *if* $^\bullet t \cap {}^\bullet t' \neq \emptyset$, *i.e., if there exists a place p with a pre arc to both t and t'.*
Given a marking $\boldsymbol{m}$, we say that transitions t and t' are in behavioral conflict *(or in* conflict *for short) if* $\boldsymbol{m} \geq \boldsymbol{Pre}[\cdot,t]$ *and* $\boldsymbol{m} \geq \boldsymbol{Pre}[\cdot,t']$ *but* $\boldsymbol{m} \not\geq \boldsymbol{Pre}[\cdot,t] + \boldsymbol{Pre}[\cdot,t']$, *i.e., they are both enabled at $\boldsymbol{m}$, but $\boldsymbol{m}$ does not contain enough tokens to allow the firing of both transitions.*

In the net in Fig. 10.1 transitions t_3 and t_4 are in structural conflict. Such conflict is also behavioral at marking $\boldsymbol{m}'' = [0\ 1\ 0\ 1]^T$ since $p_2 \in {}^\bullet t_3 \cap {}^\bullet t_4$ only contains one token that can be used for the firing of only one of the two transitions. On the contrary, the conflict is not behavioral at marking $\boldsymbol{m} = [1\ 0\ 0\ 0]^T$ since the two transitions are not enabled. Analogously, the conflict is not behavioral at marking $\boldsymbol{m}' = [0\ 2\ 0\ 0]^T$, since p_2 contains enough tokens to allow the firing of both transitions.

To a marked net $\langle N, \boldsymbol{m}_0 \rangle$ it is possible to associate a well precise dynamics, given by the set of all sequences of transitions that can fire at the initial marking.

Definition 10.8. *The* language *of a marked net $\langle N, \boldsymbol{m}_0 \rangle$ is the set of firing sequences enabled at the initial marking, i.e., the set*

$$L(N, \boldsymbol{m}_0) = \{\sigma \in T^* \mid \boldsymbol{m}_0[\sigma\rangle\}.$$

Finally, it is also possible to define the state space of a marked net.

Definition 10.9. *A marking $\boldsymbol{m}$ is* reachable *in $\langle N, \boldsymbol{m}_0 \rangle$ if there exists a firing sequence σ such that $\boldsymbol{m}_0[\sigma\rangle\boldsymbol{m}$. The* reachability set *of a marked net $\langle N, \boldsymbol{m}_0 \rangle$ is the set of markings that can be reached from the initial marking, i.e., the set*

$$R(N, \boldsymbol{m}_0) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \exists \sigma \in L(N, \boldsymbol{m}_0) :\ \boldsymbol{m}_0[\sigma\rangle\boldsymbol{m}\}.$$

Note that in the previous definition the empty sequence, that contains no transition, is also considered. Indeed, since $\boldsymbol{m}_0[\varepsilon\rangle\boldsymbol{m}_0$, it holds $\boldsymbol{m}_0 \in R(N, \boldsymbol{m}_0)$.

As an example, let us consider the marked net in Fig. 10.3(a), where the initial marking assigns a number r of tokens to p_1. The reachability set is $R(N, \boldsymbol{m}_0) = \{[i\ j\ k]^T \in \mathbb{N}^3 \mid i + j + k = r\}$ and it is thus finite. On the contrary, the language $L(N, \boldsymbol{m}_0)$ of such a net system is infinite since sequences of arbitrary length can fire.

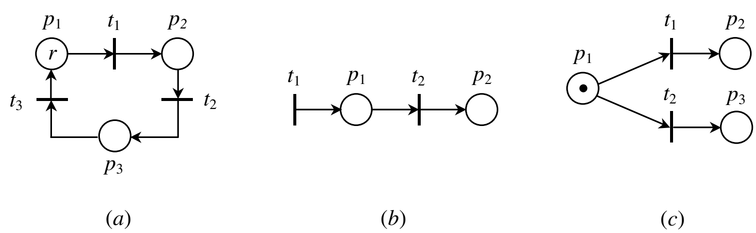

Fig. 10.3 Some examples of marked nets

In the net in Fig. 10.3(b), the reachability set is $R(N,\boldsymbol{m}_0) = \mathbb{N}^2$ and such a set is infinite. A generic marking $\boldsymbol{m}$ with $m[p_1] = i$ and $m[p_2] = j$ can be reached from $\boldsymbol{m}_0$ firing the sequence ${t_1}^{i+j}\ {t_2}^{j}$. Also the language is infinite. Finally, in the net in Fig. 10.3(c), it is $R(N,\boldsymbol{m}_0) = \{[1\ 0\ 0]^T, [0\ 1\ 0]^T, [0\ 0\ 1]^T\}$ and $L(N,\boldsymbol{m}_0) = \{\varepsilon, t_1, t_2\}$.

Summarizing, a double meaning is associated with the marking of a net: on one side the marking denotes the current state of the system; on the other side it specifies which activities can be executed, i.e., which transitions can fire. The transition firing determines the dynamical behavior of the marked net.

10.3 Modeling with Petri Nets

In this section we present some elementary P/T structures and the semantics associated with them.

In a discrete event system, the order in which events occur can be subject to constraints of different nature. In a PN model this corresponds to impose some constraints on the order in which transitions fire. In the following we present four main structures.

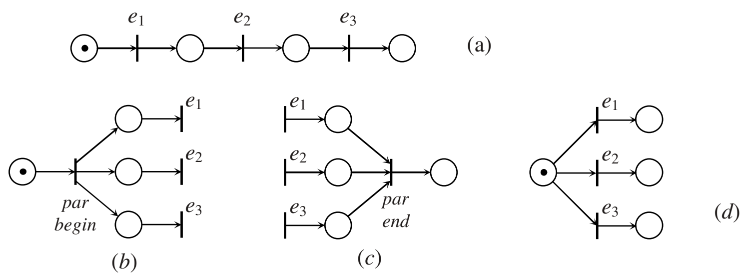

Fig. 10.4 Elementary structures of PNs: (a) sequentiality; (b) parallelism; (c) synchronization; (d) choice

Sequentiality. Events occur in a sequential order.

In Fig. 10.4(a) event e_2 can only occur after the occurrence of e_1; e_3 can occur only after the occurrence of e_2.

Parallelism (or **structural concurrency**). Events may occur with no fixed order.

In Fig. 10.4(b), after the firing of transition *par begin* (parallel begin) events e_1, e_2, and e_3 are simultaneously enabled. Parallelism implies that the three events are not in structural conflict and can occur in any order since the occurrence of any event does not modify the enabling condition of the others. Transition *par begin* creates a fork in the flow of events.

Synchronization. Several parallel events must have occurred before proceeding.

In Fig. 10.4(c), events e_1, e_2 and e_3 can occur in parallel but transition *par end* (parallel end) cannot fire until all of them have occurred. Transition *par end* creates a join in the flow of events.

Choice (or **structural conflict**). Only one event among many possible ones can occur.

In Fig. 10.4(d), only one event among e_1, e_2 and e_3 can occur, because the firing of any transition disables the others. Note that the choice is characterized by two or more transitions sharing an input place that determines a structural conflict.

To the above elementary structures it is often possible to associate a dual semantics, that takes into account the variation of markings, rather than the order in which transitions fire. In such a case, tokens represent available resources.

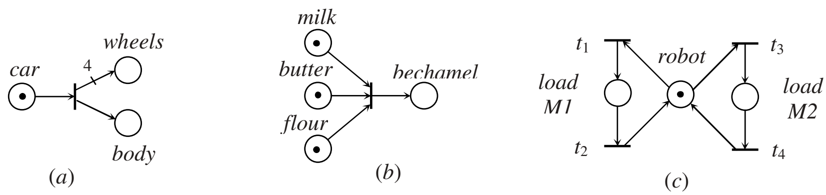

Fig. 10.5 Elementary structures of PNs: (a) disassembly; (b) assembly; (c) mutual exclusion

Disassembly. A composite element is separated into elementary parts.

In Fig. 10.5(a) the marked net represents the disassembly of a car, obtaining four wheels and a body. The transition is similar to the transition *par begin* previously introduced.

Assembly. Several parts are combined to produce a composite element.

In Fig. 10.5(b), the marked net describes the recipe to prepare bechamel sauce. The transition is similar to *par end* introduced above.

Mutual exclusion. A resource (or a set of resources) can be employed in several operations. However, while it has been acquired for a given operation, it is not available for other operations until it is released.

In Fig. 10.5(c), a single robot is available to load parts in two machines. When the place *robot* is marked the robot is available, while if either place *load M*1 or *load M*2 is marked the robot is acquired for the corresponding operation. From the situation in figure, if t_1 fires, the loading of the first machine starts and place *robot* gets empty: thus t_3, whose firing corresponds to the reservation of the robot for the loading of the second machine, is disabled until the firing of t_2 that moves the token again in place *robot*. Analogously, from the situation in figure the firing of t_3 disables t_1 until the firing of t_4. The structure is similar to "choice" in Fig. 10.4(d).

We conclude this section presenting an example taken from the manufacturing domain. Note that *manufacturing* is one of the application areas where PNs have been more extensively used since the early 1990s [4, 6].

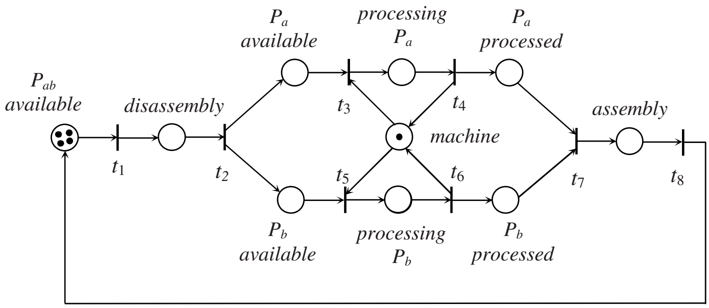

Fig. 10.6 Petri net model of a manufacturing cell

Figure 10.6 presents the Petri net model of a manufacturing cell where composite parts P_{ab} are processed. The cell consists of a single machine. A composite part is initially disassembled in two elementary parts P_a and P_b, that are, one at a time, processed by the machine. Finally, the two processed parts are assembled again and removed from the cell. The PN in Fig. 10.6 describes such a system. Places associated with resources are: P_{ab} *available*, P_a *available*, P_a *processed*, P_b *available*, P_b *processed*, *machine*. In figure, four tokens are initially assigned to place P_{ab} *available*: this denotes the presence of four parts P_{ab} that are available to be disassembled. The firing of t_1 represents the withdrawal of a part P_{ab} to be disassembled. The disassembly operation is modeled by transition t_2. After such an operation one part P_a and one part P_b are available to be processed. A single machine is available to process parts of both types. When transition t_3 fires, the machine starts processing a part of type P_a and no other part can be processed until t_4 fires, i.e., the machine is released. Analogously, transition t_5 represents the acquisition of the machine for the processing of a part of type P_b, while t_6 represents its release. Transitions t_7 models the assembly operation, that can only occur when a part of each type is available. At the end of the assembly operation, the processed part P_{ab} exits the cell and a new part to be processed enters the system. This is modeled by transition t_8. Note that this operation mode is typical of those processes where parts move on pallets, that are available in a finite number.

10.4 Analysis by Enumeration

In this section we present an important technique for the analysis of PNs based on the enumeration of the reachability set of the net and of the transition function between markings. If the reachability set is finite, an exhaustive enumeration is possible and the *reachability graph* of the net is constructed. If the reachability set is not finite, a finite *coverability graph* can still be constructed using the notion of

ω-*marking*; the coverability graph provides a larger approximation of the reachability set and of the net language.

10.4.1 Reachability Graph

The main steps for the construction of the reachability graph of a marked net $\langle N, \boldsymbol{m}_0 \rangle$ are summarized in the following algorithm.

Algorithm 10.2. (Reachability Graph).

1. *The initial node of the graph is the initial marking $\boldsymbol{m}_0$. This node is initially unlabeled.*
2. *Consider an unlabeled node $\boldsymbol{m}$ of the graph.*
   a *For each transition t enabled at $\boldsymbol{m}$, i.e., such that $\boldsymbol{m} \geq \boldsymbol{Pre}[\cdot, t]$:*
      i. *Compute the marking $\boldsymbol{m}' = \boldsymbol{m} + \boldsymbol{C}[\cdot, t]$ reached from $\boldsymbol{m}$ firing t.*
      ii. *If no node $\boldsymbol{m}'$ is on the graph, add a new node $\boldsymbol{m}'$ to the graph.*
      iii. *Add an arc t from $\boldsymbol{m}$ to node $\boldsymbol{m}'$.*
   b *Label node $\boldsymbol{m}$ "old".*
3. *If there exist nodes with no label, goto Step 2.*

In the case of nets with an infinite reachability set the algorithm does not terminate. However, a simple test to detect this case can be added at Step 2.a: if there exists a marking $\boldsymbol{m}''$, computed previously, such that the new marking $\boldsymbol{m}'$ is greater than and different from $\boldsymbol{m}''$, then stop the computation because the reachability graph is infinite.

An example of reachability graph is given in Fig. 10.7.

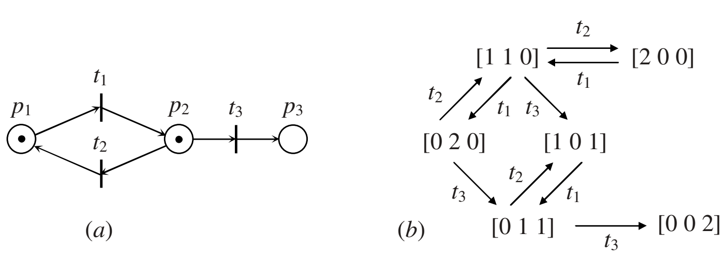

Fig. 10.7 (*a*) A bounded PN system and its reachability graph

The following proposition holds, whose proof immediately follows from the definition of reachability graph.

Proposition 10.1. *Consider a bounded marked net $\langle N, \boldsymbol{m}_0 \rangle$ and its reachability graph.*

(a) *Marking* $\boldsymbol{m}$ *belongs to the reachability set* $R(N,\boldsymbol{m}_0) \iff \boldsymbol{m}$ *is a node of the graph.*
(b) *Given* $\boldsymbol{m} \in R(N,\boldsymbol{m}_0)$, *sequence* $\sigma = t_{j_1}\ t_{j_2} \cdots$, *belongs to* $L(N,\boldsymbol{m})$ *and can be generated with the trajectory* $\boldsymbol{m}[t_{j_1}\rangle \boldsymbol{m}'[t_{j_2}\rangle \boldsymbol{m}'' \cdots \iff$ *there exists in the graph a directed path* $\gamma = \boldsymbol{m}\ t_{j_1}\ \boldsymbol{m}'\ t_{j_2}\ \boldsymbol{m}'' \cdots$.

As shown in [5], given a bounded PN, the problem of construction of the reachability graph is not primitive recursive. This implies that every method based on the reachability graph construction has an unpredictable complexity. This explains the importance of structural analysis which is the object of the following Chapter 11.

10.4.2 Coverability Graph

The procedure used for the construction of the reachability graph obviously does not terminate if the net is unbounded. Indeed in such a case, a situation like the following would surely occur. There exists a directed path that starts from $\boldsymbol{m}_0$ to $\tilde{\boldsymbol{m}}$, and from such a node there exists a directed path leading to $\boldsymbol{m}' \gneq \tilde{\boldsymbol{m}}$. To characterize the existence of sequences of transitions whose firing indefinitely increase the marking of some places, we assign a special symbol ω to all entries of $\boldsymbol{m}'$ that are strictly greater than the corresponding entries of $\tilde{\boldsymbol{m}}$.

Definition 10.10. *An* $\omega-$marking *of a net N with m places is a vector* $\boldsymbol{m}_\omega \in (\mathbb{N} \cup \{\omega\})^m$, *where one or more components may be equal to* ω.

Thus ω should be thought as "arbitrarily large" and we assume that $\forall n \in \mathbb{N}$ it holds $\omega > n$ and $\omega \pm n = \omega$.
Using the notion of $\omega-$marking, a finite approximation of the reachability graph, called *coverability graph*, can be constructed. The construction of the coverability graph first requires the construction of the *coverability tree*, a graph with no loops where duplicated nodes may exist. The following algorithm summarizes the main steps for the computation of the coverability tree of a marked net $\langle N,\boldsymbol{m}_0\rangle$ with incidence matrix $\boldsymbol{C}$.

Algorithm 10.3. (Coverability tree).

1 *The root node of the tree is the initial marking* $\boldsymbol{m}_0$. *This node is initially unlabeled.*
2 *Consider an unlabeled node* $\boldsymbol{m}$ *of the tree.*

  a *For each transition t enabled at* $\boldsymbol{m}$, *i.e., such that* $\boldsymbol{m} \geq \boldsymbol{Pre}[\cdot,t]$:
    i. *Compute the marking* $\boldsymbol{m}' = \boldsymbol{m} + \boldsymbol{C}[\cdot,t]$ *reached from* $\boldsymbol{m}$ *firing t.*
    ii. *For all markings* $\tilde{\boldsymbol{m}} \lneq \boldsymbol{m}'$ *on the path from the root node* $\boldsymbol{m}_0$ *to node* $\boldsymbol{m}$ *and for all* $p \in P$,
    if $\tilde{m}[p] < m'[p]$ *then let* $m'[p] = \omega$.
    iii. *Add a new node* $\boldsymbol{m}'$ *to the tree.*

iv. Add an arc t from **m** *to the new node* $\mathbf{m}'$.
v. If there already exists a node $\mathbf{m}'$ *in the tree, label the new node* $\mathbf{m}'$ *"duplicated".*
b Label node **m** *"old".*

3 If there exist nodes with no label, goto Step 2.

Karp and Miller [10] proved that Algorithm 10.3 always terminates in a finite number of steps even if the net has an infinite state space.

Consider as an example, the marked net in Fig. 10.8(*a*). The coverability tree is shown in Fig. 10.8(*b*) where labels "old" in the internal nodes have been omitted to make the figure more readable.

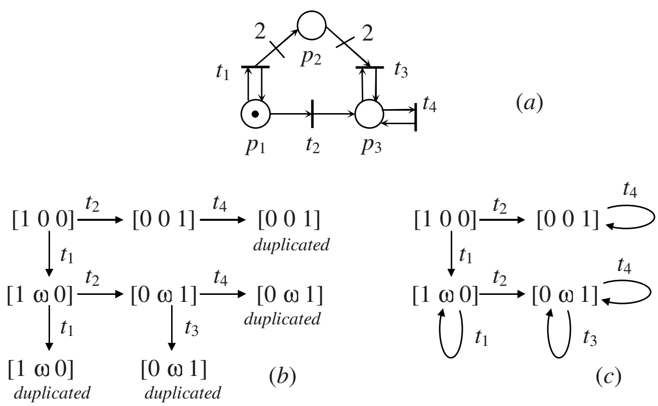

Fig. 10.8 (*a*) A PN; (*b*) coverability tree; (*c*) coverability graph

As summarized in Algorithm 10.4, "merging" duplicated nodes of the coverability tree, we obtain the *coverability graph*.

Algorithm 10.4. (Coverability graph).

1 If the tree contains no nodes with label "duplicated" goto Step 4.
2 Consider a node **m** *of the graph with label "duplicated".*
Such a node has no output arcs but an input arc t from node $\mathbf{m}'$.
Moreover, there surely exists in the graph another node **m** *with label "old".*

a Remove arc t from node $\mathbf{m}'$ *to node* **m** *"duplicated".*
b Add an arc t from node $\mathbf{m}'$ *to node* **m** *"old".*
c Remove node **m** *"duplicated".*

3 If there still exist nodes with label "duplicated" goto Step 2.
4 Remove labels from nodes.

The coverability graph of the marked net in Fig. 10.8(a) is shown in Fig. 10.8(c).

In the case of nets with an infinite reachability set, the coverability graph provides a finite description that approximates this infinite set.

Definition 10.11. *A marking $\boldsymbol{m} \in \mathbb{N}^m$ is said to be ω-*covered *by a vector $\boldsymbol{m}_\omega \in (\mathbb{N} \cup \{\omega\})^m$ if $m_\omega[p] = m[p]$ for all places p such that $m_\omega[p] \neq \omega$; this relation is denoted $\boldsymbol{m}_\omega \geq_\omega \boldsymbol{m}$.*

Thus a node $\boldsymbol{m}_\omega$ in the graph represents all markings that are ω-covered by it.

Due to the presence of ω-markings, the coverability graph does not always provide necessary and sufficient conditions to decide the reachability of a marking or the existence of a firing sequence. Such results are summarized in the following proposition, where a node that can contain ω components is denoted with the notation $\boldsymbol{m}_\omega$.

Proposition 10.2. *Consider a marked net $\langle N, \boldsymbol{m}_0 \rangle$ and its coverability graph.*

(a) Marking $\boldsymbol{m}$ is reachable $\Longrightarrow$ there exists a node $\boldsymbol{m}_\omega \geq_\omega \boldsymbol{m}$ in the graph.
(b) Given a marking $\boldsymbol{m} \in R(N, \boldsymbol{m}_0)$, sequence $\sigma = t_{j_1}\, t_{j_2} \cdots$, belongs to the language $L(N, \boldsymbol{m})$ and can be generated with a trajectory $\boldsymbol{m}[t_{j_1}\rangle \boldsymbol{m}'[t_{j_2}\rangle \boldsymbol{m}'' \cdots \Longrightarrow$ there exists in the graph a directed path $\gamma = \boldsymbol{m}_\omega\, t_{j_1}\, \boldsymbol{m}'_\omega\, t_{j_2}\, \boldsymbol{m}''_\omega \cdots$, with $\boldsymbol{m}_\omega \geq_\omega \boldsymbol{m}$, $\boldsymbol{m}'_\omega \geq_\omega \boldsymbol{m}'$ etc.

The main feature of the coverability graph is that of not providing a general algorithm, valid in all cases, to determine the reachability of a marking.

Example 10.5. Consider the marked net in Fig. 10.8 and its coverability graph. Based on Proposition 10.2(a) we conclude that marking $[0\ 0\ 1]^T$ is reachable, because it appears in the graph. On the contrary, based on Proposition 10.2(a), marking $[1\ 1\ 1]^T$ is not reachable since it is covered by no node in the graph. Finally, if we consider a marking $[0\ k\ 1]^T$ for a given value $k > 0$, it is not possible to draw a conclusion concerning its reachability, being it covered by node $[0\ \omega\ 1]^T$: as an example, $[0\ 2\ 1]^T$ is a reachable marking, while $[0\ 3\ 1]^T$ is not reachable.

Let us also observe that by Proposition 10.2(b) a coverability graph may contain directed paths associated with sequences that are not enabled. As an example, in the net in Fig. 10.8, $\sigma = t_1 t_2 t_3 t_3$ cannot fire at the initial marking: indeed in an admissible sequence, t_3 can fire at most as many times as t_1, due to the constraint imposed by place p_2 that is initially empty. However, starting from $\boldsymbol{m}_0$ there is in the graph a path whose arcs form sequence σ. ■

We conclude this section introducing the notion of *covering set*, that is a (not necessarily strict) superset of $R(N, \boldsymbol{m}_0)$.

Definition 10.12. *Given a marked net $\langle N, \boldsymbol{m}_0 \rangle$, let $V \subseteq (\{\mathbb{N} \cup \{\omega\})^m$ be the set of nodes of its coverability graph. The* covering set *of $\langle N, \boldsymbol{m}_0 \rangle$ is*

$$CS(N, \boldsymbol{m}_0) = \{\boldsymbol{m} \in \mathbb{N}^m \ \mid\ \exists \boldsymbol{m}_\omega \in V,\ \ m[p] = m_\omega[p]\ \ if\ \ m_\omega[p] \neq \omega\}\,.$$

By Proposition 10.2, we can state the following result.

Proposition 10.3. *Given a marked net $\langle N, \boldsymbol{m}_0 \rangle$, it holds $R(N, \boldsymbol{m}_0) \subseteq CS(N, \boldsymbol{m}_0)$.*

As an example, in the case of the marked net in Fig. 10.8, it holds $CS(N, \boldsymbol{m}_0) = \{[1\ 0\ 0]^T, [0\ 0\ 1]^T\} \cup \{[1\ k\ 0]^T,\ k \in \mathbb{N}\} \cup \{[0\ k\ 1]^T,\ k \in \mathbb{N}\} \subset R(N, \boldsymbol{m}_0)$. However, if N' is a new net obtained from the net in Fig. 10.8 changing the multiplicity of the arcs incident on place p_2 from 2 to 1, then $CS(N', \boldsymbol{m}_0) = R(N', \boldsymbol{m}_0)$.

Other approximations of the reachability set will be given in the following Chapter 11.

10.4.3 Behavioral Properties

In this section we define the main *behavioral properties* of a marked net, i.e., those properties that depend both on the net structure and on the initial marking.

10.4.3.1 Reachability

A fundamental problem in the PN net setting is the following, known as the *reachability problem.*

- Given a marked net $\langle N, \boldsymbol{m}_0 \rangle$ and a generic marking $\boldsymbol{m}$, is $\boldsymbol{m} \in R(N, \boldsymbol{m}_0)$?

As already discussed in the previous section, if the net has a finite state space, such a problem can be solved constructing the reachability graph. However, in the case of nets with an infinite state space, the coverability graph does not provide necessary and sufficient conditions to test if a given marking is reachable.

It is easy to show that the reachability problem is at least semi-decidable[1]. Indeed, if we consider a marked net $\langle N, \boldsymbol{m}_0 \rangle$ and a marking $\boldsymbol{m}$ whose reachability has to be verified, we can generate in an orderly fashion all sequences in $L(N, \boldsymbol{m}_0)$, starting first with those of length 1, then with those of length 2, etc., and compute the markings reached with each of these sequences. If $\boldsymbol{m}$ is reachable with a sequence of length k, at the kth step the algorithm terminates with a positive answer. However, if $\boldsymbol{m}$ is not reachable, this algorithm never halts.

In the 1980s it has been proved that the reachability problem is also *decidable*, even if the corresponding algorithm has a very high complexity [16].

[1] A problem whose solution may either be YES or NO is said to be:

- *decidable* if there exists an algorithm that, for each possible formulation of the problem, halts in a finite number of steps providing the correct solution;
- *semi-decidable* if there exists an algorithm that, for each possible formulation of the problem, halts in a finite number of steps providing the correct solution in one of the two cases (e.g., if the answer is YES), while it may not halt in the other case (e.g., if the answer is NO).

10.4.3.2 Boundedness

The *boundedness* property, associated with a place or with a net, implies that the number of tokens in the place or in the net, never exceeds a given amount. As an example, this property may imply that no overflow occurs in a buffer, or can be used to dimension the number of resources required by a process.

Definition 10.13. *A place p is $k-$*bounded *in $\langle N, \boldsymbol{m}_0 \rangle$ if for all reachable markings $\boldsymbol{m} \in R(N, \boldsymbol{m}_0)$ it holds $m[p] \leq k$. A place 1-bounded is* safe *(or* binary*). A marked net $\langle N, \boldsymbol{m}_0 \rangle$ is $k-$*bounded *if all places are $k-$bounded. A marked net that is $1-$bounded is called* safe *(or* binary*).*

When it is not important to specify the value of k, the place (net) is simply called *bounded*.

Proposition 10.4. [12] *A marked net $\langle N, \boldsymbol{m}_0 \rangle$ is bounded if and only if it has a finite reachability set.*

Proposition 10.5. *Consider a marked net $\langle N, \boldsymbol{m}_0 \rangle$ and its coverability graph.*

- *A place p is $k-$bounded $\Longleftrightarrow$ for each node $\boldsymbol{m}_\omega$ of the graph it holds $\boldsymbol{m}_\omega[p] \leq k \neq \omega$.*
- *The marked net is bounded $\Longleftrightarrow$ no node of the graph contains the symbol ω.*

The net in Fig. 10.8 is unbounded. Places p_1 and p_3 are safe, while place p_2 is unbounded. The net in Fig. 10.9 is safe.

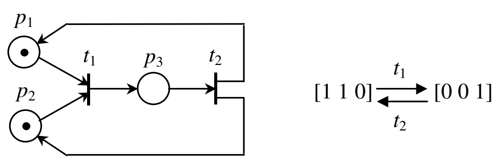

Fig. 10.9 A safe Petri net and its reachability graph

10.4.3.3 Conservativeness

A property strictly related to boundedness is *conservativeness* implying that the weighted sum of tokens in a net remains constant. Such a property ensures that resources are preserved.

Definition 10.14. *A marked net $\langle N, \boldsymbol{m}_0 \rangle$ is* strictly conservative *if for all reachable markings $\boldsymbol{m} \in R(N, \boldsymbol{m}_0)$ the number of tokens that the net can contain does not vary, i.e., if:*

$$\mathbf{1}^T \cdot \boldsymbol{m} = \sum_{p \in P} m[p] = \sum_{p \in P} m_0[p] = \mathbf{1}^T \cdot \boldsymbol{m}_0$$

It is easy to verify graphically if a marked net is strictly conservative. Indeed all transitions should have a number of "pre" arcs equal to the number of "post" arcs.

Note however, that such a condition is not necessary for strict conservativeness: there may exist a transition with a different number of "pre" and "post" arcs that never fires. The net in Fig. 10.7 is strictly conservative since the total number of tokens is always equal to two. The net in Fig. 10.8 is not strictly conservative.

A generalization of strict conservativeness is the following.

Definition 10.15. *A marked net $\langle N, \boldsymbol{m}_0 \rangle$ is* conservative *if there exists a vector of* positive *integers $\boldsymbol{x} \in \mathbb{N}_+^m$ such that for all reachable markings $\boldsymbol{m} \in R(N, \boldsymbol{m}_0)$ it is:*

$$\boldsymbol{x}^T \cdot \boldsymbol{m} = \boldsymbol{x}^T \cdot \boldsymbol{m}_0$$

i.e. the number of tokens weighted through $\boldsymbol{x}$ does not vary.

The net in Fig. 10.9 is not strictly conservative, but it is conservative. Indeed, consider the vector $\boldsymbol{x} = [1\ 1\ 2]^T$. It is easy to verify that for all reachable markings $\boldsymbol{m}$ it is $\boldsymbol{x}^T \cdot \boldsymbol{m} = \boldsymbol{x}^T \cdot \boldsymbol{m}_0 = 2$.

Conservativeness is related to boundedness.

Proposition 10.6. *If a marked net $\langle N, \boldsymbol{m}_0 \rangle$ is conservative then it is bounded.*

Note however, that there may also exist bounded nets that are not conservative. An example is given in Fig. 10.10 that shows a safe net that is not conservative: indeed its reachability set is $\{[1], [0]\}$, thus chosen an arbitrary positive integer x it holds $0 = x \cdot 0 \neq x \cdot 1 = x$.

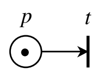

Fig. 10.10 A bounded Petri net that is not conservative

In the following Chapter 11 it will be shown how, using the incidence matrix, it is possible to compute a vector $\boldsymbol{x}$ with respect to whom the net is conservative.

10.4.3.4 Repetitiveness

Repetitiveness of a sequence of transitions ensures that the sequence can occur indefinitely.

Definition 10.16. *Given a marked net $\langle N, \boldsymbol{m}_0 \rangle$, let σ be a non empty sequence of transitions and $\boldsymbol{m} \in R(N, \boldsymbol{m}_0)$ a marking enabling it. Sequence σ is called* repetitive *if it can fire an infinite number of times at $\boldsymbol{m}$, i.e., it holds*

$$\boldsymbol{m}[\sigma\rangle \boldsymbol{m}_1 [\sigma\rangle \boldsymbol{m}_2 [\sigma\rangle \boldsymbol{m}_3 \cdots$$

A marked net $\langle N, \boldsymbol{m}_0 \rangle$ is repetitive *if there exists a repetitive sequence in $L(N, \boldsymbol{m}_0)$.*

The following proposition characterizes repetitive sequences.

Proposition 10.7. *Let σ be a non empty sequence of transitions such that $\boldsymbol{m}[\sigma\rangle\boldsymbol{m}'$. Sequence σ is repetitive if and only if $\boldsymbol{m} \leq \boldsymbol{m}'$.*

We distinguish two types of repetitive sequences.

Definition 10.17. *A repetitive sequence σ enabled at $\boldsymbol{m}$ is called:*

- stationary *if $\boldsymbol{m}[\sigma\rangle\boldsymbol{m}$,*
- increasing *if $\boldsymbol{m}[\sigma\rangle\boldsymbol{m}'$ with $\boldsymbol{m}' \gneq \boldsymbol{m}$.*

As an example, the net in Fig. 10.3(c) does not contain repetitive sequences. In the net in Fig. 10.8 repetitive sequences are t_1^k and t_4^k, with $k \in \mathbb{N}_+$: sequences t_1^k are increasing, while sequences t_4^k are stationary.

Increasing sequences exist only on unbounded nets.

Proposition 10.8. [12] *A marked net $\langle N, \boldsymbol{m}_0\rangle$ is bounded if and only if it does not admit increasing repetitive sequences.*

As discussed in Chapter 11 it is immediate to verify if a given sequence σ is repetitive (either stationary or increasing) using the incidence matrix of the net. Here we only consider the information given by the analysis of the reachability graph.

Proposition 10.9. *Consider a marked bounded net $\langle N, \boldsymbol{m}_0\rangle$ and its reachability graph. A sequence σ is stationary $\Longleftrightarrow$ there exists a directed cycle in the graph whose arcs form σ.*

In the net in Fig. 10.7 each stationary sequence corresponds to a cycle in the reachability graph. Sequences that correspond to elementary cycles are called *elementary*. As an example, t_1t_2 is an elementary sequence, while $t_1t_2t_1t_2$ is not elementary.

Proposition 10.10. *Consider a marked net $\langle N, \boldsymbol{m}_0\rangle$ and its coverability graph.*

- *A sequence σ is repetitive $\Longrightarrow$ there exists a directed cycle in the graph whose arcs form σ.*
- *A sequence σ is stationary $\Longleftarrow$ there exists a directed cycle in the graph that does not pass through markings containing ω and whose arcs form σ.*

Note that a coverability graph has always at least one cycle associated with an increasing sequence. Such is the case of sequence t_1 in the net in Fig. 10.8. Moreover, there can be cycles associated with non repetitive sequences. Such is the case of sequence t_3 in the net in Fig. 10.8: t_3 is not repetitive because its firing leads to decreasing of two units the number of tokens in p_2; however, this is hidden when t_3 fires from $[0\ \omega\ 1]^T$.

10.4.3.5 Reversibility

Reversibility implies that a system can always be reinitialized to its initial state. This is a desirable feature in many man-made systems.

Definition 10.18. *A marked net $\langle N, \boldsymbol{m}_0 \rangle$ is* reversible *if for all reachable markings $\boldsymbol{m} \in R(N, \boldsymbol{m}_0)$ it holds $\boldsymbol{m}_0 \in R(N, \boldsymbol{m})$, i.e., if from any reachable marking it is possible to reach back the initial marking $\boldsymbol{m}_0$.*

As an example, the net in Fig. 10.11(a) is reversible because from any reachable marking $m = [k]$, transition t_2 can fire k times leading the net back to the initial marking $m_0 = [0]$ in figure. On the contrary, the net in Fig. 10.11(b) is not reversible because any token that enters in p_2 can never be removed coming back to the initial marking.

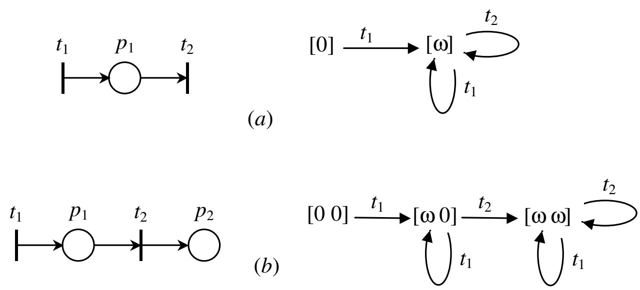

Fig. 10.11 (a) A reversible unbounded marked net and its coverability graph; (b) a non reversible unbounded marked net and its coverability graph

The reachability graph provides necessary and sufficient conditions for reversibiliy. On the contrary, the coverability graph only provides necessary conditions. This is formalized by the following two propositions whose validity derives from Propositions 10.1 and 10.2, respectively.

Proposition 10.11. *Consider a bounded marked net $\langle N, \boldsymbol{m}_0 \rangle$ and its reachability graph. The marked net is reversible $\Longleftrightarrow$ the graph is strongly connected.*

The reachability graph of the net in Fig. 10.7 is not strongly connected: as an example, there exists no directed path from marking $[1\ 0\ 1]^T$ to the initial marking.

Proposition 10.12. *Consider a marked net $\langle N, \boldsymbol{m}_0 \rangle$ and its coverability graph. The net is reversible $\Longrightarrow$ each ergodic[2] component of the graph contains a node $\boldsymbol{m}_\omega \geq_\omega \boldsymbol{m}_0$.*

As an example, the only ergodic component of the coverability graph of the reversible net in Fig. 10.11(a) contains marking $[\omega] \geq_\omega [0] = m_0$. Note however, that also the net in Fig. 10.11(b) has only one ergodic component that contains the marking $[\omega\ \omega]^T \geq_\omega [0\ 0]^T = \boldsymbol{m}_0$ and it is not reversible. Finally, it is possible to conclude

[2] Consider a maximal strongly connected component of a graph. Such a component is called *ergodic* if there are no edges leading from a node that belongs to the component to a node that does not belong to it. Otherwise the component is called *transient* [2].

by the only analysis of the coverability graph, that the net in Fig. 10.8 is not reversible: indeed it has two ergodic components, each one with a single marking $[0\ 0\ 1]^T$ and $[0\ \omega\ 1]^T$, respectively, none of them covering the initial marking.

Note, finally, that even if the coverability graph does not provide necessary and sufficient conditions for checking reversibility, such a property is decidable. In fact checking for reversibility reduces to checking if the initial marking $\boldsymbol{m}_0$ is a *home marking*, a problem that is known to be decidable [7] (see also Chapter 12, Definition 12.8).

10.4.3.6 Liveness and Deadlock

Liveness of a transition implies the possibility that it can always eventually fire, regardless of the current state of the net.

Definition 10.19. *Given a marked net $\langle N, \boldsymbol{m}_0 \rangle$, we say that a transitions t is:*

- dead *if no reachable marking enables it, i.e.,* $\forall \boldsymbol{m} \in R(N, \boldsymbol{m}_0)\ \neg \boldsymbol{m}[t\rangle$*;*
- quasi-live *if it is enabled by some reachable marking, i.e.,* $\exists \boldsymbol{m} \in R(N, \boldsymbol{m}_0)\ :\ \boldsymbol{m}[t\rangle$*;*
- live *if for all reachable markings* $\boldsymbol{m} \in R(N, \boldsymbol{m}_0)$*, t is quasi-live in* $\langle N, \boldsymbol{m} \rangle$*.*

In the net in Fig. 10.12 transition t_4 is dead, transitions t_1 and t_2 are quasi-live, transition t_3 is live. Note a fundamental difference between quasi-live transitions t_1 and t_2: t_1 can fire an infinite number of times, while t_2 may only fire once.

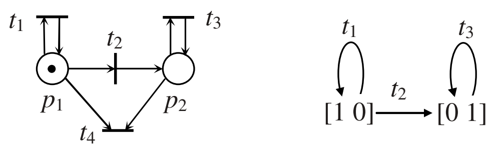

Fig. 10.12 A PN for the study of liveness

It is also possible to define the liveness property for a marked net.

Definition 10.20. *A marked net $\langle N, \boldsymbol{m}_0 \rangle$ is:*

- dead*, if all its transitions are dead;*
- not quasi-live*, if some of its transitions are dead and some are quasi-live;*
- quasi-live*, if all its transitions are quasi-live;*
- live*, if all its transitions are live.*

The net in Fig. 10.12 is not quasi-live because it contains both dead and quasi-live transitions. The two nets in Fig. 10.11 are both live.

Another important concept related to the notion of liveness is *deadlock* that denotes an anomalous state from which no further evolution is possible.

Definition 10.21. *Given a marked net* $\langle N, \boldsymbol{m}_0 \rangle$ *let* $\boldsymbol{m} \in R(N, \boldsymbol{m}_0)$ *be a reachable marking. We say that* $\boldsymbol{m}$ *is a* dead marking *if no transition is enabled at* $\boldsymbol{m}$*, i.e., if* $\langle N, \boldsymbol{m} \rangle$ *is dead. A marked net* $\langle N, \boldsymbol{m}_0 \rangle$ *is* deadlocking *if there exists a dead reachable marking.*

The net in Fig. 10.7 is deadlocking: marking $[0\ 0\ 2]^T$ is dead.

Once again the reachability graph provides necessary and sufficient conditions for the verification of liveness and deadlock.

Proposition 10.13. *Consider a bounded marked net* $\langle N, \boldsymbol{m}_0 \rangle$ *and its reachability graph.*

- *Transition* t *is dead* $\Longleftrightarrow$ *no arc labeled* t *belongs the graph.*
- *Transition* t *is quasi-live* $\Longleftrightarrow$ *an arc labeled* t *belongs the graph.*
- *Transition* t *is live* $\Longleftrightarrow$ *an arc labeled* t *belongs to each ergodic component of the graph.*
- *Reachable marking* $\boldsymbol{m}$ *is dead* $\Longleftrightarrow$ *node* $\boldsymbol{m}$ *in the graph has no output arc.*

The coverability graph provides necessary and sufficient conditions for the analysis of quasi-liveness, but only necessary conditions for the analysis of liveness.

Proposition 10.14. *Consider a marked net* $\langle N, \boldsymbol{m}_0 \rangle$ *and its coverability graph.*

- *Transition* t *is dead* $\Longleftrightarrow$ *no arc labeled* t *belongs the graph.*
- *Transition* t *is quasi-live* $\Longleftrightarrow$ *an arc labeled* t *belongs to the graph.*
- *Transition* t *is live* $\Longrightarrow$ *an arc labeled* t *belongs each ergodic component of the graph.*
- *Reachable marking* $\boldsymbol{m}$ *is dead* $\Longleftarrow$ *node* $\boldsymbol{m}_\omega$ *in the graph has no output arc and* $\boldsymbol{m}_\omega \geq_\omega \boldsymbol{m}$.

Note, finally, that even if the coverability graph does not provide necessary and sufficient conditions for checking liveness, such a property is decidable. In fact checking for liveness can be reduced to a reachability problem [13]. Thus liveness of a net is a decidable property.

10.5 Further Reading

Further details on the proposed topics can be found in the survey paper by Murata [12] and on the books of Peterson [13] and David and Alla [1].

Finally, we address to the book of Girault and Valk [8] for a discussion on the effectiveness of model checking in the verification of the properties introduced in Section 10.4.

References

1. David, R., Alla, H.: Discrete, Continuous and Hybrid Petri Nets. Springer (2005)
2. Diestel, R.: Graph Theory, 4th edn. Springer (2010)
3. Desel, J., Esparza, J.: Free Choice Petri Nets. Cambridge University Press (1995)
4. Desrochers, A.A., Al-Jaar, R.Y.: Applications of Petri Nets in Manufacturing Systems. IEEE Press (1995)
5. Diaz, M. (ed.): Petri nets. Fundamental Models, Verification and Applications. John Wiley and Sons, Inc. (2009)
6. Di Cesare, F., Harhalakis, G., Proth, J.M., Silva, M., Vernadat, F.B.: Petri Net Synthesis for Discrete Event Control of Manufacturing Systems. Kluwer (1993)
7. Finkel, A., Johnen, C.: The home state problem in transition systems. In: Rapport de Recherche, vol. 471, Univ. de Paris-Sud., Centre d'Orsay (1989)
8. Girault, C., Valk, R.: Petri Nets for Systems Engineering. Springer (2003)
9. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley (1979)
10. Karp, R., Miller, R.: Parallel program schemata. Journal of Computer and System Sciences 3(2), 147–195 (1969)
11. Martinez, J., Silva, M.: A simple and fast algorithm to obtain all invariants of a generalized Petri net. In: Informatik-Fachberichte: Application and Theory of Petri Nets, vol. 52. Springer (1982)
12. Murata, T.: Petri nets: properties, analysis and applications. Proceedings IEEE 77(4), 541–580 (1989)
13. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice-Hall (1981)
14. Petri, C.A.: Kommunication Mit Automaten. Institut für Instrumentelle, Mathematik (Bonn, Germany), Schriften des IIM 3 (1962)
15. Reisig, W.: Petri Nets: An Introduction. EATCS Monographs on Theoretical Computer Science (1985)
16. Reutenauer, C.: Aspects Mathématiques des Réseaux de Petri. Prentice-Hall International (1990)