Structure Theory of Petri Nets: the Free Choice Hiatus

Eike Best

Institut für methodische Grundlagen
Gesellschaft für Mathematik und Datenverarbeitung
D-5205 St.Augustin

ABSTRACT

Structure theory asks whether a relationship can be found between the behaviour of a marked net and the structure of the underlying unmarked net. From the rich body of structure theoretical results that exists in Petri net theory, this paper selects a few examples which are deemed to be typical. The class of free choice nets, whose structure theory is particularly agreeable, is studied in some detail.

1 Introduction

By the 'structure' of a P/T-system we mean marking-independent properties depending on the way in which the places and the transitions of the underlying net are interconnected by the flow relation. By the 'behaviour' of a P/T-system we denote marking-dependent properties relating to the token flow effected by the transition rule, depending on the set of processes, the set of reachable markings, the reachability graph, and so on.

The behaviour of a marked net is, in general, less easily analysable than its structure. But it is the behavioural properties that are of foremost interest in the analysis of systems. They include, for example, the property of deadlock-freeness, the existence of invariant assertions, safeness properties, the validity of intermediate assertions, and others.

Structure theory asks whether a relationship can be found between the behaviour of a marked net and the structure of the underlying unmarked net. It asks questions such as: Can one deduce, from certain 'nice' structural properties of a net, that its behaviour will also be 'nice'? Or, conversely: Does certain 'bad' behaviour preordain some 'bad' structure? In any case one may hope that the (behavioural) properties which are of interest may be reduced to easier-to-investigate (structural) properties.

A rich body of structure theoretical results exists in net theory. From this body, we shall select some typical examples, neither too many in order not to let the paper grow out of size, nor too few let the reader get an idea of the kind of reasoning employed in structure theory (hopefully).

There is a class of nets which has an interesting motivation and allows for a very satisfactory structure theory. This class is called free choice nets. While being a non-trivial class of nets, their theory is so nice that it has sometimes jokingly been said that every conjecture is true for free choice nets and false for other nets. Although we will exhibit some 'counterexamples' to this statement, a good part of these notes will be dedicated to the study of free choice nets.

These notes are organised as follows. In section 2 we introduce and explain almost all notions we need, but we will rely on [39] for some definitions and explanations. We introduce some basic behavioural properties (liveness and safeness), and we show that they have an impact in terms of the connectedness of a system. Sometimes it is necessary to compare nets with each other and to state that one is 'similar' to another one. In section 3 we define a notion of simulation to capture this idea. In sections 4-6 we introduce various subclasses of nets (free choice nets, amongst others) and we investigate some basic properties of these classes. In sections 7 and 8 we deal almost exclusively with free choice nets, listing and explaining some more advanced results about their structure and behaviour.

Most of the results reported in this paper are drawn from published literature, and appropriate references will always be given. However, some proofs are not easily accessible and some belong to the 'folklore'. Because of the size of the material we have done a selection of the proofs. We give in detail only those proofs that cannot be retrieved easily from the literature. All other proofs will be given by outline only, or will even be omitted, and a reference will be supplied instead.

2 Basic definitions and general results

Our object of study are P/T-systems in the sense of [39]. However, we will restrict ourselves to P/T-systems without capacity constraints (that is, we will assume all capacities to be infinite) and with a trivial weight function (i.e. the weight equals 1 on every F-arrow). For the sake of simplicity and brevity, we will change the sixtuple notation of [39](1.5) into a fourtuple notation; thus, $\Sigma = (S,T;F,M_0)$ will henceforth denote a P/T-system (with infinite capacities and trivial weight function). As usual, S,T,F and M_0 are the set of places, the set of transitions, the flow relation and the initial marking, respectively. We mention that the relation $F \subseteq (S \times T) \cup (T \times S)$ could equivalently be viewed as a function

$$F\colon\ (S \times T) \cup (T \times S) \ \rightarrow\ \{0,1\}$$

such that $(x,y) \notin F$ or $(x,y) \in F$ (in the relational view) iff $F(x,y) = 0$ or $F(x,y) = 1$, respectively (in the functional view). We will sometimes make use of the functional view in order to shorten formulae.

Without repeating their definitions, we shall use the following concepts: the transition rule [39](1.7a-d), the set of occurrence sequences of Σ [39](1.11a,b), the set of transition sequences of Σ [39](1.11c), the set $[M_0\rangle$ of forward reachable markings of Σ [39](1.7e), the coverability graph (sometimes also called the reachability graph) of Σ [39](2.4), and the notion of a side condition being a place $s \in S$ with ${}^\bullet s \cap s^\bullet \neq \emptyset$ (called a 'loop' in [39](1.17)). As usual, ${}^\bullet x$ denotes the set of F-predecessors of $x \in X$ and $x^\bullet$ denotes the set of F-successors of $x \in X$. The reader may also consult [6] for the various formal definitions.

We will now introduce two restrictions, the purpose of which is to focus our scope of concern on such P/T-systems as are of primary interest.

Restriction 2.1 *Finiteness of* Σ

From now on, we will always assume Σ to be finite, that is, $S \cup T$ to be a finite set. ∎ 2.1

The reason for restricting ourselves to finite systems is simply a pragmatic one: they are the main cases of practical interest. Besides, a theory of infinite systems exists only in rudimentary form.

A (finite) system Σ may consist of two or more parts which are unconnected with each other in the sense that no (undirected) F-path leads from one part to the other. To all intents and purposes, it is then sufficient to study the two (or more) parts in isolation. The next definition and the restriction following it are intended to capture this property.

Definition 2.2 *Weak connectedness*

$\Sigma = (S,T;F,M_0)$ is weakly connected *iff* all $x,y \in S \cup T$ are in the relation $(F \cup F^{-1})^*$. ∎ 2.2

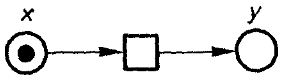


Figure 1: A system which is weakly connected but not strongly connected

Restriction 2.3

From now on, we will always assume Σ to be weakly connected. ■ 2.3

Weak connectedness means that one can always travel from x to y along some F-arcs, be it in forward or in backward direction. It does not mean that there is always a *directed* path from x to y. Consider the simple system shown in Figure 1. It is weakly connected and there is a directed F-path from x to y but not from y to x. The existence of directed paths is captured by the next definition.

Definition 2.4 *Strong connectedness*

$\Sigma = (S,T;F,M_0)$ is strongly connected *iff* all $x,y \in S \cup T$ are in the relation F^*. ■ 2.4

Weak connectedness is a much weaker property that strong connectedness. We will not require the latter universally because there are interesting non-strongly connected nets and because it is not easily possible to split such nets into strongly connected components without disrupting their behaviour.

The terms 'weak connectedness' and 'strong connectedness' are generally agreed upon in graph theory. The above definitions are applications of this general terminology. We will sometimes use the definitions in a more general sense, for example applying to the reachability graph which is (by definition) always weakly connected.

We will often be interested in the set $[M_0\rangle$ of forward reachable markings and its properties. After all, this set models the set of states the P/T-system Σ may be in. We assume the usual definition of $[M_0\rangle$ (as given in [6], for instance) which works, essentially, using occurrence sequences.

It is well known that in order to represent concurrency directly, one should replace occurrence sequences by processes [35,18,5]. It is possible to define the set $[M_0\rangle$ using processes instead of occurrence sequences. However, it is also known [5] that for finite P/T-systems, the two definitions coincide, so that the definition of $[M_0\rangle$ using occurrence sequences is sufficient.

On the other hand, we will then have to investigate concurrency indirectly via the notion of two transitions being concurrently enabled by some marking. We repeat this definition from [6], in a simplified form implied by the fact that all capacities are infinite.

Definition 2.5 *Concurrent enabling*

For $\Sigma = (S,T;F,M_0)$, let $M \in [M_0\rangle$ be a marking and $t_1,t_2 \in T$ two transitions.
t_1 and t_2 are concurrently enabled by M *iff* $\forall s \in S\colon F(s,t_1) + F(s,t_2) \leq M(s)$.
(Here F is viewed as a function to $\{0,1\}$, as explained above.) ■ 2.5

For concurrently enabled transitions, the following simple fact is true:

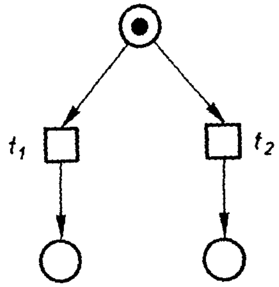


Figure 2: A conflict

Fact 2.6 *Exchanging concurrently enabled transitions in an occurrence sequence*

If t_1 and t_2 are concurrently enabled by M then both $Mt_1M't_2M''$ and $Mt_2\hat{M}t_1M''$ are occurrence sequences starting with M and ending with M'' (but not necessarily $M' = \hat{M}$).

Proof: From the transition rule and from $M(s) \geq F(s,t_1) + F(s,t_2)$ it follows immediately that $M'(s,t_2) \geq F(s,t_2)$ and $\hat{M}(s,t_1) \geq F(s,t_1)$. ■ 2.6

If M enables both t_1 and t_2 then it is by no means necessarily true that t_1 and t_2 are concurrently enabled by M; failing the latter, the situation is called a 'conflict' (see Figure 2).

Definition 2.7 *Conflict at a marking*

For $\Sigma = (S,T;F,M_0)$, let $M \in [M_0\rangle$ be a marking and t_1, t_2 two transitions.
Then t_1 and t_2 are in conflict at M *iff* M enables both t_1 and t_2 but does not concurrently enable t_1 and t_2. ■ 2.7

Fact 2.8 *Characterisation of conflict*

t_1 and t_2 are in conflict at M iff both are enabled at M and $M(p) = 1$ for some $p \in {}^\bullet t_1 \cap {}^\bullet t_2$.

Proof: Easy from the definitions.
This result depends on the facts that all capacities are infinite and that all arc weights equal 1; otherwise it becomes false. ■ 2.8

If t_1 and t_2 do not share an input place then their combined enabling implies their concurrent enabling, as shown by the next simple fact.

Fact 2.9 *Sufficient conditions for concurrent enabling*

(a) *If ${}^\bullet t_1 \cap {}^\bullet t_2 = \emptyset$ and M enables both t_1, t_2 then M concurrently enables t_1 and t_2.*

(b) *If $t_1^\bullet \cap {}^\bullet t_2 = \emptyset$ and $Mt_1M't_2M''$ is an occurrence sequence then M concurrently enables t_1 and t_2.*

Proof: (a) M enables both t_1 and t_2 implies $M(s) \geq F(s,t_1)$ and $M(s) \geq F(s,t_2)$ (for all $s \in S$). But

$$\left.\begin{array}{lcl} F(s,t_1) \neq 0 & \Rightarrow & F(s,t_2) = 0 \\ F(s,t_2) \neq 0 & \Rightarrow & F(s,t_1) = 0 \end{array}\right\} \text{ since } {}^\bullet t_1 \cap {}^\bullet t_2 = \emptyset.$$

Hence $M(s) \geq F(s,t_1) + F(s,t_2)$.

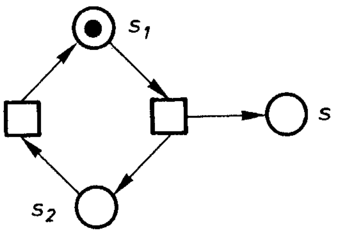


Figure 3: An unsafe system

(b) We have $\forall s \in S: M'(s) \geq F(s,t_2)$ and $M'(s) = M(s) - F(s,t_1) + F(t_1,s)$.
Hence $M(s) \geq F(s,t_2) - F(t_1,s) + F(s,t_1)$. But

$$\left.\begin{array}{lcl} F(s,t_2) \neq 0 & \Rightarrow & F(t_1,s) = 0 \\ F(t_1,s) \neq 0 & \Rightarrow & F(s,t_2) = 0 \end{array}\right\} \text{since } t_1^\bullet \cap {}^\bullet t_2 = \emptyset.$$

Hence $M(s) \geq F(s,t_2) + F(s,t_1)$.

■ 2.9

Definition 2.5 and facts 2.6-2.9 provide a means of rearranging occurrence sequences. The typical argument is the following: if σ is an occurrence sequence in which some marking concurrently enables the two next transitions, then the latter can be switched around to give another occurrence sequence which agrees with σ except in the order of the two transitions and the marking between them.

We will now define two simple but important behavioural properties of Σ. One of them (safeness) is concerned with the places of Σ while the other (liveness) is concerned with the transitions of Σ. On a given place s of S (in Σ), more than one token may assemble in the course of a behaviour of Σ; indeed, s may have (finitely many but) more than one token already in the initial marking. There may not even be a bound on how many tokens may assemble on s. For example, in the system shown in Figure 3, for any given natural number $n \in \mathbf{N}$, s may receive more than n tokens; however, s_1 and s_2 may always carry at most one token each. Safeness introduced in the next definition, measures the amount of tokens that can assemble on a given place; in the literature, safeness is often also called 'boundedness'.

Definition 2.10 *Safeness*

Let $\Sigma = (S,T;F,M_0)$ be a place/transition system.

(a) $s \in S$ is n-safe ($n \in \mathbf{N}$) *iff* $\forall M \in [M_0\rangle: M(s) \leq n$.

(b) Σ is n-safe ($n \in \mathbf{N}$) *iff* $\forall s \in S: s$ is n-safe.

■ 2.10

The case that $n = 1$, that is 1-safeness, plays a particular rôle. Every place can then contain at most one token. A 1-safe place can be interpreted as a condition which either 'holds' (if $M(s) = 1$) or 'does not hold' (if $M(s) = 0$). 1-safeness is a very important property which holds in many practical situations (e.g.: a variable always has exactly one value; in a sequential program, control is always at exactly one location). We will focus our attention on 1-safe systems (and we will sometimes write 'safe' instead of '1-safe', but only when the '1' is unimportant, i.e. could be replaced by 'n').

A given transition t of T (in Σ) may be repeated infinitely often, or there may be bounds on the number of times it can be repeated (including zero, in which case it is called 'dead'). It may even be the case that a system has the choice of entering a state in which t is 'dead' or entering a state in which t can be repeated over and over again, never becoming 'dead'. Liveness measures whether or not it is possible to 'kill' a transition.

Definition 2.11 *Liveness*

Let $\Sigma = (S, T; F, M_0)$ be a place/transition system.

(a) $t \in T$ is live *iff* for all $M \in [M_0\rangle$ there is some $M' \in [M\rangle$ such that M' enables t.

(b) Σ is live *iff* $\forall t \in T{:}\, t$ is live.

■ 2.11

Notation 2.12 *LS systems*

If Σ is live and 1-safe then we shall call Σ an LS system, for short. ■ 2.12

The next result relates three interesting notions introduced so far, i.e. safeness, liveness and strong connectedness, to each other. We intend to show that a net which is not strongly connected cannot be live and 1-safe at the same time, i.e. that the (structural) property of strong connectedness is necessary for the existence of a live and 1-safe marking. The previous examples show that a system could be either live or 1-safe and non-strongly connected: The systems shown in Figures 1 and 2 are not strongly connected but 1-safe (but they are not live), while the system shown in Figure 3 is not strongly connected but live (but it is not safe).

Theorem 2.13 *Liveness and safeness implies strong connectedness*

Let $\Sigma = (S, T; F, M_0)$ be a (finite, weakly connected) P/T-system which is live and 1-safe. Then Σ is strongly connected.

Proof: For any arbitrarily chosen $x, y \in S \cup T$ we have to prove that $(x, y) \in F^*$, i.e. that there is a directed F-chain from x to y. Weak connectedness implies only that there is a $(F \cup F^{-1})$-chain from x to y, i.e. there is a sequence

$$x_0, \ldots, x_m \; (m \geq 0, x_i \in S \cup T)$$

such that $x = x_0$, $x_m = y$ and $(x_i, x_{i+1}) \in (F \cup F^{-1})$ for $0 \leq i < m$. Let us now try to construct an F-chain from x to y. We start with $x = x_0$. If $(x_0, x_1) \in F$ then we may pass on to x_1; if $(x_1, x_2) \in F$, we may then pass on to x_2, and so on. The bad case is that $(x_i, x_{i+1}) \in F^{-1}$ (rather than $\in F$) for some $0 \leq i < m$. But if we then have $(x_i, x_{i+1}) \in F^*$, we may still pass from x_i to x_{i+1} along an F-chain. So the really bad case is that $(x_i, x_{i+1}) \in F^{-1}$, but no F-chain leads from x_i to x_{i+1}.
So, we assume now that we have $(x_i, x_{i+1}) \in F^{-1}$, but $(x_i, x_{i+1}) \notin F^*$; the proof is done if we can derive a contradiction from this assumption. Because we will be concerned exclusively with deriving this contradiction, we may drop the index i from now on and re-use the letters x and y.
There are two cases:

Case 1: $(x, y) \in F \cap (S \times T)$ and $(y, x) \notin F^*$ (see Figure 4(i));

Case 2: $(y, x) \in F \cap (T \times S)$ and $(x, y) \notin F^*$ (see Figure 4(ii)).

If we succeed in obtaining a contradiction in both cases then the theorem is proved.

Let us consider Case 1 first. That is, we assume $(x, y) \in F \cap (S \times T)$, but no directed F-path leads from y to x. We consider the set of places and the set of transitions from which a directed path leads to x, calling them S_1 and T_1, respectively:

$$S_1 = \{s \in S \mid (s, x) \in F^*\}, \quad T_1 = \{t \in T \mid (t, x) \in F^*\}.$$

and no directed path from y to x

Case 1

and no directed path from x to y

Case 2

Figure 4: Illustrating the case distinction in the proof

$$M_0[\ldots\rangle\ \underbrace{M_1}_{\text{enables } y}\ [y\rangle\ \underbrace{M_2}_{<M_1 \text{ on } S_1}\ [\ \ \ldots\tau\ldots\ \ \rangle\ \underbrace{M_3}_{\text{enables } y}$$

Figure 5: Illustrating Case 1: M_1, M_2, M_3

By this definition we have:

$$\begin{array}{l} x \in S_1 \\ y \notin T_1 \ \ (\text{since } (y,x) \notin F^*) \\ y^\bullet \cap S_1 = \emptyset \ \text{ and} \\ {}^\bullet T_1 \subseteq S_1. \end{array}$$

By the liveness of Σ, a reachable marking M_1 can be found which enables y, i.e.:

$$\exists M_1 \in [M_0\rangle : M_1 \text{ enables } y.$$

Let us fix such a marking M_1 and let us consider the successor marking M_2 under y, i.e. $M_1[y\rangle M_2$.
Because $y^\bullet \cap S_1 = \emptyset$, the token load on S_1 cannot be increased by the occurrence of y, that is:

$$\forall s \in S_1 : M_1(s) \geq M_2(s).$$

Furthermore, we have

$$1 \ = \ M_1(x) \ > \ M_2(x) \ = \ 0,$$

that is, there is a token on x under M_1 but not under M_2; abbreviating this, we may say that M_1 is 'strictly bigger' than M_2 on S_1.
Again by the liveness of Σ, there is a successor marking M_3 of M_2 which enables y; let τ be the transition sequence which leads from M_2 to M_3 (see Figure 5).

The sequence τ may contain transitions from T_1 and transitions from $T \backslash T_1$. We claim, however, that it is possible to rearrange the transitions in τ in such a way that all transitions in T_1 come first. To this end, assume that τ has the following form:

$$\tau \ = \ \ldots tt' \ldots$$

with $t \in T \backslash T_1$, $t' \in T_1$. We plan to show that $t^\bullet \cap {}^\bullet t' = \emptyset$; then facts 2.9(b) and 2.6 can be applied to show that t and t' can be exchanged in τ. But suppose that $s \in t^\bullet \cap {}^\bullet t'$; Then there is a directed path (of length 2) from t to t', and hence also from t' to x, contradicting the assumption that $t' \in T \backslash T_1$.
Hence $t^\bullet \cap {}^\bullet t' = \emptyset$ and t and t' may be exchanged in τ.
By repeating such exchanges exhaustively, τ can be rearranged into a transition sequence

$$M_0[\ldots\rangle \underbrace{M_1}_{\text{enables } y} [y\rangle \underbrace{M_2}_{\text{no token on } x} [\tau_1\rangle \underbrace{M'}_{\text{token on } x} [\tau_2\rangle \underbrace{M_3}_{\text{enables } y}$$

Figure 6: Illustrating Case 1: τ_1 and τ_2

$$M_0[\ldots\rangle \underbrace{M_1}_{\text{enables } y} [\tau_1\rangle \underbrace{M''}_{\text{2 tokens on } x!}$$

Figure 7: Illustrating Case 1: M''

τ' which also transforms M_2 into M_3 and which can be split as $\tau' = \tau_1\tau_2$ where τ_1 contains only transitions from T_1 while τ_2 contains only transitions from $T \backslash T_1$ (see Figure 6; it could happen that τ_2 is empty).

Now we consider the intermediate marking M' reached from M_2 after τ_1; is there a token on x in this marking or not? There is surely a token on x in M_3 because M_3 enables y; this token can have come there only through a transition in ${}^\bullet x$, i.e. in T_1, but since between M' and M_3 no such transition occurs, the token must have been on x already in M'. Hence $M'(x) = 1$.
On the other hand, the transitions in τ_1 (between M_2 and M') need only tokens from S_1, since ${}^\bullet T_1 \subseteq S_1$. But because M_1 is bigger than M_2 on S_1, this implies that τ_1 is also a transition sequence from M_1 rather than M_2; let M'' denote the marking reached from M_1 after τ_1 (see Figure 7).

Let us now count the number of tokens on x in the marking M''. In M_1, there is one token on x because M_1 enables y. But this token is not needed in the course of τ_1 because τ_1 is enabled in M_2 and we have $M_2(x) = 0$; hence we may consider it to remain unmoved on x during the sequence $M_1[\tau_1\rangle M''$. On the other hand, we have just seen that $M'(x) = 1$, hence τ_1 creates another token on x. Together, we have 2 tokens on x in M''. This contradicts 1-safeness. Hence the assumptions made in Case 1 are wrong, and we have $(x,y) \in F \cap (S \times T) \Rightarrow (y,x) \in F^*$.
It remains to consider Case 2. That is, we assume $(y,x) \in F \cap (T \times S)$ and $(x,y) \notin F^*$ (as in Figure 4(ii)).
It is tempting to think that this case can be reduced to Case 1, but this is not easy. However, the reasoning is quite similar and we therefore give a shortened account[1]. We define:

$$S_2 = \{s \in S \mid (x,s) \in F^*\} \text{ and } T_1 = \{t \in T \mid (t,y) \in F^*\}.$$

S_2 is the set of places to which a directed path leads from x; T_1 is the set of transitions from which a directed path leads to y.
By liveness, we find $M_1 \in [M_0\rangle$ such that M_1 enables y; define M_2 such that $M_1[y\rangle M_2$.
We have $M_2(s) \geq M_1(s)$ for all $s \in S_2$ and $1 = M_2(x) > M_1(x) = 0$; that is, M_2 is strictly bigger than M_1 on S_2.
By liveness, again, there is a transition sequence τ transforming M_2 into M_3 such that M_3 enables y (see Figure 8).

By an argument which is similar to that used above, τ can be rearranged to a sequence $\tau_1\tau_2$

[1] Recently, Wolfgang Reisig has produced a modification of the proof in which the two cases are treated more analogously.

$$M_0[\ldots\rangle M_1[y\rangle M_2\underbrace{[\quad\tau\quad\rangle}_{\tau_1 M'\tau_2}\underbrace{M_3}_{\text{enables } y}$$

Figure 8: Illustrating Case 2: τ_1, τ_2 and M'

such that τ_1 contains only transitions from T_1 and τ_2 contains only transitions from $T\backslash T_1$ (see Figure 8).
Let M' be the marking reached from M_2 by τ_1. In M', y must be enabled because otherwise, y could not be enabled in M_3. On the other hand, x carries a token in M' because the transitions in T_1 cannot take away that token. This contradicts 1-safeness, showing that Case 2 cannot arise and completing the proof. ∎ 2.13

By iterating the argument in the proof, it is easy to see that the premise of 1-safeness in theorem 2.13 can be weakened to n-safeness (for any n).

3 A notion of simulation

The result 2.13 gives a necessary condition for liveness and safeness. It is general in the sense that it applies to all (finite, weakly connected) systems. Such results are rare. In particular, non-trivial sufficient conditions for liveness and safeness are not known. Most known results are more specialised. One may ask, for instance: what happens if we don't allow conflict? Or: what happens if there is no concurrency? This means that one focusses on particular classes of nets. In the main body of this paper we will investigate a range of classes of nets.

Quite often when arguing about net classes, one is led to say that a certain class of nets is 'essentially the same' as another (maybe simpler) class. Usually, one can give a construction which translates every net in the first class into a 'similar' net of the second (simpler) class. We will consider some such constructions. To accommodate these constructions, we will define a general concept which captures the idea that a P/T-system 'simulates' another P/T-system. If a system simulates another one then we may say that they are in some sense 'similar'.

In order to be able to state the definition of simulation we need have a preparatory look at functions on strings which are induced by functions on letters. Let $f\colon A \to A'$ be an injective function from an alphabet A into an alphabet A'. Then f may be extended to a function $f\colon A^* \to A'^*$ in the canonical way[2], i.e. $f(\epsilon_A) = \epsilon_{A'}$, $f(va) = f(v)f(a)$ $(v \in A^*, a \in A)$. Furthermore, f^{-1} is a relation in $A' \times A$ and can be extended to a *function* $f^{-1}\colon A'^* \to A^*$ in the following way:

$$f^{-1}(\epsilon_{A'}) = \epsilon_A,\ f^{-1}(wa) = \begin{cases} f^{-1}(w) & \text{if } a \notin f(A) \\ f^{-1}(w)f^{-1}(a) & \text{if } a \in f(A) \end{cases},w \in A'^*, a \in A'$$

(The injectivity of f is used in the last clause of this definition.) We are now ready to state the definition of simulation; we shall explain the definition after giving it.

Definition 3.1 *Σ' simulates Σ*

Let $\Sigma = (S,T;F,M_0)$ and $\Sigma' = (S',T';F',M_0')$ be two P/T-systems and $f\colon T \to T'$ an injection. We shall say that Σ' simulates Σ (with respect to f) *iff* there is a surjection $\beta\colon [M_0'\rangle \to [M_0\rangle$ such that the following holds:

(i) $M_0 = \beta(M_0')$.

[2] A^* is the set of strings over the alphabet A, including the empty string ϵ_A

(ii) Suppose $M_1 = \beta(M_1')$, $M_1' \in [M_0'\rangle$ and $M_1 \in [M_0\rangle$;

(a) whenever $M_1[t\rangle M_2$ with $t \in T$, $M_2 \in [M_0\rangle$ then $\exists M_2' \in \beta^{-1}(M_2)\ \exists w \in T'^*: M_1'[w\rangle M_2' \wedge f^{-1}(w) = t$;

(b) whenever $M_1'[w\rangle M_2'$ with $w \in T'^*$, $M_2' \in [M_0'\rangle$ then $M_1[f^{-1}(w)\rangle\beta(M_2')$.

(iii) $\forall M \in [M_0\rangle: |\beta^{-1}(M)| < \infty$.

■ 3.1

Because f is an injection, Σ' has at least as many transitions as Σ. The extra transitions of Σ' (i.e. those in $T' \backslash f(T)$) should be thought of as 'silent internal actions' of Σ'. The transitions in $f(T)$ simulate the transitions of Σ. A reachable marking M' of Σ' should be thought of as 'representing' the marking $\beta(M')$ of Σ. There may be more than one marking of Σ' representing the same marking of Σ, but every marking of Σ should be covered; hence the surjection requirement for β. Requirement 3.1(i) means that the initial marking of Σ' must represent the initial marking of Σ. Requirement 3.1(iia) states that any occurrence of the transition t in Σ must be simulatable in Σ' by a transition sequence w which involves $f(t)$ and (possibly) a few intermediate 'silent' occurrences. Moreover, the new marking M_2' of Σ' must represent M_2 (the requirement $M_2' \in \beta^{-1}(M_2)$ is important). 3.1(iib) requires that every occurrence sequence in Σ' corresponds, via f^{-1}, to an occurrence sequence in N which, moreover, respects the representation function β. Requirement 3.1(iii) simply implies that n-safe systems can only be simulated by n-safe systems; it is a somewhat arbitrary requirement that could be dropped if only n-safe systems are under consideration. Some examples are given by Figure 9. In the examples, we indicate the function f by labelling the transitions in $f(T)$ with the names of their counterparts in T, and the transitions in $T' \backslash f(T)$ by a 'τ'; this terminology is borrowed from R.Milner's CCS [31].

We will now show that the simulation relation preserves n-safeness and, essentially, also liveness.

Theorem 3.2 *Simulation preserves n-safeness*

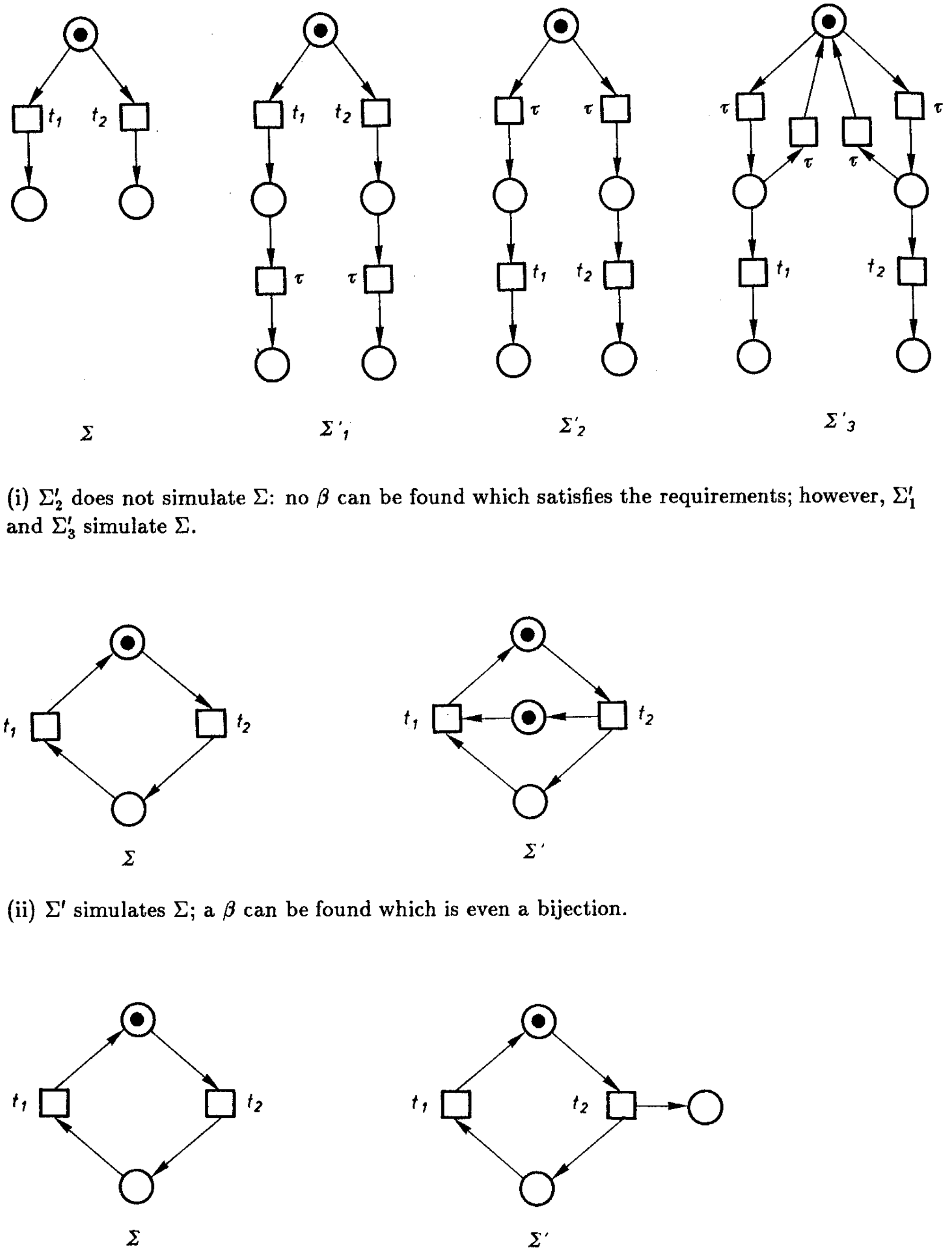
Suppose $\Sigma = (S,T;F,M_0)$, $\Sigma' = (S',T';F',M_0')$, $f: T \rightarrow T'$ *injective and* Σ' *simulates* Σ *with respect to* f.
Then Σ *is n-safe* $\Longleftrightarrow$ Σ' *is n'-safe.* (n need not be the same as n'.)

Proof: Σ is n-safe
$\Rightarrow [M_0\rangle$ is finite
$\Rightarrow \bigcup_{M \in [M_0\rangle} \beta^{-1}(M)$ is finite (with 3.1(iii))
$\Rightarrow |[M_0'\rangle|$ is finite (because $|[M_0'\rangle| \leq \Sigma_{M \in [M_0\rangle} |\beta^{-1}(M)|$).
Conversely, Σ' is n'-safe
$\Rightarrow [M_0'\rangle$ is finite
$\Rightarrow [M_0\rangle$ is finite ($|[M_0\rangle| \leq |[M_0'\rangle|$ because β is surjective). ■ 3.2

Theorem 3.3 *Simulation preserves liveness*

Suppose $\Sigma = (S,T;F,M_0)$, $\Sigma' = (S',T';F',M_0')$, $f: T \rightarrow T'$ *injective and* Σ' *simulates* Σ *with respect to* f.
Then Σ *is live* $\Longleftrightarrow$ *for all* $t' \in f(T)$: t' *is live in* Σ'.

Proof: $\Rightarrow$: Assume that Σ is live.
Let $M_0'[w\rangle M_1'$, $w \in T'^*$ and let $t' \in f(T)$, i.e. $t' = f(t)$ with $t \in T$.
By 3.1(i) and (iib), $M_0[f^{-1}(w)\rangle M_1$, where $M_1 = \beta(M_1')$.
Because Σ is live, $\exists v \in T^*\ \exists M_2 \in [M_0\rangle: M_1[v\rangle M_2$ and t occurs in v.
By 3.1(iia) and because β is a surjection, there are a sequence $v' \in T'^*$ and a marking $M_2' \in [M_0'\rangle$ such that $M_1'[v'\rangle M_2'$ and t' occurs in v'.
Hence all $t' \in f(T)$ are live in Σ'.



(i) Σ_2' does not simulate Σ: no β can be found which satisfies the requirements; however, Σ_1' and Σ_3' simulate Σ.

(ii) Σ' simulates Σ; a β can be found which is even a bijection.

(iii) Σ' does not simulate Σ (only 3.1(iii) is violated).

Figure 9: Illustrating the notion of simulation

$\Leftarrow$: Assume that all $t' \in f(T)$ are live in Σ'.
Let $M_0[v\rangle M_1$, $v = t_1 \dots t_m \in T^*$ and let $t \in T$.
By 3.1(i) and (iia), $\exists w_1, \dots, w_m: M_0'[w_1 \dots w_m\rangle M_1'$ and $M_1 = \beta(M_1')$.
Since all $t' \in f(T)$ are live in Σ', there are a sequence $w' \in T'^*$ and a marking $M_2' \in [M_0'\rangle$ such that $M_1'[w'\rangle M_2'$ and $f(t)$ occurs in w'.
By 3.1(iib), $M_1[f^{-1}(w')\rangle M_2$ for $M_2 = \beta(M_2')$ and t occurs in $f^{-1}(w')$ by the definition of f^{-1}.
Hence Σ is live.

■ 3.3

Theorem 3.3 states that, while Σ' may contain non-live internal transitions, the liveness or otherwise of Σ' on the relevant sets of transitions, viz. on $f(T)$, coincides with the liveness or otherwise of Σ.

Remark 3.4 *Relationship to other literature*

The concept of simulation defined above is much related to the notion of bisimulation introduced by D.Park in [34], and applied to nets by M.Nielsen and P.S.Thiagarajan [33]. Bisimulation does not include the requirement 3.1(iii) and specifies β to be only a relation, rather than a surjective function, on the sets $[M_0\rangle$ and $[M_0'\rangle$. Bisimulation is known to be essentially equivalent to R.Milner's notion of observational equivalence [31], [10]. L.Pomello has shown that the latter is a comparatively strong notion of equivalence [36]. Hence we suggest that our notion of simulation is a comparatively strong one, but we shall not pursue these connections any further in these notes. More recently, K.Voss has investigated a notion of simulation which is similar to the above but is based on step sequences (i.e. sequences of concurrently enabled transitions) rather than transition sequences [42]. A related notion of simulation has also been defined by L.Priese [37]. Readers wishing to compare various definitions are referred to the paper [32] by H.Müller. ■ 3.4

4 T-systems

In the next three sections we introduce, and start to study, three important classes of nets of increasing generality. The first class, called T-nets (or T-systems if a marking is implied), admits concurrency and synchronisation, but no conflict. In the literature, T-systems are often known as 'marked graphs' or 'synchronisation graphs'.

Definition 4.1 *T-nets and T-systems*

$\Sigma = (S, T; F, M_0)$ is a T-system *iff*
its underlying net is a T-net, i.e. for all $s \in S$: $|{}^\bullet s| \le 1 \wedge |s^\bullet| \le 1$. ■ 4.1

In a T-system there is never any conflict, simply because there are no (forward) branched places. A token can be taken away from a place only by its unique (if existing) output transition. T-systems are very well understood. The basic references are [12,17], and [27,19] may be consulted for further reading.

The cycles of a T-system play an important rôle in its analysis.

Definition 4.2 *Cycles and paths*

A cycle of a net $(S,T;F)$ is a sequence $x_0,\ldots,x_m$ with $x_i \in S \cup T$ $(0 \leq i \leq m)$, $(x_i, x_{i+1}) \in F$ $(0 \leq i < m)$ and $x_0 = x_m$. A cycle is called simple *iff* no element (except $x_0 = x_m$) appears twice in it, i.e. $\forall k,j$: $(0 \leq k \leq m \wedge 1 \leq j \leq m-1 \wedge k \neq j) \Rightarrow x_k \neq x_j$.
For later use, we define the (simple) paths of a net exactly as the (simple) cycles, except that the requirement $x_0 = x_m$ is omitted. ∎ 4.2

We shall say that a net $(S,T;F)$ is covered by (simple) cycles *iff* every $x \in S \cup T$ lies on some (simple) cycle. The reader should be cautious with definition 4.2, because it may occur that an element of a cycle has more than one F-predecessors (or successors) on the same cycle, and similarly for paths. In cases of doubt, it may be advisable to include F-arrows that are meant to belong to the cycle or to the path explicitly in the definition.

Theorem 4.3 *Characterisation of the liveness of T-systems*

A T-system $\Sigma = (S,T;F,M_0)$ is live iff all of its simple cycles carry at least one token and for all places $s \in S$: $|{}^\bullet s| = 1$.

Proof: See [12], theorem 1 and [17], theorem (8S); but when looking up these references, the reader should be cautious because in both of them, $|{}^\bullet s| = 1 = |s^\bullet|$ is required in place of $|{}^\bullet s| \leq 1 \geq |s^\bullet|$. ∎ 4.3

Theorem 4.4 *Characterisation of the safeness of live T-systems*

A live T-system $\Sigma = (S,T;F,M_0)$ is 1-safe iff it is covered by simple cycles which carry at most one token.

Proof: See [12], theorem 2 and [17], theorem (28S). ∎ 4.4

The last theorem can be generalised by dropping the liveness assumption; see [17], theorem (28S). Often one needs to use the fact that markings are 'reproducible'. In T-systems, reproducibility is very much related to liveness.

Definition 4.5 *Reproducibility*

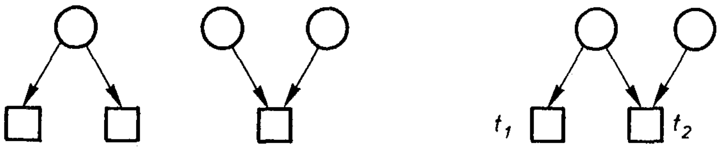
A marking $M \in [M_0\rangle$ of $\Sigma = (S,T;F,M_0)$ is reproducible *iff* there is an occurrence sequence σ of non-zero length such that $M = first(\sigma)$ and $M = last(\sigma)$, that is, σ starts with M and ends with M. ∎ 4.5

Theorem 4.6 *Link between reproducibility and liveness*

A strongly connected T-system $\Sigma = (S,T;F,M_0)$ is live iff its initial marking M_0 can be reproduced by σ in such a way that every transition occurs exactly once in σ.

Proof: For the direction ($\Rightarrow$) of this theorem, see [12], theorem 7 and [17], theorem (15S).
The direction ($\Leftarrow$) is easy to prove: the reproducing sequence necessitates at least one token on each cycle, and liveness follows with theorem 4.3. ∎ 4.6

This theorem can be generalised by dropping the strong connectedness assumption (see [17], (14S) and (15S)). However, we will not bother to do so, since the chief interest is in strongly connected T-nets.



(i) included

(ii) excluded

Figure 10: Illustrating the free choice structure

Corollary 4.7 *Characterisation of liveness in strongly connected T-systems*

In a strongly connected T-system Σ the following are equivalent:

(i) *Σ is live.*

(ii) *All simple cycles of Σ carry at least one token.*

(iii) *The initial marking is reproducible such that every transition occurs exactly once.*

■ 4.7

In fact, strong connectedness nicely characterises the existence of a live and safe marking:

Theorem 4.8 *Existence of live and 1-safe markings of T-nets*

A T-net N can be endowed with a live and 1-safe marking iff it is strongly connected.

Proof: See [12], theorem 4, and [17], theorem (32S). ■ 4.8

The class of nets which is dual to T-nets is called S-nets [6]; their characterising property is that $|{}^\bullet t| \leq 1$ and $|t^\bullet| \leq 1$ always holds for $t \in T$. S-nets allow conflict, but no synchronisation. The behavioural theory of S-nets is rather simpler than the theory of T-nets[3], unless one is interested in information flow: [25] have developed a nice and non-trivial theory of information flow in S-nets. S-nets play a particularly important rôle as substructures of larger nets, a topic which will not be studied here (but see [3]).

5 Free choice systems

Free choice nets have been invented as a common generalisation of S-nets and T-nets, with the aim of retaining as much as possible of the nice theory of these classes. They allow synchronisation (but only in the 'T-net way') and conflicts (but only in the 'S-net way'). The former is to say that if two places share a common output transition then they may not have any further output transitions, and the latter is to say that if two transitions share a common input place then they may not have any further input places. But these two properties are equivalent! They allow the structures shown in Figure 10(i) but exclude the structure shown in Figure 10(ii).

[3] The reader may check that a strongly connected S-net is live iff it carries at least one token and 1-safe iff it carries at most one token.

Definition 5.1 *Free choice nets and free choice systems*

> $\Sigma = (S, T; F, M_0)$ is called a free choice system (abbreviated FC system) *iff* its underlying net is free choice, i.e. for all $t_1, t_2 \in T, t_1 \neq t_2: {}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset \Rightarrow |{}^\bullet t_1| = 1 = |{}^\bullet t_2|$. ■ 5.1

It is clear that every T-net, as well as every S-net, is free choice; that is, free choice nets are indeed a common generalisation. F.Commoner and M.Hack have shown that there exist generalisations of the two theorems 4.3 and 4.4 about liveness and safeness of T-systems. These generalisations will be described below in sections 8.1 and 8.2, respectively.

An essential consequence of the free choice property is that if t_1 and t_2 share a common input place then it can never be the case that one of them is enabled while the other is not. That is, every marking enables either both of them or none of them. This may be contrasted with the (excluded) case of Figure 10(ii) where a marking can be found which enables t_1 but not t_2. The next result shows that, in the sense of the simulation notion defined in section 3, the property just explained is a characteristic one.

Definition 5.2 *Extended and behavioural free choice systems*

> $\Sigma = (S, T; F, M_0)$ is
>
> (a) extended free choice (EFC) *iff* $\forall t_1, t_2 \in T: {}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset \Rightarrow {}^\bullet t_1 = {}^\bullet t_2$.
>
> (b) behaviourally free choice (BFC) *iff*
> $\forall t_1, t_2 \in T: {}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset \Rightarrow \forall M \in [M_0\rangle: M$ enables $t_1 \Longleftrightarrow M$ enables t_2.
>
> ■ 5.2

It is immediate that every FC system is also EFC. Furthermore, every EFC system satisfies the BFC Property. Conversely:

Theorem 5.3 *Equivalence of FC, EFC and BFC systems w.r.t. simulation*

> (i) *Every BFC system can be simulated by an EFC system.*
>
> (ii) *Every EFC system can be simulated by an FC system.*

Proof: See [7]; the two easy constructions are sketched in Figure 11. ■ 5.3

Theorem 5.3 shows that the three classes of FC systems, EFC systems and BFC systems are 'the same modulo simulation'. In the sequel we shall consider FC nets only.

We will take a closer look at liveness in FC systems. Our aim is to relate liveness to a weaker, easier-to-check, property called deadlock-freeness.

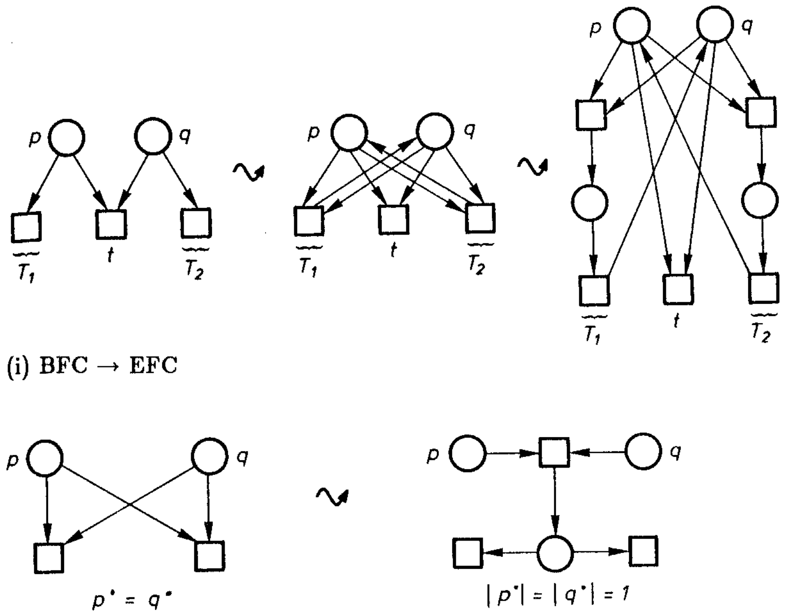
Definition 5.4 *Deadlock-freeness*

> $\Sigma = (S, T; F, M_0)$ is called deadlock-free *iff* $\forall M \in [M_0\rangle \; \exists t \in T: M$ enables t. ■ 5.4

A system is deadlock-free if it may always go on working as a whole; no global system 'stop' is possible.

Liveness implies deadlock-freeness in general, simply because by definition, every net contains at least one transition. Conversely, Figure 12 shows a P/T-system (even a 1-safe free choice one) which is deadlock-free but not live.

However, it is possible to characterise those transitions whose liveness is guaranteed by deadlock-freeness:



(i) BFC → EFC

(ii) EFC → FC

Figure 11: Constructions reducing EFC and BFC systems to FC systems

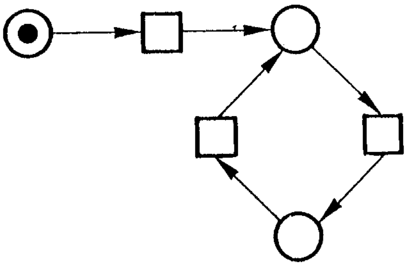


Figure 12: A 1-safe FC system which is deadlock-free but not live

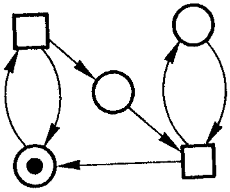


Figure 13: A strongly connected FC system which is deadlock-free but not live

Theorem 5.5 *Relationship between deadlock-freeness and liveness in FC systems*

> *Let $\Sigma = (S, T; F, M_0)$ be a 1-safe FC system which is deadlock-free and let $t \in T$ be such that $\forall t' \in T\colon (t', t) \in F^*$ (i.e. t can be reached from every other transition by a directed path). Then t is live.*

Proof: The proof proceeds by contradiction. Assume that t is not live; then there exists a marking $M_1 \in [M_0\rangle$ at which t is dead, i.e. no successor marking of M_1 enables t.

Consider any $p \in {}^\bullet t$; then, by the free choice property, all $t' \in p^\bullet$ (not just t itself) are dead at M_1. But this implies that any token put on p after M_1 will remain there.

By 1-safeness, it follows that the transitions in ${}^\bullet p$ can occur at most once, i.e. there is a marking $M_2 \in [M_1\rangle$ at which all transitions in ${}^\bullet p$ are dead.

Since this holds for all $p \in {}^\bullet t$ (and since the net is finite), there is some $M_3 \in [M_1\rangle$ at which all transitions in ${}^\bullet({}^\bullet t)$ are dead.

Repeating this argument shows that every transition in the set $\{t' \in T \mid (t', t) \in F^*\}$ can be made dead; but since by assumption, the latter set equals T, this means that a deadlock can be reached. ■ 5.5

Corollary 5.6 *Liveness equals deadlock-freeness in strongly connected FC systems*

> *Let Σ be a 1-safe strongly connected FC system.*
> *Then Σ is live iff Σ is deadlock-free.* ■ 5.6

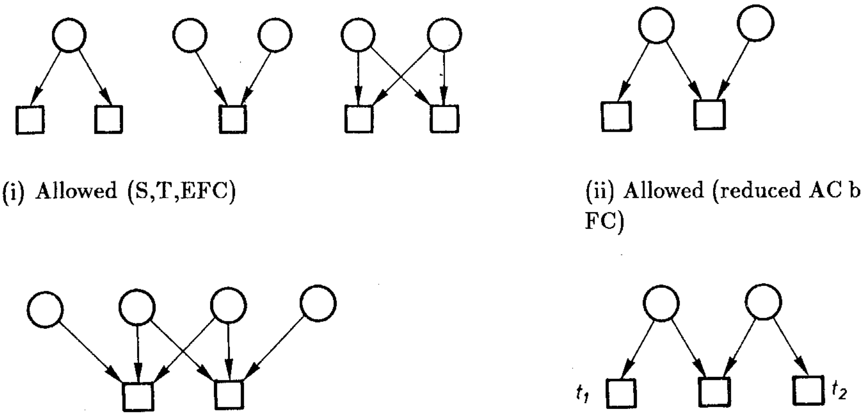
It is seen readily that the 1-safeness assumption in 5.5 and 5.6 can be weakened to n-safeness (for any n). However, Figure 13 shows that it cannot be omitted altogether; this observation, as well as the proof of 5.5/5.6 are due to D.Hillen [24].

6 Asymmetric choice systems

The EFC (extended free choice) property can be rewritten equivalently as follows:

$$\forall s_1, s_2 \in S\colon s_1^\bullet \cap s_2^\bullet \neq \emptyset \Rightarrow s_1^\bullet = s_2^\bullet.$$

It has turned out that some of the free choice results hold also if a weaker condition is assumed, namely $(s_1^\bullet \subseteq s_2^\bullet \vee s_2^\bullet \subseteq s_1^\bullet)$ instead of $s_1^\bullet = s_2^\bullet$ in the above formula. The resulting class of nets will be called asymmetric choice nets.



(i) Allowed (S,T,EFC)

(ii) Allowed (reduced AC but not FC)

(iii) Allowed (AC but not EFC)

(iv) Excluded

Figure 14: Illustrating the AC structure

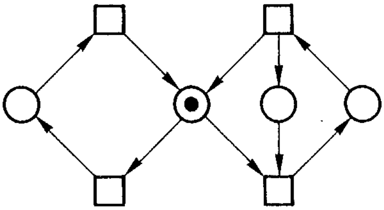


Figure 15: A simple AC system

Definition 6.1 *Asymmetric choice systems*

$\Sigma = (S, T; F, M_0)$ is called asymmetric choice (AC) system *iff*
$\forall s_1, s_2 \in S\colon s_1^\bullet \cap s_2^\bullet \neq \emptyset \Rightarrow (s_1^\bullet \subseteq s_2^\bullet \vee s_2^\bullet \subseteq s_1^\bullet)$. ■ 6.1

Figures 14(i)-(iii) show some allowed structures while Figure 14(iv) shows the typical structure which is excluded by the asymmetric choice property. It is immediate that every (extended) free choice net is also asymmetric choice.

There are many theorems which hold for FC systems but fail to hold for AC systems. An example is corollary 5.6: Figure 15 shows an AC system which is 1-safe, strongly connected and deadlock-free but not live.

Just as for FC systems, there is a property which explains AC systems in behavioural terms; but it is slightly more complicated. We introduce it by means of Figure 14(iv). In this Figure, a marking which enables t_1 but not t_2, as well as a marking which enables t_2 but not t_1, can be found. The behavioural correspondent of the asymmetric choice property excludes just this possibility. That is, if t_1 and t_2 are related to each other as in Figure 14(iv) (and this means: if there is some $t_0 \in ({}^\bullet t_1)^\bullet \cap ({}^\bullet t_2)^\bullet$) then either the enablement of t_1 must always imply the enablement of t_2, or the other way round (hence the name: 'asymmetric choice property'). This property is captured by part (i) of the next definition.

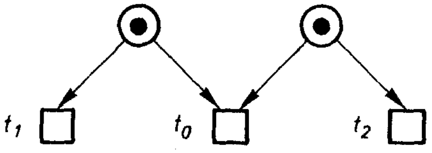


Figure 16: A non-transitive conflict situation

Definition 6.2 *Behavioural and reduced asymmetric choice property*

Let $\Sigma = (S, T; F, M_0)$ be a P/T-system.

(i) Σ is behaviourally asymmetric choice (BAC) *iff* for all $t_1, t_2 \in T$:

$$\begin{aligned}({}^\bullet t_1)^\bullet \cap ({}^\bullet t_2)^\bullet = \emptyset \;&\vee\; (\forall M \in [M_0\rangle : M \text{ enables } t_1 \Rightarrow M \text{ enables } t_2)\\ &\vee\; (\forall M \in [M_0\rangle : M \text{ enables } t_2 \Rightarrow M \text{ enables } t_1).\end{aligned}$$

(ii) Σ is reducedly asymmetric choice (RAC) *iff* for all $p, q \in S$:

$$\begin{aligned}p^\bullet \cap q^\bullet = \emptyset \;&\vee\; (|p^\bullet| = 1 \wedge |q^\bullet| \leq 2 \wedge {}^\bullet(q^\bullet) = \{p, q\})\\ &\vee\; (|q^\bullet| = 1 \wedge |p^\bullet| \leq 2 \wedge {}^\bullet(p^\bullet) = \{p, q\}).\end{aligned}$$

■ 6.2

The reduced AC property allows all free choice structures having no more than two input places for each transitions, and only one type of non-FC structure, namely the very simplest AC structure shown in Figures 10(ii) and 14(ii). It excludes AC structures such as shown in Figure 14(iii) and even EFC structures such as the third net of Figure 14(i). Thus RAC systems seem to be a 'tiny' subclass of AC systems. However, there is the following:

Theorem 6.3 *Characterisation of asymmetric choice nets*

Let Σ be a P/T-system.

(a) *If Σ is RAC then Σ is BAC.*

(b) *If Σ is BAC then Σ can be simulated by Σ' such that Σ' is AC.*

(c) *If Σ is AC then Σ can be simulated by Σ' such that Σ' is RAC.*

Proof: (a) is obvious from the definitions.
For (b), see [7], theorem 3.4. For (c), see [7] and [2], theorem 4. ■ 6.3

Part (c) of this theorem means that all the complexity of AC nets is already hidden in reducedly AC nets. But notice that (unlike in the FC case) not all (structurally) AC systems are behaviourally AC.

A further useful fact about AC systems is that the conflict relation is transitive. To see that this need not always be true, consider the 'typical' non-AC net of Figure 14(iv) with the marking shown in Figure 16. In this marking, both t_1 and t_2 are in conflict with t_0, but they are concurrently enabled, i.e. not in conflict with each other.

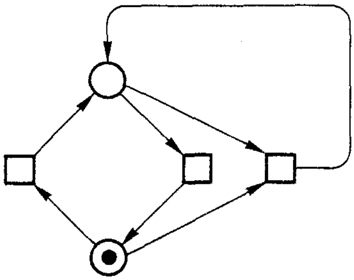


Figure 17: A system which is place-live but not live

Theorem 6.4 *Conflict is transitive in AC systems*

Let $\Sigma = (S,T;F,M_0)$ be an AC system, let $M \in [M_0\rangle$ be a marking and let t_0, t_1, t_2 in T be such that both t_1 and t_2 are in conflict with t_0 at M.
Then t_1 is in conflict with t_2 at M.

Proof: By lemma 2.8($\Rightarrow$), we may pick $p \in {}^\bullet t_1 \cap {}^\bullet t_0$ and $q \in {}^\bullet t_0 \cap {}^\bullet t_2$ such that $M(p) = 1$ and $M(q) = 1$. By the AC property, either $p^\bullet \subseteq q^\bullet$ or $q^\bullet \subseteq p^\bullet$. If $p^\bullet \subseteq q^\bullet$ then $q \in {}^\bullet t_1 \cap {}^\bullet t_2$ and hence, by lemma 2.8($\Leftarrow$), t_1 and t_2 are in conflict at M. If $q^\bullet \subseteq p^\bullet$ then $p \in {}^\bullet t_1 \cap {}^\bullet t_2$, yielding the same conclusion. ■ 6.4

We will prove another typical property of AC nets. It yields a characterisation of liveness (which we will call 'place-liveness') which is easier to check. Place-liveness captures the idea that no place of a net can ever become empty and unable to receive a token again.

Definition 6.5 *Place-liveness*

$\Sigma = (S,T;F,M_0)$ is place-live *iff* $\forall M_1 \in [M_0\rangle\ \forall s \in S\ \exists M \in [M_1\rangle : M(s) > 0$. ■ 6.5

Lemma 6.6 *Liveness implies place-liveness*

If $\Sigma = (S,T;F,M_0)$ is live then Σ is place-live.

Proof: If, for $s \in S$, ${}^\bullet s \neq \emptyset$ then whenever $t \in {}^\bullet s$ occurs, a token is put (or remains) on s.
If $s^\bullet \neq \emptyset$ then whenever $t \in s^\bullet$ occurs, a token must previously have been on s.
The case ${}^\bullet s = \emptyset = s^\bullet$ is excluded by the definition of a net. ■ 6.6

Figure 17 shows the typical example of a (non-AC) system which is place-live but not live.

We intend to show next that such a case cannot occur in an AC net. For the purpose of proving this, the following small technical lemma is helpful.

Lemma 6.7 *A technical lemma*

Let $\Sigma = (S, T; F, M_0)$, $M \in [M_0\rangle$ and $t, t' \in T$ such that:

(i) $^\bullet t = \{s_1, \ldots, s_m\}$ *and* $s_1^\bullet \subseteq s_2^\bullet \subseteq \ldots \subseteq s_m^\bullet$*;*

(ii) *for some* i*,* $1 \leq i \leq m$*,* $s_1, \ldots, s_i$ *are marked under* M*, i.e.* $M(s_1), \ldots, M(s_i) > 0$*;*

(iii) $^\bullet t' \cap \{s_1, \ldots, s_i\} \neq \emptyset$*;*

(iv) M *enables* t'*.*

Then M *also enables* t*.*

Proof: Suppose not, then as a consequence of (ii), $\exists q \in \{s_{i+1}, \ldots, s_m\}: M(q) = 0$.
But by (iii), $^\bullet t' \cap \{s_1, \ldots, s_i\} \neq \emptyset$, say $p \in {^\bullet t'} \cap \{s_1, \ldots, s_i\}$.
By (i), $p^\bullet \subseteq q^\bullet$, which implies $t' \in q^\bullet$.
Hence M does not enable t', contradicting (iv). ■ 6.7

Theorem 6.8 *Equivalence of liveness and place-liveness for AC systems*

Let $\Sigma = (S, T; F, M_0)$ be an AC system. Then Σ is live iff Σ is place-live.

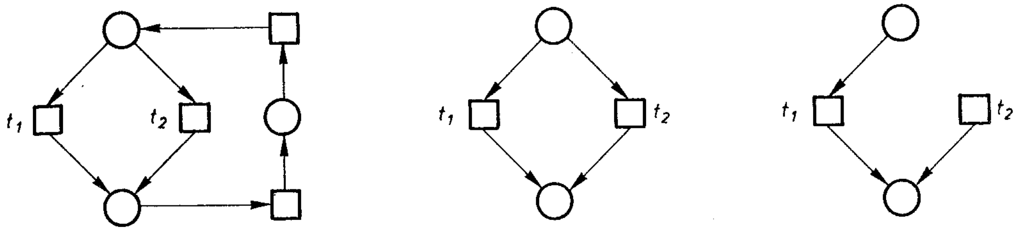
Proof: ($\Rightarrow$) follows from 6.6.
To prove ($\Leftarrow$), let $M \in [M_0\rangle$, $t \in T$ and $^\bullet t = \{s_1, \ldots, s_m\}$; we have to prove that t can be enabled from M.
The AC property implies that the $s_1, \ldots, s_m$ can be linearly ordered as in 6.7(i), so without loss of generality we may assume $s_1^\bullet \subseteq \ldots \subseteq s_m^\bullet$.
We construct a reachable marking which enables t by putting tokens on $s_1, \ldots, s_m$, one after the other, in this order.
By place-liveness, there exists a marking $M_1 \in [M\rangle$ which marks s_1, i.e. $M_1(s_1) > 0$.
Suppose that a marking M_i has been reached which marks all of $s_1, \ldots, s_i$, i.e. satisfying 6.7(ii), for some $i, 1 \leq i < m$. Then by place-liveness, a marking M_{i+1} in $[M_i\rangle$ exists which marks s_{i+1}, i.e. $M_{i+1}(s_{i+1}) > 0$. Two cases are possible:

(1) In the transition from M_i to M_{i+1}, an output transition t' of $\{s_1, \ldots, s_i\}$ has occurred, removing a token from one of $s_1, \ldots, s_i$. In this case, lemma 6.7 can be applied to show that the marking which enables t' also enables t, and the proof is done.

(2) In the transition from M_i to M_{i+1}, all tokens have remained on $s_1, \ldots, s_i$. In this case, M_{i+1} marks $s_1, \ldots, s_{i+1}$, and the construction can be repeated; eventually, all $s_1, \ldots, s_m$ are marked and t is enabled.

■ 6.8

A very similar result is lemma 4.3 of [26] which states that every dead transition in an AC system has an input place which remains unmarked. Theorem 6.8 holds even if the AC premise is changed to BAC. The proof is given in [7] (proposition 3.8); it is different from the above because it is not obvious that the construction which associates a simulating AC system to each BAC system preserves place-liveness.



(i) A net (ii) A subnet (iii) Not a subnet

Figure 18: Illustrating the definition of a subnet

7 T-component covers of free choice systems

The small results presented so far hardly scratch the surface of the rich and elegant structure theory of free choice nets uncovered by F.Commoner [11] and M.Hack [21], and amplified (and made accessible to a wider audience) by other authors [41,40,15,9,38].

It is impossible to explain all of the results in detail in these notes. We shall concentrate on only one of them and explain its proof in full. This is M.Hack's theorem that a live and 1-safe FC system is covered by strongly connected T-components. It is the basis for many further results about free choice nets.

This theorem was chosen for two reasons. Firstly, its proof highlights some typical ways of arguing about free choice nets. Secondly, its proof is rather 'hidden' in the published literature. Hack's original proof in [21] is not very easily accessible. Besides, it is not exactly perspicuous and contains some serious mistakes which have been corrected in [22] and in [14]. Thus, at present, one would have to examine closely at least three papers in order to understand it.

To introduce T-components we first need the notion of a subnet.

Definition 7.1 *Subnet*

Let $N = (S,T;F)$ and $N_1 = (S_1,T_1;F_1)$ be two nets.
N_1 is a subnet of N *iff* $S_1 \subseteq S$, $T_1 \subseteq T$ and $F_1 = F \cap ((S_1 \times T_1) \cup (T_1 \times S_1))$. ■ 7.1

The restriction on F_1 in this definition deserves an explanation. It means that F_1 must contain not just some subset of the F-arrows between elements of $S_1 \cup T_1$, but even *all* such F-arrows. Figure 18 explains the difference.

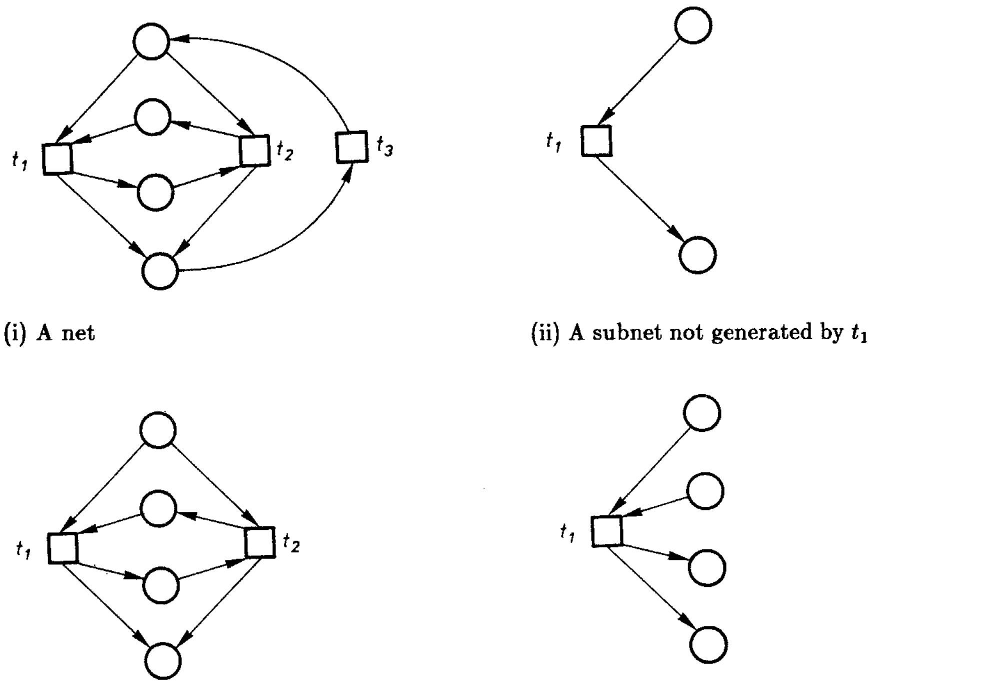
T-components are special subnets which satisfy two further conditions. Firstly, if a T-component contains a transition then it must contain all of its bordering places as well. In this case we say that the subnet is 'generated' by its transitions:

Definition 7.2 *Transition generated subnets*

Let $N_1 = (S_1,T_1;F_1)$ be a subnet of $N = (S,T;F)$. N_1 is generated by T_1 *iff*
$S_1 = {}^\bullet T_1 \cup T_1^\bullet$ (where the presets and postsets are taken w.r.t. F). ■ 7.2

For any $T_1 \subseteq T$ there is always exactly one subnet generated by T_1, namely the net $N_1 = (S_1,T_1;F_1)$ with $S_1 = {}^\bullet T_1 \cup T_1^\bullet$ and $F_1 = F \cap ((S_1 \times T_1) \cup (T_1 \times S_1))$.

The second requirement to be satisfied by a T-component is that 'by itself' it must be a T-net. We capture this as follows:



(i) A net

(ii) A subnet not generated by t_1

(iii) A subnet generated by $\{t_1, t_2\}$ which is not a T-component

(iv) A non-strongly connected T-component (generated by t_1)

Figure 19: Illustrating the definition of T-components

Definition 7.3 *T-components*

> $N_1 = (S_1, T_1; F_1)$ is called a T-component of $N = (S, T; F)$ *iff* N_1 is the subnet generated by T_1 and, in addition, $\forall s \in S_1\colon |{}^\bullet s \cap T_1| \leq 1 \wedge |s^\bullet \cap T_1| \leq 1$ (where the preset and the postset are taken w.r.t. F, but it would come to the same if they were taken w.r.t. F_1). ■ 7.3

A T-component N_1 will be called strongly connected iff it is strongly connected as a T-net 'by itself', that is, if there is a directed F_1-path between any two distinct elements of N_1. Definitions 7.1-7.3 are quite subtle and deserve careful study. Figure 19 explains them further. The net shown in Figure 19(i) has no strongly connected T-component. We will give examples of strongly connected T-components later in this section (Figure 21).

The reader should be cautioned that T-components are different from T-invariants [6]. While the set $\{t_1, t_2, t_3\}$ in Figure 19(i) defines a T-invariant (where one has to count t_3 twice), it does not define a T-component. On the other hand, every strongly connected T-component (and those are of chief interest) defines also a T-invariant. In this sense, T-invariants are more general. In the following, the notion of a T-invariant will play no further rôle, but see [28].

The objective of this section is to prove, by elementary means, Hack's result that every live and 1-safe free choice system Σ is covered by strongly connected T-components. That is to say: every place and every transition of a live and safe free choice system is contained in some strongly connected T-component. A moment's reflection reveals that the theorem need only be proved for transitions, since the covering of the places can be deduced immediately from that of the transitions (using the property that T-components are generated by their transitions). Hence the essential statement which we wish to prove is the following:

> Every transition $\hat{t}$ of Σ lies on some strongly connected T-component $\hat{N}$ of Σ.

We solve this problem by defining an algorithm which constructs $\hat{N}$, given $\hat{t}$. Before defining this algorithm we give several examples. Figure 20 shows that the three preconditions (liveness, safeness and the free choice property) are necessary to establish the theorem.

The example in Figure 21 explains the construction of the T-component $\hat{N}$ covering $\hat{t}$. (In passing, the system shown in Figure 21(i) is live, 1-safe and FC, but its initial marking in not reproducible, so that corollary 4.7 fails to hold for FC nets.)

The task of the algorithm is to grow a T-component from a given single transition $\hat{t}$. Because the T-component is generated by its transitions, it stands to reason to extend the 'current' $\hat{N}$ at those transitions that have input places or output places *not* in $\hat{N}$. For instance, if $\hat{N}$ consists only of $\hat{t} = t_3$ initially (see Figure 21(i)), we might extend $\hat{N}$ by the output place s_5 of t_3 and hence include t_7 in it as well (to make $\hat{N}$ strongly connected). But one has to be careful because in the next step, $\hat{N}$ may not be further extended by t_2 (this being an output transition of an output place of t_7), since no T-component will result; we will have to choose t_1 rather than t_2.

To account for this, our algorithm extends $\hat{N}$ not in terms of single elements but in terms of certain 'nice' directed F-paths. The initial $\hat{N}$ (which equals $\hat{t}$) in Figure 21(i) will be extended, in the first iteration, by the cycle consisting of $\hat{t}$, t_7 and t_1 as a whole (see Figure 21(ii)). The next iteration detects that t_1 has an output place not yet in $\hat{N}$, and another path including t_4 will be added to $\hat{N}$. The construction then finishes because the subnet shown in Figure 21(iii) is already a T-component of the original net. Formally, the algorithm is defined as follows:

Algorithm 7.4 *Algorithm to construct T-components*

Let $\Sigma = (S, T; F, M_0)$ be a live and 1-safe free choice system and let $\hat{t} \in T$. We construct inductively a triple $\hat{N} = (\hat{S}, \hat{T}; \hat{F})$ which will turn out to be a strongly connected T-component containing $\hat{t}$.

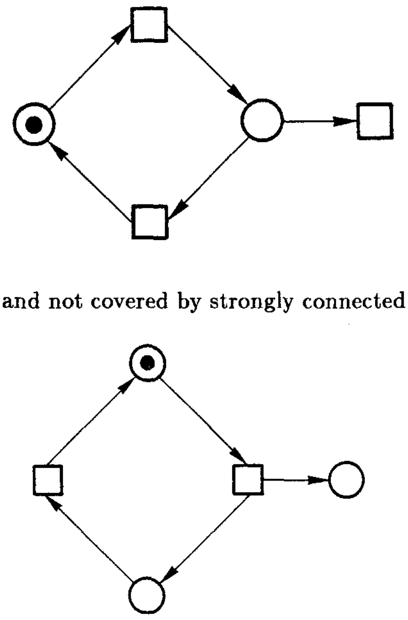
Step 1: $\hat{S} := \emptyset$, $\hat{T} := \{\hat{t}\}$, $\hat{F} := \emptyset$ and $\hat{N} := (\hat{S}, \hat{T}; \hat{F})$.

Step 2: Repeat the following exhaustively: If there is $t \in \hat{T}$ with $t^\bullet \not\subseteq \hat{S}$ then choose $s \in t^\bullet \backslash \hat{S}$ arbitrarily and $t' \in s^\bullet$ in such a way that there is a nice path $p = \{t_0, s_1, t_1, \ldots, s_m, t_m\}$ from $t' = t_0$ to $\hat{N}$ (see below for what this means); then put

$$\begin{aligned} \hat{S} &:= \hat{S} \cup \{s\} \cup \{s_1, \ldots, s_m\} \\ \hat{T} &:= \hat{T} \cup \{t_0, \ldots, t_m\} \\ \hat{F} &:= \hat{F} \cup \{(t, s), (s, t')\} \cup \{(t_0, s_1), (s_1, t_1), \ldots, (s_m, t_m)\} \\ \text{and} \quad \hat{N} &:= (\hat{S}, \hat{T}; \hat{F}). \end{aligned}$$

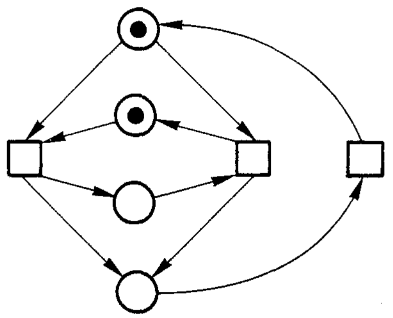
■ 7.4

Notice that we immediately have $\hat{S} \subseteq S$, $\hat{T} \subseteq T$ and $\hat{F} \subseteq F$ always. However, it is not even clear that $\hat{N}$ is a subnet, leave alone a T-component, of Σ.



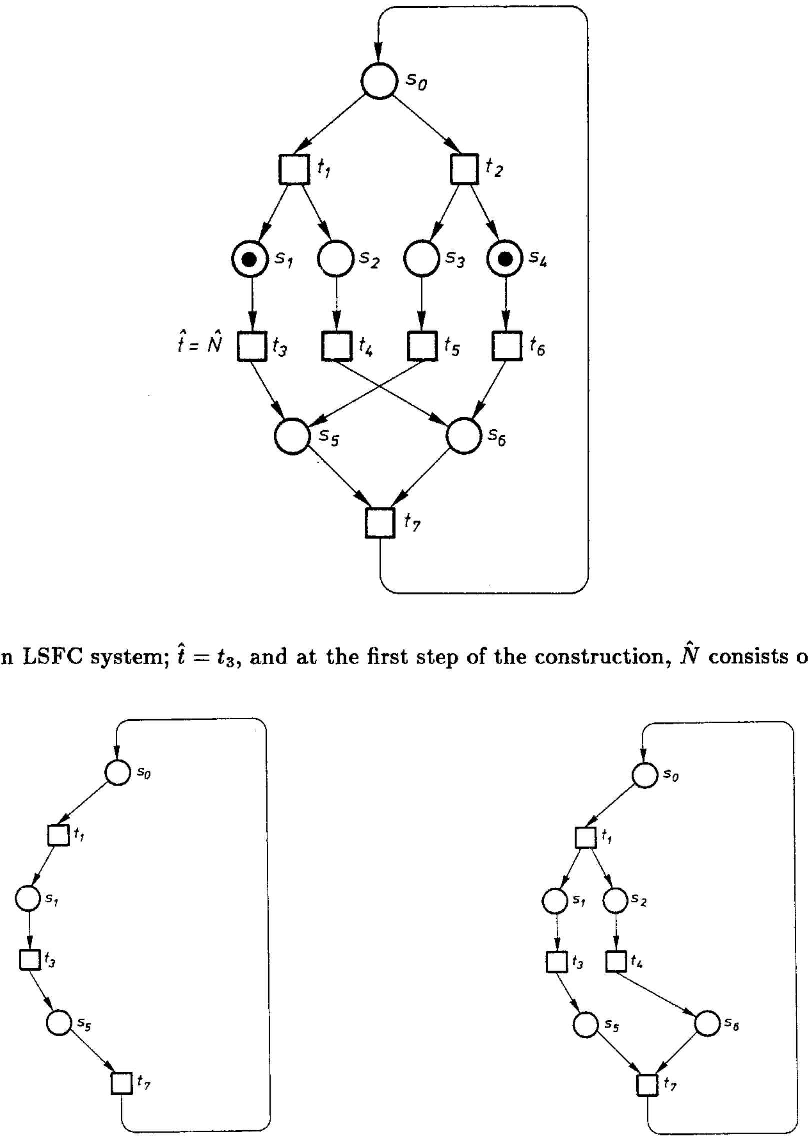
(i) 1-safe, FC and not covered by strongly connected T-components (but not live)

(ii) live, FC and not covered by strongly connected T-components (but not safe)



(iii) live, 1-safe and not covered by strongly connected T-components (AC but not FC)

Figure 20: Illustrating the preconditions of the main theorem



(i) An LSFC system; $\hat{t} = t_3$, and at the first step of the construction, $\hat{N}$ consists only of $\hat{t}$.

(ii) Second step of the construction: $\hat{N}$ is just a simple cycle.

(iii) Third and last step of the construction: the final $\hat{N}$.

Figure 21: Illustrating the algorithm

Definition 7.5 *Nice paths*

With the notation as in 7.4, $p = \{t_0, s_1, \ldots, s_m, t_m\}$ is a nice path from t' to $\hat{N}$ *iff*

- $(t_{j-1}, s_j) \in F \wedge (s_j, t_j) \in F\,(1 \leq j \leq m)$ (p is a directed path)
- p is simple (i.e. no element occurs twice in it)
- $t' = t_0$ and $t_m \in \hat{T}$
- $s_j \notin \hat{S}$ $(1 \leq j \leq m)$ and $t_j \notin \hat{T}$ $(1 \leq j < m)$.

The special case that $m = 0$ is allowed; we then have $t' = t_0 = t_m \in \hat{T}$. ■ 7.5

p being a 'nice path' simply means that p must start with t' and lead back into $\hat{N}$ at a *transition*; it is important that p may not directly lead to a place of $\hat{N}$. If $t' \in \hat{T}$ already, then p is the trivial path $\{t'\}$, and all that gets added to $\hat{N}$ by 7.4 is the place s. Otherwise, both s and the path p are included in the new $\hat{N}$. Hence what is added to $\hat{N}$ at each step of 7.4 is always a simple path which leaves $\hat{N}$ at a transition (namely at t) and re-enters $\hat{N}$ at another transition (namely at t_m, which might equal t'). It could even be the case that $t = t'$; then s is a side condition of t which will be included in the new $\hat{N}$.

The reader should carefully check this construction on the example of Figure 21(i) and convince himself that with $\hat{t} = t_3$ only the two iterations as shown in Figures 21(ii) and 21(iii) are possible. However, the construction need not be deterministic; to see this, the reader could try $\hat{t} = t_7$ in Figure 21(i). The reader may also wish to try the example of Figure 20(iii) to find out that the algorithm works as well, but may yield a triple $\hat{N}$ which is not a subnet.

It has to be shown that a transition $t' \in s^\bullet$ with the properties demanded in 7.4 always exists. Furthermore, it has to be shown that when the construction is completed, $\hat{N}$ is a strongly connected T-component. The rest of this section is devoted to the proofs of these two statements.

Let us first collect a few simple facts about the construction. First, every place is handled (i.e. added to $\hat{N}$) only once, either as some $s \in t^\bullet \backslash \hat{S}$ or as an s_j in some path p of construction 7.4. As a result, every place in $\hat{S}$ has exactly one incoming $\hat{F}$-arc and exactly one outgoing $\hat{F}$-arc (although it may have many other incoming and outgoing arcs). Moreover, every transition in $\hat{T}$ has at least one incoming $\hat{F}$-arc and at least one outgoing $\hat{F}$-arc, except at the very beginning when $\hat{N}$ consists only of $\hat{t}$. Also, at any stage of the construction, $\hat{N}$ is strongly connected in terms of $\hat{F}$ (i.e. every two distinct elements of $\hat{N}$ are connected by a directed $\hat{F}$-path), since at the very beginning, $\hat{N}$ is trivially strongly connected and adding directed paths emanating from $\hat{N}$ and leading back to $\hat{N}$ does not destroy the strong connectedness of $\hat{N}$. These four properties hold at every iteration of the algorithm (with the only exception that initially, $\hat{t}$ does not have any bordering $\hat{F}$-arcs).

Let us now turn to the proof that a transition $t' \in s^\bullet$ exists with the properties required in 7.4, provided $s \notin \hat{S}$ and $t \in \hat{T} \cap {}^\bullet s$ are as in 7.4. Figure 22 shows the setup.

In this proof we will encounter the same type of arguments that have been employed in the basic lemmata of [41]. In particular, we will use the notion of a maximal marking which plays an important rôle in lemmata 3.1 and 3.2 of [41]. Unfortunatley, it is not possible to apply these lemmata directly here, because their proofs in [41] make actual use of Hack's results, particularly the dual of the one we wish to prove here, and because the class of nets considered in [41] slightly differs from the class we are interested in. Hence we will have to do the proof 'from scratch'.

Now let us consider the initial setup of the problem shown in Figure 22. We know that $t \in \hat{T}$, $s \notin \hat{S}$ and $s \in {}^\bullet t$, and we wish to prove that there is some $t' \in s^\bullet$ from which a simple directed path re-enters $\hat{N}$ at a transition[4]. First, let us settle the case that s is also an input place of t, i.e.

[4] After reading the first version of this paper, P.S.Thiagarajan has noticed that, in fact, *every* transition $t' \in s^\bullet$ can be used to start some simple path which re-enters $\hat{N}$ at a transition. For a proof which yields this result, the reader is referred to a forthcoming paper [8]. Also, J.Desel has recently found a direct proof of this fact [13].

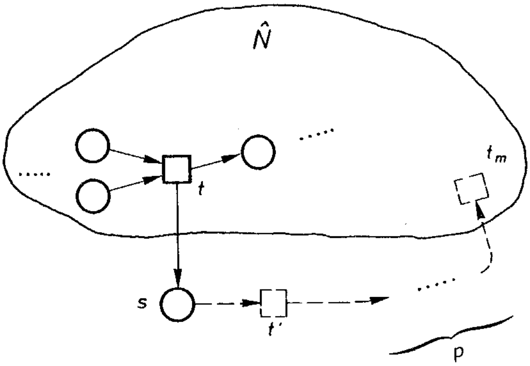


Figure 22: Knowing t and s, t' and p must be found

a side condition of t. Then the choice $t = t'$ already satisfies the requirements, so that we need not consider this case any further.

Since the system is live, the transition t can be enabled. After one occurrence of t, it can be re-enabled. But between two occurrences of t, some token must have disappeared from s because of 1-safeness, so that some $t' \in s^\bullet$ must have occurred between the two occurrences of t. The idea is that if the reproduction of the occurrence of t is properly chosen, then this t' might be a suitable transition which satisfies the requirements of the algorithm.

Let us make this argument more precise. We can find a sequence of the following form:

$$M_0 \ldots M[t\rangle M'[\ldots \tau \ldots\rangle M''[t\rangle M''',$$

where τ is a transition sequence leading from M' to M'' which re-enables t. We have just seen that because of the 1-safeness of s (and because $s \notin {}^\bullet t$), τ must contain some $t' \in s^\bullet$. Let us, in order to discount 'unimportant' occurrences (of transitions that have nothing to do with the part of the net presently under consideration) assume that τ is a *minimal* sequence with the above properties; that is, τ cannot be made any shorter. Then define $t_0 \in s^\bullet$ such that t_0 occurs in τ and consider the *last* occurrence of t_0 in τ:

$$M'\tau M'' \quad = \quad M' \ldots t_0 \underbrace{\ldots}_{\text{no } t_0} M''$$

If t_0 is in $\hat{T}$ then there is nothing more to prove; we have found a path of the required kind, namely $\{t_0\}$. If t_0 is not in $\hat{T}$ then we must consider the set of output places of t_0, i.e. $t_0^\bullet$, in order to prolong the path. Let us suppose, for the moment, that *none* of the output places of t_0 are in $\hat{S}$, i.e. that $t_0^\bullet \cap \hat{S} = \emptyset$. Because τ is minimal, the output places of t_0 cannot remain all marked from (the last occurrence of) t_0 to M'' in τ (if so, the last occurrence of t_0 could be omitted from τ). Hence between (the last occurrence of) t_0 and M'' in τ, some $t_1 \in t_0^{\bullet\bullet}$ must have occurred; consider again the last such occurrence. If t_1 is in $\hat{T}$ then there is again nothing more to prove, because a nice path leads from t_0 to t_1. If t_1 is not in $\hat{T}$, however, then we have to consider the output places of t_1. Assume again that we can prove that $t_1^\bullet \cap \hat{S} = \emptyset$. Then the argument can be repeated, using the minimality of τ again, to show that between (the last occurrence of) t_1 and M'' in τ, some $t_2 \in t_1^{\bullet\bullet}$ must have occurred. However, this cannot go on forever, since τ is a finite sequence. Hence eventually, some nice path must result; the *last* occurrences were always taken in order to ensure that this path is simple.

The above argument depends on the assumption that we have $t_i^\bullet \cap \hat{S} = \emptyset$ whenever $t_i \notin \hat{T}$ in τ. We now turn to look how this assumption can be ensured. Apparently, we have to examine more closely the possible shape of the sequence τ. Let us first see what it would mean for a transition

t (we are now re-using the previously fixed name t for an arbitrary transition) to have an output place in $\hat{S}$ but not to be contained in $\hat{T}$: $t^\bullet \cap \hat{S} \neq \emptyset$ and $t \notin \hat{T}$. Then every occurrence of t puts at least one token on the set of places $\hat{S}$. We call t an 'input transition' of $\hat{N}$ and define

$$T^{in} = \{t \in T \mid t^\bullet \cap \hat{S} \neq \emptyset \wedge t \notin \hat{T}\}$$

(or shorter: $T^{in} = {}^\bullet\hat{S} \backslash \hat{T}$) as the set of input transitions of $\hat{N}$. Symmetrically,

$$T^{out} = \{t \in T \mid {}^\bullet t \cap \hat{S} \neq \emptyset \wedge t \notin \hat{T}\}$$

(or shorter: $T^{out} = \hat{S}^\bullet \backslash \hat{T}$) is defined to be the set of output transitions of $\hat{N}$. Notice that T^{in} and T^{out} are defined for every step of the construction 7.4; they depend on (and vary with) $\hat{N}$. It could well be true that $T^{in} \cap T^{out} \neq \emptyset$ (even for the final $\hat{N}$), but we have, by definition, $T^{in} \cap \hat{T} = \emptyset$ as well as $T^{out} \cap \hat{T} = \emptyset$.

In terms of these definitions, we may rephrase our assumption: we must find a transition sequence τ with the above properties which does not contain any T^{in}-transitions. Our aim now becomes to show that every transition in $\hat{T}$ can be enabled and re-enabled by some τ which does not contain any transitions from T^{in}.

Since $\hat{N}$ always likens a (strongly connected) T-net, we may examine corollary 4.7 to find that, at least, all the cycles of $\hat{N}$ should be filled with tokens if there is to be a chance of $\hat{T}$-transitions being reproduced without the occurrence of input transitions. In order to achieve this, we may try to put as many tokens as possible on $\hat{N}$. To this end, let us call a marking M (again, *any* marking, not just the one considered previously) to be $\hat{N}$-maximal iff, starting from M, no further tokens can be put on $\hat{N}$ without some tokens having to be taken away first. More precisely, we define a marking M to be $\hat{N}$-maximal *iff* :

> $\forall\tau$: if τ is a transition sequence from M and τ does not contain any transitions from T^{out}, then τ does not contain any transitions from T^{in} either.

This means that $\hat{N}$ is 'saturated' at the marking M. We may hope that from a 'saturating' marking of $\hat{N}$, the transitions of $\hat{N}$ may be reproduced without the use of transitions from T^{in}.

Hence, our next subtask becomes to show that for every $\hat{N}$ that may arise in construction 7.4, an $\hat{N}$-maximal marking exists and can be reached from the initial marking. We prove this statement by counting the number of tokens on the simple cycles of $\hat{N}$. First we recall that (except at the very beginning) $\hat{N}$ is covered by simple cycles since it is strongly connected. Now consider any marking M and the number $\hat{n}(M)$, defined as follows:

$$\hat{n}(M) = \sum_{\hat{c} \text{ a simple cycle of } \hat{N}} M(\hat{c}),$$

where $M(\hat{c}) = \sum_{s \in S \cap \hat{c}} M(s)$. This number is simply the count of the tokens on the simple cycles of $\hat{N}$, such that each token is counted as often as it is covered by a simple cycle. (Initially, when the sum is empty, we may put $\hat{n}(M) = 0$ by definition.) We will now classify the transitions of the system in accordance with whether their occurrence increases $\hat{n}(M)$, decreases $\hat{n}(M)$, or leaves $\hat{n}(M)$ invariant.

First consider the transitions that have nothing to do with $\hat{N}$, i.e. are neither in $\hat{T}$ nor in T^{in} nor in T^{out}: $t \in T \backslash (\hat{T} \cup T^{in} \cup T^{out})$. Then, clearly, ${}^\bullet t \cap \hat{S} = \emptyset = t^\bullet \cap \hat{S}$, and hence, the occurrence of t changes nothing on the marking of $\hat{S}$. In particular, $\hat{n}(M)$ is left untouched since it depends only on the tokens of $\hat{S}$.

Next, consider a transition t in $T^{in} \backslash T^{out}$. Such a transition could be called a 'proper' input of $\hat{N}$. Because $t \notin T^{out}$ and $t \notin \hat{T}$, we have ${}^\bullet t \cap \hat{S} = \emptyset$; that is, the occurrence of t cannot decrease the

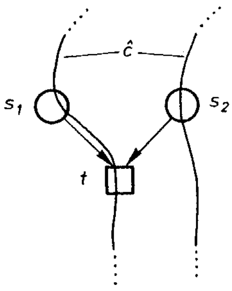


Figure 23: Illustrating the case that $|{}^\bullet t \cap \hat{c}| > 1$

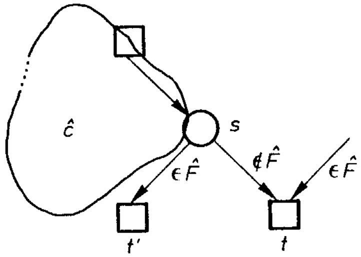


Figure 24: Illustrating the case that $|t^\bullet \cap \hat{c}| = 0$

number $\hat{n}(M)$. But can $\hat{n}(M)$ be left invariant, or is it always increased by the occurrence of t? $t \in T^{in}$ implies that for at least one $s \in S$ we have $s \in t^\bullet \cap \hat{S}$. But since $\hat{N}$ is strongly connected, there is some simple cycle $\hat{c}$ of $\hat{N}$ which contains s. The number of tokens on this cycle is increased by the occurrence of t. Hence the occurrence of $t \in T^{in} \backslash T^{out}$ always properly increases $\hat{n}(M)$.

Next, consider a transition $t \in \hat{T}$; this is the hardest case to analyse. The result we will eventually get is that the occurrence of t leaves the number $\hat{n}(M)$ invariant. First, however, we will only prove that the occurrence of t does not decrease $\hat{n}(M)$, which is sufficient for our present purposes. In order to prove this claim, we will have to examine what t can do to the cycles of $\hat{N}$.

Let $\hat{c}$ be any simple cycle of $\hat{N}$. We claim that $|{}^\bullet t \cap \hat{c}| \leq 1$ (where the preset is taken w.r.t. F). Suppose otherwise, i.e. $|{}^\bullet t \cap \hat{c}| > 1$, say $\{s_1, s_2\} \subseteq {}^\bullet t \cap \hat{c}$ (see Figure 23).

Then by the FC property, we have $s_1^\bullet = s_2^\bullet = \{t\}$, and hence $\hat{c}$ cannot be a *simple* cycle (t must occur at least twice in it); this holds even if s_1 or s_2 (or both) are side conditions of t. Hence, in general, $|{}^\bullet t \cap \hat{c}| \leq 1$. Thus, the only way that t could possibly reduce the number of tokens on $\hat{c}$ is if the following holds true:

$$|{}^\bullet t \cap \hat{c}| = 1 \text{ and } |t^\bullet \cap \hat{c}| = 0;$$

suppose this is true and $s \in {}^\bullet t \cap \hat{c}$. However, t cannot then be the output transition of s in $\hat{c}$, so there must be another one, say t' (see Figure 24).

We have $(s, t') \in \hat{F}$ since $\hat{c}$ is a cycle in $\hat{N}$, and hence $(s, t) \notin \hat{F}$ since, as we have seen previously, every place in $\hat{S}$ has exactly one $\hat{F}$-output arc. But since $t \in \hat{T}$, t must have at least one $\hat{F}$-input arc, a contradiction to the FC property. This finishes the proof that the occurrence of a transition $t \in \hat{T}$ cannot decrease the number $\hat{n}(M)$.

The last class of transitions to be considered is the set T^{out}. However, let us recall our current aim, which is to find an $\hat{N}$-maximal marking M of $\hat{N}$. The existence of such a marking can be shown even without considering the effect of the T^{out}-transitions on $\hat{n}(M)$. Because of 1-safeness, the number $\hat{n}(M)$ has a marking-independent upper bound (for instance, the number of simple cycles of $\hat{N}$ times the number of places on them). Hence there is some maximal number of times that transitions of $T^{in}\backslash T^{out}$ can occur without transitions of T^{out} necessarily having to occur, since the former properly increase the number $\hat{n}(M)$ while only the latter could possibly decrease it. To construct a $\hat{N}$-maximal marking, it is therefore sufficient to let $T^{in}\backslash T^{out}$-transitions (but not T^{out}-transitions) occur until no longer possible.

This ends our first subtask, namely to show that an $\hat{N}$-maximal marking exists. Our next task is to show that from an $\hat{N}$-maximal marking, any $t \in \hat{T}$ can be enabled and reproduced without any occurrences of T^{in}-transitions. In fact we will show that this can be done even without the occurrences of any T^{out}-transitions (which implies, by $\hat{N}$-maximality, that T^{in}-transitions do not occur either). Instead of the above, we will prove the following, even stronger, statement:

> Let $t \in \hat{T}$ be given. Every $\hat{N}$-maximal marking M can be transformed into another $\hat{N}$-maximal marking M' by $M[\tau\rangle M'$ such that τ does not contain any transitions from T^{out} (an hence also none from T^{in}), and M' enables t.

First of all, we show that $\hat{N}$-maximality is preserved by the occurrences of transitions not in T^{out}. Suppose that M is $\hat{N}$-maximal, $t \notin T^{out}$ and $M[t\rangle M'$; if τ were a transition sequence from M' which contains transitions from T^{in} but not T^{out}, then $t\tau$ would be a similar sequence from M. Hence M' must also be $\hat{N}$-maximal.

Next, we remark that every $t' \in T^{out}$ has a conflicting transition $t'' \in \hat{T}$, i.e. $t'' \in ({}^\bullet t')^\bullet \cap \hat{T}$. This simply follows because any place $s \in {}^\bullet t' \cap \hat{S}$ has (exactly) one $\hat{F}$-arc leading to $t'' \in \hat{T}$.

Now let t be an arbitrary transition in $\hat{T}$ and M be an arbitrary $\hat{N}$-maximal marking. We have to show that there is a transition sequence τ and a marking M' such that $M[\tau\rangle M'$, τ does not contain T^{out}-transitions and M' enables t. The idea is to construct τ in such a way that whenever some $t' \in T^{out}$ is in danger of occurring, we choose the conflicting $t'' \in \hat{T} \cap ({}^\bullet t')^\bullet$ instead. This can happen only a finite number of times before the given $t \in \hat{T}$ must of needs occur.

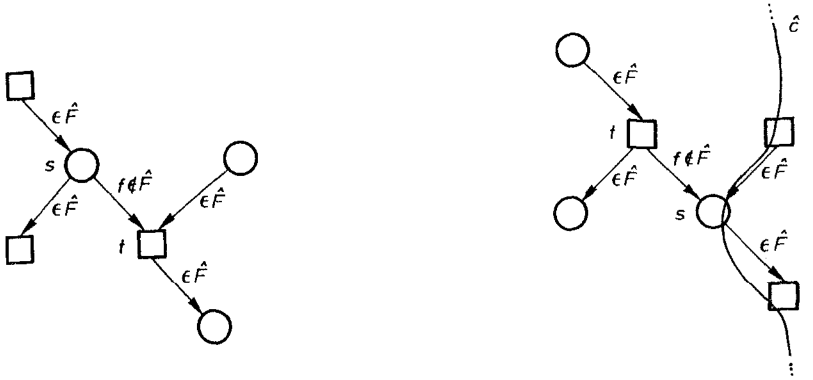
More precisely, we consider a first $t' \in T^{out}$ that can be enabled from M and let its corresponding $t'' \in \hat{T} \cap ({}^\bullet t')^\bullet$ occur; this is possible because of the (behavioural) free choice property. From the resulting marking, another $t' \in T^{out}$ can be chosen to be enabled, and so on. Thus we may construct a transition sequence which contains arbitrarily many $\hat{T}$-transitions but no T^{out}-transitions. However, there cannot be arbitrarily many $\hat{T}\backslash\{t\}$-occurrences! This follows as in theorem 5.5: any $t^1 \in \hat{T} \cap {}^{\bullet\bullet}t$ can occur at most once before t has to occur; any $t^2 \in \hat{T} \cap {}^{\bullet\bullet\bullet\bullet}t$ can occur at most twice before t has to occur, etc. The claim follows because $\hat{N}$ is strongly connected and finite.

This concludes the proof that at any stage of the construction 7.4, a transition $t' \in s^\bullet$ can be found which satisfies the requirements, i.e. from which a nice path leads back to a transition of $\hat{N}$. What remains to be done now is to show that the end result of construction 7.4 is indeed a T-component of $\hat{N}$ (that $\hat{N}$ is strongly connected has already been shown, and of course it contains $\hat{t}$).

From now on, let $\hat{N} = (\hat{S}, \hat{T}; \hat{F})$ denote the final result of construction 7.4. The proof that $\hat{N}$ is a T-component involves three steps:

(1) Show that $\hat{N}$ is a subnet.
(2) Show that $\hat{N}$ is generated by $\hat{T}$.
(3) Show that $\hat{N}$ satisfies 7.3.

(1) means that we have to show that whenever $x \in \hat{S} \cup \hat{T}$, $y \in \hat{T} \cup \hat{S}$ and $(x,y) \in F$ then $(x,y) \in \hat{F}$. For (2), we have to show that every $t \in \hat{T}$ satisfies $t^\bullet \subseteq \hat{S}$ and ${}^\bullet t \subseteq \hat{S}$. (3) means that $|{}^\bullet t \cap \hat{S}| = 1 = |t^\bullet \cap \hat{S}|$ (the equality can be taken rather than $\leq$ since $\hat{N}$ is strongly connected). But



(i) Case 1: $f = (s,t) \in (F\backslash\hat{F})$ (ii) Case 2: $f = (t,s) \in (F\backslash\hat{F})$

Figure 25: Illustrating the proof that $\hat{N}$ is a subnet

(3) follows immediately from (1), together with the fact that each $s \in \hat{S}$ has exactly one incoming $\hat{F}$-arc and exactly one outgoing $\hat{F}$-arc. Thus, all that remains to be proved are the statements (1) and (2).

Let us deal with (1) first. (1) is wrong if there is an F-arc f between two elements of $\hat{N}$ which is not also an $\hat{F}$-arc. Two cases are possible: f leads from a place to a transition, or f leads from a transition to a place. Let us first consider the case that $f = (s,t)$ with $s \in \hat{S}$ and $t \in \hat{T}$. If $f \notin \hat{F}$ then there must be $\hat{F}$-arcs bordering on s and t in the way shown in Figure 25(i); but this is excluded by the FC property, whence f must be in $\hat{F}$.

Let us then consider the case that $f = (t,s)$ with $t \in \hat{T}$ and $s \in \hat{S}$. If $f \notin \hat{F}$ then there must be $\hat{F}$-arcs bordering on t and s as shown in Figure 25(ii). We claim that the occurrence of t properly increases the number $\hat{n}(M)$ defined above! To see this, consider any simple cycle $\hat{c}$ of $\hat{N}$ which contains s. There are two possibilities: if t is not included in $\hat{c}$ then $|{}^\bullet t \cap \hat{c}| = 0$ and $|t^\bullet \cap \hat{c}| = 1$; if t is included in $\hat{c}$ then $|{}^\bullet t \cap \hat{c}| = 1$ and $|t^\bullet \cap \hat{c}| = 2$; in both cases, $|{}^\bullet t \cap \hat{c}| < |t^\bullet \cap \hat{c}|$ and hence the occurrence of t increases the number of tokens on $\hat{c}$. On the other hand we have seen that t can occur arbitrarily often in some transition sequence which does not contain any T^{out}-transitions. This contradicts the fact that $\hat{n}(M)$ is bounded by a marking-independent constant number, and thus we must have $f \in \hat{F}$. This finishes the proof of (1), i.e. that $\hat{N}$ is a subnet. (In passing, the last argument also implies that $|t^\bullet \cap \hat{c}| \leq 1$ for any $t \in \hat{T}$ and simple cycle $\hat{c}$ of $\hat{N}$ and that, as has been claimed above, the occurrence of $t \in \hat{T}$ leaves the number $\hat{n}(M)$ invariant.)

The last step in the whole proof is to show (2), i.e. the fact that $\hat{N}$ is generated by $\hat{T}$. One half of this is trivial, because from construction 7.4 it follows immediately that $t^\bullet \subseteq \hat{S}$ for all $t \in \hat{T}$ (in fact, this is the termination condition of the algorithm). We have to exclude the case that $t \in \hat{T}$ but ${}^\bullet t \not\subseteq \hat{S}$; to do so, we shall assume $t \in \hat{T}$, $s \in {}^\bullet t\backslash\hat{S}$ (see Figure 26) and construct a contradiction.

We claim that from an $\hat{N}$-maximal marking which enables $t \in \hat{T}$, t can be reproduced by occurrences of $\hat{T}$-transitions only! Indeed, let

$$M_0 \ldots M[t\rangle M'[\ldots\tau\ldots\rangle M''[t\rangle M'''$$

be a sequence such that M is $\hat{N}$-maximal and τ contains no T^{out}-transitions (and consequently no T^{in}-transitions). Suppose that τ is of the form

$$\tau \;=\; \ldots t't'' \ldots$$

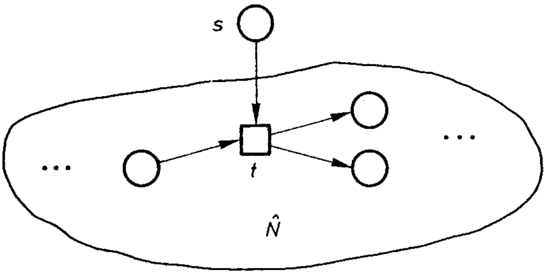


Figure 26: Illustrating the proof that $\hat{N}$ is generated by $\hat{T}$

where $t' \in \hat{T}$ and $t'' \in T\backslash\hat{T}$. We claim that $t'^{\bullet} \cap {}^{\bullet}t'' = \emptyset$. To see this, suppose $s' \in t'^{\bullet} \cap {}^{\bullet}t''$; by $t' \in \hat{T}$ and $t'^{\bullet} \subseteq \hat{S}$ we have $s \in \hat{S}$, but then by $t'' \notin \hat{T}$, we have $t'' \in T^{out}$, contradicting our assumption that τ contains no T^{out}-transitions. Hence $t'^{\bullet} \cap {}^{\bullet}t'' = \emptyset$, and lemmata 2.9(b) and 2.6 can be used to show that t' and t'' can be exchanged in τ. Repeating this, if necessary, the above sequence can be rearranged as follows:

$$M_0 \ldots M[\ldots \tau_1 \ldots \rangle M^1[t\rangle M^2[\ldots \tau_2 \ldots \rangle M''[t\rangle M''',$$

such that τ_1 contains only transitions from $T\backslash\hat{T}$ (even from $T\backslash(\hat{T} \cup T^{in} \cup T^{out})$, i.e. transitions that have nothing to do with $\hat{N}$) and τ_2 contains only transitions from $\hat{T}$.

Now assume that $t \in \hat{T}$ has an input place s which is not in $\hat{S}$, as in Figure 26. Then $M^1(s) = 1$ since M^1 enables t, and $M^2(s) = 0$ since s cannot also be an output place of t (since $t^{\bullet} \subseteq \hat{S}$ and $s \notin \hat{S}$). But $M''(s) = 1$ again because M'' also enables t. Hence in τ_2 some transition $t' \in {}^{\bullet}s$ occurs, but t' cannot be in $\hat{T}$ since $\hat{T}^{\bullet} \subseteq \hat{S}$. This gives a contradiction to the fact that τ_2 contains only $\hat{T}$-transitions, showing that the assumption ${}^{\bullet}t \not\subseteq \hat{S}$ is false. Hence (2) is also proved.

This completes the proof of the main theorem. We may remark that on two occasions (namely in the proof of the fact that $\hat{T}$-transitions do not decrease $\hat{n}(M)$ and in the proof that $\hat{N}$ is a subnet) we have used the FC property in a strict way. That is to say, if only the EFC property is assumed in place of the FC property, then the proof does not go through. As a matter of fact, construction 7.4 does not always produce the desired results for EFC systems. The interested reader may wish to find a counterexample and a modification to the algorithm which works for EFC nets as well.

It should be mentioned that M.Hack also proves a dual of the above theorem, namely that an LSFC system can be covered by S-components which carry exactly one token each (see section 8.2 below). Furthermore, he shows that theorem 4.8, i.e. the necessary and sufficient condition for the existence of a live and 1-safe marking in a T-net, can be generalised. Also, he shows that an FC net N has a live and 1-safe marking if and only if its reverse-dual net (i.e. the net $(N^{-1})^d = (N^d)^{-1}$ [6]) has a live and 1-safe marking. For the proof of these additional results, the reader is referred to [21,22,14].

In [23], M.Hack shows how some of his constructions can be generalised to EFC nets and to ESMA nets, which are another generalised class of nets; for the latter class, [26] can also be consulted. Further generalisations are described by G.Memmi [29,30] and by W.Griese [20]. These generalisations illustrate various essential aspects of the free choice property.

Finally, we mention that the free choice property (or a close analogon thereof) has been translated into other formalisms of concurrent systems, notably to COSY by M.W.Shields [40] and to FIFO nets by A.Finkel [16].

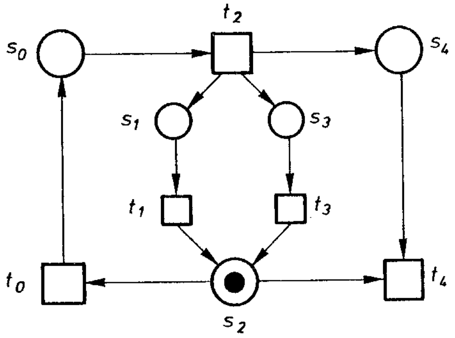


Figure 27: A live AC system which does not satisfy the dt-property

8 Some further results

The proof of section 7 was given in detail in order to explain the kind of reasoning employed in structure theory (of free choice nets). Once the reader is acquainted with this proof then, hopefully, he will find it easier to understand (the proofs of) the various other results that exist about free choice nets (and other net classes). We will give a selection of such other results in this section. They will only be explained briefly, and their proofs will be omitted.

8.1 A liveness criterion

An exact condition for the liveness of an FC system is due to F.Commoner and M.Hack [11,21]. A proof is also in [38] and, partly, in [26]. The theorem states that an FC system $\Sigma = (S, T; F, M_0)$ is live if, and only if it has the so-called dt-property, i.e., by definition, every 'deadlock' $S_0 \subseteq S$ contains a trap $S_1 \subseteq S$ which is marked under M_0, i.e. $M_0(S_1) > 0$. S_0 being a 'deadlock' means that ${}^\bullet S_0 \subseteq S_0^\bullet$; S_1 being a trap means that $S_1^\bullet \subseteq {}^\bullet S_1$. (We put 'deadlock' in quotes here since it is not a satisfactory term.)

F.Commoner has show in [11] that the dt-property is even a sufficient condition for the liveness of AC systems. However, conversely, it is not a necessary condition: Figure 27 shows an AC system which is live, even though the 'deadlock' $S_0 = \{s_0, s_1, s_2, s_3\}$ contains no marked trap (as the reader is invited to check). As an exercise, the reader may also wish to find out why the Commoner/Hack theorem reduces to theorem 4.3 for the special case of T-systems (a hint is that, according to the strict definition, no net may have isolated places).

8.2 A safeness criterion for live FC systems

In [21], M.Hack has proved a generalisation of theorem 4.4 as well. This states that a live FC system is 1-safe iff it is covered by S-components which carry exactly one token each. The concept of an S-component is defined dually to that of a T-component; for a definition, the reader is referred to [6]. In this safeness criterion, the S-components replace the simple cycles of theorem 4.4.

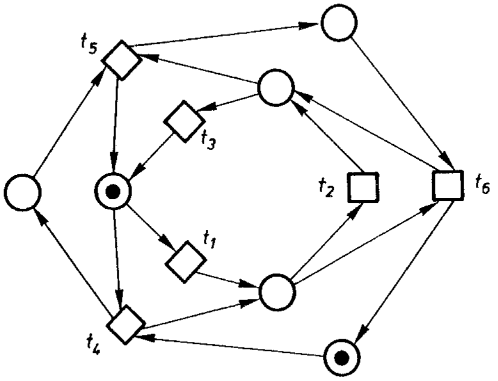


Figure 28: A live and 1-safe AC system violating the promptness theorem

8.3 A monotonicity result

In [15], K.Döpp has proved that a number of properties of an FC system $\Sigma = (S, T; F, M_0)$ do not change if its initial marking is increased by adding tokens to it, that is, by considering the new FC system $\Sigma' = (S, T; F, M_0')$ where $M_0' \geq M_0$. In particular, if Σ is live then so is Σ'. Also, if Σ is deadlock-free then so is Σ'. Furthermore, if Σ has certain reachability properties the so does Σ'. See [15] for the details. In general nets, this monotonicity property is false: adding a token to a live system may well 'kill' it; the reader may find examples by himself, or consult [38].

8.4 Promptness

Of the many nice results of [41], we shall mention only a few here. Promptness (of a system Σ with respect to a set of transitions T_0) means that the system may not go on working indefinitely without using transitions from T_0. T_0 may be thought of as a set of 'external' transitions of Σ; then Σ is prompt relative to T_0 if there is some maximal length of those behaviours that consist of 'internal' occurrences only.

Theorem 4.1 of [41] shows that an LSFC system Σ is prompt relative to T_0 iff T_0 has a non-empty intersection with every strongly connected T-component of Σ. This means that the behaviour of Σ is, in a sense, generated by its strongly connected T-components. Figure 28 shows that the theorem is not true for AC systems: with $T_0 = \{t_1, t_2, t_3\}$, no transition sequence can have two non-T_0-elements in succession, yet there is a T-component (namely that generated by $\{t_4, t_5, t_6\}$) which does not intersect with T_0.

The promptness theorem can be interesting when T_0 is interpreted as the 'interface' of the system to its environment.

8.5 A containment property

In section 5 of [41], it has been proved that every strongly connected subnet $N_1 = (S_1, T_1; F_1)$ of an LSFC system $\Sigma = (S, T; F, M_0)$ which is a T-net by itself (that is, N_1 satisfies $\forall s \in S_1 \colon |{}^\bullet s \cap T_1| \leq 1 \geq |s^\bullet \cap T_1|$) is contained in some strongly connected T-component of Σ. This result also follows directly from the arguments given in section 7 of these notes, since any strongly connected subnet which is also a T-net *could* be the result of an intermediate step of construction 7.4.

8.6 Home states

A marking $\hat{M}$ is a home state of a system $\Sigma = (S, T; F, M_0)$ if for all $M \in [M_0\rangle$: $\hat{M} \in [M\rangle$; that is, $\hat{M}$ always remains reachable. If the reachability graph of Σ is strongly connected then every marking is a home state. On the other hand, systems whose reachability graph is not strongly connected could still have home states; Figure 21(i) shows an example which is even an LSFC system.

[9] prove that an LSFC system always has at least one home state. The proof makes essential use of the T-component covering proved in section 7. It is a challenging exercise for the reader to prove that the theorem does not hold in general, i.e. to construct a live and 1-safe system which has no home states. K.Voss has even found an AC system with this property [9]. Home states may be useful in protocol validation [1].

8.7 Fairness

[41] and [4] show that in FC systems and in AC systems, fairness considerations are greatly simplified. Section 6 of [41] shows that global fairness of an LSFC system can be achieved locally, i.e. by taking care that every local conflict is resolved fairly. [4] shows that in AC systems (and a fortiori, in FC systems), there is no notion of 'proper conspiracy' (of, say, two processes against a third one), such as it may occur in the well-known 'five philosophers' example. The details of these results will be omitted here.

Acknowledgements

I wish to thank Ulrich Grude, Claudia Toussaint, P.S.Thiagarajan, K.Döpp and Wolfgang Reisig for giving me helpful comments on the structure and the contents of this manuscript. I am sincerely grateful to Frau Elisabeth Münch for producing the excellent drawings.

References

[1] G.Berthelot and R.Terrat: Petri Nets for the Correctness of Protocols. IEEE Trans. Comm. 30, 2497-2505 (1982).

[2] E.Best: Adequacy Properties of Path Programs. TCS Vol.18, 149-171 (1982).

[3] E.Best: COSY: its Relation to Nets and to CSP. These Notes.

[4] E.Best: Fairness and Conspiracies. IPL Vol.18, 215-220 (1984).

[5] E.Best and R.Devillers: Concurrent Behaviour: Sequences, Processes and Programming Languages. Studien der GMD No.99 (1985). A revised version of this report is due to appear in TCS (1987).

[6] E.Best and C.Fernández: Notations and Terminology on Petri Net Theory. Arbeitspapiere der GMD No.195 (1986). Also: Petri Net Newsletters No.23, 21-46 (April 1986).

[7] E.Best and M.W.Shields: Some Equivalence Results on Free Choice Nets and Simple Nets, and on the Periodicity of Live Free Choice Nets. Springer Lecture Notes in Computer Science Vol.159, 141-154 (1983).

[8] E.Best and P.S.Thiagarajan: (Forthcoming paper.)

[9] E.Best and K.Voss: Free Choice Systems have Home States. Acta Informatica 21, 89-100 (1984).

[10] S.D.Brookes and W.C.Rounds: Behavioural Equivalence Notions Induced by Programming Logic. Springer Lecture Notes in Computer Science Vol.154, 97-108 (1983).

[11] F.Commoner: Deadlocks in Petri Nets. Report, Applied Data Inc., CA-7206-2311 (1972).

[12] F.Commoner, A.W Holt, S.Even and A.Pnueli: Marked Directed Graphs. JCSS Vol.5, 511-523 (1971).

[13] J.Desel: (Forthcoming paper in Petri Net Newsletters.)

[14] K.Döpp: Zum Hack'schen Wohlformungssatz für Free-Choice-Petrinetze. EIK 19/1-2, 3-15 (1983).

[15] K.Döpp: Ein Satz über Free-Choice-Petrinetze. EIK 19/3, 107-113 (1983).

[16] A.Finkel: Boundedness and Liveness for Monogenous FIFO Nets and for Free Choice FIFO Nets — Applications to the Analysis of Protocols. Univ. Paris-Sud, L.R.I. Report No.205 (1985).

[17] H.J.Genrich and K.Lautenbach: Synchronisationsgraphen. Acta Informatica Vol.2, 143-161 (1973).

[18] H.J.Genrich, K.Lautenbach and P.S.Thiagarajan: Elements of General Net Theory. Springer Lecture Notes in Computer Science Vol.84, 21-163 (1981).

[19] H.J.Genrich and P.S.Thiagarajan: A Theory of Bipolar Synchronisation Schemes. TCS Vol.30, 241-318 (1984).

[20] W.Griese: Liveness in NSC Nets. In: Discrete Structures and Algorithms (ed. U.Pape), Carl Hanser Verlag, Munich, 256-264 (1980).

[21] M.Hack: Analysis of Production Schemata by Petri Nets. TR-94, MIT-MAC (1972).

[22] M.Hack: Corrections to MAC-TR-94. Computation Structure Notes 17, MIT-MAC (1974).

[23] M.Hack: Extended State-Machine Allocatable Nets, an Extension of Free Choice Petri Net Results. Computation Structures Group Memo 78-1, MIT-MAC (1974).

[24] D.Hillen: Relationship between Deadlock-freeness and Liveness in Free Choice Nets. Petri Net Newsletters No.19, 28-32 (1985).

[25] A.W.Holt: State Machines and Information. MIT-MAC Report (1970).

[26] M.Jantzen and R.Valk: Formal Properties of Place/Transition-Nets. Springer Lecture Notes in Computer Science Vol.84, 165-212 (1981).

[27] R.Johnsonbaugh and T.Murata: Additional Methods for Reduction and Expansion of Marked Graphs. IEEE Tr. on Circuits and Systems, Vol.28/10, 1009-1014 (1981).

[28] K.Lautenbach: Linear Algebraic Techniques for Place/transition Nets. These Notes.

[29] G.Memmi: Fuites et graphes à choix non imposé dans les réseaux de Petri. 3ème coll. int. sur la programmation, Dunod-Paris (1978).

[30] G.Memmi: Leakage Notion. Springer Informatik-Fachberichte No.52, 172-177 (1982).

[31] R.Milner: A Calculus of Communicating Systems. Springer Lecture Notes in Computer Science Vol.92 (1980).

[32] H.Müller: Prompt and hangup-free simulation of place/transition nets by pure nets without multiple arcs. Petri Net Newsletters No.15, 16-21 (October 1983).

[33] M.Nielsen and P.S.Thiagarajan: Degrees of Nondeterminism and Concurrency: A Petri Net View. DAIMI PB-180, University of Århus (1984). Also: 4th Conf. on Foundations of Software Technology and Theoretical Computer Science, Springer Lecture Notes in Computer Science, 89-117 (1984).

[34] D.Park: Concurrency and Automata on Finite Sequences. Computer Science Department, University of Warwick (1981).

[35] C.A.Petri: Nonsequential Processes. GMD-ISF Report 77.05 (1977).

[36] L.Pomello: Some Equivalence Notions for Concurrent Systems: An Overview. Arbeitspapiere der GMD No.103 (1984). Also: Springer Lecture Notes in Computer Science Vol.222, 381-400 (1985).

[37] L.Priese: Automata and Concurrency. TCS Vol.25(3), 221-265 (1982).

[38] W.Reisig: Petri Nets — An Introduction. Springer EATCS Monographs (1985).

[39] W.Reisig: Place/transition Systems. These Notes.

[40] M.W.Shields: On the Nonsequential Behaviour of Systems Possessing a Generalised Free Choice Property. Report CRS-92-81, Edinburgh University (1981).

[41] P.S.Thiagarajan and K.Voss: A Fresh Look at Free Choice Nets. Information and Control, Vol.61/2, 85-113 (1984).

[42] K.Voss: System Specification with Labelled Nets and the Notion of Interface Equivalence. Arbeitspapiere der GMD No.211 (June 1986).