Elementary Net Systems

Grzegorz Rozenberg and Joost Engelfriet

Department of Computer Science, Leiden University
P.O.Box 9512, 2300 RA Leiden, The Netherlands
e-mail: {rozenber | engelfri}@wi.leidenuniv.nl

1 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]). 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 the framework of EN systems a concurrent system is seen as consisting of local states, local transitions (between local states), and the neighbourhood relationship between the local transitions and the local states. The global state of a system (its configuration) is simply the collection of all local states that (con)currently hold. The extent of change caused by a (local) transition is fixed and is restricted to the neighbourhood of the transition; it does not depend on the part of the global state that is outside that neighbourhood. This simple and elegant setup lends itself to a nice graphical representation of both the static structure of the system and its dynamic behaviour.

The EN system model has resulted from a number of modifications of the basic system model called Condition/Event Systems, or C/E systems, introduced by Petri (see, e.g., [Rei82, Thi87]). Perhaps the most significant difference is that in C/E systems transitions can also be reversed, recovering in this way the history of the system. An EN system can also be viewed as a special case of a P/T system.

In this chapter we present a comprehensive introduction to the theory of EN systems, covering both their structure and behaviour. The chapter is organized as follows. It consists of eight sections, of which this is the first. The second section recalls some standard mathematical notions needed in this chapter. Section 3 is a basic introduction to EN systems, both informally and formally. It begins with an informal introduction of EN systems. This is followed by the formalization of the notion of a net which represents the structure of an EN system. It consists of places (i.e., local states) and transitions (i.e., local transitions), connected by the neighbourhood relation. Then the dynamic execution of a net in both a sequential and a concurrent setting is discussed. An EN system is a net with an initial state, where all its executions begin. The sequential executions of the system that start from the initial state are called its firing sequences. Considering all states that are reachable by such a firing sequence leads to the state space of an EN system, formalized by the notion of a sequential configuration graph. The sequential configuration graph is extended to the (full) configuration graph of the system, in which also concurrent execution steps are represented. It is shown that the full configuration graph of an EN system is completely determined by its sequential configuration graph. Finally we demonstrate how fundamental situations of concurrent systems can be naturally expressed in the model of EN systems.

Section 4 discusses a number of structural and/or behavioural normal forms for EN systems. This is done in the framework of several fundamental behavioural equivalence relations for EN systems, based on their state spaces. A first, basic, normal form is that for every EN system an equivalent EN system can be constructed that has no redundant transitions or places. For such "reduced" EN systems, a structural characterization is given for the behavioural notion of a sequential EN system, i.e., an EN system of which every global state consists of one local state only. After formalizing the notion of a sequential component of an EN system (i.e., a subsystem that, on its own, is a sequential EN system), a second, more involved, normal form is shown: for every EN system an equivalent EN system can be constructed that can be viewed as a concurrent set of communicating sequential components. Such systems are conceptually easier to understand.

In the last three sections we turn to the partial order view of concurrent behaviour, where the partial order represents the causal dependencies between the events in a run of the EN system. In Section 5 the notion of a concurrent run of an EN system is formalized; as usual in Petri net theory, it is called a "process". It is one of the nice features of Petri net theory that such a process is in fact itself an EN system, called a process net (or causal net); it consists moreover of a mapping that labels the places and transitions of the process net with those of the given EN system. Through this mapping, the firing sequences of the processes correspond to a partition of the set of firing sequences of the EN system. Since a process net is acyclic, the transitions of a process naturally represent a partial order that is labelled by the transitions of the EN system. There is a one-to-one correspondence between the processes of an EN system and the corresponding labelled partial orders. EN systems with the same set of labelled partial orders are said to be lpo-equivalent; this is a natural concurrent behavioural equivalence of EN systems.

In Section 6.1 it is demonstrated that two EN systems are lpo-equivalent if and only if they have the same firing sequences. This shows that all various notions of behaviour considered in this chapter actually amount to two equivalences: configuration equivalence (where two EN systems have the same sequential configuration graph) and firing sequence equivalence (where two EN systems have the same firing sequences). It is remarkable that these two equivalences formalize the *sequential* behaviour of EN systems. This means that the concurrent behaviour of an EN system (as, e.g., formalized by lpo-equivalence) can be derived from its sequential behaviour. In Section 6.2 it is demonstrated that the above partition of the set of firing sequences of an EN system corresponds to the so-called trace-equivalence relation on firing sequences: two firing sequences are trace-equivalent if the one can be obtained from the other by the interchange of causally independent transitions.

Section 7 discusses the notion of a branching process of an EN system, which formalizes the combination of a finite number of concurrent runs of the system, showing where these runs are in conflict and, thus, have to branch. The theory of branching processes (or unfoldings) of EN systems is very similar to the theory of processes, with the notion of partial order replaced by the notion of event structure, which is a partial order with a conflict relation. Finally, Section 8 contains a conclusion.

We note here that we only consider the *finite* behaviour of an EN system, i.e., we only formalize finite executions or runs of the system. Thus, infinite executions or runs of the systems should be understood through their finite initial parts.

This chapter is based on [Rei82, RozThi86, Thi87, Roz87] and on the lecture notes for the lecture "Theory of Concurrency I" given for several years already at the Department of Computer Science of Leiden University, The Netherlands.

2 Preliminaries

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

The sets of non-negative and positive integers are denoted by $\mathrm{N} = \{0, 1, 2, \ldots\}$ and $\mathrm{N}_+ = \{1, 2, 3, \ldots\}$, 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. We consider total functions only, i.e., if $f : A \to A'$, then $f(a)$ is defined (and in A') for every $a \in A$. If $f : A \to A'$ and $B \subseteq A$, then $f \upharpoonright B$ denotes the restriction of f to B, i.e., the function $B \to A'$ defined by $(f \upharpoonright B)(b) = f(b)$. A function $f : A \to A'$ 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 $a_1, \ldots, a_n \in A$, 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 $h : \Sigma \to \Delta$. 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.

3 EN Systems

3.1 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 Fig. 1 shows an elementary net system 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

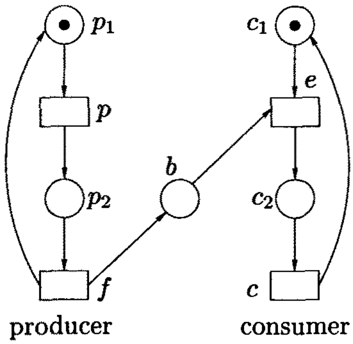


Fig. 1. The producer/consumer problem.

(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 Fig. 2 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

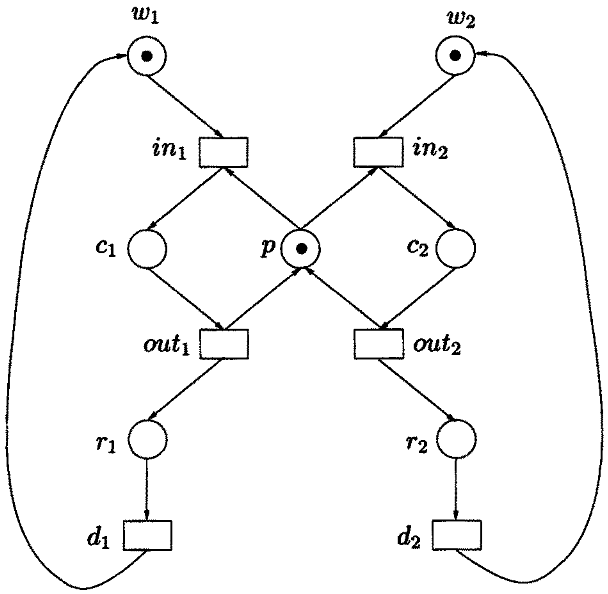


Fig. 2. The mutual exclusion problem.

used), c_1 (the resource is used by component 1), and c_2 (the resource is used by component 1), and can perform the actions in_i and out_i, $i = 1, 2$.

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

3.2 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 $P \cap T = \varnothing$,
(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., $N = (\varnothing, \varnothing, \varnothing)$.

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: ${}^\bullet t \neq \varnothing$, $t^\bullet \neq \varnothing$, and ${}^\bullet t \cap t^\bullet = \varnothing$. 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. 3.

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 $p \in P$, $\mathbf{nbh}(p) \neq \varnothing$.

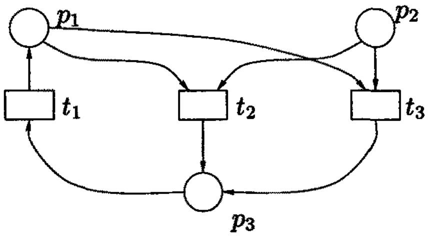


Fig. 3. A net.

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

Example 2. The net N from Example 1 (Fig. 3) 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. 4 is P-simple and T-simple. It has an isolated place and is cyclic.

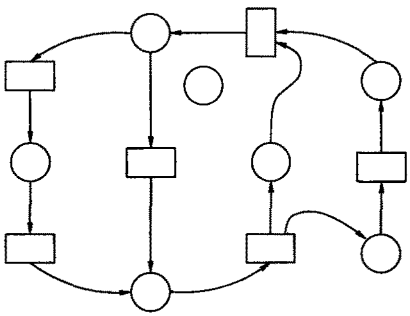


Fig. 4. A net that is P-simple and T-simple.

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 $\alpha : P \to P'$ and $\beta : T \to T'$, such that for every $p \in P$ and $t \in T$,
$(p, t) \in F$ iff $(\alpha(p), \beta(t)) \in F'$ and
$(t, p) \in F$ 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 $N \equiv_\beta^\alpha N'$. 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. 5 and 6, 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!

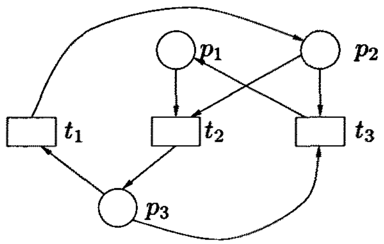


Fig. 5. A net N.

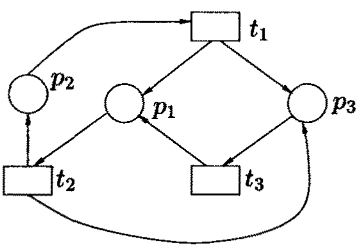


Fig. 6. A net N', not isomorphic with the net N of Fig. 5.

(2) Let N'' be the net in Fig. 7. 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 $N \equiv^{\alpha}_{\beta} N''$. The isomorphism (α, β) does preserve the sorts: it maps transitions into transitions, and places into places.

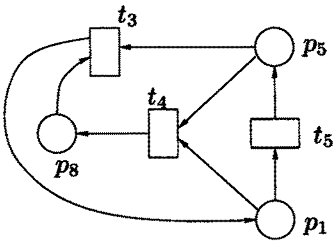


Fig. 7. A net N'', isomorphic with the net N of Fig. 5.

3.3 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., $M = (\varnothing, \varnothing, \varnothing, \varnothing)$.

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. 8. 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. 9. Note that M is the producer/consumer system of Fig. 1, with a full buffer.

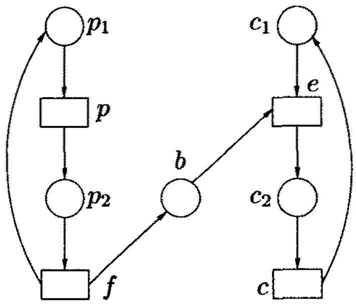


Fig. 8. The underlying net of the EN system of Fig. 9

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

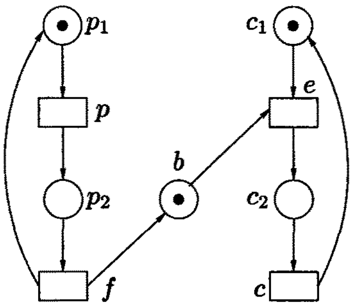


Fig. 9. An EN system.

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 $t^\bullet \cap C = \varnothing$, written as t **con** C.

(2) Let $C, D \subseteq P$. Then t *fires from* C *to* D if t **con** C and $D = (C - {}^\bullet t) \cup t^\bullet$, written as $C[t\rangle D$; 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 $C[t\rangle_M D$ instead of t **con** C and $C[t\rangle D$. In the literature $C[t\rangle$ is often written instead of t **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., $t^\bullet \cap C = \varnothing$). Note also that, though we write $C[t\rangle D$, the new configuration D is uniquely determined by C and t, i.e., if $C[t\rangle D_1$ and $C[t\rangle D_2$, 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. 10. In this way, the token marking that corresponds to the

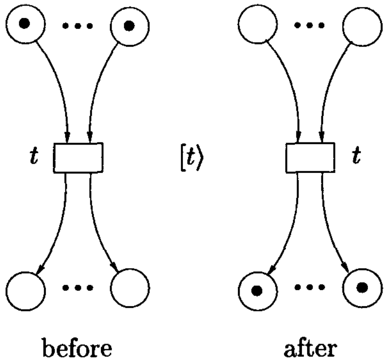


Fig. 10. Firing of transition t.

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 ${}^\bullet t \cap t^\bullet = \varnothing$ due to Definition 1(4)).

Example 5. Let M be the EN system from Example 4 (Fig. 9). Then transitions p and e have concession in C_{in}, i.e., p **con** C_{in} and e **con** C_{in}; transitions f and c do not have concession in C_{in}. Both $C_{in}[p\rangle\{p_2, b, c_1\}$ and $C_{in}[e\rangle\{p_1, c_2\}$ hold. The configuration $\{p_1, c_2\}$ is drawn in Fig. 11. Also, e.g., $\{p_2, b, c_1\}[e\rangle\{p_2, c_2\}$, $\{p_2, c_2\}[f\rangle\{p_1, b, c_2\}$, and $\{p_1, b, c_2\}[c\rangle C_{in}$ hold.

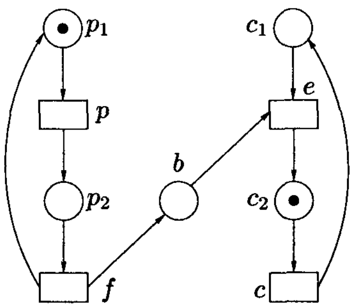


Fig. 11. The EN system of Fig. 9 after firing transition e.

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., $t^\bullet \cap C = \varnothing$). 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 ${}^\bullet t = \{p_1, \ldots, p_m\}$ and $t^\bullet = \{q_1, \ldots, q_n\}$ change their value:

if p_1 **and** $\cdots$ **and** p_m **and** (**not** q_1) **and** $\cdots$ **and** (**not** q_n)
then begin $p_1 :=$ false ; ... ; $p_m :=$ false ;
$q_1 :=$ true ; ... ; $q_n :=$ 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 $C[t\rangle D$ 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) $A \subseteq C$, $B \cap C = \varnothing$, and $D = (C - A) \cup B$, (2) $C - D = A$ and $D - C = B$. □

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 ($C[t\rangle D$ 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:

$(\forall t \in T)(\forall C_1, D_1, C_2, D_2 \subseteq P)$: if $C_1[t\rangle D_1$ and $C_2[t\rangle D_2$, then
$C_1 - D_1 = C_2 - D_2$ 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*:

$(\forall t_1, t_2 \in T)(\forall C_1, D_1, C_2, D_2 \subseteq P)$: if $C_1[t_1\rangle D_1$ and $C_2[t_2\rangle D_2$, then
$t_1 = t_2$ iff $C_1 - D_1 = C_2 - D_2$ 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 $t_1, \ldots, t_n \in T$. Let $C, D \subseteq P$. Then $t_1 \cdots t_n$ *fires from* C *to* D if there exist configurations $C_0, C_1, \ldots, C_n \subseteq P$ with $C_0 = C$, $C_n = D$ and $C_{i-1}[t_i\rangle C_i$ for all $1 \leq i \leq n$, written as $C[t_1 \cdots t_n\rangle D$.

(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 $C[x\rangle D$, written as x **con** C.

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

(4) $C \subseteq P$ is a *reachable configuration of* M if there exists an $x \in \mathrm{FS}(M)$ with $C_{in}[x\rangle C$. 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 **con** C. The set of useful transitions of M is denoted by $\mathbf{use}_M(T)$, or just $\mathbf{use}(T)$ 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 **con** C.

Note that a configuration C is reachable (i.e., $C \in \mathbb{C}_M$) if there exist transitions $t_1, \ldots, t_n \in T$ (with $n \geq 0$) and configurations $C_1, \ldots, C_n$ such that $C_n = C$ and $C_{in}[t_1\rangle C_1[t_2\rangle C_2 \cdots [t_n\rangle C_n$. 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 $C[t\rangle D$ 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 $C[t\rangle D$ 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. 9). 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 $\{p_1, b, c_1\}[pef\rangle\{p_1, b, c_2\}$. Consequently, pef has concession in $C_{in} = \{p_1, b, c_1\}$, written as pef **con** $\{p_1, b, c_1\}$, and pef is a firing sequence of M, written as $pef \in \mathrm{FS}(M)$. 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 $\{p_1, b, c_1\}$, $\{p_2, b, c_1\}$, $\{p_2, c_2\}$, $\{p_1, b, c_2\} \in \mathbb{C}_M$. Since c **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. 12 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.

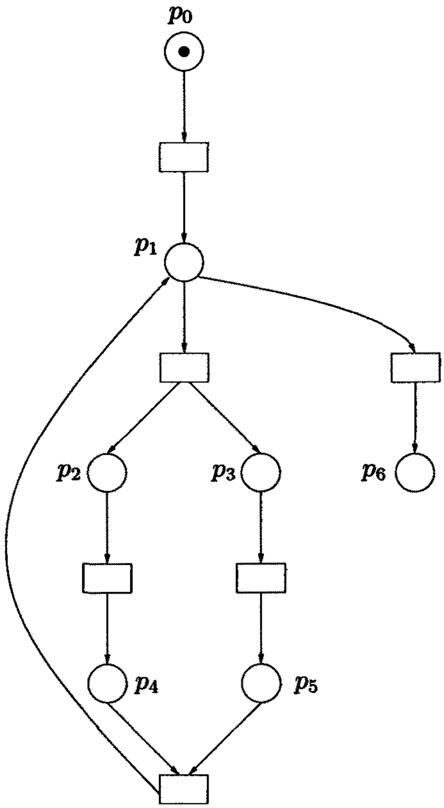


Fig. 12. An EN system of which all transitions are useful, but none is live.

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 $\mathbb{C}_M$ 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} 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 $\alpha : V_1 \to V_2$ and $\beta : \Sigma_1 \to \Sigma_2$ 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 $G_1 \equiv^{\alpha}_{\beta} G_2$.

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 $\mathrm{SCG}(M)$, 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 $\Gamma = \{(C, t, D) \mid C, D \in \mathbb{C}_M,\ t \in T_M,\ C[t\rangle_M D\}$.

Example 7. For the EN system M from Example 4 (Fig. 9), $\mathrm{SCG}(M)$ is given in Fig. 13. The "wriggly" 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 $\{a_1, a_2, \ldots, a_n\}$.

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 $\mathrm{SCG}(M)$ there must exist a path to a node with an outgoing edge labelled by t. Thus, for the EN system M of Fig. 9, $\mathrm{SCG}(M)$ of Fig. 13 shows that all transitions of M are live.

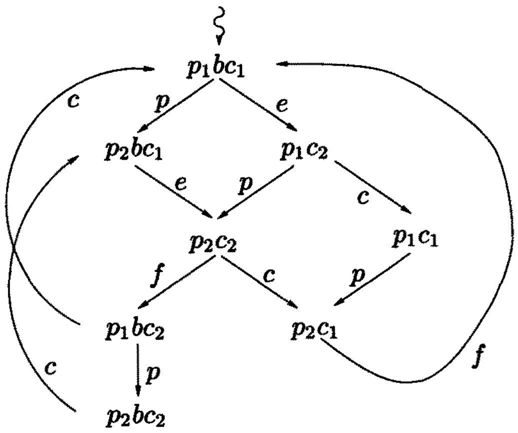


Fig. 13. A sequential configuration graph.

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

The sequential configuration graph 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 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 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 SCG(M) that starts in C_{in}. Hence, the firing sequences of M are precisely the words that are accepted by the finite automaton SCG(M). This gives the following result.

Theorem 12. *For every EN system M, FS(M) is a regular language.*

Proof. Let $M = (P, T, F, C_{in})$. Consider the (deterministic) finite automaton $\mathcal{A}$ with input alphabet **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 $\{(C, t, D) \mid C[t\rangle_M D\}$. Then FS($M$) is the language accepted by $\mathcal{A}$. □

Note that the language FS(M) is prefix-closed, i.e., if a word x is contained in FS(M), then also each prefix of x is contained in 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 $C[t\rangle D$, then ${}^\bullet t \cap D = \varnothing$, and hence t does not have concession in D).

3.4 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 5. 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 $U \neq \varnothing$ and for every two distinct transitions $t_1, t_2 \in U$: $\mathbf{nbh}(t_1) \cap \mathbf{nbh}(t_2) = \varnothing$; 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 $U^\bullet \cap C = \varnothing$; this is denoted by U **con** C.

(3) Let $U \subseteq T$ and let $C, D \subseteq P$. Then U *fires from* C *to* D, written as $C[U\rangle D$, if U **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* $C[U\rangle D$ *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. 9). Then $\{p, e\}$ is a disjoint set of transitions (thus $\mathbf{disj}(\{p, e\})$), 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 $\{p, e\}$ **con** $\{p_1, b, c_1\}$), and $\{p_1, b, c_1\}[\{p, e\}\rangle\{p_2, c_2\}$.

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 $U \neq \varnothing$. Then U* **con** *C iff*
(1) t **con** *C for all $t \in U$, and*
(2) for all $t_1, t_2 \in U$ with $t_1 \neq t_2$, ${}^\bullet t_1 \cap {}^\bullet t_2 = \varnothing$ and $t_1{}^\bullet \cap t_2{}^\bullet = \varnothing$.

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

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., $U = U_1 \cup U_2$, $U_1 \cap U_2 = \varnothing$ and $U_1, U_2 \neq \varnothing$. If $C[U\rangle D$, then there exists a configuration $E \subseteq P$ such that $C[U_1\rangle E$ and $E[U_2\rangle D$.*

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, $\mathbf{disj}(U_1)$ and $\mathbf{disj}(U_2)$ follow from $\mathbf{disj}(U)$. 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 $\mathbf{disj}(U)$ 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 $C[U_1\rangle E$ according to Lemma 14. Likewise $E - D = {}^\bullet U_2$ and $D - E = U_2{}^\bullet$, and thus $E[U_2\rangle D$ according to Lemma 14. □

This lemma expresses a so-called *diamond property*: if $C[U\rangle D$ and $\{U_1, U_2\}$ is a partition of U, then there exist two configurations E_1 and E_2 such that $C[U_1\rangle E_1[U_2\rangle D$ and $C[U_2\rangle E_2[U_1\rangle D$. A drawing (in Fig. 14) of the four steps $C[U_1\rangle E_1$, $E_1[U_2\rangle D$, $C[U_2\rangle E_2$, and $E_2[U_1\rangle D$ gives a "diamond"; the step $C[U\rangle D$ is then a diagonal of this diamond. More general diamond properties are studied in [HooRoz91].

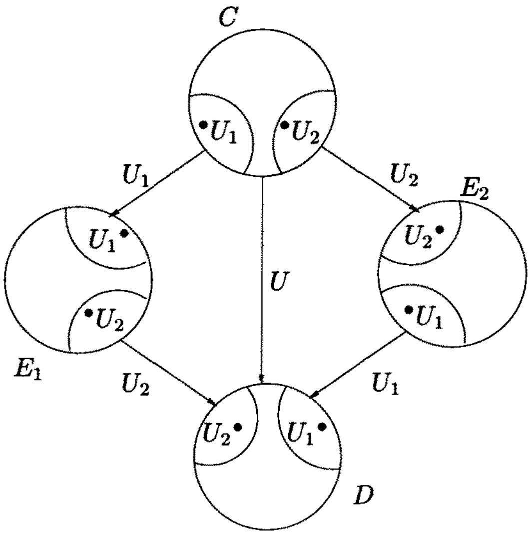


Fig. 14. A diamond.

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* $C[U\rangle D$, *then* $C[t_1 \cdots t_n\rangle D$ *for each ordering* $(t_1, \ldots, t_n)$ *of the elements of* U.

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

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 $\mathrm{CG}(M)$, 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 $\Gamma = \{(C, U, D) \mid C, D \in \mathbb{C}_M,\ U \subseteq T_M,\ C[U\rangle_M D\}$.

Note that if $C[U\rangle_M D$ 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. 9). The configuration graph $\mathrm{CG}(M)$ is drawn in Fig. 15. Compare $\mathrm{CG}(M)$ with the sequential configuration graph $\mathrm{SCG}(M)$ of Fig. 13. Also note that $\mathrm{CG}(M)$ contains several "diamonds" (see Fig. 14) such as, e.g., the diamond at the concurrent step $\{p_1, b, c_1\}[\{p, e\}\rangle\{p_2, c_2\}$.

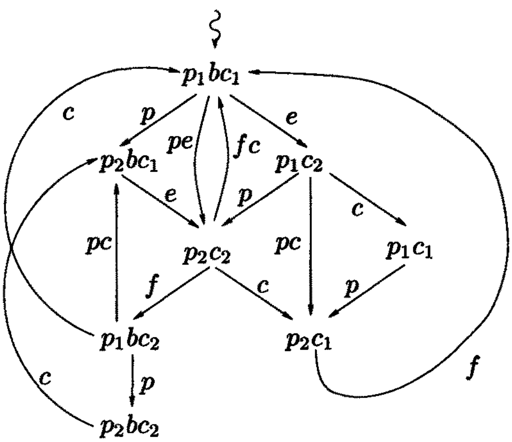


Fig. 15. A configuration graph.

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 **con** C *and* t **con** C, *then* $\{s, t\}$ **con** C.

Proof. Since st **con** C, s **con** C certainly holds. Since t **con** C also holds, it suffices according to Lemma 15 to show that ${}^\bullet s \cap {}^\bullet t = \varnothing$ and $s^\bullet \cap t^\bullet = \varnothing$. Let $C[s\rangle D$. Then ${}^\bullet s \cap D = \varnothing$ and $s^\bullet \subseteq D$. Since t **con** D, ${}^\bullet t \subseteq D$ and $t^\bullet \cap D = \varnothing$. Hence ${}^\bullet s \cap {}^\bullet t = \varnothing$ and $s^\bullet \cap t^\bullet = \varnothing$. □

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 $U \neq \varnothing$. Then*

*(1) U **con** C iff $t_1 \cdots t_n$ **con** C for every ordering $(t_1, \ldots, t_n)$ of the elements of U, and*

(2) $C[U\rangle D$ iff $C[t_1 \cdots t_n\rangle D$ for every ordering $(t_1, \ldots, t_n)$ 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 $(t_1, \ldots, t_n)$ of U with $t_1 = t$. This implies that t **con** C for all $t \in U$. What remains to be proved (see Lemma 15), is **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 **con** C. If we now consider an ordering of U of the form $(s, t, t_3, \ldots, t_n)$, then, by assumption, $stt_3 \cdots t_n$ **con** C. Hence st **con** C, and so, by Lemma 19, $\{s,t\}$ **con** C. Thus **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). □

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 CG(M) of an EN system M from its sequential configuration graph 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',*
SCG(M) $\equiv$ SCG(M') iff CG(M) $\equiv$ CG(M').

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

(Only-if) Assume that SCG(M) $\equiv^{\alpha}_{\beta}$ SCG(M'). This means that for all $C, D \in \mathbb{C}_M$ and $t \in$ **use**(T_M), $C[t\rangle_M D$ iff $\alpha(C)[\beta(t)\rangle_{M'}\alpha(D)$. It is now easy to prove, by induction on $|x|$, that for all $C, D \in \mathbb{C}_M$ and $x \in$ **use**$(T_M)^*$, $C[x\rangle_M D$ iff $\alpha(C)[\beta(x)\rangle_{M'}\alpha(D)$; note that, for $x = t_1 \cdots t_n$, $\beta(x) = \beta(t_1) \cdots \beta(t_n)$ according to Section 2. From this and Theorem 20 it follows that for all $C, D \in \mathbb{C}_M$ and $U \subseteq$ **use**(T_M), $C[U\rangle_M D$ iff $\alpha(C)[\beta(U)\rangle_{M'}\alpha(D)$. Thus CG($M$) $\equiv^{\alpha}_{\beta}$ CG(M'). □

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 $\mathcal{P}(\Sigma_2)$).

3.5 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. 16 and 17: t_2 **con** C does not hold, but $t_1 t_2$ **con** C does hold. Thus, t_1 needs to occur to grant concession to t_2 (input-concession in Fig. 16, and output-concession in Fig. 17). In other words, t_1 is one of the causes of t_2. An equivalent formal definition is: $t_1 t_2$ **con** C holds, but $t_1^\bullet \cap {}^\bullet t_2 \neq \varnothing$ or ${}^\bullet t_1 \cap t_2^\bullet \neq \varnothing$.

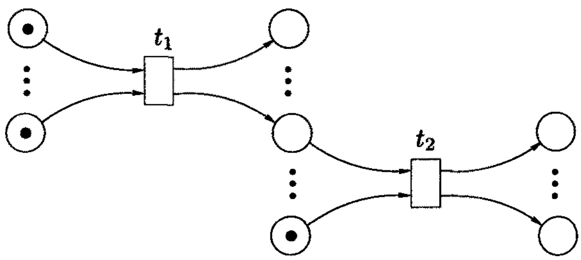


Fig. 16. Causality: $t_1^\bullet \cap {}^\bullet t_2 \neq \varnothing$.

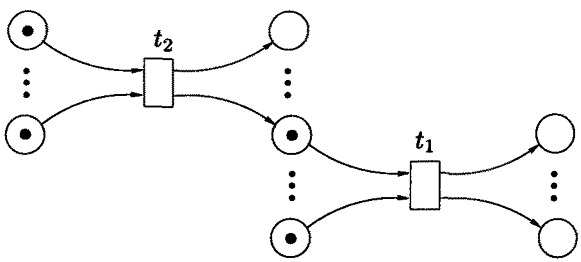


Fig. 17. Causality: ${}^\bullet t_1 \cap t_2^\bullet \neq \varnothing$.

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

This notion is illustrated in Fig. 18: $\{t_1, t_2\}$ **con** C. Figure 19 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.

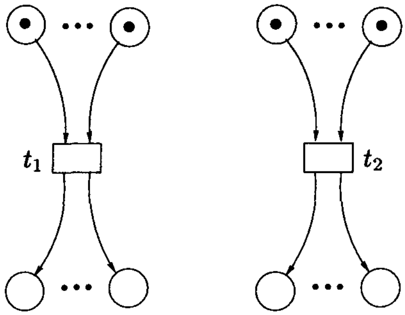


Fig. 18. Concurrency.

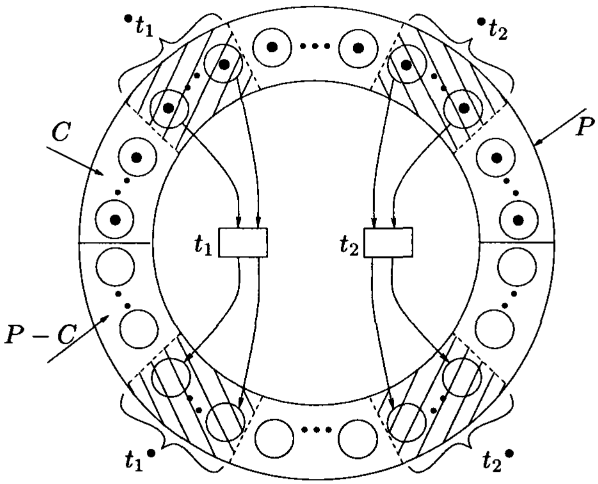


Fig. 19. Concurrency, the complete picture.

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 **con** C does not hold, then there is causality, and if t_2 **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. 20 and 21: both t_1 **con** C and t_2 **con** C hold, but $\{t_1, t_2\}$ **con** C does not hold. Consequently, by Lemma 15, ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \varnothing$ or $t_1{}^\bullet \cap t_2{}^\bullet \neq \varnothing$. If the former holds, then we have an input-conflict (represented in Fig. 20), and if the latter holds, then we have an output-conflict (represented in Fig. 21). Obviously, input-conflict and output-conflict can also both be present.

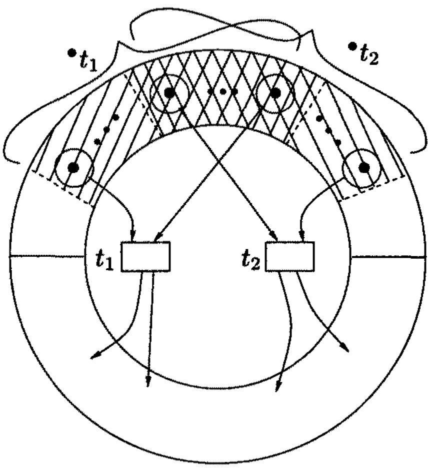


Fig. 20. Input-conflict.

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 **con** C and t_2 **con** C, then $\{t_1, t_2\}$ **con** C.

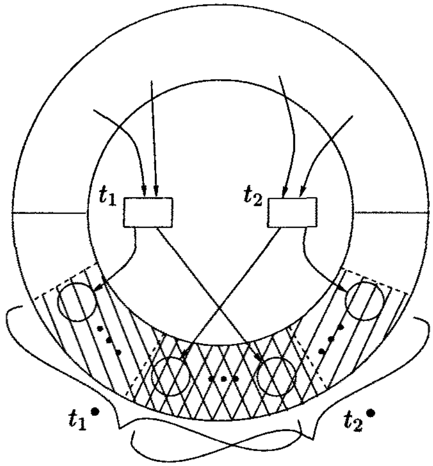


Fig. 21. Output-conflict.

Since in a conflict-free EN system choices are never made, it has only one run. The EN system of Fig. 1 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: $\{t_1, t_2\}$ **con** C follows from ${}^\bullet t_1 \cap {}^\bullet t_2 = \varnothing$ and $t_1{}^\bullet \cap t_2{}^\bullet = \varnothing$, 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. 22, hence $C_{in} = \{p_1, p_2, p_3\}$. Let $C = \{p_4, p_5\}$; thus $C_{in}[\{t_1, t_2\}\rangle C$. Different sequential realizations (t_1t_2 and t_2t_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_1t_2) and O_2 (corresponding to t_2t_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".

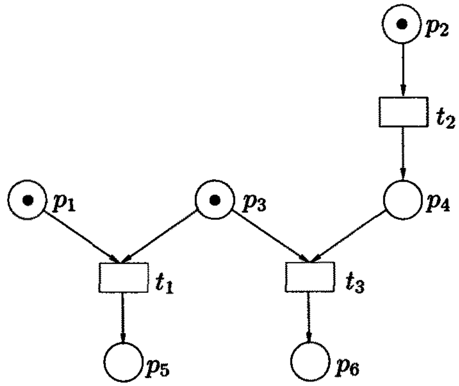


Fig. 22. A conflict-increasing confusion.

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 3.1 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 **con** C. The *conflict set of* t *in* C, denoted by $\mathbf{cfl}(t, C)$, is the set $\{t' \in T \mid t' \textbf{ con } C \text{ and } \neg \{t, t'\} \textbf{ con } C\}$.

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 $t_1 \neq t_2$, $\{t_1, t_2\}$ **con** C, and $\mathbf{cfl}(t_1, C) \neq \mathbf{cfl}(t_1, D)$, where $C[t_2\rangle D$. 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. 22. For the configuration $C = \{p_1, p_2, p_3\} = C_{in}$, $\mathbf{cfl}(t_1, C) = \varnothing$. Hence (C, t_1, t_2) is a confusion, because $\{t_1, t_2\}$ **con** C and $\mathbf{cfl}(t_1, C) = \varnothing \neq \{t_3\} = \mathbf{cfl}(t_1, D)$, 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 $C[t_2\rangle D$.

(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. 22). 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. 23. For $C = C_{in} = \{p_1, p_2\}$, (C, t_1, t_2) is a confusion because $\{t_1, t_2\}$ **con** C and $\mathbf{cfl}(t_1, C) = \{t_3\} \neq \varnothing = \mathbf{cfl}(t_1, D)$, 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.

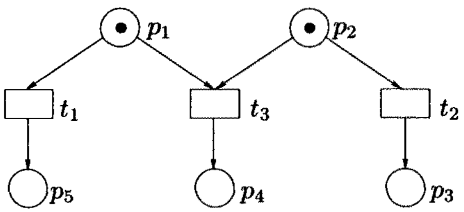


Fig. 23. A conflict-decreasing confusion.

(3) Consider the EN system in Fig. 24. 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.

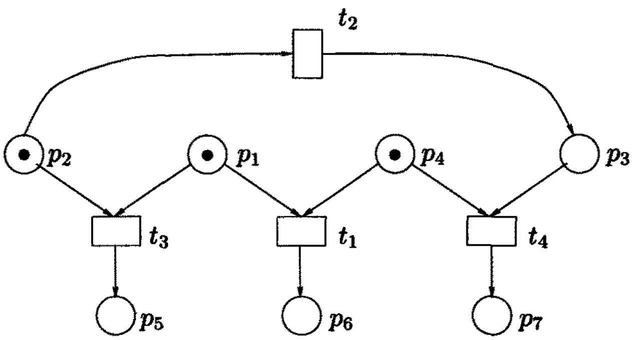


Fig. 24. A confusion that is neither conflict-increasing nor conflict-decreasing.

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. 22). 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. 25. 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.

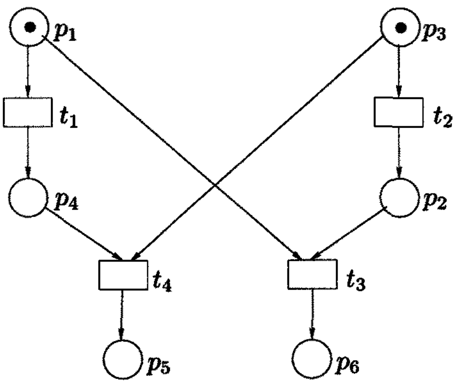


Fig. 25. A symmetric confusion.

(3) The confusion (C, t_1, t_2) from Example 11(2) (Fig. 23) is a symmetric cd confusion.

(4) The confusion (C, t_1, t_2) from Example 11(3) (Fig. 24) 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 **con** C and t is in conflict with both t_1 and t_2).

An EN system is said to be *free-choice* if, for all transitions t_1 and t_2, ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \varnothing$ implies ${}^\bullet t_1 = {}^\bullet t_2$. This is a structural restriction on EN systems that guarantees the absence of confusion as far as input-conflicts are concerned. Requiring additionally that the system is contact-free (see Section 4.5), it is not difficult to show that there is no confusion. The class of free-choice EN (and P/T) systems is a large class with many interesting properties, see [Hac72, DesEsp95].

4 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].

4.1 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 $\alpha : P \to P'$ and $\beta : T \to T'$ such that $\mathbf{und}(M) \equiv^{\alpha}_{\beta} \mathbf{und}(M')$ 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 correpondences 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 $\alpha : \mathbb{C}_M \to \mathbb{C}_{M'}$ and $\beta : \mathbf{use}_M(T) \to \mathbf{use}_{M'}(T')$ such that
(1) $\alpha(C_{in}) = C'_{in}$ and
(2) for all $C, D \in \mathbb{C}_M$ and $t \in \mathbf{use}_M(T)$, $C[t\rangle_M D$ iff $\alpha(C)[\beta(t)\rangle_{M'}\alpha(D)$.

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

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. 26 and M' the EN system of Fig. 27. 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 $\alpha : \mathbb{C}_M \to \mathbb{C}_{M'}$ and $\beta : T_M \to T_{M'}$ 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. 28 and 29. 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.

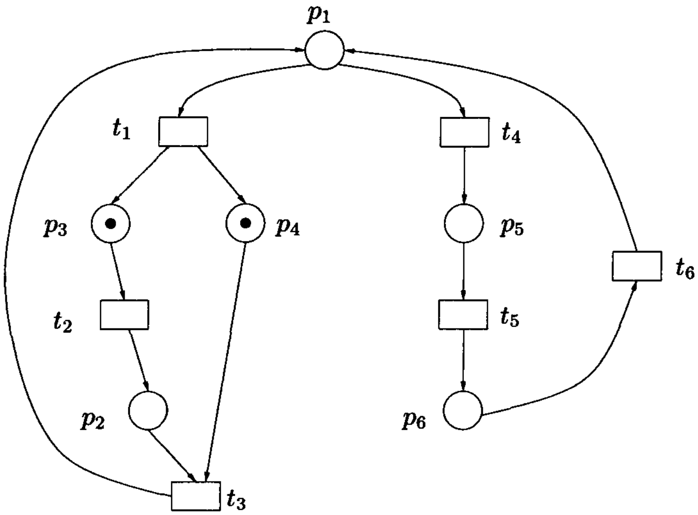


Fig. 26. An EN system M.

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, $\alpha : \mathbb{C}_M \to \mathcal{P}(P')$, and β is a bijective function, $\beta : \mathbf{use}_M(T) \to T'$, such that*
(1) $\alpha(C_{in}) = C'_{in}$ and
(2) for all $C, D \in \mathbb{C}_M$ and $t \in \mathbf{use}_M(T)$,
$C[t\rangle_M D$ implies $\alpha(C)[\beta(t)\rangle_{M'} \alpha(D)$, 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 $\alpha(C)[\beta(t)\rangle_{M'} \alpha(D)$. Then $\beta(t)$ $\mathbf{con}_{M'}$ $\alpha(C)$ and thus t $\mathbf{con}_M$ C. Let $C[t\rangle_M E$. Then $\alpha(C)[\beta(t)\rangle_{M'} \alpha(E)$. Hence $\alpha(E) = \alpha(D)$. Since α is injective, $E = D$ and thus $C[t\rangle_M D$.

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

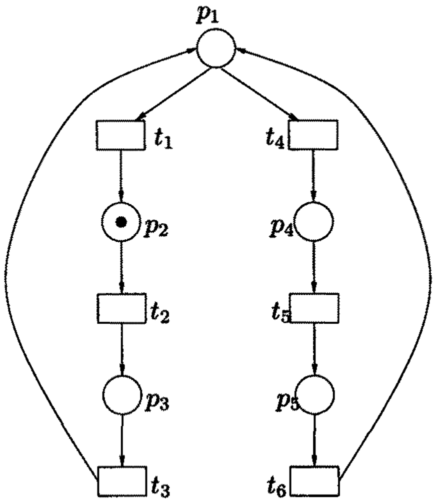


Fig. 27. An EN system M' configuration equivalent with the EN system M of Fig. 26.

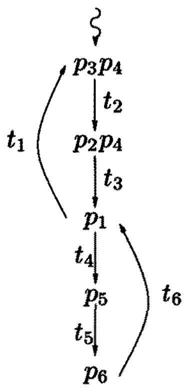


Fig. 28. The configuration graph of the EN system M of Fig. 26.

$\alpha(C) \in \mathbb{C}_{M'}$ and let $C[t\rangle_M D$. We have to prove that $\alpha(D) \in \mathbb{C}_{M'}$. From $C[t\rangle_M D$ and (2) it follows that $\alpha(C)[\beta(t)\rangle_{M'}\alpha(D)$. 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 $C'[t'\rangle_{M'}D'$. Since β is a bijection, there is a $t \in \mathbf{use}(T)$ with $\beta(t) = t'$. Hence $\beta(t)$ **con** $\alpha(C)$ and thus t **con** C according to (2). Let $C[t\rangle_M D$. Hence $D \in \mathbb{C}_M$. Then $\alpha(C)[\beta(t)\rangle_{M'}\alpha(D)$; in other words

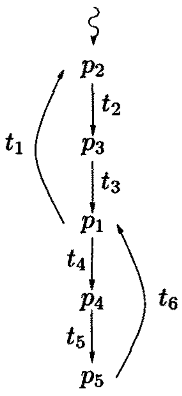


Fig. 29. The configuration graph of the EN system M' of Fig. 27.

$C'[t'\rangle_{M'}\alpha(D)$, 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 $C[t\rangle_M D$. Hence $\alpha(C)[\beta(t)\rangle_{M'}\alpha(D)$. Since $\alpha(C) \in \mathbb{C}_{M'}$, $\beta(t) \in \mathbf{use}_{M'}(T')$. □

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 $\beta : \mathbf{use}(T) \to \mathbf{use}(T')$, 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 $C[t\rangle_M D$ and $(C, C') \in \alpha$, then there is a $D' \in \mathbb{C}_{M'}$ such that $C'[\beta(t)\rangle_{M'} D'$ 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 $C'[t'\rangle_{M'} D'$ and $(C, C') \in \alpha$, then there is a $D \in \mathbb{C}_M$ such that $C[\beta^{-1}(t')\rangle_M D$ 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 $C[U\rangle_M D$ and $C'[\beta(U)\rangle_{M'} D'$.

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. 30 and 32, respectively; their (sequential) configuration graphs are given in Figs. 31 and 33, respectively. Since SCG(M) has less nodes than SCG(M'), SCG(M) and 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.

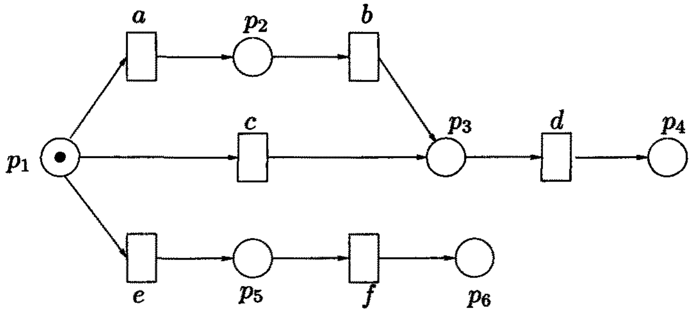


Fig. 30. An EN system M, weakly equivalent with the EN system M' of Fig. 32.

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 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 $\beta : \mathbf{use}(T_M) \to \mathbf{use}(T_{M'})$, where M and M' are EN systems, then $\beta(\text{FS}(M)) \subseteq \mathbf{use}(T_{M'})^*$, see Section 2.

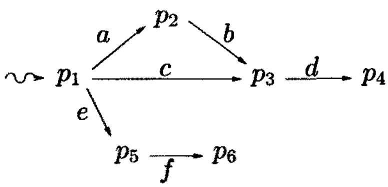


Fig. 31. The configuration graph of the EN system M of Fig. 30.

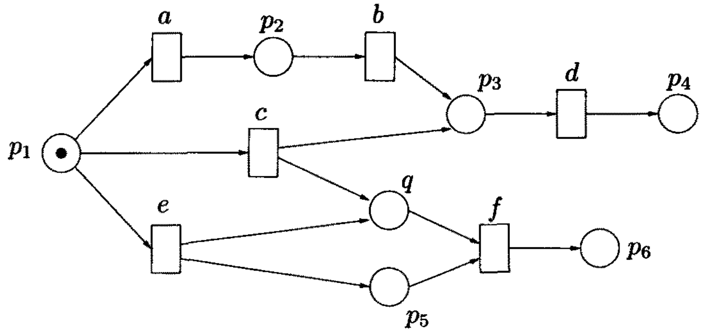


Fig. 32. An EN system M', weakly equivalent with the EN system M of Fig. 30.

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 $\beta : \mathbf{use}(T_M) \to \mathbf{use}(T_{M'})$ such that $\beta(\mathrm{FS}(M)) = \mathrm{FS}(M')$.

Example 15. The (weakly configuration equivalent) EN systems M and M' of Example 14 (Figs. 30 and 32) are firing sequence equivalent because $\mathrm{FS}(M) = \mathrm{FS}(M') = \{\lambda, a, c, e, ab, cd, ef, abd\}$.

Even though firing sequence equivalence abstracts completely from the configurations, it turns out to be the same as weak configuration equivalence! This

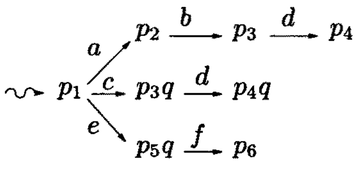


Fig. 33. The configuration graph of the EN system M' of Fig. 32.

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 $\beta(\mathrm{FS}(M)) \subseteq \mathrm{FS}(M')$: by symmetry this also proves that $\beta^{-1}(\mathrm{FS}(M')) \subseteq \mathrm{FS}(M)$.

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 $x \in \mathbf{use}(T_M)^*$:

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

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

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 $\alpha \subseteq \mathbb{C}_M \times \mathbb{C}_{M'}$:

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

It is straightforward to verify that α is a bisimulation, i.e., that α and β satisfy the three conditions in Definition 31. For condition (1) take $x = \lambda$. For condition (2), assume that $C_{in}[x\rangle_M C$ and $C'_{in}[\beta(x)\rangle_{M'} C'$, and that $C[t\rangle_M D$. Then $C_{in}[xt\rangle_M D$. Thus xt is a firing sequence of M, and so $\beta(xt)$ is a firing sequence of M'. Hence there exists $D' \in \mathbb{C}_{M'}$ such that $C'_{in}[\beta(xt)\rangle_{M'} D'$. Clearly $(D, D') \in \alpha$. It remains to prove that $C'[\beta(t)\rangle_{M'} D'$. Since $\beta(xt) = \beta(x)\beta(t)$, there exists $E' \in \mathbb{C}_{M'}$ such that $C'_{in}[\beta(x)\rangle_{M'} E'$ and $E'[\beta(t)\rangle_{M'} D'$. Since the firing of $\beta(x)$ leads from C'_{in} to a unique configuration, $E' = C'$ and thus indeed $C'[\beta(t)\rangle_{M'} D'$. Condition (3) can be proved analogously. □

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 $\mathrm{FS}(M)$ and $\mathrm{FS}(M')$ 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',

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

4.2 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. □

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 $\mathbf{use}(P) = \{p \in P \mid \mathbf{nbh}(p) \neq \varnothing\}$; 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 $\alpha : \mathbb{C}_M \to \mathcal{P}(P')$ with $\alpha(C) = C \cap P'$ for all $C \in \mathbb{C}_M$ and let β be the identity on T. We prove that $M \approx^{\alpha}_{\beta} M'$ 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 $C[t\rangle_M D$ iff $C \cap P'[t\rangle_{M'} D \cap P'$. 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. □

Example 16. According to the constructions in the proofs of Theorems 36 and 37, the EN system M in Fig. 34 is transformed into the equivalent strongly reduced EN system M' of Fig. 35: first the useless transition t_3 is removed and then the isolated places p_7 and p_8 are removed.

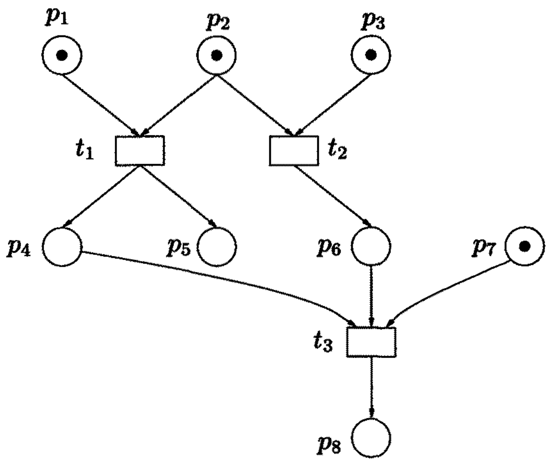


Fig. 34. An EN system with useless transition t_3.

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 $p \notin D$.*

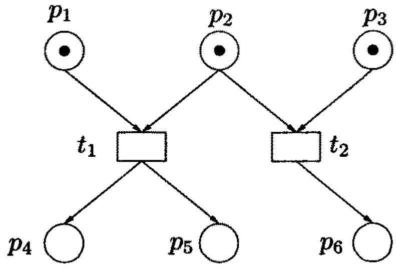


Fig. 35. A strongly reduced EN system, configuration equivalent with the EN system of Fig. 34.

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 **con** E. Let $E[t\rangle E'$. If $t \in p^\bullet$, then $p \in E$ and $p \notin E'$. And if $t \in {}^\bullet p$, then $p \in E'$ and $p \notin E$. □

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 $xsus \in \mathrm{FS}(M')$, then $s^\bullet \subseteq {}^\bullet u$ and ${}^\bullet s \subseteq u^\bullet$. This is proved as follows. Since $xsus \in \mathrm{FS}(M')$, there exist configurations C, D_1, D_2, and E of M' such that $C[s\rangle D_1$, $D_1[u\rangle D_2$, and $D_2[s\rangle E$. Since $D_1 - D_2 = {}^\bullet u$, $s^\bullet \subseteq D_1$, and $s^\bullet \cap D_2 = \varnothing$, we have $s^\bullet \subseteq {}^\bullet u$. Since $D_2 - D_1 = u^\bullet$, ${}^\bullet s \cap D_1 = \varnothing$, and ${}^\bullet s \subseteq D_2$, we have ${}^\bullet s \subseteq u^\bullet$.

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 $\mathrm{FS}(M') = \mathrm{FS}(M)$ (i.e., that β is the identity). Since $sus \in \mathrm{FS}(M) = \mathrm{FS}(M')$, 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 $susu \in \mathrm{FS}(M')$ 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. □

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 $x \in \mathrm{FS}(M)$ there exists $y \in T_M^*$ such that $xyt \in \mathrm{FS}(M)$. This implies that if $\beta(\mathrm{FS}(M)) = \mathrm{FS}(M')$, 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.

4.3 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 3.3).

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. 27 and 30 are sequential. This can be seen from their configuration graphs, which are given in Figs. 29 and 31, 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 $C \in \mathbb{C}_M$. □

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* $\mathbb{C}_M = \{\{p\} \mid p \in P\}$.

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 3.3), i.e., that concurrent steps do not occur (cf. the discussion of the notions of sequentiality and concurrency in Section 3.5). EN systems satisfying this property are "concurrency-free".

Definition 44. An EN system M is *concurrency-free* if there do not exist $C \in \mathbb{C}_M$ and $t_1, t_2 \in T_M$ such that $\{t_1, t_2\}$ **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. 26 and 32 are concurrency-free (as can be seen from their configuration graphs in Figs. 28 and 33, 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. To show this we observe that for a sequential EN system $M = (P, T, F, C_{in})$ and for $x, y, z \in T^*$ and $t \in T$, if $xt, yt, xz \in \text{FS}(M)$ then $yz \in \text{FS}(M)$; this is because if $C_{in}[x\rangle C$ and $C_{in}[y\rangle D$, then $C = {}^\bullet t = D$. The concurrency-free EN system of Fig. 36 does not satisfy this property: take $x = t_1$, $y = t_2$, $t = t_3$, and $z = t_3 t_2$.

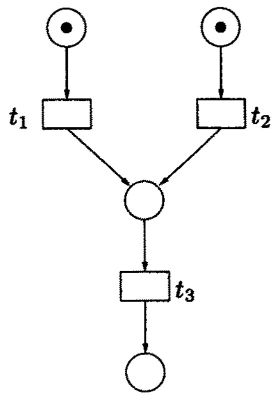


Fig. 36. A concurrency-free EN system that is not firing sequence equivalent with any sequential EN system.

Finally we observe that Theorem 43 does not hold for concurrency-free systems: the EN systems M (Fig. 26) and M' (Fig. 27) from Example 13 are configuration equivalent, but not isomorphic.

4.4 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 inital 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) $\forall p \in P'$: if $p \in C_{in}$, 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 $(\varnothing, \varnothing, \varnothing, \varnothing)$ 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 $t \in T - T'$, $\mathbf{nbh}_M(t) \cap P' = \varnothing$,
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: $(t^\bullet)_{M'} = \varnothing$ (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. □

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 $t \in T$, ${}^\bullet t \cap S \neq \varnothing$ iff $t^\bullet \cap S \neq \varnothing$. 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. 37 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. 9 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 3.1, where the buffer was viewed as a component of the system. In Example 23 (Fig. 47) we will see that there is a slight variation of M in which the buffer is also a sequential component.

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 $M = (P,T,F,C_{in})$.*

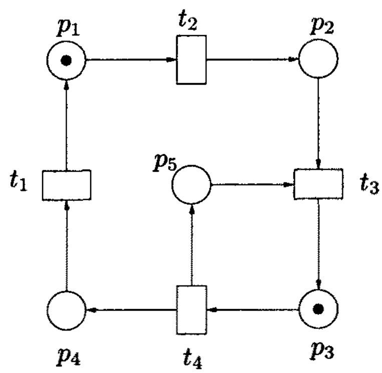


Fig. 37. An EN system with two nontrivial subsystems: $\{p_3, p_5\}$ and $\{p_1, p_2, p_3, p_4\}$.

(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 $C[t\rangle_M D$. Then we claim the following: if $t \notin T'$ then $D \cap S = C \cap S$, and if $t \in T'$ then $(C \cap S)[t\rangle_{M'}(D \cap S)$. 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), $({}^\bullet t)_M \cap S = \varnothing$ and $(t^\bullet)_M \cap S = \varnothing$, 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, $(C \cap S)[t\rangle_{M'}(D \cap S)$. 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 $C[t\rangle_M D$ 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 $(C \cap S)[t\rangle_{M'}(D \cap S)$. 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 $C[t\rangle_M D$. By (1), $C \cap S \in \mathbb{C}_{M'}$. Thus, by the above claim, $(C \cap S)[t\rangle_{M'}(D \cap S)$. Hence t $\mathbf{con}_{M'}$ $C \cap S$. □

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. 38. Its configuration graph is drawn in Fig. 39.

(1) Figure 40 shows a subsystem M_1 of M, and Fig. 41 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.

(2) Figure 42 shows a subsystem M_2 of M (and of M_1), and Fig. 43 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}.

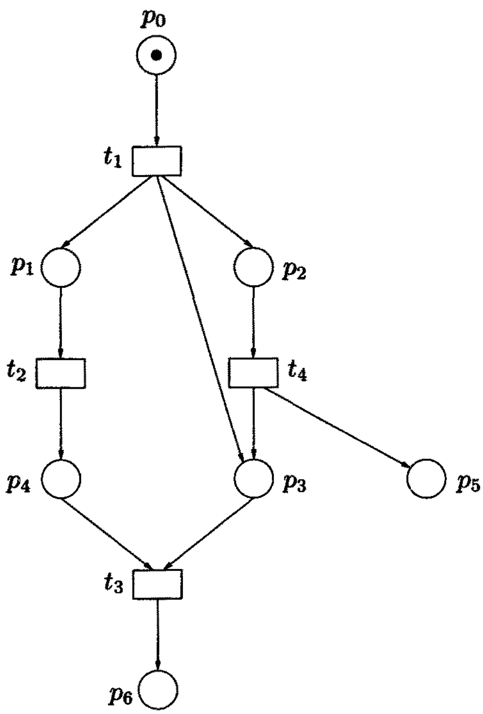


Fig. 38. An EN system M.

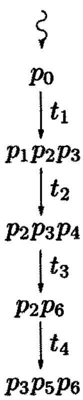


Fig. 39. The configuration graph of M.

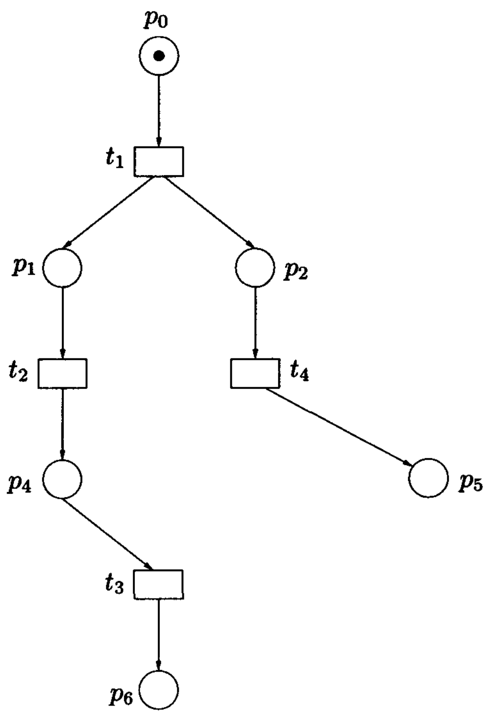


Fig. 40. A subsystem M_1 of M.

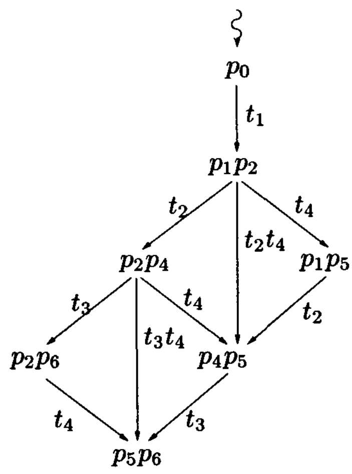


Fig. 41. The configuration graph of M_1.

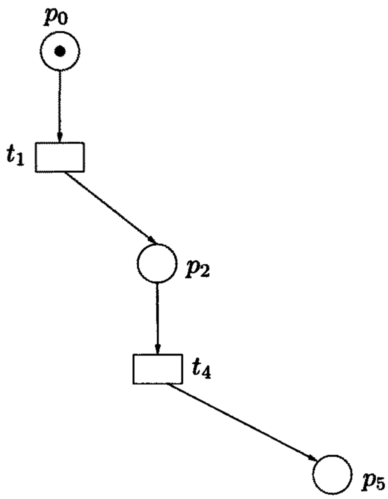


Fig. 42. A subsystem M_2 of M.

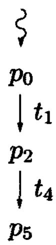


Fig. 43. The configuration graph of M_2.

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 $\#({}^\bullet t \cap S) = \#(t^\bullet \cap S) = 0$.

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 **con** C. Since $\#(C \cap S) = 1$, $\#({}^\bullet t \cap S) \leq 1$. Now let $C[t\rangle D$. 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. □

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 $S = \varnothing$ or $S = P$. Assume that $S \neq \varnothing$; then it remains to prove that $S = P$. Since M is strongly reduced, $\mathbb{C}_M = \{\{p\} \mid p \in P\}$ according to Lemma 42(2). Thus for all $t \in T$ and $p, q \in P$, if $\{p\}[t\rangle_M\{q\}$, then ($p \in S$ iff $q \in S$). This implies that for all $x \in T^*$ and $p, q \in P$, if $\{p\}[x\rangle_M\{q\}$, then ($p \in S$ iff $q \in S$). Now let $C_{in} = \{p_0\}$. Take a $q \in S$. Since $\{q\} \in \mathbb{C}_M$, $\{p_0\}[x\rangle_M\{q\}$ for some $x \in T^*$. Thus $p_0 \in S$. To prove that $P \subseteq S$, consider an arbitrary $p \in P$. Since $\{p\} \in \mathbb{C}_M$, $\{p_0\}[x\rangle_M\{p\}$ for some $x \in T^*$. Thus $p \in S$, and so $S = P$. □

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. □

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. 38). Figure 44 shows a subsystem M_3 of M that is not sequential. It is straightforward to verify that M_3 has only trivial subsystems.

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]).

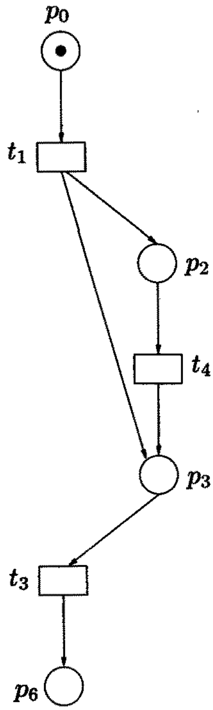


Fig. 44. A subsystem M_3 of M.

Definition 52. Let $M = (P, T, F, C_{in})$ be an EN system.
(1) A set $\{M_1, \ldots, M_n\}$ 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 $\{M_1, \ldots, M_n\}$, $n \geq 0$, 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 $\{M_1, \ldots, M_n\}$ is a covering of M iff $P = \bigcup_{i=1}^{n} S_i$.*

An EN system that is covered by sequential components $M_1, \ldots, M_n$ can intuitively be viewed as a system consisting of the communicating concurrent subsystems $M_1, \ldots, M_n$, 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 $S_1, \ldots, S_n$ of $M_1, \ldots, M_n$, 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. 45 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.

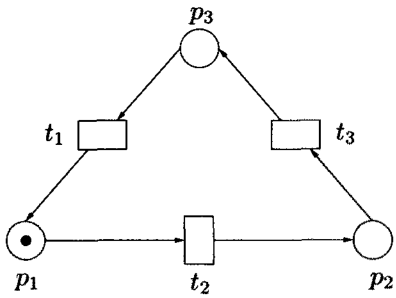


Fig. 45. A sequential EN system.

(2) The only sequential component of the EN system of Fig. 37 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. 12 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. 2 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. 9 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. 38, and M_1 and M_3 are the subsystems of M in Figs. 40 and 44, 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 **com** q, if $p^\bullet = {}^\bullet q$ and ${}^\bullet p = q^\bullet$.

Example 22. Let M be the EN system of Fig. 37. It is clear that p_3 **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 **com** q.*

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

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
$P' = P \cup \{q_0\}$, where q_0 is a new place (i.e., $q_0 \notin P \cup T$),
$T' = T$,
$F' = F \cup \{(q_0, t) \mid (t, p_0) \in F\} \cup \{(t, q_0) \mid (p_0, t) \in F\}$, and
$C'_{in} = C_{in}$ if $p_0 \in C_{in}$, and $C'_{in} = C_{in} \cup \{q_0\}$ if $p_0 \notin C_{in}$.
To prove that M and M' are configuration equivalent, we define the function $\alpha : \mathbb{C}_M \to \mathcal{P}(P')$ 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 $t \in T$,
$t\ \mathbf{con}_{M'}\ \alpha(C)$ implies $t\ \mathbf{con}_M\ C$, and (*)
$C[t\rangle_M D$ implies $\alpha(C)[t\rangle_{M'}\alpha(D)$. (**)
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). □

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.

Example 23. (1) The EN system of Fig. 46 is obtained from the one of Fig. 37 by complementing places p_1, p_2, and p_4. (2) The EN system of Fig. 47 is obtained from the EN system of Fig. 9 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 3.1. 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).

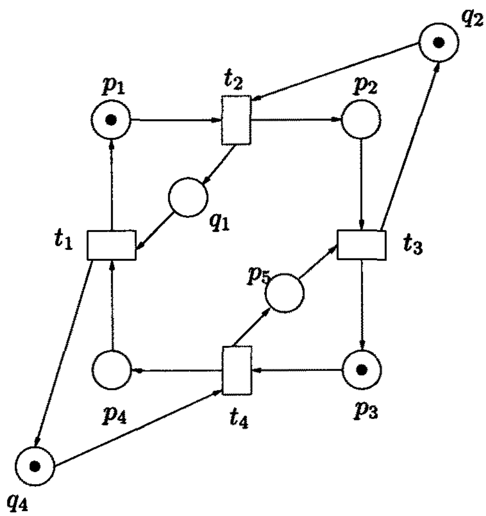


Fig. 46. The result of complementing places p_1, p_2, and p_4 of the EN system of Fig. 37.

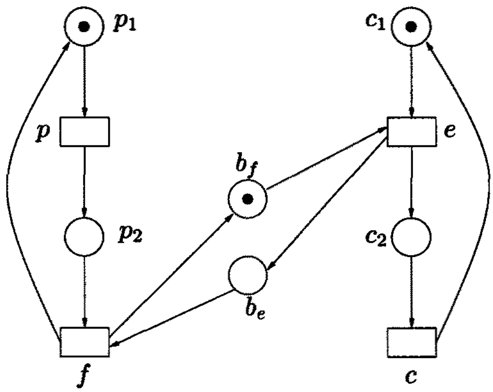


Fig. 47. The producer/consumer system with three sequential components.

4.5 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. 48. As an example, transition f has contact in configuration $\{p_2, b, c_1\}$ of the producer/consumer system in Fig. 9; 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. 47 transition f does not have input-concession.

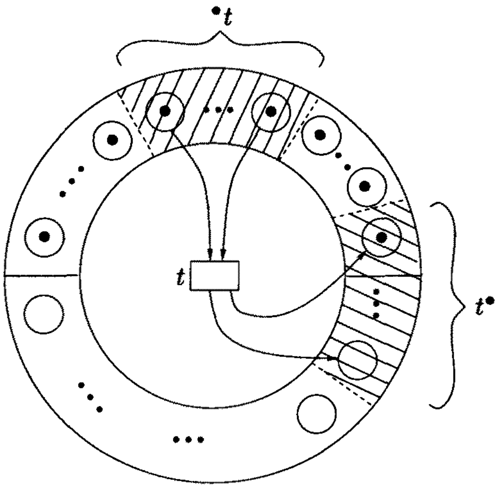


Fig. 48. Contact.

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 $t^\bullet \cap C = \varnothing$.

Contact-free EN systems are also called *safe* EN systems. In fact, they are the same as the safe P/T systems (assuming that the nets of P/T systems are defined as in Definition 1).

Example 24. The EN system of Fig. 45 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 $t^\bullet \cap C \neq \varnothing$. 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). □

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. 49 has no sequential components, but it is contact-free. By (the proofs of) Theorems 3.15 and 3.18 of [DesEsp95] (see also Section 7.3 of [Rei82]), the converse of Theorem 59 *does* hold for the so-called *T-systems* (or *marked graphs*); in such systems both $p^\bullet$ and ${}^\bullet p$ contain exactly one transition, for every $p \in P$. In fact, assuming the EN system to be strongly reduced, the converse of Theorem 59 even holds for the (larger) class of EN systems that are free-choice (see the end of Section 3.5) and that have live transitions only (see Definition 8(6)); see Section 4.2 of [Hac72] or Theorem 5.6 of [DesEsp95] (the S-coverability Theorem). In these cases, the sequential components are even strongly connected.

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. 45, 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. 12 is covered by two sequential components (see Example 21(3)). Hence M is contact-free.

(3) The EN system M of Fig. 2 is covered by three sequential components (see Example 21(4)) and is thus also contact-free.

(4) The EN systems of Figs. 46 and 47 are covered by sequential components (see Example 23) and are thus contact-free.

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. 47. 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 $C \in \mathbb{C}_M$ and all $U \subseteq T$ with $U \neq \varnothing$, if ${}^\bullet U \subseteq C$, then U **con** *C.*

If two transitions of a contact-free EN system have an output-conflict, in some configuration, then they also have an input-conflict in that configuration (cf. (3) of Section 3.5). Using this it can be shown that there is no confusion in contact-free free-choice systems (see the end of Section 3.5).

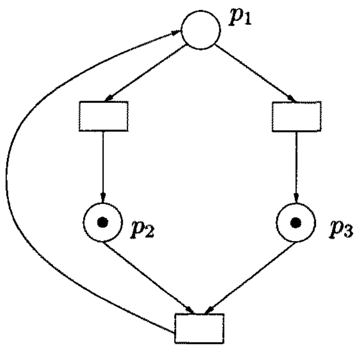


Fig. 49. A contact-free EN system without sequential components.

5 Processes

The way in which we have formalized the concurrent behaviour of an EN system in Section 3.4 still has a sequential flavour. The only difference with the sequential behaviour in Section 3.3 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 3.5). 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 5.1 we recall a number of notions concerning partial orders, and in Section 5.2 we consider process nets and some of their formal properties. In Sections 5.3 and 5.4 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]).

5.1 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 $\mathbf{li}_\rho$ and $\mathbf{co}_\rho$.

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), $\mathbf{li}_\rho$ is called the *line relation of* ρ and $\mathbf{co}_\rho$ 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 $\mathbf{li}_\rho$ and $\mathbf{co}_\rho$ play an important part in what follows. Maximal cliques are now defined for arbitrary reflexive symmetric relations (and note that $\mathbf{li}_\rho$ and $\mathbf{co}_\rho$ 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 $B \subseteq B_n$. □

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 $\mathbf{li}_\rho$-clique is a linearly ordered subset of A, and a $\mathbf{co}_\rho$-clique is a set of mutually incomparable elements.

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

Note that if $A = \varnothing$ (and thus $\rho = \varnothing$), then the empty set is both a line and a cut of ρ. If $A \neq \varnothing$, 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 nonempty 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
$({}^{\rightarrow}B)_\rho = \{a \in A \mid \exists b \in B : a \; \rho \; b \text{ or } a = b\}$,
$(B^{\rightarrow})_\rho = \{a \in A \mid \exists b \in B : b \; \rho \; a \text{ or } b = a\}$,
$({}^{\circ}B)_\rho = \{b \in B \mid \neg\exists b' \in B : b' \; \rho \; b\}$, and
$(B^{\circ})_\rho = \{b \in B \mid \neg\exists b' \in B : b \; \rho \; b'\}$.

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 $\rho^{-1} = \{(a, b) \mid (b, a) \in \rho\}$. □

Lemma 66 is a special case of (the second inclusion of) Lemma 70, for the partially ordered set $(\mathbb{C}, \subsetneq)$ where $\mathbb{C}$ 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. □

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 $A \neq \varnothing$ and let L be a line of ρ. Then $L \cap {}^\circ A \neq \varnothing$ and $L \cap A^\circ \neq \varnothing$.*

Proof. By duality it suffices to prove that $L \cap {}^\circ A \neq \varnothing$. 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 $m = a$). □

5.2 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 3.2) 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. □

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 $X_N{}^\circ$ (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 $\alpha : P_N \to P_{N'}$, then, obviously, $\alpha({}^\circ N) = {}^\circ N'$. In other words: then they are also isomorphic as EN systems (cf. Definition 27).

Example 26. Fig. 50 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.

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 ${}^\bullet U \subseteq C$.*

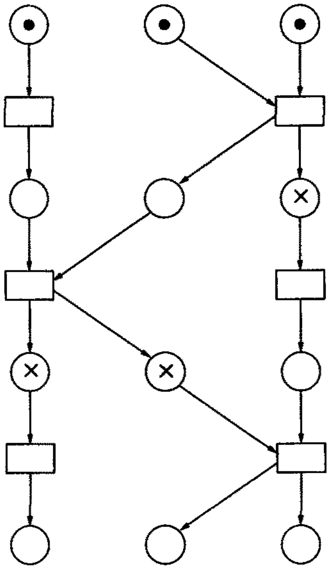


Fig. 50. A process net.

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

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

(4) For every slice C and every transition t, if $t \in {}^{\to}C$, then $\mathbf{nbh}(t) \subseteq {}^{\to}C$.

(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, $t^\bullet \cap C = \varnothing$. 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 $C \neq {}^\circ N$, $T \cap {}^{\rightarrow}C \neq \varnothing$. 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 $t' \in T \cap {}^{\rightarrow}C$. □

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 ${}^{\rightarrow}C \cap T = \varnothing$, then $C = {}^\circ N$. If ${}^{\rightarrow}C \cap T \neq \varnothing$, 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), ${}^\bullet t \cap C = \varnothing$ 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 $D[t\rangle C$ and hence $C \in \mathbb{C}_N$. □

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 $U \neq \varnothing$. 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 $U \subseteq T$, $(\exists C \in \mathbb{C}_N : U$ **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 **con** C for a $C \in \mathbb{C}_N$. The other way around, if U **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. □

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 $\mathbf{li}_N$-clique, then $L \cup \mathbf{nbh}(L \cap P)$ is a $\mathbf{li}_N$-clique.
(2) If L is a line of N, then for every $t \in T$:
${}^\bullet t \cap L \neq \varnothing$ *iff* $t \in L$ *iff* $t^\bullet \cap L \neq \varnothing$.

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 ${}^\bullet t \cap L \neq \varnothing$ iff $t \in L$ (the proof that $t \in L$ iff $t^\bullet \cap L \neq \varnothing$ 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 $Y \neq \varnothing$, and so, by Lemma 70, $Y^\circ \neq \varnothing$. 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 $p \in {}^\bullet t \cap L$. □

Theorem 83. *Let $N = (P, T, F, {}^\circ N)$ be a process net with $P \neq \varnothing$.*
(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 **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 **li** q does not hold. Now consider $t \in T - S^\bullet$. Then ${}^\bullet t \cap S = \varnothing$. 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 **co** q for every $p \in {}^\bullet t$ it follows that t **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 $L \cap C \neq \varnothing$. 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 $C[t\rangle D$ and ${}^\bullet t \cap L \neq \varnothing$, then $t^\bullet \cap L \neq \varnothing$. □

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. 50) 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 $L \cap C \neq \varnothing$ 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 $P \neq \varnothing$, 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), $L \cap C' \neq \varnothing$. 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 $t \in L \cap C$. □

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

5.3 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. 16. 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. 51. We start by recording the initial configuration $\{p_1, p_2\}$ (see Fig. 52). Next we record the occurrence of event t_1 (see Fig. 53). 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. 54. 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. 55). Finally we can (for example)

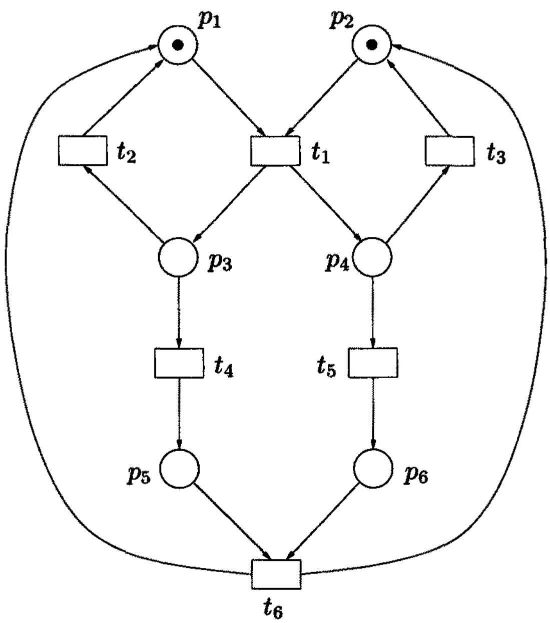


Fig. 51. An EN system M.

Fig. 52. A process of the EN system M of Fig. 51.

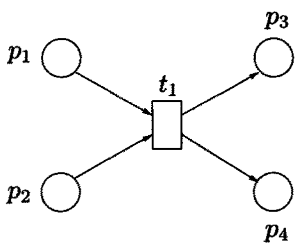


Fig. 53. A process that extends the process of Fig. 52.

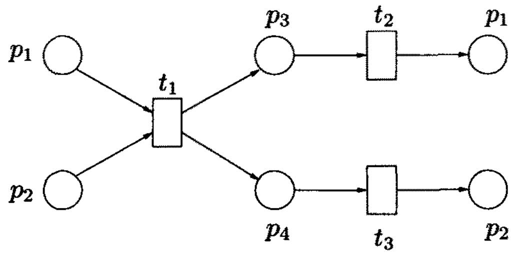


Fig. 54. A process that extends the process of Fig. 53.

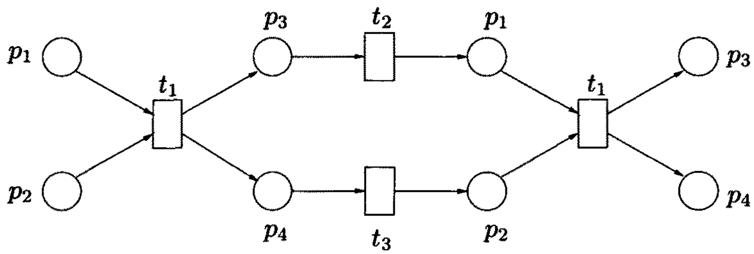


Fig. 55. A process that extends the process of Fig. 54.

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. 56).

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 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

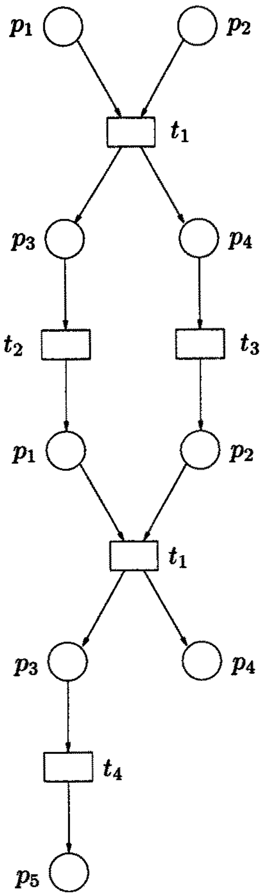


Fig. 56. A process of the EN system M of Fig. 51, leading from $\{p_1,p_2\}$ to $\{p_4,p_5\}$.

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 $\alpha : \Sigma_1 \to \Sigma_1'$, $\beta : \Sigma_2 \to \Sigma_2'$, $\gamma : P \to P'$, and $\delta : T \to T'$, 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 $\alpha : \Sigma_1 \to \Sigma_1'$ and $\beta : \Sigma_2 \to \Sigma_2'$, 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 2 that $f \restriction B$ denotes the restriction of function $f : A \to A'$ to the set $B \subseteq A$. The requirement that $f \restriction B$ 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) $\phi_1 \restriction {}^\circ N$ is injective,
(3) $\phi_1({}^\circ N) = C_{in}$,
(4) for every $t \in T_N$, $\phi_1 \restriction {}^\bullet t$ is injective and $\phi_1 \restriction t^\bullet$ 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, 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. 47. A process N of M is drawn in Fig. 57 (the underlying process net is the one of Fig. 50). 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. 2 (mutual exclusion) is drawn in Fig. 58. It leads from the initial configuration $\{w_1, p, w_2\}$ to the configuration $\{c_1, w_2\}$.

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 $\phi \restriction C$ is injective, $\phi(C) \in \mathbb{C}_M$, and $C[t\rangle_N D$, then $\phi \restriction D$ is injective and $\phi(C)[\phi(t)\rangle_M \phi(D)$.*

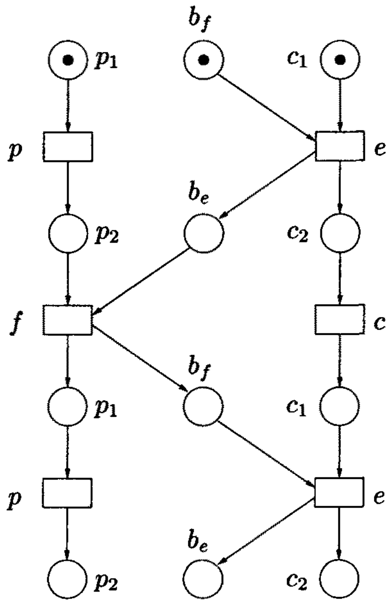


Fig. 57. A process of the producer/consumer system of Fig. 47.

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, $\phi(t)^\bullet \cap \phi(C) = \varnothing$ and hence, again by requirement (5), $\phi(t^\bullet) \cap \phi(C) = \varnothing$. This, and the fact that $\phi \restriction t^\bullet$ is injective (requirement (4)), implies that $\phi \restriction (C \cup D)$ 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 $\phi(C)[\phi(t)\rangle_M \phi(D)$. □

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 $C \in \mathbb{C}_N$, $\phi \restriction C$ is injective and $\phi(C) \in \mathbb{C}_M$.
(2) For every $C, D \in \mathbb{C}_N$ and $t \in T_N$, if $C[t\rangle_N D$ then $\phi(C)[\phi(t)\rangle_M \phi(D)$.

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. □

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, $\phi \restriction D$ is injective.*
(2) For every $C, D \in \mathbb{C}_N$ and $U \subseteq T_N$, if $C[U\rangle_N D$ then $\phi(C)[\phi(U)\rangle_M \phi(D)$.
(3) For every **co**-*clique $U \subseteq T_N$ there exist $C, D \in \mathbb{C}_M$ such that $C[\phi(U)\rangle_M D$.*

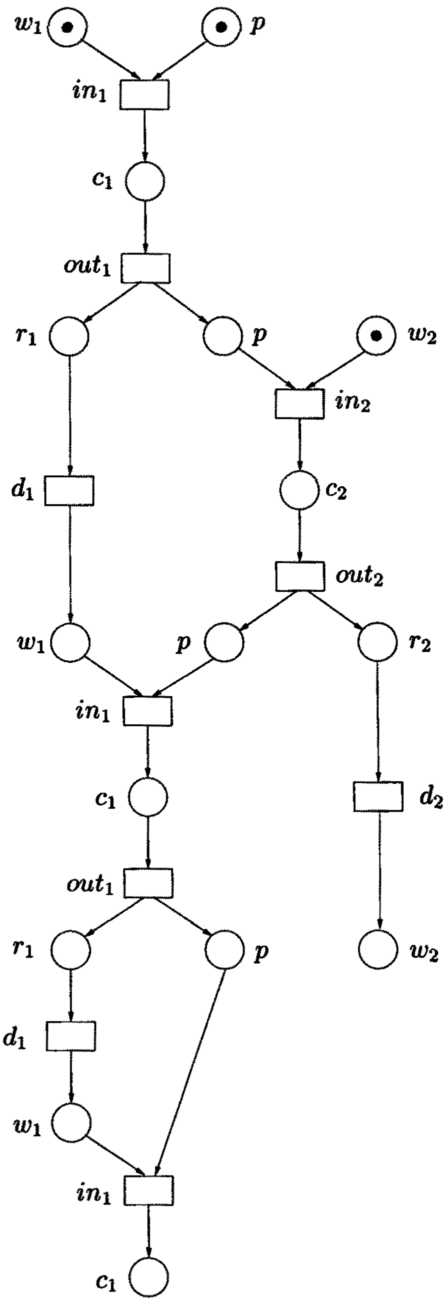


Fig. 58. A process of the mutual exclusion system of Fig. 2.

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 $\phi \restriction \{x,y\}$ 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, $\phi \restriction ({}^\bullet x \cup {}^\bullet y)$ 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 $C[U\rangle_N D$. By Theorem 81(2), U is a **co**-clique. Then (1) implies that $\phi \restriction U$ is injective. This means that the set of orderings of the elements of $\phi(U)$ equals $\{(\phi(t_1), \ldots, \phi(t_n)) \mid (t_1, \ldots, t_n)$ is an ordering of the elements of $U\}$. This, together with Theorems 20(2) and 90(2), implies that $\phi(C)[\phi(U)\rangle_M \phi(D)$.

(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). □

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. 52 to 56).

Theorem 92. *Let $M = (P,T,F,C_{in})$ be a contact-free EN system, let $t_1, \ldots, t_n$ be transitions in T, and let $C \subseteq P$. Then $C_{in}[t_1 \cdots t_n\rangle_M C$ iff there exists a process $N = (P_N, T_N, F_N, \phi_1, \phi_2)$ of M and there exist transitions $s_1, \ldots, s_n$ in T_N such that*
(1) $\phi(s_i) = t_i$ for $1 \leq i \leq n$,
(2) $\phi(N^\circ) = C$, and
(3) ${}^\circ N[s_1 \cdots s_n\rangle_N N^\circ$.

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 $C_{in} = \{q_1, \ldots, q_m\}$, then such a process N is defined by: $P_N = \{p_1, \ldots, p_m\}$, $T_N = \varnothing$, $F_N = \varnothing$, $\phi_2 = \varnothing$, and $\phi_1(p_i) = q_i$ for $1 \leq i \leq m$.

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

It is also straightforward to prove that, for given firing sequence $t_1 \cdots t_n$, the process N is unique (modulo isomorphism, i.e., modulo $\equiv_\beta^\alpha$, 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 $t_1 \cdots t_n$ can lead to the same process N. The resulting equivalence relation between firing sequences is studied in Section 6.2. 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, $\mathrm{PROC}(M) \neq \varnothing$ (since λ is always a firing sequence of M).

Example 28. Consider the EN system M of Fig. 47 and the process N of M of Fig. 57. Examples of firing sequences of M that correspond to process N in the way indicated in Theorem 92 are: *pefcep* and *ecpfpe*.

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 $C[t\rangle_M D$, then there exists a process N of M and there exist $C', D' \in \mathbb{C}_N$ and $s \in T_N$, such that $C'[s\rangle_N D'$, $\phi_N(C') = C$, $\phi_N(s) = t$, and $\phi_N(D') = D$.*

Proof. There is a firing sequence x such that $C_{in}[x\rangle_M C$. If we now apply Theorem 92 to $C_{in}[xt\rangle_M D$, 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 ${}^\circ N[ys\rangle_N N^\circ$ (where $\phi = \phi_N$). Assume that ${}^\circ N[y\rangle_N C'[s\rangle_N N^\circ$ and let $D' = N^\circ$. By Theorem 90, $\phi({}^\circ N)[\phi(y)\rangle_M \phi(C')$ holds, i.e., $C_{in}[x\rangle_M \phi(C')$. Hence $\phi(C') = C$. □

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 $C[U\rangle_M D$, 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 $C'[V\rangle_N D'$, $\phi_N(C') = C$, $\phi_N(V) = U$, 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 $M \equiv^\alpha_\beta M'$, then $\mathrm{PROC}(M) \equiv^\alpha_\beta \mathrm{PROC}(M')$. Now assume that $\mathrm{PROC}(M) \equiv^\alpha_\beta \mathrm{PROC}(M')$. Then α and β are bijections, $\alpha : P \to P'$ and $\beta : T \to T'$. 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 $\mathrm{PROC}(M) \neq \varnothing$) 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 $C[t\rangle_M D$. 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 $\alpha(t^\bullet) = \beta(t)^\bullet$. □

5.4 Pruned Contracted Processes

We can interpret the set $\mathrm{PROC}(M)$ 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 $\mathrm{PROC}(M)$, 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 4.1), 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 $\beta : \Sigma \to \Sigma'$ and $\delta : V \to V'$, 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 $\beta : \Sigma \to \Sigma'$, 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 59 shows a graph with its transitive closure and its pruned version. The labels have been omitted.

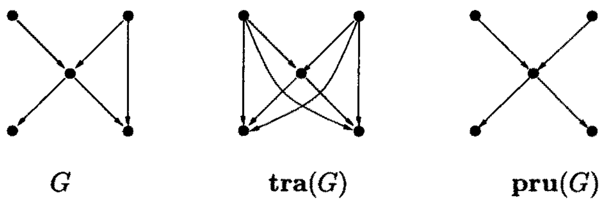


Fig. 59. A graph with its transitive closure and its pruned version.

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. □

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 $\beta : \Sigma \to \Sigma'$ 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. □

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. 16.

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 $s^\bullet \cap {}^\bullet t \neq \varnothing$.

(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 $(s,t) \in F^+$.

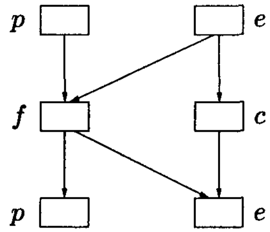
If N is a process of an EN system M, then $\mathbf{ctr}(N)$ is called a *contracted process of* M and $\mathbf{pru}(\mathbf{ctr}(N))$ a *pruned contracted process of* M. For a contact-free EN system M we denote by $\mathrm{LPO}(M)$ the set of all pruned contracted processes of M (where LPO stands for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

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

Example 30. (1) Let M be the EN system of Fig. 47 (the producer/consumer system) and let N be the process of M in Fig. 57. Then $\mathbf{ctr}(N) = \mathbf{pru}(\mathbf{ctr}(N))$ is given in Fig. 60. A larger pruned contracted process of M is drawn in Fig. 61.

(2) Let M be the EN system of Fig. 2 (the mutual exclusion system) and let N be the process of M in Fig. 58. Then $\mathbf{ctr}(N)$ and $\mathbf{pru}(\mathbf{ctr}(N))$ are given in Fig. 62.

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 $\mathrm{LPO}(M)$, modulo isomorphism.



or

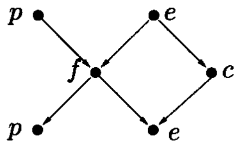


Fig. 60. The (pruned) contracted version of the process of Fig. 57.

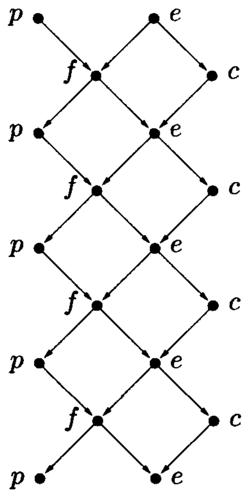


Fig. 61. Another pruned contracted process of the producer/consumer system of Fig. 47.

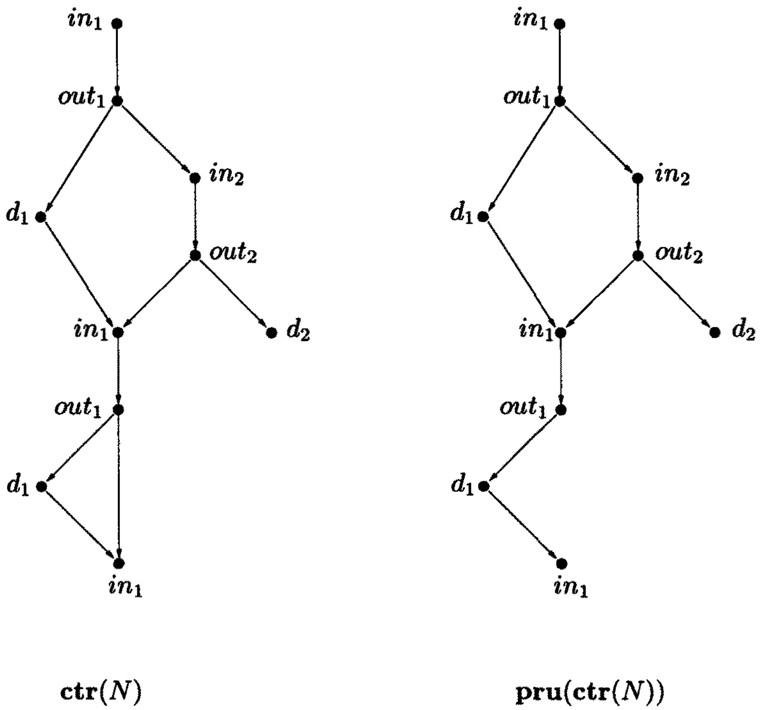


Fig. 62. The contracted process and the pruned contracted process corresponding to the process of Fig. 58.

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

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 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. 30 and 32 have isomorphic sets of contracted processes, and those contracted processes are already pruned, i.e., contain no transitive edges. Hence $\text{LPO}(M) \equiv \text{LPO}(M')$. Note that in this case the contracted processes correspond to $\text{FS}(M) = \text{FS}(M') = \{\lambda, a, c, e, ab, cd, ef, abd\}$, see Example 15.

Another example of two lpo-equivalent EN systems M and M' that are not isomorphic is depicted in Fig. 63. The sets of contracted processes of M and M' are not isomorphic, but the sets of pruned contracted processes are. □

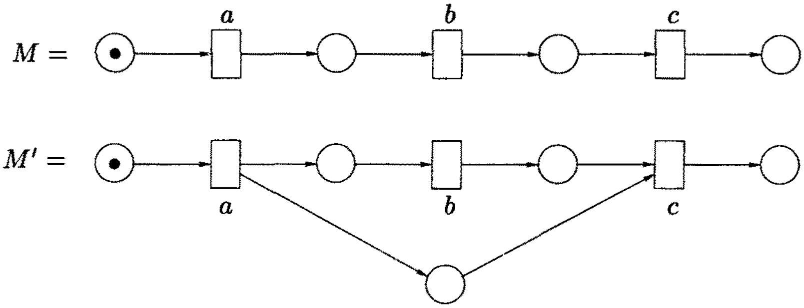


Fig. 63. Two non-isomorphic lpo-equivalent EN systems.

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

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 $f : \text{PROC}(M) \to \text{PROC}(M')$ 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 **ctr** and **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 ${}^\bullet q_1 = \varnothing$ *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 **co** q_2 does not hold. Hence q_1 **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. □

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* ${}^\bullet q = \varnothing$, $s_2 \in q^\bullet$, *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). □

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 **pructr** : PROC(M) → LPO(M) is a bijection modulo isomorphism, it suffices to define a function **proc** : LPO(M) → PROC(M) such that

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

It is straightforward to define the function **proc**, on the basis of Lemma 109, as follows. Let $G = (T, \Gamma, \mathbf{use}(T_M), \phi)$ be an element of LPO(M). Then **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 **proc**(G), and the places $q_{s,p}$ form the post-set of the transition s in **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

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

and all pairs (q_p, s_2) such that

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

This ends the definition of the function **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 **proc**(**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 **proc**(**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 **proc**(**pructr**(N)) and N. Note that this also shows that **proc**(G) is in PROC(M), for every $G \in$ LPO(M).

This ends the proof of Theorem 107.

6 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].

6.1 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) $V = \{u_1, \ldots, u_n\}$, 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., $\mathbf{top}(G) \neq \varnothing$. 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 $\mathbf{tra}(G) = \mathbf{tra}(G')$, *then* $\mathbf{top}(G) = \mathbf{top}(G')$ *and* $\mathbf{words}(G) = \mathbf{words}(G')$.

Example 31. Consider the acyclic labelled graph G shown in Fig. 64, with $V_G = \{v_1, \ldots, v_6\}$ 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 $v_3v_4v_1v_2v_6v_5$.

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

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

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.

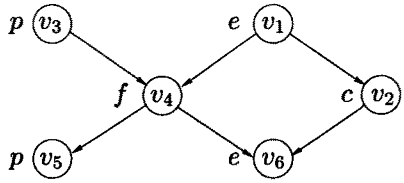


Fig. 64. An acyclic labelled graph.

More precisely, $\mathrm{FS}(M) = \{\phi_N(s_1)\cdots\phi_N(s_n) \mid {}^\circ N[s_1\cdots s_n\rangle N^\circ$ for a process N of $M\}$. Now it only remains to show that these complete firing sequences $s_1\cdots s_n$ 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 $t_1, \ldots, t_n \in T$. Then ${}^\circ N[t_1\cdots t_n\rangle N^\circ$ iff*
(1) all t_i are distinct,
(2) $T = \{t_1, \ldots, t_n\}$, 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 $t_1, \ldots, t_n \in T$. Then ${}^\circ N[t_1\cdots t_n\rangle C$ iff*
(1) all t_i are distinct,
(2) ${}^{\to}C \cap T = \{t_1, \ldots, t_n\}$,
(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 ${}^{\to}C \cap T$ (with respect to F^+) and hence $t_n{}^\bullet \subseteq C$. □

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, $\mathrm{FS}(M) = \{\phi_N(s_1)\cdots\phi_N(s_n) \mid \exists N \in \mathrm{PROC}(M) : {}^\circ N[s_1\cdots s_n\rangle N^\circ\}$. For a process $N = (P, T, F, \phi_1, \phi_2)$ of M, Theorem 112 now implies that ${}^\circ N[s_1\cdots s_n\rangle N^\circ$ 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 $\mathrm{FS}(M) = \{\phi_G(u_1)\cdots\phi_G(u_n) \mid u_1\cdots u_n$ is a topological order of $G \in \mathrm{LPO}(M)\} = \bigcup\{\mathbf{words}(G) \mid G \in \mathrm{LPO}(M)\}$. □

Example 32. Let M be the EN system of Fig. 47 and N the process of M in Fig. 57, with $\mathbf{pru}(\mathbf{ctr}(N))$ shown in Fig. 60. The words of $\mathbf{pru}(\mathbf{ctr}(N))$ that are given in Example 31 (see Fig. 64) 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, $\mathbf{words}(\mathbf{pru}(\mathbf{ctr}(N))) = \{\phi_N(s_1) \cdots \phi_N(s_n) \mid {}^\circ N[s_1 \cdots s_n\rangle N^\circ\}$ and similarly for N'. From this and the uniqueness (modulo $\equiv_\beta^\alpha$) of the process in the statement of Theorem 92, it follows that $N \equiv_\beta^\alpha N'$. Note that, consequently, the function $\mathbf{proc} : \mathrm{LPO}(M) \to \mathrm{PROC}(M)$ mentioned in the proof of Theorem 107 can simply be defined as follows: for $G \in \mathrm{LPO}(M)$, 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 $\mathrm{FS}(M)$ in such a way that partial orders in $\mathrm{LPO}(M)$ 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. 47. 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. 15.

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 FS(M) (without knowing M).

Lemma 118. *Let $M = (P, T, F, C_{in})$ be an EN system. Then* $\mathbf{ind}(M) = \{(s,t) \in T \times T \mid \exists x \in T^* : xst \in FS(M) \text{ and } xts \in FS(M)\}$.

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

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 LPO(M). In the next definition we use a predetermined countable set of nodes $\{v_1, v_2, v_3, \ldots\}$. 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 $t_1, \ldots, t_n \in \Sigma$.

(1) The *dependency graph of x (over I)*, denoted by $\mathbf{dep}_I(x)$, is the labelled graph $(V, \Gamma, \Sigma, \phi)$, where $V = \{v_1, \ldots, v_n\}$, $\phi(v_i) = t_i$ 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 $\mathbf{dep}_{\mathbf{ind}(M)}(x)$.

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. 65 and $\mathbf{pru}(\mathbf{dep}_I(ecpfpe))$ in Fig. 66. Since the graphs in Figs. 64 and 66 are the same, $\mathbf{pru}(\mathbf{dep}_I(ecpfpe))$ can thus also be found in Fig. 64.

We now want to show that, for every contact-free EN system M, LPO$(M) \equiv \{\mathbf{pru}(\mathbf{dep}_M(x)) \mid x \in \text{FS}(M)\}$. 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 $s^\bullet \cap {}^\bullet t \neq \varnothing$, then $(\phi(s), \phi(t)) \in \mathbf{dep}(M)$.

Proof. (1) This follows directly from Theorem 91(3).
(2) If $s^\bullet \cap {}^\bullet t \neq \varnothing$, then also $\phi(s)^\bullet \cap {}^\bullet\phi(t) \neq \varnothing$. □

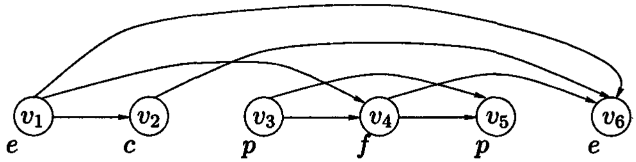


Fig. 65. A dependency graph.

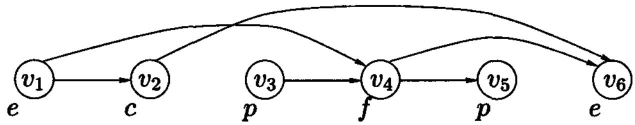


Fig. 66. A pruned dependency graph.

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 ${}^\circ N[s_1 \cdots s_n\rangle N^\circ$ with $T = \{s_1, \ldots, s_n\}$, and let $G = \mathbf{dep}_M(\phi(s_1) \cdots \phi(s_n))$ Then for all $1 \le i, j \le n$: $(s_i, s_j) \in F_N^+$ iff $(v_i, v_j) \in \Gamma_G^+$.*

Proof. (Only-if) If $s_i{}^\bullet \cap {}^\bullet s_j \neq \varnothing$, 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), $s_i\ \mathbf{co}_N\ s_j$ 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 $(s_i, s_j) \in F_N^+$. □

Theorem 122. *Let $N = (P, T, F, \phi_1, \phi_2)$ be a process of a contact-free EN system M and let ${}^\circ N[s_1 \cdots s_n\rangle N^\circ$. 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. □

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 $\equiv_\beta^\alpha$) of this process N, cf. the remark following Theorem 92.

Example 35. Let M be the EN system of Fig. 47 and N the process of M in Fig. 57. Then there is a firing sequence $^\circ N[s_1 \cdots s_n\rangle N^\circ$ with $\phi(s_1)\cdots\phi(s_n) = ecpfpe$ (see Example 28). The graphs $\mathbf{pru}(\mathbf{ctr}(N))$ and $\mathbf{pru}(\mathbf{dep}_M(ecpfpe))$ are drawn in Figs. 60 and 64, 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. □

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 $\beta : \mathbf{use}(T_M) \to \mathbf{use}(T_{M'})$ 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 $\beta(\mathrm{FS}(M)) = \mathrm{FS}(M')$ then, by Lemma 118, $\mathbf{ind}(M') = \{(\beta(s), \beta(t)) \mid (s,t) \in \mathbf{ind}(M)\}$. Then, for $x \in \mathrm{FS}(M)$, $\mathbf{pru}(\mathbf{dep}_M(x)) \equiv_\beta \mathbf{pru}(\mathbf{dep}_{M'}(\beta(x)))$. □

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. □

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 $\beta(\mathrm{FS}(M)) = \mathrm{FS}(M')$. We first construct $\mathrm{SCG}(M)$ and $\mathrm{SCG}(M')$. From these we can obtain $\mathbf{use}(T_M)$ and $\mathbf{use}(T_{M'})$. Now we test all bijections β. The languages $\mathrm{FS}(M)$, $\beta(\mathrm{FS}(M))$, and $\mathrm{FS}(M')$ are regular, and finite automata can easily be constructed for them (see Theorem 12). Since 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 $\beta(\mathrm{FS}(M))$ and $\mathrm{FS}(M')$. □

For an arbitrary (not necessarily contact-free) EN system M, let $\text{PD}(M) = \{\mathbf{pru}(\mathbf{dep}_M(x)) \mid x \in \text{FS}(M)\}$: the set of pruned dependency graphs of M. Then Theorem 123 says that for every contact-free EN system M: $\text{LPO}(M) \equiv_\beta \text{PD}(M)$, where β is the identity on $\mathbf{use}(T_M)$. Now note that $\text{PD}(M)$ 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 $\text{PD}(M) \equiv \text{PD}(M')$. 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 $\text{PD}(M) \equiv \text{PD}(M') \equiv \text{LPO}(M')$; in other words $\text{PD}(M)$ is isomorphic with $\text{LPO}(M')$. Hence we can view $\text{PD}(M)$ 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 $\text{PD}(M) \equiv \text{PD}(M')$. The above results can then easily be extended to arbitrary EN systems (e.g., Lemma 124 would then be the generalization of Theorem 125).

6.2 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 ${}^\circ N[y\rangle N^\circ$, $\phi_N(y) = x$, ${}^\circ N[y'\rangle N^\circ$, and $\phi_N(y') = x'$.

Note that the process N is unique modulo $\equiv_\beta^\alpha$ (cf. the remarks following Theorems 92 and 122). Note also that, by Theorem 112, $x \approx_{lpo} x'$ iff $x, x' \in \mathbf{words}(G)$ 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 $x, x' \in FS(M)$. 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 ${}^\circ N[y\rangle N^\circ$, $\phi_N(y) = x$, ${}^\circ N[y'\rangle N^\circ$, 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, $^\circ N[y\rangle N^\circ$. Since $\mathbf{pru}(\mathbf{ctr}(N)) \equiv_\beta \mathbf{pru}(\mathbf{dep}_M(x'))$, analogously there exists y' with $\phi_N(y') = x'$ and $^\circ N[y'\rangle N^\circ$. □

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 $\doteq_I \subseteq \Sigma^* \times \Sigma^*$ is defined as follows: for $x, y \in \Sigma^*$,
$x \doteq_I y$ iff there exist $a, b \in \Sigma$ and $x_1, x_2 \in \Sigma^*$, such that
$x = x_1abx_2$, $y = x_1bax_2$, and $(a, b) \in I$.
The relation $\approx_I \subseteq \Sigma^* \times \Sigma^*$ is then defined as the smallest equivalence relation that contains $\doteq_I$. 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 $x_0, \ldots, x_n \in \Sigma^*$ such that $x_0 = x$, $x_n = y$, and $x_{i-1} \doteq_I x_i$ 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 $xsty \in \mathrm{FS}(M)$ and $(s, t) \in \mathbf{ind}(M)$, then $xtsy \in \mathrm{FS}(M)$. The proof of this property is easy: If $(s, t) \in \mathbf{ind}(M)$ then $\mathbf{disj}(\{s, t\})$. Let $C_{in}[x\rangle C[st\rangle D$. Then st **con** C and $\mathbf{disj}(\{s, t\})$. This implies that $\{s, t\}$ **con** C and thus (by Lemma 17) that $C[ts\rangle D$. Hence $xtsy \in \mathrm{FS}(M)$. In other words, transitions in $\mathbf{ind}(M)$ are interchangeable in a firing sequence, because they actually occur concurrently. Thus, $\mathrm{FS}(M)$ is closed under $\approx_{\mathbf{ind}(M)}$: if $x \in \mathrm{FS}(M)$ and $x \approx_{\mathbf{ind}(M)} y$, then $y \in \mathrm{FS}(M)$. A similar argument shows that, for $x, y \in \mathrm{FS}(M)$, $x \approx_{\mathbf{ind}(M)} y$ 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 $[pefc]_I = \{pefc, pecf, epfc, epcf, ecpf\}$.

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 $t_k, \ldots, t_{m-1}$ succeed u_k in the topological order $u_1 \cdots u_n$. Hence there are no edges $(t_k, t_m), \ldots, (t_{m-1}, t_m)$ in Γ (and of course no edges in the other direction because $t_1 \cdots t_n$ is a topological order). Thus $(t_k, t_m), \ldots, (t_{m-1}, t_m) \in J$. 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 $k + 1$. □

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 $y = \phi(w_1) \cdots \phi(w_n)$. □

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) $\mathbf{dep}_I(x) \equiv_\beta \mathbf{dep}_I(y)$, *and*
(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 $x = x_1abx_2$ and $y = x_1bax_2$, 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 $\delta : \{v_1, \ldots, v_n\} \to \{v_1, \ldots, v_n\}$ 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 $\mathbf{dep}_I(x)$. □

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 $\mathrm{TR}(M) = \{[x]_{\mathbf{ind}(M)} \mid x \in \mathrm{FS}(M)\}$, i.e., the language $\mathrm{FS}(M)$ in which trace equivalent words are grouped together. According to Theorems 123 and 135 the function $\mathbf{prudep} : \mathrm{TR}(M) \to \mathrm{LPO}(M)$, defined by $\mathbf{prudep}([x]) = \mathbf{pru}(\mathbf{dep}_M(x))$, is a bijection between $\mathrm{TR}(M)$ and $\mathrm{LPO}(M)$, modulo isomorphism (and note that, by Theorem 134, **words** is its inverse). Thus, $\mathrm{TR}(M)$ 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 $\mathrm{TR}(M)$ and $\mathrm{PROC}(M)$, modulo isomorphism. We can define two EN systems M and M' to be *trace equivalent* if there exists a bijection $\beta : \mathbf{use}(T_M) \to \mathbf{use}(T_{M'})$ such that $\beta(\mathrm{TR}(M)) = \mathrm{TR}(M')$, where $\beta(\mathrm{TR}(M))$ is defined in the obvious way. Then, clearly, trace equivalence is the same as lpo-equivalence.

7 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 ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \varnothing$ 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 ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \varnothing$, then $\{x \in X_N \mid t_1 \ F^* \ x\} \cap \{x \in X_N \mid t_2 \ F^* \ x\} = \varnothing$. 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. 22, 23, 24, and 25 are b-process nets. This shows that confusion can be present in b-process nets. The acyclic EN system in Fig. 30 is not a b-process net because $\#({}^\bullet p_3) = 2$. The EN system of Fig. 34 satisfies (1) and (2) above but it is not a b-process net, because $t_3 \otimes t_3$. In fact, ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \varnothing$ 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. 23. 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 $C \subseteq P$. $C \in \mathbb{C}_N$ 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 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. 51. A branching process N of M is drawn in Fig. 67. Note that the process of M that is given in Fig. 56 is "part" of N, i.e., it is one of the runs of M that is combined in the branching run corresponding to N.

(2) A b-process of the EN system of Fig. 2 (mutual exclusion) is drawn in Fig. 68. 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. 58, corresponding to one run during which components 1,2,1, and 1 get permission, respectively.

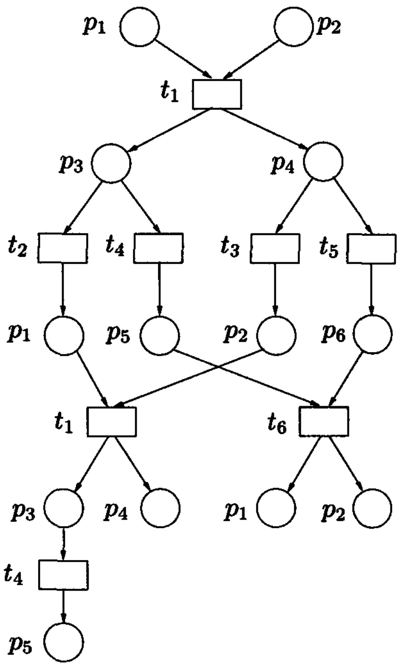


Fig. 67. A branching process of the EN system of Fig. 51.

It can be proved that Lemma 89 and Theorems 90 and 91 are also true for b-processes (again with $\mathbf{co}_N \cap \overline{\otimes}_N$ 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 5.4). 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.

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 $(A, \rho, \otimes)$ where
(1) (A, ρ) is a partially ordered set,

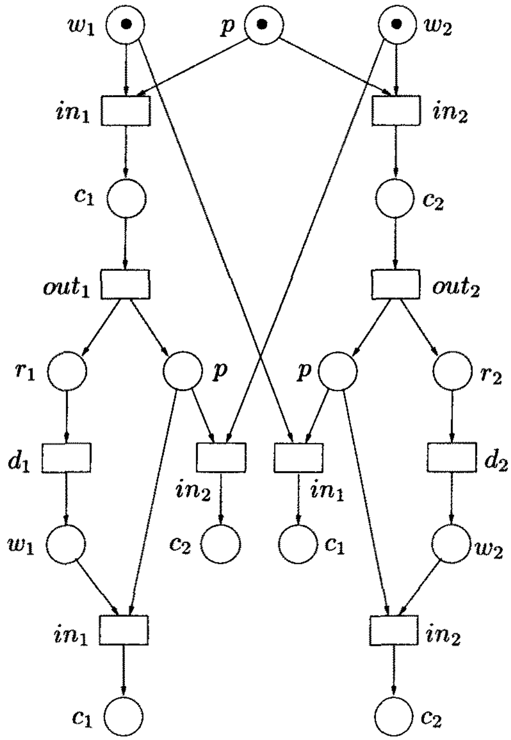


Fig. 68. A b-process of the mutual exclusion system of Fig. 2.

(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 69 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

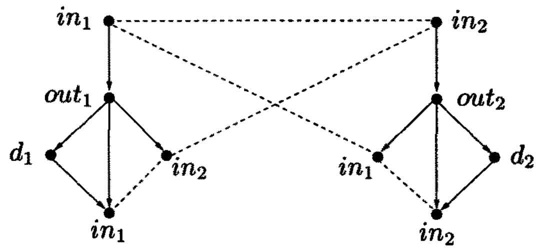


Fig. 69. A labelled b-graph.

by the transitions of the mutual exclusion system of Fig. 2. 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.

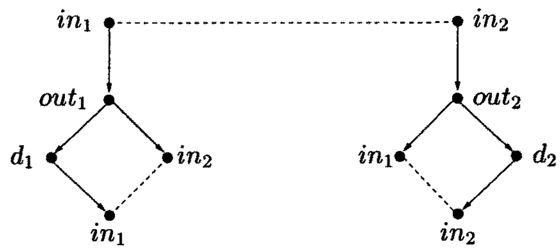


Fig. 70. The pruned version of the labelled b-graph of Fig. 69.

Example 42. Figure 70 shows the pruned version $\mathbf{pru}(G)$ of the labelled b-graph G of Fig. 69.

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 $s^\bullet \cap {}^\bullet t \neq \varnothing$, and $(s,t) \in \otimes$ iff ${}^\bullet s \cap {}^\bullet t \neq \varnothing$.

(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. 68. Its contracted version $\mathbf{ctr}(N)$ and pruned contracted version $\mathbf{pru}(\mathbf{ctr}(N))$ are shown in Figs. 69 and 70, respectively. They can be compared with the contracted and pruned contracted process of Fig. 62.

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 LES(M) the set of all pruned contracted b-processes of M (where LES stands for Labelled Event Structures). Hence

$$\mathrm{LES}(M) = \{\mathbf{pru}(\mathbf{ctr}(N)) \mid N \in \mathrm{BPROC}(M)\}.$$

Definition 147. Two contact-free EN systems M and M' are *les-equivalent* if $\mathrm{LES}(M) \equiv \mathrm{LES}(M')$.

Obviously, for every contact-free EN system M, $\mathrm{LPO}(M) = \{G \in \mathrm{LES}(M) \mid \otimes_G = \varnothing\}$. 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 $(R, \Gamma \cap (R \times R), \Sigma, \phi \restriction R)$.

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, LES(M) can be recovered from 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, $G \in LES(M)$ iff
(1) $\mathbf{pru}(G) = G$,
(2) $G[R] \in LPO(M)$ 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 ${}^\bullet s_1 \cap {}^\bullet s_2 \neq \varnothing$ by Definition 146. This implies that ${}^\bullet\phi(s_1) \cap {}^\bullet\phi(s_2) \neq \varnothing$, 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 ${}^\bullet s_1' \cap {}^\bullet s_2' \neq \varnothing$, $s_i'\ F^*\ s_i$ 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 $G \in$ LPO(M)) in the proof of the If-part of Theorem 107, at the end of Section 5. 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 $\phi \restriction {}^\bullet s$ 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 $G[R_s] \in \mathrm{LPO}(M)$. 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

$$\text{for } s, s' \in T, \quad s' \,F^+\, s \text{ iff } s' \,\Gamma^+\, s. \tag{1}$$

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

$$\text{for } s_1, s_2 \in T, \quad s_1 \otimes_N s_2 \text{ iff } (s_1, s_2) \in \mathbf{tra}(\otimes), \tag{2}$$

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 ${}^\bullet s_1 \cap {}^\bullet s_2 \neq \varnothing$. 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 $\mathrm{LPO}(M)$. Again, this implies that also ${}^\bullet s_1 \cap {}^\bullet s_2 \neq \varnothing$ 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 $\phi(s_1) \neq \phi(s_2)$. □

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 $LPO(M) \equiv_\beta LPO(M')$, then $LES(M) \equiv_\beta LES(M')$.

Proof. By Theorem 115, $\beta(\mathrm{FS}(M)) = \mathrm{FS}(M')$. 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 $\mathrm{LES}(M) \equiv_\beta \mathrm{LES}(M')$. □

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 BPROC(M) and 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 6.2. Possibly, this could be used to give an alternative proof of Theorem 151 that is similar to the one of Theorem 126 in Section 6.1.

8 Conclusion

In this chapter we have presented a comprehensive introduction to the theory of Elementary Net systems. We have discussed here both the structural and the behavioural aspects of the theory.

However, the behavioural aspects are prevalent in developing the theory of EN systems, because even such structural issues like reduction and decomposition make sense only modulo some behavioural equivalence of the systems. For example, if a normal form for EN systems is proved, then it is proved with respect to a specific equivalence. Thus typically we say: "For every EN system there exists an *equivalent* EN system that satisfies the conditions of the normal form". In this sense, the study of (behavioural) equivalences is central to the theory of EN systems.

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.

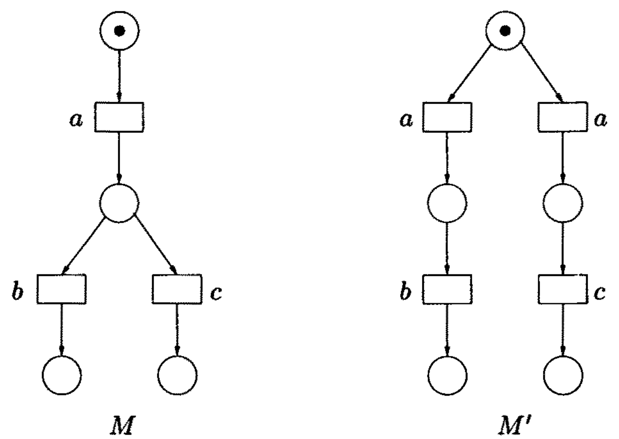


Fig. 71. Two labelled EN systems M and M' that are lpo-equivalent but not weakly configuration equivalent.

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)

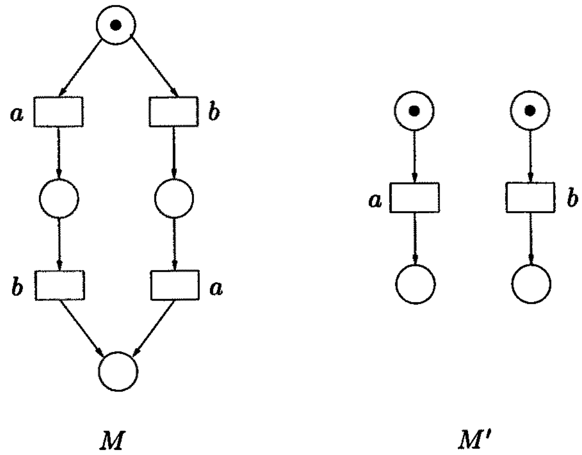


Fig. 72. Two labelled EN systems M and M' that are configuration equivalent but not lpo-equivalent.

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).

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. 71 and 72 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 4 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 which, in its turn, implies lpo-equivalence in the new sense.

Acknowledgments

We wish to thank IJsbrand Jan Aalbersberg, Iko Keesmaat, George Leih, 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 grateful to Maurice ter Beek, Jurriaan Hage, and Nikè van Vugt for their assistance with the preparation of this paper.

References

[AalRoz88] IJ.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 → (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

[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

[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

[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]