

Table of Contents

1 Preface	2
2 Introduction	2
3 Preliminaries	4
4 EN Systems	6
4.1 Informal Introduction	6
The producer/consumer problem	7
The mutual exclusion problem	8
4.2 Nets	9
4.3 The Firing of Transitions	13
4.4 Concurrency	22
4.5 Fundamental Situations	27
(1) Causality	27
(2) Concurrency	28
(3) Conflict	29
(4) Confusion	30
5 Equivalences and Normal Forms	34
5.1 Equivalence	34
5.2 Reduction	42
5.3 Sequential EN Systems	45
5.4 Subsystems and Sequential Components	46
5.5 Contact-freeness	58
6 Processes	61
6.1 Partial Orders	62
6.2 Process Nets	64
6.3 Processes	69
6.4 Pruned Contracted Processes	79
7 Comparison of Partial and Linear Order	88
7.1 LPO-Equivalence and Firing Sequence Equivalence	88
7.2 Equivalence of Firing Sequences	96
8 Branching Processes	99
9 P/T Systems	109
9.1 Informal Introduction	109
9.2 Definitions	113
9.3 A Finiteness Algorithm	117
9.4 Covering Sets	121
9.5 Place Invariants	129
9.6 Marked Graphs	139
9.7 Free-Choice Systems	143

1 Preface

The aim of the course Theory of Concurrency 1 is to present an introduction into fundamental notions concerning concurrent systems, on the basis of a specific theoretical model (the so-called Petri Net). These lecture notes are meant to accompany the lectures. They contain a, mostly concise, overview of the treated material. The lectures are important for the associated intuitions, motivations, and proof techniques. The exercise class is meant to acquire the necessary technical skills.

These lecture notes contain several parts that are only meant for the more interested reader. Such parts start with an * and end at the first \square following it. These parts do not have to be known for the exam.

Acknowledgements. We wish to thank IJsbrand Jan Aalbersberg, George Leih, Iko Keesmaat, and Peter Hoogers for their contributions to these lecture notes. We thank Hendrik Jan Hoogeboom, Jetty Kleijn, and P.S.Thiagarajan for valuable comments throughout the years. We are especially grateful to Maurice ter Beek for his excellent job of translating the original text into English, and to Jurriaan Hage and Nikè van Vugt for their nice work of drawing all the figures.

2 Introduction

The area of Petri Nets was initiated by C.A.Petri in the early sixties ([Pet62]). Since then this area has been developed tremendously in both the theory and the applications. Although many other models of concurrent and distributed systems have been developed in the meantime, the Petri net model is still a central model for such systems. It is also often used as a yardstick for other models.

One of the main attractions of Petri nets is the way in which the basic aspects of concurrent systems are identified both conceptually and mathematically. The ease of conceptual modelling (based also on a natural graphical notation) makes Petri nets the model of choice in many applications. The natural way in which Petri nets allow to formally capture many of the basic notions and issues of concurrent systems contributed greatly to the development of a rich theory of concurrent systems based on Petri nets.

Petri nets is actually a generic name for a whole class of net-based models which can be divided into three main layers. The first layer is the most fundamental and is especially well suited for a thorough investigation of foundational issues of concurrent systems. The basic model here is that of Elementary Net Systems, or EN systems (introduced in [RozThi86, Thi87, Roz87]). This model is not suitable for practical applications because the size of the model explodes even for simple but nontrivial applications. The second layer is an “intermediate” model where one folds some repetitive features of EN systems in order to get more compact representations. The basic model here is Place/Transition Systems, or P/T systems (see for example [Pet81, Rei82]). An EN system can also be viewed as a special case of a P/T system. Finally, the third layer is that

of high-level nets, where one uses essentially algebraic and logical tools to yield “compact nets” that are suited to real-life applications. Predicate/Transition Nets (see, e.g., [Gen87]) and Coloured Petri Nets (see, e.g., [Jen92]) are the best known high-level models.

In these lecture notes, which are partly based on [Rei82, RozThi86, Thi87, Roz87], we present a comprehensive introduction to the theory of EN systems and P/T systems, covering both their structure and behaviour.

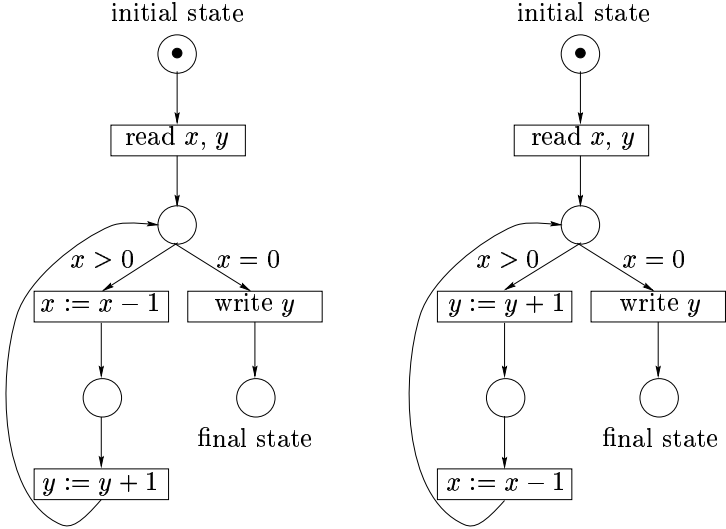


Fig. 1. Sequential addition programs.

Let us start with some very simple examples of concurrency. Figure 1 shows two sequential programs that compute the sum of two nonnegative integers. During execution of each program, the current point in the program (i.e., the current state of the program) can be indicated by marking the corresponding circle with a “token”. Execution starts in the initial state, as shown in Fig. 1. After execution (with the program in the final state), the value of y is the sum of the two values that were read for x and y , and x has value 0. Obviously, the order of execution of the two assignments $x := x - 1$ and $y := y + 1$ is irrelevant, and thus they may as well be executed concurrently. A program in which they are indeed executed concurrently is shown in Fig. 2. The effect of the fork instruction is that the pre-states of both assignments are activated, i.e., both program points are currently executed, independent of each other. As soon as execution arrives in the post-states of the assignments, and thus both assignments have been executed, the join instruction activates the program point where x is tested on 0. Note that during execution of the assignments there were

two tokens in the program, but now there is just one again.

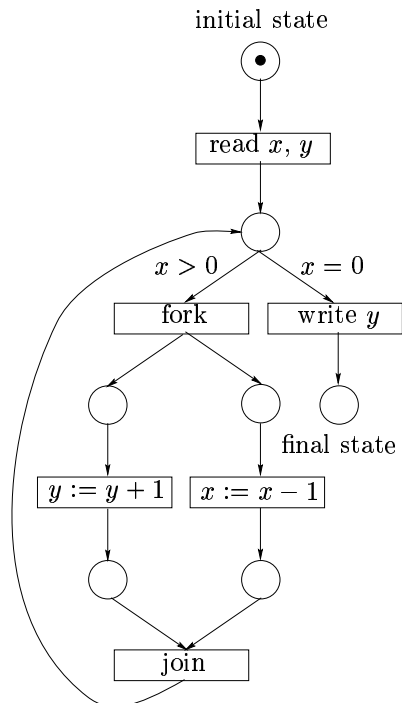


Fig. 2. A concurrent addition program.

Obviously, the main advantage of concurrent programs is that, in general, they are faster than their sequential counterparts. To illustrate this, consider the concurrent program in Fig. 3. If A and B are two given disjoint sets of integers (both finite and nonempty), then the program divides $A \cup B$ into two disjoint subsets A' and B' of the same cardinality as A and B , respectively, such that $\max(A') < \min(B')$. It does this by repeatedly computing $\max(A)$ and $\min(B)$, and interchanging them when they do not satisfy the requirement. Since operations on sets (such as finding the maximum) are slow in comparison to operations on integers, they should be executed concurrently as often as possible.

3 Preliminaries

In this section we recall some well-known concepts and notation concerning sets and words.

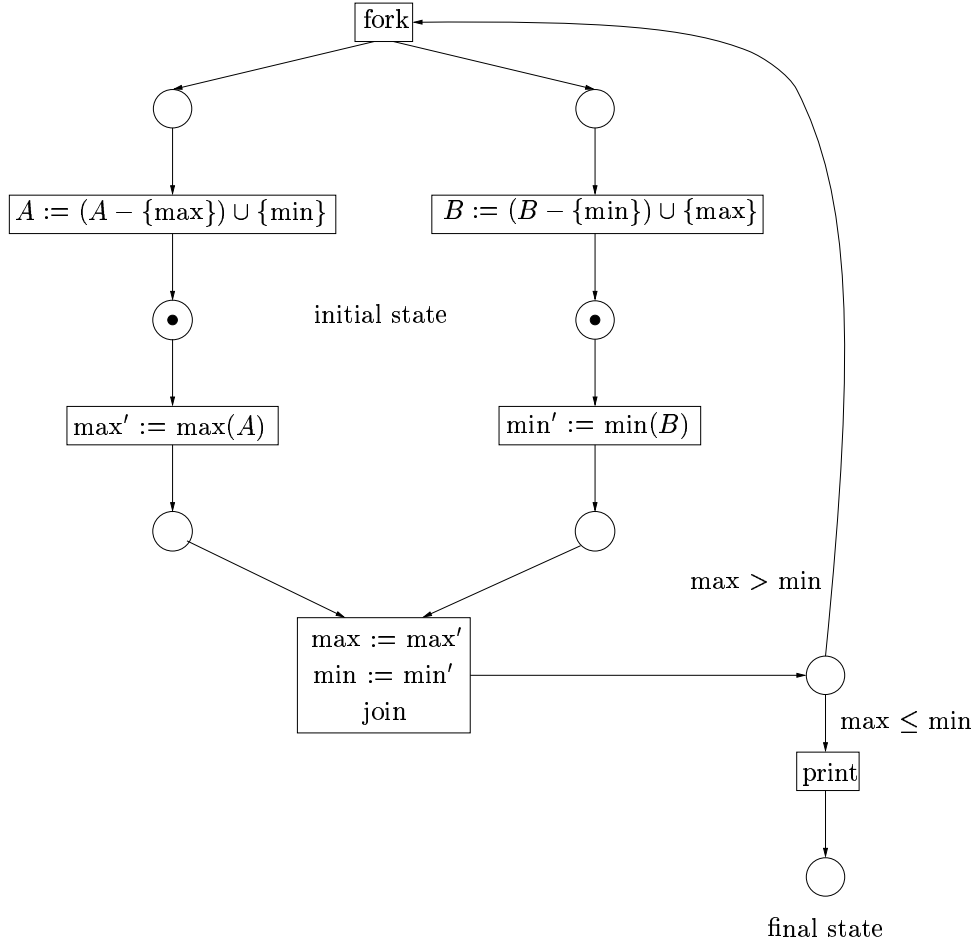


Fig. 3. Concurrent operations on sets.

The set of integers is denoted by $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$. The sets of non-negative and positive integers are denoted by $\mathbb{N} = \{0, 1, 2, \dots\}$ and $\mathbb{N}_+ = \{1, 2, 3, \dots\}$, respectively.

For a set A , $\mathcal{P}(A)$ is the set of all subsets of A , and $\#A$ denotes the number of elements of A . For $B \subseteq A$, $C_B : A \rightarrow \{0, 1\}$ denotes the characteristic function of B , i.e., $C_B(a) = 1$ if $a \in B$ and $C_B(a) = 0$ if $a \notin B$. We consider total functions only, i.e., if $f : A \rightarrow A'$, then $f(a)$ is defined (and in A') for every $a \in A$. If $f : A \rightarrow A'$ and $B \subseteq A$, then $f \upharpoonright B$ denotes the restriction of f to B , i.e., the function $B \rightarrow A'$ defined by $(f \upharpoonright B)(b) = f(b)$. A function $f : A \rightarrow A'$ is injective if $f(a_1) \neq f(a_2)$ whenever $a_1 \neq a_2$, f is surjective if for every $a' \in A'$

there exists an $a \in A$ such that $f(a) = a'$, and f is a bijection (between A and A') if f is injective and surjective.

For a binary relation $R \subseteq A \times A$, the transitive (and reflexive) closure of R is denoted by R^+ (R^* , respectively). Hence, $(a, b) \in R^+$ iff there exist $a_1, \dots, a_n \in A$, with $n \geq 1$, such that $a_1 = a, a_n = b$ and $(a_i, a_{i+1}) \in R$ for all $1 \leq i \leq n-1$, and $(a, b) \in R^*$ iff $(a, b) \in R^+$ or $a = b$.

For an alphabet Σ , Σ^* is the set of all words over Σ , and λ is the empty word. The length of a word $x \in \Sigma^*$ is denoted by $|x|$; thus, $|\lambda| = 0$. A word y is a prefix of a word x if there exists a word z such that $x = yz$. A language over Σ is a subset of Σ^* . We note that any finite set can be viewed as an alphabet.

Let Σ and Δ be alphabets and let $h : \Sigma \rightarrow \Delta$. We extend h to a function from Σ^* to Δ^* in the following way: $h(\lambda) = \lambda$ and, for a word $x = \sigma_1 \cdots \sigma_n$, with $n > 0$ and $\sigma_i \in \Sigma$ for all $1 \leq i \leq n$, $h(x) = h(\sigma_1) \cdots h(\sigma_n)$. Furthermore, for a language $L \subseteq \Sigma^*$ we define $h(L) = \{h(x) \mid x \in L\}$. In formal language theory, h is called a letter-to-letter homomorphism.

4 EN Systems

4.1 Informal Introduction

Elementary net systems form the most fundamental class of Petri nets. Like most of the models that fall under the generic name Petri nets, elementary net systems are a net-based model. The basic intuition behind net-based models is that such a model consists of a net, and of the rules of a token game played on the net. The net describes the static structure of a concurrent system, and the rules describe the dynamic behaviour of the concurrent system. Different classes of Petri nets differ by the sort of underlying nets and/or rules of the dynamic token game.

In this section we introduce the notion of a net for elementary net systems and the rules for playing the token game on such a net. These rules tell us how to get from one global state of the (elementary net) system to another global state. They give thus a potential state space of the system. In order to get the actual state space one has to specify the initial global state of the system: the actual state space is then obtained by starting from this initial global state and using the rules of the game. Hence a specific elementary net system is given by a net and an initial global state. The rules of the token game are the same for all elementary net systems.

A characteristic property of Petri nets is that the global state is a set (or multiset) of local states and that the transition from a global state to another global state is given by one or more local transitions, where local transitions act on local states. More precisely, a local transition replaces a subset of the local states of a global state by another such subset. Thus, Petri nets can be viewed as set transition systems or set replacement systems (as opposed to the string rewriting systems of formal language theory).

The local states of an elementary net system are also called places, and the local transitions are just called transitions. The net is a finite directed graph;

its nodes are the potential places and transitions of the system, and its edges assign input and output places to each transition. Thus the net defines for each transition which local states are replaced by which other local states. The actual global state is indicated by putting tokens on the places that are the actual local states, and the token game is played by moving the tokens around, according to the transitions that are executed.

Thus, in elementary net systems there are only finitely many potential local states, and a global state is a finite set of local states. An important consequence of this setup is that the state space of an elementary net system is finite. This means that elementary net systems are a model of finite-state concurrent systems (just as finite automata model finite-state sequential systems).

This section discusses the basic notions of a net, the token game played on the net, and the state space of an elementary net system. However, before turning to the formal definitions, we wish to give the reader some intuitive feeling for the net-based approach to concurrent systems, through two easy and well-known examples.

Very often, a concurrent system consists of several communicating concurrent components, each of which is sequential (i.e., executes its actions one after the other). Concurrency of the components means that one component can perform (some of) its actions independently of (and thus simultaneously with) the actions of another component. From time to time the components communicate with each other, which means that they have to wait for each other and interact by synchronizing their actions. In modelling such a concurrent system by an elementary net system, a local state is a state of a component, and a local transition is either an action of one component that is independent of the other components, or consists of a synchronized action shared by several communicating components.

The producer/consumer problem Figure 4 shows an elementary net sys-

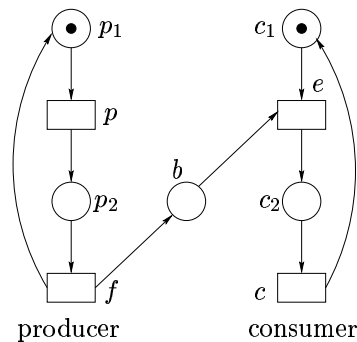


Fig. 4. The producer/consumer problem.

tem that models the well-known producer/consumer problem. Places (i.e., local states) are indicated by circles, and transitions by rectangles. For each transition, the local states with an edge to the transition (the input places) are replaced by the local states with an edge from the transition (the output places).

The system can be viewed as consisting of three components: the producer, the consumer, and the buffer. The producer puts “production units” in the buffer (which can contain only one such unit), and the consumer takes units from the buffer.

The producer is always in exactly one of its local states p_1 or p_2 . If it is in state p_1 , it can execute transition p (i.e., produce a unit) and go into state p_2 (which means that the token is moved from p_1 to p_2). Then it can synchronize with the buffer component through the execution of the shared transition f (i.e., fill the buffer). The producer then returns to state p_1 , and the buffer place b now also contains a token, to indicate that the buffer is full. The buffer can be emptied again by the shared transition e (i.e., empty the buffer) in which the consumer and the buffer synchronize, and the consumer moves to its local state c_2 . Then the consumer can execute transition c (i.e., consume the unit) and return to its local state c_1 . In the meantime the producer may have executed transition p independently, and may have executed transition f as soon as the buffer was empty.

Note that the producer and the consumer never communicate directly, i.e., never perform shared actions. However, they cooperate asynchronously via the buffer.

The mutual exclusion problem Figure 5 shows an elementary net system that models the well-known mutual exclusion problem. The system consists of three components: component 1, component 2, and the “permission” component. Components 1 and 2 compete for access to the same shared resource (such as a printer). At any moment of time at most one of these components can use the resource. It is the task of the permission component to schedule access to the resource. The availability of the resource is represented by a “permission” in place p ; this permission is indicated by the presence of a token in place p .

A “critical section” is a part of a component which uses the shared resource and hence needs protection against a possible “disturbance” by the other component. The critical section of component 1 is represented by place c_1 and that of component 2 by place c_2 . The noncritical part of component i (for $i = 1, 2$) is represented by places r_i (the remainder) and w_i (wait). Thus, component i has local states c_i , r_i , and w_i . Component i can perform the actions in_i , out_i , and d_i (entering the critical section, exiting the critical section, and an action outside the critical section). To enter or exit the critical section it has to synchronize with the permission component, which has local states p (the resource is not used), c_1 (the resource is used by component 1), and c_2 (the resource is used by component 2), and can perform the actions in_i and out_i , $i = 1, 2$.

In the global state given in Fig. 5, components 1 and 2 compete for permission. Clearly, only one of the transitions in_1 or in_2 can be executed, permitting

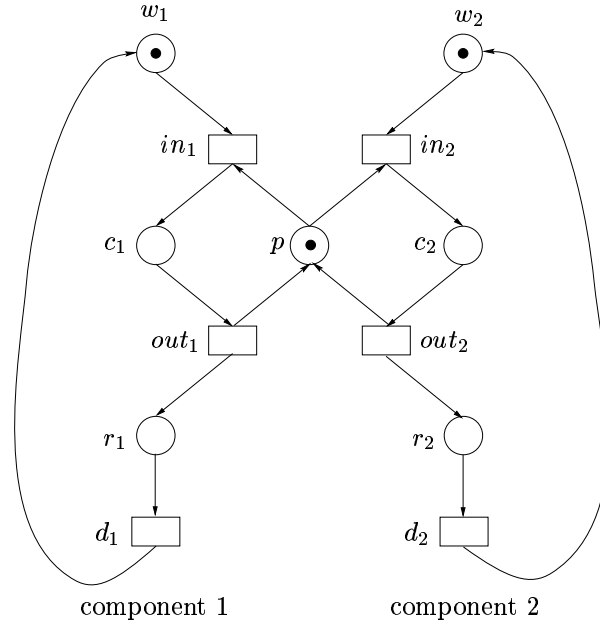


Fig. 5. The mutual exclusion problem.

either component 1 or component 2 (respectively) to access the resource. When component i has obtained permission and has finished using the resource, it returns the permission to place p through transition out_i . Note that components 1 and 2 never communicate directly, i.e., never perform shared actions. However, they solve their conflict by communicating with the permission component. Note also that the permission component has places in common with the other two components. Thus, place c_1 represents both a local state of component 1 and a local state of the permission component.

4.2 Nets

The main part of an elementary net system is its net, which is defined as follows.

Definition 1. A *net* is a triple $N = (P, T, F)$, where:

- (1) P and T are finite sets with $P \cap T = \emptyset$,
- (2) $F \subseteq (P \times T) \cup (T \times P)$,
- (3) for every $t \in T$ there exist $p, q \in P$ such that $(p, t), (t, q) \in F$, and
- (4) for every $t \in T$ and $p, q \in P$, if $(p, t), (t, q) \in F$, then $p \neq q$.

Elements of P are called *places*, elements of T are called *transitions*, elements of $X = P \cup T$ are called *elements (of N)*, and F is called the *flow relation (of*

N). We will also use the notations P_N , T_N , X_N , and F_N for P , T , X , and F , respectively. Note that N can be the empty net, i.e., $N = (\emptyset, \emptyset, \emptyset)$.

For each $x \in X$, $\bullet x = \{y \in X \mid (y, x) \in F\}$ is the *input-set* (or *pre-set*) of x and $x^\bullet = \{y \in X \mid (x, y) \in F\}$ is the *output-set* (or *post-set*) of x ; the set $\bullet x \cup x^\bullet$ is called the *neighbourhood* of x , denoted by $\mathbf{nbh}(x)$. Whenever we want to indicate the net under consideration, the notations $(\bullet x)_N$, $(x^\bullet)_N$, and $\mathbf{nbh}_N(x)$ will be used. For $Y \subseteq X$, we write $\bullet Y = \bigcup_{x \in Y} \bullet x$, $Y^\bullet = \bigcup_{x \in Y} x^\bullet$, and $\mathbf{nbh}(Y) = \bullet Y \cup Y^\bullet$ and the terminology is carried over correspondingly. Note that the flow relation F is completely determined if $\bullet t$ and t^\bullet are known for every transition $t \in T$.

Conditions (3) and (4) of Definition 1 can now also be formulated as follows. For each transition t : $\bullet t \neq \emptyset$, $t^\bullet \neq \emptyset$, and $\bullet t \cap t^\bullet = \emptyset$. These requirements do not always appear in the literature, but we use them for two reasons. Firstly because they are quite natural, and secondly because they allow us to avoid many unnecessary technicalities. In the sequel, we will now and then indicate where these conditions are used.

It is clear that a net $N = (P, T, F)$ can in fact be seen as a directed graph G_N : the nodes of G_N are the elements of X and there is an edge from x to y iff $(x, y) \in F$. Thus, the reflexive and transitive closure F^* and the transitive closure F^+ indicate the paths in G_N : $(x, y) \in F^*$ iff there is a (possibly empty) directed path from x to y , and $(x, y) \in F^+$ iff there is a nonempty directed path from x to y .

Note that in fact G_N is a bipartite graph since $\{P, T\}$ is a partition of X and an edge leads either from P to T or from T to P . A difference between a net and an arbitrary bipartite graph is that the partition is explicitly given and that an explicit distinction is made between the two sets P and T of the partition, by the order in which they appear in the tuple (P, T, F) .

Since nets are graphs, the standard conventions for drawing graphs can be applied to nets. In a drawing of a net, places are represented by circles and transitions by rectangles.

Example 1. Let $N = (P, T, F)$ be the net with $P = \{p_1, p_2, p_3\}$, $T = \{t_1, t_2, t_3\}$, and $F = \{(p_1, t_2), (p_1, t_3), (p_2, t_2), (p_2, t_3), (p_3, t_1), (t_1, p_1), (t_2, p_3), (t_3, p_3)\}$. N is drawn in Fig. 6.

Since every net N corresponds in a natural way to a graph G_N , one can classify nets by “structural”, i.e., graph-theoretic properties. In particular, the following notions will be used in the sequel.

Definition 2. A net $N = (P, T, F)$

- (1) is *acyclic* if, for every $x \in X$, $(x, x) \notin F^+$,
- (2) is *P-simple* if, for all $p, q \in P$, $(\bullet p = \bullet q$ and $p^\bullet = q^\bullet)$ implies $p = q$,
- (3) is *T-simple* if, for all $s, t \in T$, $(\bullet s = \bullet t$ and $s^\bullet = t^\bullet)$ implies $s = t$, and
- (4) has *no isolated places* if, for all $p \in P$, $\mathbf{nbh}(p) \neq \emptyset$.

Note that, due to condition (3) in Definition 1, transitions are never isolated.

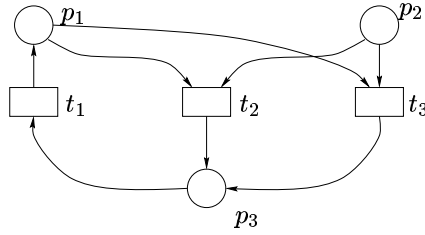


Fig. 6. A net.

Example 2. The net N from Example 1 (Fig. 6) is cyclic since, e.g., $(p_1, p_1) \in F^+$. N is P-simple, but not T-simple because $\bullet t_2 = \bullet t_3$ and $t_2^\bullet = t_3^\bullet$. N has no isolated places. The net in Fig. 7 is P-simple and T-simple. It has an isolated place and is cyclic.

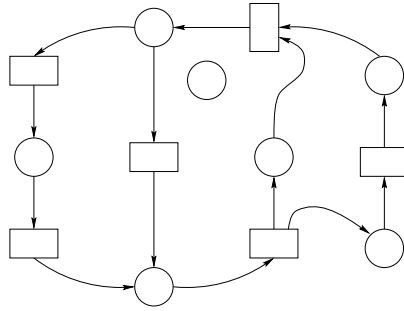


Fig. 7. A net that is P-simple and T-simple.

In considerations concerning the structure of a net, the concrete elements of the net are not important. To express this, we use the following notion of isomorphism. Note that, in the definition, the order of P and T in the tuple (P, T, F) is taken into account; this is the way in which places and transitions are distinguished formally (cf. Example 3(1)).

Definition 3. Two nets $N = (P, T, F)$ and $N' = (P', T', F')$ are *isomorphic*, denoted by $N \equiv N'$, if there exist two bijections $\alpha : P \rightarrow P'$ and $\beta : T \rightarrow T'$, such that for every $p \in P$ and $t \in T$,

$$(p, t) \in F \text{ iff } (\alpha(p), \beta(t)) \in F' \text{ and}$$

$$(t, p) \in F \text{ iff } (\beta(t), \alpha(p)) \in F'.$$

If we want to be more specific, we can say in the above situation that N and N' are (α, β) -isomorphic, denoted by $N \equiv_{\alpha, \beta}^{\alpha} N'$. The ordered pair (α, β) is called an *isomorphism between N and N'* .

The conditions for α and β in Definition 3 can also be formulated as follows: for every $t \in T$, $\bullet\beta(t) = \alpha(\bullet t)$ and $\beta(t)\bullet = \alpha(t\bullet)$, where, clearly, $\bullet\beta(t)$ stands for $(\bullet\beta(t))_{N'}$, $\bullet t$ stands for $(\bullet t)_N$, and likewise for the output-sets.

Example 3. (1) Let N and N' be the two nets in Figs. 8 and 9, respectively. It is clear that $N \equiv N'$ does not hold; in fact, in N the output-set of every transition is a singleton but in N' there are transitions with an output-set of cardinality 2. However, the graphs G_N and $G_{N'}$ are isomorphic graphs: there is a graph isomorphism γ between G_N and $G_{N'}$, viz. γ defined by $\gamma(t_1) = p_2$, $\gamma(t_2) = p_1$, $\gamma(t_3) = p_3$, $\gamma(p_1) = t_3$, $\gamma(p_2) = t_1$ and $\gamma(p_3) = t_2$. This isomorphism does not preserve the two sorts of the bipartition of the nets because it maps transitions into places, and places into transitions!

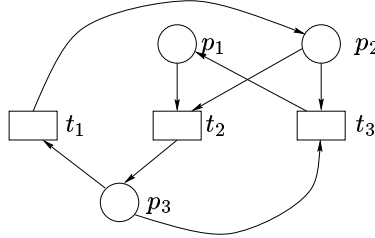


Fig. 8. A net N .

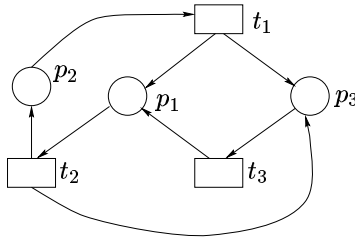


Fig. 9. A net N' , not isomorphic with the net N of Fig. 8.

(2) Let N'' be the net in Fig. 10. Let α and β be the bijections from P_N to $P_{N''}$ and from T_N to $T_{N''}$ defined by $\alpha(p_1) = p_8$, $\alpha(p_2) = p_5$, $\alpha(p_3) = p_1$,

$\beta(t_1) = t_5$, $\beta(t_2) = t_3$, and $\beta(t_3) = t_4$. Now N and N'' are (α, β) -isomorphic, and so $N \equiv_{\beta}^{\alpha} N''$. The isomorphism (α, β) does preserve the sorts: it maps transitions into transitions, and places into places.

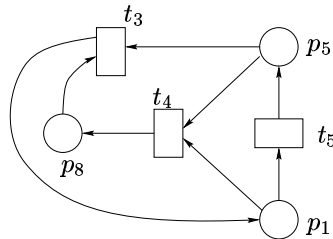


Fig. 10. A net N'' , isomorphic with the net N of Fig. 8.

4.3 The Firing of Transitions

A global state of a concurrent system consists of (is a set of) local states, where a local state can often be viewed as the state of a component of the system. A global state transition consists of local transitions. During a local transition a “small” number of local states change, for instance as the result of a communication between some components of the system. Thus, the global states are “distributed”, and so are the global state transitions.

When modelling a concurrent system by a net, the local states of the system are represented by the places of the net and its local transitions are represented by the transitions of the net, together with the flow relation. A global state of the system is thus represented by a set of places; such a set of places will be called a configuration.

Definition 4. A *configuration* of a net $N = (P, T, F)$ is a subset of P .

Graphically, a configuration $C \subseteq P$ is represented by placing a “token” (i.e., a fat dot, or a thumbtack when we use transparencies) in every circle corresponding to a place in C . Hence a single token represents a local state of the system, often corresponding to the state of a component of the system. Since we mark the net with tokens, a configuration is also called a *marking* of the net.

We are now ready to define the most fundamental Petri Net model, the “elementary net system”. It is a net together with the initial configuration of the system.

Definition 5. An *elementary net system*, EN system for short, is a quadruple $M = (P, T, F, C_{in})$, where:

- (1) (P, T, F) is a net and
- (2) $C_{in} \subseteq P$ is the *initial configuration*.

For an EN system $M = (P, T, F, C_{in})$, we denote by $\mathbf{und}(M)$, P_M , T_M , F_M , and $(C_{in})_M$ the net (P, T, F) , P , T , F , and C_{in} , respectively. Furthermore, we will call $\mathbf{und}(M)$ the *underlying net of M* , the places will also be called *conditions*, and the transitions will also be called *events* (in the literature, when places are called conditions, a configuration is often called a *case*). Finally, the terminology and notations concerning nets carry over to EN systems through their underlying nets. Note that M can be the empty EN system, i.e., $M = (\emptyset, \emptyset, \emptyset, \emptyset)$.

We represent an EN system M graphically by representing $\mathbf{und}(M)$ graphically and marking the initial configuration $(C_{in})_M$ by tokens.

Example 4. Let $N = (P, T, F)$ be the net in Fig. 11. Then $M = (P, T, F, C_{in})$, with $C_{in} = \{p_1, b, c_1\}$, is an EN system with underlying net N . The EN system M is drawn in Fig. 12. Note that M is the producer/consumer system of Fig. 4, with a full buffer.

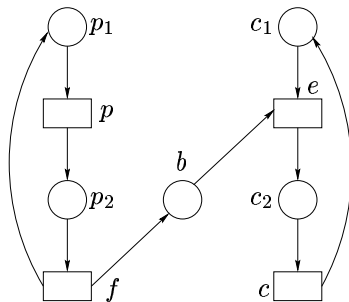


Fig. 11. The underlying net of the EN system of Fig. 12.

Until now we have considered nets and EN systems only as static objects. We now turn to the definition of the dynamic behaviour of EN systems. This, however, is not as easy as it is for sequential systems (such as finite-state automata). In fact, it is hard to express in a mathematical, convincing way that certain events, such as the local state transitions of a concurrent system, occur independently, “at the same time”. For this reason we will consider several definitions of the behaviour of an EN system, in which the concurrency of events is captured in different ways (without going into philosophical discussions about the concept of time). We start with the simplest case, where we consider how a configuration of an EN system is transformed by the execution of one transition

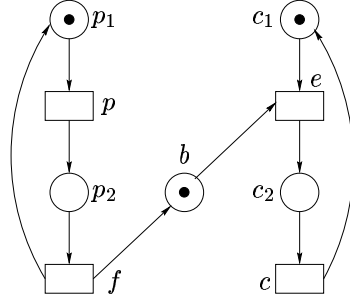


Fig. 12. An EN system.

of the system. Hence, in this case we do not yet consider concurrent events. In other words, we start by defining the sequential behaviour of the EN system.

A local transition of a concurrent system replaces a subset of the local states of a global state by another such subset. In the corresponding EN system, when the input-set of the transition is a subset of the current configuration, execution of the transition consists of replacing the input-set by the output-set of the transition. This is also called a firing of the transition.

Definition 6. Let $M = (P, T, F, C_{in})$ be an EN system and let $t \in T$.

(1) Let $C \subseteq P$ be a configuration. Then t has *concession* in C (or t can be fired in C , or t is enabled in C) if $\bullet t \subseteq C$ and $t^\bullet \cap C = \emptyset$, written as $t \mathbf{con} C$.

(2) Let $C, D \subseteq P$. Then t fires from C to D if $t \mathbf{con} C$ and $D = (C - \bullet t) \cup t^\bullet$, written as $C[t]D$; t is also called a *sequential step* from C to D .

When it is necessary to indicate the EN system M under consideration, the index M is used: we then write $t \mathbf{con}_M C$ and $C[t]_M D$ instead of $t \mathbf{con} C$ and $C[t]D$. In the literature $C[t]$ is often written instead of $t \mathbf{con} C$.

Note that a transition t can only be fired if it has both *input-concession* (i.e., $\bullet t \subseteq C$) and *output-concession* (i.e., $t^\bullet \cap C = \emptyset$). Note also that, though we write $C[t]D$, the new configuration D is uniquely determined by C and t , i.e., if $C[t]D_1$ and $C[t]D_2$, then $D_1 = D_2$.

The firing of a transition t is also called the *occurrence* of transition t , or the occurrence of event t . To represent the firing of a transition graphically, we play the so-called “token game” as follows. If, in a given configuration (token marking) C , there exists a transition (rectangle) t such that every circle that corresponds to an element of the input-set of t contains a token, and every circle that corresponds to an element of the output-set of t does not contain a token, then t can be fired (i.e., t has concession). The firing of t consists of removing a token from every circle that corresponds to an element of the input-set of t and placing a token in every circle that corresponds to an element of the output-set of t , cf. Fig. 13. In this way, the token marking that corresponds to the

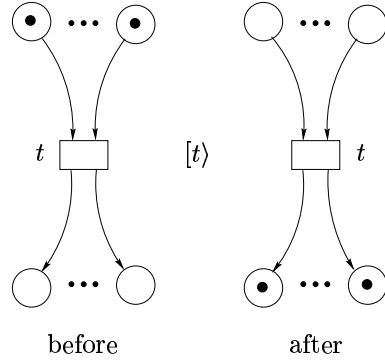


Fig. 13. Firing of transition t .

configuration $(C - \bullet t) \cup t \bullet$ is obtained. Note that this configuration can also be written as $(C \cup t \bullet) - \bullet t$, since for arbitrary sets A , B and C the following holds if A and B are disjoint: $(C - A) \cup B = (C \cup B) - A$ (and $\bullet t \cap t \bullet = \emptyset$ due to Definition 1(4)).

Example 5. Let M be the EN system from Example 4 (Fig. 12). Then transitions p and e have concession in C_{in} , i.e., $p \text{ con } C_{in}$ and $e \text{ con } C_{in}$; transitions f and c do not have concession in C_{in} . Both $C_{in}[p]\{p_2, b, c_1\}$ and $C_{in}[e]\{p_1, c_2\}$ hold. The configuration $\{p_1, c_2\}$ is drawn in Fig. 14. Also, e.g., $\{p_2, b, c_1\}[e]\{p_2, c_2\}$, $\{p_2, c_2\}[f]\{p_1, b, c_2\}$, and $\{p_1, b, c_2\}[c]C_{in}$ hold.

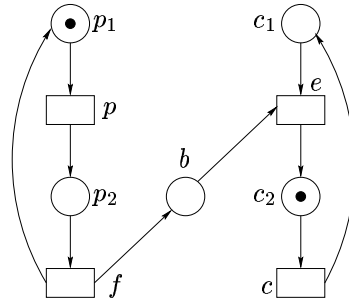


Fig. 14. The EN system of Fig. 12 after firing transition e .

We can also view the occurrence of event t as a *change of the values of certain conditions* (= places). The idea behind the definition of concession (Definition 6)

is that the result of the occurrence of an event must be observable in each component of the system that is involved in this event. Hence the event t can only occur if all pre-conditions hold (i.e., $\bullet t \subseteq C$) and all post-conditions do not hold (i.e., $t^\bullet \cap C = \emptyset$). After the event has occurred, the post-conditions hold and the pre-conditions no longer hold. Hence, viewing places as booleans, the occurrence of t can be intuitively described by the execution of the following conditional statement, in which all booleans in $\bullet t = \{p_1, \dots, p_m\}$ and $t^\bullet = \{q_1, \dots, q_n\}$ change their value:

```

if  $p_1$  and  $\dots$  and  $p_m$  and (not  $q_1$ ) and  $\dots$  and (not  $q_n$ )
then begin  $p_1 := \text{false}$  ;  $\dots$  ;  $p_m := \text{false}$  ;
            $q_1 := \text{true}$  ;  $\dots$  ;  $q_n := \text{true}$ 
end.

```

It must be observed that the execution of this statement is an atomic, indivisible, action and that the execution only takes place if the boolean condition (between the **if** and the **then**) is satisfied. Also note that the order of the assignments is irrelevant.

The next result shows that there is an alternative, symmetric, way of defining the firing of a transition.

Lemma 7. *Let $M = (P, T, F, C_{in})$ be an EN system. Let $t \in T$ and let $C, D \subseteq P$. Then $C[t]D$ iff $C - D = \bullet t$ and $D - C = t^\bullet$.*

Proof. For arbitrary sets A, B, C, D , the following two statements are equivalent: (1) $A \subseteq C$, $B \cap C = \emptyset$, and $D = (C - A) \cup B$, (2) $C - D = A$ and $D - C = B$. \square

Let $M = (P, T, F, C_{in})$ be an EN system. Assume that an event t takes place in a configuration C and leads to configuration D ($C[t]D$ in our notation). Then the amount of change of the configuration that is caused by t , can be given by the pair $(C - D, D - C)$: the conditions in $C - D$ stop being valid and the conditions in $D - C$ start being valid. From Lemma 7 now follows that the amount of change caused by an occurrence of an event is determined only by the event itself, i.e., it is independent of the configuration in which it takes place. Thus we have:

$(\forall t \in T)(\forall C_1, D_1, C_2, D_2 \subseteq P) : \text{if } C_1[t]D_1 \text{ and } C_2[t]D_2, \text{ then}$
 $C_1 - D_1 = C_2 - D_2 \text{ and } D_1 - C_1 = D_2 - C_2.$

This holds because, according to Lemma 7, $C_1 - D_1 = \bullet t = C_2 - D_2$ and $D_1 - C_1 = t^\bullet = D_2 - C_2$. Thus, each event t determines unambiguously the amount of change it causes (when it occurs); this amount of change is given by its *characteristic pair* $\mathbf{cp}(t) = (\bullet t, t^\bullet)$. However, there can be two distinct events t_1 and t_2 that cause the same amount of change, i.e., they have the same characteristic pair: $\bullet t_1 = \bullet t_2$ and $t_1^\bullet = t_2^\bullet$. This cannot happen in a T-simple EN system, because in such a system a transition t is completely determined by its characteristic pair. Consequently, a T-simple EN system satisfies the following *principle of extensionality*:

$$(\forall t_1, t_2 \in T)(\forall C_1, D_1, C_2, D_2 \subseteq P) : \text{if } C_1[t_1]D_1 \text{ and } C_2[t_2]D_2, \text{ then} \\ t_1 = t_2 \text{ iff } C_1 - D_1 = C_2 - D_2 \text{ and } D_1 - C_1 = D_2 - C_2.$$

This means that, in a T-simple EN system, distinct transitions cause different amounts of change in all configurations.

For a given EN system we are only interested in those configurations that can be reached from the initial configuration by the repeated firing of transitions, and in the (sequences of) transitions that lead to these configurations. Knowing the effect of firing a single transition allows us to define the effect of several transitions that fire one after the other. Such a sequence of transitions can also be considered as an observation of the system by a sequential observer (i.e., an observer that can observe only one event at a time). Formally, a sequence of transitions is a word over the alphabet T , i.e., an element of T^* .

Definition 8. Let $M = (P, T, F, C_{in})$ be an EN system.

(1) Let $t_1 \cdots t_n \in T^*$, with $n \geq 0$ and $t_1, \dots, t_n \in T$. Let $C, D \subseteq P$. Then $t_1 \cdots t_n$ *fires from* C *to* D if there exist configurations $C_0, C_1, \dots, C_n \subseteq P$ with $C_0 = C$, $C_n = D$ and $C_{i-1}[t_i]C_i$ for all $1 \leq i \leq n$, written as $C[t_1 \cdots t_n]D$.

(2) Let $x \in T^*$ and $C \subseteq P$. Then x *has concession in* C (or x *can be fired in* C , or x *is enabled in* C) if there exists a $D \subseteq P$ such that $C[x]D$, written as $x \text{ con } C$.

(3) $x \in T^*$ is a *firing sequence of* M if $x \text{ con } C_{in}$. The set of all firing sequences of M is denoted by $\text{FS}(M)$.

(4) $C \subseteq P$ is a *reachable configuration of* M if there exists an $x \in \text{FS}(M)$ with $C_{in}[x]C$. The set of all reachable configurations of M is denoted by \mathbb{C}_M .

(5) $t \in T$ is a *useful transition of* M if there exists a reachable configuration C of M such that $t \text{ con } C$. The set of useful transitions of M is denoted by $\text{use}_M(T)$, or just $\text{use}(T)$ when M is clear from the context.

(6) $t \in T$ is a *live transition of* M if for each $C \in \mathbb{C}_M$ there exists an $x \in T^*$ with $xt \text{ con } C$.

Note that a configuration C is reachable (i.e., $C \in \mathbb{C}_M$) if there exist transitions $t_1, \dots, t_n \in T$ (with $n \geq 0$) and configurations C_1, \dots, C_n such that $C_n = C$ and $C_{in}[t_1]C_1[t_2]C_2 \cdots [t_n]C_n$. Hence, \mathbb{C}_M can also be defined as the smallest set of configurations for which the following holds: (1) $C_{in} \in \mathbb{C}_M$ and (2) if $C \in \mathbb{C}_M$ and $C[t]D$ for some $t \in T$, then $D \in \mathbb{C}_M$. Properties of reachable configurations can thus be proved by induction in the following way. Let $P(C)$ be a property of (reachable) configurations C of M . Assume that it has been proved that (1) $P(C_{in})$ and (2) if $C[t]D$ and $P(C)$, then $P(D)$. Then $P(C)$ holds for all $C \in \mathbb{C}_M$. We call this a *proof by induction on* C of the statement $\forall C \in \mathbb{C}_M : P(C)$.

Intuitively, a transition is useful when it can eventually be fired starting from the initial configuration, and it is live when it can eventually be fired starting from any reachable configuration. Thus every live transition is useful.

Example 6. Let $M = (P, T, F, C_{in})$ be the EN system of Example 4 (Fig. 12). From the discussion in Example 5 it follows that the sequence of transitions

$pef \in T^*$ fires from $\{p_1, b, c_1\}$ to $\{p_1, b, c_2\}$, and so $\{p_1, b, c_1\}[pef]\{p_1, b, c_2\}$. Consequently, pef has concession in $C_{in} = \{p_1, b, c_1\}$, written as $pef \mathbf{con} \{p_1, b, c_1\}$, and pef is a firing sequence of M , written as $pef \in \text{FS}(M)$. The configurations $\{p_1, b, c_1\}$, $\{p_2, b, c_1\}$, $\{p_2, c_2\}$, and $\{p_1, b, c_2\}$ that are “encountered” during the firing of pef are thus reachable configurations of M , written as $\{p_1, b, c_1\}$, $\{p_2, b, c_1\}$, $\{p_2, c_2\}$, $\{p_1, b, c_2\} \in \mathcal{C}_M$. Since $c \mathbf{con} \{p_1, b, c_2\}$, all transitions of M are useful. In the comments following Example 7 we observe that all transitions of M are even live. The transitions of the EN system of Fig. 15 are all useful, but none is live: from the (reachable) configuration $\{p_6\}$ no other configuration can be reached, and so no transition can be fired anymore.

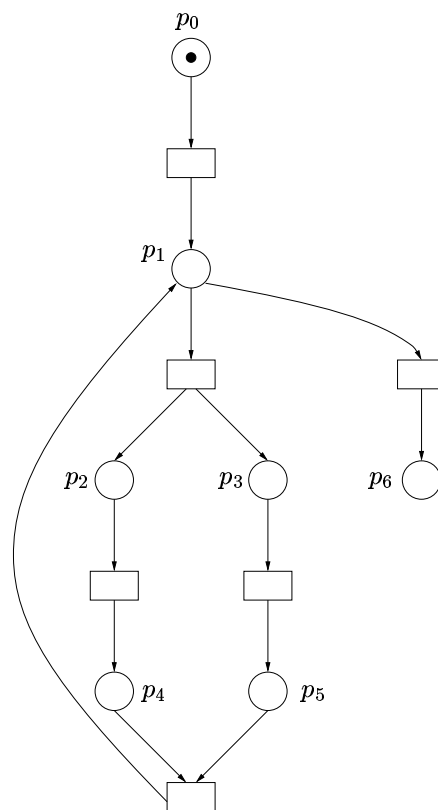


Fig. 15. An EN system of which all transitions are useful, but none is live.

When analyzing the (sequential) behaviour of an EN system M , its (sequential) configuration graph is often useful. It directly represents the way in which the set \mathcal{C}_M is constructed from C_{in} by firing transitions.

Before presenting its formal definition, we first recall the notion of a (directed) edge-labelled graph. Here we want such a graph to be “initialized”, i.e., to have an “initial node”. Also, we allow an edge to have a set of labels rather than a single label, because this will be needed when we consider concurrent configuration graphs. In the literature, initialized edge-labelled graphs are also known as *transition systems*.

Definition 9. An (initialized) *edge-labelled graph* is a quadruple $(V, \Gamma, \Sigma, v_{in})$, where V is a finite set of *nodes*, $v_{in} \in V$ is the *initial node*, Σ is a finite set of (*edge-*) *labels*, and $\Gamma \subseteq V \times \mathcal{P}(\Sigma) \times V$ is a set of (*labelled*) *edges*.

For an edge $e = (v, U, w) \in \Gamma$ of such a graph $G = (V, \Gamma, \Sigma, v_{in})$, $U \subseteq \Sigma$ is the *set of labels of e*. If U is a singleton, $U = \{\sigma\}$, then we also write the edge e as (v, σ, w) rather than $(v, \{\sigma\}, w)$.

Isomorphism of edge-labelled graphs is defined in the following way. Note that this notion of isomorphism also allows the labels to change.

Definition 10. Let $G_1 = (V_1, \Gamma_1, \Sigma_1, v_1)$ and $G_2 = (V_2, \Gamma_2, \Sigma_2, v_2)$ be two edge-labelled graphs. Then G_1 and G_2 are *isomorphic*, denoted by $G_1 \equiv G_2$, if there exist two bijections $\alpha : V_1 \rightarrow V_2$ and $\beta : \Sigma_1 \rightarrow \Sigma_2$ such that $\alpha(v_1) = v_2$ and, for all $v, w \in V_1$ and all $U \subseteq \Sigma_1$, $(v, U, w) \in \Gamma_1$ iff $(\alpha(v), \beta(U), \alpha(w)) \in \Gamma_2$.

Here, as usual, $\beta(U) = \{\beta(\sigma) \mid \sigma \in U\}$. If we want to be more specific, then we say that G_1 and G_2 are (α, β) -*isomorphic*, written as $G_1 \equiv_{\beta}^{\alpha} G_2$.

The (sequential) configuration graph of an EN system is now formally defined as follows; here each edge has exactly one label, i.e., $\Gamma \subseteq V \times \Sigma \times V$. In the literature it is often called the sequential *case graph*, or the *transition system* of the EN system.

Definition 11. Let M be an EN system. The *sequential configuration graph of M*, denoted by $\text{SCG}(M)$, is the edge-labelled graph $(V, \Gamma, \Sigma, v_{in})$, where $V = \mathbb{C}_M$, $v_{in} = (C_{in})_M$, $\Sigma = \mathbf{use}(T_M)$, and $\Gamma = \{(C, t, D) \mid C, D \in \mathbb{C}_M, t \in T_M, C\{t\}_M D\}$.

Example 7. For the EN system M from Example 4 (Fig. 12), $\text{SCG}(M)$ is given in Fig. 16. The “wiggly” arrow indicates its initial node (i.e., the initial configuration of M). To simplify notation, we use $a_1 a_2 \cdots a_n$ for the set $\{a_1, a_2, \dots, a_n\}$.

Note that the useful transitions of an EN system M are precisely the labels that actually occur on the edges of the sequential configuration graph of M . Also the liveness of a transition t of an EN system M can easily be decided by analyzing the sequential configuration graph of M : from each node of $\text{SCG}(M)$ there must exist a path to a node with an outgoing edge labelled by t . Thus, for the EN system M of Fig. 12, $\text{SCG}(M)$ of Fig. 16 shows that all transitions of M are live.

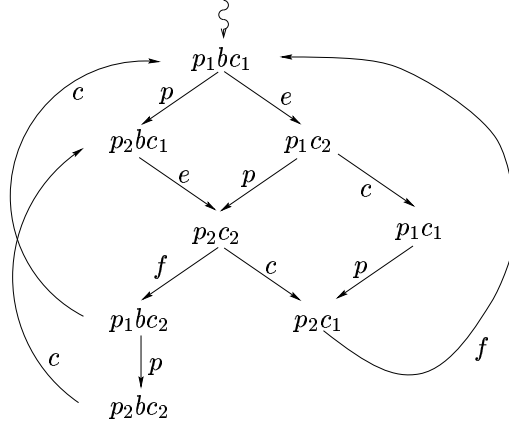


Fig. 16. A sequential configuration graph.

The class of transition systems corresponding to EN systems (i.e., the class of all sequential configuration graphs $\text{SCG}(M)$) is characterized and studied in [EhrRoz90, NieRozThi92a, NieRozThi92b, NieRozThi95].

The sequential configuration graph $\text{SCG}(M)$ can be considered as the (sequential) “state space” of the EN system M . Since M has only finitely many configurations, its state space is finite. Thus, an elementary net system M is a model of a finite-state concurrent system, and $\text{SCG}(M)$ models its sequential behaviour. Since finite automata model finite-state sequential systems, there is a clear relationship between elementary net systems and finite automata. In fact, to the reader familiar with formal language theory (see, e.g., [HopUll79]) it should be clear that the sequential configuration graph $\text{SCG}(M)$ can be seen as a finite automaton with initial state C_{in} (and all states being final states). A firing sequence of M is a word in T_M^* that forms the concatenation of the labels of a path in $\text{SCG}(M)$ that starts in C_{in} . Hence, the firing sequences of M are precisely the words that are accepted by the finite automaton $\text{SCG}(M)$. This gives the following result.

Theorem 12. *For every EN system M , $\text{FS}(M)$ is a regular language.*

Proof. Let $M = (P, T, F, C_{in})$. Consider the (deterministic) finite automaton \mathcal{A} with input alphabet $\mathbf{use}(T)$, set of states \mathbb{C}_M , initial state C_{in} , set of final states \mathbb{C}_M (thus each state is a final state), and set of state transitions $\{(C, t, D) \mid C[t]_M D\}$. Then $\text{FS}(M)$ is the language accepted by \mathcal{A} . \square

Note that the language $\text{FS}(M)$ is prefix-closed, i.e., if a word x is contained in $\text{FS}(M)$, then also each prefix of x is contained in $\text{FS}(M)$. This follows directly from the definition of a firing sequence (Definition 8) and explains why all states of the automaton \mathcal{A} are taken as final states in the above proof. Hence not

every regular language is the set of firing sequences of an EN system (e.g., the language $\{ab\}$, with $a, b \in \Sigma$, is regular but not prefix-closed). Also, not every prefix-closed regular language is the set of firing sequences of an EN system; e.g., $\{\lambda, t, tt\}$ with $t \in \Sigma$ is such a language, because no transition can fire twice consecutively (if $C[t]D$, then $\bullet t \cap D = \emptyset$, and hence t does not have concession in D).

4.4 Concurrency

Intuitively it is not so difficult to get an idea of a “run” of an EN system in which transitions can fire concurrently, i.e., independently of each other. In fact, in everyday life, people can be viewed as communicating concurrent components of a large system. However, to formalize this intuition is not so easy. We will now give a first attempt to capture the concurrent firing of transitions in a formal definition. The second, more sophisticated, attempt is presented in Section 6. In this first attempt, we still view the behaviour of an EN system in terms of global state transitions that are made in a step-wise fashion. However, as opposed to Definition 6(2), we now allow several transitions to fire in one such step.

Intuitively, two transitions such that each of them has concession in a given configuration, can be fired concurrently provided they are disjoint, i.e., have no common places. Also, it is intuitively clear how the configuration is transformed by firing both transitions. We will now formalize these intuitions; we do this right away for an arbitrary number of transitions (rather than two).

Definition 13. Let $M = (P, T, F, C_{in})$ be an EN system.

(1) Let $U \subseteq T$. U is a *disjoint set of transitions* if $U \neq \emptyset$ and for every two distinct transitions $t_1, t_2 \in U$: $\mathbf{nbh}(t_1) \cap \mathbf{nbh}(t_2) = \emptyset$; this is denoted by $\mathbf{disj}(U)$.

(2) Let $U \subseteq T$ and let $C \subseteq P$. Then U *has concession in C* (or U *can be fired in C* , or U *is enabled in C*) if $\mathbf{disj}(U)$, $\bullet U \subseteq C$, and $U \bullet \cap C = \emptyset$; this is denoted by $U \mathbf{con} C$.

(3) Let $U \subseteq T$ and let $C, D \subseteq P$. Then U *fires from C to D* , written as $C[U]D$, if $U \mathbf{con} C$ and $D = (C - \bullet U) \cup U \bullet$. If $\#U \geq 2$, then we also say that U is a *concurrent step from C to D* .

The firing of a disjoint set of transitions U can be seen as a global state transition of the system M , consisting of local state transitions (namely the transitions in U). Thus, states and state transitions are now treated in a similar way: global states (configurations) are sets of local states (places) and global state transitions (concurrent steps) are sets of local state transitions (sequential steps).

Analogous to Lemma 7 we obtain the following lemma.

Lemma 14. *Let $M = (P, T, F, C_{in})$ be an EN system. Let $U \subseteq T$ and let $C, D \subseteq P$. Then $C[U]D$ holds iff $\mathbf{disj}(U)$, $C - D = \bullet U$, and $D - C = U \bullet$.*

The amount of change caused by a set of transitions is thus cumulative, i.e., it equals the sum of the changes caused by the separate transitions. As in the case of single transitions this means that the amount of change caused by the occurrence of several disjoint events is determined only by these events themselves, not by the configuration in which they occur. However, in a T-simple EN system the principle of extensionality does not necessarily hold for concurrent steps. For example, it is easy to construct an EN system with three transitions t_1, t_2, t_3 for which $\bullet t_3 = \bullet\{t_1, t_2\}$ and $t_3^\bullet = \{t_1, t_2\}^\bullet$.

Example 8. Let M be once again the EN system of Example 4 (Fig. 12). Then $\{p, e\}$ is a disjoint set of transitions (thus $\text{disj}(\{p, e\})$), but $\{f, e\}$ is not. The only disjoint sets U with $\#U \geq 2$ are $\{p, e\}$, $\{p, c\}$ and $\{f, c\}$. The set $\{p, e\}$ has concession in C_{in} (thus $\{p, e\} \text{ con } \{p_1, b, c_1\}$), and $\{p_1, b, c_1\}\{p, e\}\{p_2, c_2\}$.

The concession of a set U of transitions (in a given configuration) can be expressed through the concessions of the transitions in U , together with a simplified disjointness condition, as follows.

Lemma 15. *Let $M = (P, T, F, C_{in})$ be an EN system. Let $C \subseteq P$ and let $U \subseteq T$ with $U \neq \emptyset$. Then $U \text{ con } C$ iff*

- (1) $t \text{ con } C$ for all $t \in U$, and
- (2) for all $t_1, t_2 \in U$ with $t_1 \neq t_2$, $\bullet t_1 \cap \bullet t_2 = \emptyset$ and $t_1^\bullet \cap t_2^\bullet = \emptyset$.

Proof. Obviously, $\bullet U \subseteq C$ and $U^\bullet \cap C = \emptyset$ if and only if $t \text{ con } C$ for all $t \in U$. Moreover, if $t_1 \text{ con } C$ and $t_2 \text{ con } C$, then $\bullet t_1 \cap \bullet t_2 = \emptyset$ and $t_1^\bullet \cap t_2^\bullet = \emptyset$. \square

For a concurrent step U from configuration C to configuration D , each splitting of U into nonempty sets U_1 and U_2 yields two (concurrent or sequential) steps U_1, U_2 which, when executed sequentially in arbitrary order (first U_1 then U_2 , or first U_2 then U_1), lead to the same configuration, i.e., fire from C to D . This is formally expressed as follows.

Lemma 16. *Let $M = (P, T, F, C_{in})$ be an EN system, let $C, D \subseteq P$, and let $U \subseteq T$. Let $\{U_1, U_2\}$ be a partition of U , i.e., $U = U_1 \cup U_2$, $U_1 \cap U_2 = \emptyset$ and $U_1, U_2 \neq \emptyset$. If $C[U]D$, then there exists a configuration $E \subseteq P$ such that $C[U_1]E$ and $E[U_2]D$.*

Proof. It is intuitively clear that E is the configuration that is obtained by firing U_1 , i.e., $E = (C - \bullet U_1) \cup U_1^\bullet$. The formal proof is as follows.

To begin with, $\text{disj}(U_1)$ and $\text{disj}(U_2)$ follow from $\text{disj}(U)$. From Lemma 14 it follows that $C \cap D$, $\bullet U$, and U^\bullet are mutually disjoint sets with $C = (C \cap D) \cup \bullet U$ and $D = (C \cap D) \cup U^\bullet$. From $\text{disj}(U)$ and the fact that $\{U_1, U_2\}$ is a partition of U it then follows that $C \cap D$, $\bullet U_1$, $\bullet U_2$, U_1^\bullet , and U_2^\bullet are mutually disjoint sets, with $\bullet U = \bullet U_1 \cup \bullet U_2$ and $U^\bullet = U_1^\bullet \cup U_2^\bullet$. Hence $C = (C \cap D) \cup \bullet U_1 \cup \bullet U_2$ and $D = (C \cap D) \cup U_1^\bullet \cup U_2^\bullet$. Now consider the configuration $E = (C \cap D) \cup U_1^\bullet \cup \bullet U_2$ (i.e., $E = (C - \bullet U_1) \cup U_1^\bullet$). Then $C - E = \bullet U_1$ and $E - C = U_1^\bullet$, and thus $C[U_1]E$ according to Lemma 14. Likewise $E - D = \bullet U_2$ and $D - E = U_2^\bullet$, and thus $E[U_2]D$ according to Lemma 14. \square

This lemma expresses a so-called *diamond property*: if $C[U]D$ and $\{U_1, U_2\}$ is a partition of U , then there exist two configurations E_1 and E_2 such that $C[U_1]E_1[U_2]D$ and $C[U_2]E_2[U_1]D$. A drawing (in Fig. 17) of the four steps $C[U_1]E_1$, $E_1[U_2]D$, $C[U_2]E_2$, and $E_2[U_1]D$ gives a “diamond”; the step $C[U]D$ is then a diagonal of this diamond. More general diamond properties are studied in [HooRoz91].

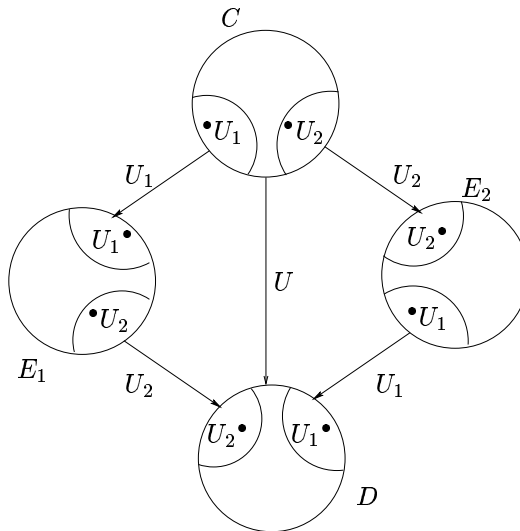


Fig. 17. A diamond.

It follows directly from Lemma 16 that a concurrent step can be realized by the firing of its elements in an arbitrary order. Such a realization of a concurrent step intuitively corresponds to a possible way in which a sequential observer sees the step as a sequence of sequential steps. This is expressed by the following lemma.

Lemma 17. *Let $M = (P, T, F, C_{in})$ be an EN system, let $C, D \subseteq P$ and let $U \subseteq T$. If $C[U]D$, then $C[t_1 \cdots t_n]D$ for each ordering (t_1, \dots, t_n) of the elements of U .*

Proof. By induction on $\#U$. The induction step follows directly from Lemma 16. \square

This thus means that by allowing concurrent steps no new reachable configurations are obtained (and no new useful transitions). Adding all steps (C, U, D) with $\#U \geq 2$ to the sequential configuration graph leads only to new edges labelled by sets of transitions.

Definition 18. Let M be an EN system. The *configuration graph* of M , denoted by $\text{CG}(M)$, is the edge-labelled graph $(V, \Gamma, \Sigma, v_{in})$, where $V = \mathbb{C}_M$, $v_{in} = (C_{in})_M$, $\Sigma = \mathbf{use}(T_M)$, and $\Gamma = \{(C, U, D) \mid C, D \in \mathbb{C}_M, U \subseteq T_M, C[U]_M D\}$.

Note that if $C[U]_M D$ and $C \in \mathbb{C}_M$, then $U \subseteq \mathbf{use}(T_M)$, according to Lemma 15.

Example 9. Let M be the EN system from Example 4 (Fig. 12). The configuration graph $\text{CG}(M)$ is drawn in Fig. 18. Compare $\text{CG}(M)$ with the sequential configuration graph $\text{SCG}(M)$ of Fig. 16. Also note that $\text{CG}(M)$ contains several “diamonds” (see Fig. 17) such as, e.g., the diamond at the concurrent step $\{p_1, b, c_1\} \{p, e\} \{p_2, c_2\}$.

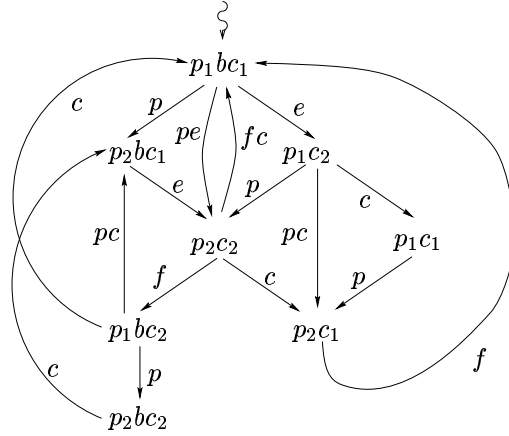


Fig. 18. A configuration graph.

Rather surprisingly, Lemma 17 also holds the other way around. To prove this (in Theorem 20) we use the following diamond property, which shows that a diamond exists whenever three of its sides are given, cf. [HooRoz91].

Lemma 19. Let $M = (P, T, F, C_{in})$ be an EN system, let $C \subseteq P$ and let $s, t \in T$. If $st \mathbf{con} C$ and $t \mathbf{con} C$, then $\{s, t\} \mathbf{con} C$.

Proof. Since $st \mathbf{con} C$, $s \mathbf{con} C$ certainly holds. Since $t \mathbf{con} C$ also holds, it suffices according to Lemma 15 to show that $\bullet s \cap \bullet t = \emptyset$ and $s \bullet \cap t \bullet = \emptyset$. Let $C[s]D$. Then $\bullet s \cap D = \emptyset$ and $s \bullet \subseteq D$. Since $t \mathbf{con} D$, $\bullet t \subseteq D$ and $t \bullet \cap D = \emptyset$. Hence $\bullet s \cap \bullet t = \emptyset$ and $s \bullet \cap t \bullet = \emptyset$. \square

Now we prove the so-called *sequentialization property*.

Theorem 20. *Let $M = (P, T, F, C_{in})$ be an EN system, let $C, D \subseteq P$ and let $U \subseteq T$ with $U \neq \emptyset$. Then*

- (1) $U \mathbf{con} C$ iff $t_1 \cdots t_n \mathbf{con} C$ for every ordering (t_1, \dots, t_n) of the elements of U , and
- (2) $C[U]D$ iff $C[t_1 \cdots t_n]D$ for every ordering (t_1, \dots, t_n) of the elements of U .

Proof. (Only-if) The only-if-part of (2) is Lemma 17. From this the only-if-part of (1) directly follows.

(If) We first prove the if-part of (1). For each $t \in U$ there exists an ordering (t_1, \dots, t_n) of U with $t_1 = t$. This implies that $t \mathbf{con} C$ for all $t \in U$. What remains to be proved (see Lemma 15), is $\mathbf{disj}(\{s, t\})$ for every two distinct elements s and t of U . To this aim, we use Lemma 19. We already know that $t \mathbf{con} C$. If we now consider an ordering of U of the form (s, t, t_3, \dots, t_n) , then, by assumption, $stt_3 \cdots t_n \mathbf{con} C$. Hence $st \mathbf{con} C$, and so, by Lemma 19, $\{s, t\} \mathbf{con} C$. Thus $\mathbf{disj}(\{s, t\})$.

The if-part of (2) can now easily be deduced from the if-part of (1) and the only-if-part of (2). \square

According to this theorem, the concession and the effect of a concurrent step are completely determined by the concession and the effect of sequences of sequential steps. Hence we can use the properties of firing sequences of transitions when reasoning about concurrent steps. In particular, it follows from Theorem 20 that we can construct the configuration graph $\mathbf{CG}(M)$ of an EN system M from its sequential configuration graph $\mathbf{SCG}(M)$, even if we do not know M itself (see Section 4 of [HooRoz91]). Thus, the sequential configuration graph contains already all the information about concurrency! This is formally expressed in the next theorem.

Theorem 21. *For EN systems M and M' ,*
 $\mathbf{SCG}(M) \equiv \mathbf{SCG}(M') \iff \mathbf{CG}(M) \equiv \mathbf{CG}(M')$.

Proof. (If) It follows directly from the definitions that if $\mathbf{CG}(M) \equiv_{\beta}^{\alpha} \mathbf{CG}(M')$, then $\mathbf{SCG}(M) \equiv_{\beta}^{\alpha} \mathbf{SCG}(M')$.

(Only-if) Assume that $\mathbf{SCG}(M) \equiv_{\beta}^{\alpha} \mathbf{SCG}(M')$. This means that for all $C, D \in \mathbb{C}_M$ and $t \in \mathbf{use}(T_M)$, $C[t]_M D$ iff $\alpha(C)[\beta(t)]_{M'} \alpha(D)$. It is now easy to prove, by induction on $|x|$, that for all $C, D \in \mathbb{C}_M$ and $x \in \mathbf{use}(T_M)^*$, $C[x]_M D$ iff $\alpha(C)[\beta(x)]_{M'} \alpha(D)$; note that, for $x = t_1 \cdots t_n$, $\beta(x) = \beta(t_1) \cdots \beta(t_n)$ according to Section 3. From this and Theorem 20 it follows that for all $C, D \in \mathbb{C}_M$ and $U \subseteq \mathbf{use}(T_M)$, $C[U]_M D$ iff $\alpha(C)[\beta(U)]_{M'} \alpha(D)$. Thus $\mathbf{CG}(M) \equiv_{\beta}^{\alpha} \mathbf{CG}(M')$. \square

It is shown in [HooRoz91] that this result (and in particular its If direction) is still true if the isomorphism between the configuration graphs disregards the fact that the edges are labelled by sets, viewing them as abstract symbols (i.e., if in Definition 10, β is taken as a partial injective function from $\mathcal{P}(\Sigma_1)$ to $\mathcal{P}(\Sigma_2)$).

4.5 Fundamental Situations

An attractive feature of Petri Net models is that important notions concerning concurrent systems can be formulated in terms of, e.g., EN systems in a very natural way. To illustrate this aspect of EN systems, we discuss several fundamental situations that may occur in the dynamic behaviour of (concurrent systems that can be modelled by) EN systems. In what follows, we assume that an EN system $M = (P, T, F, C_{in})$ is given.

There are three fundamental relationships that may hold between two events t_1 and t_2 in a given configuration C : causality, concurrency, and conflict.

(1) Causality (of events t_1 and t_2 in configuration C).

This notion is illustrated in Figs. 19 and 20: $t_2 \mathbf{con} C$ does not hold, but $t_1 t_2 \mathbf{con} C$ does hold. Thus, t_1 needs to occur to grant concession to t_2 (input-concession in Fig. 19, and output-concession in Fig. 20). In other words, t_1 is one of the causes of t_2 . An equivalent formal definition is: $t_1 t_2 \mathbf{con} C$ holds, but $t_1^\bullet \cap \bullet t_2 \neq \emptyset$ or ${}^\bullet t_1 \cap t_2^\bullet \neq \emptyset$ (verify this).

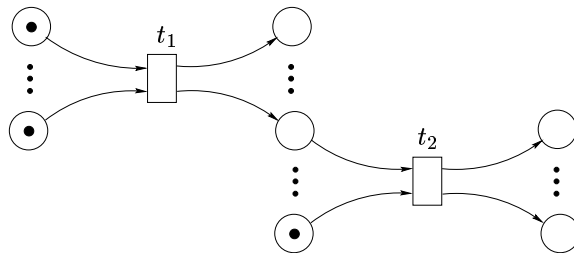


Fig. 19. Causality: $t_1^\bullet \cap \bullet t_2 \neq \emptyset$.

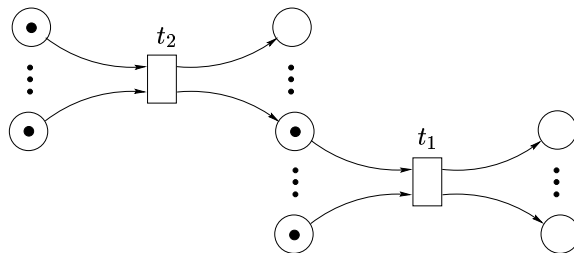


Fig. 20. Causality: ${}^\bullet t_1 \cap t_2^\bullet \neq \emptyset$.

(2) **Concurrency** (of events t_1 and t_2 in configuration C).

This notion is illustrated in Fig. 21: $\{t_1, t_2\}$ **con** C . Figure 22 shows a more complete picture of the situation: it gives a representation of C and $P - C$, and it shows how $\bullet t_1, \bullet t_2$ and $t_1\bullet, t_2\bullet$ fit into C and $P - C$, respectively.

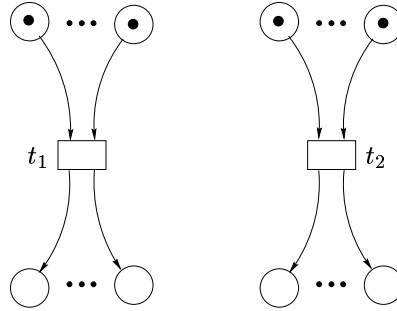


Fig. 21. Concurrency.

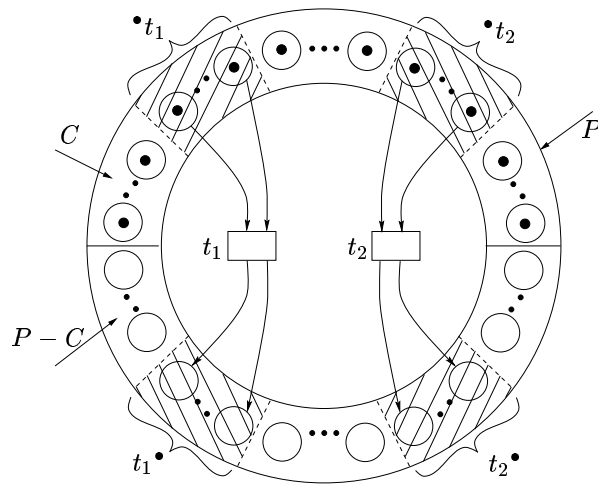


Fig. 22. Concurrency, the complete picture.

Hence if $t_1 t_2$ has concession in C , then t_1 and t_2 are related by either causality

or concurrency: if $t_2 \text{ con } C$ does not hold, then there is causality, and if $t_2 \text{ con } C$ does hold, then (by Lemma 19) there is concurrency.

(3) Conflict (between events t_1 and t_2 in configuration C).

This notion is illustrated in Figs. 23 and 24: both $t_1 \text{ con } C$ and $t_2 \text{ con } C$ hold, but $\{t_1, t_2\} \text{ con } C$ does not hold. Consequently, by Lemma 15, $\bullet t_1 \cap \bullet t_2 \neq \emptyset$ or $t_1^\bullet \cap t_2^\bullet \neq \emptyset$. If the former holds, then we have an input-conflict (represented in Fig. 23), and if the latter holds, then we have an output-conflict (represented in Fig. 24). Obviously, input-conflict and output-conflict can also both be present.

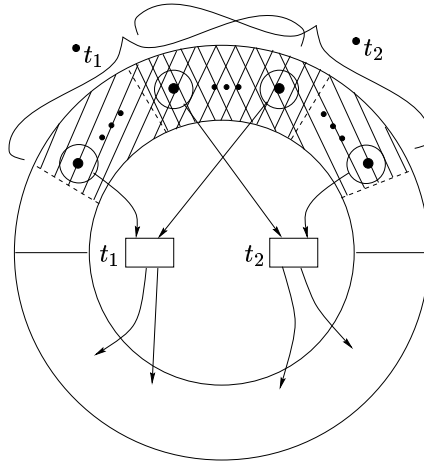


Fig. 23. Input-conflict.

Hence if t_1 and t_2 both have concession in configuration C , then there is either conflict or concurrency. In the case of concurrency t_1 and t_2 are independent, whereas in the case of a conflict they are not independent. That is why a conflict intuitively leads to a nondeterministic choice between the transitions (either t_1 occurs, or t_2 occurs). Clearly there is no need for choice if t_1 and t_2 are concurrent. EN systems in which no choice is made are particularly easy to understand; they are the concurrent equivalent of deterministic finite automata in the sequential case.

Definition 22. An EN system $M = (P, T, F, C_{in})$ is *conflict-free* if for every $C \in \mathbb{C}_M$ and all transitions $t_1, t_2 \in T$: if $t_1 \text{ con } C$ and $t_2 \text{ con } C$, then $\{t_1, t_2\} \text{ con } C$.

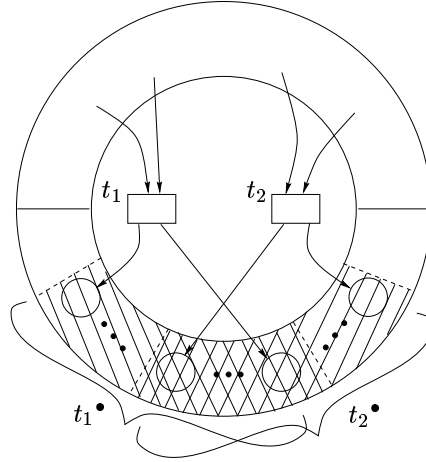


Fig. 24. Output-conflict.

Since in a conflict-free EN system choices are never made, it has only one run. The EN system of Fig. 4 is conflict-free, for “structural reasons”: both p^\bullet and ${}^\bullet p$ contain at most one transition, for every $p \in P$; such systems are clearly conflict-free: $\{t_1, t_2\} \text{ con } C$ follows from ${}^\bullet t_1 \cap {}^\bullet t_2 = \emptyset$ and $t_1^\bullet \cap t_2^\bullet = \emptyset$, by Lemma 15. In the literature (see, e.g., [LanRob78]) conflict-free systems are usually called *persistent*, in which case the term ‘conflict-free’ is reserved for a structural subclass such as the one above.

The interplay between concurrency and conflict may be quite intricate, and in particular it can lead to confusion - a phenomenon that seems to be fundamentally present in nature and appears in various disguises, depending on the chosen level and way of description of a concurrent system. Here we discuss confusion in the framework of EN systems.

(4) Confusion.

Consider the EN system in Fig. 25, hence $C_{in} = \{p_1, p_2, p_3\}$. Let $C = \{p_4, p_5\}$; thus $C_{in}[\{t_1, t_2\}]C$. Different sequential realizations $(t_1 t_2$ and $t_2 t_1)$ of this concurrent step have drastically different properties. Since sequential realizations of a concurrent step correspond to observations of the step by sequential observers, assume that we have two honest sequential observers O_1 (corresponding to $t_1 t_2$) and O_2 (corresponding to $t_2 t_1$). They will report their observations as follows:

O_1 : “ t_1 occurred first without having been in conflict with another event; then t_2 occurred”, and

O_2 : “ t_2 occurred first; this resulted in a conflict between t_1 and t_3 which was resolved in favour of t_1 , and so t_1 occurred”.

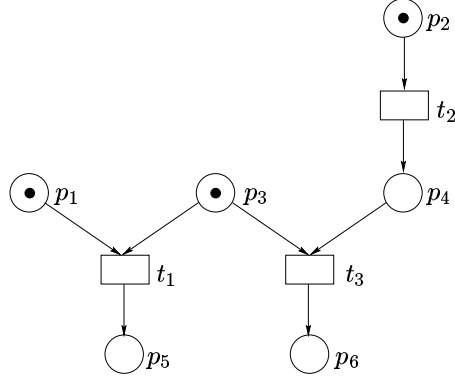


Fig. 25. A conflict-increasing confusion.

This is therefore a confusing situation, which resulted from the interplay between concurrency (between t_1 and t_2) and conflict (between t_1 and t_3). Systems where confusion occurs are in general difficult to analyze. This is due to the fact that the intermediate configurations determined by the different sequential realizations of a concurrent step can differ drastically from each other, as we have seen in the example above. Consequently, in general one may have to analyze all possible sequentializations of the step rather than just one. The theory of Petri Nets suggests that it is not the combination of concurrency and conflict as such that causes difficulties. Only those combinations of concurrency and conflict that result in confusion create problems. Unfortunately it is not always possible to avoid confusion. Even the rather simple “mutual exclusion problem” discussed in Section 4.1 contains confusion (in configuration $\{w_1, p, r_2\}$).

We now turn to a formal discussion of confusion.

Definition 23. Let $M = (P, T, F, C_{in})$ be an EN system, let $C \in \mathbb{C}_M$, and let $t \in T$ be such that $t \text{ con } C$. The *conflict set of t in C* , denoted by $\mathbf{cfl}(t, C)$, is the set $\{t' \in T \mid t' \text{ con } C \text{ and } \neg \{t, t'\} \text{ con } C\}$.

Hence the conflict set of an event t in a configuration C is the set of all events that are in conflict with t in C . Note that t is in conflict in C only if t itself has concession in C .

Definition 24. Let $M = (P, T, F, C_{in})$ be an EN system, let $C \in \mathbb{C}_M$, and let $t_1, t_2 \in T$. The triple (C, t_1, t_2) is called a *confusion (in C)* if $t_1 \neq t_2$, $\{t_1, t_2\} \text{ con } C$, and $\mathbf{cfl}(t_1, C) \neq \mathbf{cfl}(t_1, D)$, where $C[t_2]D$. Then M is *confused in C* if there is a confusion in C .

Hence a triple (C, t_1, t_2) is a confusion if $\{t_1, t_2\}$ is a step in C and the occurrence of t_2 in C changes the conflict set of t_1 .

Example 10. Consider the EN system M depicted in Fig. 25. For the configuration $C = \{p_1, p_2, p_3\} = C_{in}$, $\mathbf{cfl}(t_1, C) = \emptyset$. Hence (C, t_1, t_2) is a confusion, because $\{t_1, t_2\} \mathbf{con} C$ and $\mathbf{cfl}(t_1, C) = \emptyset \neq \{t_3\} = \mathbf{cfl}(t_1, D)$, where $D = \{p_1, p_3, p_4\}$.

It is natural to distinguish the following two types of confusion.

Definition 25. Let $M = (P, T, F, C_{in})$ be an EN system, let $C \in \mathbb{C}_M$, let $t_1, t_2 \in T$, let $\gamma = (C, t_1, t_2)$ be a confusion, and let $C[t_2]D$.

(1) γ is a *conflict-increasing confusion*, *ci confusion* for short, if $\mathbf{cfl}(t_1, D) \supsetneq \mathbf{cfl}(t_1, C)$.

(2) γ is a *conflict-decreasing confusion*, *cd confusion* for short, if $\mathbf{cfl}(t_1, D) \subsetneq \mathbf{cfl}(t_1, C)$.

Example 11. (1) Consider the EN system M and the confusion (C, t_1, t_2) from Example 10 (Fig. 25). Since $\mathbf{cfl}(t_1, D) \supsetneq \mathbf{cfl}(t_1, C)$, (C, t_1, t_2) is a ci confusion.

(2) Consider the EN system in Fig. 26. For $C = C_{in} = \{p_1, p_2\}$, (C, t_1, t_2) is a confusion because $\{t_1, t_2\} \mathbf{con} C$ and $\mathbf{cfl}(t_1, C) = \{t_3\} \neq \emptyset = \mathbf{cfl}(t_1, D)$, where $D = \{p_1, p_3\}$. Furthermore, since $\mathbf{cfl}(t_1, D) \subsetneq \mathbf{cfl}(t_1, C)$, (C, t_1, t_2) is a cd confusion.

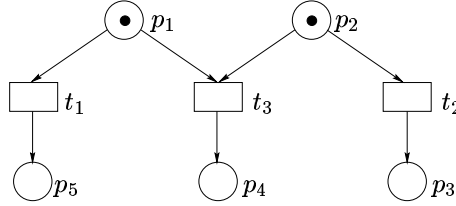


Fig. 26. A conflict-decreasing confusion.

(3) Consider the EN system in Fig. 27. For $C = C_{in} = \{p_1, p_2, p_4\}$, (C, t_1, t_2) is a confusion because $\mathbf{cfl}(t_1, C) = \{t_3\} \neq \{t_4\} = \mathbf{cfl}(t_1, D)$, where $D = \{p_1, p_3, p_4\}$. Note that (C, t_1, t_2) is neither a ci confusion nor a cd confusion.

As we have seen in the above example, the classification of confusions into ci confusions and cd confusions is not exhaustive: there exist confusions that are neither ci nor cd.

If (C, t_1, t_2) is a confusion, then the occurrence of t_2 in C has a rather strong impact on the occurrence of t_1 in C : it changes the conflict set of t_1 . Hence the fact that (C, t_1, t_2) is a confusion shows an influence of t_2 on t_1 (in C). In order to better understand the mutual dependency between t_1 and t_2 it is important to know whether t_1 has the same sort of influence on t_2 (in C). This consideration leads to the notion of symmetric confusion.

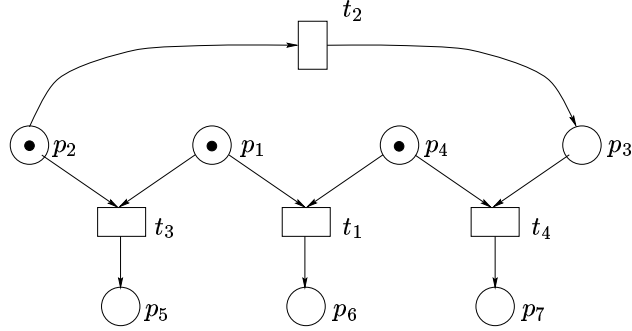


Fig. 27. A confusion that is neither conflict-increasing nor conflict-decreasing.

Definition 26. Let $M = (P, T, F, C_{in})$ be an EN system, let $C \in \mathbb{C}_M$, let $t_1, t_2 \in T$ and let $\gamma = (C, t_1, t_2)$ be a confusion. Then γ is *symmetric* if (C, t_2, t_1) is also a confusion, otherwise γ is *asymmetric*.

Example 12. (1) Consider the EN system M and the confusion (C, t_1, t_2) from Example 10 (Fig. 25). Since (C, t_2, t_1) is not a confusion, (C, t_1, t_2) is a ci confusion that is asymmetric (see also Example 11(1)).

(2) Consider the EN system in Fig. 28. For $C = C_{in} = \{p_1, p_3\}$, (C, t_1, t_2) is a ci confusion and (C, t_2, t_1) is a ci confusion. Hence (C, t_1, t_2) is a ci confusion that is symmetric.

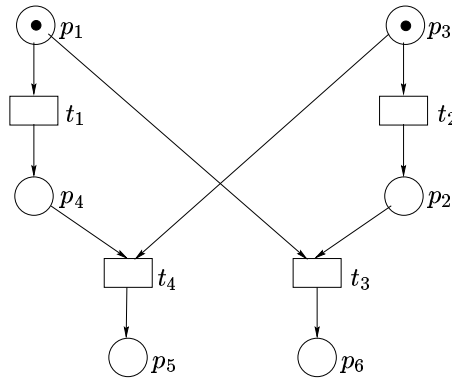


Fig. 28. A symmetric confusion.

(3) The confusion (C, t_1, t_2) from Example 11(2) (Fig. 26) is a symmetric cd

confusion.

(4) The confusion (C, t_1, t_2) from Example 11(3) (Fig. 27) is a symmetric confusion that is neither a ci confusion nor a cd confusion.

The above example shows that the division into ci and cd confusions is rather independent from the division into symmetric and asymmetric confusions. We will not consider the topic of confusion in more detail here, however we would like to mention that the only nontrivial relation between these dividing lines is that cd confusions are always symmetric (in fact, if (C, t_1, t_2) is a cd confusion, then there is a transition t such that $t \mathbf{con} C$ and t is in conflict with both t_1 and t_2).

A structural restriction on EN systems that guarantees the absence of confusion will be discussed in Section 9.7.

5 Equivalences and Normal Forms

When we wish to express in a formal, mathematical, way that two systems “are similar to each other” or that they “behave in the same way”, then we have to define an equivalence relation on the class of all systems, such that the systems that “are similar to each other” or “behave in the same way” form an equivalence class. Such equivalence relations are particularly useful when one wants to transform or optimize a system without changing its “behaviour”, i.e., transform a system into a “better”, equivalent system. If R is a property of EN systems such that for each EN system there is an equivalent one satisfying R , then we say that R is a “normal form” for the class of EN systems. In this section we consider several notions of equivalence and normal forms for EN systems. In particular, we formalize the notion of a component of an EN system and show that every EN system is equivalent with one that can be viewed as consisting of communicating concurrent components (where each of the components is sequential). Such decompositions of Petri nets are studied, e.g., in [Hac72, DesEsp95]. A survey of various notions of equivalence for EN systems is presented in [PomRozSim92].

5.1 Equivalence

The simplest (and least interesting) definition of equivalence is isomorphism. Two EN systems are isomorphic if their underlying nets are isomorphic in such a way that the initial configurations correspond to each other.

Definition 27. Two EN systems $M = (P, T, F, C_{in})$ and $M' = (P', T', F', C'_{in})$ are *isomorphic*, denoted by $M \equiv M'$, if there exist two bijections $\alpha : P \rightarrow P'$ and $\beta : T \rightarrow T'$ such that $\mathbf{und}(M) \equiv_{\beta}^{\alpha} \mathbf{und}(M')$ and $\alpha(C_{in}) = C'_{in}$.

Thus, isomorphic EN systems have the same static structure. Now we will try to capture the dynamic “behaviour” of EN systems through an equivalence relation, as discussed above. For EN systems there are many possibilities to define such a notion of equivalence, some weaker than others. If two systems are

isomorphic, then they will be equivalent for any of these notions. The underlying idea for all the notions of equivalence that we will consider, is that “behaviour” is mainly concerned with the actions (transitions) that occur during a run of the system and not so much with the distribution of the global state of the system (over the places).

Our first notion of dynamic equivalence defines two EN systems to be equivalent if there exist one-to-one correspondences between the transitions and the configurations (not the places!), such that the correspondence between configurations is preserved by firing corresponding transitions. This means that the systems “simulate each other’s behaviour”.

Definition 28. Let $M = (P, T, F, C_{in})$ and $M' = (P', T', F', C'_{in})$ be two EN systems. Then M and M' are *configuration equivalent*, denoted by $M \approx M'$, if there exist two bijections $\alpha : \mathbb{C}_M \rightarrow \mathbb{C}_{M'}$ and $\beta : \mathbf{use}_M(T) \rightarrow \mathbf{use}_{M'}(T')$ such that

- (1) $\alpha(C_{in}) = C'_{in}$ and
- (2) for all $C, D \in \mathbb{C}_M$ and $t \in \mathbf{use}_M(T)$, $C[t]_M D$ iff $\alpha(C)[\beta(t)]_{M'} \alpha(D)$.

If we want to be more specific, then we say that M and M' are (α, β) -*configuration equivalent*, denoted by $M \approx_{\beta}^{\alpha} M'$.

It can directly be seen that \approx is indeed an equivalence relation on the class of EN systems. In fact, configuration equivalence is strongly related to the configuration graphs. It immediately follows from the definitions that two EN systems are configuration equivalent iff their sequential configuration graphs are isomorphic. Hence here the behaviour of an EN system is identified with (represented by) its sequential configuration graph, modulo isomorphism. According to Theorem 21 this notion of behaviour covers also the concurrent behaviour of the system.

Theorem 29. *Let M and M' be two EN systems. Then $M \approx M'$ iff $SCG(M) \equiv SCG(M')$ iff $CG(M) \equiv CG(M')$.*

It is easy to see that $M \equiv M'$ implies $M \approx M'$, i.e., isomorphic EN systems are configuration equivalent. On the other hand there exist configuration equivalent EN systems that are not isomorphic (see the next example). Configuration equivalence is thus weaker than isomorphism of EN systems.

Example 13. Let M be the EN system of Fig. 29 and M' the EN system of Fig. 30. M and M' are not isomorphic. It is clear that $\mathbb{C}_M = \{\{p_1\}, \{p_3, p_4\}, \{p_2, p_4\}, \{p_5\}, \{p_6\}\}$ and $\mathbb{C}_{M'} = \{\{p_1\}, \{p_2\}, \{p_3\}, \{p_4\}, \{p_5\}\}$. All transitions are useful. Let $\alpha : \mathbb{C}_M \rightarrow \mathbb{C}_{M'}$ and $\beta : T_M \rightarrow T_{M'}$ be the bijections defined as follows: $\alpha(\{p_1\}) = \{p_1\}$, $\alpha(\{p_3, p_4\}) = \{p_2\}$, $\alpha(\{p_2, p_4\}) = \{p_3\}$, $\alpha(\{p_5\}) = \{p_4\}$, $\alpha(\{p_6\}) = \{p_5\}$, and $\beta(t_i) = t_i$ for all $1 \leq i \leq 6$. Now it should be clear that $M \approx_{\beta}^{\alpha} M'$. The sequential configuration graphs $SCG(M)$ and $SCG(M')$ are given in Figs. 31 and 32. It is easy to see that $SCG(M) \equiv_{\beta}^{\alpha} SCG(M')$. Note that for both M and M' the configuration graph is the same as the sequential configuration graph.

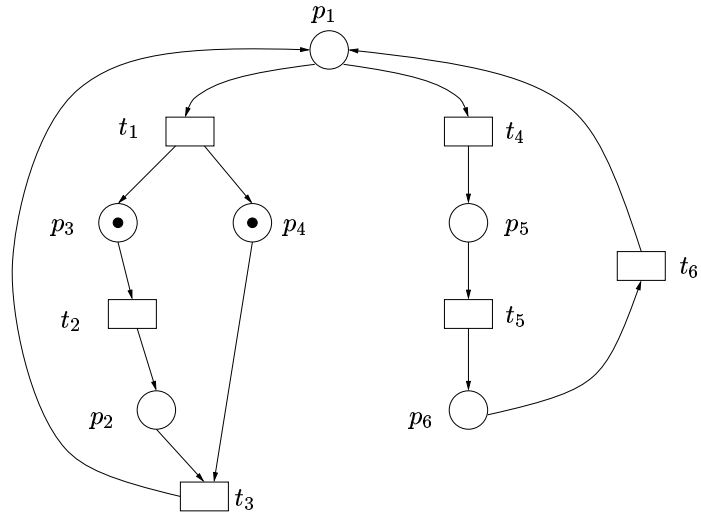


Fig. 29. An EN system M .

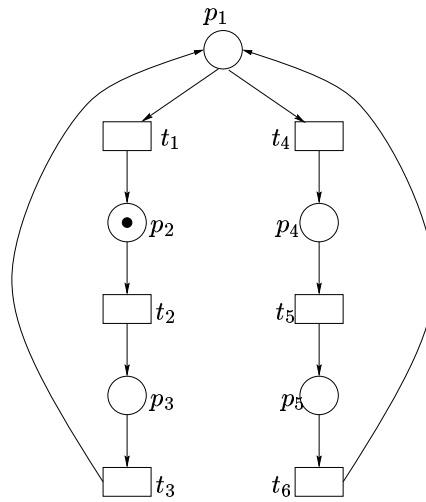


Fig. 30. An EN system M' configuration equivalent with the EN system M of Fig. 29.

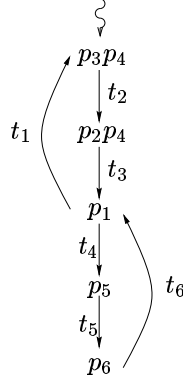


Fig. 31. The configuration graph of the EN system M of Fig. 29.

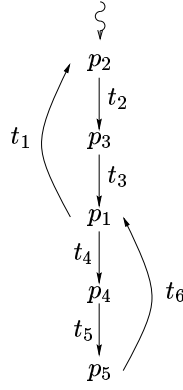


Fig. 32. The configuration graph of the EN system M' of Fig. 30.

The following technical lemma is often useful in proofs of configuration equivalence.

Lemma 30. *Let $M = (P, T, F, C_{in})$ and $M' = (P', T', F', C'_{in})$ be two EN systems. If α is an injective function, $\alpha : \mathbb{C}_M \rightarrow \mathcal{P}(P')$, and β is a bijective function, $\beta : \mathbf{use}_M(T) \rightarrow T'$, such that*

- (1) $\alpha(C_{in}) = C'_{in}$ and
 - (2) for all $C, D \in \mathbb{C}_M$ and $t \in \mathbf{use}_M(T)$,
 $C[t]_M D$ implies $\alpha(C)[\beta(t)]_{M'} \alpha(D)$, and
 $\beta(t) \mathbf{con}_{M'} \alpha(C)$ implies $t \mathbf{con}_M C$,
- then $M \approx_{\beta}^{\alpha} M'$.

Proof. We first prove the if-part of Definition 28(2). Let $\alpha(C)[\beta(t)]_{M'}\alpha(D)$. Then $\beta(t) \mathbf{con}_{M'} \alpha(C)$ and thus $t \mathbf{con}_M C$. Let $C[t]_M E$. Then $\alpha(C)[\beta(t)]_{M'}\alpha(E)$. Hence $\alpha(E) = \alpha(D)$. Since α is injective, $E = D$ and thus $C[t]_M D$.

* To prove that α is a bijection between \mathbb{C}_M and $\mathbb{C}_{M'}$ we first show that $\alpha(C) \in \mathbb{C}_{M'}$ for every $C \in \mathbb{C}_M$. This is done by induction on C . For $C = C_{in}$, $\alpha(C_{in}) = C'_{in} \in \mathbb{C}_{M'}$ by (1). Now assume (as induction hypothesis) that $\alpha(C) \in \mathbb{C}_{M'}$ and let $C[t]_M D$. We have to prove that $\alpha(D) \in \mathbb{C}_{M'}$. From $C[t]_M D$ and (2) it follows that $\alpha(C)[\beta(t)]_{M'}\alpha(D)$. Since $\alpha(C) \in \mathbb{C}_{M'}$, we obtain that $\alpha(D) \in \mathbb{C}_{M'}$. Next we show that α is surjective, i.e., that for every $C' \in \mathbb{C}_{M'}$ there exists $C \in \mathbb{C}_M$ with $\alpha(C) = C'$. This is done in the same way, by induction on C' . For $C' = C'_{in}$, according to (1), we can take $C = C_{in}$. Now assume that $\alpha(C) = C'$ for $C \in \mathbb{C}_M$ and let $C'[t']_{M'} D'$. Since β is a bijection, there is a $t \in \mathbf{use}(T)$ with $\beta(t) = t'$. Hence $\beta(t) \mathbf{con} \alpha(C)$ and thus $t \mathbf{con} C$ according to (2). Let $C[t]_M D$. Hence $D \in \mathbb{C}_M$. Then $\alpha(C)[\beta(t)]_{M'}\alpha(D)$; in other words $C'[t']_{M'}\alpha(D)$, and thus $\alpha(D) = D'$.

We still have to prove that β is a bijection from $\mathbf{use}_M(T)$ to $\mathbf{use}_{M'}(T')$, i.e., that $\beta(t) \in \mathbf{use}_{M'}(T')$ for every $t \in \mathbf{use}_M(T)$. If $t \in \mathbf{use}_M(T)$, then there exist $C, D \in \mathbb{C}_M$ such that $C[t]_M D$. Hence $\alpha(C)[\beta(t)]_{M'}\alpha(D)$. Since $\alpha(C) \in \mathbb{C}_{M'}$, $\beta(t) \in \mathbf{use}_{M'}(T')$. \square

A similar, but less stringent, definition of equivalence is obtained by no longer requiring the correspondence α between configurations to be a bijection, but just a relation. In that way the systems can still simulate each other's behaviour "step by step", but the configuration graphs no longer need to be isomorphic. Again, the correspondence between the transitions is a bijection because we are mainly interested in the transitions (modulo their identity) of a system.

Definition 31. Let $M = (P, T, F, C_{in})$ and $M' = (P', T', F', C'_{in})$ be two EN systems. M and M' are *weakly configuration equivalent*, denoted by $M \approx_w M'$, if there exists a relation $\alpha \subseteq \mathbb{C}_M \times \mathbb{C}_{M'}$ and a bijection $\beta : \mathbf{use}(T) \rightarrow \mathbf{use}(T')$, such that

- (1) $(C_{in}, C'_{in}) \in \alpha$,
- (2) for all $C, D \in \mathbb{C}_M$, $C' \in \mathbb{C}_{M'}$, and $t \in \mathbf{use}(T)$: if $C[t]_M D$ and $(C, C') \in \alpha$, then there is a $D' \in \mathbb{C}_{M'}$ such that $C'[\beta(t)]_{M'} D'$ and $(D, D') \in \alpha$, and
- (3) for all $C', D' \in \mathbb{C}_{M'}$, $C \in \mathbb{C}_M$, and $t' \in \mathbf{use}(T')$: if $C'[t']_{M'} D'$ and $(C, C') \in \alpha$, then there is a $D \in \mathbb{C}_M$ such that $C[\beta^{-1}(t')]_M D$ and $(D, D') \in \alpha$.

In the literature the relation α is often called a *bisimulation*, and weak configuration equivalence is then called *bisimilarity* or *observation equivalence* (see, e.g., [Mil89]).

Condition (1) means that both systems start in corresponding configurations. Condition (2) means the following: if both systems are in corresponding configurations and M takes a step by firing transition t , then M' can simulate that step by firing the corresponding transition $\beta(t)$, after which M and M' are in corresponding configurations again. Condition (3) says the same as condition (2), with the roles of M and M' reversed.

As in the case of configuration equivalence, it can be shown also here (using Theorem 20) that conditions (2) and (3) also hold for concurrent steps $C[U]_M D$ and $C'[\beta(U)]_{M'} D'$.

It is clear that configuration equivalent EN systems are also weakly configuration equivalent. On the other hand there exist weakly configuration equivalent EN systems that are not configuration equivalent (see the next example). Hence weak configuration equivalence is weaker than configuration equivalence (as the name already suggests!).

Example 14. Let M and M' be the EN systems in Figs. 33 and 35, respectively; their (sequential) configuration graphs are given in Figs. 34 and 36, respectively. Since $\text{SCG}(M)$ has less nodes than $\text{SCG}(M')$, $\text{SCG}(M)$ and $\text{SCG}(M')$ are not isomorphic, and so M and M' are not configuration equivalent. They are however weakly configuration equivalent, with the bisimulation α that consists of the following pairs: $(\{p_1\}, \{p_1\})$, $(\{p_2\}, \{p_2\})$, $(\{p_3\}, \{p_3\})$, $(\{p_3\}, \{p_3, q\})$, $(\{p_4\}, \{p_4\})$, $(\{p_4\}, \{p_4, q\})$, $(\{p_5\}, \{p_5, q\})$, and $(\{p_6\}, \{p_6\})$. For β we take the identity.

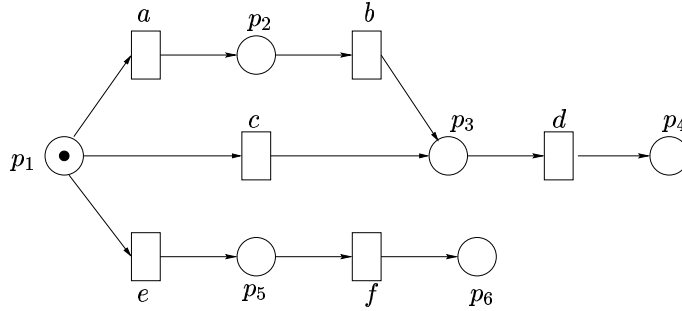


Fig. 33. An EN system M , weakly equivalent with the EN system M' of Fig. 35.

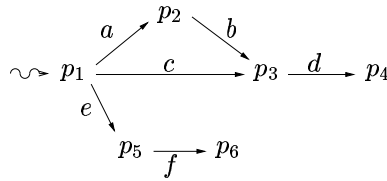


Fig. 34. The configuration graph of the EN system M of Fig. 33.

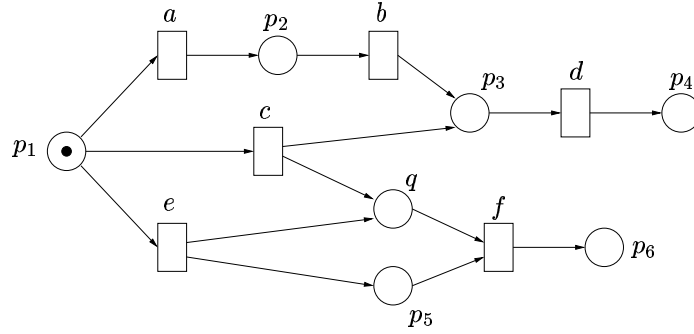


Fig. 35. An EN system M' , weakly equivalent with the EN system M of Fig. 33.

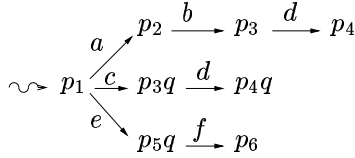


Fig. 36. The configuration graph of the EN system M' of Fig. 35.

An even less stringent definition of equivalence is obtained by requiring only that the firing sequences of two equivalent EN systems (bijectively) correspond to each other, with no requirement at all on the configurations of the two EN systems. Thus here the behaviour of an EN system M is defined as (represented by) its set $\text{FS}(M)$ of firing sequences, modulo the identity of the transitions. In this way we abstract completely from the global states of the system.

Note that if $\beta : \text{use}(T_M) \rightarrow \text{use}(T_{M'})$, where M and M' are EN systems, then $\beta(\text{FS}(M)) \subseteq \text{use}(T_{M'})^*$, see Section 3.

Definition 32. Let M and M' be two EN systems. M and M' are *firing sequence equivalent*, denoted by $M \approx_{fs} M'$, if there exists a bijection $\beta : \text{use}(T_M) \rightarrow \text{use}(T_{M'})$ such that $\beta(\text{FS}(M)) = \text{FS}(M')$.

Example 15. The (weakly configuration equivalent) EN systems M and M' of Example 14 (Figs. 33 and 35) are firing sequence equivalent because $\text{FS}(M) = \text{FS}(M') = \{\lambda, a, c, e, ab, cd, ef, abd\}$.

Even though firing sequence equivalence abstracts completely from the configurations, it turns out to be the same as weak configuration equivalence! This is essentially due to the fact that firing a transition leads from a configuration to a unique next configuration (see [Eng85, Mil89, PomRozSim92]).

Theorem 33. *Two EN systems are firing sequence equivalent iff they are weakly configuration equivalent.*

Proof. Let M and M' be two EN systems.

First assume that $M \approx_w M'$, with bisimulation α and bijection β . It suffices to prove that $\beta(\text{FS}(M)) \subseteq \text{FS}(M')$: by symmetry this also proves that $\beta^{-1}(\text{FS}(M')) \subseteq \text{FS}(M)$.

It follows directly from the definition of weak configuration equivalence that each event t of M can be simulated by event $\beta(t)$ of M' in such a way that the relation α between the (old and new) configurations continues to hold. Hence each firing sequence x of M can be simulated in such a way by the sequence $\beta(x)$ of M' . This implies that $\beta(x)$ is a firing sequence of M' . Formally, we use the following extension of Definition 31(2), which can easily be proved by induction on $|x|$: for all $C, D \in \mathbb{C}_M$, $C' \in \mathbb{C}_{M'}$, and $x \in \text{use}(T_M)^*$:

if $C[x]_M D$ and $(C, C') \in \alpha$, then there exists $D' \in \mathbb{C}_{M'}$ such that
 $C'[\beta(x)]_{M'} D'$ and $(D, D') \in \alpha$.

Taking $C = C_{in}$ and $C' = C'_{in}$, this shows, by Definition 31(1), that if $x \in \text{FS}(M)$ then $\beta(x) \in \text{FS}(M')$.

Now assume that $M = (P, T, F, C_{in})$ and $M' = (P', T', F', C'_{in})$ are firing sequence equivalent, with bijection β . Hence, if x is a firing sequence of M , then $\beta(x)$ is a firing sequence of M' , and vice versa. This leads to the following definition of the relation $\alpha \subseteq \mathbb{C}_M \times \mathbb{C}_{M'}$:

$$(C, C') \in \alpha \text{ iff } \exists x \in T^* : C_{in}[x]_M C \text{ and } C'_{in}[\beta(x)]_{M'} C'.$$

It is straightforward to verify that α is a bisimulation, i.e., that α and β satisfy the three conditions in Definition 31. The details of that proof will be considered in Exercise 5.2. \square

Since configuration equivalence implies weak configuration equivalence, the following corollary is obtained from the above theorem.

Corollary 34. *If two EN systems are configuration equivalent, then they are also firing sequence equivalent.*

This corollary can also be seen directly: if two EN systems M and M' are configuration equivalent, then they have isomorphic configuration graphs and hence $\text{FS}(M)$ and $\text{FS}(M')$ are recognized by isomorphic finite automata (see Theorem 12).

To recapitulate, in this subsection we have introduced four equivalence relations for EN systems, of which one is static (isomorphism) and the other three are dynamic, depending on the intuitive notion of “behaviour”. The following relationships hold between these four equivalences: isomorphism implies configuration equivalence, which in turn implies firing sequence equivalence, which equals weak configuration equivalence. Thus in formal notation we have, for EN systems M and M' ,

$$M \equiv M' \Rightarrow M \approx M' \Rightarrow M \approx_{fs} M' \Leftrightarrow M \approx_w M'.$$

5.2 Reduction

We now turn to normal forms for EN systems.

It is intuitively natural to assume that an EN system contains no “superfluous” transitions and places, hence in particular no useless transitions and no isolated places. Such transitions and places clearly play no part in the behaviour of the system. We will formalize this intuition by showing that for every EN system M there exists an equivalent EN system M' containing useful transitions and nonisolated places only. This holds for the strongest dynamic notion of equivalence from the previous subsection, viz. configuration equivalence.

Definition 35. An EN system M is *reduced* if all transitions of M are useful. M is *strongly reduced* if M is reduced and has no isolated places.

Our first normal form result is the following.

Theorem 36. *For every EN system M there exists a reduced EN system M' such that $M \approx M'$.*

Proof. Let $M = (P, T, F, C_{in})$. We construct M' by simply removing all useless transitions. Let M' be the EN system (P, T', F', C_{in}) with $T' = \mathbf{use}_M(T)$ and $F' = F \cap ((P \times T') \cup (T' \times P))$. It is clear that M and M' even have the same sequential configuration graph and thus are configuration equivalent. Hence M' contains only useful transitions and is thus reduced. \square

This theorem can be strengthened as follows.

Theorem 37. *For every EN system M there exists a strongly reduced EN system M' such that $M \approx M'$.*

Proof. Let $M = (P, T, F, C_{in})$. According to Theorem 36 we may assume that M is reduced, i.e., $\mathbf{use}_M(T) = T$. We now simply construct M' by removing all isolated places. Define $\mathbf{use}(P) = \{p \in P \mid \mathbf{nbh}(p) \neq \emptyset\}$; thus $\mathbf{use}(P)$ is the set of nonisolated places of M . Let $M' = (P', T, F, C'_{in})$ with $P' = \mathbf{use}(P)$ and $C'_{in} = C_{in} \cap P'$. Obviously M' has no isolated places.

Define the function $\alpha : \mathbb{C}_M \rightarrow \mathcal{P}(P')$ with $\alpha(C) = C \cap P'$ for all $C \in \mathbb{C}_M$ and let β be the identity on T . We prove that $M \approx_{\beta}^{\alpha} M'$ using Lemma 30. It is easy to prove by induction on C that, for all $C \in \mathbb{C}_M$, $C - P' = C_{in} - P'$ (intuitively this holds because the marking of the isolated places never changes). This implies that α is injective. Also $\alpha(C_{in}) = C'_{in}$. By Lemma 30 it now suffices to prove that $C[t]_M D$ iff $C \cap P'[t]_{M'} D \cap P'$. This can easily be proved using Lemma 7, because $(C \cap P') - (D \cap P') = (C - D) \cap P'$ (for arbitrary sets) and $\bullet t \cap P' = \bullet t$, and analogously for $D - C$ and $t \bullet$. Hence M' is (α, β) -configuration equivalent with M . Finally, by Definition 28, β is a bijection between $\mathbf{use}_M(T)$ and $\mathbf{use}_{M'}(T)$. This implies that $\mathbf{use}_{M'}(T) = T$. Hence M' is reduced and thus strongly reduced. \square

Example 16. According to the constructions in the proofs of Theorems 36 and 37, the EN system M in Fig. 37 is transformed into the equivalent strongly reduced EN system M' of Fig. 38: first the useless transition t_3 is removed and then the isolated places p_7 and p_8 are removed.

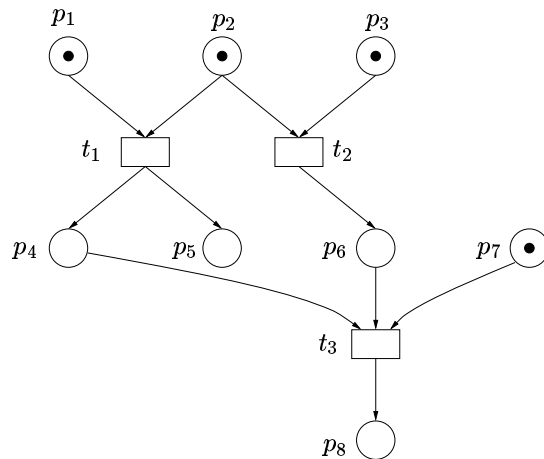


Fig. 37. An EN system with useless transition t_3 .

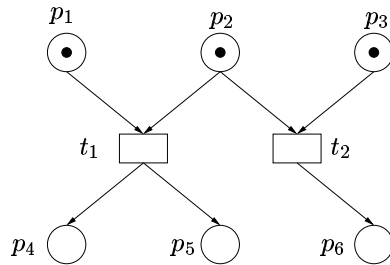


Fig. 38. A strongly reduced EN system, configuration equivalent with the EN system of Fig. 37.

We will now show that a strongly reduced EN system contains no “superfluous” places in the following sense: for every condition there exists a configuration

in which this condition holds, and there exists a configuration in which this condition does not hold. Hence there are no void, static conditions that either always hold or never hold.

Theorem 38. *Let $M = (P, T, F, C_{in})$ be a strongly reduced EN system. For every $p \in P$ there exist configurations $C, D \in \mathbb{C}_M$ such that $p \in C$ and $p \notin D$.*

Proof. Since p is not isolated, there is a transition t such that $t \in \mathbf{nbh}(p)$. Since t is useful, there is a reachable configuration E with $t \mathbf{con} E$. Let $E[t]E'$. If $t \in p^\bullet$, then $p \in E$ and $p \notin E'$. And if $t \in \bullet p$, then $p \in E'$ and $p \notin E$. \square

We have now proved that isolated places and useless transitions can be removed from an EN system M . But, in general, there are also other kinds of “superfluous” places and transitions, in particular when M is not simple (see Definition 2(2,3)). There are two types of simplicity, which we will now consider from this viewpoint.

First assume that M is not P-simple. Then there are two distinct places p and q such that $\bullet p = \bullet q$ and $p^\bullet = q^\bullet$. It is intuitively clear that one of these two places is superfluous. It can easily be shown that the removal of either p or q results in an EN system that is configuration equivalent with M . Hence for every EN system there exists a configuration equivalent strongly reduced EN system that is P-simple.

Now assume that M is not T-simple. Then there are two distinct transitions s and t such that $\bullet s = \bullet t$ and $s^\bullet = t^\bullet$. Intuitively one of these two transitions is again superfluous. However, if s and t are useful, then the removal of either s or t does not result in a configuration equivalent EN system, simply because the number of useful transitions of two configuration equivalent EN systems must be equal (β is a bijection). We will now show that T-simplicity is not a normal form for EN systems, not even with respect to the weakest kind of equivalence (viz. firing sequence equivalence).

Theorem 39. *There exists an EN system M such that for every EN system M' : if $M' \approx_{fs} M$, then M' is not T-simple.*

Proof. The following technical property of an (arbitrary) EN system M' will be useful in our proof: if $s, u \in T_{M'}$ and there exists an $x \in T_{M'}^*$, such that $xsus \in \mathbf{FS}(M')$, then $s^\bullet \subseteq \bullet u$ and $\bullet s \subseteq u^\bullet$ (verify that; see Exercise 4.7).

Now consider the EN system $M = (P, T, F, C_{in})$ with $P = \{p, q\}$, $T = \{s, t, u\}$, $C_{in} = \{p\}$, $\bullet s = \bullet t = u^\bullet = \{p\}$ and $s^\bullet = t^\bullet = \bullet u = \{q\}$. Suppose that M' is an EN system that is firing sequence equivalent with M , and assume for the sake of simplicity that $\mathbf{FS}(M') = \mathbf{FS}(M)$ (i.e., that β is the identity). Since $sus \in \mathbf{FS}(M) = \mathbf{FS}(M')$, the technical property given above implies that $s^\bullet \subseteq \bullet u$ and $\bullet s \subseteq u^\bullet$ hold in M' . Likewise it follows from $susu \in \mathbf{FS}(M')$ that $u^\bullet \subseteq \bullet s$ and $\bullet u \subseteq s^\bullet$ (take $x = s$ and interchange s and u in the statement of the above technical property). Hence $s^\bullet = \bullet u$ and $\bullet s = u^\bullet$ in M' . Analogously we have $t^\bullet = \bullet u$ and $\bullet t = u^\bullet$ in M' . Hence $s^\bullet = t^\bullet$ and $\bullet s = \bullet t$ in M' , and M' is thus not T-simple. \square

We would like to point out that the difference between P-simplicity and T-simplicity demonstrated above is a direct consequence of the fact that we are more interested in transitions than in places (which is formally expressed in our definitions of equivalence by requiring the existence of a bijection β between the useful transitions).

Note that it is also not possible to find for every EN system M a firing sequence equivalent EN system M' that has only live transitions. In fact, liveness of a transition t of M means (cf. Definition 8(6)) that for every $x \in \text{FS}(M)$ there exists $y \in T_M^*$ such that $xyt \in \text{FS}(M)$. This implies that if $\beta(\text{FS}(M)) = \text{FS}(M')$, then $t \in T_M$ is live iff $\beta(t) \in T_{M'}$ is live. Hence liveness of transitions is preserved by firing sequence equivalence.

5.3 Sequential EN Systems

An EN system can often be better understood when it can be seen as several communicating concurrent subsystems (or components), where each such subsystem is “simpler” (can be easier understood) than the whole system itself. It would be particularly desirable if the considered EN system could be decomposed into subsystems that no longer contain concurrency themselves, i.e., into sequential EN systems. Though this is not always directly possible, we will show in the next subsection that every EN system is equivalent with such a system. Hence “sequentially decomposable” systems are a normal form for EN systems.

In this subsection we define sequential EN systems and study several of their properties. We call an EN system sequential if its global states are not distributed (see the beginning of Section 4.3).

Definition 40. An EN system M is *sequential* if $\#C = 1$ for all $C \in \mathbb{C}_M$.

Example 17. The EN systems of Figs. 30 and 33 are sequential. This can be seen from their configuration graphs, which are given in Figs. 32 and 34, respectively.

Here is a sufficient (structural) condition for sequentiality; it is satisfied by the EN systems of the previous example.

Lemma 41. *If $M = (P, T, F, C_{in})$ is an EN system for which*

- (1) $\#C_{in} = 1$, and
 - (2) $\#(\bullet t) = \#(t\bullet) = 1$ for all $t \in T$,
- then M is sequential.*

Proof. It is easy to prove by induction on C that $\#C = 1$ for all $C \in \mathbb{C}_M$. \square

Systems that just satisfy the second property of this lemma are called *state machines* or *S-systems* in the literature, see, e.g., [Hac72, DesEsp95].

Finite automata, which are the usual model of finite-state sequential systems, are closely related to EN systems with the two properties of Lemma 41. The differences are that finite automata also have final states, that their transitions are labeled, and that transitions t with $\bullet t = t\bullet$ are allowed (cf. Definition 1(4)).

These additional features have also been considered for Petri nets in the literature (see, e.g., [Pet81, Tau89, Och95]).

It should be clear that in the case of reduced EN systems the two properties of Lemma 41 characterize sequentiality. In fact, for a sequential EN system M , a transition with more than one place in its input- or output-set must be useless. If, moreover, M is strongly reduced, then every configuration $\{p\}$ is reachable, because every place p is the input- or output-set of at least one useful transition.

Lemma 42. *Let $M = (P, T, F, C_{in})$ be a reduced EN system.*

- (1) *M is sequential iff*
 - (i) *$\#C_{in} = 1$, and*
 - (ii) *$\#(\bullet t) = \#(t\bullet) = 1$ for all $t \in T$.*
- (2) *If M is strongly reduced and sequential, then $\mathcal{C}_M = \{\{p\} \mid p \in P\}$.*

From this lemma it easily follows that isomorphism and configuration equivalence coincide for strongly reduced sequential systems.

Theorem 43. *Let M and M' be two strongly reduced sequential EN systems. Then $M \approx M'$ iff $M \equiv M'$.*

Every strongly reduced sequential system is in fact isomorphic with the finite automaton constructed in the proof of Theorem 12.

The definition of a sequential EN system (Definition 40) is “place oriented”: the global state of the system is not distributed. Another possibility is a “transition oriented” definition, where we require that the global state transitions are not distributed (see the beginning of Section 4.3), i.e., that concurrent steps do not occur (cf. the discussion of the notions of sequentiality and concurrency in Section 4.5). EN systems satisfying this property are “concurrency-free”.

Definition 44. An EN system M is *concurrency-free* if there do not exist $C \in \mathcal{C}_M$ and $t_1, t_2 \in T_M$ such that $\{t_1, t_2\} \mathbf{con} C$.

It should be clear that every sequential EN system is concurrency-free. This does not hold the other way around: the EN systems of Figs. 29 and 35 are concurrency-free (as can be seen from their configuration graphs in Figs. 31 and 36, respectively), but not sequential. It is shown in Examples 13 and 14 that these EN systems are (weakly) configuration equivalent with sequential EN systems. However, there exist concurrency-free EN systems that are not even firing sequence equivalent with any sequential EN system (this will be proved in Exercise 5.6).

Finally we observe that Theorem 43 does not hold for concurrency-free systems: the EN systems M (Fig. 29) and M' (Fig. 30) from Example 13 are configuration equivalent, but not isomorphic.

5.4 Subsystems and Sequential Components

In this subsection we will show that every EN system is configuration equivalent with an EN system that can be decomposed into sequential subsystems, i.e.,

in subsystems that, considered on their own, are sequential EN systems (see Definition 40). Sequential subsystems will also be called sequential components.

We first define the notions ‘subsystem’ and ‘sequential component’ and then study several of their characteristics. These notions, or variants of them, can be found in, e.g., [Hac72, DesEsp95].

Intuitively, a subsystem M' of M consists of a set of places of M (the local states of the subsystem) together with all transitions of M that can put tokens in these places and/or can remove tokens from these places, i.e., all transitions that belong to the neighbourhoods of these places. The flow relation F' of M' is completely determined by the flow relation F of M : a place and a transition of M' are connected by F' iff they are connected by F . The same holds for the initial configuration: a place of M' belongs to the initial configuration of M' iff it belongs to the initial configuration of M . Hence the subsystem is uniquely determined by its set of places.

Definition 45. Let $M = (P, T, F, C_{in})$ and $M' = (P', T', F', C'_{in})$ be EN systems. M' is a *subsystem* of M if:

- (1) $P' \subseteq P$, $T' \subseteq T$, $F' \subseteq F$, $C'_{in} \subseteq C_{in}$,
- (2) $\forall p \in P' : \mathbf{nbh}_M(p) \subseteq \mathbf{nbh}_{M'}(p)$, and
- (3) $\forall p \in P' : \text{if } p \in C_{in}, \text{ then } p \in C'_{in}$.

If, moreover, M' is a sequential EN system, then M' is a *sequential component* of M .

Note that the empty EN system $(\emptyset, \emptyset, \emptyset, \emptyset)$ and M itself are subsystems of M . They will be called the *trivial* subsystems of M .

In the next lemma we state some properties of subsystems that are easy to prove.

Lemma 46. Let $M = (P, T, F, C_{in})$ and $M' = (P', T', F', C'_{in})$ be EN systems.

- (1) M' is a *subsystem* of M iff
 $P' \subseteq P$, $T' = \mathbf{nbh}_M(P')$, $F' = F \cap ((P' \times T') \cup (T' \times P'))$, and $C'_{in} = C_{in} \cap P'$.
- (2) If M' is a *subsystem* of M then:
for every $t \in T'$, $(\bullet t)_{M'} = (\bullet t)_M \cap P'$ and $(t \bullet)_{M'} = (t \bullet)_M \cap P'$,
for every $t \in T - T'$, $\mathbf{nbh}_M(t) \cap P' = \emptyset$,
for every $p \in P'$, $(\bullet p)_{M'} = (\bullet p)_M$ and $(p \bullet)_{M'} = (p \bullet)_M$.

Lemma 46(1) says that a subsystem is indeed completely determined by its set of places. However, not every set of places of an EN system determines a subsystem! Formally this means that, for a given EN system $M = (P, T, F, C_{in})$ and a given subset P' of P , the 4-tuple $M' = (P', T', F', C'_{in})$ with $T' = \mathbf{nbh}_M(P')$, $F' = F \cap ((P' \times T') \cup (T' \times P'))$, and $C'_{in} = C_{in} \cap P'$ does not have to be an EN system. For example, take $P = \{p, q\}$, $T = \{t\}$, $F = \{(p, t), (t, q)\}$, and $C_{in} = \{p\}$, and consider $P' = \{p\}$. Then $T' = \{t\}$, $F' = \{(p, t)\}$, and $C'_{in} = \{p\}$. But M' is not an EN system because (P', T', F') is not a net: $(t \bullet)_{M'} = \emptyset$ (see Definition 1(3)).

If the 4-tuple M' determined by the set P' (as in Lemma 46(1)) is an EN system, then we call M' the *subsystem* of M *determined by* P' . We now characterize the sets of places that determine subsystems.

Lemma 47. *Let $M = (P, T, F, C_{in})$ be an EN system and let $S \subseteq P$. There exists a subsystem M' of M with $P_{M'} = S$ iff $\bullet S = S^\bullet$.*

Proof. (Only-if) Assume that $M' = (S, T', F', C'_{in})$ is a subsystem of M . We first prove that $\bullet S \subseteq S^\bullet$ (where the \bullet is of course the one of M). Take a $t \in \bullet S$ and let $p \in S$ such that $t \in \bullet p$. Then, according to Definition 45(2), $t \in \mathbf{nbh}_{M'}(p)$ and thus $t \in T'$. Definition 1(3) then implies that there exists $q \in S$ with $(q, t) \in F'$, and thus $(q, t) \in F$. This means that $t \in S^\bullet$. The inclusion $S^\bullet \subseteq \bullet S$ can be proved analogously.

(If) Assume that $\bullet S = S^\bullet$. Define $M' = (P', T', F', C'_{in})$ with $P' = S$ and T', F' , and C'_{in} as in Lemma 46(1). It is easy to check that M' is an EN system, i.e., M' satisfies the conditions from Definitions 1 and 5. In particular, the equality $\bullet S = S^\bullet$ guarantees that Definition 1(3) is satisfied. According to Lemma 46(1), M' is then a subsystem of M . \square

Note that condition (3) from Definition 1 is essentially used in the above proof.

Since a subsystem is uniquely determined by its set of places, in the sequel we will *often make no distinction between a subsystem and its set of places*. Thus, according to Lemma 47: a set of places S is a subsystem iff $\bullet S = S^\bullet$. This property of S has the following equivalent formulation: for all $t \in T$, $\bullet t \cap S \neq \emptyset$ iff $t^\bullet \cap S \neq \emptyset$. Note also that, for a subsystem $M' = (S, T', F', C'_{in})$ of M , $T' = \mathbf{nbh}(S) = \bullet S \cup S^\bullet$ according to Lemma 46(1), and hence $T' = \bullet S = S^\bullet$. Finally note that, according to Lemma 47, the union of two subsystems is again a subsystem (or more precisely: the union of the sets of places of two subsystems determines a subsystem).

Example 18. (1) The subset $\{p_3, p_5\}$ of the set of places of the EN system M from Fig. 39 is a subsystem of M ; this can easily be proved using Lemma 47. The subsystem determined by $\{p_3, p_5\}$ is $M' = (S, T, F, C_{in})$ with $S = \{p_3, p_5\}$, $T = \{t_3, t_4\}$, $F = \{(p_3, t_4), (t_4, p_5), (p_5, t_3), (t_3, p_3)\}$, and $C_{in} = \{p_3\}$. Since M' is sequential, M' is thus a sequential component of M . Another subsystem of M is $\{p_1, p_2, p_3, p_4\}$; for this subsystem $T = \{t_1, t_2, t_3, t_4\}$, $F = \{(t_1, p_1), (p_1, t_2), (t_2, p_2), (p_2, t_3), (t_3, p_3), (p_3, t_4), (t_4, p_4), (p_4, t_1)\}$, and $C_{in} = \{p_1, p_3\}$. Since $\#C_{in} > 1$ this is not a sequential component of M . Using Lemma 47 one can check that the only other subsystems of M are the trivial ones, i.e., the empty subsystem and the system M itself.

(2) Subsystems of the EN system M of Fig. 12 are $\{p_1, p_2\}$ (the consumer), $\{c_1, c_2\}$ (the producer), and $\{p_1, p_2, c_1, c_2\}$ (the union of the producer and the consumer). The producer and the consumer are both sequential components of M , but their union is not. M has no other nontrivial subsystems. In particular $\{b\}$ (the buffer) is not a subsystem of M . Thus, our notion of subsystem is more restricted than suggested in the discussion of the producer/consumer problem in Section 4.1, where the buffer was viewed as a component of the system. In Example 23 (Fig. 49) we will see that there is a slight variation of M in which the buffer is also a sequential component.

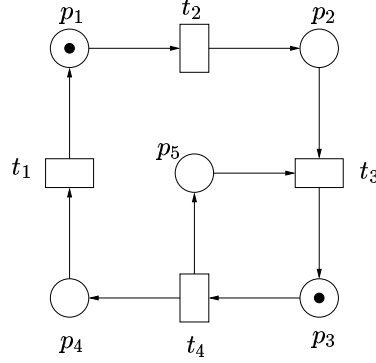


Fig. 39. An EN system with two nontrivial subsystems: $\{p_3, p_5\}$ and $\{p_1, p_2, p_3, p_4\}$.

We will give (in Theorem 49) a characterization of the sets of places that determine sequential components. First we prove an important property of subsystems: when we restrict a reachable configuration of an EN system to a subsystem, then we obtain again a reachable configuration of the subsystem.

Lemma 48. Let $M' = (S, T', F', C'_{in})$ be a subsystem of an EN system $M = (P, T, F, C_{in})$.

- (1) For all $C \subseteq P$, if $C \in \mathbb{C}_M$ then $C \cap S \in \mathbb{C}_{M'}$.
- (2) For all $t \in T'$, if $t \in \mathbf{use}_M(T)$ then $t \in \mathbf{use}_{M'}(T')$.

Proof. Before proving (1) and (2), let us consider configurations C, D of M and a transition t of M such that $C \cap S \in \mathbb{C}_{M'}$ and $C[t]_M D$. Then we claim the following: if $t \notin T'$ then $D \cap S = C \cap S$, and if $t \in T'$ then $(C \cap S)[t]_{M'}(D \cap S)$. To prove this, note first that, by Lemma 7, $C - D = (\bullet t)_M$ and $D - C = (t \bullet)_M$, and hence $(C \cap S) - (D \cap S) = (\bullet t)_M \cap S$ and $(D \cap S) - (C \cap S) = (t \bullet)_M \cap S$. If $t \notin T'$, then, by Lemma 46(2), $(\bullet t)_M \cap S = \emptyset$ and $(t \bullet)_M \cap S = \emptyset$, and so $D \cap S = C \cap S$. If $t \in T'$, then, by Lemma 46(2), $(\bullet t)_{M'} = (\bullet t)_M \cap S$ and $(t \bullet)_{M'} = (t \bullet)_M \cap S$, and so, by Lemma 7, $(C \cap S)[t]_{M'}(D \cap S)$. This proves our claim.

(1) We prove this by induction on C . By Lemma 46(1) the statement holds for $C = C_{in}$. Now assume that $C \cap S \in \mathbb{C}_{M'}$ and let $C[t]_M D$ for a $t \in T$. By the above claim, either $t \notin T'$ and $D \cap S = C \cap S$, or $t \in T'$ and $(C \cap S)[t]_{M'}(D \cap S)$. In both cases, $D \cap S \in \mathbb{C}_{M'}$.

(2) Assume that $t \mathbf{con}_M C$ for a $C \in \mathbb{C}_M$ and let $C[t]_M D$. By (1), $C \cap S \in \mathbb{C}_{M'}$. Thus, by the above claim, $(C \cap S)[t]_{M'}(D \cap S)$. Hence $t \mathbf{con}_{M'} C \cap S$. \square

In the next example we will show that the mapping from \mathbb{C}_M to $\mathbb{C}_{M'}$ that maps C to $C \cap S$ (see Lemma 48(1)) need not be surjective nor injective.

Example 19. Let M be the EN system of Fig. 40. Its configuration graph is drawn in Fig. 41.

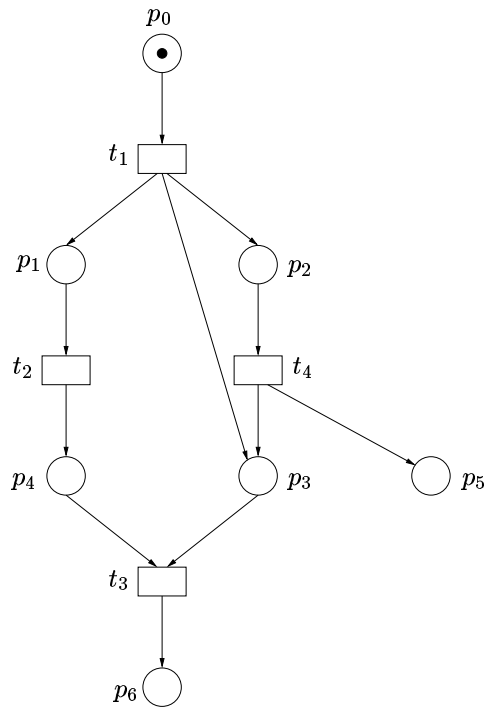


Fig. 40. An EN system M .

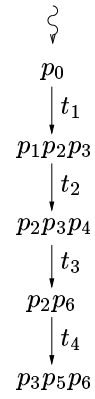


Fig. 41. The configuration graph of M .

(1) Figure 42 shows a subsystem M_1 of M , and Fig. 43 gives the configuration graph of M_1 . For $\{p_1, p_5\} \in \mathbb{C}_{M_1}$ there does not exist $C \in \mathbb{C}_M$ such that $C \cap P_{M_1} = \{p_1, p_5\}$; this also holds for $\{p_4, p_5\} \in \mathbb{C}_{M_1}$. Furthermore, note that though M is a concurrency-free EN system, M_1 is not.

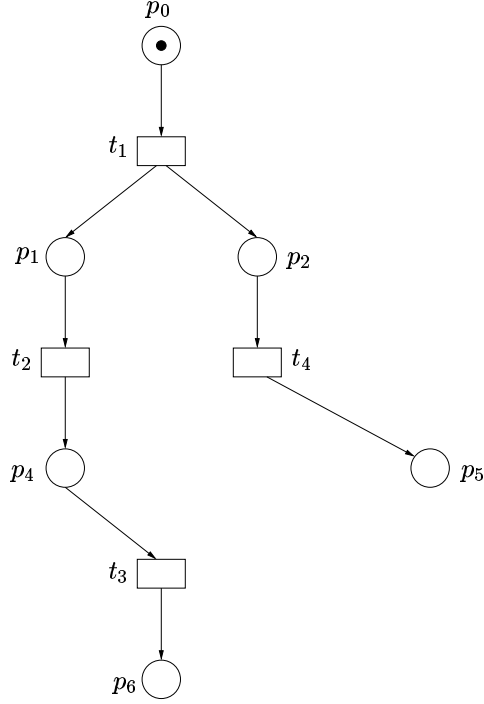


Fig. 42. A subsystem M_1 of M .

(2) Figure 44 shows a subsystem M_2 of M (and of M_1), and Fig. 45 gives the configuration graph of M_2 . This subsystem is a sequential component of M . The configurations $\{p_1, p_2, p_3\}, \{p_2, p_3, p_4\}, \{p_2, p_6\} \in \mathbb{C}_M$ all give the configuration $\{p_2\} \in \mathbb{C}_{M_2}$ when intersected with P_{M_2} .

Compare the following characterization with Lemma 42(1).

Theorem 49. *Let $M = (P, T, F, C_{in})$ be a reduced EN system and let $S \subseteq P$. Then the following statements are equivalent.*

- (1) *There is a sequential component M' of M with $P_{M'} = S$,*
- (2) *$\#(C \cap S) = 1$ for all $C \in \mathbb{C}_M$,*
- (3) *(i) $\#(C_{in} \cap S) = 1$, and*
(ii) $\forall t \in T : \#(\bullet t \cap S) = \#(t^\bullet \cap S) = 1$ or $\#(\bullet t \cap S) = \#(t^\bullet \cap S) = 0$.

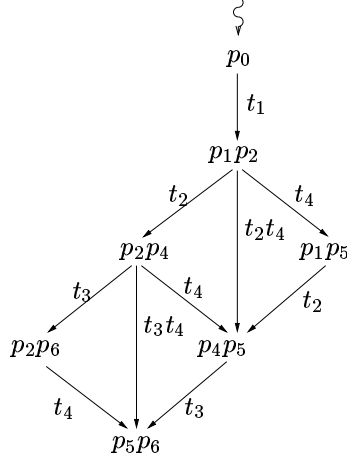


Fig. 43. The configuration graph of M_1 .

Proof. (1) implies (2): Follows directly from Lemma 48(1) and the definition of a sequential EN system (Definition 40).

(2) implies (3): To prove (ii), consider a $t \in T$. Since M is reduced, there is a $C \in \mathbb{C}_M$ with $t \text{ con } C$. Since $\#(C \cap S) = 1$, $\#(\bullet t \cap S) \leq 1$. Now let $C[t]D$. Since $\#(D \cap S) = 1$, it is easy to see that $\#(\bullet t \cap S) = \#(t \bullet \cap S)$.

(3) implies (1): Condition (3)(ii) implies that $\bullet S = S \bullet$. Hence, according to Lemma 47, there exists a (unique) subsystem M' of M with $P_{M'} = S$. Then, by Lemma 46(1), $\#((C_{in})_{M'}) = 1$ and, by Lemma 46(2), $\#((\bullet t)_{M'}) = \#((t \bullet)_{M'}) = 1$ for all $t \in T_{M'}$. Lemma 41 now implies that M' is sequential. \square

According to Theorem 49(2) a sequential component of an EN system is always in exactly one local state (there is always exactly one token in its set of places).

Subsystems S that just satisfy property 3(ii) of Theorem 49 and, moreover, are strongly connected (viewed as graphs), are called *state machine components* or *S-components* in the literature (see, e.g., [DesEsp95]). Such subsystems contain an arbitrary, but fixed number of tokens.

We will now show that, for strongly reduced EN systems, sequential components themselves do not have nontrivial subsystems and are thus not decomposable.

Lemma 50. *Let M be a strongly reduced sequential EN system and let M' be a subsystem of M . Then M' is trivial.*

Proof. Let $M = (P, T, F, C_{in})$ and $M' = (S, T', F', C'_{in})$. By Lemma 47, $\bullet S = S \bullet$. By Lemma 46 it suffices to prove that $S = \emptyset$ or $S = P$. Assume that $S \neq \emptyset$; then it remains to prove that $S = P$. Since M is strongly reduced,

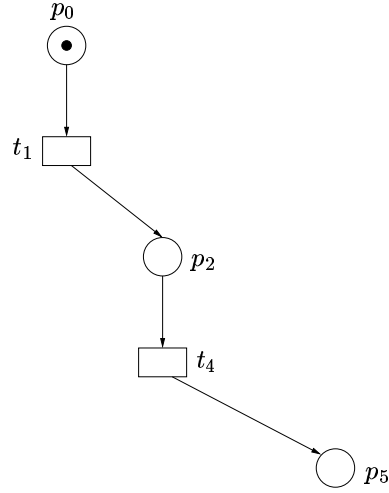


Fig. 44. A subsystem M_2 of M .

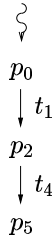


Fig. 45. The configuration graph of M_2 .

$\mathbb{C}_M = \{\{p\} \mid p \in P\}$ according to Lemma 42(2). Thus for all $t \in T$ and $p, q \in P$, if $\{p\}[t]_M\{q\}$, then $(p \in S \text{ iff } q \in S)$. This implies that for all $x \in T^*$ and $p, q \in P$, if $\{p\}[x]_M\{q\}$, then $(p \in S \text{ iff } q \in S)$. Now let $C_{in} = \{p_0\}$. Take a $q \in S$. Since $\{q\} \in \mathbb{C}_M$, $\{p_0\}[x]_M\{q\}$ for some $x \in T^*$. Thus $p_0 \in S$. To prove that $P \subseteq S$, consider an arbitrary $p \in P$. Since $\{p\} \in \mathbb{C}_M$, $\{p_0\}[x]_M\{p\}$ for some $x \in T^*$. Thus $p \in S$, and so $S = P$. \square

Theorem 51. *Let M be a strongly reduced EN system, and let M' be a sequential component of M . Then M' has no nontrivial subsystems.*

Proof. Since M is reduced, Lemma 48(2) implies that M' is also reduced. Since M has no isolated places, neither does M' (by Definition 45(2)). Hence M' is strongly reduced, and so, by Lemma 50 has no nontrivial subsystems. \square

The following example demonstrates that this theorem does not hold the other way around.

Example 20. Let M be the EN system from Example 19 (Fig. 40). Figure 46 shows a subsystem M_3 of M that is not sequential. It is straightforward to verify that M_3 has only trivial subsystems.

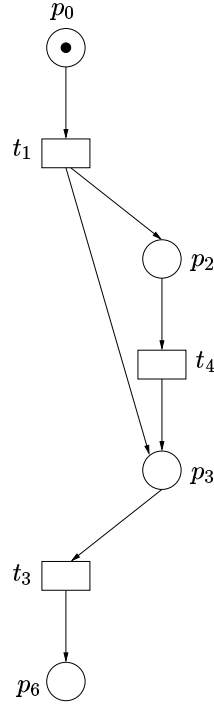


Fig. 46. A subsystem M_3 of M .

We now define the notion of a decomposition of an EN system into subsystems, and in particular into sequential components. It is called a “covering” of the EN system (see, e.g., [Hac72] or Chapter 5 of [DesEsp95]).

Definition 52. Let $M = (P, T, F, C_{in})$ be an EN system.

(1) A set $\{M_1, \dots, M_n\}$ of subsystems of M , $n \geq 0$, with $M_i = (S_i, T_i, F_i, (C_{in})_i)$ for $1 \leq i \leq n$, is a *covering* of M if $P = \bigcup_{i=1}^n S_i$, $T = \bigcup_{i=1}^n T_i$, $F = \bigcup_{i=1}^n F_i$, and $C_{in} = \bigcup_{i=1}^n (C_{in})_i$.

(2) M is *covered by sequential components* if there exists a covering $\{M_1, \dots, M_n\}$, $n \geq 0$, of M such that M_i is a sequential component of M for every $1 \leq i \leq n$.

Since a subsystem can be identified with its set of places, it should be clear that a set of subsystems of M is a covering of M if their union is the set of places of M .

Lemma 53. *Let $M = (P, T, F, C_{in})$ be an EN system, and let, for every $1 \leq i \leq n$ (with $n \geq 0$), $M_i = (S_i, T_i, F_i, (C_{in})_i)$ be a subsystem of M . Then $\{M_1, \dots, M_n\}$ is a covering of M iff $P = \bigcup_{i=1}^n S_i$.*

An EN system that is covered by sequential components M_1, \dots, M_n can intuitively be viewed as a system consisting of the communicating concurrent subsystems M_1, \dots, M_n , where each subsystem M_i is sequential. The communication between components takes place through synchronization on shared transitions. Components may also share places, meaning that the sets of places S_1, \dots, S_n of M_1, \dots, M_n , respectively, need not be disjoint. If there is a token in place $p \in S_1 \cap S_2$, then this means that both components M_1 and M_2 are in the same local state p . Hence a token in p represents all components to which p belongs. Note that, in general, an EN system can have several coverings by sequential components, i.e., several interpretations as a set of communicating components.

Example 21. (1) The EN system M of Fig. 47 is covered by one sequential component, viz. M itself. The system itself is sequential. With $C_{in} = \{p_1, p_2\}$ the system has no covering by sequential components.

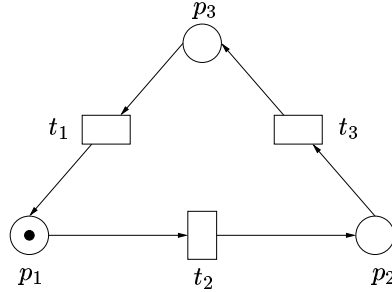


Fig. 47. A sequential EN system.

(2) The only sequential component of the EN system of Fig. 39 is $\{p_3, p_5\}$, see Example 18(1). We already saw in that example that Lemma 47 can be used to conclude that p_1 , p_2 , and p_4 do not belong to any sequential component. It is often quicker to use Theorem 49 for that purpose: for example, if S is the set of places of a sequential component and $p_2 \in S$, then $p_1 \in S$ and $p_3 \in S$ by Theorem 49(3)(ii), and hence $\#(C_{in} \cap S) = 2$, contradicting Theorem 49(2).

(3) The EN system of Fig. 15 has a covering by two sequential components, viz. $\{p_0, p_1, p_2, p_4, p_6\}$ and $\{p_0, p_1, p_3, p_5, p_6\}$. Intuitively, these two components work on the same job in places p_0, p_1 , and p_6 , but work on different jobs when they are in places p_2, p_4 and p_3, p_5 , respectively.

(4) The EN system of Fig. 5 is covered by three sequential components. In terms of the sets of places, the components are: $\{w_1, c_1, r_1\}$ (component 1), $\{w_2, c_2, r_2\}$ (component 2), and $\{p, c_1, c_2\}$ (the permission component). If we give the complete specification, then component 1 is the EN system (S, T, F, C_{in}) with $S = \{w_1, c_1, r_1\}$, $T = \{in_1, out_1, d_1\}$, $F = \{(w_1, in_1), (in_1, c_1), (c_1, out_1), (out_1, r_1), (r_1, d_1), (d_1, w_1)\}$, and $C_{in} = \{w_1\}$; and analogously for component 2. The permission component is then the EN system (S, T, F, C_{in}) with $S = \{p, c_1, c_2\}$, $T = \{in_1, out_1, in_2, out_2\}$, $F = \{(p, in_1), (in_1, c_1), (c_1, out_1), (out_1, p), (p, in_2), (in_2, c_2), (c_2, out_2), (out_2, p)\}$, and $C_{in} = \{p\}$.

(5) The EN system of Fig. 12 is not covered by sequential components. Both the producer ($\{p_1, p_2\}$) and the consumer ($\{c_1, c_2\}$) are sequential components, but the buffer b belongs to no sequential component (see also Example 18(2)).

(6) See Examples 19 and 20. M is the EN system in Fig. 40, and M_1 and M_3 are the subsystems of M in Figs. 42 and 46, respectively. It should be clear that $\{M_1, M_3\}$ is a covering of M . However, there does not exist a covering of M by sequential components, since each subsystem M' that contains p_3 must also contain p_2 and t_1 ; then $\#(t_1^\bullet \cap P_{M'}) \geq 2$, and so, by Theorem 49, M' is not a sequential component. M_1 is covered by sequential components, but M_3 is not.

We will now show that for every EN system M there exists a configuration equivalent EN system M' that is covered by sequential components.

Theorem 54. *For every EN system M there exists a reduced EN system M' that is configuration equivalent with M and that is covered by at most $\#P_M$ sequential components.*

In the remainder of this subsection we will prove this theorem. Note that according to Theorem 36 we may assume that M is reduced. The proof technique is based on the so-called *complement construction*.

Definition 55. Let M be an EN system and let $p, q \in P_M$. Then p and q are *complementary*, denoted by $p \mathbf{com} q$, if $p^\bullet = \bullet q$ and $\bullet p = q^\bullet$.

Example 22. Let M be the EN system of Fig. 39. It is clear that $p_3 \mathbf{com} p_5$, and that there are no other complementary places.

In general a place can have several complementary places. If the EN system is P-simple, then each place has at most one complementary place.

The complement construction is based on the following property of two complementary places.

Lemma 56. *Let $M = (P, T, F, C_{in})$ be a reduced EN system. For all $p, q \in P$, $\{p, q\}$ is a sequential component of M iff $\#(C_{in} \cap \{p, q\}) = 1$ and $p \mathbf{com} q$.*

Proof. We use Theorem 49(3). It is easy to check that, for arbitrary $p, q \in P$, $p \mathbf{com} q$ iff $S = \{p, q\}$ satisfies condition (ii) of Theorem 49(3) (where we use Definition 1(4): $\bullet t \cap t \bullet = \emptyset$ for all $t \in T$). \square

If, moreover, M is strongly reduced, then the condition $\#(C_{in} \cap \{p, q\}) = 1$ can be omitted from the statement of Lemma 56. Thus, for strongly reduced EN systems, the complementary places are exactly the sequential components of size two.

By Lemma 56, in a reduced EN system two complementary places p and q (of which exactly one is in C_{in}) form a sequential component. That means that there is a token in q iff there is no token in p . Viewing p and q as booleans, q is the negation of p . The complement construction constructs a complement for those places that do not yet belong to a sequential component. We do this place by place, as follows.

Theorem 57. *Let M be a reduced EN system and let $p_0 \in P_M$. Then there exists a reduced EN system M' that is configuration equivalent with M , such that:*

- (1) $P_{M'} = P_M \cup \{q_0\}$ with $q_0 \notin P_M$,
- (2) $\{p_0, q_0\}$ is a sequential component of M' , and
- (3) for every $S \subseteq P_M$,

S is a sequential component of M iff S is a sequential component of M' .

Proof. Let $M = (P, T, F, C_{in})$. We define $M' = (P', T', F', C'_{in})$ by setting

$$P' = P \cup \{q_0\}, \text{ where } q_0 \text{ is a new place (i.e., } q_0 \notin P \cup T),$$

$$T' = T,$$

$$F' = F \cup \{(q_0, t) \mid (t, p_0) \in F\} \cup \{(t, q_0) \mid (p_0, t) \in F\}, \text{ and}$$

$$C'_{in} = C_{in} \text{ if } p_0 \in C_{in}, \text{ and } C'_{in} = C_{in} \cup \{q_0\} \text{ if } p_0 \notin C_{in}.$$

To prove that M and M' are configuration equivalent, we define the function $\alpha : \mathbb{C}_M \rightarrow \mathcal{P}(P')$ by $\alpha(C) = C$ if $p_0 \in C$, and $\alpha(C) = C \cup \{q_0\}$ if $p_0 \notin C$. Note that α is injective (because $\alpha(C) \cap P = C$) and that $\alpha(C_{in}) = C'_{in}$. According to Lemma 30 (with β the identity on T) it now suffices to prove that for all $C, D \in \mathbb{C}_M$ and $t \in T$,

$$t \mathbf{con}_{M'} \alpha(C) \text{ implies } t \mathbf{con}_M C, \text{ and} \quad (*)$$

$$C[t]_M D \text{ implies } \alpha(C)[t]_{M'} \alpha(D). \quad (**)$$

Implication (*) follows from: $(\bullet t)_M = (\bullet t)_{M'} \cap P$, $(t \bullet)_M = (t \bullet)_{M'} \cap P$, and $\alpha(C) \cap P = C$. To prove (**) we distinguish three cases.

Case 1: $p_0 \in (t \bullet)_M$. Then $(\bullet t)_{M'} = (\bullet t)_M \cup \{q_0\}$ and $(t \bullet)_{M'} = (t \bullet)_M$. Moreover $p_0 \notin C$ and $p_0 \in D$, and so $\alpha(C) = C \cup \{q_0\}$ and $\alpha(D) = D$. Hence $\alpha(C) - \alpha(D) = (C - D) \cup \{q_0\}$ and $\alpha(D) - \alpha(C) = D - C$. Thus $C - D = (\bullet t)_M$ implies $\alpha(C) - \alpha(D) = (\bullet t)_{M'}$, and $D - C = (t \bullet)_M$ implies $\alpha(D) - \alpha(C) = (t \bullet)_{M'}$. By Lemma 7, this proves implication (**) for this case.

Case 2 ($p_0 \in (\bullet t)_M$) and Case 3 ($p_0 \notin (\bullet t)_M \cup (t \bullet)_M$) can be proved in a completely analogous way. Hence, by Lemma 30, M and M' are (α, β) -configuration equivalent. By Definition 28, $\mathbf{use}_{M'}(T') = T$, and so M' is reduced.

It is clear that (1) holds, (2) follows from Lemma 56, and (3) follows easily from Theorem 49(3). \square

Now, Theorem 54 follows directly from the repeated (at most $\#P_M$ times) application of Theorem 57. In this way, each place that did not yet belong to a sequential component will be complemented and consequently, by Lemma 56, covered by a sequential component.

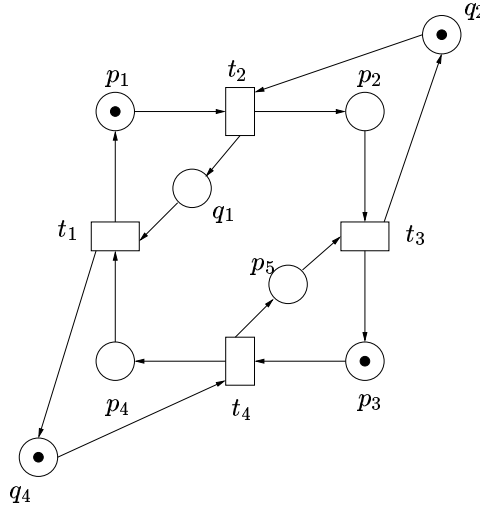


Fig. 48. The result of complementing places $p_1, p_2,$ and p_4 of the EN system of Fig. 39.

Example 23. (1) The EN system of Fig. 48 is obtained from the one of Fig. 39 by complementing places $p_1, p_2,$ and p_4 . (2) The EN system of Fig. 49 is obtained from the EN system of Fig. 12 by complementing the buffer place $b = b_f$. It is covered by three sequential components: the producer $\{p_1, p_2\}$, the consumer $\{c_1, c_2\}$, and the buffer $\{b_f, b_e\}$. It should be clear that this EN system still models the producer/consumer problem as discussed in Section 4.1. Rather than treating place b itself as the buffer (which is full iff it contains a token), we now represent the two possible states of the buffer by the two places b_f (full buffer) and b_e (empty buffer).

5.5 Contact-freeness

In an EN system M , an event t has concession in a configuration C if it has both input-concession (i.e., $\bullet t \subseteq C$) and output-concession (i.e., $t^\bullet \cap C = \emptyset$). A transition that has input-concession in C , need not have output-concession in C . This is called *contact*, and is illustrated in Fig. 50. As an example, transition f has contact in configuration $\{p_2, b, c_1\}$ of the producer/consumer system

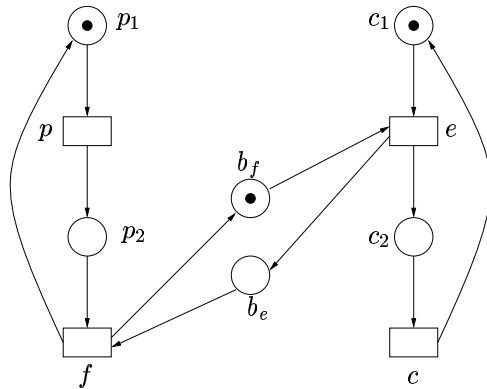


Fig. 49. The producer/consumer system with three sequential components.

in Fig. 12; it cannot be fired because the buffer is full. Note that in the corresponding configuration $\{p_2, b_f, c_1\}$ of the producer/consumer system of Fig. 49 transition f does not have input-concession.

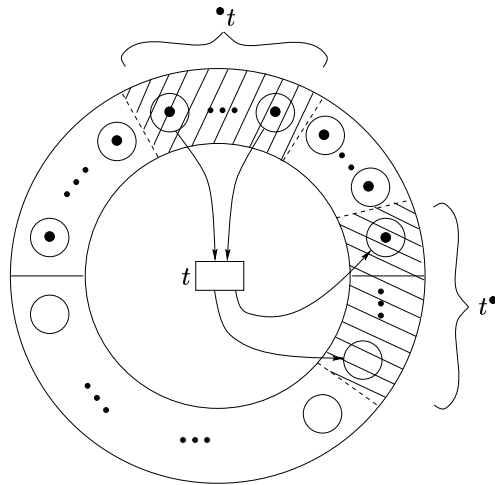


Fig. 50. Contact.

In general, to decide whether t has concession in C , one has to check both the pre-conditions and the post-conditions of t . However, if there is never contact in M , then t has concession in C iff all the pre-conditions of t in C are satisfied:

thus one does not have to check the post-conditions of $t!$

For this reason EN systems without contact play an important part in the theory of the behaviour of EN systems. They are formally defined as follows.

Definition 58. Let $M = (P, T, F, C_{in})$ be an EN system.

M is *contact-free* if for all $t \in T$ and $C \in \mathbb{C}_M$, if $\bullet t \subseteq C$ then $t^\bullet \cap C = \emptyset$.

Contact-free EN systems are also called *safe* EN systems. In fact, they are the same as the safe P/T systems, cf. Definition 158.

Example 24. The EN system of Fig. 47 is contact-free (because it is sequential). With $C_{in} = \{p_1, p_2\}$ it is not contact-free.

We will now show that for every EN system there exists a configuration equivalent reduced EN system that is contact-free. Hence contact-free EN systems are a normal form for EN systems. This is a simple corollary of the normal form from the previous subsection.

Theorem 59. *If a reduced EN system M is covered by sequential components, then M is contact-free.*

Proof. Let $M = (P, T, F, C_{in})$ and let $C \in \mathbb{C}_M$. Let $\bullet t \subseteq C$ with $t \in T$ and assume that $t^\bullet \cap C \neq \emptyset$. Let $p \in t^\bullet \cap C$ and let $S \subseteq P$ be a sequential component of M with $p \in S$. Then $p \in t^\bullet \cap S$ and so $\#(t^\bullet \cap S) = 1 = \#(\bullet t \cap S)$ by Theorem 49(3). Hence there is a place $q \in \bullet t \cap S \subseteq C \cap S$. Thus $\#(C \cap S) \geq 2$, contradicting Theorem 49(2). \square

Theorems 54 and 59 imply that contact-freeness is a normal form.

Theorem 60. *For every EN system there exists a configuration equivalent reduced contact-free EN system.*

Theorem 59 does not hold the other way around! The EN system of Fig. 51 has no sequential components, but it is contact-free. Systems for which the converse of Theorem 59 holds are discussed in Sections 9.6 and 9.7 (see Corollary 207 and Theorem 213).

Theorem 59 also provides a method to deduce contact-freeness, as demonstrated by the following example.

Example 25. (1) The EN system M in Fig. 47, with $C_{in} = \{p_1\}$, is sequential and hence contact-free according to Theorem 59. With $C_{in} = \{p_1, p_2\}$ the system clearly is not contact-free and hence, according to Theorem 59, it is not covered by sequential components (see Example 21(1)).

(2) The EN system M of Fig. 15 is covered by two sequential components (see Example 21(3)). Hence M is contact-free.

(3) The EN system M of Fig. 5 is covered by three sequential components (see Example 21(4)) and is thus also contact-free.

(4) The EN systems of Figs. 48 and 49 are covered by sequential components (see Example 23) and are thus contact-free.

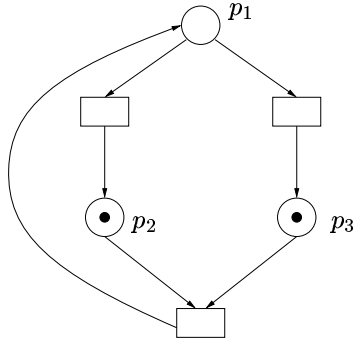


Fig. 51. A contact-free EN system without sequential components.

The notion of ‘contact-freeness’ should not be confused with the notion of ‘conflict-freeness’ (see Definition 22). An example of an EN system that is both contact-free and conflict-free, is the producer/consumer system of Fig. 49. Such systems can be characterized as follows.

Theorem 61. *Let $M = (P, T, F, C_{in})$ be an EN system. M is contact-free and conflict-free iff for all $C \in \mathcal{C}_M$ and all $U \subseteq T$ with $U \neq \emptyset$, if $\bullet U \subseteq C$, then $U \text{ con } C$.*

6 Processes

The way in which we have formalized the concurrent behaviour of an EN system in Section 4.4 still has a sequential flavour. The only difference with the sequential behaviour in Section 4.3 is that at each step, i.e., during each global state transition, several transitions can be fired simultaneously. This may be viewed as a formalization of simultaneity rather than concurrency. However, in general, actions do not occur simultaneously but they may overlap in time, in an arbitrary fashion. Thus, one component of a system can execute six actions while, independently, another component executes two actions and part of a third. In this section we will define the notion of a “process” of an EN system, which formalizes a concurrent run of the system, taking into account this feature of concurrency. In order to abstract from the notion of time, as we did before, we will only formalize that one action should be executed “before” another action, or, that one action is one of the “causes” of another action (cf. (1) in Section 4.5). Such a notion of causality between the events that occur during a run of the system, is, in general, a partial order. It will be represented by a special type of acyclic net, called “process net” (or causal net, or occurrence net).

In Section 6.1 we recall a number of notions concerning partial orders, and in Section 6.2 we consider process nets and some of their formal properties. In Sections 6.3 and 6.4 we introduce and study the processes of an EN system.

The theory of process nets and processes originated in [Pet76], and is presented in detail in [BesDev87, BesFer88] (for EN systems see in particular Section 4.4 of [BesFer88]).

6.1 Partial Orders

We start this subsection with the usual definition of a (strict) partial order.

Definition 62. Let A be a finite set. A binary relation $\rho \subseteq A \times A$ is a *partial order on A* if ρ is irreflexive and transitive; (A, ρ) is also called a *partially ordered set*. A subset B of A is *linearly ordered* if for all $a, b \in B$: $a \rho b$ or $b \rho a$ or $a = b$.

With every partial order ρ we can associate two important relations \mathbf{li}_ρ and \mathbf{co}_ρ .

Definition 63. Let (A, ρ) be a partially ordered set. Then $\mathbf{li}_\rho \subseteq A \times A$ and $\mathbf{co}_\rho \subseteq A \times A$ are the binary relations such that, for every $a, b \in A$,

- (1) $a \mathbf{li}_\rho b$ iff $a \rho b$ or $b \rho a$ or $a = b$, and
- (2) $a \mathbf{co}_\rho b$ iff $\neg a \rho b$ and $\neg b \rho a$.

For a partially ordered set (A, ρ) , \mathbf{li}_ρ is called the *line relation of ρ* and \mathbf{co}_ρ is called the *concurrency relation of ρ* .

Note that the irreflexivity of ρ implies that $a \mathbf{co}_\rho a$ for every $a \in A$. Two distinct elements of A are either comparable (**li**) or incomparable (**co**).

Lemma 64. Let (A, ρ) be a partially ordered set. Then, for every $a, b \in A$,

- (1) $a \mathbf{li}_\rho b$ or $a \mathbf{co}_\rho b$, and
- (2) ($a \mathbf{li}_\rho b$ and $a \mathbf{co}_\rho b$) iff $a = b$.

The maximal cliques of \mathbf{li}_ρ and \mathbf{co}_ρ play an important part in what follows. Maximal cliques are now defined for arbitrary reflexive symmetric relations (and note that \mathbf{li}_ρ and \mathbf{co}_ρ are reflexive and symmetric).

Definition 65. Let A be a finite set, let $\sigma \subseteq A \times A$ be a reflexive symmetric relation, and let $B \subseteq A$. B is a σ -*clique* if $a \sigma b$ for all $a, b \in B$, and B is a *maximal σ -clique* if B is a σ -clique and for every $a \in A - B$ there exists $b \in B$ such that $\neg a \sigma b$.

Lemma 66. Let A be a finite set and let $\sigma \subseteq A \times A$ be a reflexive symmetric relation. For every σ -clique B there exists a maximal σ -clique C with $B \subseteq C$.

Proof. If B is maximal then we are ready. Otherwise there exists $a_1 \notin B$ such that $a_1 \sigma b$ for all $b \in B$. Then $B_1 = B \cup \{a_1\}$ is a σ -clique. If B_1 is maximal then we are ready. Otherwise there exists $a_2 \notin B_1$ such that $B_2 = B_1 \cup \{a_2\}$ is a σ -clique. We iterate this procedure. Since A is finite it must terminate with a maximal σ -clique B_n . Clearly $B \subseteq B_n$. \square

In particular, for every $a \in A$ there exists a maximal σ -clique C with $a \in C$ (because every singleton $\{a\}$ is a σ -clique, by the reflexivity of σ).

For a partially ordered set (A, ρ) , a \mathbf{li}_ρ -clique is a linearly ordered subset of A , and a \mathbf{co}_ρ -clique is a set of mutually incomparable elements.

Definition 67. Let (A, ρ) be a partially ordered set. A maximal \mathbf{li}_ρ -clique is a *line of ρ* and a maximal \mathbf{co}_ρ -clique is a *cut of ρ* .

Note that if $A = \emptyset$ (and thus $\rho = \emptyset$), then the empty set is both a line and a cut of ρ . If $A \neq \emptyset$, then lines and cuts are nonempty sets.

It is clear from Lemma 64(2) that, for every line L and every cut C of ρ , $\#(L \cap C) \leq 1$. This leads to the following definition (see [Pet76] and Section 2.3 of [BesFer88], where it is called K-density).

Definition 68. Let (A, ρ) be a partially ordered set. The ordering ρ is *dense* if every line and every cut of ρ have a nonempty intersection.

The empty partially ordered set is clearly not dense. There also exist non-empty partially ordered sets that are not dense. Consider, e.g., the N-shaped partially ordered set (A, ρ) with $A = \{a, b, c, d\}$ and $\rho = \{(a, b), (c, b), (c, d)\}$. Then the line $\{c, b\}$ and the cut $\{a, d\}$ do not intersect.

A cut divides a partially ordered set into two parts: the part “before” (preceding) the cut, and the part “after” (following) the cut. This can be defined, for arbitrary subsets of A instead of cuts, as follows. We also define the sets of minimal and maximal elements of (a subset of) A .

Definition 69. Let (A, ρ) be a partially ordered set and let $B \subseteq A$. Then

$$\begin{aligned} (\rightarrow B)_\rho &= \{a \in A \mid \exists b \in B : a \rho b \text{ or } a = b\}, \\ (B \rightarrow)_\rho &= \{a \in A \mid \exists b \in B : b \rho a \text{ or } b = a\}, \\ ({}^\circ B)_\rho &= \{b \in B \mid \neg \exists b' \in B : b' \rho b\}, \text{ and} \\ (B^\circ)_\rho &= \{b \in B \mid \neg \exists b' \in B : b \rho b'\}. \end{aligned}$$

If ρ is clear from the context, then we will just write $\rightarrow B$, $B \rightarrow$, ${}^\circ B$, and B° . Intuitively, $\rightarrow B$ is the part of A *before* B (including B), $B \rightarrow$ is the part of A *after* B (including B), ${}^\circ B$ is the *initial* part of B , and B° is the *final* part of B . In the literature, $\rightarrow B$ and $B \rightarrow$ are often denoted $\downarrow B$ and $\uparrow B$, respectively.

The following technical lemma shows that every element of B is after its initial part and before its final part.

Lemma 70. *Let (A, ρ) be a partially ordered set and let $B \subseteq A$. Then $B \subseteq ({}^\circ B) \rightarrow$ and $B \subseteq \rightarrow (B^\circ)$.*

Proof. To prove that $B \subseteq ({}^\circ B) \rightarrow$, we have to show that for every $b \in B$ there exists $m \in {}^\circ B$ such that $m \rho b$ or $m = b$. Let $b \in B$. If $b \in {}^\circ B$ then we are ready. Otherwise there exists $b_1 \in B$ with $b_1 \rho b$. If $b_1 \in {}^\circ B$ then we are ready. Otherwise there exists $b_2 \in B$ with $b_2 \rho b_1$ and thus $b_2 \rho b$. We iterate this procedure. Since B is finite it must terminate with a $b_n \in {}^\circ B$ such that $b_n \rho b$. Formally all this can be proved by induction on $\#\{b' \in B \mid b' \rho b\}$.

The proof of $B \subseteq \rightarrow(B^\circ)$ is “dual”, i.e., follows from the above by considering the partially ordered set (A, ρ^{-1}) , with $\rho^{-1} = \{(a, b) \mid (b, a) \in \rho\}$. \square

Lemma 66 is a special case of (the second inclusion of) Lemma 70, for the partially ordered set (Σ, \subsetneq) where Σ is the set of all σ -cliques.

The following elementary properties of cuts can easily be proved from the definitions, using Lemma 70.

Theorem 71. *Let (A, ρ) be a partially ordered set and let B be a cut of ρ .*

- (1) ${}^\circ A$ and A° are cuts of ρ ,
- (2) $({}^\circ A)^\rightarrow = A$, $\rightarrow({}^\circ A) = {}^\circ A$, $(A^\circ)^\rightarrow = A^\circ$, and $\rightarrow(A^\circ) = A$,
- (3) $\rightarrow B \cup B^\rightarrow = A$ and $\rightarrow B \cap B^\rightarrow = B$,
- (4) ${}^\circ(\rightarrow B) = {}^\circ A$, $(\rightarrow B)^\circ = B$, ${}^\circ(B^\rightarrow) = B$, and $(B^\rightarrow)^\circ = A^\circ$.

Proof. As an example we prove that ${}^\circ A$ is a cut. From Definition 69 it is obvious that ${}^\circ A$ is a **co**-clique. By Lemma 70, for every $a \in A - {}^\circ A$ there exists $m \in {}^\circ A$ such that $m \rho a$. This shows that ${}^\circ A$ is a maximal **co**-clique. \square

We now present another result that uses Lemma 70: every line intersects the initial and the final cut (hence a “partial” density).

Lemma 72. *Let (A, ρ) be a partially ordered set with $A \neq \emptyset$ and let L be a line of ρ . Then $L \cap {}^\circ A \neq \emptyset$ and $L \cap A^\circ \neq \emptyset$.*

Proof. By duality it suffices to prove that $L \cap {}^\circ A \neq \emptyset$. Since A is nonempty, L is nonempty. Then, by Lemma 70 (with $B = L$), ${}^\circ L$ is nonempty. Since L is a li-clique, ${}^\circ L$ consists of precisely one element, say a . By Lemma 70 (with $B = A$) there exists $m \in {}^\circ A$ with $m \rho a$ or $m = a$. This implies that $L \cup \{m\}$ is a li-clique. Since L is maximal, $m \in L$. Hence $m \in L \cap {}^\circ A$ (and $m = a$). \square

6.2 Process Nets

For the description of the concurrent runs of an EN system we will define so-called processes. In defining processes, nets of a special kind are used: process nets. These nets will be treated in this subsection.

Definition 73. A net $N = (P, T, F)$ is a *process net* if:

- (1) N is acyclic, and
- (2) $\#(\bullet p) \leq 1$ and $\#(p^\bullet) \leq 1$ for all $p \in P$.

Hence, process nets are nets without cycles and with “unbranching” places only. They are also called *occurrence* nets or *causal* nets.

For every acyclic directed graph with edge relation F , the relation F^+ (which indicates the nonempty paths in the graphs) is a partial order on the set of nodes of the graph. Applying this to the directed graph G_N corresponding to the net N (see Section 4.2) gives the next result.

Lemma 74. *For every process net N , F_N^+ is a partial order on X_N .*

The above result allows us to consider a process net $N = (P, T, F)$ as a partially ordered set (X, F^+) . In this way, all terminology and notations concerning partial orders introduced so far can be carried over to process nets. In particular, we write \mathbf{li}_N and \mathbf{co}_N instead of \mathbf{li}_{F^+} and \mathbf{co}_{F^+} and speak about lines and cuts of N instead of lines and cuts of F^+ . Thus, for $x, y \in X$, $x \mathbf{co}_N y$ iff $\neg x F^+ y$ and $\neg y F^+ x$, and $x \mathbf{li}_N y$ iff $x F^+ y$ or $y F^+ x$ or $x = y$.

We are especially interested in cuts of a process net that consist of places only, i.e., that are configurations of the process net.

Definition 75. A *slice* of a process net N is a cut C of N such that $C \subseteq P_N$.

Lemma 76. Let $N = (P, T, F)$ be a process net and let $C \subseteq P$.

C is a slice of N iff

- (1) for all $p, q \in C$, $p \mathbf{co}_N q$, and
- (2) for every $p \in P - C$ there exists $q \in C$ such that $\neg p \mathbf{co}_N q$.

Proof. If C is a slice of N , then (1) and (2) follow directly from the definition of a cut of N . Now assume that (1) and (2) hold. We have to prove that C is a cut of N . By (1), C is a \mathbf{co}_N -clique. To show that it is maximal, it suffices, by (2), to prove that for every $t \in T$ there exists $q \in C$ such that $t F^+ q$ or $q F^+ t$. Let $p \in t^\bullet$, i.e., $t F p$. If $p \in C$, then we are ready. Now assume $p \notin C$. According to (2) there exists $q \in C$, such that $p F^+ q$ or $q F^+ p$. If $p F^+ q$, then $t F^+ q$ (since $t F p$). If $q F^+ p$, then $q F^+ t$, because ${}^\bullet p = \{t\}$. This last fact is based on property (2) in the definition of a process net. \square

This lemma says that a slice is the same as a maximal clique of the relation \mathbf{co}_N restricted to the set P . The next result then follows immediately from Lemma 66.

Lemma 77. Let $N = (P, T, F)$ be a process net. For every \mathbf{co}_N -clique $B \subseteq P$ there exists a slice C of N with $B \subseteq C$.

From now on we write ${}^\circ N$ for ${}^\circ X_N$ (the minimal elements of the net N). Likewise we write N° for X_N° (the maximal elements of N). By Theorem 71(1), ${}^\circ N$ and N° are cuts. They are even slices of N , because, by Definition 1(3), ${}^\bullet t \neq \emptyset$ and $t^\bullet \neq \emptyset$, for every transition t . Note that ${}^\circ N = \{p \in P_N \mid {}^\bullet p = \emptyset\}$ and $N^\circ = \{p \in P_N \mid p^\bullet = \emptyset\}$.

In the sequel, we will view each process net $N = (P, T, F)$ as the EN system $(P, T, F, {}^\circ N)$, i.e., with the initial slice ${}^\circ N$ as initial configuration. Property (2) of a process net (Definition 73) guarantees that this EN system is conflict-free, see the discussion following Definition 22.

If two process nets N and N' are isomorphic via a bijection $\alpha : P_N \rightarrow P_{N'}$, then, obviously, $\alpha({}^\circ N) = {}^\circ N'$. In other words: then they are also isomorphic as EN systems (cf. Definition 27).

Example 26. Fig. 52 shows an example of a process net N . The places in ${}^\circ N$ are marked with tokens. The places marked with crosses show another slice C of N . Note that $\rightarrow C$ is the part of N above (and including) the crossed places, whereas $C \rightarrow$ is the part below (and including) them.

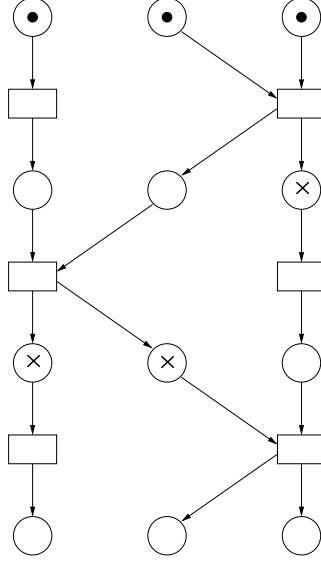


Fig. 52. A process net.

Here are some basic properties of **co**-cliques and slices of process nets.

Lemma 78. *Let $N = (P, T, F)$ be a process net.*

(1) *For every $U \subseteq T$, if U is a **co**-clique, then $\bullet U$ and U^\bullet are **co**-cliques. In particular, $\bullet t$ and t^\bullet are **co**-cliques for every $t \in T$.*

(2) *For every **co**-clique $U \subseteq T$ there exists a slice C such that $\bullet U \subseteq C$.*

(3a) *For every slice C and every $t \in T$, if $\bullet t \subseteq C$, then $t^\bullet \cap C = \emptyset$ and $D = (C - \bullet t) \cup t^\bullet$ is a slice such that $\rightarrow D = \rightarrow C \cup \{t\} \cup t^\bullet$.*

(3b) *For every slice C and every $t \in T$, if $t^\bullet \subseteq C$, then $\bullet t \cap C = \emptyset$ and $D = (C - t^\bullet) \cup \bullet t$ is a slice such that $\rightarrow D = \rightarrow C - t^\bullet - \{t\}$.*

(4) *For every slice C and every transition t , if $t \in \rightarrow C$, then $\text{nbh}(t) \subseteq \rightarrow C$.*

(5) *For every slice $C \neq \circ N$ there exists $t \in T$ such that $t^\bullet \subseteq C$.*

Proof. (1) Let U be a **co**-clique. To show that $\bullet U$ is a **co**-clique, let $p_1 \in \bullet t_1$ and $p_2 \in \bullet t_2$ with $t_1, t_2 \in U$, and suppose that $p_1 F^+ p_2$. Since $p_1^\bullet = \{t_1\}$ by Definition 73(2), $t_1 F^+ p_2$ and so $t_1 F^+ t_2$, contradicting the fact that U is a **co**-clique. In the same way it can be shown that U^\bullet is a **co**-clique.

(2) follows directly from (1) and Lemma 77.

* (3a) Let $\bullet t \subseteq C$. Since C is a **co**-clique, $t^\bullet \cap C = \emptyset$. To prove that D is a slice, we first show that it is a **co**-clique. Since both C and t^\bullet are **co**-cliques, it suffices to consider places $p \in C - \bullet t$ and $q \in t^\bullet$ such that $p F^+ q$ or $q F^+ p$. If $p F^+ q$, then, since $\bullet q = \{t\}$, there exists $q' \in \bullet t$ such that $p F^+ q'$. If $q F^+ p$, then $q' F^+ p$ for any $q' \in \bullet t$. In both cases, since $p, q' \in C$, this contradicts the fact

that C is a **co**-clique. Using Lemma 76, the maximality of D easily follows from the maximality of C . Thus, D is a slice. The proof of the remaining property of D is left to the reader.

(3b) Here t “fires backwards” in C yielding D . The proof is similar to the proof of (3a).

(4) Let $t \in \rightarrow C$. Thus, $t F^+ q$ for some $q \in C$. Obviously $\bullet t \subseteq \rightarrow C$. Now consider $p \in t^\bullet$ and suppose that $p \notin \rightarrow C$. Then, by Theorem 71(3), $p \in C^\rightarrow - C$ and so there exists $q' \in C$ such that $q' F^+ p$. Hence $q' F^+ t$, and so $q' F^+ q$, contradicting the fact that C is a **co**-clique.

(5) Since $C \neq \circ N$, $T \cap \rightarrow C \neq \emptyset$. Lemma 70 then implies that $(T \cap \rightarrow C)^\circ$ is also nonempty. Take a transition t that is a maximal element of $T \cap \rightarrow C$. We claim that $t^\bullet \subseteq C$. In fact, suppose that there exists $p \in t^\bullet$ such that $p \notin C$. Since, by (4), $p \in \rightarrow C$, there exists a place $q \in C$ such that $p F^+ q$. Hence there is a transition t' such that $p F t' F^+ q$. Consequently $t F^+ t'$, which contradicts the maximality of t , because $t' \in T \cap \rightarrow C$. \square

We will now show that the reachable configurations of a process net N are exactly the slices of N .

Theorem 79. *Let $N = (P, T, F, \circ N)$ be a process net and let $C \subseteq P$. $C \in \mathbb{C}_N$ iff C is a slice of N .*

Proof. (Only-if) By induction on C . The base of the induction holds because $\circ N$ is a slice. The induction step follows directly from Lemma 78(3a).

(If) By induction on $\#(\rightarrow C \cap T)$, i.e., the number of events before the slice C . If $\rightarrow C \cap T = \emptyset$, then $C = \circ N$ (verify this). If $\rightarrow C \cap T \neq \emptyset$, then $C \neq \circ N$ and so, by Lemma 78(5), there is a $t \in T$ such that $t^\bullet \subseteq C$. Then, by Lemma 78(3b), $\bullet t \cap C = \emptyset$ and $D = (C - t^\bullet) \cup \bullet t$ is a slice. Also $\rightarrow D = \rightarrow C - t^\bullet - \{t\}$ and hence $\#(\rightarrow D \cap T) = \#(\rightarrow C \cap T) - 1$. Thus, by the induction hypothesis, $D \in \mathbb{C}_N$. Obviously $D[t]C$ and hence $C \in \mathbb{C}_N$. \square

Theorem 79 and Lemma 78(3a) imply that every process net is contact-free. The conflict-freeness and contact-freeness of a process net can be expressed together as follows (see Theorem 61).

Theorem 80. *Let N be a process net, let $C \in \mathbb{C}_N$, and let $U \subseteq T_N$ with $U \neq \emptyset$. If $\bullet U \subseteq C$, then U **con** C .*

In general a process net is not strongly reduced, because it may contain isolated places. However, as we will show now, a process net is always reduced. We also give a characterization of the concurrent steps of a process net (i.e., the sets of labels that appear in its configuration graph).

Theorem 81. *Let $N = (P, T, F, \circ N)$ be a process net.*

- (1) N is reduced.
- (2) For every $U \subseteq T$, $(\exists C \in \mathbb{C}_N : U \text{ con } C)$ iff U is a **co**-clique.

Proof. (1) follows from (2), because for every $t \in T$, $\{t\}$ is a **co**-clique.

(2) If U is a **co**-clique then, by Lemma 78(2) and by Theorems 79 and 80, $U \text{ con } C$ for a $C \in \mathbb{C}_N$. The other way around, if $U \text{ con } C$, then $\bullet U \subseteq C$ and hence $\bullet U$ is a **co**-clique by Theorem 79. To show that U is a **co**-clique, suppose that $t_1 F^+ t_2$ for $t_1, t_2 \in U$. Then there exists $p_2 \in \bullet t_2$ such that $t_1 F^+ p_2$. Hence, by Definition 1(3), there exists $p_1 \in \bullet t_1$ such that $p_1 F^+ p_2$, contradicting the fact that $\bullet U$ is a **co**-clique. \square

Theorem 79 gives a characterization of the reachable configurations of a process net in terms of the partial order F^+ . We will now do the same for sequential components. Whereas reachable configurations correspond to slices, sequential components correspond to lines.

Lemma 82. *Let $N = (P, T, F, \circ N)$ be a process net.*

(1) *If L is a **li** $_N$ -clique, then $L \cup \text{nbh}(L \cap P)$ is a **li** $_N$ -clique.*

(2) *If L is a line of N , then for every $t \in T$:*

$\bullet t \cap L \neq \emptyset$ iff $t \in L$ iff $t \bullet \cap L \neq \emptyset$.

Proof. (1) Let L be a **li**-clique. We have to prove that $L \cup \bullet(L \cap P) \cup (L \cap P) \bullet$ is a **li**-clique. Take $x \in L$ and $t \in (L \cap P) \bullet$, hence $(p, t) \in F$ for a $p \in L$. If $x F^+ p$ then $x F^+ t$, and if $p F^+ x$ then $t F^* x$ (by Definition 73(2)). A similar reasoning holds for all other cases.

(2) We will prove that $\bullet t \cap L \neq \emptyset$ iff $t \in L$ (the proof that $t \in L$ iff $t \bullet \cap L \neq \emptyset$ can be done in the same way). First let $p \in \bullet t \cap L$. Then $t \in (L \cap P) \bullet$ and hence $L \cup \{t\}$ is a **li**-clique according to (1). Then $t \in L$, because L is a line. Now, for the implication in the other direction, let us assume that $t \in L$ and let $Y = \{x \in L \mid x F^+ t\}$. Lemma 72 implies that $Y \neq \emptyset$, and so, by Lemma 70, $Y^\circ \neq \emptyset$. Since L is a **li**-clique, Y° consists of one element, say x_m . Since $x_m F^+ t$, there exists $p \in \bullet t$ with $x_m F^* p$. This implies that $L \cup \{p\}$ is a **li**-clique and hence that $p \in L$. Thus $p \in \bullet t \cap L$. \square

Theorem 83. *Let $N = (P, T, F, \circ N)$ be a process net with $P \neq \emptyset$.*

(1) *If M is a sequential component of N , then $P_M \cup T_M$ is a line of N .*

(2) *If L is a line of N , then $(L \cap P, L \cap T, (L \times L) \cap F, L \cap \circ N)$ is a sequential component of N .*

Proof. (1) Let M be a sequential component of N and let $S = P_M$. Then $T_M = S^\bullet$ (see Lemmas 46 and 47). We have to show that $S \cup S^\bullet$ is a line of N . We first show that S is a **li**-clique. Let $p, q \in S$ and assume that $p \neq q$ and $p \text{ co } q$. Then, by Lemma 77, there is a slice C with $p, q \in C$. Theorem 79 implies that $C \in \mathbb{C}_N$. But then $\#(C \cap S) \geq 2$, contradicting Theorem 49(2) (which is applicable because, by Theorem 81(1), N is reduced). Hence S is a **li**-clique. Then, by Lemma 82(1), $S \cup S^\bullet$ is a **li**-clique. Now it remains to prove that $S \cup S^\bullet$ is maximal. First consider $p \in P - S$. By Lemma 77 and Theorem 79 there is a slice $C \in \mathbb{C}_N$ with $p \in C$. Since $\#(C \cap S) = 1$, there exists $q \in S$ with $q \in C$. Then, for this $q \in S \cup S^\bullet$, $p \text{ li } q$ does not hold. Now consider $t \in T - S^\bullet$. Then $\bullet t \cap S = \emptyset$. Again there is a slice $C \in \mathbb{C}_N$ with $\bullet t \subseteq C$ (see also Lemma 78(1)) and again

there exists $q \in C \cap S$. From the fact that $p \mathbf{co} q$ for every $p \in \bullet t$ it follows that $t \mathbf{li} q$ does not hold for this $q \in S \cup S^\bullet$.

(2) Let L be a line and $S = L \cap P$. It is easy to check that $M = (S, L \cap T, (L \times L) \cap F, L \cap \circ N)$ is a subsystem of N (where condition (2) of Definition 45 follows from Lemma 82(1) and the fact that L is a line). Now it remains to prove that M is sequential. By Theorem 49(2) and the fact that M is determined by S , it suffices to prove that $\#(C \cap S) = 1$ for every slice $C \in \mathbb{C}_N$. Since $\#(L \cap C) \leq 1$ and $C \subseteq P$, we only need to show that $L \cap C \neq \emptyset$. This is done by induction on C . The base of the induction follows from Lemma 72 and the induction step from Lemma 82(2): if $C[t]D$ and $\bullet t \cap L \neq \emptyset$, then $t^\bullet \cap L \neq \emptyset$. \square

From Lemma 53, Theorem 83(2), and Lemma 66 (for $\sigma = \mathbf{li}$) it follows that every process net is not only contact-free (as shown after Theorem 79), but is even covered by sequential components (cf. Theorem 59). As an example, the process net of Example 26 (Fig. 52) is covered by the three sequential components that correspond to the four places in each vertical row.

Also, Theorems 83(2), 79, and 49(2) imply that $L \cap C \neq \emptyset$ for every line L and every slice C . This is already close to density (see Definition 68).

Theorem 84. *Every process net $N = (P, T, F, \circ N)$, with $P \neq \emptyset$, is dense.*

Proof. Let L be a line and C be a cut of N . It is not hard to prove (analogously to Lemma 78(3a)) that $C' = (C \cap P) \cup (C \cap T)^\bullet$ is a slice. By Theorems 83(2), 79, and 49(2), $L \cap C' \neq \emptyset$. Let $p \in L \cap C'$. If $p \in C \cap P$, then $p \in L \cap C$. If $p \in (C \cap T)^\bullet$, then there is a $t \in C$ such that $p \in t^\bullet \cap L$. Lemma 82(2) then implies that $t \in L$, and hence $t \in L \cap C$. \square

Hence, density is an abstract version of the fact that in every reachable configuration every sequential component is in one particular state.

6.3 Processes

Process nets will be used in defining the notion of a process, which formalizes a concurrent run of a system. Informally speaking, a process of an EN system M describes a transformation of the initial configuration C_{in} of M to a configuration C of M ; it is a record of all occurrences of events that lead from C_{in} to C , together with all conditions involved in these events. Two occurrences of events are (partially) ordered in this record if there is a condition that starts to hold as a result of the first occurrence of an event, and ceases to hold as a result of the second (or if the first is connected to the second by a chain of occurrences of events related in this way). This partial order represents the causal connection between the occurrences of the events, cf. Fig. 19. Note that the linear order of occurrences of events in a firing sequence also represents the fact that they are observed by a sequential observer.

Consider for example the (contact-free) EN system $M = (P, T, F, C_{in})$ of Fig. 53. We start by recording the initial configuration $\{p_1, p_2\}$ (see Fig. 54).

Next we record the occurrence of event t_1 (see Fig. 55). What we obtain in this way, is a process (a record of a transformation) from the configuration $C_{in} = \{p_1, p_2\}$ to the configuration $C = \{p_3, p_4\}$. We can continue by also recording the occurrence of (for example) the concurrent step $\{t_2, t_3\}$ in configuration C , see Fig. 56. What we obtain now, is a process (a record of a transformation) leading from the configuration $C_{in} = \{p_1, p_2\}$ to itself. By continuing one more step, we again record the occurrence of t_1 and again obtain a process (a record of a transformation) from C_{in} to C (see Fig. 57). Finally we can (for example) record the occurrence of event t_4 and thus obtain a process leading from C_{in} to the configuration $\{p_4, p_5\}$ (see Fig. 58).

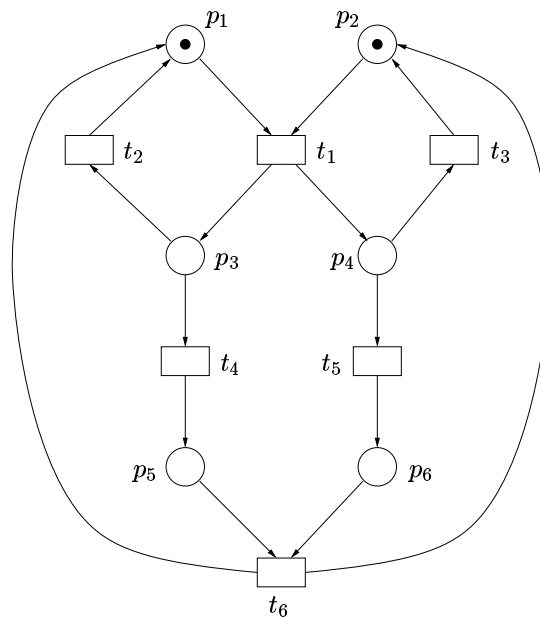


Fig. 53. An EN system M .

Thus, a process does not describe a complete run of the system (which can be infinite), but rather an initial finite part of it.

Note that our processes are themselves nets. These nets record the *occurrences of events* together with the *occurrences of the conditions that belong to these events*. Moreover, processes are nets of a special kind, viz. process nets: (1) if an event occurs, a possible conflict is resolved, and (2) different occurrences of the same condition and different occurrences of the same event are recorded by different copies of the corresponding condition and event, respectively. Therefore all places are unbranched and no cycles occur. The flow relation



Fig. 54. A process of the EN system M of Fig. 53.

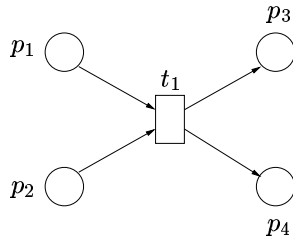


Fig. 55. A process that extends the process of Fig. 54.

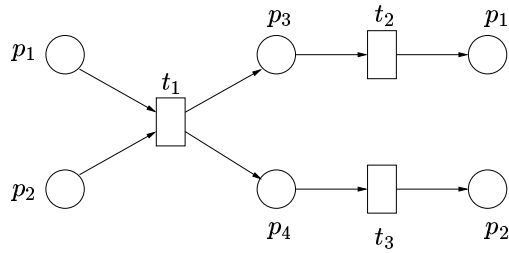


Fig. 56. A process that extends the process of Fig. 55.

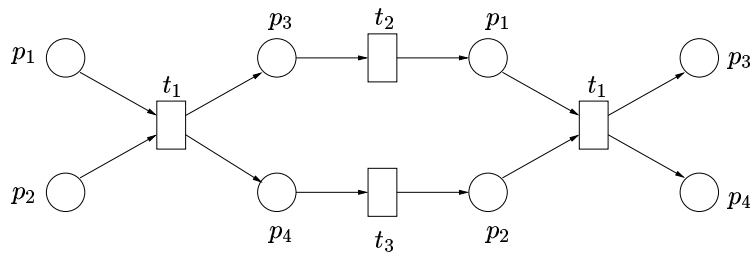


Fig. 57. A process that extends the process of Fig. 56.

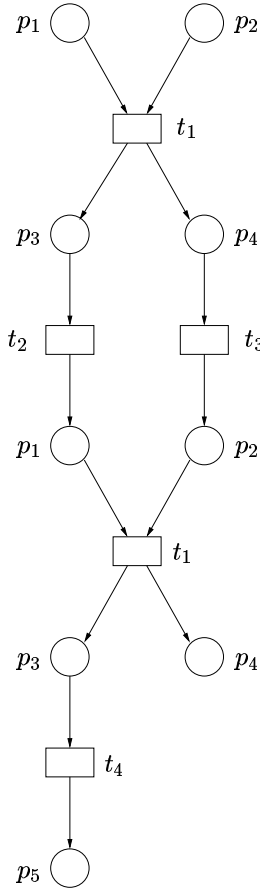


Fig. 58. A process of the EN system M of Fig. 53, leading from $\{p_1, p_2\}$ to $\{p_4, p_5\}$.

of the net represents the causal partial order between the events and conditions (see Lemma 74).

It is important to note that our processes only record that conditions hold, not that conditions do *not* hold. Hence a process faithfully describes a run of the system under consideration *only if* this system is contact-free, i.e., we need not record which conditions do *not* hold when we record the occurrence of an event. This is the reason for restricting our attention to processes of contact-free EN systems. Fortunately, Theorem 60 says that for every EN system there exists a configuration equivalent contact-free EN system. Hence, without loss of generality, we can assume contact-freeness in our study of the behaviour of EN systems. More specifically, when dealing with an EN system M that is not contact-free, if we want to study its behaviour through its processes, then we will

study the processes of a contact-free EN system that is configuration equivalent with M (e.g., the system that is obtained by the complement construction in the proof of Theorem 57).

The above considerations lead to the formal notion of a process in a contact-free EN system. However, before giving the formal definition, we need a number of auxiliary notions.

Definition 85. Let Σ_1 and Σ_2 be disjoint alphabets. A (Σ_1, Σ_2) -labelled net is a 5-tuple $N = (P, T, F, \phi_1, \phi_2)$, where (P, T, F) is a net (the *underlying net of N* , denoted by $\mathbf{und}(N)$), ϕ_1 is a function from P to Σ_1 (the *place labelling of N*), and ϕ_2 is a function from T to Σ_2 (the *transition labelling of N*).

N is also called a *labelled net*, and, if $\mathbf{und}(N)$ is a process net, then N is also called a (Σ_1, Σ_2) -labelled *process net* or simply a *labelled process net*. All notations and terminology concerning (process) nets carry over, through the underlying nets, to labelled (process) nets. We will also use the notation ϕ_{1N}, ϕ_{2N} for ϕ_1, ϕ_2 , respectively.

To compare labelled nets we need the following notion of isomorphism, which expresses the fact that the identity of the places, the transitions, and the labels is irrelevant.

Definition 86. Let $N = (P, T, F, \phi_1, \phi_2)$ and $N' = (P', T', F', \phi'_1, \phi'_2)$ be two (Σ_1, Σ_2) -, respectively (Σ'_1, Σ'_2) -labelled nets. Then N and N' are *isomorphic*, denoted by $N \equiv N'$, if there exist bijections $\alpha : \Sigma_1 \rightarrow \Sigma'_1$, $\beta : \Sigma_2 \rightarrow \Sigma'_2$, $\gamma : P \rightarrow P'$, and $\delta : T \rightarrow T'$, such that:

- (1) $\mathbf{und}(N) \equiv_{\delta}^{\gamma} \mathbf{und}(N')$,
- (2) for all $p \in P$, $\phi'_1(\gamma(p)) = \alpha(\phi_1(p))$, and
- (3) for all $t \in T$, $\phi'_2(\delta(t)) = \beta(\phi_2(t))$.

For isomorphic N and N' as above, we also say that N and N' are (α, β) -*isomorphic*, denoted by $N \equiv_{\beta}^{\alpha} N'$.

Condition (1) above means that the underlying nets are isomorphic, condition (2) means that corresponding places (via γ) have corresponding labels (via α), and condition (3) means that corresponding transitions (via δ) also have corresponding labels (via β).

This notion of isomorphism between labelled nets can naturally be extended to isomorphism between sets of labelled nets in the following way.

Definition 87. Let \mathcal{P} and \mathcal{P}' be two sets of (Σ_1, Σ_2) -, respectively (Σ'_1, Σ'_2) -labelled nets. Then \mathcal{P} and \mathcal{P}' are *isomorphic*, denoted by $\mathcal{P} \equiv \mathcal{P}'$, if there exist bijections $\alpha : \Sigma_1 \rightarrow \Sigma'_1$ and $\beta : \Sigma_2 \rightarrow \Sigma'_2$, such that

- (1) for every $N \in \mathcal{P}$ there exists $N' \in \mathcal{P}'$ such that $N \equiv_{\beta}^{\alpha} N'$, and
- (2) for every $N' \in \mathcal{P}'$ there exists $N \in \mathcal{P}$ such that $N \equiv_{\beta}^{\alpha} N'$.

For isomorphic \mathcal{P} and \mathcal{P}' as above, we also say that \mathcal{P} and \mathcal{P}' are (α, β) -*isomorphic*, denoted by $\mathcal{P} \equiv_{\beta}^{\alpha} \mathcal{P}'$.

Condition (1) means that every net in \mathcal{P} is isomorphic with a net in \mathcal{P}' and condition (2) means that, the other way around, every net in \mathcal{P}' is isomorphic with a net in \mathcal{P} . Note that the isomorphisms from (1) and (2) always use the same, fixed a priori, bijections α and β between the alphabets.

We now present the formal definition of the notion of a process. Recall from Section 3 that $f \upharpoonright B$ denotes the restriction of function $f : A \rightarrow A'$ to the set $B \subseteq A$. The requirement that $f \upharpoonright B$ is injective thus means that for all $b_1, b_2 \in B$, if $b_1 \neq b_2$ then $f(b_1) \neq f(b_2)$.

Definition 88. Let $N = (P_N, T_N, F_N, \phi_1, \phi_2)$ be a (Σ_1, Σ_2) -labelled process net and let $M = (P, T, F, C_{in})$ be a contact-free EN system.

Then N is a *process of M* if

- (1) $\Sigma_1 = P$ and $\Sigma_2 = \mathbf{use}(T)$,
- (2) $\phi_1 \upharpoonright {}^\circ N$ is injective,
- (3) $\phi_1({}^\circ N) = C_{in}$,
- (4) for every $t \in T_N$, $\phi_1 \upharpoonright \bullet t$ is injective and $\phi_1 \upharpoonright t^\bullet$ is injective, and
- (5) for every $t \in T_N$, $\phi_1(\bullet t) = \bullet(\phi_2(t))$ and $\phi_1(t^\bullet) = (\phi_2(t))^\bullet$.

From now on, for the sake of simplicity, we will also write ϕ instead of ϕ_1 and ϕ_2 , when the subscript is clear from the context.

For a contact-free EN system M , $\text{PROC}(M)$ denotes the set of all processes of M .

Requirement (1) above says that the places of the process net are labelled with the places of the system, and the transitions of the process net are labelled with the useful transitions of the system. Requirements (2) and (3) say that, via the labelling ϕ , the minimal places of the process net N are in one-to-one correspondence with the initial configuration of the system. The non-minimal places of N represent conditions that are set by the occurrence of some event. Thus, ${}^\circ N$ faithfully records the conditions that hold initially. Requirements (4) and (5) say that, via the labelling ϕ , the places in the input- and output-set of a transition s of the process net are in one-to-one correspondence with the places in the input- and output-set, respectively, of a transition t of the system. This means that s is indeed a faithful record of the occurrence of the event $t = \phi(s)$. Note that requirement (5) implies that for all $x, y \in X_N$, if $x F_N y$ then $\phi(x) F \phi(y)$.

Example 27. (1) Let M be the (contact-free) EN system of Fig. 49. A process N of M is drawn in Fig. 59 (the underlying process net is the one of Fig. 52). Note that the symbols next to the places and transitions of N are labels, i.e., places and transitions of M (the identities of the places and transitions of N are not given in the figure). The process N leads from the initial configuration $\{p_1, b_f, c_1\}$ of M to the configuration $\{p_2, b_e, c_2\}$ of M .

(2) A process of the EN system of Fig. 5 (mutual exclusion) is drawn in Fig. 60. It leads from the initial configuration $\{w_1, p, w_2\}$ to the configuration $\{c_1, w_2\}$.

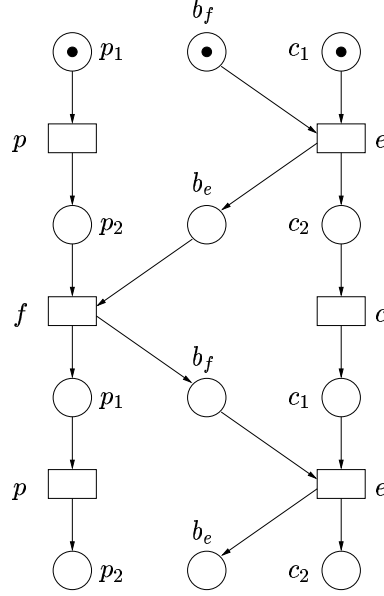


Fig. 59. A process of the producer/consumer system of Fig. 49.

We will now show that the firing of transitions in a process is mapped (by ϕ) to the firing of transitions in the system. Hence, playing the token game in a process corresponds to playing the token game in the system. To prove this, we need the following lemma.

Lemma 89. *Let $M = (P, T, F, C_{in})$ be a contact-free EN system and let $N = (P_N, T_N, F_N, \phi_1, \phi_2)$ be a process of M . Let $C, D \in \mathbb{C}_N$ and $t \in T_N$. If $\phi \upharpoonright C$ is injective, $\phi(C) \in \mathbb{C}_M$, and $C[t]_N D$, then $\phi \upharpoonright D$ is injective and $\phi(C)[\phi(t)]_M \phi(D)$.*

Proof. From $\bullet t \subseteq C$ it follows that $\phi(\bullet t) \subseteq \phi(C)$ and hence, by requirement (5) of Definition 88, that $\bullet \phi(t) \subseteq \phi(C)$. Since M is contact-free, $\phi(t) \bullet \cap \phi(C) = \emptyset$ and hence, again by requirement (5), $\phi(t) \bullet \cap \phi(C) = \emptyset$. This, and the fact that $\phi \upharpoonright t^\bullet$ is injective (requirement (4)), implies that $\phi \upharpoonright (C \cup D)$ is injective. Consequently $\phi(C - D) = \phi(C) - \phi(D)$ and $\phi(D - C) = \phi(D) - \phi(C)$. By Lemma 7, $\bullet t = C - D$ and $t^\bullet = D - C$. Hence $\bullet \phi(t) = \phi(\bullet t) = \phi(C - D) = \phi(C) - \phi(D)$ and similarly $\phi(t) \bullet = \phi(D) - \phi(C)$. Lemma 7 now implies that $\phi(C)[\phi(t)]_M \phi(D)$. \square

Theorem 90. *Let $M = (P, T, F, C_{in})$ be a contact-free EN system and let $N = (P_N, T_N, F_N, \phi_1, \phi_2)$ be a process of M .*

- (1) *For every $C \in \mathbb{C}_N$, $\phi \upharpoonright C$ is injective and $\phi(C) \in \mathbb{C}_M$.*
- (2) *For every $C, D \in \mathbb{C}_N$ and $t \in T_N$, if $C[t]_N D$ then $\phi(C)[\phi(t)]_M \phi(D)$.*

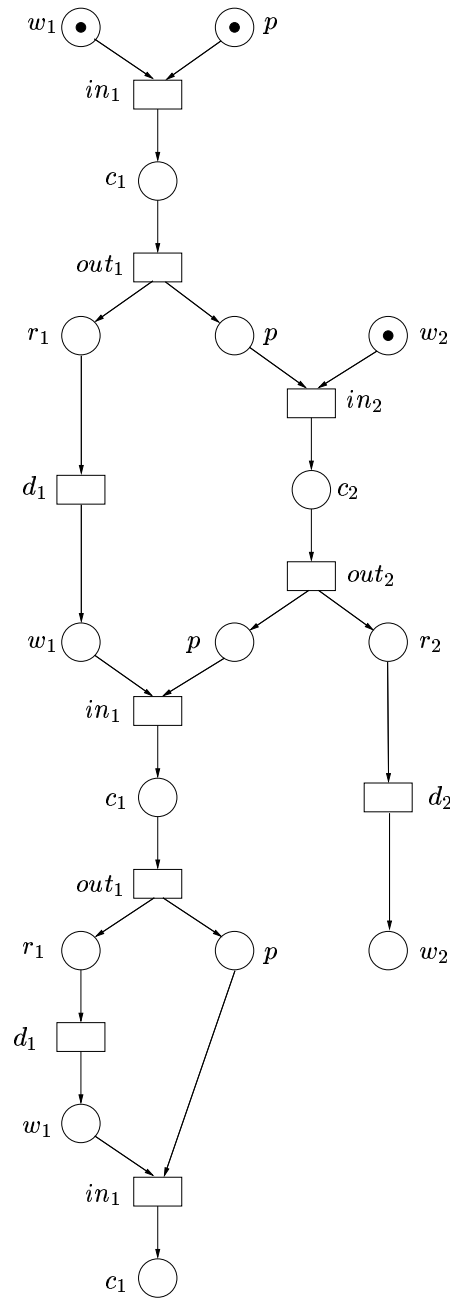


Fig. 60. A process of the mutual exclusion system of Fig. 5.

Proof. (1) can easily be proved by induction on C . The base of the induction (i.e., $C = {}^\circ N$) directly follows from requirements (2) and (3) of Definition 88, and the induction step follows from Lemma 89.

(2) directly follows from (1) and Lemma 89. \square

Theorem 91. *Let $M = (P, T, F, C_{in})$ be a contact-free EN system and let $N = (P_N, T_N, F_N, \phi_1, \phi_2)$ be a process of M .*

(1) *For every **co**-clique D of X_N , $\phi \upharpoonright D$ is injective.*

(2) *For every $C, D \in \mathbb{C}_N$ and $U \subseteq T_N$, if $C[U]_N D$ then $\phi(C)[\phi(U)]_M \phi(D)$.*

(3) *For every **co**-clique $U \subseteq T_N$ there exist $C, D \in \mathbb{C}_M$ such that $C[\phi(U)]_M D$.*

Proof. (1) We have to show that for every two distinct elements x and y of X_N , if x **co** y then $\phi(x) \neq \phi(y)$. This is obvious if one of the two is a place and the other a transition. If $\{x, y\} \subseteq P_N$, then by Lemma 77 there exists a slice $C \in \mathbb{C}_N$ with $\{x, y\} \subseteq C$, and Theorem 90(1) then implies that $\phi \upharpoonright \{x, y\}$ is injective. If $\{x, y\} \subseteq T_N$, then Lemma 78(1) implies that $\bullet x \cup \bullet y$ is a **co**-clique. According to the previous case, $\phi \upharpoonright (\bullet x \cup \bullet y)$ is injective. Definition 73(2) now implies that $\phi(\bullet x) \neq \phi(\bullet y)$. Hence $\bullet \phi(x) \neq \bullet \phi(y)$ and thus $\phi(x) \neq \phi(y)$.

(2) Assume that $C[U]_N D$. By Theorem 81(2), U is a **co**-clique. Then (1) implies that $\phi \upharpoonright U$ is injective. This means that the set of orderings of the elements of $\phi(U)$ equals $\{(\phi(t_1), \dots, \phi(t_n)) \mid (t_1, \dots, t_n) \text{ is an ordering of the elements of } U\}$. This, together with Theorems 20(2) and 90(2), implies that $\phi(C)[\phi(U)]_M \phi(D)$.

(3) If $U \subseteq T_N$ is a **co**-clique, then there exists $C \in \mathbb{C}_N$ with U **con** C by Theorem 81(2). Theorem 90(1) implies that $\phi(C) \in \mathbb{C}_M$. Now apply (2). \square

There is a clear connection between the firing sequences of an EN system and the firing sequences of its processes. Theorem 90 implies that every firing sequence of a process N (of an EN system M) is mapped to a firing sequence of M by ϕ . We will now show that this also holds the other way around: for every firing sequence of M there exists a firing sequence of a process N of M which is mapped to it by ϕ ; moreover, we can guarantee that this is a “complete” firing sequence of N , i.e., a firing sequence from ${}^\circ N$ to N° (note that, by Theorem 79, every process net has such a complete firing sequence). This resembles the construction which we presented as an example at the beginning of this subsection (in Figs. 54 to 58).

Theorem 92. *Let $M = (P, T, F, C_{in})$ be a contact-free EN system, let t_1, \dots, t_n be transitions in T , and let $C \subseteq P$. Then $C_{in}[t_1 \dots t_n]_M C$ iff there exists a process $N = (P_N, T_N, F_N, \phi_1, \phi_2)$ of M and there exist transitions s_1, \dots, s_n in T_N such that*

(1) $\phi(s_i) = t_i$ for $1 \leq i \leq n$,

(2) $\phi(N^\circ) = C$, and

(3) ${}^\circ N[s_1 \dots s_n]_N N^\circ$.

Proof. (If) This follows from Definition 88(3) and Theorem 90(2).

(Only-if) The proof is by induction on n .

For $n = 0$ we have to show the existence of a process N with ${}^\circ N = N^\circ$. If $C_{in} = \{q_1, \dots, q_m\}$, then such a process N is defined by: $P_N = \{p_1, \dots, p_m\}$, $T_N = \emptyset$, $F_N = \emptyset$, $\phi_2 = \emptyset$, and $\phi_1(p_i) = q_i$ for $1 \leq i \leq m$.

Now assume that $C_{in}[t_1 \dots t_n]C[t]D$ and assume (the induction hypothesis) that there exists a process N satisfying requirements (1-3). Let $\bullet t = \{q_1, \dots, q_k\}$ and $t^\bullet = \{q'_1, \dots, q'_m\}$. Since $C = \phi(N^\circ)$ and $\bullet t \subseteq C$, there are (unique) places $p_1, \dots, p_k \in N^\circ$ such that $\phi(p_i) = q_i$ for $1 \leq i \leq k$. Now take an $s \notin T_N$ and $p'_1, \dots, p'_m \notin P_N$. We extend N by adding transition s and places p'_1, \dots, p'_m , in such a way that $\bullet s = \{p_1, \dots, p_k\}$, $s^\bullet = \{p'_1, \dots, p'_m\}$, $\phi(s) = t$, and $\phi(p'_i) = q'_i$ for $1 \leq i \leq m$. In this way a new process is obtained that satisfies requirements (1-3) for $C_{in}[t_1 \dots t_n t]D$ (verify this). \square

It is also straightforward to prove that, for given firing sequence $t_1 \dots t_n$, the process N is unique (modulo isomorphism, i.e., modulo \equiv_β^α , where α is the identity on P_M and β the identity on $\mathbf{use}(T_M)$). Since every process has a complete firing sequence, this implies that *every* process of M is obtained in this way from a firing sequence of M (cf. [BesDev87], Theorem 3.5.3 of [BesFer88], and Section 3 of [NieRozThi90]). Note, however, that different firing sequences $t_1 \dots t_n$ can lead to the same process N . The resulting equivalence relation between firing sequences is studied in Section 7.2. The uniqueness of N will also follow from later results (see the remark after Theorem 122).

Note that Theorem 92 implies that, for every contact-free EN system M , $\text{PROC}(M) \neq \emptyset$ (since λ is always a firing sequence of M).

Example 28. Consider the EN system M of Fig. 49 and the process N of M of Fig. 59. Examples of firing sequences of M that correspond to process N in the way indicated in Theorem 92 are: *pefcep* and *ecpffe*.

We now prove the converse of Theorem 90(2).

Theorem 93. *Let M be a contact-free EN system. Let $C, D \in \mathbb{C}_M$ and $t \in T_M$. If $C[t]_M D$, then there exists a process N of M and there exist $C', D' \in \mathbb{C}_N$ and $s \in T_N$, such that $C'[s]_N D'$, $\phi_N(C') = C$, $\phi_N(s) = t$, and $\phi_N(D') = D$.*

Proof. There is a firing sequence x such that $C_{in}[x]_M C$. If we now apply Theorem 92 to $C_{in}[xt]_M D$, then we obtain a process N of M and a firing sequence ys (with $y \in T_N^*$ and $s \in T_N$) such that $\phi(y) = x$, $\phi(s) = t$, $\phi(N^\circ) = D$, and ${}^\circ N[ys]_N N^\circ$ (where $\phi = \phi_N$). Assume that ${}^\circ N[y]_N C'[s]_N N^\circ$ and let $D' = N^\circ$. By Theorem 90, $\phi({}^\circ N)[\phi(y)]_M \phi(C')$ holds, i.e., $C_{in}[x]_M \phi(C')$. Hence $\phi(C') = C$. \square

Analogously the following converse of Theorem 91(2,3) can be proved. The details of the proof of (1) are left to the reader; (2) follows immediately from (1) and Theorem 81(2).

Theorem 94. *Let M be a contact-free EN system. Let $C, D \in \mathbb{C}_M$ and let $U \subseteq T_M$. If $C[U]_M D$, then:*

- (1) *there exists a process N of M and there exist $C', D' \in \mathbb{C}_N$ and $V \subseteq T_N$, such that $C'[V]_N D'$, $\phi_N(C') = C$, $\phi_N(V) = U$, and $\phi_N(D') = D$,*
- (2) *there exist a process N of M and a **co**-clique $V \subseteq T_N$ such that $\phi_N(V) = U$.*

It would be natural to say that two EN systems are equivalent if they have isomorphic sets of processes (see Definition 87). The next result shows that two (reduced) EN systems are equivalent in this sense iff they are isomorphic. Thus, as discussed in the next subsection, this equivalence relation does not capture equivalent behaviour of EN systems.

Theorem 95. *Let M and M' be two contact-free reduced EN systems. Then $PROC(M) \equiv PROC(M')$ iff $M \equiv M'$.*

Proof. Let $M = (P, T, F, C_{in})$ and $M' = (P', T', F', C'_{in})$. It is easy to see that if $M \equiv_{\beta}^{\alpha} M'$, then $PROC(M) \equiv_{\beta}^{\alpha} PROC(M')$.

* Now assume that $PROC(M) \equiv_{\beta}^{\alpha} PROC(M')$. Then α and β are bijections, $\alpha : P \rightarrow P'$ and $\beta : T \rightarrow T'$. It now suffices to show that (1) $\alpha(C_{in}) = C'_{in}$ and that (2) for every $t \in T$, $\alpha(\bullet t) = \bullet \beta(t)$ and $\alpha(t\bullet) = \beta(t)\bullet$.

(1) Consider a process N of M (this is possible because $PROC(M) \neq \emptyset$) and consider a process N' of M' that is (α, β) -isomorphic with N . Then $\circ N$ corresponds to $\circ N'$ and hence $\alpha(\phi_N(\circ N)) = \phi_{N'}(\circ N')$. Since $\phi_N(\circ N) = C_{in}$ and $\phi_{N'}(\circ N') = C'_{in}$, this means that $\alpha(C_{in}) = C'_{in}$.

(2) Since M is reduced, there exist $C, D \in \mathbb{C}_M$ with $C[t]_M D$. Then, by Theorem 93, there exist a process N of M and an $s \in T_N$ such that $\phi_N(s) = t$. Let N' be a process of M' that is (α, β) -isomorphic with N and assume that $s' \in T_{N'}$ corresponds to s . Then $\phi_{N'}(s') = \beta(t)$. Furthermore, $\bullet s$ corresponds to $\bullet(s')$ and $s\bullet$ to $(s')\bullet$, and hence $\alpha(\phi_N(\bullet s)) = \phi_{N'}(\bullet(s'))$ and $\alpha(\phi_N(s\bullet)) = \phi_{N'}((s')\bullet)$. Now $\phi_N(\bullet s) = \bullet \phi_N(s) = \bullet t$, $\phi_{N'}(\bullet(s')) = \bullet \phi_{N'}(s') = \bullet \beta(t)$, $\phi_N(s\bullet) = \phi_N(s)\bullet = t\bullet$, and $\phi_{N'}((s')\bullet) = \phi_{N'}(s')\bullet = \beta(t)\bullet$, and hence $\alpha(\bullet t) = \bullet \beta(t)$ and $\alpha(t\bullet) = \beta(t)\bullet$. \square

6.4 Pruned Contracted Processes

We can interpret the set $PROC(M)$ of processes of an EN system M as the behaviour of M : it is the set of all concurrent runs of M . However, by Theorem 95, this definition of the behaviour of an EN system is too strong, since the system M is uniquely determined by $PROC(M)$, modulo isomorphism, which means that the behaviour of the EN system would be identified with its structure! In this way it would be impossible to transform systems while preserving their behaviour. The intuitive reason why Theorem 95 holds, is that in a process both the events and the conditions are recorded: in that way we are able to read the flow relation of the system from the flow relation of its processes and their labels (see Definition 88(4,5)). However, since we are more interested in the events

than in the conditions of the system when defining behaviour (see Section 5.1), we will, in this subsection, remove the conditions from every process N . In this way the set T_N of recorded events remains, together with their causal order (the partial order F_N^+ , restricted to T_N). This can be considered as a “labelled partially ordered set”, with labels in T_M (or, more precisely, in $\mathbf{use}(T_M)$). Then we will define the behaviour of M as the set of all labelled partially ordered sets obtained in this way. With this definition of the behaviour of M , the system M is no longer uniquely determined, i.e., Theorem 95 no longer holds. Note that such partially ordered sets of occurrences of events are similar to firing sequences, which are linearly ordered sets of occurrences of events.

Modulo isomorphism, labelled partially ordered sets (with labels in an alphabet Σ) are also called partially ordered multisets (with elements in Σ) or *pom-sets*, see, e.g., [Pra86]. Here, we model labelled partially ordered sets by node-labelled acyclic graphs, which are easier to compare to processes. There is a clear connection between partial orders and acyclic graphs. A partially ordered set (A, ρ) with $\rho \subseteq A \times A$ is a (special kind of) acyclic directed graph. The other way around it is clear that for every acyclic directed graph (V, Γ) , (V, Γ^+) is a partially ordered set (cf. Lemma 74). We will say that the graph (V, Γ) represents the partially ordered set (V, Γ^+) . In general a partially ordered set can thus be represented by several acyclic graphs.

To begin with, we recall several notions concerning node-labelled directed graphs which, for the sake of simplicity, we will simply call labelled graphs. Note that we have so far only considered (initialized) edge-labelled graphs (in particular configuration graphs). We have also considered labelled nets (in particular processes) in the previous section. When we view a net as a graph, labelled nets can be viewed as node-labelled graphs. The following definitions are thus analogous to those for labelled nets.

Definition 96. Let Σ be an alphabet. A (Σ) -labelled graph is a quadruple $G = (V, \Gamma, \Sigma, \phi)$, where V is a finite set of *nodes*, $\Gamma \subseteq V \times V$ is a set of *edges*, and ϕ is a function from V to Σ (the *labelling of G*). G is *acyclic* if Γ^+ is irreflexive.

The components of G are also indicated by V_G , Γ_G , Σ_G , and ϕ_G .

To compare graphs labelled over distinct alphabets we need the following notion of isomorphism.

Definition 97. Let Σ and Σ' be two alphabets and let $G = (V, \Gamma, \Sigma, \phi)$ and $G' = (V', \Gamma', \Sigma', \phi')$ be two Σ -, respectively Σ' -labelled graphs. Then G and G' are *isomorphic*, denoted by $G \equiv G'$, if there exist bijections $\beta : \Sigma \rightarrow \Sigma'$ and $\delta : V \rightarrow V'$, such that

- (1) for all $v, w \in V$, $(v, w) \in \Gamma$ iff $(\delta(v), \delta(w)) \in \Gamma'$, and
- (2) for all $v \in V$, $\phi'(\delta(v)) = \beta(\phi(v))$.

For isomorphic G and G' as above, we also say that G and G' are β -*isomorphic*, denoted by $G \equiv_\beta G'$.

The above notion of isomorphism between labelled graphs can naturally be extended to isomorphism between sets of labelled graphs in the following way.

Definition 98. Let \mathcal{P} and \mathcal{P}' be two sets of Σ -, respectively Σ' -labelled graphs. Then \mathcal{P} and \mathcal{P}' are *isomorphic*, denoted by $\mathcal{P} \equiv \mathcal{P}'$, if there exists a bijection $\beta : \Sigma \rightarrow \Sigma'$, such that

- (1) for every $G \in \mathcal{P}$ there exists $G' \in \mathcal{P}'$ such that $G \equiv_{\beta} G'$, and
- (2) for every $G' \in \mathcal{P}'$ there exists $G \in \mathcal{P}$ such that $G \equiv_{\beta} G'$.

For isomorphic \mathcal{P} and \mathcal{P}' as above, we also say that \mathcal{P} and \mathcal{P}' are β -*isomorphic*, denoted by $\mathcal{P} \equiv_{\beta} \mathcal{P}'$. Note that the isomorphisms from (1) and (2) always use the same, fixed a priori, bijection β between the alphabets (cf. Definition 87).

A labelled graph $G = (V, \Gamma, \Sigma, \phi)$ for which (V, Γ) is a partially ordered set is also called a *labelled partially ordered set* or, shorter, a *labelled partial order*. To every acyclic labelled graph such a labelled partial order is naturally associated.

Definition 99. Let $G = (V, \Gamma, \Sigma, \phi)$ be an acyclic labelled graph. The *transitive closure* of G , denoted by $\mathbf{tra}(G)$, is the labelled graph $(V, \Gamma^+, \Sigma, \phi)$. We also say that G *represents* $\mathbf{tra}(G)$.

Note that $\mathbf{tra}(G)$ is a labelled partial order. In general a labelled partial order is represented by several labelled graphs. One of these graphs is the “smallest” (and is also known as the “Hasse diagram” of the partial order).

Definition 100. Let $G = (V, \Gamma, \Sigma, \phi)$ be an acyclic labelled graph. The *pruned version* of G , denoted by $\mathbf{pru}(G)$, is the labelled graph $(V, \Gamma', \Sigma, \phi)$ with $\Gamma' = \{(v, w) \in \Gamma \mid \neg \exists u \in V : (v, u) \in \Gamma^+ \text{ and } (u, w) \in \Gamma^+\}$.

This means that $\mathbf{pru}(G)$ is the graph that is obtained from G by removing all edges (v, w) such that there exists a path of length ≥ 2 from v to w in G , i.e., by removing the so-called *transitive* edges.

Example 29. Figure 61 shows a graph with its transitive closure and its pruned version. The labels have been omitted.

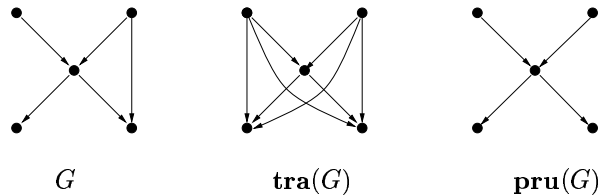


Fig. 61. A graph with its transitive closure and its pruned version.

We now recall the well-known fact that $\mathbf{tra}(G)$ is represented by $\mathbf{pru}(G)$, and that $\mathbf{pru}(G)$ is uniquely determined by $\mathbf{tra}(G)$.

Theorem 101. *For every acyclic labelled graph G , $\mathbf{tra}(\mathbf{pru}(G)) = \mathbf{tra}(G)$ and $\mathbf{pru}(\mathbf{tra}(G)) = \mathbf{pru}(G)$.*

Proof. Let $G = (V, \Gamma, \Sigma, \phi)$. We will prove the inclusion $\mathbf{tra}(G) \subseteq \mathbf{tra}(\mathbf{pru}(G))$; the other inclusions are immediate. We have to prove that, for all $v, w \in V$, if there is a path from v to w , then there is a path from v to w that uses only non-transitive edges. This is done by induction on the length $n(v, w)$ of the longest path from v to w . If $n(v, w) = 1$, then the longest path from v to w is the edge (v, w) . Hence that edge is not transitive. Now take $n(v, w) \geq 2$ and assume that the claim holds for all lengths $< n(v, w)$. Consider the longest path from v to w and take a node u on that path that is distinct from v and w . Then the subpath from v to u is also the longest path from v to u , and hence $n(v, u) < n(v, w)$. This implies that there is a path from v to u with non-transitive edges only. Likewise there is a path from u to w with non-transitive edges only. \square

This implies that two labelled partial orders are isomorphic iff their pruned versions are isomorphic.

Theorem 102. *Let Σ and Σ' be alphabets. Let G and G' be two acyclic Σ -, respectively Σ' -labelled graphs and let $\beta : \Sigma \rightarrow \Sigma'$ be a bijection. Then $\mathbf{tra}(G) \equiv_{\beta} \mathbf{tra}(G')$ iff $\mathbf{pru}(G) \equiv_{\beta} \mathbf{pru}(G')$.*

Proof. It is clear that, for labelled graphs G_1 and G_2 , if $G_1 \equiv_{\beta} G_2$, then $\mathbf{tra}(G_1) \equiv_{\beta} \mathbf{tra}(G_2)$ and $\mathbf{pru}(G_1) \equiv_{\beta} \mathbf{pru}(G_2)$. Hence, if $\mathbf{tra}(G) \equiv_{\beta} \mathbf{tra}(G')$, then $\mathbf{pru}(\mathbf{tra}(G)) \equiv_{\beta} \mathbf{pru}(\mathbf{tra}(G'))$ and thus $\mathbf{pru}(G) \equiv_{\beta} \mathbf{pru}(G')$. Likewise in the other direction. \square

We now return to the processes of an EN system M . As observed before, we are interested in the labelled partial order $(T_N, F_N^+ \cap (T_N \times T_N), T_M, \phi_{2N})$ for a process N of M . There is an easy way to construct a labelled graph that represents this labelled partial order: we remove the places from N and replace them by edges. When we then remove the transitive edges from this graph we obtain a unique representation of the labelled partial order. The precise formulation of this is given in the following definition, cf. Fig. 19.

Definition 103. Let $N = (P, T, F, \phi_1, \phi_2)$ be an acyclic (Σ_1, Σ_2) -labelled net.

- (1) The *contracted version* of N , denoted by $\mathbf{ctr}(N)$, is the labelled graph $(T, \Gamma, \Sigma_2, \phi_2)$ such that, for all $s, t \in T$, $(s, t) \in \Gamma$ iff $s^{\bullet} \cap t^{\bullet} \neq \emptyset$.
- (2) The *pruned contracted version* of N is the labelled graph $\mathbf{pru}(\mathbf{ctr}(N))$.

It is easy to see that $\mathbf{ctr}(N)$ indeed represents the labelled partial order $(T_N, F_N^+ \cap (T_N \times T_N), \Sigma_2, \phi_{2N})$. Then, by Theorem 101, the same holds for $\mathbf{pru}(\mathbf{ctr}(N))$.

Lemma 104. *Let $N = (P, T, F, \phi_1, \phi_2)$ be an acyclic (Σ_1, Σ_2) -labelled net and let $\mathbf{pru}(\mathbf{ctr}(N)) = (T, \Gamma, \Sigma_2, \phi_2)$. Then, for all $s, t \in T$, $(s, t) \in \Gamma^+$ iff $(s, t) \in F^+$.*

If N is a process of an EN system M , then $\mathbf{ctr}(N)$ is called a *contracted process of M* and $\mathbf{pru}(\mathbf{ctr}(N))$ a *pruned contracted process of M* . For a contact-free EN system M we denote by $\mathbf{LPO}(M)$ the set of all pruned contracted processes of M (where LPO stands for Labelled Partial Orders). Hence

$$\mathbf{LPO}(M) = \{\mathbf{pru}(\mathbf{ctr}(N)) \mid N \in \mathbf{PROC}(M)\}.$$

Example 30. (1) Let M be the EN system of Fig. 49 (the producer/consumer system) and let N be the process of M in Fig. 59. Then $\mathbf{ctr}(N) = \mathbf{pru}(\mathbf{ctr}(N))$ is given in Fig. 62. A larger pruned contracted process of M is drawn in Fig. 63.

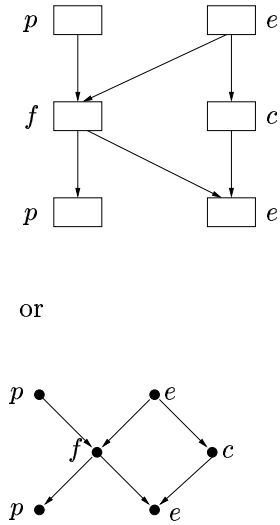


Fig. 62. The (pruned) contracted version of the process of Fig. 59.

(2) Let M be the EN system of Fig. 5 (the mutual exclusion system) and let N be the process of M in Fig. 60. Then $\mathbf{ctr}(N)$ and $\mathbf{pru}(\mathbf{ctr}(N))$ are given in Fig. 64.

We can now call two EN systems equivalent if their sets of pruned contracted processes are isomorphic (see Definition 98). The behaviour of an EN system M is thus defined as $\mathbf{LPO}(M)$, modulo isomorphism.

Definition 105. Two contact-free EN systems M and M' are *lpo-equivalent* if $\mathbf{LPO}(M) \equiv \mathbf{LPO}(M')$.

An intuitive way to view a concurrent system, is to see it as a system with its actions not linearly ordered but partially ordered, where a linear order can be

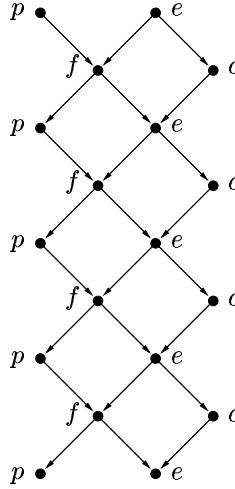


Fig. 63. Another pruned contracted process of the producer/consumer system of Fig. 49.

interpreted as an ordering in time, whereas a partial order can be interpreted as showing the causal relationships between the actions. This means that the above definition of behaviour is intuitively attractive and perhaps the most natural one.

We now show (as mentioned already in the introduction of this subsection) that, in contrast with ordinary processes (see Theorem 95), the set of pruned contracted processes does not uniquely determine an EN system (modulo isomorphism).

Theorem 106. *There exist reduced contact-free EN systems M and M' such that $LPO(M) \equiv LPO(M')$ but $M \equiv M'$ does not hold.*

Proof. The EN systems M and M' of Figs. 33 and 35 have isomorphic sets of contracted processes, and those contracted processes are already pruned, i.e., contain no transitive edges. Hence $LPO(M) \equiv LPO(M')$. Note that in this case the contracted processes correspond to $FS(M) = FS(M') = \{\lambda, a, c, e, ab, cd, ef, abd\}$, see Example 15.

Another example of two lpo-equivalent EN systems M and M' that are not isomorphic is depicted in Fig. 65. The sets of contracted processes of M and M' are not isomorphic, but the sets of pruned contracted processes are. \square

In the remainder of this subsection we will prove that the function **prctr** : $PROC(M) \rightarrow LPO(M)$, defined by **prctr**(N) = **pru**(**ctr**(N)), is a bijection (modulo isomorphism); i.e., though certain information is lost by the function **prctr**, it does not identify distinct processes. Hence $LPO(M)$ is still a “faithful” modelling of the behaviour of M . This is formulated as follows.

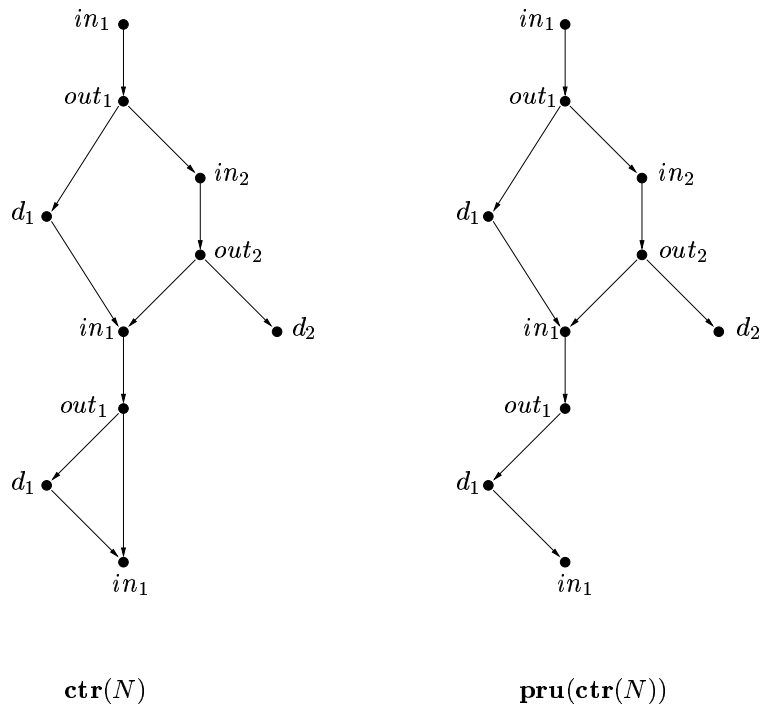


Fig. 64. The contracted process and the pruned contracted process corresponding to the process of Fig. 60.

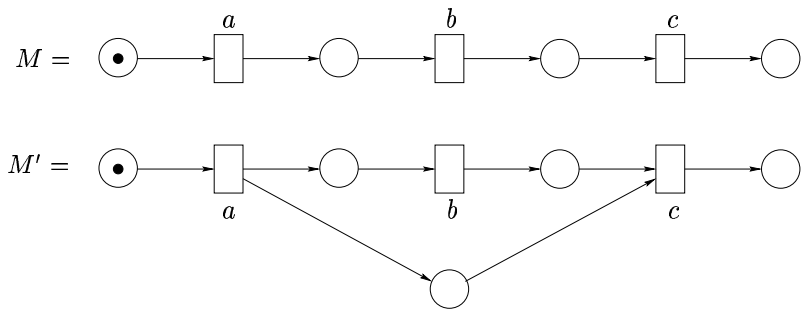


Fig. 65. Two non-isomorphic lpo-equivalent EN systems.

Theorem 107. *Let M be a contact-free EN system and let N, N' be two processes of M . Let α be the identity on P_M and β the identity on $\mathbf{use}(T_M)$. Then $N \equiv_{\beta}^{\alpha} N'$ iff $\mathbf{pru}(\mathbf{ctr}(N)) \equiv_{\beta} \mathbf{pru}(\mathbf{ctr}(N'))$.*

This implies that if M and M' are lpo-equivalent EN systems, then there is a bijection $f : \text{PROC}(M) \rightarrow \text{PROC}(M')$ between their processes such that $\mathbf{pru}(\mathbf{ctr}(N)) \equiv_{\beta} \mathbf{pru}(\mathbf{ctr}(f(N)))$ for every process N of M .

We now turn to the *proof of Theorem 107*. The Only-if direction of the proof is clear (both \mathbf{ctr} and \mathbf{pru} preserve isomorphism). A simple proof of the implication in the other direction will be given after Theorem 115. The proof that follows is, however, more transparent, because it shows how to reconstruct the process N directly from the pruned contracted process $\mathbf{pru}(\mathbf{ctr}(N))$, i.e., from $(T_N, F_N^+ \cap (T_N \times T_N), T_M, \phi_{2N})$. This technique will be useful in Theorem 149.

* To explain the construction we need two lemmas.

Lemma 108. *Let $N = (P, T, F, \phi_1, \phi_2)$ be a process of a contact-free EN system M and let $q_1, q_2 \in P$.*

(1) *For all $s_1, s_2 \in T$,*

if $s_1 \in \bullet q_1, s_2 \in q_2 \bullet, (s_1, s_2) \in F^+$, and $\phi(q_1) = \phi(q_2)$, then $(q_1, q_2) \in F^$.*

(2) *If $\bullet q_1 = \emptyset$ and $\phi(q_1) = \phi(q_2)$, then $(q_1, q_2) \in F^*$.*

Proof. (1) If $q_1 \neq q_2$, then, by Theorem 91(1), $q_1 \mathbf{co} q_2$ does not hold. Hence $q_1 \mathbf{li} q_2$. Then $(q_1, q_2) \in F^+$, because $(q_2, q_1) \in F^+$ implies that $(s_2, s_1) \in F^*$ and hence that N is cyclic. The proof of (2) is analogous. \square

* In the next lemma we show how the position of the places of N can be determined from the labelled partial order $(T_N, F_N^+ \cap (T_N \times T_N), T_M, \phi_{2N})$.

Lemma 109. *Let $N = (P, T, F, \phi_1, \phi_2)$ be a process of a contact-free EN system M and let $p \in P_M$.*

(1) *For all $s_1, s_2 \in T$:*

$\exists q \in P$ with $s_1 \in \bullet q, s_2 \in q \bullet$, and $\phi(q) = p$

iff

$\phi(s_1) \in \bullet p, \phi(s_2) \in p \bullet, (s_1, s_2) \in F^+$, and

$\neg \exists s \in T : \phi(s) \in p \bullet$ and $(s_1, s) \in F^+$ and $(s, s_2) \in F^+$.

(2) *For all $s_2 \in T$:*

$\exists q \in P$ with $\bullet q = \emptyset, s_2 \in q \bullet$, and $\phi(q) = p$

iff

$p \in (C_{in})_M, \phi(s_2) \in p \bullet$, and

$\neg \exists s \in T : \phi(s) \in p \bullet$ and $(s, s_2) \in F^+$.

Proof. First note that the right-hand side of the equivalence in (1) means that s_2 is minimal (with respect to the partial order F^+) in the set of all $s \in T$ with $\phi(s) \in p \bullet$ and $(s_1, s) \in F^+$ (and, in fact, it is the minimum of that set). Similarly for (2), s_2 is minimal in the set of all $s \in T$ with $\phi(s) \in p \bullet$.

(1) (Only-if) Obviously $(s_1, s_2) \in F^+$ and, by Definition 88(5), $\phi(s_1) \in \bullet p$ and $\phi(s_2) \in p^\bullet$. Now assume that $\phi(s) \in p^\bullet$, $(s_1, s) \in F^+$, and $(s, s_2) \in F^+$. By Definition 88(5) there exists $q' \in P$ with $s \in (q')^\bullet$ and $\phi(q') = p$. Hence, by Lemma 108(1), $(q, q') \in F^*$. Then, since $q^\bullet = \{s_2\}$, also $(s_2, s) \in F^*$. This contradicts the fact that N is acyclic.

(If) Since $\phi(s_1) \in \bullet p$, there exists $q_1 \in P$ with $s_1 \in \bullet q_1$ and $\phi(q_1) = p$. Likewise there exists q_2 with $s_2 \in q_2^\bullet$ and $\phi(q_2) = p$. We prove that $q_1 = q_2$ (which is then the required q). By Lemma 108(1), $(q_1, q_2) \in F^*$. Suppose that $q_1 \neq q_2$. Then there is a transition s such that $q_1 F s F^+ q_2$. Clearly $\phi(s) \in p^\bullet$, $(s_1, s) \in F^+$, and $(s, s_2) \in F^+$, contradicting the assumption.

(2) can be proved analogously, using Lemma 108(2) and Definition 88(3). \square

* We now prove the If-part of Theorem 107. Let $M = (P_M, \mathbf{use}(T_M), F_M, C_{in})$. To show that the function $\mathbf{pructr} : \text{PROC}(M) \rightarrow \text{LPO}(M)$ is a bijection modulo isomorphism, it suffices to define a function $\mathbf{proc} : \text{LPO}(M) \rightarrow \text{PROC}(M)$ such that

- (1) for all $G, G' \in \text{LPO}(M)$, if $G \equiv_\beta G'$, then $\mathbf{proc}(G) \equiv_\beta^\alpha \mathbf{proc}(G')$, and
- (2) for all $N \in \text{PROC}(M)$, $\mathbf{proc}(\mathbf{pructr}(N)) \equiv_\beta^\alpha N$.

It is straightforward to define the function \mathbf{proc} , on the basis of Lemma 109, as follows. Let $G = (T, \Gamma, \mathbf{use}(T_M), \phi)$ be an element of $\text{LPO}(M)$. Then $\mathbf{proc}(G)$ is the $(P_M, \mathbf{use}(T_M))$ -labelled net (P, T, F, ϕ_1, ϕ) , with P , ϕ_1 , and F defined as follows. P consists of newly created places of two types: all places q_p with $p \in (C_{in})_M$, and all places $q_{s,p}$ with $s \in T$ and $p \in \phi(s)^\bullet$. Their labels are defined by $\phi_1(q_p) = \phi_1(q_{s,p}) = p$. Intuitively, the places q_p form the initial slice of $\mathbf{proc}(G)$, and the places $q_{s,p}$ form the output-set of the transition s in $\mathbf{proc}(G)$. With this in mind we define $F \cap (T \times P) = \{(s, q_{s,p}) \mid s \in T, p \in \phi(s)^\bullet\}$. The remaining part of F is defined on the basis of Lemma 109 (and Lemma 104): $F \cap (P \times T)$ consists of all pairs $(q_{s_1,p}, s_2)$ such that

$$\begin{aligned} & \phi(s_2) \in p^\bullet, (s_1, s_2) \in \Gamma^+, \text{ and} \\ & \neg \exists s \in T : \phi(s) \in p^\bullet \text{ and } (s_1, s) \in \Gamma^+ \text{ and } (s, s_2) \in \Gamma^+, \end{aligned}$$

and all pairs (q_p, s_2) such that

$$\begin{aligned} & \phi(s_2) \in p^\bullet \text{ and} \\ & \neg \exists s \in T : \phi(s) \in p^\bullet \text{ and } (s, s_2) \in \Gamma^+. \end{aligned}$$

This ends the definition of the function \mathbf{proc} . It should be clear that it satisfies property (1) above. To show property (2), let $N = (P, T, F, \phi_1, \phi_2)$. Then $\mathbf{proc}(\mathbf{pructr}(N))$ and N have the same transitions, with the same labels. Note now that, in N , the sets ${}^\circ N$ and all s^\bullet , $s \in T$, form a partition of P . Thus, by Definition 88, there is an obvious bijection between the places of $\mathbf{proc}(\mathbf{pructr}(N))$ and N : q_p corresponds to the unique place in ${}^\circ N$ with label p , and $q_{s,p}$ corresponds to the unique place in s^\bullet with label p . It is straightforward to show from Lemmas 109 and 104 that this correspondence defines an isomorphism between $\mathbf{proc}(\mathbf{pructr}(N))$ and N . Note that this also shows that $\mathbf{proc}(G)$ is in $\text{PROC}(M)$, for every $G \in \text{LPO}(M)$.

This ends the proof of Theorem 107. \square

7 Comparison of Partial and Linear Order

In this section we compare the partial order behaviour $LPO(M)$ of an EN system M with its linear order behaviour $FS(M)$. In the first subsection we show that lpo-equivalence and firing sequence equivalence are the same (cf. [Pom88, PomRozSim92]). The basic concepts used to prove this are the independency relation between the transitions of an EN system, and the dependency graph of a firing sequence. These concepts are at the basis of the so-called theory of traces (see, e.g., [AalRoz88] and [DieRoz95], in particular [Maz95] and [HooRoz95]). In the second subsection we show that $LPO(M)$ can be viewed as the set of equivalence classes of a natural equivalence relation on $FS(M)$ that models concurrency. This equivalence relation, called lpo-equivalence of firing sequences, is proved to be the same as the trace equivalence of firing sequences based on the independency relation (see [Maz95]).

This section is largely based on Section 5 of [AalRoz88].

7.1 LPO-Equivalence and Firing Sequence Equivalence

To recapitulate, we now have in total three definitions of the behaviour of a contact-free EN system M : $SCG(M)$ by configuration equivalence, $FS(M)$ by firing sequence equivalence, and $LPO(M)$ by lpo-equivalence, where firing sequence equivalence is weaker than configuration equivalence (Corollary 34). Though firing sequence equivalence describes the sequential (linearly ordered) behaviour of M and lpo-equivalence the non-sequential (partially ordered) behaviour of M , we will prove in this subsection the rather surprising result that two contact-free EN systems are lpo-equivalent iff they are firing sequence equivalent. In one direction this result is as expected: if two EN systems are lpo-equivalent, then they are firing sequence equivalent. This intuitively holds because we can obtain the linear orders in $FS(M)$ from the acyclic graphs in $LPO(M)$ by ordering these graphs topologically. Topological order of acyclic graphs, as defined next, gives a fundamental connection between partial orders and linear orders.

Definition 110. A *topological order* of an acyclic labelled graph $G = (V, \Gamma, \Sigma, \phi)$ is a sequence $u_1 \cdots u_n \in V^*$, with $u_i \in V$ for $1 \leq i \leq n$, and

- (1) all u_i are distinct,
- (2) $V = \{u_1, \dots, u_n\}$, and
- (3) for all $1 \leq i, j \leq n$, if $(u_i, u_j) \in \Gamma$, then $i < j$.

A *word of G* is a word $\phi(u_1) \cdots \phi(u_n) \in \Sigma^*$, where $u_1 \cdots u_n$ is a topological order of G .

For an acyclic labelled graph $G = (V, \Gamma, \Sigma, \phi)$, the set of all topological orders of G is denoted by $\mathbf{top}(G)$. It is well known that every acyclic graph can be ordered topologically, i.e., $\mathbf{top}(G) \neq \emptyset$. Furthermore, $\mathbf{words}(G)$ denotes the set of all words of G , i.e., $\mathbf{words}(G) = \{\phi(u_1) \cdots \phi(u_n) \mid u_1 \cdots u_n \in \mathbf{top}(G)\}$.

Two graphs G and G' that represent the same partial order have the same topological orders (and hence the same words).

Lemma 111. *Let G and G' be acyclic labelled graphs.*

If $\text{tra}(G) = \text{tra}(G')$, then $\text{top}(G) = \text{top}(G')$ and $\text{words}(G) = \text{words}(G')$.

Example 31. Consider the acyclic labelled graph G shown in Fig. 66, with $V_G = \{v_1, \dots, v_6\}$ and $\Sigma_G = \{p, f, e, c\}$. Then, e.g., $v_3v_1v_4v_2v_6v_5$ and $v_1v_2v_3v_4v_5v_6$ are topological orders of G , and $pefcep$ and $ecpfpe$ are the corresponding words of G . Sequences that are *not* topological orders of G are, e.g., $v_3v_4v_6v_4v_1$ and $v_3v_4v_1v_2v_6v_5$.

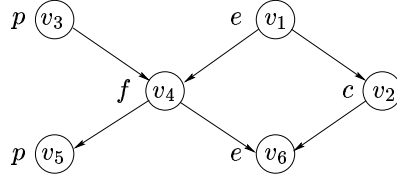


Fig. 66. An acyclic labelled graph.

We want to show that for every contact-free EN system M ,

$$\text{FS}(M) = \bigcup \{ \text{words}(G) \mid G \in \text{LPO}(M) \},$$

i.e., the firing sequences of M are the words of the pruned contracted processes of M . We know from Theorem 92 that there is a relationship between the firing sequences of M and the “complete” firing sequences of the processes of M . More precisely, $\text{FS}(M) = \{ \phi_N(s_1) \dots \phi_N(s_n) \mid {}^\circ N[s_1 \dots s_n] N^\circ \text{ for a process } N \text{ of } M \}$. Now it only remains to show that these complete firing sequences $s_1 \dots s_n$ of N are precisely the topological orders of the partially ordered set $(T_N, F_N^+ \cap (T_N \times T_N))$.

Theorem 112. *Let $N = (P, T, F)$ be a process net and let $t_1, \dots, t_n \in T$. Then ${}^\circ N[t_1 \dots t_n] N^\circ$ iff*

- (1) *all t_i are distinct,*
- (2) *$T = \{t_1, \dots, t_n\}$, and*
- (3) *for all $1 \leq i, j \leq n$, if $(t_i, t_j) \in F^+$, then $i < j$.*

This theorem is an immediate consequence (using Theorem 71(2)) of the following lemma, which is a simple extension of Theorem 79.

Lemma 113. *Let $N = (P, T, F, {}^\circ N)$ be a process net, $C \subseteq P$, and $t_1, \dots, t_n \in T$. Then ${}^\circ N[t_1 \dots t_n] C$ iff*

- (1) *all t_i are distinct,*
- (2) *${}^\rightarrow C \cap T = \{t_1, \dots, t_n\}$,*
- (3) *for all $1 \leq i, j \leq n$, if $(t_i, t_j) \in F^+$, then $i < j$, and*
- (4) *C is a slice.*

Proof. The proof is by induction on n , and is analogous to the proof of Theorem 79. The details are left to the reader. Note that if conditions (1-4) hold, then t_n is a maximal element of $\rightarrow C \cap T$ (with respect to F^+) and hence $t_n \bullet \subseteq C$. \square

Theorem 114. *Let M be a contact-free EN system. Then*
 $FS(M) = \bigcup \{\mathbf{words}(G) \mid G \in LPO(M)\}$.

Proof. By Theorem 92, $FS(M) = \{\phi_N(s_1) \cdots \phi_N(s_n) \mid \exists N \in \text{PROC}(M) : \circ N[s_1 \cdots s_n]N^\circ\}$. For a process $N = (P, T, F, \phi_1, \phi_2)$ of M , Theorem 112 now implies that $\circ N[s_1 \cdots s_n]N^\circ$ iff $s_1 \cdots s_n$ is a topological order of $\mathbf{ctr}(N)$, i.e., iff $s_1 \cdots s_n$ is a topological order of $\mathbf{pru}(\mathbf{ctr}(N))$, see Theorem 101 and Lemma 111. Note that the labelling ϕ of $\mathbf{pru}(\mathbf{ctr}(N))$ is the restriction of ϕ_N to T_N . This implies that $FS(M) = \{\phi_G(u_1) \cdots \phi_G(u_n) \mid u_1 \cdots u_n \text{ is a topological order of } G \in LPO(M)\} = \bigcup \{\mathbf{words}(G) \mid G \in LPO(M)\}$. \square

Example 32. Let M be the EN system of Fig. 49 and N the process of M in Fig. 59, with $\mathbf{pru}(\mathbf{ctr}(N))$ shown in Fig. 62. The words of $\mathbf{pru}(\mathbf{ctr}(N))$ that are given in Example 31 (see Fig. 66) are the firing sequences of M given in Example 28.

The next theorem now follows from Theorem 114 and the following simple fact: if G and G' are labelled graphs such that $G \equiv_\beta G'$, and δ is the corresponding bijection between V_G and $V_{G'}$ (see Definition 97), then:

- (1) $u_1 \cdots u_n \in \mathbf{top}(G)$ iff $\delta(u_1) \cdots \delta(u_n) \in \mathbf{top}(G')$, and
- (2) $\mathbf{words}(G') = \beta(\mathbf{words}(G))$.

Theorem 115. *Let M and M' be two contact-free EN systems and let β be a bijection from $\mathbf{use}(T_M)$ to $\mathbf{use}(T_{M'})$. If $LPO(M) \equiv_\beta LPO(M')$ then $\beta(FS(M)) = FS(M')$.*

Similar arguments can be used to give an alternative proof of Theorem 107, as follows. Let $\mathbf{pru}(\mathbf{ctr}(N)) \equiv_\beta \mathbf{pru}(\mathbf{ctr}(N'))$ where β is the identity on $\mathbf{use}(T_M)$. Then, by the simple fact above, $\mathbf{words}(\mathbf{pru}(\mathbf{ctr}(N))) = \mathbf{words}(\mathbf{pru}(\mathbf{ctr}(N')))$. By Theorem 112, $\mathbf{words}(\mathbf{pru}(\mathbf{ctr}(N))) = \{\phi_N(s_1) \cdots \phi_N(s_n) \mid \circ N[s_1 \cdots s_n]N^\circ\}$ and similarly for N' . From this and the uniqueness (modulo \equiv_β) of the process in the statement of Theorem 92, it follows that $N \equiv_\beta N'$. Note that, consequently, the function $\mathbf{proc} : LPO(M) \rightarrow \text{PROC}(M)$ mentioned in the proof of Theorem 107 can simply be defined as follows: for $G \in LPO(M)$, let $t_1 \cdots t_n$ be any element of $\mathbf{words}(G)$; then $\mathbf{proc}(G)$ is the process N corresponding to $t_1 \cdots t_n$ as constructed in the proof of Theorem 92.

We are now going to prove the, more surprising, converse of Theorem 115 (see Theorem 125). More specifically, we will show that we can “break” the linear orders in $FS(M)$ in such a way that partial orders in $LPO(M)$ are obtained. To this purpose we use the “independency relation” between the transitions of an EN system M , defined as follows.

Definition 116. Let Σ be an alphabet. A relation $I \subseteq \Sigma \times \Sigma$ is an *independency relation (over Σ)* if I is irreflexive and symmetric.

Definition 117. Let $M = (P, T, F, C_{in})$ be an EN system.

(1) The *independency relation* of M is the independency relation $\mathbf{ind}(M)$ over $\mathbf{use}(T)$ defined by

$$\mathbf{ind}(M) = \{(s, t) \in T \times T \mid s \neq t \text{ and } \exists C \in \mathbb{C}_M : \{s, t\} \mathbf{con} C\}.$$

(2) The *dependency relation* of M is the relation $\mathbf{dep}(M)$ defined by $\mathbf{dep}(M) = (\mathbf{use}(T) \times \mathbf{use}(T)) - \mathbf{ind}(M)$.

Example 33. Consider the EN system M of Fig. 49. Then $\mathbf{ind}(M) = \{(p, e), (e, p), (p, c), (c, p), (f, c), (c, f)\}$ and $\mathbf{dep}(M) = \{(p, f), (f, p), (f, e), (e, f), (e, c), (c, e)\} \cup \{(x, x) \mid x \in \{p, f, e, c\}\}$. See also Examples 8 and 9, and Fig. 18.

Usually (see [Maz95]), the independency relation of M is defined to be $\{(s, t) \in T_M \times T_M \mid s \neq t \text{ and } \mathbf{disj}(\{s, t\})\}$. The above, stronger definition serves the same purposes (cf. [Hoo94]), and moreover it satisfies the next lemma.

We will show that, using $\mathbf{ind}(M)$, we can construct for every firing sequence $t_1 \cdots t_n$ of M a pruned contracted process G of M such that $t_1 \cdots t_n$ is a word of G (without knowing the system M). First we show that we can determine $\mathbf{ind}(M)$ from $\mathbf{FS}(M)$ (without knowing M).

Lemma 118. *Let $M = (P, T, F, C_{in})$ be an EN system. Then*

$$\mathbf{ind}(M) = \{(s, t) \in T \times T \mid \exists x \in T^* : xst \in \mathbf{FS}(M) \text{ and } xts \in \mathbf{FS}(M)\}.$$

Proof. If $(s, t) \in \mathbf{ind}(M)$, then there is a $C \in \mathbb{C}_M$ such that $\{s, t\} \mathbf{con} C$. Hence, by Lemma 17, $st \mathbf{con} C$ and $ts \mathbf{con} C$. Let $x \in T^*$ with $C_{in}[x]C$. Then xst and xts are firing sequences of M . The other way around, assume that $xst, xts \in \mathbf{FS}(M)$. Let $C \in \mathbb{C}_M$ with $C_{in}[x]C$. Then $st \mathbf{con} C$ and $ts \mathbf{con} C$. Lemma 19 then implies that $\{s, t\} \mathbf{con} C$, and hence $(s, t) \in \mathbf{ind}(M)$. \square

We now “break” the linear order of every firing sequence of M , i.e., for every firing sequence we construct an acyclic graph, and then show that this graph is an element of $\mathbf{LPO}(M)$. In the next definition we use a predetermined countable set of nodes $\{v_1, v_2, v_3, \dots\}$. This set is used to canonically construct graphs. For later usage we present the definition for an arbitrary independency relation, see [HooRoz95].

Definition 119. Let Σ be an alphabet and I an independency relation over Σ . Let $x = t_1 \cdots t_n \in \Sigma^*$, with $n \geq 0$ and $t_1, \dots, t_n \in \Sigma$.

(1) The *dependency graph* of x (over I), denoted by $\mathbf{dep}_I(x)$, is the labelled graph $(V, \Gamma, \Sigma, \phi)$, where $V = \{v_1, \dots, v_n\}$, $\phi(v_i) = t_i$ for all $1 \leq i \leq n$, and, for all $1 \leq i, j \leq n$, $(v_i, v_j) \in \Gamma$ iff $i < j$ and $(t_i, t_j) \notin I$.

(2) The *pruned dependency graph* of x (over I) is $\mathbf{pru}(\mathbf{dep}_I(x))$.

Note that $v_1 \cdots v_n$ is a topological order of $\mathbf{dep}_I(x)$, and hence x is a word of $\mathbf{dep}_I(x)$.

For an EN system M we write $\mathbf{dep}_M(x)$ instead of $\mathbf{dep}_{\mathbf{ind}(M)}(x)$.

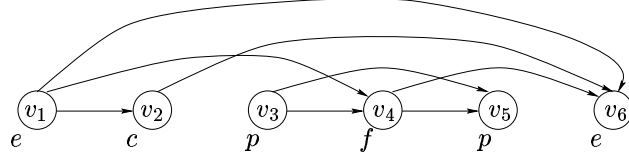


Fig. 67. A dependency graph.

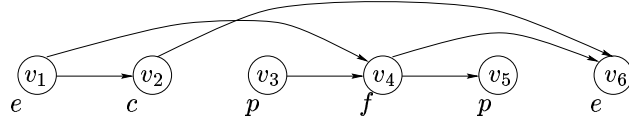


Fig. 68. A pruned dependency graph.

Example 34. Take $\Sigma = \{p, f, e, c\}$ and $I = \mathbf{ind}(M)$ as in Example 33. Then $\mathbf{dep}_I(ecpfpe)$ is given in Fig. 67 and $\mathbf{pru}(\mathbf{dep}_I(ecpfpe))$ in Fig. 68. Since the graphs in Figs. 66 and 68 are the same, $\mathbf{pru}(\mathbf{dep}_I(ecpfpe))$ can thus also be found in Fig. 66.

We now want to show that, for every contact-free EN system M , $\mathbf{LPO}(M) \equiv \{\mathbf{pru}(\mathbf{dep}_M(x)) \mid x \in \mathbf{FS}(M)\}$.

* First we show a simple connection between $\mathbf{ind}(M)$ and the processes of an EN system M .

Lemma 120. *Let $N = (P, T, F, \phi_1, \phi_2)$ be a process of a contact-free EN system M and let s, t be distinct elements of T . Then:*

- (1) *if $s \mathbf{co}_N t$, then $(\phi(s), \phi(t)) \in \mathbf{ind}(M)$,*
- (2) *if $s^\bullet \cap t^\bullet \neq \emptyset$, then $(\phi(s), \phi(t)) \in \mathbf{dep}(M)$.*

Proof. (1) This follows directly from Theorem 91(3).

(2) If $s^\bullet \cap t^\bullet \neq \emptyset$, then also $\phi(s)^\bullet \cap \phi(t)^\bullet \neq \emptyset$. □

In the next theorem we show the connection between a complete firing sequence of a process N of M and the dependency graph of the corresponding firing sequence of M .

* To begin with, we prove the following lemma.

Lemma 121. *Let $N = (P, T, F, \phi_1, \phi_2)$ be a process of a contact-free EN system M . Let ${}^\circ N[s_1 \cdots s_n] N^\circ$ with $T = \{s_1, \dots, s_n\}$, and let $G = \mathbf{dep}_M(\phi(s_1) \cdots \phi(s_n))$. Then for all $1 \leq i, j \leq n$: $(s_i, s_j) \in F_N^+$ iff $(v_i, v_j) \in \Gamma_G^+$.*

Proof. (Only-if) If $s_i^\bullet \cap s_j^\bullet \neq \emptyset$, then, by Lemma 120(2), $(\phi(s_i), \phi(s_j)) \in \mathbf{dep}(M)$ and, by Theorem 112(3), $i < j$. Hence $(v_i, v_j) \in \Gamma_G$ according to Definition 119.

(If) If $(v_i, v_j) \in \Gamma_G$, then $i < j$ and $(\phi(s_i), \phi(s_j)) \in \mathbf{dep}(M)$. Then, by Lemma 120(1), $s_i \mathbf{con} s_j$ does not hold. Hence $(s_i, s_j) \in F_N^+$ or $(s_j, s_i) \in F_N^+$. Theorem 112(3) then implies that $(s_i, s_j) \in F_N^+$. \square

Theorem 122. *Let $N = (P, T, F, \phi_1, \phi_2)$ be a process of a contact-free EN system M and let ${}^\circ N[s_1 \cdots s_n]N^\circ$. Let β be the identity on $\mathbf{use}(T_M)$. Then $\mathbf{pru}(\mathbf{ctr}(N)) \equiv_\beta \mathbf{pru}(\mathbf{dep}_M(\phi(s_1) \cdots \phi(s_n)))$.*

*

Proof. Lemma 121 implies $\mathbf{tra}(\mathbf{ctr}(N)) \equiv_\beta \mathbf{tra}(\mathbf{dep}_M(\phi(s_1) \cdots \phi(s_n)))$ via the bijection δ with $\delta(s_i) = v_i$ between the nodes of these labelled graphs. Now, by Theorem 102, the result holds. \square

Thus, for every firing sequence $t_1 \cdots t_n$ of the EN system M , its dependency graph $\mathbf{dep}_M(t_1 \cdots t_n)$ represents the same labelled partial order as (the contracted version of) a process N of M corresponding to $t_1 \cdots t_n$, as expressed in Theorem 92. Together with Theorem 107 this proves the uniqueness (modulo \equiv_β^α) of this process N , cf. the remark following Theorem 92.

Example 35. Let M be the EN system of Fig. 49 and N the process of M in Fig. 59. Then there is a firing sequence ${}^\circ N[s_1 \cdots s_n]N^\circ$ with $\phi(s_1) \cdots \phi(s_n) = \mathit{ecpfpe}$ (see Example 28). The graphs $\mathbf{pru}(\mathbf{ctr}(N))$ and $\mathbf{pru}(\mathbf{dep}_M(\mathit{ecpfpe}))$ are drawn in Figs. 62 and 66, respectively.

Theorem 123. *Let M be a contact-free EN system, and let β be the identity on $\mathbf{use}(T_M)$. Then $LPO(M) \equiv_\beta \{\mathbf{pru}(\mathbf{dep}_M(x)) \mid x \in FS(M)\}$.*

Proof. Directly by Theorems 92 and 122. \square

We need one more lemma to show that firing sequence equivalence implies lpo-equivalence.

Lemma 124. *Let M and M' be EN systems and let $\beta : \mathbf{use}(T_M) \rightarrow \mathbf{use}(T_{M'})$ be a bijection. If $\beta(FS(M)) = FS(M')$ then $\{\mathbf{pru}(\mathbf{dep}_M(x)) \mid x \in FS(M)\} \equiv_\beta \{\mathbf{pru}(\mathbf{dep}_{M'}(x)) \mid x \in FS(M')\}$.*

Proof. If $\beta(FS(M)) = FS(M')$ then, by Lemma 118, $\mathbf{ind}(M') = \{(\beta(s), \beta(t)) \mid (s, t) \in \mathbf{ind}(M)\}$. Then, for $x \in FS(M)$, $\mathbf{pru}(\mathbf{dep}_M(x)) \equiv_\beta \mathbf{pru}(\mathbf{dep}_{M'}(\beta(x)))$. \square

We are now ready to prove the main result of this section.

Theorem 125. *Let M and M' be two contact-free EN systems and let β be a bijection from $\mathbf{use}(T_M)$ to $\mathbf{use}(T_{M'})$. If $\beta(FS(M)) = FS(M')$ then $LPO(M) \equiv_\beta LPO(M')$.*

Proof. By Theorem 123 and Lemma 124. \square

Theorem 126. *Two contact-free EN systems are lpo-equivalent iff they are firing sequence equivalent.*

Proof. By Theorems 115 and 125. □

The following two corollaries of this characterization are interesting.

Corollary 127. *If two contact-free EN systems are configuration equivalent, then they are lpo-equivalent.*

Proof. Directly by Corollary 34 and Theorem 126. □

Corollary 128. *There is an algorithm that, for two arbitrary contact-free EN systems M and M' , decides whether or not M and M' are lpo-equivalent.*

Proof. According to Theorem 126 we have to check whether there exists a bijection β from $\mathbf{use}(T_M)$ to $\mathbf{use}(T_{M'})$ such that $\beta(\mathbf{FS}(M)) = \mathbf{FS}(M')$. We first construct $\mathbf{SCG}(M)$ and $\mathbf{SCG}(M')$. From these we can obtain $\mathbf{use}(T_M)$ and $\mathbf{use}(T_{M'})$. Now we test all bijections β . The languages $\mathbf{FS}(M)$, $\beta(\mathbf{FS}(M))$, and $\mathbf{FS}(M')$ are regular, and finite automata can easily be constructed for them (see Theorem 12). Since it is known from formal language theory that there is an algorithm to decide whether two finite automata are equivalent (see, e.g., [HopUll79]), we can now apply it to the automata for $\beta(\mathbf{FS}(M))$ and $\mathbf{FS}(M')$. □

For an arbitrary (not necessarily contact-free) EN system M , let $\mathbf{PD}(M) = \{\mathbf{pru}(\mathbf{dep}_M(x)) \mid x \in \mathbf{FS}(M)\}$: the set of pruned dependency graphs of M . Then Theorem 123 says that for every contact-free EN system M : $\mathbf{LPO}(M) \equiv_\beta \mathbf{PD}(M)$, where β is the identity on $\mathbf{use}(T_M)$. Now note that $\mathbf{PD}(M)$ is also defined (and meaningful) for EN systems M that are not contact-free. Lemma 124 implies that, for arbitrary EN systems M and M' , if $M \approx M'$ then $\mathbf{PD}(M) \equiv \mathbf{PD}(M')$. As we have observed already, for an EN system M that is not contact-free we consider the processes of a contact-free EN system M' that is configuration equivalent with M . Consequently, in such a case $\mathbf{PD}(M) \equiv \mathbf{PD}(M') \equiv \mathbf{LPO}(M')$; in other words $\mathbf{PD}(M)$ is isomorphic with $\mathbf{LPO}(M')$. Hence we can view $\mathbf{PD}(M)$ as the behaviour of M , and we can thus meaningfully define lpo-equivalence for arbitrary EN systems M and M' : M and M' are lpo-equivalent if $\mathbf{PD}(M) \equiv \mathbf{PD}(M')$. The above results can then easily be extended to arbitrary EN systems (e.g., Lemma 124 would then be the generalization of Theorem 125).

Finally we make a general remark concerning Theorem 126 which says that lpo-equivalence and firing sequence equivalence are the same. A technical reason for this surprising result is that all our considerations have been centered around the transitions: we compare (by means of equivalences) only EN systems with essentially the same transitions (where ‘essentially’ means: modulo bijections). In a more general approach, also EN systems with distinct transitions could be compared. To indicate that distinct transitions actually perform the same task we give them the same label (where the label thus in fact represents the task)

from a given alphabet. Then, in all definitions of behaviour, the labels of the transitions are used instead of the transitions themselves. Hence, instead of firing sequences we take the sequences of labels corresponding with the firing sequences, and the labelled partial orders are not labelled with transitions of the system, but with their labels. Note that this may drastically change the properties of the behaviour of EN systems. For example, even such a fundamental property that a transition cannot fire twice consecutively (cf. the discussion following Theorem 12), does not hold for labelled EN systems: the same label can occur twice consecutively in a firing sequence (take any two transitions that fire one after the other, and give them the same label).

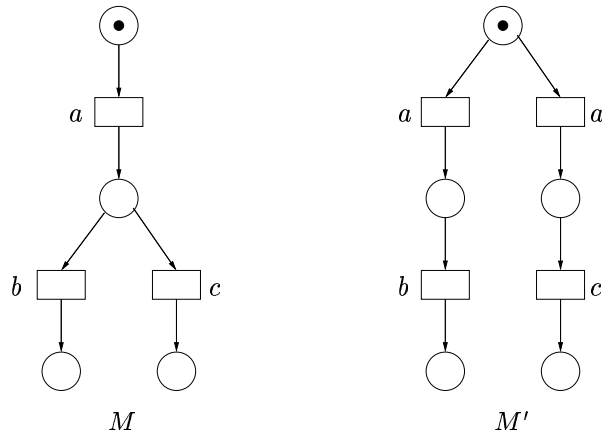


Fig. 69. Two labelled EN systems M and M' that are lpo-equivalent but not weakly configuration equivalent.

Under this approach Theorem 126 most certainly no longer holds, i.e., the new lpo-equivalence still implies the new firing sequence equivalence, but not the other way around. Similarly, Theorem 33 is no longer true: the new weak configuration equivalence still implies the new firing sequence equivalence, but not the other way around. More precisely, the new lpo-equivalence is incomparable with the new (weak) configuration equivalence; see Figs. 69 and 70 for two well-known counter-examples (where a , b , and c are the labels of the transitions rather than the transitions themselves). The relationships between various equivalences of labelled EN systems are presented in [PomRozSim92].

Note that each (old) equivalence relation implies the corresponding new one. Thus all normal forms discussed in Section 5 also hold for the new equivalences. As an example, Theorem 54 implies that for every EN system there is an lpo-equivalent (in the new sense) reduced EN system that is covered by sequential components; this is because configuration equivalence implies lpo-equivalence

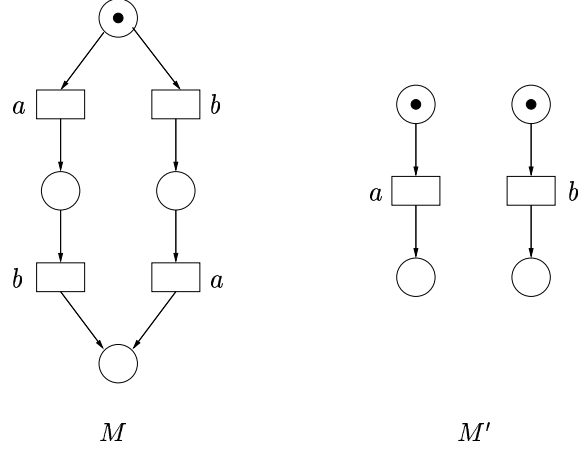


Fig. 70. Two labelled EN systems M and M' that are configuration equivalent but not lpo-equivalent.

which, in its turn, implies lpo-equivalence in the new sense.

7.2 Equivalence of Firing Sequences

Another way to understand the concurrent behaviour of an EN system is by calling two firing sequences of an EN system equivalent if they correspond to two distinct sequential observations of the same run of the system. This notion of equivalence can be formulated as follows.

Definition 129. Let M be a contact-free EN system and let $x, x' \in \text{FS}(M)$. Then x and x' are *lpo-equivalent*, denoted by $x \approx_{lpo} x'$, if there exist a process N of M and two firing sequences $y, y' \in T_N^*$ such that ${}^\circ N[y]N^\circ$, $\phi_N(y) = x$, ${}^\circ N[y']N^\circ$, and $\phi_N(y') = x'$.

Note that the process N is unique modulo \equiv_β^α (cf. the remarks following Theorems 92 and 122). Note also that, by Theorem 112, $x \approx_{lpo} x'$ iff $x, x' \in \text{words}(G)$ for some $G \in \text{LPO}(M)$.

Example 36. The two firing sequences $pefcep$ and $ecpfpe$ given in Example 28 are lpo-equivalent, i.e., $pefcep \approx_{lpo} ecpfpe$.

Due to the relationship between pruned contracted processes and dependency graphs of firing sequences discussed in the previous subsection, lpo-equivalence can be characterized in terms of dependency graphs as follows (and for non-contact-free EN systems we can take this as the definition of lpo-equivalence). This characterization is a part of Theorem 3.12 of [NieRozThi90].

Theorem 130. *Let M be a contact-free EN system and let $x, x' \in \text{FS}(M)$. Then $x \approx_{lpo} x'$ iff $\mathbf{pru}(\mathbf{dep}_M(x)) \equiv_\beta \mathbf{pru}(\mathbf{dep}_M(x'))$, where β is the identity on $\mathbf{use}(T_M)$.*

Proof. (Only-if) If $x \approx_{lpo} x'$, then there exist a process N of M and two firing sequences y, y' of N with ${}^\circ N[y]N^\circ$, $\phi_N(y) = x$, ${}^\circ N[y']N^\circ$, and $\phi_N(y') = x'$. Hence, by Theorem 122, $\mathbf{pru}(\mathbf{ctr}(N)) \equiv_\beta \mathbf{pru}(\mathbf{dep}_M(\phi_N(y)))$ and $\mathbf{pru}(\mathbf{ctr}(N)) \equiv_\beta \mathbf{pru}(\mathbf{dep}_M(\phi_N(y')))$.

(If) By Theorem 123 there exists a process N of M such that $\mathbf{pru}(\mathbf{ctr}(N)) \equiv_\beta \mathbf{pru}(\mathbf{dep}_M(x))$. Since $v_1 \cdots v_n$ is a topological order of $\mathbf{pru}(\mathbf{dep}_M(x))$ (where $n = |x|$), there is a topological order y of $\mathbf{pru}(\mathbf{ctr}(N))$ with the same labels, i.e., $\phi_N(y) = x$. Then, by Theorem 112, ${}^\circ N[y]N^\circ$. Since $\mathbf{pru}(\mathbf{ctr}(N)) \equiv_\beta \mathbf{pru}(\mathbf{dep}_M(x'))$, analogously there exists y' with $\phi_N(y') = x'$ and ${}^\circ N[y']N^\circ$. \square

We will now show a very simple characterization of isomorphism of dependency graphs, in terms of the independency relation $\mathbf{ind}(M)$ induced by the EN system M . This characterization can be proved very generally, without reference to EN systems; it is part of the theory of traces, see [HooRoz95].

For a given independency relation we define an associated equivalence relation on words as follows.

Definition 131. Let I be an independency relation over Σ .

The relation $\dot{=}_I \subseteq \Sigma^* \times \Sigma^*$ is defined as follows: for $x, y \in \Sigma^*$,

$x \dot{=}_I y$ iff there exist $a, b \in \Sigma$ and $x_1, x_2 \in \Sigma^*$, such that

$x = x_1 a b x_2$, $y = x_1 b a x_2$, and $(a, b) \in I$.

The relation $\approx_I \subseteq \Sigma^* \times \Sigma^*$ is then defined as the smallest equivalence relation that contains $\dot{=}_I$. If $x \approx_I y$ then x and y are *trace equivalent (over I)*.

Thus, two words are trace equivalent if one can be obtained from the other by interchanging independent symbols, repeatedly.

Lemma 132. *Let I be an independency relation over Σ , and let $x, y \in \Sigma^*$.*

(1) $x \approx_I y$ iff there exist $n \geq 0$ and $x_0, \dots, x_n \in \Sigma^*$ such that $x_0 = x$, $x_n = y$, and $x_{i-1} \dot{=}_I x_i$ for all $1 \leq i \leq n$.

(2) $x \approx_I y$ implies $|x| = |y|$.

Intuitively the definition of trace equivalence is based on the following property of firing sequences of an EN system $M = (P, T, F, C_{in})$: for $x, y \in T^*$ and $s, t \in T$, if $xsty \in \text{FS}(M)$ and $(s, t) \in \mathbf{ind}(M)$, then $xtsy \in \text{FS}(M)$. The proof of this property is easy: If $(s, t) \in \mathbf{ind}(M)$ then $\mathbf{disj}(\{s, t\})$. Let $C_{in}[x]C[st]D$. Then $st \mathbf{con} C$ and $\mathbf{disj}(\{s, t\})$. This implies that $\{s, t\} \mathbf{con} C$ and thus (by Lemma 17) that $C[ts]D$. Hence $xtsy \in \text{FS}(M)$. In other words, transitions in $\mathbf{ind}(M)$ are interchangeable in a firing sequence, because they actually occur concurrently. Thus, $\text{FS}(M)$ is closed under $\approx_{\mathbf{ind}(M)}$: if $x \in \text{FS}(M)$ and $x \approx_{\mathbf{ind}(M)} y$, then $y \in \text{FS}(M)$. A similar argument shows that, for $x, y \in \text{FS}(M)$, $x \approx_{\mathbf{ind}(M)} y$ iff $x \approx_I y$, where $I = \{(s, t) \in T_M \times T_M \mid s \neq t, \mathbf{disj}(\{s, t\})\}$ (the ‘‘usual’’ independency relation, cf. the remark following Definition 117).

Example 37. Again, let $\Sigma = \{p, f, e, c\}$ and $I = \{(p, e), (e, p), (p, c), (c, p), (f, c), (c, f)\}$. Then I is an independency relation, $pefc \doteq_I epfc \doteq_I epcf$, $pefc \approx_I epcf$, and $[pefc]_I = \{pefc, ppcf, epfc, epcf, ecpf\}$.

For an independency relation I over Σ and an $x \in \Sigma^*$, $[x]_I$ denotes the equivalence class of \approx_I that contains x . An equivalence class of \approx_I is called a *trace* over I ; intuitively it is the set of all sequential observations of one run of some system. In Example 37 $[pefc]_I$ is a trace over I . We will prove that $[x]_I = \mathbf{words}(\mathbf{dep}_I(x))$. To this purpose we use the following well-known result from graph theory (where the labelling of the graph is irrelevant).

Lemma 133. *Let $G = (V, \Gamma, \Sigma, \phi)$ be an acyclic graph. Let $J \subseteq V \times V$ be the independency relation (of G) defined as follows: for all $u, w \in V$, $(u, w) \in J$ iff $u \neq w$, $(u, w) \notin \Gamma$, and $(w, u) \notin \Gamma$. Then, for every topological order $u_1 \cdots u_n$ of G , $\mathbf{top}(G) = \{w_1 \cdots w_n \in V^* \mid u_1 \cdots u_n \approx_J w_1 \cdots w_n\}$.*

*

Proof. (1) It is clear that $\{w_1 \cdots w_n \in V^* \mid u_1 \cdots u_n \approx_J w_1 \cdots w_n\} \subseteq \mathbf{top}(G)$ for every $u_1 \cdots u_n \in \mathbf{top}(G)$.

(2) Now it remains to show that $\mathbf{top}(G) \subseteq \{w_1 \cdots w_n \in V^* \mid u_1 \cdots u_n \approx_J w_1 \cdots w_n\}$. Let $w_1 \cdots w_n \in \mathbf{top}(G)$. We prove by induction on k that for every k , $1 \leq k \leq n + 1$, there exists a topological order $t_1 \cdots t_n$ of G such that $t_1 \cdots t_n \approx_J w_1 \cdots w_n$ and $t_i = u_i$ for all $1 \leq i \leq k - 1$. For $k = 1$ we take $t_1 \cdots t_n = w_1 \cdots w_n$. For the induction step, assume that the statement holds for k , and consider the topological order $t_1 \cdots t_n$ which satisfies the requirements for k . There exists m , $k \leq m \leq n$, such that $u_k = t_m$. Then t_k, \dots, t_{m-1} succeed u_k in the topological order $u_1 \cdots u_n$. Hence there are no edges $(t_k, t_m), \dots, (t_{m-1}, t_m)$ in Γ (and of course no edges in the other direction because $t_1 \cdots t_n$ is a topological order). Thus $(t_k, t_m), \dots, (t_{m-1}, t_m) \in J$. This implies that $t_1 \cdots t_n \approx_J t_1 \cdots t_{k-1} t_m t_k \cdots t_{m-1} t_{m+1} \cdots t_n$. And this is a topological order (according to (1)) that satisfies the requirements for $k + 1$. \square

For $I = \mathbf{ind}(M)$, the next result is a refinement of Theorem 114, as can be seen from Theorem 123.

Theorem 134. *Let I be an independency relation over Σ , and $x \in \Sigma^*$. Then $[x]_I = \mathbf{words}(\mathbf{dep}_I(x))$.*

*

Proof. Let J be the independency relation of the graph $G = \mathbf{dep}_I(x)$, as defined in Lemma 133. Then it is easy to check that, for $u, w \in V_G$, $(u, w) \in J$ iff $(\phi(u), \phi(w)) \in I$. Hence, for all $y \in \Sigma^*$, $x \approx_I y$ iff there is a $w_1 \cdots w_n \in V_G^*$ with $v_1 \cdots v_n \approx_J w_1 \cdots w_n$ and $y = \phi(w_1) \cdots \phi(w_n)$. Now $v_1 \cdots v_n$ is a topological order of $\mathbf{dep}_I(x)$. Hence, according to Lemma 133, $v_1 \cdots v_n \approx_J w_1 \cdots w_n$ iff $w_1 \cdots w_n$ is a topological order. Thus, for all $y \in \Sigma^*$, $x \approx_I y$ iff there exists a topological order $w_1 \cdots w_n$ of $\mathbf{dep}_I(x)$ such that $y = \phi(w_1) \cdots \phi(w_n)$. \square

This implies that two words are trace equivalent iff they have isomorphic dependency graphs.

Theorem 135. *Let I be an independency relation over Σ , and let $x, y \in \Sigma^*$. Then the following four statements are equivalent.*

- (1) $x \approx_I y$,
 - (2) $[x]_I = [y]_I$,
 - (3) $\mathbf{dep}_I(x) \equiv_\beta \mathbf{dep}_I(y)$,
 - (4) $\mathbf{pru}(\mathbf{dep}_I(x)) \equiv_\beta \mathbf{pru}(\mathbf{dep}_I(y))$,
- where β is the identity on Σ .

Proof. (1) implies (3): Assume $x = x_1 a b x_2$ and $y = x_1 b a x_2$, with $x_1, x_2 \in \Sigma^*$ and $(a, b) \in I$. Then it is easy to see that $\mathbf{dep}_I(x)$ and $\mathbf{dep}_I(y)$ are isomorphic: if $|x_1| = i - 1$, then take the bijection $\delta : \{v_1, \dots, v_n\} \rightarrow \{v_1, \dots, v_n\}$ such that $\delta(v_i) = v_{i+1}$, $\delta(v_{i+1}) = v_i$, and $\delta(v_j) = v_j$ for $j \neq i, i + 1$.

(3) implies (2): This follows from Theorem 134, because $\mathbf{dep}_I(x)$ and $\mathbf{dep}_I(y)$ have corresponding topological orders with the same labels. Likewise (4) implies (2), because $\mathbf{pru}(\mathbf{dep}_I(x))$ has the same topological orders as $\mathbf{dep}_I(x)$. \square

Note that Theorems 134 and 135 do not refer to EN systems. The equality of lpo-equivalence and trace equivalence of firing sequences of EN systems is now a direct consequence of Theorems 130 and 135.

Theorem 136. *Let M be a contact-free EN system and $x, x' \in FS(M)$. Then $x \approx_{lpo} x'$ iff $x \approx_{\mathbf{ind}(M)} x'$.*

A *trace language* (over an independency relation I) is a set of equivalence classes of \approx_I . Now the behaviour of an EN system M can also be defined as the trace language $\mathbf{TR}(M) = \{[x]_{\mathbf{ind}(M)} \mid x \in FS(M)\}$, i.e., the language $FS(M)$ in which trace equivalent words are grouped together. According to Theorems 123 and 135 the function $\mathbf{prudep} : \mathbf{TR}(M) \rightarrow \mathbf{LPO}(M)$, defined by $\mathbf{prudep}([x]) = \mathbf{pru}(\mathbf{dep}_M(x))$, is a bijection between $\mathbf{TR}(M)$ and $\mathbf{LPO}(M)$, modulo isomorphism (and note that, by Theorem 134, \mathbf{words} is its inverse). Thus, $\mathbf{TR}(M)$ can also be seen as a formalization of the set of runs of the system M (see Theorem 107). Theorems 130 and 136 (i.e., Theorem 3.12 of [NieRozThi90]) show that the mapping that assigns a process with each firing sequence, as defined in Theorem 92, is a bijection between $\mathbf{TR}(M)$ and $\mathbf{PROC}(M)$, modulo isomorphism. We can define two EN systems M and M' to be *trace equivalent* if there exists a bijection $\beta : \mathbf{use}(T_M) \rightarrow \mathbf{use}(T_{M'})$ such that $\beta(\mathbf{TR}(M)) = \mathbf{TR}(M')$, where $\beta(\mathbf{TR}(M))$ is defined in the obvious way. Then, clearly, trace equivalence is the same as lpo-equivalence.

8 Branching Processes

To obtain a more complete picture of the relationship between different runs of a system that is not conflict-free, we will consider “branching runs” (also called

“unfoldings”). Intuitively, a branching run combines several conflicting runs of the system, with an indication of where the conflicts occur. At each point of conflict, one may view the system as splitting into several “parallel” copies of itself, one for each resolution of the conflict (just as a splitting universe in science-fiction). In this section we model these branching runs by “branching processes”, which are a natural extension of the processes that we have considered in the previous sections.

The theory of branching processes or unfoldings was initiated in [NiePloWin81], and developed in, e.g., [Win87, NieRozThi90, RozThi91, Eng91, NieRozThi95, WinNie95a, WinNie95b]. In [McM93, McM95, Esp94, EspRömVog96] branching processes are used to develop an efficient model checker for contact-free EN systems, i.e., an algorithm that verifies logical properties of such systems (for a fragment of so-called branching time temporal logic in [Esp94]). Most papers consider the unique maximal unfolding of the EN system, which is a (usually infinite) branching run containing *all* runs of the system. Here we consider arbitrary (finite) branching runs, as in [Eng91].

Just as processes are based on process nets, branching processes are based on branching process nets. A branching process net is like a process net, except that its places may have arbitrary output-sets. A conflict is modelled by a place with more than one transition in its output-set. Whenever a conflict occurs, the conflicting parts of the branching run should be separated after the conflict. This is formalized by requiring the following “conflict relation” to be irreflexive.

Definition 137. Let $N = (P, T, F)$ be a net. The *conflict relation* of N is the binary relation $\otimes \subseteq X_N \times X_N$ defined as follows: for all $x_1, x_2 \in X_N$, $x_1 \otimes x_2$ if there exist distinct transitions $t_1, t_2 \in T$ such that $\bullet t_1 \cap \bullet t_2 \neq \emptyset$ and $t_i F^* x_i$ for $i = 1, 2$.

The conflict relation of a net N will also be denoted by \otimes_N . Note that it is a symmetric relation. We now use it to define branching process nets, introduced in [NiePloWin81] (where they are called occurrence nets).

Definition 138. A net $N = (P, T, F)$ is a *branching process net*, abbreviated *b-process net*, if:

- (1) N is acyclic,
- (2) $\#(\bullet p) \leq 1$ for all $p \in P$, and
- (3) \otimes_N is irreflexive.

Note that the irreflexivity of \otimes_N can also be expressed as follows: for all distinct transitions $t_1, t_2 \in T$, if $\bullet t_1 \cap \bullet t_2 \neq \emptyset$, then $\{x \in X_N \mid t_1 F^* x\} \cap \{x \in X_N \mid t_2 F^* x\} = \emptyset$. Intuitively this means that conflicting transitions t_1 and t_2 have disjoint futures.

As for process nets, we will view a b-process net N as an EN system with initial configuration $\circ N$. Every process net is a b-process net, because \otimes_N is empty for a process net N .

Example 38. The EN systems in Figs. 25, 26, 27, and 28 are b-process nets. This shows that confusion can be present in b-process nets. The acyclic EN system

in Fig. 33 is not a b-process net because $\#(\bullet p_3) = 2$. The EN system of Fig. 37 satisfies (1) and (2) above but it is not a b-process net, because $t_3 \otimes t_3$. In fact, $\bullet t_1 \cap \bullet t_2 \neq \emptyset$ and both $t_1 F^* t_3$ and $t_2 F^* t_3$. Thus, t_3 is in the conflict relation with itself and so \otimes is not irreflexive.

In a b-process net N we define the notion of a slice in the same way as for a process net, but we additionally require that its places are not in the conflict relation \otimes_N . We will denote the complement of \otimes_N by $\overline{\otimes}_N$; note that it is a reflexive symmetric relation (cf. Definition 65).

Definition 139. A *slice* of a b-process net N is a maximal $(\mathbf{co}_N \cap \overline{\otimes}_N)$ -clique C of N such that $C \subseteq P_N$.

Example 39. Consider the b-process net N of Fig. 26. Its conflict relation is $\otimes_N = \{(x, y), (y, x) \mid x \in \{t_3, p_4\}, y \in \{t_1, p_5, t_2, p_3\}\}$. The slices of N are $\{p_1, p_2\}$, $\{p_4\}$, $\{p_5, p_2\}$, $\{p_1, p_3\}$, and $\{p_5, p_3\}$. Note that $\{p_4\}$ and $\{p_5, p_3\}$ are not cuts.

If one systematically considers the relation $\mathbf{co}_N \cap \overline{\otimes}_N$ instead of the relation \mathbf{co}_N , then Lemmas 76, 77, and 78, and Theorems 79 and 81, also hold for b-process nets. The details of the proofs are left to the reader. Thus, by Lemma 78 and Theorem 81, b-process nets are contact-free and reduced. In the next theorem we state the generalization of Theorem 79: the slices of a b-process net are exactly its reachable configurations.

Theorem 140. Let $N = (P, T, F, \circ N)$ be a b-process net and let $C \subseteq P$. $C \in \mathcal{C}_N$ iff C is a slice of N .

Based on b-process nets, we now define branching processes of an EN system, see [Eng91]. They can be viewed as records of all events that occur during a branching run of the system.

Definition 141. Let $N = (P_N, T_N, F_N, \phi_1, \phi_2)$ be a (Σ_1, Σ_2) -labelled b-process net and let $M = (P, T, F, C_{in})$ be a contact-free EN system.

Then N is a *branching process* of M , abbreviated *b-process*, if

- (1)–(5) of Definition 88 hold, and
- (6) for all $s, t \in T_N$, if $\bullet s = \bullet t$ and $\phi_2(s) = \phi_2(t)$, then $s = t$.

For a contact-free EN system M , we denote the set of all b-processes of M by $\text{BPROC}(M)$.

Condition (6) above says that a conflict is always between two distinct transitions of M . This is a natural requirement that prevents the same run to appear twice in the record of a branching run. However, many properties of b-processes also hold without the requirement.

Example 40. (1) Let M be the (contact-free) EN system of Fig. 53. A branching process N of M is drawn in Fig. 71. Note that the process of M that is given in Fig. 58 is “part” of N , i.e., it is one of the runs of M that is combined in the branching run corresponding to N .

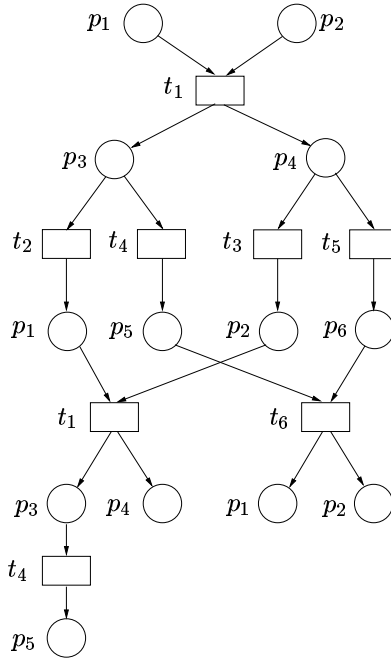


Fig. 71. A branching process of the EN system of Fig. 53.

(2) A b-process of the EN system of Fig. 5 (mutual exclusion) is drawn in Fig. 72. Intuitively, it is a combination of four possible runs of the system: component i gets permission to access its critical section, and then component j gets permission, for every combination of $i, j \in \{1, 2\}$. This can be compared with the process in Fig. 60, corresponding to one run during which components 1,2,1, and 1 get permission, respectively.

It can be proved that Lemma 89 and Theorems 90 and 91 are also true for b-processes (again with $\mathbf{co}_N \cap \bar{\otimes}_N$ instead of \mathbf{co}_N). In fact, they are even true when condition (6) is dropped from the definition of b-process (Definition 141); this will be needed in the proof of Theorem 149.

As for processes, to compare the behaviour of two different EN systems we are mainly interested in the events, and their relationships, rather than in the conditions of a branching process. Thus, we will remove the conditions and consider pruned contracted b-processes (see Section 6.4). In the case of branching processes we are not only interested in the causal relationship between events but also in their conflict relation. However, since the conflicts in a b-process are modelled by places with more than one transition in their output-set, the conflict relation is lost when the places are removed. Consequently, in pruned contracted b-processes, we have to model the conflict relation explicitly.

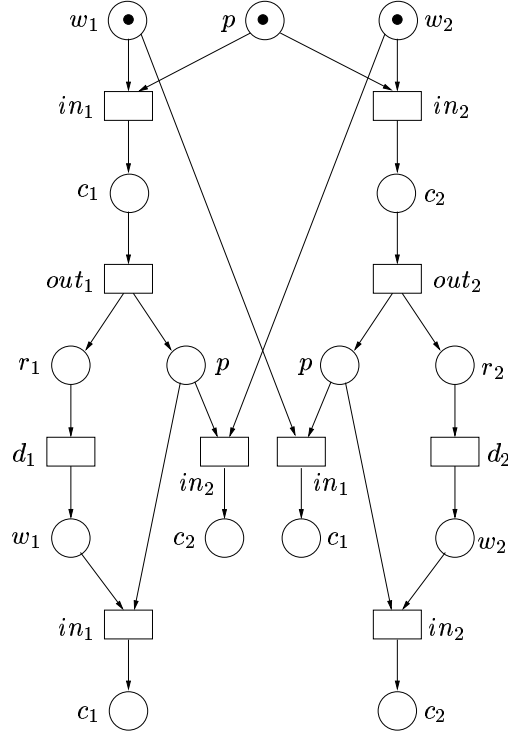


Fig. 72. A b-process of the mutual exclusion system of Fig. 5.

Just as a pruned contracted process represents a labelled partial order, a pruned contracted b-process will represent a labelled partial order together with a conflict relation. Such a partial order with conflict relation is called an “event structure”, introduced in [NiePloWin81] (see also, e.g., [Win87, WinNie95a]).

Definition 142. An *event structure* is a triple (A, ρ, \otimes) where

- (1) (A, ρ) is a partially ordered set,
- (2) $\otimes \subseteq A \times A$ is an irreflexive symmetric relation, and
- (3) for all $a, b, a', b' \in A$, if $a \otimes b$, $a \rho a'$, and $b \rho b'$, then $a' \otimes b'$.

Condition (3) relates the causal relation ρ to the conflict relation \otimes . It expresses the fact that ρ inherits \otimes , in the sense that a conflict between a and b is inherited by all ρ -descendants of a and b (as in a vendetta).

It is easy to see that for every b-process net N , (X_N, F_N^+, \otimes_N) is an event structure (cf. Lemma 74). For a b-process N of an EN system M , we will be interested in the “labelled event structure”

$$(T_N, F_N^+ \cap (T_N \times T_N), \otimes_N \cap (T_N \times T_N), T_M, \phi_{2N}).$$

As in the case of processes and labelled partial orders, we will represent such a labelled event structure by an acyclic labelled graph, with an additional relation that models conflict. We will call this a “labelled branching graph”, defined as follows.

Definition 143. Let Σ be an alphabet. A (Σ) -labelled b-graph is a quintuple $G = (V, \Gamma, \otimes, \Sigma, \phi)$, where $(V, \Gamma, \Sigma, \phi)$ is an acyclic Σ -labelled graph, and $\otimes \subseteq V \times V$ is the *conflict relation* of G , such that $\neg \exists v_1, v_2, v \in V : v_1 \otimes v_2, v_1 \Gamma^* v$, and $v_2 \Gamma^* v$.

It is left to the reader to define the appropriate notion of isomorphism of labelled b-graphs and sets of labelled b-graphs (cf. Definitions 97 and 98). Note that a labelled b-graph G can be viewed as a graph with two types of edges: directed edges in Γ_G , and undirected edges in \otimes_G .

Example 41. Figure 73 shows an example of a labelled b-graph G . The conflict relation of G is indicated by undirected dashed lines. The nodes of G are labeled

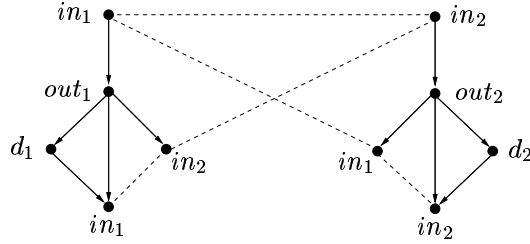


Fig. 73. A labelled b-graph.

by the transitions of the mutual exclusion system of Fig. 5. We will see later that G is in fact a contracted b-process of that EN system.

We now define the labelled event structure that is represented by a labelled b-graph, cf. Definition 99.

Definition 144. Let $G = (V, \Gamma, \otimes, \Sigma, \phi)$ be a labelled b-graph. The *transitive closure* of G , denoted by $\mathbf{tra}(G)$, is the labelled b-graph $(V, \Gamma^+, \mathbf{tra}(\otimes), \Sigma, \phi)$, where $\mathbf{tra}(\otimes) = \{(v_1, v_2) \mid \exists v'_1, v'_2 \in V : v'_1 \otimes v'_2 \text{ and } v'_i \Gamma^* v_i \text{ for } i = 1, 2\}$. We also say that G *represents* $\mathbf{tra}(G)$.

It should be clear that $(V, \Gamma^+, \mathbf{tra}(\otimes))$ is an event structure, and so $\mathbf{tra}(G)$ is a labelled event structure. The pruned version of a labelled b-graph is defined next, cf. Definition 100.

Definition 145. Let $G = (V, \Gamma, \otimes, \Sigma, \phi)$ be a labelled b-graph. The *pruned version* of G , denoted by $\mathbf{pru}(G)$, is the labelled b-graph $(V, \Gamma', \otimes', \Sigma, \phi)$ with $\Gamma' = \{(v, w) \in \Gamma \mid \neg \exists u \in V : (v, u) \in \Gamma^+ \text{ and } (u, w) \in \Gamma^+\}$ and $\otimes' = \{(v_1, v_2) \in \otimes \mid \neg \exists (v'_1, v'_2) \in \otimes : (v'_1, v'_2) \neq (v_1, v_2) \text{ and } v'_i \Gamma^* v_i \text{ for } i = 1, 2\}$.

It is not difficult to show that Theorems 101 and 102 still hold for labelled b-graphs.

Example 42. Figure 74 shows the pruned version $\mathbf{pru}(G)$ of the labelled b-graph G of Fig. 73.

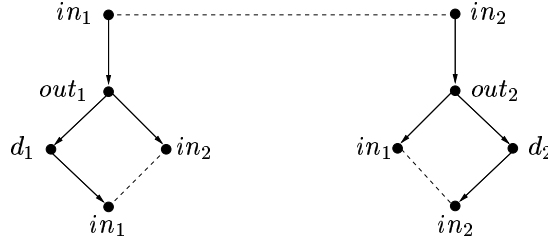


Fig. 74. The pruned version of the labelled b-graph of Fig. 73.

After discussing labelled b-graphs, we return to b-processes and show how to contract and prune them, cf. Definition 103.

Definition 146. Let $N = (P, T, F, \phi_1, \phi_2)$ be a b-process of an EN system M .

(1) The *contracted version* of N , denoted by $\mathbf{ctr}(N)$, is the labelled b-graph $(T, \Gamma, \otimes, T_M, \phi_2)$ such that, for all $s, t \in T$,

$(s, t) \in \Gamma$ iff $s \bullet \cap \bullet t \neq \emptyset$, and $(s, t) \in \otimes$ iff $\bullet s \cap \bullet t \neq \emptyset$.

(2) The *pruned contracted version* of N is the labelled b-graph $\mathbf{pru}(\mathbf{ctr}(N))$.

Example 43. Let N be the b-process given in Fig. 72. Its contracted version $\mathbf{ctr}(N)$ and pruned contracted version $\mathbf{pru}(\mathbf{ctr}(N))$ are shown in Figs. 73 and 74, respectively. They can be compared with the contracted and pruned contracted process of Fig. 64.

It is easy to see (using the analogue of Lemma 104 for $\mathbf{ctr}(N)$) that, for $\mathbf{ctr}(N) = (T_N, \Gamma, \otimes, \mathbf{use}(T_M), \phi_2)$, $\mathbf{tra}(\otimes) = \otimes_N \cap (T_N \times T_N)$. This implies that $\mathbf{ctr}(N)$ is indeed a labelled b-graph, and that both $\mathbf{ctr}(N)$ and $\mathbf{pru}(\mathbf{ctr}(N))$ represent the labelled event structure $(T_N, F_N^+ \cap (T_N \times T_N), \otimes_N \cap (T_N \times T_N), T_M, \phi_{2N})$. We note here that, by condition (6) of Definition 141, this labelled event structure is deterministic in the sense of [Vaa91].

For a contact-free EN system M we denote by $\text{LES}(M)$ the set of all pruned contracted b-processes of M (where LES stands for Labelled Event Structures). Hence

$$\text{LES}(M) = \{\mathbf{pru}(\mathbf{ctr}(N)) \mid N \in \text{BPROC}(M)\}.$$

Definition 147. Two contact-free EN systems M and M' are *les-equivalent* if $\text{LES}(M) \equiv \text{LES}(M')$.

Obviously, for every contact-free EN system M , $\text{LPO}(M) = \{G \in \text{LES}(M) \mid \otimes_G = \emptyset\}$. This implies that if two EN systems are les-equivalent, then they are lpo-equivalent. To show that this also holds the other way around, we need the concept of a “configuration” of an event structure. We define it for labelled b-graphs. Intuitively, a configuration of a branching run is one of the conflict-free runs that the branching run consists of.

Definition 148. Let $G = (V, \Gamma, \otimes, \Sigma, \phi)$ be a labelled b-graph.

A *configuration* of G is a subset R of V such that

- (1) $\forall v_1, v_2 \in R : \neg v_1 \otimes v_2$, and
- (2) $\forall v_1, v_2 \in V$: if $v_2 \in R$ and $v_1 \Gamma v_2$, then $v_1 \in R$.

For a configuration R of G , the (non-branching) labelled graph *induced by R* , denoted $G[R]$, is $(R, \Gamma \cap (R \times R), \Sigma, \phi \upharpoonright R)$.

It is well known that configurations of unfoldings of an EN system correspond to processes of the system (cf. Theorem 4.6 of [NieRozThi90] and the discussion at the end of that paper). In the next theorem we show that, due to this correspondence, $\text{LES}(M)$ can be recovered from $\text{LPO}(M)$ without knowing M .

Theorem 149. Let M be a contact-free EN system, and let $G = (T, \Gamma, \otimes, \Sigma, \phi)$ be a labelled b-graph with $\Sigma = \mathbf{use}(T_M)$.

Then, $G \in \text{LES}(M)$ iff

- (1) $\mathbf{pru}(G) = G$,
- (2) $G[R] \in \text{LPO}(M)$ for every configuration R of G , and
- (3) $\forall s_1, s_2 \in T$: if $s_1 \otimes s_2$, then $(\phi(s_1), \phi(s_2)) \in \mathbf{dep}(M)$ and $\phi(s_1) \neq \phi(s_2)$.

*

Proof. (Only-if) This direction of the proof consists of verifying a number of rather obvious properties of pruned contracted b-processes. Let $G = \mathbf{pru}(\mathbf{ctr}(N))$ for a b-process $N = (P, T, F, \phi_1, \phi)$ of M . Property (1) is obvious. For a configuration $R \subseteq T$, let $N[R]$ be the labelled net $(P', T', F', \phi'_1, \phi')$ with $T' = R$, $P' = {}^\circ N \cup \mathbf{nbh}(R)$, and F', ϕ'_1, ϕ' are the restrictions of F, ϕ_1, ϕ to P' and T' . It is straightforward to show that $N[R]$ is a process of M with $\mathbf{pru}(\mathbf{ctr}(N[R])) = G[R]$. This proves property (2). To show property (3), let $s_1, s_2 \in T$ with $s_1 \otimes s_2$. Then $s_1 \neq s_2$ by Definition 143, and $\bullet s_1 \cap \bullet s_2 \neq \emptyset$ by Definition 146. This implies that $\bullet \phi(s_1) \cap \bullet \phi(s_2) \neq \emptyset$, and so $(\phi(s_1), \phi(s_2)) \in \mathbf{dep}(M)$. Now we assume that $\phi(s_1) = \phi(s_2)$ and derive a contradiction. Since $s_1 \neq s_2$, condition (6) of Definition 141 implies that $\bullet s_1 \neq \bullet s_2$. Thus, there exist distinct places $q_1 \in \bullet s_1$ and $q_2 \in \bullet s_2$ with the same label. Hence, by the b-analogue of Theorem 91(1), either $q_1 \mathbf{li}_N q_2$ or $q_1 \otimes_N q_2$. In both cases it is easy to see that there exist transitions s'_1, s'_2 such that $\bullet s'_1 \cap \bullet s'_2 \neq \emptyset$, $s'_i F^* s_i$ for $i = 1, 2$, and $(s'_1, s'_2) \neq (s_1, s_2)$. Since G is pruned, this contradicts the fact that $s_1 \otimes s_2$.

(If) This direction of the proof is based on Lemma 109 and on the proof of Theorem 107. Let $G = (T, \Gamma, \otimes, \Sigma, \phi)$ satisfy properties (1–3). We have to show the existence of a b-process N of M such that $\mathbf{pru}(\mathbf{ctr}(N)) = G$. The construction of N is exactly the same as the construction of $\mathbf{proc}(G)$ (for $G \in \mathbf{LPO}(M)$) in the proof of the If-part of Theorem 107, at the end of Section 6. Thus, we define N to be the $(P_M, \mathbf{use}(T_M))$ -labelled net (P, T, F, ϕ_1, ϕ) , where P, ϕ_1 , and F are defined in exactly the same way as for $\mathbf{proc}(G)$. In what follows we prove that N is a b-process of M such that $\mathbf{pru}(\mathbf{ctr}(N)) = G$. As usual, we will drop the subscript of ϕ_1 .

Conditions (1)–(5) of Definition 141 (see Definition 88) are all obvious, except that we have to show that for every $s \in T$, $\phi(\bullet s) = \bullet(\phi(s))$ and $\phi \upharpoonright \bullet s$ is injective. For a given s , let $R_s = \{s' \in T \mid s' \Gamma^* s\}$. It is easy to see that R_s is a configuration of G , and so $G[R_s] \in \mathbf{LPO}(M)$. From the definitions of N and $\mathbf{proc}(G[R_s])$, it should be clear that $\bullet s$ contains the same places in N and in $\mathbf{proc}(G[R_s])$, with the same labels. Since $\mathbf{proc}(G[R_s])$ is a process of M , it satisfies the above two requirements.

Next we claim that

$$\text{for } s, s' \in T, \quad s' F^+ s \text{ iff } s' \Gamma^+ s. \quad (1)$$

In the Only-if direction this follows directly from the definition of N . In the If direction it follows from an argument about $\mathbf{proc}(G[R_s])$ similar to the one above, using the fact that the partial order on the transitions of $\mathbf{proc}(G[R_s])$ is represented by $G[R_s]$, see Lemma 104. Equivalence (1) implies that N is acyclic, which is condition (1) of a b-process net, cf. Definition 138.

Similarly, for the conflict relations we claim that

$$\text{for } s_1, s_2 \in T, \quad s_1 \otimes_N s_2 \text{ iff } (s_1, s_2) \in \mathbf{tra}(\otimes), \quad (2)$$

where \otimes is the conflict relation of G .

We first show the Only-if direction of equivalence (2). By equivalence (1), it suffices to prove this for the case that $\bullet s_1 \cap \bullet s_2 \neq \emptyset$. Suppose that $(s_1, s_2) \notin \mathbf{tra}(\otimes)$. Then $R = \{s \in T \mid s \Gamma^* s_1 \text{ or } s \Gamma^* s_2\}$ is a configuration of G , and so $G[R]$ is in $\mathbf{LPO}(M)$. Again, this implies that also $\bullet s_1 \cap \bullet s_2 \neq \emptyset$ in $\mathbf{proc}(G[R])$, contradicting the fact that this is a process of M . From this Only-if direction, and the fact that $\mathbf{tra}(\otimes)$ is irreflexive, it follows that \otimes_N is irreflexive (condition (3) of a b-process net, cf. Definition 138). Since condition (2) of a b-process net is immediate from the definition of N , this shows that N is a b-process net. Thus, we have almost proved that N is a b-process: only condition (6) of Definition 141 is still missing.

Next we show the If direction of equivalence (2). By equivalence (1), it suffices to prove this for the case that $s_1 \otimes s_2$. This implies, by property (3), that $(\phi(s_1), \phi(s_2)) \in \mathbf{dep}(M)$. Assume that $\neg s_1 \otimes_N s_2$. If $s_1 \mathbf{li}_N s_2$, then, by equivalence (1), $s_1 \Gamma^* s_2$ or $s_2 \Gamma^* s_1$, which contradicts the irreflexivity of $\mathbf{tra}(\otimes)$. If $s_1 \mathbf{co}_N s_2$, then (s_1, s_2) is in the relation $\mathbf{co}_N \cap \overline{\otimes}_N$. Thus, by the b-analogue of Theorem 91(3) (which is also valid without condition (6) of Definition 141), there exists $C \in \mathbb{C}_M$ such that $\{\phi(s_1), \phi(s_2)\} \mathbf{con} C$. This contradicts the fact that $(\phi(s_1), \phi(s_2)) \in \mathbf{dep}(M)$.

Equivalences (1) and (2) show that $\mathbf{pru}(\mathbf{ctr}(N))$ and G represent the same labelled event structure. Hence, because they are both pruned, they are the same (see Theorem 101). Note that we have applied \mathbf{ctr} to N without knowing whether condition (6) of Definition 141 is satisfied; it should be clear that Definition 146 can also be used in this case.

It remains to show condition (6) of Definition 141. Consider distinct transitions $s_1, s_2 \in T$ with $\bullet s_1 = \bullet s_2$. It is straightforward to show that (s_1, s_2) is in the conflict relation of $\mathbf{pru}(\mathbf{ctr}(N))$. Thus, since $\mathbf{pru}(\mathbf{ctr}(N)) = G$, $s_1 \otimes s_2$. It now follows from property (3) that $\phi(s_1) \neq \phi(s_2)$. \square

We note here that the condition $\phi(s_1) \neq \phi(s_2)$ in property (3) of Theorem 149 corresponds precisely to condition (6) of Definition 141. Theorem 149 is still true when both conditions are dropped.

Theorem 149 allows us to show that lpo-equivalence implies les-equivalence. This result can also be deduced from the results in [NieRozThi90, Eng91] together with Lemma 5.3 of [Vaa91].

Theorem 150. *Let M and M' be two contact-free EN systems and let β be a bijection from $\mathbf{use}(T_M)$ to $\mathbf{use}(T_{M'})$. If $\mathbf{LPO}(M) \equiv_\beta \mathbf{LPO}(M')$, then $\mathbf{LES}(M) \equiv_\beta \mathbf{LES}(M')$.*

Proof. By Theorem 115, $\beta(\mathbf{FS}(M)) = \mathbf{FS}(M')$. Hence, by Lemma 118, $\mathbf{ind}(M') = \{(\beta(s), \beta(t)) \mid (s, t) \in \mathbf{ind}(M)\}$ (as in the proof of Lemma 124). It now follows from Theorem 149 that $\mathbf{LES}(M) \equiv_\beta \mathbf{LES}(M')$. \square

Together with the remark after Definition 147 this shows that les-equivalence is the same as lpo-equivalence (and hence the same as firing sequence equivalence by Theorem 126).

Theorem 151. *Two contact-free EN systems are les-equivalent iff they are lpo-equivalent.*

The analogue of Theorem 107 also holds for b-processes, i.e., the function **pructr** is a bijection between $\text{BPROC}(M)$ and $\text{LES}(M)$; the proof is by Lemmas 108 and 109, which also hold for b-processes (even when the conflict relations are dropped from the pruned contracted b-processes, because in this case the system M is known).

It is shown in, e.g., [RozThi91, WinNie95a] that there is a general relationship between trace languages and event structures, similar to the relationship between traces and dependency graphs considered in Section 7.2. Possibly, this could be used to give an alternative proof of Theorem 151 that is similar to the one of Theorem 126 in Section 7.1.

In Sections 4-8 we have studied a number of equivalence relations between contact-free EN systems and the relationships between them: isomorphism, which implies configuration equivalence, which in its turn implies firing sequence equivalence, which equals lpo-equivalence, weak configuration equivalence, trace equivalence, and les-equivalence. This leaves us with one notion of equivalent structure of EN systems (isomorphism), and essentially two notions of equivalent behaviour of EN systems, one stronger than the other: configuration equivalence, for which we require the two systems to have the same state space, and firing sequence equivalence (or lpo-equivalence), for which we require the two systems to have the same runs (i.e., “paths” in the state space). These are the two notions of behaviour that play an important role in system theory in general.

9 P/T Systems

9.1 Informal Introduction

For the modelling of real-life systems, EN systems are not well suited in general, because the EN systems that are obtained in this way are usually much too large. This has led to the development of P/T systems, which yield much more manageable size, because they allow to “fold” different parts of an EN system. This is best illustrated by the following simple example. Recall from Section 4.1 the *producer/consumer problem*, and now consider the EN system of Fig. 75 which models the producer/consumer problem with three producers and two consumers. Since all producers have the same structure, and so have all consumers, this system can be modelled in a more compact way. In fact, by “folding” the three producers and the two consumers, we obtain the P/T system of Fig. 76 in which each producer and each consumer is represented by a token. Thus, the main difference with the EN system model is that places may now contain more than one token. As a matter of fact, this difference is quite drastic because one may even have P/T systems where no a priori bound can be given on the number of tokens that may be present in a given place during the execution of the system (think, e.g., of the dynamic creation of new producers and consumers). Thus, there may be places with an unbounded number of tokens.

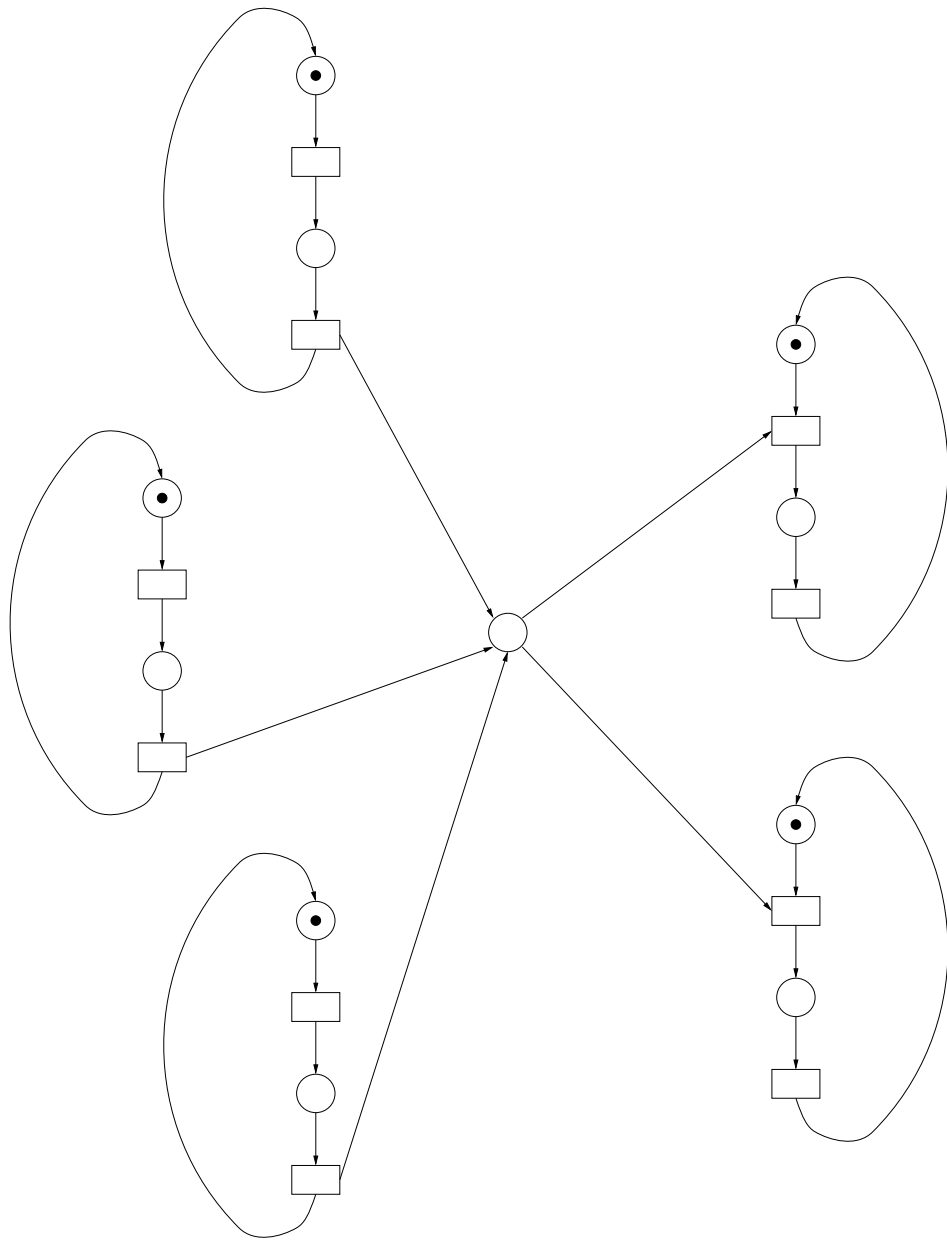


Fig. 75. EN system for 3 producers and 2 consumers.

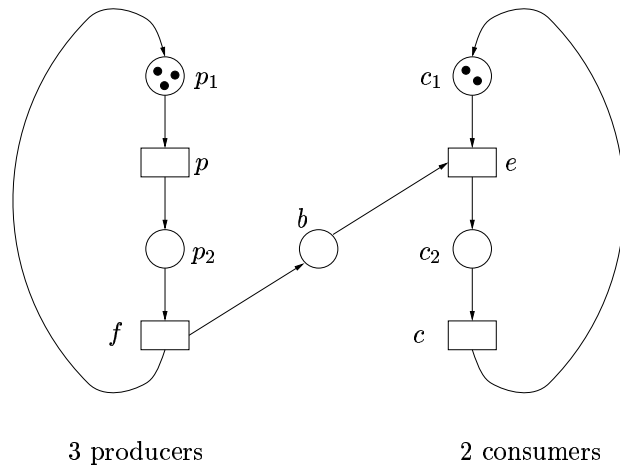


Fig. 76. P/T system for 3 producers and 2 consumers.

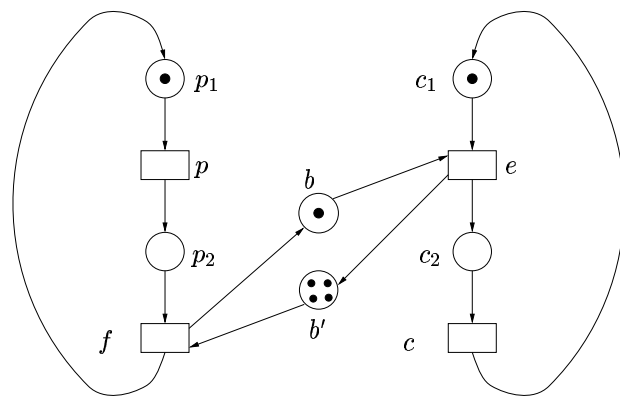


Fig. 77. A buffer with capacity 5.

The most important technical consequence of this fact is that the configuration graphs of (even very simple) P/T systems may have an infinite number of nodes. In fact, since in the P/T system model a token may be added to a place even if it already contains tokens, the P/T system of Fig. 76 has an infinite configuration graph because the buffer can contain arbitrarily many tokens (if the consumers are not as fast as the producers). To model the producer/consumer problem with a bounded buffer capacity of, say, five production units, one possibility is to explicitly declare the buffer place b to be of capacity 5. In this way EN systems are the same as P/T systems with places of capacity 1 only. Another possibility

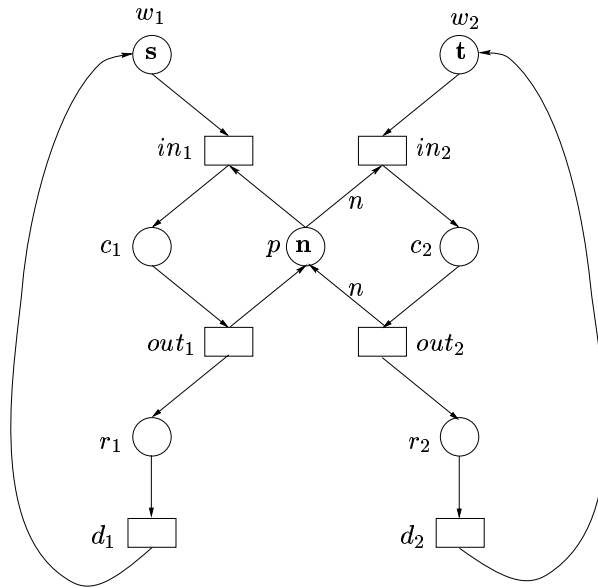


Fig. 78. The readers/writers problem.

to obtain a buffer of capacity 5 is illustrated in Fig. 77: b has a complementary place b' which, at each moment, contains the remaining capacity of the buffer b (in the figure, b contains one token and at most four tokens can be added to it).

As a last example of a P/T system we consider the *readers/writers problem*, depicted in Fig. 78, which generalizes the mutual exclusion problem of Section 4.1. There are s readers and t writers, that wish to obtain access to the same piece of memory, to read from it or write into it. They are represented by s tokens in place w_1 and t tokens in place w_2 . Writers change the content of the memory, but readers do not. Thus, a writer should have exclusive access to memory, i.e., all other writers and readers have to wait. We also assume that at most n readers can have simultaneous access to memory. To realize these requirements, n permits are used, represented by n tokens in place p . A reader needs just one permit, but a writer needs all n permits. Thus, when a writer wishes to enter its critical section it has to take n tokens from place p , and when it leaves its critical section it has to put n tokens back. This guarantees that the writer has unique access to the memory. Thus, in a P/T system, a transition can remove (add) more than one token from (to) a place, indicated by a positive integer weight on the edge from (to) that place.

9.2 Definitions

We first generalize the notion of a net and then the notion of an EN system. These generalizations will be called multi-net and P/T system. The places of a multi-net can contain more than one token, and a transition can remove more than one token from an input place and add more than one token to an output place. Each place has a capacity (in $\mathbb{N}_+ \cup \{\omega\}$): the maximal number of tokens the place can contain (where capacity ω means that the number of tokens is unlimited).

Definition 152. A *multi-net* is a tuple $N = (P, T, F, W, K)$, where

- (1) (P, T, F) is a net,
- (2) $W : F \rightarrow \mathbb{N}_+$ is the *weight function*, and
- (3) $K : P \rightarrow \mathbb{N}_+ \cup \{\omega\}$ is the *capacity function*.

In a multi-net a configuration is thus determined by recording for each place the number of tokens in that place. Hence, formally a configuration is represented by a function $C : P \rightarrow \mathbb{N}$, where, for each place p , $C(p)$ indicates the number of tokens in p . Such a function is also called a *multi-set*: it is a “set” of places such that each place can occur several times in the “set”.

Definition 153. A *configuration* of a multi-net $N = (P, T, F, W, K)$ is a function $C : P \rightarrow \mathbb{N}$, such that $\forall p \in P : \text{if } K(p) \in \mathbb{N}_+, \text{ then } C(p) \leq K(p)$.

A configuration C of a multi-net can also be seen as a vector: if $P = \{p_1, \dots, p_n\}$, then C corresponds to the n -dimensional vector (v_1, \dots, v_n) for which $v_i = C(p_i)$ for $1 \leq i \leq n$. We will return to this point of view at the end of this section.

We now define a P/T system as a multi-net with an initial configuration.

Definition 154. A *place/transition system*, P/T system for short, is a tuple $M = (P, T, F, W, K, C_{in})$, where

- (1) (P, T, F, W, K) is a multi-net, and
- (2) $C_{in} : P \rightarrow \mathbb{N}$ is a configuration of that multi-net: the *initial configuration*.

For a P/T system M as above we will use P_M, T_M, F_M, W_M, K_M , and $(C_{in})_M$ to denote P, T, F, W, K , and C_{in} , respectively.

Graphically we represent a configuration $C : P \rightarrow \mathbb{N}$ by marking each place p with $C(p)$ tokens (and thus a configuration is once again called a *marking* of the net as well). In the graphical representation of a P/T system $M = (P, T, F, W, K, C_{in})$ we label an edge $f \in F$ with $W(f)$, but we usually omit the label 1. We also label each place $p \in P$ with its capacity $K(p)$, but we usually omit the label ω . Finally, we mark the initial configuration C_{in} by tokens.

Example 44. Figure 79 is a graphical representation of a P/T system $M = (P, T, F, W, K, C_{in})$ with: $W(p_2, t_2) = 2$, $W(p_5, t_4) = 3$, $W(t_4, p_6) = 2$, and $W(p, t) = W(t, p) = 1$ for all other (p, t) and (t, p) in F ; $K(p_2) = 2$ and $K(p) = \omega$ for all other p in P ; $C_{in}(p_2) = 2$, $C_{in}(p_4) = 1$, and $C_{in}(p) = 0$ for all other p in P . Figures 76, 77, and 78 are three other examples of a P/T system.

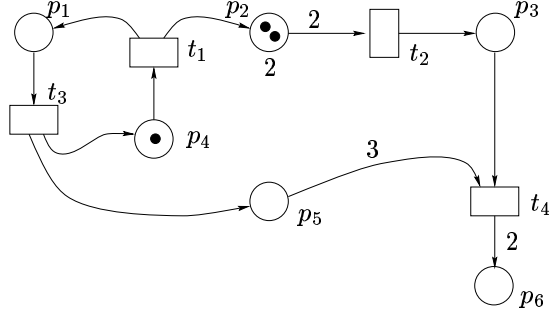


Fig. 79. A P/T system.

A P/T system is a multi-net with an initial configuration, while an EN system is a net with an initial configuration. The dynamic (sequential) behaviour of a P/T system is now defined completely analogous to that of an EN system, taking into consideration the difference between a net and a multi-net (i.e., the weight and capacity function). All terminology is carried over as much as possible.

Definition 155. Let $M = (P, T, F, W, K, C_{in})$ be a P/T system and let $t \in T$.

(1) Let $C : P \rightarrow \mathbb{N}$ be a configuration of M . Then t has concession in C (or t can be fired in C , or t is enabled in C), written as $t \mathbf{con} C$, if:

$$\forall p \in \bullet t: W(p, t) \leq C(p) \text{ and}$$

$$\forall p \in t^\bullet: \text{if } K(p) \in \mathbb{N}_+, \text{ then } C(p) + W(t, p) \leq K(p).$$

(2) Let C and D be configurations of M . Then t fires from C to D , written as $C[t]D$, if $t \mathbf{con} C$ and, for every $p \in P$,

$$D(p) = C(p) - W(p, t) \quad \text{if } p \in \bullet t,$$

$$D(p) = C(p) + W(t, p) \quad \text{if } p \in t^\bullet, \text{ and}$$

$$D(p) = C(p) \quad \text{otherwise.}$$

Note that the case distinction in the definition of $D(p)$ is allowed, because $\bullet t \cap t^\bullet = \emptyset$ according to Definition 1(4).

Definition 8 for EN systems can now be carried over literally to P/T systems (except that now $M = (P, T, F, W, K, C_{in})$, and configurations are functions from P to \mathbb{N} rather than subsets of P). In this way we obtain, e.g., the definition of the set $\text{FS}(M)$ of firing sequences of M , of the set \mathbb{C}_M of reachable configurations of M , of the set $\mathbf{use}_M(T)$ of useful transitions of M , and of liveness of transitions.

Example 45. (1) Consider the P/T system of Fig. 79. In the initial configuration C_{in} only t_2 has concession. Firing t_2 leads to the configuration C with $C(p_2) = 0$, $C(p_3) = 1$, and $C(p) = C_{in}(p)$ for the remaining places p . In C only t_1 has concession. $\text{FS}(M)$ contains, e.g., for every n the firing sequence $(t_2 t_1 t_3 t_1 t_3)^{3n} t_4^{2n}$; this firing sequence leads from C_{in} to the reachable configuration D_n , with $D_n(p_1) = 0$, $D_n(p_2) = 2$, $D_n(p_3) = n$, $D_n(p_4) = 1$, $D_n(p_5) = 0$, and $D_n(p_6) = 4n$.

(2) Consider the P/T systems in Figs. 4, 76, and 77. In all these systems $K(p) = \omega$ for every place p . In Figs. 4 and 76 the buffer b can contain arbitrarily many tokens: if the (or a) producer performs its cycle n times (and the consumers remain idle), then b contains n tokens. By taking, e.g., $K(b) = 5$ instead of $K(b) = \omega$, the capacity of the buffer is limited to 5. In Fig. 77 $K(b) = \omega$, but still there can only be a maximum of 5 tokens in b .

Completely analogous to Definition 11 for EN systems we can also define the sequential configuration graph $\text{SCG}(M)$ of a P/T system M , the only difference being that $\text{SCG}(M)$ is now an infinite (or finite) edge-labelled graph (defined as in Definition 9 but without the requirement that the set of nodes is finite). This possible infiniteness of $\text{SCG}(M)$ has the consequence that it is no longer as easy as for EN systems to verify whether or not a given configuration is reachable and whether or not a given transition is useful/live (however, these problems are decidable). Naturally, it *is* easy to verify whether a given $x \in T^*$ is a firing sequence of M . Note that Theorem 12 does not hold for P/T systems, as the following example shows.

Example 46. Consider the P/T system M in Fig. 12 (producer/consumer). It is easy to see that $\text{FS}(M) \cap (pf)^*(ec)^* = \{(pf)^n(ec)^m \mid 0 \leq m \leq n + 1\}$. Since this is not a regular language, $\text{FS}(M)$ is not regular. Similarly, an example of a P/T system M for which $\text{FS}(M)$ is not context-free can be presented.

Since there is a bijective correspondence between subsets of a set P and (characteristic) functions $P \rightarrow \{0, 1\}$, configurations C of a P/T system for which $C : P \rightarrow \{0, 1\}$ can also be considered as configurations of an EN system. Then it is easy to verify, using the definitions of $t \text{ con } C$ and $C[t]D$ for EN systems and for P/T systems, that EN systems are a special case of P/T systems.

Definition 156. (alternative) An *EN system* is a P/T system M with $K_M(p) = 1$ for all $p \in P_M$.

Note that, with this alternative definition, an EN system M can still have a transition t and a place p with $W_M(p, t) \geq 2$ or $W_M(t, p) \geq 2$. However, since all places have capacity 1, such a transition t is not useful. Hence we could directly assume (as normal form) for EN systems that $W(x, y) \leq 1$ for all $(x, y) \in F$; in this way we would get “exactly” the old definition of an EN system.

A large number of facts for EN systems can be generalized to P/T systems, but we will not do this all. We first note that Theorems 36 and 37 also hold for P/T systems: for each P/T system M there exists a configuration equivalent strongly reduced P/T system M' . The proof is the same, i.e., useless transitions and isolated places can simply be omitted. We will see later that there is an algorithm to verify whether a transition is useful.

Furthermore we note that P/T systems can also be made contact-free (cf. Theorem 60). A P/T system has contact if a transition has input-concession but does not have output-concession, because firing the transition would lead to exceeding the capacity of one of its output places.

Definition 157. Let $M = (P, T, F, K, W, C_{in})$ be a P/T system. M is *contact-free* if for all $t \in T$ and $C \in \mathbb{C}_M$, if $W(p, t) \leq C(p)$ for all $p \in \bullet t$, then $C(p) + W(t, p) \leq K(p)$ for all $p \in t^\bullet$ with $K(p) \in \mathbb{N}_+$.

Making a P/T system contact-free is analogous to the complement construction in Theorem 57. For a given place p_0 with $K(p_0) \in \mathbb{N}_+$ we add to M a new place q_0 with $K(q_0) = K(p_0)$ (but more is also allowed), $C_{in}(q_0) = K(p_0) - C_{in}(p_0)$, $\bullet q_0 = p_0^\bullet$, $q_0^\bullet = \bullet p_0$ and, for all $t \in T$, $W(q_0, t) = W(t, p_0)$ and $W(t, q_0) = W(p_0, t)$. Then for every reachable configuration C in the new system $C(q_0) = K(p_0) - C(p_0)$, i.e., q_0 contains the remaining capacity of p_0 . Then there is no risk anymore of exceeding the capacity of p_0 (and of q_0). As an example, the P/T system of Fig. 77 has been constructed in this way from the P/T system of Fig. 12 with $K(b) = 5$.

For this reason we will from now on consider contact-free P/T systems only. Note that every P/T system M with $K_M(p) = \omega$ for all $p \in P_M$ is contact-free. The other way around we can assume just as well that, for a contact-free P/T system M , $K_M(p) = \omega$ for all $p \in P_M$ (without changing the behaviour of the system). In other words: **from now on we assume for every P/T system M that $K_M(p) = \omega$ for all $p \in P_M$, and we denote M by (P, T, F, W, C_{in}) .** Note that a configuration is now just a function $C : P \rightarrow \mathbb{N}$ and that $t \text{ con } C \Leftrightarrow \forall p \in \bullet t : W(p, t) \leq C(p)$.

Contact-free EN systems now correspond to those P/T systems that contain at most one token in every place of a reachable configuration.

Definition 158. A P/T system $M = (P, T, F, W, C_{in})$ is *safe* if $C(p) \in \{0, 1\}$ for all $C \in \mathbb{C}_M$ and $p \in P$.

Definition 159. (alternative) A *contact-free EN system* is a safe P/T system.

We now introduce some terminology for configurations of a P/T system $M = (P, T, F, W, C_{in})$, based on the fact that a configuration $C : P \rightarrow \mathbb{N}$ of M actually is a vector in \mathbb{N}^n , with $n = \#P$. When we fix an ordering (p_1, \dots, p_n) of the elements of P , C corresponds to the vector (v_1, \dots, v_n) with $v_i = C(p_i)$ for all $1 \leq i \leq n$. For later usage we also introduce this terminology for vectors in \mathbb{Z}^n , i.e., for functions $C : P \rightarrow \mathbb{Z}$ (which can be viewed as multi-sets in which places can also occur negatively).

First we define a *partial order* for vectors. For vectors $C, D : P \rightarrow \mathbb{Z}$, $C \leq D$ iff $C(p) \leq D(p)$ for every $p \in P$ (this is the generalization of set inclusion to multi-sets). We write $C < D$ for $C \leq D$ and $C \neq D$ (in other words: $C \leq D$ and there exists a p for which $C(p) < D(p)$). It is easy to see that $<$ is a partial order on the set of vectors $P \rightarrow \mathbb{Z}$ (and also on the set of vectors $P \rightarrow \mathbb{N}$, i.e., the set of configurations of M).

Furthermore, the usual operations on vectors are defined. For example, the *sum* of two vectors C and D is the vector $C + D$ defined by $(C + D)(p) = C(p) + D(p)$ for every $p \in P$. Also, for $i \in \mathbb{Z}$, iC is the vector with $(iC)(p) = i \cdot C(p)$, i.e., the usual *scalar multiplication* of C by i .

9.3 A Finiteness Algorithm

Though the configuration graph of a P/T system can be infinite, there are a large number of properties of P/T systems for which algorithms exist that decide whether or not these properties hold. To get an idea of this kind of (quite complicated) algorithms we will treat in this section an algorithm to decide, for a given P/T system M , whether or not \mathbb{C}_M is finite, i.e., whether or not M 's sequential configuration graph $\text{SCG}(M)$ is finite. This section and the next are based on [Rei82, Fin90].

We first give a characterization of infiniteness of \mathbb{C}_M . It is easy to give a necessary condition, based on the following fundamental lemmas.

Lemma 160. *Let $M = (P, T, F, W, C_{in})$ be a P/T system, $x \in T^*$, and C, D, E configurations of M . If $C[x]D$, then $C + E[x]D + E$.*

Note that this lemma does not hold for P/T systems with a capacity function (and neither for EN systems): the addition of tokens (by E) can cause capacity overflow.

Lemma 161. *Let $M = (P, T, F, W, C_{in})$ be a P/T system, $x \in T^*$, and C, D configurations of M . If $C[x]D$ and $C \leq D$ then, for every $n \in \mathbb{N}$, $C[x^n]D_n$, where $D_n(p) = C(p) + n \cdot (D(p) - C(p))$ for every $p \in P$.*

Proof. Let E be the configuration with $E(p) = D(p) - C(p)$ for all $p \in P$. Then $C + E = D$. Now, by Lemma 160 and with induction on n , for every $n \in \mathbb{N}$, $C[x^n]C + nE$. Note that $D_n = C + nE$. \square

The necessary condition can now be stated as follows.

Lemma 162. *Let $M = (P, T, F, W, C_{in})$ be a P/T system, $x \in T^*$, and $C, D \in \mathbb{C}_M$. If $C[x]D$ and $C < D$, then \mathbb{C}_M is infinite.*

Proof. Directly from Lemma 161: $D_n \in \mathbb{C}_M$. And since there is at least one $p \in P$ such that $C(p) < D(p)$, all D_n are different. \square

Example 47. (1) Consider the P/T system of Fig. 12. Let $C_{in}[pf]D$. Then $C_{in} < D$. Repeatedly firing pf thus leads to infinitely many reachable configurations.

(2) Consider the P/T system that is obtained from Fig. 79 by first complementing place p_2 and then turning all capacities into ω . Let $C_{in}[t_2t_1t_3t_1t_3]D$. Then $C_{in} < D$. Compare this to Example 45(1).

We want to show that Lemma 162 is the only reason that \mathbb{C}_M can be infinite, i.e., that the lemma also holds the other way around. This leads to the following theorem.

Theorem 163. *Let $M = (P, T, F, W, C_{in})$ be a P/T system. Then \mathbb{C}_M is infinite iff there exist $x \in T^*$ and $C, D \in \mathbb{C}_M$ such that $C[x]D$ and $C < D$.*

The if-part of this theorem is Lemma 162. To prove the only-if-part we need two basic lemmas: Dickson's Lemma (cf. [Kru72]) and König's Lemma. First we prove Dickson's Lemma (in which we only use that configurations are vectors over \mathbb{N}). We say that an infinite sequence of configurations C_1, C_2, C_3, \dots of a P/T system is *increasing* if $C_i \leq C_{i+1}$ for all $i \geq 1$.

Lemma 164. (*Dickson's Lemma*) *Every infinite sequence of configurations of a P/T system has an infinite increasing subsequence.*

Proof. By induction on n we prove that every infinite sequence of vectors in \mathbb{N}^n contains an infinite increasing subsequence. We start with the case $n = 1$. Consider an infinite sequence of natural numbers a_1, a_2, \dots with $a_i \in \mathbb{N}$ for all $i \geq 1$. If there is a number that occurs infinitely many times in the sequence, then we are done. If this is not the case, then for every k there exists an $m > k$ such that $a_k \leq a_m$ (because there are only finitely many natural numbers smaller than a_k and each of these occurs only finitely often). This implies that there exists a subsequence $a_{i_1} \leq a_{i_2} \leq a_{i_3} \leq \dots$ (with for example $i_1 = 1$).

Now assume that it is proved for $n - 1$ and consider the case n . Let $a_1 b_1, a_2 b_2, a_3 b_3, \dots$ be an infinite sequence of elements of \mathbb{N}^n , with $a_i \in \mathbb{N}$ and $b_i \in \mathbb{N}^{n-1}$. According to the case $n = 1$, proved above, there is an infinite increasing subsequence $a_{i_1} \leq a_{i_2} \leq a_{i_3} \leq \dots$ of a_1, a_2, a_3, \dots . From the induction hypothesis follows that the infinite sequence $b_{i_1}, b_{i_2}, b_{i_3}, \dots$ contains an infinite increasing subsequence: $b_{i_{j_1}} \leq b_{i_{j_2}} \leq b_{i_{j_3}} \leq \dots$. Then the sequence $a_{i_{j_1}} b_{i_{j_1}}, a_{i_{j_2}} b_{i_{j_2}}, a_{i_{j_3}} b_{i_{j_3}}, \dots$ is an infinite increasing subsequence of $a_1 b_1, a_2 b_2, a_3 b_3, \dots$ \square

To formulate König's Lemma we use the following definition.

Definition 165. Let Σ be an alphabet and $L \subseteq \Sigma^*$ a language over Σ . L is *prefix-closed* if for every $x \in L$ also all prefixes of x are element of L .

Note that, for a P/T system M , $\text{FS}(M)$ is a prefix-closed language over T_M (see the remarks following Theorem 12). Intuitively, König's Lemma means that an infinite tree of which each node has finitely many children, has an infinite path.

Lemma 166. (*König's Lemma*) *If $L \subseteq \Sigma^*$ is an infinite prefix-closed language, then there is an infinite sequence t_1, t_2, t_3, \dots with $t_i \in \Sigma$, such that, for all $n \geq 0$, $t_1 \dots t_n \in L$.*

Proof. Choose $t_1 \in \Sigma$ such that $L \cap t_1 \Sigma^*$ is infinite; such a t_1 must exist because L is infinite and because L is the union of the (finitely many) sets $L \cap t \Sigma^*$, with $t \in \Sigma$. Since L is prefix-closed, $t_1 \in L$. If t_1, \dots, t_{n-1} are chosen such that $L \cap (t_1 \dots t_{n-1} \Sigma^*)$ is infinite, then choose t_n such that $L \cap (t_1 \dots t_{n-1} t_n \Sigma^*)$ is infinite. \square

We now prove the only-if-part of Theorem 163. Let $M = (P, T, F, W, C_{in})$ be a P/T system and assume \mathbb{C}_M to be infinite.

A firing sequence $t_1 \cdots t_n$, with $t_i \in T$, is *acyclic* if C_{in}, C_1, \dots, C_n are all different, where $C_{in}[t_1 \cdots t_i]C_i$ for all $1 \leq i \leq n$. We denote the set of all acyclic firing sequences of M by $\text{AFS}(M)$. Note that an acyclic firing sequence corresponds to an acyclic path in $\text{SCG}(M)$. Because every reachable configuration of M can be reached via an acyclic firing sequence, also $\text{AFS}(M)$ is infinite. Since $\text{AFS}(M)$ is prefix-closed, Lemma 166 implies that there exists an infinite sequence of transitions t_1, t_2, t_3, \dots such that $t_1 \cdots t_n \in \text{AFS}(M)$ for all n . Intuitively this is an infinite acyclic firing sequence of M . Let $C_{in}[t_1 \cdots t_n]C_n$ and consider the infinite sequence of configurations C_1, C_2, C_3, \dots . Since $t_1 \cdots t_n \in \text{AFS}(M)$, all C_1, C_2, C_3, \dots are different. By Lemma 164 there exists an infinite increasing subsequence of C_1, C_2, C_3, \dots . Hence there certainly exist configurations C_i and C_j , with $i < j$, such that $C_i \leq C_j$. Since all configurations in the sequence are different, $C_i < C_j$. Now let $x = t_{i+1} \cdots t_j$, $C = C_i$ and $D = C_j$. Then $C[x]D$ and $C < D$.

On the basis of Theorem 163 we now present an algorithm to decide finiteness of \mathbb{C}_M .

Theorem 167. *It is decidable, for an arbitrary P/T system M , whether \mathbb{C}_M is finite or infinite.*

Proof. Let $M = (P, T, F, W, C_{in})$. We try to find, in a systematical way, configurations C and D in \mathbb{C}_M and a firing sequence x that satisfy the requirements of Theorem 163. For this purpose we construct, step by step, the sequential configuration graph $\text{SCG}(M)$ of M , meanwhile continuously verifying whether we have already found C , D and x . If \mathbb{C}_M is finite, then the algorithm will halt with the (finite) sequential configuration graph and if \mathbb{C}_M is infinite, then the algorithm will halt as soon as we find C , D and x .

The algorithm uses a (finite) edge-labelled graph $G = (V, \Gamma, \Sigma, v_{in})$ with $V \subseteq \mathbb{C}_M$, $v_{in} = C_{in}$, $\Sigma \subseteq \mathbf{use}(T)$, and $\Gamma \subseteq \{(C, t, D) \mid C[t]_M D\}$ as variable. Hence G is a finite subgraph of $\text{SCG}(M)$. The algorithm works as follows.

- (1) Initialization: $V := \{C_{in}\}$, $\Gamma := \emptyset$, $\Sigma := \emptyset$ and $v_{in} := C_{in}$.
- (2) For all $C \in V$ and $t \in T$ do:
if $C[t]D$ then $V := V \cup \{D\}$, $\Gamma := \Gamma \cup \{(C, t, D)\}$, and $\Sigma := \Sigma \cup \{t\}$.
- (3) If G did not change in Step (2), then \mathbb{C}_M is finite and $G = \text{SCG}(M)$.
If G did change in Step (2), then test
whether there are $C, D \in V$ such that: $C < D$ and there is a path in G from C to D .
If this is the case, then \mathbb{C}_M is infinite. If not, then goto (2).

We conclude with some remarks on the correctness of the algorithm.

If \mathbb{C}_M is finite, then according to Theorem 163 the (second) test in Step (3) is always negative. If in $\text{SCG}(M)$ every $C \in \mathbb{C}_M$ is reachable in $\leq k$ steps, then after $\leq k$ iterations of Step (2) $V = \mathbb{C}_M$, after $\leq k + 1$ iterations $\Gamma = \{(C, t, D) \mid C, D \in \mathbb{C}_M \text{ and } C[t]D\}$ and $\Sigma = \mathbf{use}(T)$, and after $\leq k + 2$ iterations G no longer changes and the algorithm halts with $G = \text{SCG}(M)$.

If \mathbb{C}_M is infinite, then, by Theorem 163, there exist $C, D \in \mathbb{C}_M$ and $x \in T^*$ such that $C[x]D$ and $C < D$. Then after $|x|$ iterations of Step (2) the test in Step (3) becomes negative and the algorithm halts. \square

Example 48. (1) Consider the P/T system M that is given in Fig. 80. The algorithm then constructs the edge-labelled graph in Fig. 81 where a configuration C is indicated as the vector $(C(p_1), C(p_2), C(p_3), C(p_4))$. After the 4-th iteration of Step (2) the algorithm finds, in the test of Step (3), $(3, 6, 1, 0)[t_1 t_1 t_2 t_3](6, 6, 1, 0)$ and $(3, 6, 1, 0) < (6, 6, 1, 0)$, and concludes that \mathbb{C}_M is infinite.

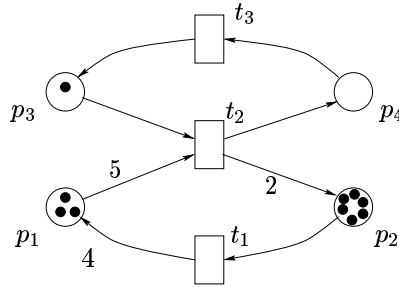


Fig. 80. A P/T system M with infinite $\text{SCG}(M)$.

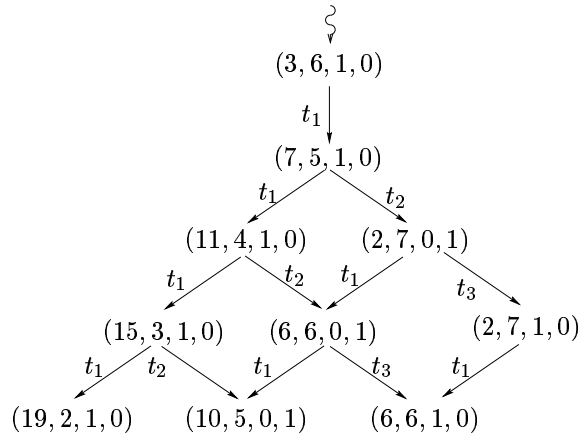


Fig. 81. The computed part of $\text{SCG}(M)$, of the P/T system of Fig. 80.

(2) Now consider the same P/T system M of Fig. 80 with the following

be computed. For this we use vectors $C : P \rightarrow \mathbb{N} \cup \{\omega\}$ which we will call ω -configurations of M . For ω -configurations C_1, C_2 , $C_1 \leq C_2$ is defined as for usual vectors, where $n \leq \omega$ for all $n \in \mathbb{N} \cup \{\omega\}$. Furthermore we define, for an ω -configuration C , $\Omega(C) = \{p \in P \mid C(p) = \omega\}$, the set of ω -places of C . Note that C is a (usual) configuration iff $\Omega(C) = \emptyset$. For ω -configurations $C, D : P \rightarrow \mathbb{N} \cup \{\omega\}$ and $t \in T$ we define $t \mathbf{con} C$ and $C[t]D$ in the same way as for configurations in Definition 155, where $\omega + n = \omega - n = \omega$ for all $n \in \mathbb{N}$ (intuitively $C(p) = \omega$ means that p contains infinitely many tokens). Note that if $C[t]D$, then $\Omega(C) = \Omega(D)$.

We now define covering sets.

Definition 170. Let $M = (P, T, F, W, C_{in})$ be a P/T system. A *covering set* of M is a finite set \mathbb{V} of ω -configurations for which:

- (1) for every $C \in \mathbb{C}_M$ there exists $D \in \mathbb{V}$ such that $C \leq D$, and
- (2) for every $D \in \mathbb{V}$ and every $n \in \mathbb{N}$ there exists $C \in \mathbb{C}_M$ such that $\forall p \in P : C(p) = D(p)$ if $D(p) \neq \omega$, and $C(p) \geq n$ if $D(p) = \omega$.

If A is a set of ω -configurations, then $(A, <)$ is a partially ordered set. Let A° denote the set of maximal elements of A , even if A is infinite (see Definition 69).

Lemma 171. *If \mathbb{V} is a covering set of a P/T system M , then so is \mathbb{V}° .*

Proof. Every subset of \mathbb{V} also satisfies requirement (2) of Definition 170. Hence it suffices to prove that for each $C \in \mathbb{V}$ there exists a $D \in \mathbb{V}^\circ$ with $C \leq D$. This follows from Lemma 70: $\mathbb{V} \subseteq \rightarrow(\mathbb{V}^\circ)$. \square

Theorem 172. *There is an algorithm that constructs a covering set for each P/T system M .*

In what follows we prove this theorem. For a P/T system $M = (P, T, F, W, C_{in})$ we first define a specific set \mathbb{V}_M of ω -configurations and then show that \mathbb{V}_M is a covering set of M (and hence \mathbb{V}_M° too). This \mathbb{V}_M is defined on the basis of a prefix-closed set $L \subseteq T^*$ and a function γ that associates an ω -configuration $\gamma(x)$ with each $x \in L$; then \mathbb{V}_M is $\{\gamma(x) \mid x \in L\}$. The pair (L, γ) is called the *covering tree* of M , since informally L and γ can be drawn as a (node- and edge-labelled) tree, in the following way: the elements of L are the nodes of the tree; if $x \in L$ and $xt \in L$ (with $x \in T^*$ and $t \in T$), then there is an edge from x to xt labelled by t ; λ is the root of the tree; if $x \in L$, then $\gamma(x)$ is the label of node x . Note that the names of the nodes can be omitted, because for every node x the sequence of labels on the path from the root to x equals x . Note also that, in this tree, node u is an ancestor of node x iff u is a prefix of x . A priori the covering tree can be infinite (i.e., L can be infinite), but we will prove that it is always finite.

Definition 173. Let $M = (P, T, F, W, C_{in})$ be a P/T system. The *covering tree* of M is the pair (L, γ) , recursively defined in (1) and (2).

- (1) $\lambda \in L$ and $\gamma(\lambda) = C_{in}$.

- (2) If $x \in L$, $t \in T$, $\gamma(x)[t]D$, and there is no prefix u of x with $D \leq \gamma(u)$, then $xt \in L$ with $\gamma(xt)$ as defined in (3).
- (3) If there is no prefix u of x such that both $\gamma(u) < D$ and $\Omega(\gamma(u)) = \Omega(D)$, then $\gamma(xt) = D$, otherwise $\gamma(xt) = D'$, as defined in (4).
- (4) Let u be the longest prefix of x such that $\gamma(u) < D$ and $\Omega(\gamma(u)) = \Omega(D)$. Let $\gamma(u) = C$ and thus $C < D$ and $\Omega(C) = \Omega(D)$. Then, for all $p \in P$,

$$D'(p) = \begin{cases} D(p) & \text{if } C(p) = D(p) \\ \omega & \text{if } C(p) < D(p). \end{cases}$$

Finally $\mathbb{V}_M = \{\gamma(x) \mid x \in L\}$.

We observe that it is irrelevant that we take the longest prefix in (4); we could just as well have taken, e.g., the shortest one.

It is clear that the covering tree can be computed systematically in a top-down fashion, if it is finite (and this we will prove).

Example 49. The covering tree of the P/T system M_1 of Fig. 83 is given in Fig. 84. The covering tree of the P/T system M_2 of Fig. 85 is given in Fig. 86.

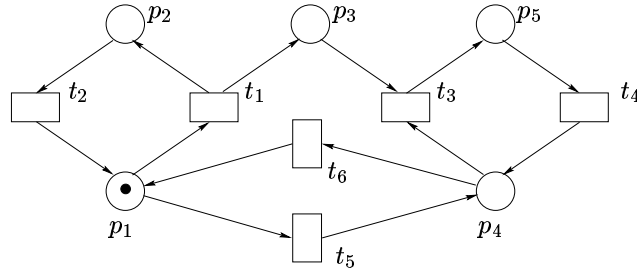


Fig. 83. P/T system M_1 .

The following properties of the covering tree can easily be seen.

Lemma 174. *Let (L, γ) be the covering tree of a P/T system M and let $t \in T_M$.*

- (1) L is prefix-closed.
- (2) If $x \in L$ and $\gamma(x)[t]D$, then there exists $y \in L$ such that $D \leq \gamma(y)$.
- (3) If $x \in L$ and $xt \in L$, with $\gamma(x) = C$ and $\gamma(xt) = D$, then $\Omega(C) \subseteq \Omega(D)$ and $\Omega(C) = \Omega(D) \Leftrightarrow C[t]D$.

Proof. (2) If there is a prefix u of x with $D \leq \gamma(u)$, then $y = u$, and otherwise $D \leq \gamma(xt)$ and thus $y = xt$. \square

We now prove that the covering tree is finite.

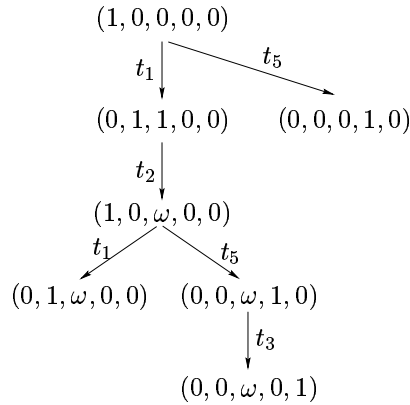


Fig. 84. The covering tree of P/T system M_1 of Fig. 83.

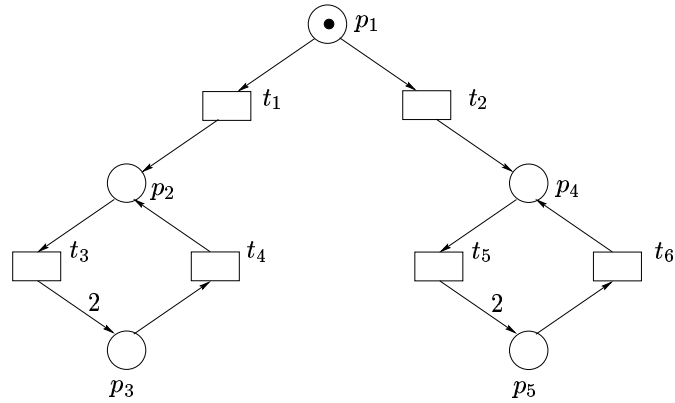


Fig. 85. P/T system M_2 .

Lemma 175. Let (L, γ) be the covering tree of a P/T system M . Then L is finite.

Proof. Suppose that L is infinite. Since L is prefix-closed, according to König's Lemma (Lemma 166) there exists an infinite sequence of transitions t_1, t_2, \dots such that $t_1 \cdots t_n \in L$ for every $n \in \mathbb{N}$. Let $C_n = \gamma(t_1 \cdots t_n)$. By Lemma 174(3) and the finiteness of P_M , there exists $k \in \mathbb{N}$ such that for all $m, n \geq k$: $\Omega(C_m) = \Omega(C_n)$. It is not difficult to see that Dickson's Lemma (Lemma 164) also holds for ω -configurations. Hence there is an infinite increasing subsequence of the sequence C_k, C_{k+1}, \dots . Thus there definitely exist C_i and C_j with $i < j$, $C_i \leq C_j$, and $\Omega(C_i) = \Omega(C_j)$. Then Lemma 174(3) implies $C_{j-1}[t_j]C_j$. Now consider

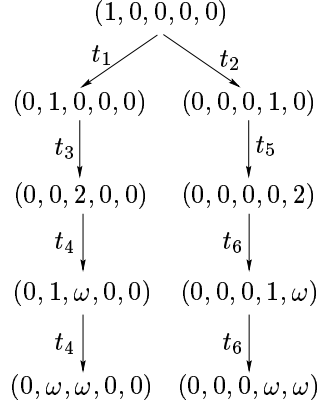


Fig. 86. The covering tree of P/T system M_2 of Fig. 85.

Definition 173, with $x = t_1 \cdots t_{j-1}$, $t = t_j$, $\gamma(x) = C_{j-1}$, and $D = C_j$. From the fact that $xt \in L$ and $\gamma(x)[t]D$ it follows (by Definition 173(2)) that there is no prefix u of x with $C_j \leq \gamma(u)$. For $u = t_1 \cdots t_i$ this implies that $C_j \neq C_i$ and thus $C_i < C_j$. Then u is a prefix of x such that $\gamma(u) = C_i < C_j = D$ and $\Omega(\gamma(u)) = \Omega(D)$. Hence, according to Definition 173(3,4), $\gamma(xt) \neq D$ (because $\Omega(D)$ is a proper subset of $\Omega(\gamma(xt))$). This contradicts the fact that $\gamma(xt) = C_j$. \square

We now show that \mathbb{V}_M is a covering set, i.e., has properties (1) and (2) of Definition 170. In connection with property (2) we first introduce the following terminology and prove a lemma.

Definition 176. An ω -configuration D of a P/T system M is a *covering vector* of M if for every $n \in \mathbb{N}$ there exists a $C \in \mathbb{C}_M$ such that, for all $p \in P$, $C(p) = D(p)$ if $D(p) \neq \omega$, and $C(p) \geq n$ if $D(p) = \omega$.

The set of all covering vectors of M is denoted by \mathbb{O}_M . Note that $\mathbb{C}_M \subseteq \mathbb{O}_M$. Also note that a covering set is the same as a finite subset of \mathbb{O}_M that satisfies requirement (1) of Definition 170.

Lemma 177. Let $M = (P, T, F, W, C_{in})$ be a P/T system. For all ω -configurations D_1, D_2 of M and $t \in T$, if $D_1[t]D_2$ and $D_1 \in \mathbb{O}_M$, then $D_2 \in \mathbb{O}_M$.

Proof. Define $m = \max\{W(p, t) \mid (p, t) \in F\}$. Since $D_1 \in \mathbb{O}_M$, for a given $n \in \mathbb{N}$ there exists a $C_1 \in \mathbb{C}_M$ such that for all $p \in P$: $C_1(p) = D_1(p)$ if $D_1(p) \neq \omega$, and $C_1(p) \geq n + m$ if $D_1(p) = \omega$. Then there exists (by the choice of m) a C_2 with $C_1[t]C_2$ and for all $p \in P$: $C_2(p) = D_2(p)$ if $D_2(p) \neq \omega$ and $C_2(p) \geq n$ if $D_2(p) = \omega$. This implies $D_2 \in \mathbb{O}_M$. \square

Theorem 178. *For every P/T system M , \mathbb{V}_M is a covering set.*

Proof. We prove that \mathbb{V}_M satisfies (1) and (2) of Definition 170. Let $M = (p, T, F, W, C_{in})$ and let (L, γ) be the covering tree of M .

(1) For every $C \in \mathbb{C}_M$ there exists $D \in \mathbb{V}_M$ with $C \leq D$.

We prove this by induction on C . It holds for $C = C_{in}$ since $C_{in} \in \mathbb{V}_M$ by Definition 173(1). Now consider $C \in \mathbb{C}_M$ and $C[t]C'$, and assume that $D \in \mathbb{V}_M$ and $C \leq D$. Determine the ω -configuration E with $C + E = D$. It is easy to verify that Lemma 160 also holds for ω -configurations (with $\omega + \omega = \omega$). Hence $D[t]C' + E$. Then according to Lemma 174(2) there exists $D' \in \mathbb{V}_M$ such that $C' + E \leq D'$. And thus $C' \leq D'$.

(2) $\mathbb{V}_M \subseteq \mathbb{O}_M$.

We thus must prove that $\gamma(x)$ is a covering vector of M for every $x \in L$. We do this by induction on $|x|$. It is clear for $x = \lambda$ (since $\gamma(\lambda) = C_{in} \in \mathbb{C}_M$). Now take $x \in L$ and assume (as induction hypothesis) that $\gamma(u) \in \mathbb{O}_M$ for every prefix u of x . Now consider a $t \in T$ and a D such that $\gamma(x)[t]D$ and there does not exist a prefix u of x with $D \leq \gamma(u)$. It remains to prove that $\gamma(xt) \in \mathbb{O}_M$. If $\gamma(xt) = D$ (see Definition 173(3)), then $\gamma(xt) \in \mathbb{O}_M$ by Lemma 177.

* The other possibility (see Definition 173(4)) is that there is a prefix u of x , such that for $\gamma(u) = C$: $C < D$, $\Omega(C) = \Omega(D)$, and $\gamma(xt) = D'$ with for every $p \in P$:

$$D'(p) = D(p) \text{ if } C(p) = D(p), \text{ and } D'(p) = \omega \text{ if } C(p) < D(p).$$

From $\Omega(C) = \Omega(D)$ and Lemma 174(3) follows that $C[y]D$, where $xt = uy$. Now take an arbitrary $n \in \mathbb{N}$. Since $C < D$ and $C[y]D$, Lemma 161 (for ω -configurations) implies that $C[y^n]D_n$ for some D_n with, for every $p \in P$:

$$D_n(p) = D(p) \text{ if } C(p) = D(p), \text{ and } D_n(p) \geq n \text{ if } C(p) < D(p).$$

By the induction hypothesis it follows that $C \in \mathbb{O}_M$ and thus Lemma 177 implies that $D_n \in \mathbb{O}_M$. Hence there exists $C_n \in \mathbb{C}_M$ such that

$$C_n(p) = D_n(p) \text{ if } D_n(p) \neq \omega, \text{ and } C_n(p) \geq n \text{ if } D_n(p) = \omega.$$

Then also

$$C_n(p) = D'(p) \text{ if } D'(p) \neq \omega, \text{ and } C_n(p) \geq n \text{ if } D'(p) = \omega.$$

In fact, in the case that $C(p) = D(p)$ it follows from the previous line because $D'(p) = D(p) = D_n(p)$, and in the case that $C(p) < D(p)$ it follows from $D'(p) = \omega$ and $D_n(p) \geq n$ (because $D_n(p) \geq n$ implies $C_n(p) \geq n$). This proves that $D' \in \mathbb{O}_M$. \square

All of this proves Theorem 172. According to Lemma 171 also \mathbb{V}_M° is a covering set. We now additionally show that, for all covering sets \mathbb{V} , the covering sets \mathbb{V}° are all the same. We begin with a lemma that generalizes Definition 170(1).

Lemma 179. *Let \mathbb{V} be a covering set of a P/T system M . For every $C \in \mathbb{O}_M$ there exists $D \in \mathbb{V}$ such that $C \leq D$.*

Proof. Since $C \in \mathbb{O}_M$, there exist configurations $C_n \in \mathbb{C}_M$, for every $n \in \mathbb{N}$, such that for all $p \in P$, $C_n(p) = C(p)$ if $C(p) \neq \omega$ and $C_n(p) \geq n$ if $C(p) = \omega$. According to Definition 170(1) there exist ω -configurations $D_n \in \mathbb{V}$ with $C_n \leq D_n$. Since \mathbb{V} is finite, there exist a $D \in \mathbb{V}$ and an infinite subset I of \mathbb{N} such that $D_n = D$ for all $n \in I$. Hence $D \geq C_n$ for all $n \in I$ and thus, for all $p \in P$, $D(p) \geq C(p)$ if $C(p) \neq \omega$ and $D(p) \geq n$ if $C(p) = \omega$. Hence if $C(p) = \omega$, then $D(p) = \omega$. This implies that $D \geq C$. \square

Theorem 180. *For every covering set \mathbb{V} of a P/T system M , $\mathbb{V}^\circ = \mathbb{O}_M^\circ$.*

Proof. ($\mathbb{V}^\circ \subseteq \mathbb{O}_M^\circ$). Let $C \in \mathbb{V}^\circ$. Then $C \in \mathbb{O}_M$ because $\mathbb{V} \subseteq \mathbb{O}_M$. Now it remains to prove that C is a maximal element of \mathbb{O}_M . Let $D \in \mathbb{O}_M$ and $C \leq D$. We have to prove that $C = D$. By Lemma 179 there exists $E \in \mathbb{V}$ with $D \leq E$. Hence $C \leq E$. Since C is maximal in \mathbb{V} , $C = E$ and thus $C = D$.

($\mathbb{O}_M^\circ \subseteq \mathbb{V}^\circ$). Let $C \in \mathbb{O}_M^\circ$. By Lemma 179 there exists $D \in \mathbb{V}$ with $C \leq D$. Since C is maximal in \mathbb{O}_M and $D \in \mathbb{O}_M$, $C = D$. Hence $C \in \mathbb{V}$. Since C is maximal in \mathbb{O}_M , it is certainly maximal in \mathbb{V} (because $\mathbb{V} \subseteq \mathbb{O}_M$). \square

This implies that $\mathbb{V}_M^\circ = \mathbb{O}_M^\circ$ is the (unique) smallest covering set of M : firstly \mathbb{V}_M° is a covering set of M and secondly (by Theorem 180) $\mathbb{O}_M^\circ \subseteq \mathbb{V}$ for every covering set \mathbb{V} of M .

We now present several applications of Theorem 172. First we show that now we can also verify for each place independently whether it is bounded (see Definition 168).

Definition 181. A place p of a P/T system M is *bounded* if there exists a $k \in \mathbb{N}$ such that $C(p) \leq k$ for every $C \in \mathbb{C}_M$.

Note that a P/T system is bounded iff all its places are bounded.

Theorem 182. *It is decidable, for an arbitrary P/T system M and an arbitrary place $p \in P_M$, whether or not p is bounded.*

Proof. If \mathbb{V} is a covering set of M , then: p is unbounded iff there exists a $C \in \mathbb{V}$ such that $C(p) = \omega$. Verify this yourself from Definition 170. \square

Example 50. (1) Consider the P/T system M_1 of Fig. 83, with the covering tree in Fig. 84. Then $\mathbb{V}_{M_1} = \{(1, 0, 0, 0, 0), (0, 0, 0, 1, 0), (0, 1, 1, 0, 0), (1, 0, \omega, 0, 0), (0, 1, \omega, 0, 0), (0, 0, \omega, 1, 0), (0, 0, \omega, 0, 1)\}$ is a covering set of M_1 . This implies that p_3 is unbounded and all other places are bounded (with bound 1). The unique smallest covering set of M_1 is $\mathbb{V}_{M_1}^\circ = \{(1, 0, \omega, 0, 0), (0, 1, \omega, 0, 0), (0, 0, \omega, 1, 0), (0, 0, \omega, 0, 1)\}$.

(2) From Figs. 85 and 86 one can see that p_1 is bounded and that all other places are unbounded. Here $\mathbb{V}_{M_2}^\circ = \{(1, 0, 0, 0, 0), (0, \omega, \omega, 0, 0), (0, 0, 0, \omega, \omega)\}$.

(3) Consider the P/T system M_3 in Fig. 87. The definition of covering tree is such that we can also construct one path through the tree instead of the whole tree. When we construct the path that corresponds to the sequence of

transitions $t_1 t_4 t_3 t_4$ then we get successively $\gamma(\lambda) = (1, 1, 0, 0)$, $\gamma(t_1) = (0, 1, 1, 1)$, $\gamma(t_1 t_4) = (1, \omega, \omega, 0)$, $\gamma(t_1 t_4 t_3) = (1, \omega, \omega, \omega)$, and $\gamma(t_1 t_4 t_3 t_4) = (\omega, \omega, \omega, \omega)$. This implies that all places are unbounded. It also implies that $\mathbb{V}_{M_3}^\circ = \{(\omega, \omega, \omega, \omega)\}$.

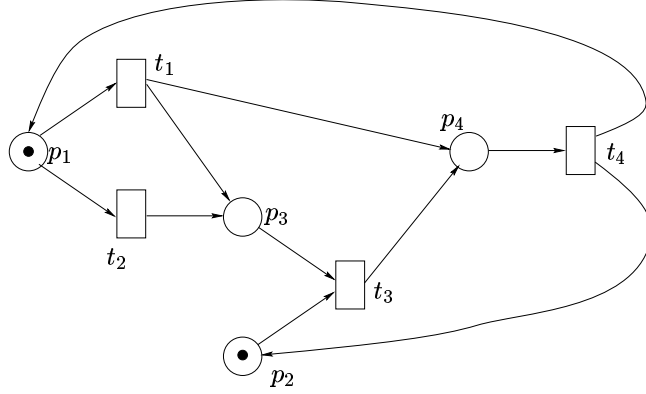


Fig. 87. P/T system M_3 .

We now show that usefulness of transitions is decidable.

Lemma 183. *It is decidable for an arbitrary P/T system M and an arbitrary configuration C of M , whether or not there exists $D \in \mathbb{C}_M$ with $C \leq D$.*

Proof. If \mathbb{V} is a covering set of M , then: there exists $D \in \mathbb{C}_M$ with $C \leq D$ iff there exists a $D \in \mathbb{V}$ with $C \leq D$. \square

Theorem 184. *It is decidable, for an arbitrary P/T system M and an arbitrary transition $t \in T_M$, whether or not t is useful.*

Proof. Let C_t be the configuration of M with $C_t(p) = W(p, t)$ if $p \in \bullet t$ and $C_t(p) = 0$ otherwise. Then: t is useful iff there exists $D \in \mathbb{C}_M$ with $C_t \leq D$. Now use Lemma 183. From the proof of that lemma it follows that t is useful iff there exists $D \in \mathbb{V}_M^\circ$ with $C_t \leq D$. \square

Corollary 185. *It is decidable, for an arbitrary P/T system M , whether or not M is reduced.*

Example 51. Consider the P/T system $M = M_1$ of Fig. 83. For \mathbb{V}_M° see Example 50(1). Then, for $t = t_3$, $C_t = (0, 0, 1, 1, 0)$. Now $(0, 0, \omega, 1, 0) \in \mathbb{V}_M^\circ$ and $(0, 0, 1, 1, 0) \leq (0, 0, \omega, 1, 0)$. Hence t_3 is useful. In the same way one can see that

all transitions of M are useful. Hence M is reduced (and since M has no isolated places, even strongly reduced).

If M would additionally have a transition t_7 with $\bullet t_7 = \{p_1, p_4\}$, then M would still have the same covering tree. For $t = t_7$, $C_t = (1, 0, 0, 1, 0)$ and there is no $C \in \mathbb{V}_M^\circ$ with $C_t \leq C$. Hence we find that t_7 is not useful. Removal of this useless transition obviously results again in the M of Fig. 83.

The next two theorems (cf. Lemma 183 and Theorem 184) are hard to prove and are thus stated without proof. The first theorem states that the so-called *reachability problem* is decidable; it took more than ten years before this theorem could be proved. For a proof see, e.g., [Reu90].

Theorem 186. *It is decidable, for an arbitrary P/T system M and an arbitrary configuration C of M , whether or not $C \in \mathbb{C}_M$.*

Theorem 187. *It is decidable, for an arbitrary P/T system M and an arbitrary configuration t of M , whether or not t is live.*

9.5 Place Invariants

An *invariant* of a P/T system M is, in general, a function f of the configurations of M which has the same value for all reachable configurations of M , i.e., $f(C) = f(C_{in})$ for all $C \in \mathbb{C}_M$. Invariants are used to prove specific properties of the system, i.e., to prove the “correctness” of the system (analogous to the use of invariants for showing program correctness). In this section we treat invariants of a special type, so-called “place-invariants”; these place-invariants can be determined automatically from the system M . We have already seen a simple case of them for EN systems: for a sequential component S of a system M , $\#(C \cap S) = 1$ for all $C \in \mathbb{C}_M$ (Theorem 49), i.e., the function f with $f(C) = \#(C \cap S)$ is an invariant of the system. We have already seen that such invariants can serve to show the contact-freeness of a system (Theorem 59).

Consider as an example the P/T system M of Fig. 88. For this system the total number of tokens in all places always equals 1, i.e., $C(p_1) + C(p_2) = 1$ for all $C \in \mathbb{C}_M$; this means that $C(p_1) + C(p_2)$ is an invariant of M (or more precisely, the function f with $f(C) = C(p_1) + C(p_2)$ is). If we add a place p_3 as output-place of t_2 (resulting in the P/T system M' of Fig. 89), then the total number of tokens in all places no longer equals 1 (since p_3 is even unbounded). But the fact that the sum of the number of tokens in p_1 and the number of tokens in p_2 always equals 1 does continue to hold, i.e., $C(p_1) + C(p_2)$ is an invariant of M' , with value 1. Note that this means that $\{p_1, p_2\}$ is a sequential component of M' (if we would have defined the notion of sequential component of a P/T system similar to Definitions 40 and 45 for EN systems). However, if we increase the weight of the edges (t_1, p_2) and (p_2, t_2) to 2 (resulting in the P/T system M'' of Fig. 90), then this no longer holds. But now, the sum of two times the number of tokens in p_1 and the number of tokens in p_2 is always equal to 2, i.e., $2C(p_1) + C(p_2) = 2$ for all $C \in \mathbb{C}_{M''}$. This means that $2C(p_1) + C(p_2)$

is an invariant of M'' , with value 2. Finally we consider the P/T system M_1 of Fig. 91. For this system the sum of the number of tokens in p_1, p_2, p_3 always equals the number of tokens in p_4 . Hence $C(p_1) + C(p_2) + C(p_3) = C(p_4)$ for all $C \in \mathbb{C}_{M_1}$. This means that $C(p_1) + C(p_2) + C(p_3) - C(p_4)$ is an invariant of M_1 , with value 0.

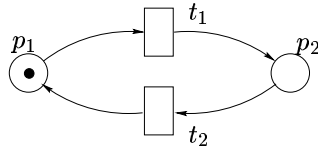


Fig. 88. P/T system M with invariant $C(p_1) + C(p_2) = 1$.

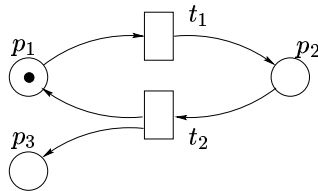


Fig. 89. P/T system M' with invariant $C(p_1) + C(p_2) = 1$.

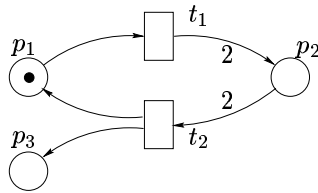


Fig. 90. P/T system M'' with invariant $2C(p_1) + C(p_2) = 2$.

We will only consider invariants that, like in the above examples, are linear combinations of the number of tokens in the places of the system, i.e., invariants of the form $i_1C(p_1) + i_2C(p_2) + \dots + i_nC(p_n)$, where $P_M = \{p_1, \dots, p_n\}$

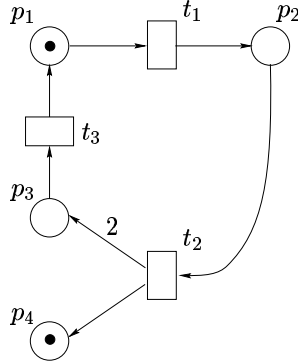


Fig. 91. P/T system M_1 with invariant $C(p_1) + C(p_2) + C(p_3) - C(p_4) = 0$.

and $i_1, \dots, i_n \in \mathbb{Z}$. Such an invariant is completely determined by the vector $i = (i_1, \dots, i_n)$. Intuitively, this means that the weighted sum of all tokens is constant, where a token in place p_k has weight i_k . For two vectors $v_1, v_2 : P \rightarrow \mathbb{Z}$ we define, as usual, their *inner product* as $v_1 \cdot v_2 = \sum_{p \in P} v_1(p) \cdot v_2(p)$. Hence, the above invariant can now be written as $i \cdot C$, or as $C \cdot i$. We now give a formal definition of this kind of invariants. For technical reasons we not only require that $C \cdot i = C_{in} \cdot i$ for all $C \in \mathbb{C}_M$, but even that for all configurations C, D of M and all transitions t of M : if $C[t]D$ then $C \cdot i = D \cdot i$. In Theorem 192 we will show that these requirements are equivalent whenever M is reduced.

Definition 188. Let $M = (P, T, F, W, C_{in})$ be a P/T system. A vector $i : P \rightarrow \mathbb{Z}$ is a *place invariant* (*p-invariant* for short) of M if for all configurations $C, D : P \rightarrow \mathbb{N}$ and all $t \in T$: if $C[t]D$, then $C \cdot i = D \cdot i$. The *value* of i is the integer $C_{in} \cdot i$.

A p-invariant $i : P \rightarrow \mathbb{N}$ is a *positive* p-invariant of M .

A p-invariant $i : P \rightarrow \{0, 1\}$ is a *characteristic* p-invariant of M (note that i is the characteristic function of the set of places $\{p \in P \mid i(p) = 1\}$).

A characteristic p-invariant with value 1 is a *sequential component* of M .

Later we will see that this definition of sequential component is consistent with the one for EN systems.

Lemma 189. Let $M = (P, T, F, W, C_{in})$ be a P/T system and let $i : P \rightarrow \mathbb{Z}$ be a p-invariant of M . Then $C \cdot i = C_{in} \cdot i$ for all $C \in \mathbb{C}_M$.

Proof. We do this by induction on C . It is trivial for C_{in} . Now assume that it holds for $C \in \mathbb{C}_M$ and that $C[t]D$. Then it holds for D because $D \cdot i = C \cdot i = C_{in} \cdot i$. \square

Example 52. (a) Let us verify that the invariant $i = (2, 1, 0)$ discussed above, is indeed a (positive) p-invariant of the P/T system M'' of Fig. 90. Assume that

$C, D : P \rightarrow \mathbb{N}$. If $C[t_1]D$, then $2D(p_1) + D(p_2) = 2(C(p_1) - 1) + (C(p_2) + 2) = 2C(p_1) + C(p_2) - 2 + 2 = 2C(p_1) + C(p_2)$. If $C[t_2]D$, then $2D(p_1) + D(p_2) = 2(C(p_1) + 1) + (C(p_2) - 2) = 2C(p_1) + C(p_2) + 2 - 2 = 2C(p_1) + C(p_2)$.

(b) Likewise it is easy to verify that $(1, 1, 1, -1)$ is a p-invariant of the P/T system M_1 of Fig. 91.

For a p-invariant i , $C \cdot i - D \cdot i$ must be zero, if $C[t]D$. Hence $(C - D) \cdot i$ must be zero, where $C - D$ is the usual difference of the vectors C and D . The difference vector $C - D$ is completely determined by t . This is defined formally in the next definition.

Definition 190. Let $M = (P, T, F, W, C_{in})$ be a P/T system.

(1) For every transition $t \in T$ the vector $\underline{t} : P \rightarrow \mathbb{Z}$ is defined as follows: for every $p \in P$,

$$\begin{aligned} \underline{t}(p) &= W(t, p) && \text{if } p \in t^\bullet, \\ \underline{t}(p) &= -W(p, t) && \text{if } p \in {}^\bullet t, \text{ and} \\ \underline{t}(p) &= 0 && \text{otherwise.} \end{aligned}$$

(2) The matrix $\underline{M} : T \times P \rightarrow \mathbb{Z}$ is defined such that, for every $t \in T$ and every $p \in P$, $\underline{M}(t, p) = \underline{t}(p)$.

Matrix multiplication is defined in the usual way; in particular, if $v : P \rightarrow \mathbb{Z}$, then $\underline{M} \cdot v$ is the vector $T \rightarrow \mathbb{Z}$ with for all $t \in T$, $(\underline{M} \cdot v)(t) = \underline{t} \cdot v$ (the inner product of \underline{t} and v).

Note that the matrix \underline{M} is, in principle, the usual incidence matrix of the graph that represents the multi-net (P, T, F, W) .

For a transition t , the vector \underline{t} is the difference between the configurations *before* and *after* firing t .

Lemma 191. Let M be a P/T system, let C and D be two configurations of M , and let $t \in T_M$. Then $C[t]D$ iff $D = C + \underline{t}$.

Due to this lemma a P/T system is also called a *Vector Addition System*. The token game of a P/T system M with n places consists of repeatedly adding the vectors $\underline{t} \in \mathbb{Z}^n$, with $t \in T_M$, starting with the vector $C_{in} \in \mathbb{N}^n$ and taking care that all resulting vectors are in \mathbb{N}^n .

Lemma 191 implies a simple linear-algebraic characterization of p-invariants.

Theorem 192. Let $M = (P, T, F, W, C_{in})$ be a P/T system and let $i : P \rightarrow \mathbb{Z}$. Then the following three statements are equivalent.

- (1) i is a p-invariant;
- (2) $\underline{t} \cdot i = 0$ for all $t \in T$;
- (3) $\underline{M} \cdot i = 0$ (where 0 is the null vector $T \rightarrow \mathbb{Z}$).

If, moreover, M is reduced, then this is also equivalent with

- (4) $C \cdot i = C_{in} \cdot i$ for all $C \in \mathbb{C}_M$.

Proof. (2) implies (1): Assume $\underline{t} \cdot i = 0$ for all $t \in T$. If $C[t]D$, then according to Lemma 191 $D = C + \underline{t}$. Hence $D \cdot i = (C + \underline{t}) \cdot i = C \cdot i + \underline{t} \cdot i = C \cdot i$. Thus i is a p-invariant. We did this calculation in Example 52(a); in this example (see Fig. 90)

$\underline{t}_1 = (-1, 2, 0)$ and $\underline{t}_2 = (1, -2, 1)$, and for $i = (2, 1, 0)$ indeed $\underline{t}_1 \cdot i = -2 + 2 = 0$ and $\underline{t}_2 \cdot i = 2 - 2 = 0$.

(1) implies (2): For every t there is a configuration $C : P \rightarrow \mathbb{N}$ such that $t \mathbf{con} C$ (take, e.g., $C(p) = W(p, t)$ for $p \in \bullet t$ and $C(p) = 0$ otherwise). Then $C[t]C + \underline{t}$ according to Lemma 191. Thus, since i is a p-invariant, $C \cdot i = (C + \underline{t}) \cdot i = C \cdot i + \underline{t} \cdot i$. Hence $\underline{t} \cdot i = 0$.

(2) \Leftrightarrow (3): This follows directly from the definition of matrix multiplication, see Definition 190(2).

(1) implies (4): That is Lemma 189.

(4) implies (2): Since t is useful, there exists $C \in \mathbb{C}_M$ with $t \mathbf{con} C$. Then also $C + \underline{t} \in \mathbb{C}_M$. Then $C \cdot i = C_{in} \cdot i$ and $(C + \underline{t}) \cdot i = C_{in} \cdot i$. Hence $C \cdot i = C \cdot i + \underline{t} \cdot i$ and so $\underline{t} \cdot i = 0$. \square

Corollary 193. *Let $M = (P, T, F, W, C_{in})$ be a P/T system and let $i : P \rightarrow \mathbb{N}$ be the characteristic function of $S \subseteq P$. Then i is a (characteristic) p-invariant iff*

$$\sum \{W(p, t) \mid p \in \bullet t \cap S\} = \sum \{W(t, p) \mid p \in t^\bullet \cap S\} \text{ for all } t \in T.$$

If, in particular, $W(x, y) = 1$ for all $(x, y) \in F$, then this is equivalent with: $\#(\bullet t \cap S) = \#(t^\bullet \cap S)$ for all $t \in T$.

Proof. The two sums are equal iff $\underline{t} \cdot i = 0$. \square

Another corollary of Theorem 192 is that for reduced EN systems the new definition of sequential component (Definition 188) is equivalent to the old one (Definition 45), via the correspondence between a set S and its characteristic function, Theorem 49(2), and the fact that an (old) sequential component is completely determined by its set of places. Corollary 193 corresponds to Theorem 49(3).

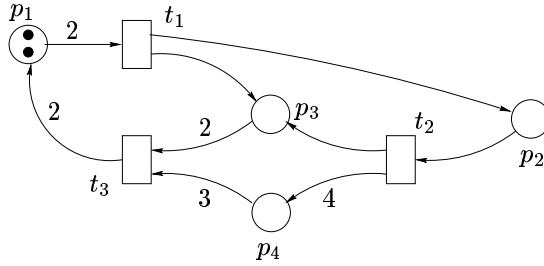


Fig. 92. A P/T system to illustrate linear algebra.

Example 53. Consider the P/T system M of Fig. 92. The changes of configuration when firing transitions of M can be denoted by (column) vectors as follows; below t_i the vector \underline{t}_i is given.

$$\begin{array}{cccccccc}
& C_{in} [t_1] & C_1 [t_2] & C_2 [t_3] & C_3 [t_1] & C_4 [t_2] & C_5 [t_3] & C_6 \\
p_1 & (2) & -2 (0) & 0 (0) & +2 (2) & -2 (0) & 0 (0) & +2 (2) \\
p_2 & (0) & +1 (1) & -1 (0) & 0 (0) & +1 (1) & -1 (0) & 0 (0) \\
p_3 & (0) & +1 (1) & +1 (2) & -2 (0) & +1 (1) & +1 (2) & -2 (0) \\
p_4 & (0) & 0 (0) & +4 (4) & -3 (1) & 0 (1) & +4 (5) & -3 (2) .
\end{array}$$

The matrix \underline{M} is

$$\begin{array}{cccc}
& p_1 & p_2 & p_3 & p_4 \\
t_1 & (-2 & +1 & +1 & 0) \\
t_2 & (0 & -1 & +1 & +4) \\
t_3 & (+2 & 0 & -2 & -3)
\end{array} .$$

It is clear that the firing of t_3 does not alter the sum of the number of tokens in p_1 , p_2 , and p_3 . Hence, for $i = (1, 1, 1, 0)$, $\underline{t}_3 \cdot i = (2, 0, -2, -3) \cdot (1, 1, 1, 0) = 2 - 2 = 0$. The same holds for t_1 and t_2 , thus

$$\underline{M} \cdot i = \begin{pmatrix} -2 & 1 & 1 & 0 \\ 0 & -1 & 1 & 4 \\ 2 & 0 & -2 & -3 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} .$$

Consequently, the vector i is a p-invariant of M , with value $C_{in} \cdot i = (2, 0, 0, 0) \cdot (1, 1, 1, 0) = 2$. Moreover, i is a characteristic p-invariant: i is the characteristic function of the set $S = \{p_1, p_2, p_3\}$. In other words, the total number of tokens in S always equals 2.

From the equivalence of (1) and (3) in Theorem 192 it follows directly that the set of p-invariants of a P/T system is a linear subspace of the linear vector space \mathbb{Z}^n (with $n = \#P$). This is because it is the set of solutions of the system of equations $\underline{M} \cdot i = 0$.

Lemma 194. *Let M be a P/T system and let $\lambda \in \mathbb{Z}$. If i_1 and i_2 are p-invariants of M , then so are $i_1 + i_2$ and $\lambda \cdot i_1$.*

With the usual methods from linear algebra we can thus determine “all” p-invariants, in the sense that we can compute a base for the set of all p-invariants (i.e., a finite set B of linearly independent p-invariants, such that every p-invariant is a linear combination of vectors from B and vice versa).

Example 54. (1) Consider again the P/T system M of Example 53 (Fig. 92). The system of equations corresponding to $\underline{M} \cdot i = 0$ is as follows, where $i = (x_1, x_2, x_3, x_4)$:

$$\begin{aligned}
-2x_1 + x_2 + x_3 &= 0, \\
-x_2 + x_3 + 4x_4 &= 0, \\
2x_1 - 2x_3 - 3x_4 &= 0.
\end{aligned}$$

Note that the first equation corresponds to t_1 , the second to t_2 , and the third to t_3 . These equations are equivalent to $x_1 = x_2 = x_3$ and $x_4 = 0$. The solution space thus consists of all vectors $\lambda \cdot i$ with $i = (1, 1, 1, 0)$ and $\lambda \in \mathbb{Z}$. Hence in a certain sense $(1, 1, 1, 0)$ is the only p-invariant of M (all other p-invariants are a multiple of it).

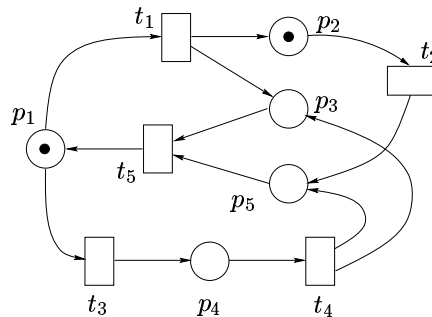


Fig. 93. A P/T system with p-invariants $\lambda \cdot (1, 1, 0, 1, 1) + \mu \cdot (1, 0, 1, 1, 0)$.

(2) Consider the P/T system M of Fig. 93. The matrix \underline{M} is

$$\begin{array}{c}
\begin{matrix} p_1 & p_2 & p_3 & p_4 & p_5 \end{matrix} \\
\begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{matrix} \begin{pmatrix} -1 & +1 & +1 & & \\ & -1 & & & +1 \\ -1 & & & +1 & \\ & & +1 & -1 & +1 \\ +1 & -1 & & -1 & \end{pmatrix}.
\end{array}$$

The p-invariants of M are therefore the solutions $i = (x_1, x_2, x_3, x_4, x_5)$ of the following system of equations:

$$\begin{aligned}
-x_1 + x_2 + x_3 &= 0, \\
-x_2 + x_3 + x_5 &= 0, \\
-x_1 + x_4 &= 0, \\
x_3 - x_4 + x_5 &= 0, \\
+x_1 - x_3 - x_5 &= 0.
\end{aligned}$$

Solving this system of equations shows that the set of p-invariants of M consists of all linear combinations of the vectors $(1, 1, 0, 1, 1)$ and $(1, 0, 1, 1, 0)$. The first of these two p-invariants has value 2 and the second has value 1. They are both characteristic p-invariants and the second is even a sequential component.

From positive p-invariants we can obtain useful structural information, as the following result shows.

Lemma 195. *Let M be a P/T system and let i be a positive p -invariant of M . For $S = \{p \in P_M \mid i(p) > 0\}$, $\bullet S = S^\bullet$.*

With the obvious generalization of the notion of “subsystem” (Definition 45) for P/T systems this means, according to Lemma 47, that S is a subsystem of M .

We now show that the “positive” places of such a p -invariant are bounded (see Definition 181). Whenever applicable, this provides a simpler method to show boundedness than Theorem 182 does.

Theorem 196. *Let M be a P/T system and let i be a positive p -invariant of M . For all $p \in P_M$, if $i(p) > 0$, then p is bounded.*

Proof. Let $C \in \mathbb{C}_M$. Since i is a p -invariant, $C \cdot i = C_{in} \cdot i$. Because i is positive, $C(p) \cdot i(p) \leq C \cdot i$. Hence $C(p) \cdot i(p) \leq C_{in} \cdot i$ and thus $C(p) \leq \frac{C_{in} \cdot i}{i(p)}$. \square

Example 55. According to Example 52(a), $(2, 1, 0)$ is a positive p -invariant of the P/T system of Fig. 90. Then Theorem 196 implies that p_1 and p_2 are bounded (but of course we already knew this).

Now consider the P/T system of Fig. 85. Though p_1 is bounded, this cannot be proved with Theorem 196. Assume that i is a positive p -invariant with $i(p_1) > 0$. Then by Lemma 195 also $i(p_2) > 0$ must hold. Then, by Theorem 196, p_2 is bounded. But p_2 is unbounded (see Example 50(2)).

Definition 197. Let $M = (P, T, F, W, C_{in})$ be a P/T system.

A collection $\{i_1, \dots, i_n\}$ of positive p -invariants of M is a *covering* of M if for every place $p \in P$ there exists k , $1 \leq k \leq n$, with $i_k(p) > 0$. Then M is *covered* by $\{i_1, \dots, i_n\}$.

In case i_1, \dots, i_n are sequential components, this definition equals Definition 52 (see Lemma 53 and the remarks after Corollary 193).

Theorem 198. *If a P/T system M is covered by positive p -invariants, then M is bounded.*

We now show that *one* positive p -invariant suffices for such a covering.

Lemma 199. *A P/T system M is covered by positive p -invariants iff there exists a p -invariant i of M such that $i(p) > 0$ for all $p \in P_M$.*

Proof. Assume that for every place p there is a positive p -invariant i_p with $i_p(p) > 0$. Then, by Lemma 194, $i = \sum\{i_p \mid p \in P\}$ is also a p -invariant of M . Obviously $i(p) > 0$ for all $p \in P$. \square

A p -invariant i with $i(p) > 0$ for all $p \in P$ means that the weighted sum of all tokens is constant, where each token in place p has a positive weight $i(p)$. Intuitively each token represents a number of “components” and the sum of all “components” is constant. A special case of this is a covering by sequential

components S_1, \dots, S_n . In that case (according to the proof of Lemma 199) there is a p-invariant i with, for all $p \in P$, $i(p) = \#\{k \mid p \in S_k, 1 \leq k \leq n\}$ and the value of i is n (verify this). This means that the “components” in this case are exactly the sequential components.

Example 56. (1) According to Example 54(2) the P/T system M of Fig. 93 is covered by the two positive p-invariants $(1, 1, 0, 1, 1)$ and $(1, 0, 1, 1, 0)$. Hence, by Lemma 199, M is covered by the p-invariant $(2, 1, 1, 2, 1)$ with value 3. By Theorem 198, M is bounded.

(2) According to Example 21(3) the P/T system of Fig. 15 is covered by two sequential components: $\{p_0, p_1, p_2, p_4, p_6\}$ and $\{p_0, p_1, p_3, p_5, p_6\}$. Then M is covered by the p-invariant i with $i(p_0) = i(p_1) = i(p_6) = 2$ and $i(p) = 1$ for the remaining places. The value of i is 2.

(3) According to Example 21(4) the P/T system M of Fig. 5 (mutual exclusion) is covered by three sequential components: $\{w_1, c_1, r_1\}$, $\{w_2, c_2, r_2\}$ and $\{p, c_1, c_2\}$. Then M is covered by the p-invariant i with $i(c_1) = i(c_2) = 2$ and $i(p') = 1$ for the remaining places p' . The value of i is 3.

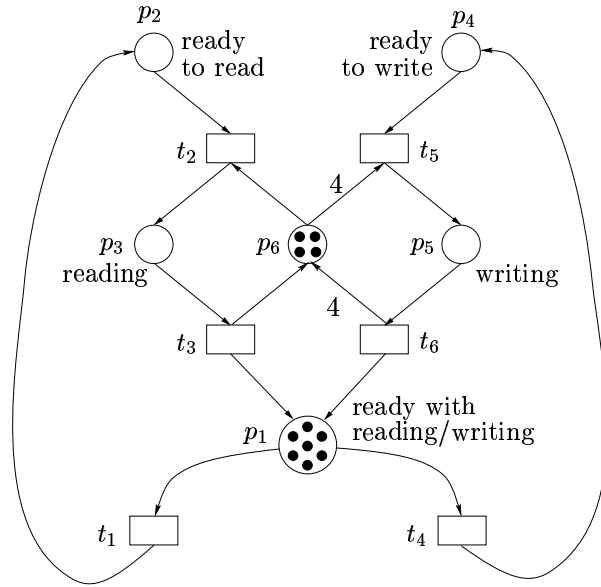


Fig. 94. Readers and writers.

We will now show by way of an example that p-invariants are useful to prove the correctness of a system. Consider the P/T system M of Fig. 94 (a variant

of Fig. 78). The system contains 7 active components (where we once again use the word “component” informally). In the initial configuration C_{in} all 7 active components are in place p_1 : every token is such a component. Each of these components has access to the same piece of memory, to read from or write to. The access to the memory should be restricted as follows: (1) if a component writes to the memory, no other component has access to the memory and (2) if no component writes, then at most 4 components are allowed to read from the memory simultaneously. To guarantee this the system uses 4 permits, which initially are in place p_6 ; each of these permits can be seen as a passive component of M . To show that requirements (1) and (2) are satisfied, we first determine the p-invariants of M . Solving the system of equations

$$\begin{array}{rcl} -x_1 + x_2 & & = 0 \\ & -x_2 + x_3 & -x_6 = 0 \\ +x_1 & -x_3 & +x_6 = 0 \\ -x_1 & +x_4 & = 0 \\ & -x_4 + x_5 - 4x_6 & = 0 \\ +x_1 & -x_5 + 4x_6 & = 0 \end{array}$$

results in the two p-invariants $i_1 = (1, 1, 1, 1, 1, 0)$ and $i_2 = (0, 0, 1, 0, 4, 1)$. The value of i_1 is $C_{in} \cdot i = (7, 0, 0, 0, 0, 4) \cdot (1, 1, 1, 1, 1, 0) = 7$ and the value of i_2 is $C_{in} \cdot i_2 = 4$. Now Lemma 189 implies that for all $C \in \mathbb{C}_M$:

$$\begin{array}{l} (i_1) \quad C(p_1) + C(p_2) + C(p_3) + C(p_4) + C(p_5) = 7 \text{ and} \\ (i_2) \quad C(p_3) + 4C(p_5) + C(p_6) = 4. \end{array}$$

From (i_1) follows that there are always 7 active components, distributed over the places p_1, \dots, p_5 . Likewise it follows from (i_2) that there are always 4 passive components, in places p_3, p_5 , and p_6 . We now take a closer look at (i_2) . If a component is writing, then $C(p_5) > 0$. From (i_2) it then follows that $C(p_5) = 1$ and $C(p_3) = 0$: exactly *one* component is writing and *no* component is reading. Now assume that no component is writing, i.e., $C(p_5) = 0$. By (i_2) we obtain $C(p_3) \leq 4$: at most 4 components are reading. This proves requirements (1) and (2) mentioned above.

We now show two more properties of M . To begin with, note that the positive p-invariants i_1 and i_2 cover M . Thus, by Theorem 198, M is bounded. By Lemma 194, M is covered by $i = i_1 + i_2 = (1, 1, 2, 1, 5, 1)$ with value 11 (see also Lemma 199). Hence there are 11 components (viz. 7 active and 4 passive ones). And, e.g., a token in p_5 represents 5 components (one active component and four permits).

Another desired property of M is that it is “live”.

Definition 200. A P/T system M is *live* if all transitions of M are live.

We now show that the system M of Fig. 94 is live. Consider an arbitrary $C \in \mathbb{C}_M$. We first show that there exists a $y \in T^*$ with $C[y]C_{in}$. If there are tokens in p_3 , then by (repeatedly) firing t_3 we can reach a configuration C' with $C'(p_3) = 0$. By firing t_6 we can moreover achieve $C'(p_5) = 0$. Since $C'(p_3) = C'(p_5) = 0$, invariant i_2 now implies $C'(p_6) = 4$. This means that if $C'(p_2) > 0$ then $t_2 t_3$ **con** C' and if $C'(p_4) > 0$ then $t_5 t_6$ **con** C' . Thus by

repeatedly firing t_2t_3 and t_5t_6 tokens in p_2 and p_4 can be transported to p_1 . For the configuration D that is reached in this way with firing sequence y we have $D(p_2) = D(p_3) = D(p_4) = D(p_5) = 0$ and thus, by invariant i_1 , $D(p_1) = 7$. By invariant i_2 moreover $D(p_6) = 4$. Hence $D = C_{in}$. Now $t_1t_2t_3 \text{ con } C_{in}$ and $t_4t_5t_6 \text{ con } C_{in}$. This implies that $yt_1 \text{ con } C$ and $yt_1t_2 \text{ con } C$ and likewise for t_3, \dots, t_6 . This proves that M is live.

9.6 Marked Graphs

In the last two sections we will put a number of (structural) restrictions on the underlying multi-nets of P/T systems, that simplify the analysis of those systems. In particular we obtain in this way simple criteria for safeness and liveness (see Definitions 158 and 200 and Theorems 169 and 187).

Definition 201. A *marked graph* (or *T-system*) is a P/T system $M = (P, T, F, W, C_{in})$, where

- (1) $W(x, y) = 1$ for all $(x, y) \in F$, and
- (2) $\#(\bullet p) = \#(p\bullet) = 1$ for all $p \in P$.

Thus, a P/T system is a marked graph if every edge has weight 1 and every place has precisely one transition in its input-set and precisely one transition in its output-set. Note that these are indeed restrictions on the underlying net. For a place p we will denote the unique transition in $\bullet p$ also by $\bullet p$ and likewise for $p\bullet$.

Since the places of a marked graph are “unbranching”, intuitively no conflicts can arise (just as was the case with process nets, see Section 6.2). In particular each marked graph which is also an EN system is conflict-free (see Definition 22 and the discussion following it).

Intuitively we can view the transitions of a marked graph M as the nodes of an edge-labelled graph G (without initial node), where each place p of M is represented by an edge of G with label p , from node $\bullet p$ to node $p\bullet$. Then the edges of G are labelled with tokens and if a node of G fires, then the tokens of the incoming edges of the node are removed and tokens are put on the outgoing edges of the node.

Example 57. The P/T system M of Fig. 95 is a marked graph. In Fig. 96 its corresponding graph G is depicted (without marking). The P/T systems of Figs. 48 and 49 are also marked graphs.

Definition 202. Let $M = (P, T, F, W, C_{in})$ be a P/T system.

(1) A sequence $\alpha = (p_0, p_1, \dots, p_m)$, with $m \geq 0$ and $p_i \in P$ for $0 \leq i \leq m$, is a *path of M , from p_0 to p_m* , if $p_i\bullet = \bullet p_{i+1}$ for every $0 \leq i \leq m - 1$. If moreover $p_m\bullet = \bullet p_0$ and all $p_i\bullet$, $0 \leq i \leq m$, are different, then α is a (*elementary*) *cycle of M* ; for $t \in T$ we say that t is *on α* if $t = p_i\bullet$ for an $0 \leq i \leq m$.

(2) Let $\alpha = (p_0, p_1, \dots, p_m)$, $m \geq 0$, be a path of M and let C be a configuration of M . Then $\sum_{i=0}^m C(p_i)$ is the *value of α in C* and is denoted by $C(\alpha)$. If $C(\alpha) > 0$, then α is *marked by C* .

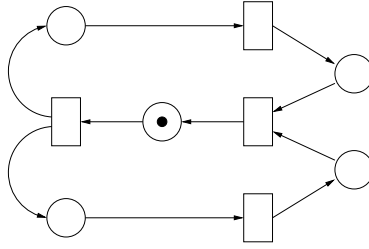


Fig. 95. A marked graph.

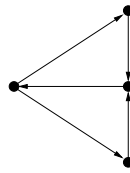


Fig. 96. The graph corresponding to the marked graph of Fig. 95 (without tokens).

Note that for a cycle $\alpha = (p_0, p_1, \dots, p_m)$ not only all transitions p_i^\bullet are different, but also all places p_i are different; note also that $m \geq 1$, see Definition 1(4). Note finally that, intuitively, $C(\alpha)$ is the total number of tokens on the path α .

The next, important, lemma shows that the value of a cycle of a marked graph always remains the same.

Lemma 203. *Let $M = (P, T, F, W, C_{in})$ be a marked graph. Let $\alpha = (p_0, \dots, p_m)$ be a cycle of M and let C_α be the characteristic function of $\{p_0, \dots, p_m\}$. Then*

- (1) C_α is a characteristic p-invariant of M , and
- (2) $C(\alpha) = C_{in}(\alpha)$ for all $C \in \mathbb{C}_M$.

Proof. (1) Let $S = \{p_0, \dots, p_m\}$. Consider a $t \in T$. Then either $\#(\bullet t \cap S) = \#(t^\bullet \cap S) = 1$ (if t is on α) or, since M is a marked graph, $\#(\bullet t \cap S) = \#(t^\bullet \cap S) = 0$ (if t is not on α). Thus, by Corollary 193, C_α is a p-invariant.

(2) It is clear that for a configuration C , $C(\alpha) = C \cdot C_\alpha$. Now use (1) and Lemma 189. \square

From now on we call $C_{in}(\alpha)$ the value of the cycle α ; note that it also is the value of the p-invariant C_α .

As we will show next, marked graphs allow an elegant analysis of liveness (see Definition 200), in terms of cycles. Moreover, a reduced marked graph is always live. Note that a marked graph has no isolated places; “reduced” thus means the same as “strongly reduced” for marked graphs.

Theorem 204. *Let $M = (P, T, F, W, C_{in})$ be a marked graph. Then the following three statements are equivalent.*

- (1) *M is live,*
- (2) *M is reduced,*
- (3) *all cycles of M have a value > 0 .*

Proof. (1) implies (2): Clear.

(2) implies (3): Assume that M is reduced, i.e., all transitions are useful. Consider a cycle $\alpha = (p_0, \dots, p_m)$ and take a transition t on α , i.e., $t = p_i^\bullet$ for some i . Since t is useful, there exists $C \in \mathbb{C}_M$ with $t \mathbf{con} C$. Hence $C(p_i) > 0$ and thus $C(\alpha) > 0$. Then $C_{in}(\alpha) > 0$ by Lemma 203(2).

* (3) implies (1): Assume that $C_{in}(\alpha) > 0$ for every cycle α . We will prove that M is live. Consider an arbitrary configuration $C \in \mathbb{C}_M$ and an arbitrary transition $t \in T$. We prove the existence of an $x \in T^*$ such that $xt \mathbf{con} C$, by induction on $\#\text{empty}(C, t)$, where $\text{empty}(C, t)$ is the set of all places $p \in P$ for which there exists a path α from p to a place in ${}^\bullet t$, such that α is not marked by C . If $\#\text{empty}(C, t) = 0$, then all places in ${}^\bullet t$ are marked by C and thus $t \mathbf{con} C$ (and we can take $x = \lambda$). Now assume that $\#\text{empty}(C, t) > 0$. By following the directed edges of the net of M from an unmarked place in ${}^\bullet t$ in the reverse direction, we can find a place $p \in \text{empty}(C, t)$ such that ${}^\bullet p \mathbf{con} C$, since otherwise there would be a cycle in $\text{empty}(C, t)$ which contradicts our assumption by Lemma 203(2). Let $t_1 = {}^\bullet p$ and let $C[t_1]C_1$. From the fact that M is a marked graph, it can be seen after some thinking that $\text{empty}(C_1, t) = \text{empty}(C, t) - t_1^\bullet$; note for instance (for the inclusion from left to right) that every path from a place of ${}^\bullet t_1$ to a place of ${}^\bullet t$, must contain a place of t_1^\bullet . Hence $\#\text{empty}(C_1, t) < \#\text{empty}(C, t)$ and so, by induction, there exists $y \in T^*$ with $yt \mathbf{con} C_1$. Thus $t_1 y t \mathbf{con} C$ (and we can take $x = t_1 y$). \square

Reduced marked graphs also have an elegant characterization of safeness (see Definition 158), in terms of cycles and in terms of sequential components.

Theorem 205. *Let M be a reduced marked graph. Then the following three statements are equivalent.*

- (1) *M is safe,*
- (2) *M is covered by sequential components,*
- (3) *every place of M belongs to a cycle of M with value 1.*

Proof. (3) implies (2): Directly clear from Lemma 203(1) and Definition 188.

(2) implies (1): Clear (for every P/T system) from Definitions 197 and 188 and Lemma 189.

(1) implies (3): Let $M = (P, T, F, W, C_{in})$. Since M is reduced, Theorem 204 implies that all cycles have a value ≥ 1 . Assume that there exists a $p \in P$ such that all cycles to which p belongs have a value > 1 (note that this includes the case that p belongs to no cycle). It remains to prove that M is not safe. Because M is reduced, there exists a $C \in \mathbb{C}_M$ such that $p^\bullet \mathbf{con} C$ and thus $C(p) > 0$. Let C_p be the characteristic function of $\{p\}$ and consider the configuration $C_1 = C - C_p$ (i.e., remove one token from p). Now $C_1(\alpha) > 0$ for all cycles α .

Thus, according to Theorem 204, the marked graph (P, T, F, W, C_1) is reduced. Hence there exist C_2 and $x \in T^*$ with $C_1[x]C_2$ such that $p^\bullet \text{ con } C_2$ and thus $C_2(p) > 0$. Then, by Lemma 160, $C_1 + C_p[x]C_2 + C_p$, i.e., $C[x]C_2 + C_p$ and thus $C_2 + C_p \in \mathbb{C}_M$. Since $(C_2 + C_p)(p) = C_2(p) + 1 > 1$, M is not safe. \square

Corollary 206. *A marked graph M is live and safe iff every cycle of M is marked by $(C_{in})_M$ and every place of M belongs to at least one cycle of which the value in $(C_{in})_M$ equals 1.*

Example 58. (a) The marked graph M of Fig. 95 has two cycles. Since both cycles have value 1 and every place belongs to one of the cycles, M is live and safe.

(b) The marked graph M of Fig. 12 (producer/consumer) has two cycles. Both cycles have value > 0 , but place b is not part of any cycle. Hence M is live, but not safe.

(c) The marked graph of Fig. 39 is live since both cycles have value > 0 , but not safe since p_1, p_2 , and p_4 do not belong to a cycle with value 1. If we remove the token from p_3 , then M is no longer live because a cycle with value 0 is created.

(d) The marked graphs of Figs. 48 and 49 are live and safe.

The next result should be compared to Theorem 59 and the remark following Theorem 60.

Corollary 207. *Let M be a reduced EN system which, moreover, is a marked graph. Then M is contact-free iff M is covered by sequential components.*

Proof. (If) Theorem 59.

(Only-if) Since M is contact-free, we can view M as a safe P/T system according to (the alternative) Definition 159. Since M is reduced, Theorem 205 then implies that M is covered by sequential components (and see the discussion following Corollary 193). \square

Given a marked graph $M = (P, T, F, W, C)$, we say that M can be started with a live and safe configuration if there exists a configuration C' such that $M' = (P, T, F, W, C')$ is live and safe. This is therefore a property of the underlying multi-net (P, T, F, W) .

Theorem 208. *Let M be a marked graph. Then M can be started with a live and safe configuration iff each place of M belongs to a cycle.*

Proof. (Only-if) Directly from Corollary 206.

* (If) Let $M = (P, T, F, W, C)$. Consider $M_1 = (P, T, F, W, C_1)$ where $C_1(p) = 1$ for all $p \in P$. By Theorem 204 M_1 is live. Now consider a $p \in P$. If every cycle to which p belongs has a value > 1 in C_1 , then we consider a configuration $C'_1 \in \mathbb{C}_{M_1}$ with $p^\bullet \text{ con } C'_1$ and thus $C'_1(p) > 0$ (which is possible since M_1 is live). Let $C_2 = C'_1 - C_p$ where C_p is the characteristic function of $\{p\}$ (i.e., we

remove a token from p). Let $M_2 = (P, T, F, W, C_2)$. Since all cycles of M_2 have a value > 0 , Theorem 204 implies that M_2 is live. Now if there is a cycle through p of which the value (in C_2) equals 1, then we are ready with this p . Otherwise we repeat the procedure above until we obtain a marked graph $M_i = (P, T, F, W, C_i)$ such that there is a cycle through p of which the value (in C_i) equals 1 (note that there is at least one cycle going through p).

We repeat this procedure for every place $p \in P$ (every time starting from the configuration where we ended up before). Now let C' be the final configuration. Then we are sure that every $p \in P$ belongs to a cycle of which the value (in C') equals 1. Theorem 204 guarantees that we always construct live marked graphs in the procedure. Hence, by Theorem 205, the marked graph $M' = (P, T, F, W, C')$ is live and safe. \square

9.7 Free-Choice Systems

Definition 209. A *free-choice system* is a P/T system $M = (P, T, F, W, C_{in})$ such that

- (1) for all $(x, y) \in F$, $W(x, y) = 1$, and
- (2) for all $(p, t) \in F \cap (P \times T)$, $p^\bullet = \{t\}$ or $\bullet t = \{p\}$.

Requirement (1) again means that all edges have weight 1. Requirement (2) means that in a free-choice system there cannot exist (different) places p and p' and (different) transitions t and t' with $(p, t'), (p, t), (p', t) \in F$, as shown in Fig. 97.

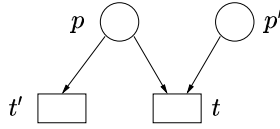


Fig. 97. Forbidden in free-choice systems.

For a safe system (i.e., a contact-free EN system) requirement (2) implies that there is no confusion in the system, see Section 4.5(4). To see this, note firstly that in a contact-free EN system there are only input-conflicts (or more precisely, if there is an output-conflict, then there is also an input-conflict, between the same two transitions). Now assume that (C, t_1, t_2) is a confusion (see Definition 24). Then there is a transition t such that firing t_2 brings t into an input-conflict with t_1 or takes it out of an input-conflict with t_1 . This means that t_1 and t can be depicted as in Fig. 97, with $t_1 = t'$ and $C(p) = 1$, and that the firing of t_2 puts a token in p' or removes a token from p' , respectively. Since Fig. 97 is not allowed in a free-choice system, there is no confusion. Compare Fig. 97 with Fig. 25 (where $p = p_3$, $t = t_3$, and $p' = p_4$), with Fig. 26 (where

$p = p_1$, $t = t_3$, and $p' = p_2$), and with Fig. 27 (where $p = p_4$, $t = t_4$, and $p' = p_3$, and also $p = p_1$, $t = t_3$, and $p' = p_2$). It shows that the P/T systems of Figs. 25 - 28 are not free-choice systems.

Note that all marked graphs are free-choice systems.

Example 59. The P/T systems of Figs. 87 and 98 are free-choice systems (ignore the dashed lines in Fig. 98). The P/T systems in Figs. 5 (mutual exclusion) and 83 are not free-choice systems.

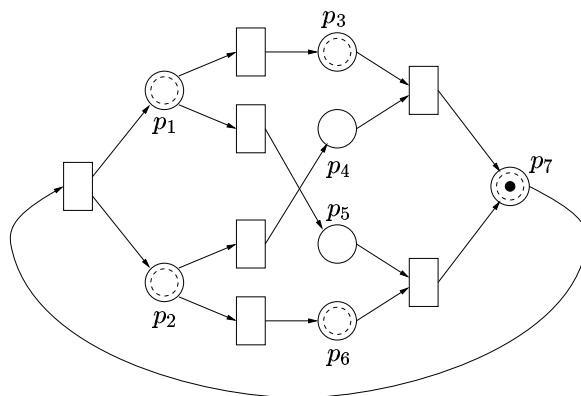


Fig. 98. A free-choice system with a siphon.

For free-choice systems there are characterizations of liveness and safeness which we will discuss now.

Definition 210. Let M be a P/T system and let $S \subseteq P_M$.

- (1) S is a *siphon* if $\bullet S \subseteq S^\bullet$.
- (2) S is a *trap* if $S^\bullet \subseteq \bullet S$.

Note that, according to Lemma 195, the “positive” places of a positive p-invariant are both a siphon and a trap. Hence, in particular, a sequential component is both a siphon and a trap, and so are the places of a cycle of a marked graph (by Lemma 203(1)). According to Lemma 47, sets of places that are both a siphon and a trap correspond to subsystems.

Intuitively Definition 210 means the following. (1) If S is a siphon and S contains no tokens in a configuration, then from this configuration S can never get to another configuration, in which S does contain tokens. (2) If S is a trap and S contains at least one token in a configuration, then S also contains at least one token in every further configuration. Hence, in short: a siphon can never be entered, and a trap can never be left.

This intuition is formally described in the following theorem.

Theorem 211. Let $M = (P, T, F, W, C_{in})$ be a P/T system and let $S \subseteq P$.

(1) If S is a siphon and $C_{in}(p) = 0$ for every $p \in S$, then $C(p) = 0$ for every $C \in \mathbb{C}_M$ and every $p \in S$.

(2) If S is a trap and there is a $p \in S$ with $C_{in}(p) > 0$, then for every $C \in \mathbb{C}_M$ there is a $p' \in S$ with $C(p') > 0$.

Example 60. Let M be the P/T system of Fig. 87. The siphons of M are \emptyset , $\{p_1, p_2, p_4\}$, $\{p_1, p_3, p_4\}$, and P_M , whereas the traps are \emptyset , $\{p_2, p_4\}$, $\{p_1, p_3, p_4\}$, $\{p_2, p_3, p_4\}$, and P_M . Note that M has a siphon that is not a trap and a trap that is not a siphon.

We now present the characterization of liveness, without (the quite complicated) proof.

Theorem 212. Let M be a free-choice system without isolated places. Then M is live iff every nonempty siphon contains a trap that has at least one token in $(C_{in})_M$.

Example 61. (1) Let M be the free-choice system of Fig. 98. Obviously, M is reduced. The set of places $S = \{p_1, p_2, p_3, p_6, p_7\}$ (indicated by dashed lines) is a siphon, because $\bullet S \subseteq S^\bullet = T_M$. However, this siphon does not contain any trap with a token. Thus, by Theorem 212, M is not live. By the way, this can also be seen immediately: in the reachable configuration C with $C(p_4) = C(p_5) = 1$ and $C(p) = 0$ for every other p , no transition has concession. Note that with any other initial configuration M neither is live!

(2) Let $M = (P, T, F, W, C_{in})$ be the free-choice system of Fig. 87 (see Example 60). Every nonempty siphon of M either contains the trap $\{p_2, p_4\}$ or the trap $\{p_1, p_3, p_4\}$. Since both traps have a token in C_{in} , Theorem 212 implies that M is live.

Now consider the free-choice system M' with the same multi-net as M , but with only a token in p_1 in the initial configuration C'_{in} . It is easy to see that M is reduced. Of course M' has the same siphons and traps as M . The siphon $\{p_1, p_2, p_4\}$ contains one trap only, viz. $\{p_2, p_4\}$, which does not contain tokens in C'_{in} . Hence, by Theorem 212, M' is not live. Note that firing t_2 (from C'_{in}) empties the siphon. Then, since each transition needs a token from the siphon, no transition has concession anymore.

The next theorem generalizes Theorem 205. Once again we do not give the (complicated) proof.

Theorem 213. Let M be a live free-choice system without isolated places. Then M is safe iff M is covered by sequential components.

Corollary 207 thus also holds for EN systems that are live free-choice systems (without isolated places).

The theory of free-choice systems is investigated in [Hac72, DesEsp95]. In [DesEsp95] a more general definition is used: there, a P/T system is said to be

free-choice if, for all transitions t_1 and t_2 , $\bullet t_1 \cap \bullet t_2 \neq \emptyset$ implies $\bullet t_1 = \bullet t_2$. A proof of Theorem 213 can be found in Section 4.2 of [Hac72] or Theorem 5.6 of [DesEsp95] (the so-called S-coverability Theorem), where the result is even shown for strongly connected sequential components (which means that the corresponding graphs are strongly connected).

References

- [AalRoz88] I.J.Aalbersberg, G.Rozenberg; Theory of traces, *Theor. Comput. Sci.* 60 (1988), 1–82
- [BesDev87] E.Best, R.Devillers; Sequential and concurrent behaviour in Petri net theory; *Theor. Comput. Sci.* 55 (1987), 87–136
- [BesFer88] E.Best, C.Fernández C.; *Nonsequential Processes*, EATCS Monographs on Theoretical Computer Science, Vol.13, Springer-Verlag, Berlin, 1988
- [BraReiRoz87] W.Brauer, W.Reisig, G.Rozenberg, eds.; *Petri Nets: Central Models and their Properties*, Lecture Notes in Computer Science 254, Springer-Verlag, Berlin, 1987
- [DesEsp95] J.Desel, J.Esparza; *Free Choice Petri Nets*, Cambridge University Press, Cambridge, 1995
- [DieRoz95] V.Diekert, G.Rozenberg, eds.; *The Book of Traces*, World Scientific, Singapore, 1995
- [EhrRoz90] A.Ehrenfeucht, G.Rozenberg; Partial 2-structures; Part II: State spaces of concurrent systems, *Acta Informatica* 27 (1990), 348–368
- [Eng85] J.Engelfriet; Determinacy \rightarrow (observation equivalence = trace equivalence), *Theor. Comput. Sci.* 36 (1985), 21–25
- [Eng91] J.Engelfriet; Branching processes of Petri nets, *Acta Informatica* 28 (1991), 575–591
- [Esp94] J.Esparza; Model checking using net unfoldings, *Science of Computer Programming* 23 (1994), 151–195
- [EspRömVog96] J.Esparza, S.Römer, W.Vogler; An improvement of McMillan's unfolding algorithm, in *Tools and Algorithms for the Construction and Analysis of Systems* (T.Margaria, B.Steffen, eds.), Proc. TACAS'96, Lecture Notes in Computer Science 1055, Springer-Verlag, Berlin, 1996, pp.87–106
- [Fin90] A.Finkel; Reduction and covering of infinite reachability trees, *Inform. and Comput.* 89 (1990), 144–179
- [Gen87] H.J.Genrich; Predicate/Transition nets, in [BraReiRoz87], pp.207–247
- [Hac72] M.H.T.Hack; *Analysis of production schemata by Petri nets*, M.S. Thesis, MIT, Project MAC, Cambridge, Mass., 1972
- [Hoo94] P.W.Hoogers; *Behavioural aspects of Petri nets*, Ph.D.Thesis, Leiden University, 1994
- [HooRoz91] H.J.Hoogeboom, G.Rozenberg; Diamond properties of elementary net systems, *Fundamenta Informaticae* 14 (1991), 287–300
- [HooRoz95] H.J.Hoogeboom, G.Rozenberg; Dependence graphs, Chapter 2 of [DieRoz95]
- [HopUll79] J.E.Hopcroft, J.D.Ullman; *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, Mass., 1979

- [Jen92] K.Jensen; *Coloured Petri Nets*, EATCS Monographs on Theoretical Computer Science, in three Volumes, Springer-Verlag, Berlin, 1992, 1994, and 1997
- [Kru72] J.B.Kruskal; The theory of well-quasi-ordering: a frequently discovered concept, *J. Combinatorial Theory (A)* 13 (1972), 297–305
- [LanRob78] L.H.Landweber, E.L.Robertson; Properties of conflict free and persistent Petri nets, *J. of the ACM* 25 (1978), 352–364
- [Maz95] A.Mazurkiewicz; Introduction to trace theory, Chapter 1 of [DieRoz95]
- [McM93] K.L.McMillan; Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits, in *Computer Aided Verification* (G.v.Bochmann, D.K.Probst, eds.), Proc. CAV'92, Lecture Notes in Computer Science 663, Springer-Verlag, Berlin, 1993, pp.164–177
- [McM95] K.L.McMillan; A technique of a state space search based on unfolding, *Formal Methods in System Design* 6 (1995), 45–65; see also [McM93]
- [Mil89] R.Milner; *Communication and Concurrency*, Prentice-Hall, Englewood Cliffs, NJ., 1989
- [NiePloWin81] M.Nielsen, G.Plotkin, G.Winskel; Petri nets, event structures and domains, Part 1, *Theor. Comput. Sci.* 13 (1981), 85–108
- [NieRozThi90] M.Nielsen, G.Rozenberg, P.S.Thiagarajan; Behavioural notions for elementary net systems, *Distributed Computing* 4 (1990), 45–59 (see also [Thi90])
- [NieRozThi92a] M.Nielsen, G.Rozenberg, P.S.Thiagarajan; Elementary transition systems, *Theor. Comput. Sci.* 96 (1992), 3–33
- [NieRozThi92b] M.Nielsen, G.Rozenberg, P.S.Thiagarajan; Elementary transition systems and refinement, *Acta Informatica* 29 (1992), 555–578
- [NieRozThi95] M.Nielsen, G.Rozenberg, P.S.Thiagarajan; Transition systems, event structures, and unfoldings, *Inform. and Comput.* 118 (1995), 191–207
- [Och95] E.Ochmański; Recognizable trace languages, Chapter 6 of [DieRoz95]
- [Pet62] C.A.Petri; *Kommunikation mit Automaten*, Rheinisch-Westfälisches Institut für Instrumentelle Mathematik an der Universität Bonn, Schrift Nr.2, 1962
- [Pet76] C.A.Petri; *Nicht-sequentielle Prozesse*, Universität Erlangen - Nürnberg, Arbeitsberichte des IMMD, Vol.9, Nr.8, 57–82, 1976
- [Pet81] J.L.Peterson; *Petri Net Theory and the Modeling of Systems*; Prentice-Hall, Englewood Cliffs, 1981
- [Pom88] L.Pomello; *Osservatore, reti di Petri, processi*, Ph.D.Thesis, Universities of Milano and Turino, 1988
- [PomRozSim92] L.Pomello, G.Rozenberg, C.Simone; A survey of equivalence notions for net based systems, in *Advances in Petri Nets 1992* (G.Rozenberg, ed.), Lecture Notes in Computer Science 609, Springer-Verlag, Berlin, 1992, pp.410–472
- [Pra86] V.R.Pratt; Modelling concurrency with partial orders, *Int. J. of Parallel Programming* 15 (1986), 33–71
- [Rei82] W.Reisig; *Petri Nets*, EATCS Monographs on Theoretical Computer Science, Vol.4, Springer-Verlag, Berlin, 1982
- [Reu90] C.Reutenauer; *The mathematics of Petri nets*, Prentice-Hall International (UK), 1990

- [Roz87] G.Rozenberg; Behaviour of elementary net systems, in [BraReiRoz87], pp.60–94
- [RozThi86] G.Rozenberg, P.S.Thiagarajan; Petri nets: basic notions, structure, behaviour, in *Current Trends in Concurrency* (J.W.de Bakker, W.-P.de Roever, G.Rozenberg, eds.), Lecture Notes in Computer Science 224, Springer-Verlag, Berlin, 1986, pp.585–668
- [RozThi91] B.Rozoy, P.S.Thiagarajan; Event structures and trace monoids, *Theor. Comput. Sci.* 91 (1991), 285–313
- [Tau89] D.Taubner; *Finite Representations of CCS and TCSP Programs by Automata and Petri Nets*, Lecture Notes in Computer Science 369, Springer-Verlag, Berlin, 1989
- [Thi87] P.S.Thiagarajan; Elementary net systems, in [BraReiRoz87], pp.26–59
- [Thi90] P.S.Thiagarajan; Some behavioural aspects of net theory, *Theor. Comput. Sci.* 71 (1990), 133–153
- [Vaa91] F.W.Vaandrager; Determinism \rightarrow (event structure isomorphism = step sequence equivalence), *Theor. Comput. Sci.* 79 (1991), 275–294
- [Win87] G.Winskel; Event structures, in *Petri Nets: Applications and Relationships to other Models of Concurrency* (W.Brauer, W.Reisig, G.Rozenberg, eds.), Lecture Notes in Computer Science 255, Springer-Verlag, Berlin, 1987, pp.325–392
- [WinNie95a] G.Winskel, M.Nielsen; Models for concurrency, in *Handbook of Logic in Computer Science* (S.Abramsky, D.M.Gabbay, T.S.E.Maibaum, eds.), Vol.4: Semantic Modelling, pp.1–148, Clarendon Press, Oxford, 1995
- [WinNie95b] G.Winskel, M.Nielsen; Trace structures and other models for concurrency, Chapter 9 of [DieRoz95]

Index

- acyclic, 10, 80, 119
- (α, β) -configuration equivalent, 35
- (α, β) -isomorphic, 12, 20, 73, 74
- asymmetric confusion, 33

- b-graph, 104
- b-process, 101
- b-process net, 100
- β -isomorphic, 81
- bisimilarity, 38
- bisimulation, 38
- bounded, 121, 127
- branching graph, 104
- branching process, 101
- branching process net, 100

- capacity function, 113
- case, 14
- case graph, 20
- causal net, 65
- causality, 27
- cd confusion, 32
- characteristic p-invariant, 131
- characteristic pair, 17
- ci confusion, 32
- clique, 62
- complement construction, 56
- complementary, 56
- concession, 15, 18, 22, 114
- concurrency, 28
- concurrency relation, 62
- concurrency-free, 46
- concurrent step, 22
- condition, 14
- configuration, 13, 106, 113
- configuration equivalent, 35
- configuration graph, 25
- conflict, 29
- conflict relation, 100, 104
- conflict set, 31
- conflict-decreasing confusion, 32
- conflict-free, 29
- conflict-increasing confusion, 32

- confused, 31
- confusion, 30, 31
- contact, 58, 115
- contact-free, 60, 116
- contracted process, 83
- contracted version, 83, 105
- covered by invariants, 136
- covered by sequential components, 54
- covering, 54, 136
- covering set, 122
- covering tree, 122
- covering vector, 125
- cut, 63
- cycle, 139

- dense, 63
- dependency graph, 91
- dependency relation, 91
- diamond property, 24
- Dickson's Lemma, 118
- disjoint set of transitions, 22

- edge, 80
- edge-labelled graph, 20
- element, 9
- elementary net system, 13
- EN system, 13
- enabled, 15, 18, 22
- event, 14
- event structure, 104

- fire, 15, 18, 22, 114
- firing sequence, 18
- firing sequence equivalent, 40
- flow relation, 9
- free-choice system, 143

- graph, 20, 80

- independency relation, 91
- induced, 106
- initial configuration, 14, 113

initial node, 20
 inner product, 131
 input-concession, 15
 input-set, 10
 isolated place, 10
 isomorphic, 11, 20, 34, 73, 74, 81

 König's Lemma, 118

 labelled b-graph, 104
 labelled branching graph, 104
 labelled edge, 20
 labelled event structure, 104
 labelled graph, 80
 labelled net, 72
 labelled partial order, 81
 labelled partially ordered set, 81
 labelled process net, 72
 labelling, 80
 les-equivalent, 106
 line, 63
 line relation, 62
 linearly ordered set, 62
 live, 138
 live transition, 18
 lpo-equivalent, 84, 96

 marked, 139
 marked graph, 139
 marking, 13, 113
 maximal clique, 62
 multi-net, 113
 multi-set, 113
 mutual exclusion, 8

 neighbourhood, 10
 net, 9
 node, 20, 80
 node-labelled graph, 80

 observation equivalence, 38
 occurrence, 15
 occurrence net, 65
 ω -configuration, 122
 ω -places, 122
 output-concession, 15
 output-set, 10

 p-invariant, 131
 P-simple, 10
 P/T system, 2, 113
 partial order, 62
 partially ordered set, 62
 path, 139
 persistent, 30
 place, 9
 place invariant, 131
 place labelling, 72
 pom-set, 80
 positive p-invariant, 131
 post-set, 10
 pre-set, 10
 prefix-closed, 118
 principle of extensionality, 17
 process, 74
 process net, 64
 producer/consumer, 8
 producer/consumer problem, 109
 proof by induction, 18
 pruned contracted process, 83
 pruned contracted version, 83, 105
 pruned dependency graph, 91
 pruned version, 81, 105

 reachability problem, 129
 reachable configuration, 18
 readers/writers problem, 112
 reduced, 42
 represent, 81, 105

 S-component, 52
 S-system, 45
 safe, 60, 116
 scalar multiplication, 116
 sequential, 45
 sequential component, 47, 131
 sequential configuration graph, 20, 115
 sequential step, 15
 sequentialization property, 25
 set of labels, 20
 σ -clique, 62
 siphon, 144
 slice, 65, 101

state machine, 45
state machine component, 52
strongly reduced, 42
subsystem, 47
subsystem determined by, 47
sum, 116
symmetric confusion, 33

T-simple, 10
T-system, 139
token, 13
token game, 15
topological order, 88
trace, 98
trace equivalent, 97, 100
trace language, 99
transition, 9
transition labelling, 72
transition system, 20
transitive closure, 81, 105
transitive edge, 81
trap, 144
trivial subsystem, 47

underlying net, 14, 72
unfolding, 100
useful transition, 18

value, 139
Vector Addition System, 132

weakly configuration equivalent, 38
weight function, 113
word, 88

Woordenlijst Engels-Nederlands

Dit dictaat is recentelijk uit het Nederlands vertaald. De opgavebundel en alle oude tentamens bestaan echter (nog) alleen in het Nederlands. Bovendien zijn ook de docenten en recidivisten vertrouwd met de Nederlandse begrippen en wordt het college in het Nederlands gegeven. Om de overschakeling naar een (overigens ook flink aangevuld en sterk verbeterd) Engels dictaat voor eenieder te vergemakkelijken, volgt hieronder een woordenlijst Engels-Nederlands. Hierin staan de belangrijkste begrippen op alfabetische volgorde. Voor sommige begrippen wordt een nieuwe terminologie gehanteerd; in zo'n geval wordt de oude (Nederlandse) term tussen haakjes vermeld.

bounded	begrensd
branching	vertakkend
capacity function	capaciteitsfunctie
causality	causaliteit (sequentialiteit)
characteristic p-invariant	karakteristieke p-invariant
characteristic pair	karakteristieke paar
clique	clique
complementary	complementair
concession	concessie
concurrency	concurrency
configuration	configuratie (situatie)
configuration equivalent	configuratie-equivalent (situatie-equivalent)
configuration graph	configuratiegraaf (situatiegraaf)
conflict	conflict
conflict relation	conflict relatie
conflict set	conflictverzameling
conflict-decreasing confusion	conflict-verminderende verwarring
conflict-free	conflict-vrij
conflict-increasing confusion	conflict-vermeerderende verwarring
confused	verward
confusion	verwarring
contact	contact
contact-free	contact-vrij
contracted process	samengetrokken proces
contracted version	samengetrokken versie
covered by invariants	overdekt door invarianten
covered by sequential components	overdekt door sequentiële componenten
covering	overdekking
covering set	overdekkingsverzameling
covering tree	overdekkingsboom
covering vector	overdekkingsvector
cut	snedes

dense	dicht (K -dicht)
dependency graph	afhankelijkheidsgraaf
dependency relation	afhankelijkheidsrelatie
diamond property	ruiteigenschap
event	gebeurtenis
event structure	event structure
fire	vuren
firing sequence	vuurrij
firing sequence equivalent	vuurrij-equivalent
flow relation	stroomrelatie
free-choice	vrije keuze
free-choice system	vrije keuze systeem
independency relation	onafhankelijkheidsrelatie
induced	geïnduceerd
initial configuration	beginconfiguratie (beginsituatie)
input-concession	input-concessie
isolated place	geïsoleerde plaats
line	lijn
line relation	lijn relatie
live	levend
live transition	levende transitie
marked	gemarkeerd
marked graph	gemarkeerde graaf
marking	markering
mutual exclusion	wederzijdse uitsluiting
neighbourhood	omgeving
occurrence	optreden, plaatsvinden
output-concession	output-concessie
permission, permit	vergunning
place	plaats
place invariant	plaats-invariant
post-set	output-verzameling
pre-set	input-verzameling
principle of extensionality	extensionaliteitsprincipe
process	proces
process net	procesnet

pruned contracted process	gesnoeid samengetrokken proces
pruned contracted version	gesnoeide samengetrokken versie
pruned dependency graph	gesnoeide afhankelijkheidsgraaf
pruned version	gesnoeide versie
reachability problem	bereikbaarheidsprobleem
reachable configuration	bereikbare configuratie (bereikbare situatie)
record	registreren
reduced	gereduceerd
represent	representeren
safe	veilig
scalar multiplication	scalaire vermenigvuldiging
sequential	sequentieel
sequential component	sequentiële component
sequential configuration graph	sequentiële configuratiegraaf (sequentiële situatiegraaf)
sequential step	sequentiële stap
sequentialization property	sequentialisatie-eigenschap
siphon	sifon (deadlock)
slice	schijf
state	toestand
strongly reduced	sterk gereduceerd
subsystem	deelsysteem
token	teken
token game	tekenspel
trace	trace
transition	transitie
trap	val
useful transition	nuttige transitie
value	waarde
weakly configuration equivalent	zwak configuratie-equivalent (zwak situatie-equivalent)
weight function	gewichtsfunctie

These lecture notes were processed using the L^AT_EX macro package with LLNCS style