Infinite Behaviour and Fairness.

Rüdiger Valk

revised version from

W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part I, Proceedings of an Advanced Course, Bad Honnef, September 1986*, volume 254 of *Lecture Notes in Computer Science*, pages 377-396. Springer-Verlag, 1987.

INFINITE BEHAVIOUR AND FAIRNESS

Rüdiger Valk
Fachbereich Informatik, Universität Hamburg
Rothenbaumchaussee 67-69, D-2000 Hamburg 13

ABSTRACT Specification techniques for the infinite behaviour of a P/T-system are introduced and their expressive power is compared. In particular, fair behaviour is related to the wellknown liveness problem. Finally, the problem is studied how to implement fair behaviour using a fair occurrence rule for transitions.

Key words: infinite behaviour, Landweber-hierarchy, fairness, liveness, resource allocation, fair and productive occurrence rule, finite delay property.

CONTENTS

1. Introduction

Petri nets are frequently used to model systems performing infinite processes like operating systems, real time control devices, communication protocols or information systems. Properties and problems concerning their behaviour should therefore be studied by a propriate model of infinite processes. Serial process models (transition sequences) have many advantages over partial order processes, which is in particular true for notions of fairness.

Liveness and fairness are important properties of systems. We will study their differences as well as their interconnections. While liveness is a commonly accepted notion - at least in net theory -, fairness has been studied less frequently. Nevertheless, many properties of real systems are based on elementary notions of fairness. In many cases, however, such fairness properties are either not mentioned explicitly or identified by completely different notions.

The classical example for a system where liveness and fairness properties are introduced are the celebrated Five Dining Philosophers [Dijkstra 71] (a formulation of the problem can be found in [Valk 83, p. 322]). Fig. 1.1 without dashed lines gives a P/T-system called FDPh1, modelling some interpretation of the Five Dining Philosophers' system. This net is not live: a deadlock occurs if all philosophers take their right-hand side fork (e.g. after the occurrence of $a_1a_2a_3a_4a_5$). The net becomes live if picking up both forks is implemented as an atomic action. This is represented by replacing a_i and a_i' by the dashed transition $pick_i$ for all $i \in \{1,\dots,5\}$. The resulting P/T-system will be referenced by FDPh2 . An alternative way to avoid deadlocks consists in supplementing FDPh1 by a common reading room where all philosophers are in their

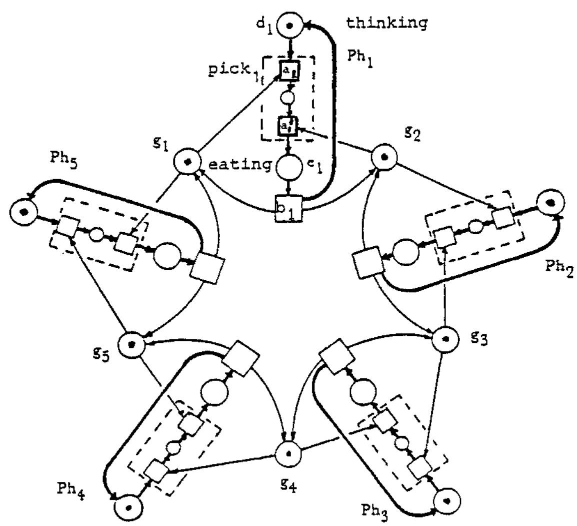


Fig. 1.1: Five philosophers P/T-system: FDPh1 , abstraction: FDPh2

initial state. A door-keeper allows at most four philosophers to be in the dining room at the same time. This extension, which is not given here, will be referenced by PDPh3 .

The behaviour of a P/T-system can be defined either as the set of <u>all</u> infinite occurrence sequences (*section 2*) or as some particular subset of this set. For instance, one might be interested in the behaviour, where philosopher 1 never eats for the last time. Such specification techniques are described in *section 3*.

In FDPh2 two philosophers can behave unfair against a third one (e.g. Ph_1 and Ph_3 against Ph_2) by picking up their forks in such a way that the latter never sees both forks simultaneously on the table (e.g. g_2 and g_3 are never marked at the same time). As a consequence transition $pick_2$ of FDPh2 cannot occur. Motivated by this example, the problem of achieving fair behaviour is also called the "problem of starvation". Using the definitions of *section 3* the fair behaviour of this system could be defined as the set of all infinite occurrence sequences, where all places d_i ("thinking") and e_i ("eating") for $1 \leq i \leq 5$ infinitely often contain a token.

Such a globally fair behaviour cannot be implemented without assuming a particular ("local") occurrence rule for transitions. By the "fair occurrence rule" (f-rule) no transition can be enabled infinitely often without occurring. The problem of implementing the fair behaviour of P/T-systems under the f-rule will be studied in *section 5*. Assuming this occurrence rule, we obtain for our example:

a) FDPh1 behaves fair but is not live

b) FDPh2 is live but behaves unfair

c) FDPh3 behaves fair and is live.

The tight relationship of the liveness and fairness problem shown by this example will be discussed in *section 4*.

2. Infinite behaviour of P/T-systems

Let $\Sigma=(S,T,F,W,m_o)$ be a Place/Transitions system (P/T-system), where (S,T,F) is a net, $W:F \to \mathbb{N}^+$ is a weight function and $m_o:S \to \mathbb{N}$ is an initial marking. (Throughout this paper we do not use finite capacities.)

By $F(\Sigma):=\{w \in T^* \mid \exists m: m_o \xrightarrow{w} m\}$ we denote the set of (finite) occurrence sequences or free Petri net language. If M_E is a finite set of (terminal) markings, then $F(\Sigma,M_E):=\{w \in T^* \mid \exists m \in M_E: m_o \xrightarrow{w} m\}$ is the free terminal Petri net language of (Σ,M_E). Assuming a labelling function $h:T \to X \cup \{\lambda\}$, then $L(\Sigma,h):=\{h(w) \mid w \in F(\Sigma)\}$ is the language of (Σ,h) and $L(\Sigma,M_E,h):=\{h(w) \mid w \in F(\Sigma,M_E)\}$ denotes the terminal language of (Σ,M_E,h). $L_{cyc}(\Sigma,h):=L(\Sigma,\{m_o\},h)$ is the cyclic language of Σ. A language is λ-free, if $h(t) \neq \lambda$ for all $t \in T$. Finally, by $\mathcal{L}^\lambda$, $\mathcal{L}$, $\mathcal{L}_o^\lambda$, $\mathcal{L}_o$ we denote the families of languages, λ-free languages, terminal languages and λ-free terminal languages of Petri nets.

We now come to the corresponding definitions for infinite occurrence sequences. To this end we first recall some definitions on infinite words. An infinite sequence or ω-word over an alphabet X is a mapping $w:\mathbb{N}^+ \to X$. For $i \in \mathbb{N}$ let $w(i)$ denote the i-th letter and $w[i]:=w(1)w(2)\dots w(i)$ the prefix of length $i \geq 1$ and $w[o]:=\lambda$ the empty word. $\tau_i(w):=w(i+1)w(i+2)\dots$ is the i-th truncation of w. As usual in literature, X^ω denotes the set of all ω-words and $X^\infty:=X^* \cup X^\omega$. A set $L \subset X^\omega$ is said to be an ω-language.

The concatenation on finite words is extended to words $u \in X^*$ and $v \in X^\omega$ by $uv:=w$ with $w[i]=u, \tau_i(w)=v$ and $i=lg(u)$. Also the finite product $v_1 v_2 \dots v_n = \prod_{i=1}^{n} v_i$ is extended to the infinite case by $w=\prod_{i=1}^{\infty} v_i$ iff $w[i]=v_1\ldots v_j$ with $i=lg(v_1)+\ldots+lg(v_j)$.

There are three classical operators that allow to define sets of infinite words from finite words:

a) for a language $L\subset X^*$ the <u>closure</u> or <u>ω-closure</u> is defined by $L^{\omega}:=\{w\in X^{\omega} \mid w=\prod_{i=1}^{\infty} w_i$ and $\forall i\in\mathbb{N}^+ : w_i\in L\}$

examples: $\{a\}^{\omega}=a^{\omega}=aaa..,ba^{\omega}=baa...$

b) for a language $L\subset X^*$ the <u>limit</u> of L [Eilenberg 74] is defined by $\lim(L):=\{w\in X^{\omega} \mid \exists_{\infty} i : w[i]\in L\}$ ($\exists_{\infty} i$ denotes: "there are infinitely many $i\in\mathbb{N}$ such that")

examples: $L_1=ba^*$ and $\lim(L_1)=ba^{\omega}$; $L_2=a^*b$ and $\lim(L_2)=\emptyset$

c) for a language $L\subset X^*$ the <u>adherence</u> of L [Boasson/Nivat 80] is defined by $Adh(L):=\{w\in X^{\omega} \mid \forall i\in\mathbb{N}^+ \exists v\in X^* : w[i]v\in L\}$

examples: $Adh(L_1)=ba^{\omega}$, $Adh(L_2)=a^{\omega}$

All these operators are interrelated. Define for $L\subset X^{\infty}$ the <u>prefix language</u> $Pref(L):=\{v\in X^* \mid \exists w\in X^{\infty} : vw\in L\}$. Then we have for arbitrary languages $L\subset X^*$: $Adh(L)=\lim(Pref(L))$, $L^{\omega}\subset \lim(L^*)$ and if L is prefixfree then $L^{\omega}=\lim(L^*)$. A particular subbehaviour is the <u>center</u> of a language $L\subset X^*$ [Boasson/Nivat 80] defined by $ctr(L):=Pref(Adh(L))$, which is the subset of those occurrence sequences of L that have an infinite continuation. In order to avoid deadlocks a system control has to allow only those occurrence sequences that belong to the center of L . For instance consider the P/T-system Σ of

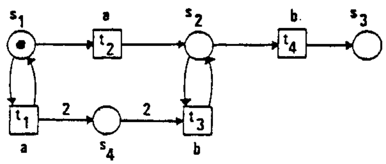


Fig. 2.1: A P/T-system with labelled transitions

Fig. 2.1 . The language of Σ is $L(\Sigma,h)=L_o=\{a^i b^j \mid i\geq j\geq o\}$. Since $Adh(L_o)=a^{\omega}$ the center of L_o is $ctr(L_o)=a^*$. In fact, whenever transitions t_2 of Σ occurs, a deadlock situation cannot be avoided furthermore.

By $F_{\omega}(\Sigma):=\{w\in T^{\omega} \mid \forall i : w[i]\in F(\Sigma)\}$ we denote the set of <u>infinite occurrence sequences</u> or <u>free ω-net-language</u> of Σ . $L_{\omega}(\Sigma,h):=\{h(w) \mid w\in F_{\omega}(\Sigma)\}$ is the <u>ω-language</u> or <u>ω-behaviour</u> of (Σ,h) . If $h(t)\neq\lambda$ for all $t\in T$ then the ω-language is <u>λ-free</u>. By $\mathcal{F}_{\omega}$, $\mathcal{L}_{\omega}$ and $\mathcal{L}_{\omega}^{\lambda}$ we denote the <u>families</u> of infinite occurrence sequences, λ-free ω-languages and ω-languages of P/T-systems, respectively.

To give an example for the P/T-system Σ of Fig. 2.1 we have $F_{\omega}(\Sigma)=t_1^{\omega}$ and $L_{\omega}(\Sigma,h)=a^{\omega}$.

<u>Lemma 2.1</u>

Let (Σ,h) be a P/T-system.

a) $F_\omega(\Sigma)=\lim(F(\Sigma))=\mathrm{Adh}(F(\Sigma))$

b) If h is λ-free then: $L_\omega(\Sigma,h)=\lim(L(\Sigma,h))=\mathrm{Adh}(L(\Sigma,h))$.

c) There are P/T-systems (Σ,h) with $L_\omega(\Sigma,h) \subsetneqq \lim(L(\Sigma,h))$

For the (easy) proofs (a) and (b) we refer to [Valk 83], but give a system satisfying (c). We redefine h of (Σ,h) in Fig. 2.1 by $h(t_1)=h(t_2)=\lambda$ and then obtain: $L_\omega(\Sigma,h)=\emptyset$ and $\lim(L(\Sigma,h))=b^\omega$.

We conclude this section by the following hierarchy of ω-behaviours, which is an analogon to the corresponding hierarchy of Petri net languages (see [Jantzen 86]).

Theorem 2.2

$$\mathcal{F}_\omega \subsetneqq \mathcal{L}_\omega \subsetneqq \mathcal{L}_\omega^\lambda \subsetneqq \mathrm{TYPEO}_\omega$$

(where TYPEO_ω is a family of ω-languages accepted by Turing machines).

In the proof of $\mathcal{L}_\omega \neq \mathcal{L}_\omega^\lambda$ we again use the system (Σ,h) of Fig. 2.1 . First we add a new transition t_5 with $W(s_3,t_5)=W(t_5,s_3)=1$ and redefine h by $h'(t_1)=h'(t_2)=h'(t_4)=\lambda, h'(t_3)=a$ and $h'(t_5)=b$. Then we obtain a P/T-system (Σ',h') with $L:=L_\omega(\Sigma',h')=a^*b^\omega$. Since $a^\omega \in \mathrm{Adh}(L)-L$ the ω-language L has not property (b) of Lemma 2.1 and therefore does not belong to $\mathcal{L}_\omega$. To prove $\mathcal{L}_\omega^\lambda \neq \mathrm{TYPEO}_\omega$ we define $L:=\{a^ib^ic \mid i\geq 1\}$. Then any reasonable definition of TYPEO_ω contains L^ω . With $v_i:=a^ib^ic$ the ω-word $v:=\prod_{i=1}^{\infty} v_i$ belongs to L^ω . Assume $v\in L_\omega(\Sigma,h)$ for some P/T-system (Σ,h) . Then the sequence of markings $m_o,m_1,m_2,\ldots$ where m_j is reached after the occurrence of $(\prod_{i=1}^{j} v_i)a^{j+1}$ contains two elements $m_r \leq m_s$ with $r<s$. But then also the ω-word $v':=(\prod_{i=1}^{s} v_i)(a^{s+1}b^{r+1})cu$ with $u=\prod_{i=r+2}^{\infty} v_i$ is in L^ω , which is impossible since $r\neq s$.
This last proof can be illustrated by showing the reader/writer-problem with unbounded numbers of readers to be not solvable by P/T-systems.

3. Specification of infinite behaviour

As mentioned in the introduction the fair behaviour of the five philosophers' P/T-system FDPh1 in Fig. 1.1 is a proper subset of the ω-behaviour, e.g. $(a_1a_1'b_1)^\omega \in F_\omega(\mathrm{FDPh1})$ is not fair. In order to specify this subset we could say that a sequence is fair if each of the places $e_1,\ldots,e_5$ infinitely often contains a token. The following definition will allow to specify such behaviour formally.

Let be $m=m(1)m(2)\ldots \in M^\omega$ be an infinite sequence. (You can imagine that m is a sequence of 'states'). Then the infinity set $\mathrm{In}(m):=\{m_1 \mid \exists_\infty i: m(i)=m_1$ is the set of elements of m occurring infinitely often.

Now we assume a finite set $\mathcal{A}=\{A_1,\dots,A_k\}$ of finite, nonempty subsets $A_i \subset M$ $(1\leq i\leq k)$ to be given. The m is called

a) <u>1-successful</u> or <u>touching</u> for $\mathcal{A}$, iff $\exists A\in\mathcal{A}\ \exists i\in\mathbb{N}^+ : m(i)\in A$

b) <u>1'-successful</u> or <u>completely enclosed</u> for $\mathcal{A}$, iff $\exists A\in\mathcal{A}\ \forall i\in\mathbb{N}^+ : m(i)\in A$

c) <u>2-successful</u> or <u>repeatedly successful</u> for $\mathcal{A}$, iff $\exists A\in\mathcal{A} : In(m)\cap A\neq\emptyset$

d) <u>2'-successful</u> or <u>eventually enclosed</u> for $\mathcal{A}$, iff $\exists A\in\mathcal{A} : \emptyset\neq In(m)\subset A$

e) <u>3-successful</u> or <u>eventually terminal</u> for $\mathcal{A}$, iff $\exists A\in\mathcal{A} : In(m)=A$

f) <u>4-successful</u> or <u>eventually containing</u> for $\mathcal{A}$, iff $\exists A\in\mathcal{A} : A\subset In(m)$

The sets $A_i\in\mathcal{A}$ are called <u>anchor sets</u> since the sequence m is attached in some sense to them. The definitions of "i-successful sequences" come from [Landweber 69] for $i\in\{1,1',2,2',3\}$. For practical reasons we add the case i=4 . For given anchor sets we have the following implications [Carstensen 86]:

$$\begin{array}{ccccc} \text{m is 4-successful} & \Rightarrow & \text{m is 2-successful} & \Rightarrow & \text{m is 1-successful} \\ \Uparrow & & \Uparrow & & \Uparrow \\ \text{m is 3-successful} & \Rightarrow & \text{m is 2'-successful} & \Rightarrow & \text{m is 1'-successful} \end{array}$$

We will apply the notion of successful sequence in three different ways. First, for a given P/T-system Σ we take $m=m_o m(1)m(2)\dots$ as infinite sequence of markings occurring with an infinite sequence $w\in F_\omega(\Sigma)$ of transitions. Here the anchor sets are finite sets of markings. Hence, given such a set $\mathcal{A}$ we define for $i\in\{1,1',2,2',3,4\}$

> $L^i_w(\Sigma,h,\mathcal{A}):=\{h(v)\in X^\omega \mid v\in F_\omega(\Sigma)$ and v occurs with a corresponding sequence of markings that is i-successful for $\mathcal{A}\}$. h is omitted if h=id is the identity map on X=T .

The <u>families</u> of these i-behaviours are denoted by $\mathcal{L}^i_\omega$. i-behaviours $\mathcal{R}^i_\omega$ for $i\in\{1,1',2,2',3\}$ of finite automata (or bounded P/T-systems) were shown in [Landweber 69] to correspond to the Borel hierarchy for a topology on X^ω .

As an exercise we specify the fair behaviour of the P/T-system FDPh1 in Fig. 1.1 as 3-behaviour. To find an appropriate set of anchor set we first define for arbitrary places p_i the following subset of reachable markings $R(\Sigma)$:

> $A(p_1,\dots,p_k):=\{A\subset R(\Sigma) \mid \forall i\in\{1,\dots,k\}\exists m\in A: m(p_i)>o\}$.

To specify that no philosopher ever eats for the last time (i.e. $e_1,\dots,e_5$ are infinitely often marked), we define $\mathcal{A}:=A(e_1\dots,e_5)$. Now the corresponding 'fair' behaviour is given by $L^3_\omega(\text{FDPh1},\mathcal{A})$.

Let $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ be families of languages. $\mathcal{L}_3$ can also be a family of ω-languages. Then we define:

$$\mathcal{L}_1 \circ \mathcal{L}_3:=\{\bigcup_{i=1}^k A_iB_i \mid A_i\in\mathcal{L}_1, B_i\in\mathcal{L}_3, k\in\mathbb{N}^+\}$$

$$\mathcal{L}_1 \circ_\omega \mathcal{L}_2:=\{\bigcup_{i=1}^k A_iB_i^\omega \mid A_i\in\mathcal{L}_1, B_i\in\mathcal{L}_2, k\in\mathbb{N}^+\}$$

Using this notation we are able to characterize i-behaviours by the following theorem, where $\mathcal{L}_{cyc}$ denotes the family of cyclic languages of P/T-systems.

<u>Theorem 3.1</u>

The families of i-behaviours with i$\in$ 1,1',2,2',3,4 are closed under finite union and

(a) $\mathcal{L}^{1}_{\omega} = \mathcal{L}_{o} \circ \mathcal{L}_{\omega}$

(b) $\mathcal{L}^{1'}_{\omega} = \mathcal{R}^{1'}_{\omega}$

(c) $\mathcal{L}^{2}_{\omega} = \mathcal{L}_{o} \circ_{\omega} \mathcal{L}_{cyc}$

(d) $\mathcal{L}^{2'}_{\omega} = \mathcal{L}_{o} \circ \mathcal{R}^{1'}_{\omega}$

(e) $\mathcal{L}^{3}_{\omega} = \mathcal{L}_{o} \circ \mathcal{R}^{3}_{\omega}$

The characterization of $\mathcal{L}^{2}_{\omega}$ in (c) expresses that the 2-behaviour of a P/T-system is the finite union of terminal languages of P/T-systems followed by an iteration of a cyclic language. In (d) and (e) the cyclic language is replaced by ω-behaviours of finite automata. Besides such intuitive interpretation the characterization was shown to be useful for proving the following hierarchy [Valk 83]. It was also used in [Pelz 86] for deriving a characterization of $\mathcal{L}^{3}_{\omega}$ by logical formulas.

<u>Theorem 3.2</u>

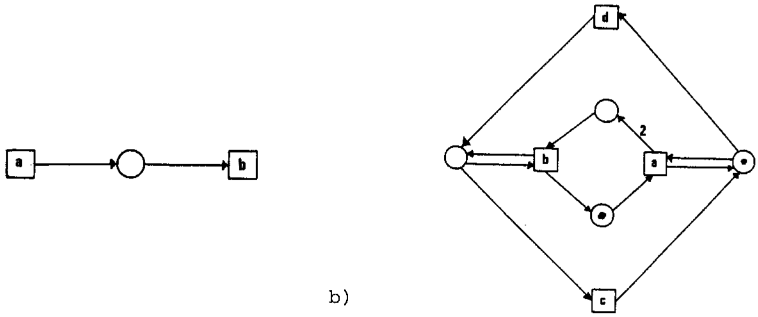
The hierarchy of i-behaviours $\mathcal{L}^{i}_{\omega}$ in Fig. 3.1 is strict and complete, i.e. there are no other inclusions.

$$
\begin{tikzcd}
\mathcal{L}^{4}_{\omega} & & & & & & & \\
\mathcal{L}^{2}_{\omega} \arrow[u] & & & \mathcal{K}^{2}_{\omega} & = & \mathcal{K}^{3}_{\omega} & = & \mathcal{K}^{4}_{\omega} \\
\mathcal{L}^{3}_{\omega} \arrow[u] & \mathcal{L}^{1}_{\omega} & & \mathcal{K}^{1}_{\omega} \arrow[u] & = & \mathcal{K}^{2'}_{\omega} & & \\
\mathcal{L}^{2'}_{\omega} \arrow[u] \arrow[ur] & \mathcal{L}_{\omega} \arrow[u] & = & \mathcal{K}^{1'}_{\omega} \arrow[u] & & & & \\
\mathcal{L}^{1'}_{\omega} \arrow[u] \arrow[ur] & & & & & & &
\end{tikzcd}
$$

Fig. 3.1: The hierarchy of i-behaviours

To give some idea of the proof for theorem 3.2, the reader is invited to show that $L_1 := L^2(\Sigma, \{\underline{O}\}) \notin \mathcal{L}^{3}_{\omega_3}$, where Σ_1 is the P/T-system of Fig. 3.2(a) and $\underline{O}$ the zero-marking, hence $\mathcal{L}^{2}_{\omega} \not\subseteq \mathcal{L}^{3}_{\omega}$. For Σ_2 in Fig. 3.2(b) we have $L_2 := L_{\omega}(\Sigma_2) \notin \mathcal{L}^{2}_{\omega}$, from which $\mathcal{L}_{\omega} \not\subseteq \mathcal{L}^{2}_{\omega}$ [Valk 83]. The results concerning $\mathcal{L}^{4}_{\omega}$ are from [Carstensen 86].

Assume now that one philosopher, say philosopher 1, has priority over philosopher 2, i.e. at any time the number of occurrences of the event "start eating" (tran-



a) b)

Fig. 3.2: Two counterexample P/T-systems

sition a_2') of philosopher 2 does not exceed the corresponding number for philosopher 1 (transition a_1'). To model this extension we add a new place p_{12} and arcs (a_1',p_{12}) and (p_{12},a_2'). Since this new place is unbounded, the anchor set $\mathcal{A}$, just defined before, becomes infinite.

Therefore we introduce a new class of i-behaviours, where anchor sets only refer to bounded places and do not care on tokens in unbounded places. For the exact definition we refer to [Valk 83]. For a given set $\mathcal{A}$ the so-called <u>bounded i-behaviour</u> of a P/T-system Σ is denoted by $B^i_\omega(\Sigma,h,\mathcal{A})$ and the corresponding class by $\mathcal{B}^i_\omega$. A similar notion is used in [Merceron 86]. The hierarchy of bounded i-behaviours is different from the hierarchy of i-behaviours, but similar to the corresponding hierarchy for nondeterministic pushdown automata [Cohen/Gold 77]. In fact, bounded places correspond to states in the finite control of pushdown automata, whereas unbounded places are weak counters.

In applications, in particular when dealing with fairness, it may be profitable to base the specification of i-behaviour on <u>events</u> instead of markings. For instance, in a fair occurrence sequence of FDPh1 each <u>transition</u> of $\{a_1',\ldots,a_5'\}$ must appear infinitely often. For a given set $\mathcal{A}$ of sets $A_i \subset T$, a P/T-system $\Sigma=(S,T,F,W,m_o)$ and a map $h:T\to X$ for $i\in\{1,1',2,2',3,4\}$ the <u>transitional i-behaviour</u> of Σ is defined by $K^i_\omega(\Sigma,h,\mathcal{A}):=\{h(v)\in X^\omega \mid v\in F_\omega(\Sigma)$ and v is i-successful for $\mathcal{A}\}$. The corresponding <u>classes of transitional i-behaviour</u> are denoted by $\mathcal{K}^i_\omega$.

It is remarkable that the classes $\mathcal{K}^i_\omega$ and $\mathcal{B}^i_\omega$ are identical for all i [Carstensen/Valk 85]! Their relative power and relationship to the classes $\mathcal{L}^i_\omega$ is shown in Fig. 3.1 .

<u>Theorem 3.3</u>

$\mathcal{K}^i_\omega = \mathcal{B}^i_\omega$ for all $i\in\{1,1',2,2',3,4\}$.

4. Live and fair behaviour

All of the classes of i-behaviour, introduced in section 3, may specify the fair behaviour of a P/T-system under a suitable interpretation. We now considerably simplify the situation and concentrate on the behaviour $K^4_\omega(\Sigma,h,\mathcal{A})$ of a P/T-system where $\mathcal{A}$ = $\{E\}$ with $E\subset T$, i.e. the set of all infinite occurrence sequences where all transitions from E appear infinitely often (among others).

Given a P/T-system $\Sigma=(S,T,F,W,m_o)$ and a subset $E\subset T$, then the E-fair behaviour of Σ is defined by $\mathrm{Fair}_E(\Sigma)=K^4_\omega(\Sigma,h,\{E\})=\{w\in F_\omega(\Sigma)\mid E\subset \mathrm{In}(w)\}$. For $E=\emptyset$ we have $\mathrm{Fair}_E(\Sigma)=F_\omega(\Sigma)$ and if $E=T$, we obtain the fair behaviour of Σ : $\mathrm{Fair}(\Sigma):=\mathrm{Fair}_T(\Sigma)$. Σ is called E-fair and fair , if $F_\omega(\Sigma)=\mathrm{Fair}_E(\Sigma)$ and $F_\omega(\Sigma)=\mathrm{Fair}(\Sigma)$, respectively. Σ is fair, if there are no infinite occurrence sequences with finite occurrence numbers of some transitions.

If Σ is E-fair, then the initial marking m_o has the following property to be "E-continual". A marking m of a P/T-system $\Sigma=(S,T,F,w,m_o)$ is E-continual, if there is an infinite occurrence sequence $v\in T^\omega$ starting in m with $E\subset \mathrm{In}(v)$. m is said to be continual if m is $\emptyset$-continual, i.e. if there is at least one infinite occurrence sequence in $\Sigma_m:=(S,T,F,W,m)$, hence $F_\omega(\Sigma_m)\neq\emptyset$.

Since we have no finite capacities (or represent them with the aid of complementary places), the set CONTINUAL(E) is right-closed. A set $K\subset \mathbb{N}^S$ of vectors is right-closed if $m\in K$ and $m'\geq m$ ($\geq$ componentwise) implies $m'\in K$. The sets NOTDEAD of nondead markings and UNBOUNDED of unbounded markings are other examples of right-closed sets. The set of minimal elements of a right-closed set K is called the residue set $\mathrm{res}(K):=\{m\in K\mid (m'\in K\wedge m'\leq m)\Rightarrow m'=m\}$ of K . As shown in [Valk/Jantzen 85] res(K) is finite for all right-closed sets K and effectively computable for $K\in\{$CONTINUAL(E), NOTDEAD, UNBOUNDED$\}$.

This implies that for given P/T-system Σ and marking m it can be decided whether there is some infinite occurrence sequence $v\in T^\omega$ such that all $t\in E$ appear infinitely often. Furthermore from $\Sigma=(S,T,F,W,m_o)$ with $m_o\in K$ a system $\Sigma_K=(S,T',F',W',m_o)$ can be effectively constructed such that in Σ_K exactly those markings are reachable that are reachable in Σ when Σ does not leave K . Furthermore, the occurrence sequences of Σ_K are essentially those of Σ when restricted to K [Valk/Jantzen, Theorem 4.2]. Therefore Σ_K is called the K-restricted P/T-system.

To give an example, if K=CONTINUAL:=CONTINUAL($\emptyset$) , then Σ_K is deadlock-free and has exactly the deadlock-free subbehaviour of Σ . Hence, the K-restriction Σ_K of Σ can be seen as supplementing Σ with a control unit to implement a desired behaviour (here deadlock freeness). Consider for instance the P/T-system Σ in Fig. 4.1 representing the wellknown banker's problem of Dijkstra as given in [Brinch Hansen 73].

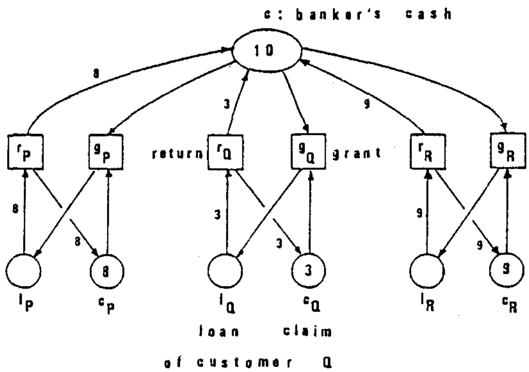


Fig. 4.1: The banker's system

The banker's problem was given in [Dijkstra 68] as an example of a resource sharing problem. A banker wishes to share a fixed capital c among a fixed number of n customers. The maximal claim of customer i is c_i . The banker will accept a customer if his claim does not exceed his initial capital, i.e. $c_i \leqq c$. The customers borrow the money unit by unit and return all the loan money some time after the maximal claim c_i was reached. Sometimes it may be necessary for a customer to wait before he can borrow another unit of money, but the banker guarantees that the customer will get the money after some finite time.

The problem for the banker is to avoid deadlock situations, i.e. states where neither the banker has any money himself nor any customer has get enough money to return his credit. There may be situations which are not deadlocks themselves, but unavoidably lead to a deadlock. Therefore to avoid deadlocks, it is not sufficient to look whether the next step leads to a deadlock directly. Dijkstra introduced the notion of 'safe' states. A state is safe if there is at least one continuation of actions from this state leading to a proper termination of all credit transactions. For the banker it is therefore important to know the safe states.

For the P/T-system of the banker's problem in Fig. 4.1 safe states are identical with T-continual markings. By the following four S-invariant equations of this P/T-system,

$$i_1: m(c) + m(l_P) + m(l_Q) + m(l_R) = 10$$
$$i_2: m(l_P) + m(c_P) = 8$$
$$i_3: m(l_Q) + m(c_Q) = 3$$
$$i_4: m(l_R) + m(c_R) = 9 ,$$

the knowledge of $(m(c_P), m(c_Q), m(c_R))$ is sufficient to uniquely describe any reachable marking m . Fig. 4.2 shows the set of reachable markings in this 3-dimensional representation. For instance, the initial marking is represented by the point (8,3,9). 24

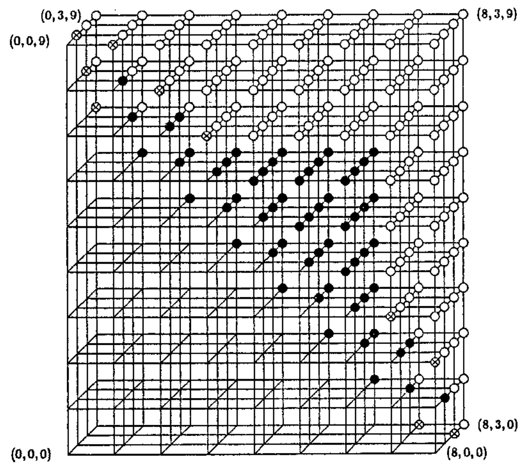


Fig. 4.2: T-continual markings of the banker's system

of these markings are deadlocks. The 137 T-continual markings are represented as white circles, which contain a cross if they are minimal. These ten minimal markings represent the residue res(K) of K=CONTINUAL(T) intersected with the reachability set. As shown in [Hauschildt/Valk 86] these 10 vectors can be obtained from only three vectors (0,1,9), (0,2,8), (0,3,7).

Hence the knowledge of three vectors is sufficient for the banker to avoid deadlocks by granting money in the wrong time. (Remember that T-continual markings are the 'safe' states of [Dijkstra 68]!) From this example we learn another aspect of fairness. The banker behaves fair against the customers by granting resources in such a way that every customer is able to perform his task!

In the introduction we saw that the properties of fairness and liveness are not independent for P/T-systems. We therefore will now have a closer look at these properties.

For any P/T-system $\Sigma=(S,T,F,W,m_o)$ and a marking m the *set of markings reachable from m* is denoted by $R(m):=\{m' \mid \exists w\in T^* \;\; m \xrightarrow{w} m'\}$. $R(\Sigma):=R(m_o)$ is the *reachability set* of Σ. A marking m of Σ is *live* if for any $t\in T$ and $m_1\in R(m)$ there is some $m_2\in R(m_1)$ such that t is enabled at m_2 . Σ is *live* if the initial marking m_o is live.

To compare this property of Σ with the property of fairness we also give a different but equivalent definition for liveness where $m \xrightarrow{w}$ denotes that w can occur from m :

(I) *Σ is live* or *behaves live* iff $\forall t\in T \forall m\in R(\Sigma) \exists w\in T^{\omega}$: $m \xrightarrow{w} \wedge t\in In(w)$

(II) *Σ is fair* or *behaves fair* iff $\forall t\in T \forall m\in R(\Sigma) \forall w\in T^{\omega}$: $m \xrightarrow{w} \Rightarrow t\in In(w)$

The difference between these notions becomes clear if you recognize that liveness means the "potential occurrence" of all transitions, whereas fairness denotes the "actual occurrence" of all transitions. To apply the construction of a K-restricted P/T-system it would be necessary that the set LIVE of all live markings is right-closed. This is, however, <u>not</u> the case as the P/T-system of Fig. 4.3(b) shows. The indicated initial marking m_o is live, but by adding a token to s_4 we obtain a marking m_1 that is not live such that $m_1 \geq m$!

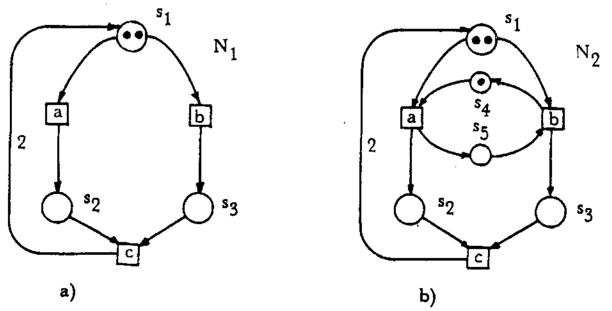


Fig. 4.3: Two P/T-systems

Obviously the system N_1 of Fig. 4.3(a) is not live, but there is a subset of $F(\Sigma)$ which is live (i.e. forbid two successive occurrences of a or b). Therefore it would be important to have a general procedure to construct a control for a P/T-system such that the system becomes live. To this end we mention the following relation between live and T-continual markings.

<u>Theorem 4.1</u>

For a P/T-system Σ a marking m is live if and only if all markings of R(m) are T-continual.

By this theorem we obtain the maximal live subbehaviour (see [Valk/Jantzen 85]) of a P/T-system, if we construct a control unit that disallows all occurrences of transitions that would transform a marking $m \in$ CONTINUAL(T) into a marking outside of this set! Hence for any P/T-system Σ with initial marking $m_o \in K$=CONTINUAL , the K-restricted system Σ_K is automatically live and has the "maximal live" subbehaviour of Σ . To be more precise, Σ_K is not live in the strict sense, since we introduce 'copies' of transitions. But if we define the same label h(t) for these copies, we can use the following variant of definition (I):

(III) a labelled P/T-system (Σ,h) where $h:T \to X$ is <u>live</u>, iff
$\forall x \in X \forall m \in R(\Sigma) \; \exists w \in T^{\omega}: \; m \xrightarrow{w} \wedge x \in \mathrm{In}(h(w))$

Using this definition and the result on K-restricted P/T-systems, we can formulate:

<u>Theorem 4.2</u>

For any P/T-system $\Sigma=(S,T,F,W,m_o)$ with $F_{\omega}(\Sigma) \neq \emptyset$ a live and labelled P/T-system (Σ',h) can be constructed such that $L(\Sigma',h) \subset F(\Sigma)$ is maximal with respect to the

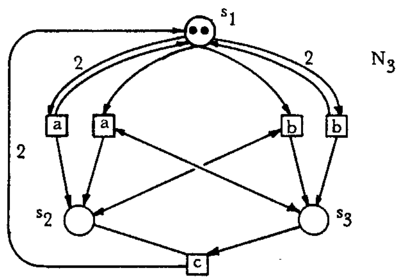


Fig. 4.4: A live labelled P/T-system

property of liveness.

Applying to the non-live P/T-system $\Sigma=N_1$ of Fig. 4.3(a), this construction gives the live labelled P/T-system $\Sigma'=N_3$ of Fig. 4.4 . Note that Σ' has the maximal live subbehaviour of Σ. This is not the case for Σ_1 in Fig. 4.3(b), since bac $\in L(\Sigma')$ but bac $\notin F(\Sigma_1)$.

5. Fair occurrence rules

In the preceeding sections we have studied the fair behaviour of P/T-systems. As in the case of live behaviour in section 4 we are interested to find some kind of control to implement fair behaviour. As discussed in the introduction this will be done by introducing two different fair occurrence rules for transitions. These occurrence rules may be thought of to be implemented in hardware or in the compiler of some programming languages. On the basis of fair occurrence rules the user of the hardware or the programmer will try to model systems or write programs that behave fair. In other words, fair processes are generated by fair systems.

Let $\Sigma=(S,T,F,W,m_o)$ be a P/T-system and $v\in F_\omega(\Sigma)$ an infinite occurrence sequence producing the infinite sequence $m_o, m_1, m_2, \ldots$ of markings.

v is productive if any transition $t\in T$ that is enabled permanently from some step on occurs infinitely often in v: $\forall t\in T(\exists_i \forall_{j\geq i} : m_j \xrightarrow{t} \Rightarrow t\in In(v))$

v is fair if any transition $t\in T$ that is enabled infinitely often occurs infinitely often in v: $\forall t\in T(\exists_\infty j : m_j \xrightarrow{t} \Rightarrow t\in In(v))$

A transition rule that disallows nonproductive and unfair occurrence sequences is called p-rule and f-rule, resp. The p-rule is also known as "finite delay property" and "just transition rule". To give some examples, $(d_1a_1a_1'b_1)^\omega$ is not productive in FDPh1 (Fig. 1.1) and $(ac)^\omega$ is productive but not fair in Fig. 5.1 .

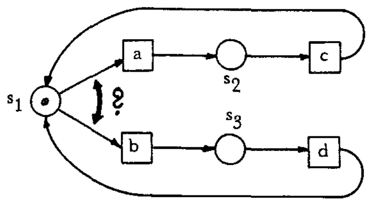


Fig. 5.1: P/T-system with unfair behaviour

Fair occurrence rules are sometimes assumed in unexpected contexts. For instance, in the "alternating-bit-protocol" the noisy half-duplex channel is modelled by two transitions, one for the correct transmission of data items (say, transition a) and another (say, transition b)) for producing an error message. The alternating-bit-protocol is not correct, when the ordinary transition rule is used. In fact, if transition b always occurs from some time one, the transmission of data stops. Hence the occurrences of errors are assumed to be 'fair', i.e. after some finite time the channel works correctly again. Therefore we have to assume the f-rule in all usual implementations of the alternating-bit-protocol.

By $L_\omega^{prod}(\Sigma,h):=\{h(v)\in X^\omega \mid v\in F_\omega(\Sigma) \text{ is productive}\}$ we denote the behaviour of a labelled P/T-system (Σ,h) under the p-rule and $L_\omega^{fair}(\Sigma,h):=\{h(v)\in X^\omega \mid v\in F_\omega(\Sigma) \text{ is fair}\}$. The corresponding families of languages are denoted by $\mathcal{L}_\omega^{prod}$ and $\mathcal{L}_\omega^{fair}$. Since every fair occurrence sequence is also productive, we have for any (Σ,h): $L_\omega^{prod}(\Sigma,h) \subset L_\omega^{fair}(\Sigma,h)$.

On the other hand, by a rather complicated construction in [Carstensen 82] it has been shown that every behaviour under the p-rule can be also obtained by another system under the f-rule:

<u>Theorem 5.1</u>

$$\mathcal{L}_\omega^{prod} \subsetneq \mathcal{L}_\omega^{fair}$$

A more important question is, however, whether every fair behaviour of a P/T-system can be generated by the p-rule. More precisely, given a P/T-system Σ, is there a (labelled) P/T-system (Σ_1,h) such that $L_\omega^{prod}(\Sigma',h) = Fair(\Sigma)$? This can be interpreted as follows: we assume a given system Σ and a specification of its fair behaviour $Fair(\Sigma) \subset F_\omega(\Sigma)$, but we have no idea how to implement it. On the other hand, we assume to have a programming environment that allows to run P/T-systems under the p-rule. Then the system (Σ',h) would be an implementation of Fair (Σ) in this environment.

Since Fair (Σ) is a particular case of transitional 4-behaviour, the problem is solved by the following theorem from [Carstensen/Valk 85]:

Theorem 5.2

$\mathcal{K}_\omega^4 \subset \mathcal{L}_\omega^{prod}$

By this theorem and other results from [Carstensen/Valk 85], the hierarchies of Fig. 3.1 obtain $\mathcal{L}_\omega^{prod}$ as common top element, i.e. we also have $\mathcal{L}_\omega^4 \subset \mathcal{L}_\omega^{prod}$ and $\mathcal{L}_\omega^1 \subset \mathcal{L}_\omega^{prod}$. The constructive proof of theorem 5.2 is, however, of less practical value, since the resulting P/T-system is not deadlock free in general, even if the original net has this property. Therefore we now restrict our investigations to deadlock free systems, for which a solution exists too.

A P/T-system Σ is deadlock free if in any reachable marking $m \in R(\Sigma)$ at least one transition is enabled. Equivalently all $m \in R(\Sigma)$ are continual, i.e. $R(\Sigma) \subset CONTINUAL(\emptyset)$. In our next theorem we have to use λ-transitions, i.e. transitions t with $h(t)=\lambda$ (that do not appear in the behavioral description of the system). But we require that after the occurrence of a λ-transition the next transition is λ-free. Such systems are called 1-prompt.

Theorem 5.3

For every P/T-system Σ having the fair behaviour $Fair_E(\Sigma)$ a deadlock free, 1-prompt and labelled system (Σ_1,h) can be effectively constructed that has ω-behaviour $Fair_E(\Sigma)$ under the p-rule: $Fair_E(\Sigma) = L_\omega^{prod}(\Sigma_1,h)$.

The proof in [Carstensen 84] is based on the result on residue sets. It is sufficient to choose $E=\{t\}$ and (since $\mathcal{K}_\omega^4 = \mathcal{K}_\omega^2$, see Fig. 3.1) to prove the theorem for $L= K_\omega^2(\Sigma,\{t\})$ instead of $Fair_E(\Sigma)$. Let $R=\{r_1,\dots,r_k\}$ be the residue set of CONTINUAL({t}) which is nonempty. Furthermore by the result on K-restricted P/T-systems in section 4 we can assume that $R(\Sigma) \subset CONTINUAL(\{t\})$. By definition of CONTINUAL({t})

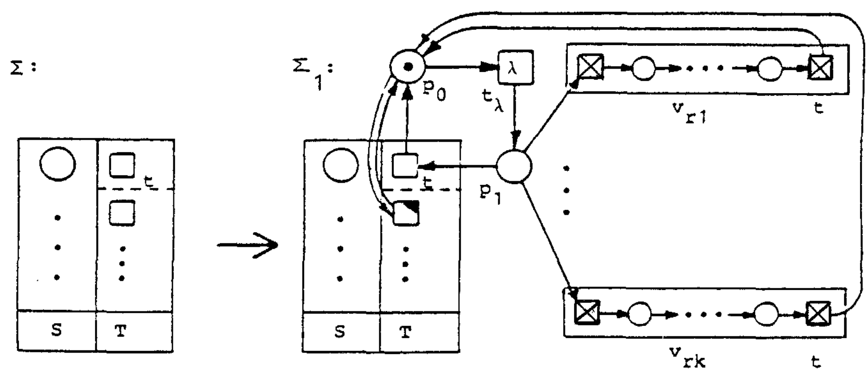


Fig. 5.2: Construction for theorem 5.3

for every marking $r \in R$, some finite occurrence sequence $v_r \in T^*$ can be computed with $\exists\, r' \in R$: $r \xrightarrow{v_r t} m$ and $m \geq r'$. In the transformation of Σ into Σ_1 in Fig. 5.2 copies (marked by a cross) of all $v_{r_1}t, \ldots, v_{r_k}t$ are introduced. Clearly, besides the additional arcs drawn in Fig. 5.2 all original arcs to places in S are assumed. All transitions of Σ with the exception of t have p_o as side condition. Hence they are blocked when p_o is empty. In Σ_1 every productive occurrence sequence v must contain infinitely often the λ-transition t_λ and hence some of the copies of t must also occur infinitely often. Moreover, every sequence in L is also in $L_\omega^{prod}(\Sigma_1, h)$ and vice versa.

To compare this general result with a wellknown special solution, we consider the fair implementation of distributed mutual exclusion of two processes A and B in

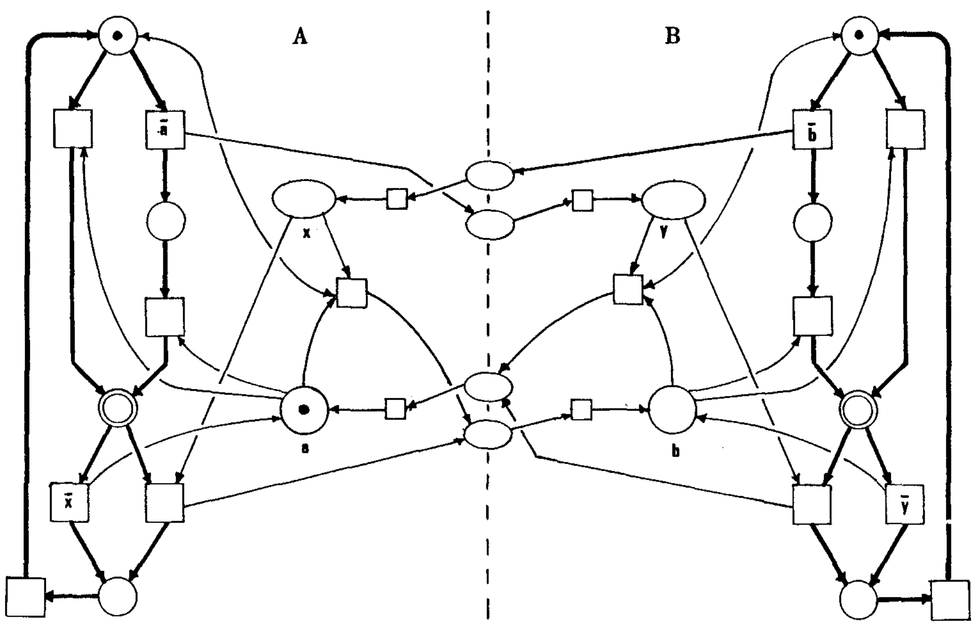


Fig. 5.3: Fair distributed mutual exclusion

in Fig. 5.3 . The critical section is specified by double circles. A and B communicate only by messages via channels on the dashed line. At most one process can have access to the critical region, namely the process that is owning the token in place a or b . To keep the net simple, transitions marked by $\bar{a}, \bar{b}, \bar{x}$ or $\bar{y}$ are assumed to be enabled only if the corresponding place is unmarked. This can be easily implemented by complementary places for a,b,x and y . In the given marking of Fig. 5.3 process A has access to the critical section, while process B has not. However, process B can fire the transition marked by $\bar{b}$, which sends a request to process A in place x . When leaving the critical section, process A will then pass the token to the place b instead of back to place a , and B has access, too. The net is modelled after an idea in [Raynal 85] modifying the solution of [Ricard/Agrawala 81].

Since the P/T-net of Fig. 5.3 behaves fair under the p-rule, it is of interest to compare it with the general construction of theorem 5.3 . To this end we start by the unfair distributed mutual exclusion of Fig. 5.4 .

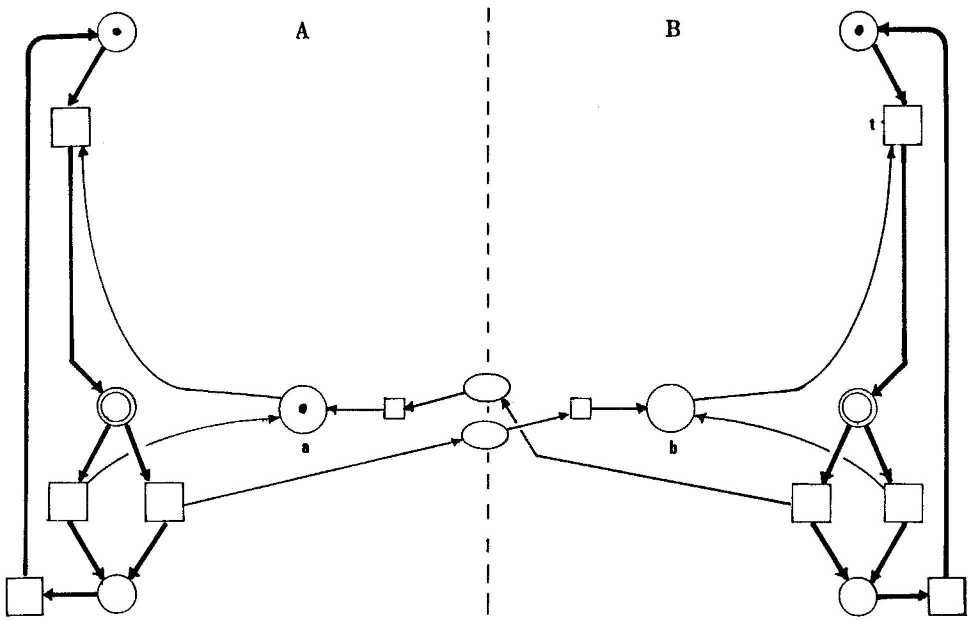


Fig. 5.4: Unfair distributed mutual exclusion

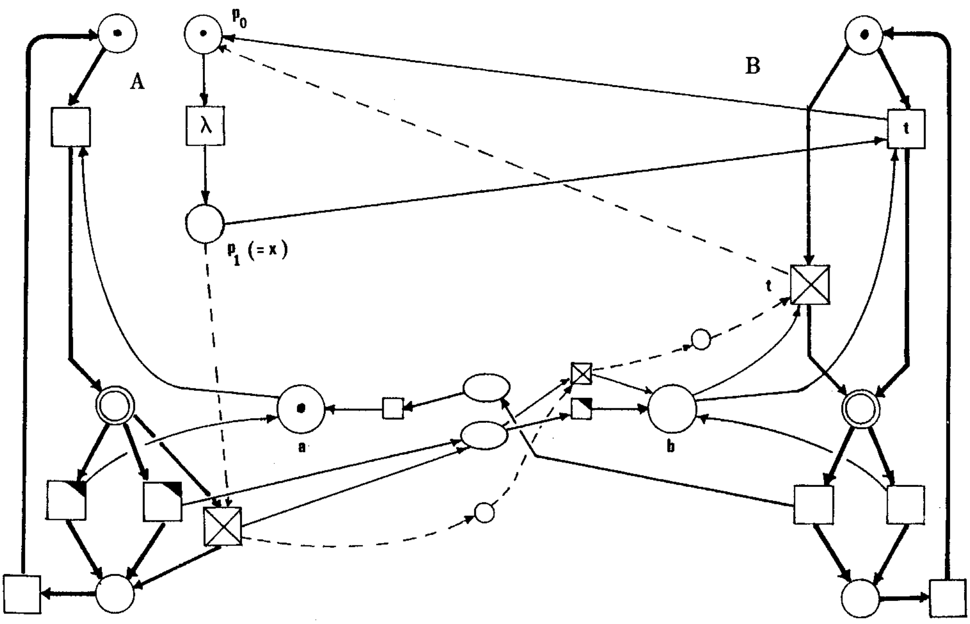


Fig. 5.5: Application of theorem 5.3

Process A can pass the token to B , but also can refuse to do so (under the p-rule). Hence the system is not fair under the p-rule (but would be under the f-rule). Therefore we apply the construction of Fig. 5.2 to transition t in Fig. 5.4 . The resulting net is given in Fig. 5.5 . Again, to simplify the construction only the case is represented, where A is in the critical section. Then after the occurrence of the λ-transition, either transition t can occur (if the token is in b) or a sequence v_1 is enabled, which is marked by a dashed line and crossed transitions. All these transitions are copies of existing ones, which are blocked by the empty side condition p_o . In the construction of theorem 5.3 all transitions having no cross and different from t are to be blocked, but in our case it is sufficient to block only the transitions bearing a black corner in their square symbol. Finally, by the occurrence of the (crossed) copy of t place p_o will be marked again. In the general case, the same construction has to be performed also for other reachable markings, which is not necessary in this particular case.

The similarities of the solutions of Figs. 5.3 and 5.5 are obvious: when leaving his critical section, process A is forced to pass the token to B . The differences lie in the cause of this action: in the first case the action is initialized by a request from B , whereas in the second case by some eventual occurrence of the λ-transition.

In [Roucairol 86] a result similar to theorem 5.3 is given using FIFO-nets. For instance, the example for implementing mutual exclusion can be interpreted in a similar way as the solution of Fig. 5.3 .

At the end of this section we mention some other proposals for fair occurrence rules. We saw in section 1 that the ordinary P/T-system of the five dining philosophers FDPh2 behaves not fair under the fair occurrence rule, i.e. Fair(FDPh2) $\subsetneq L_\omega^{fair}$(FDPh2,id). This would not be the case if we also take into account transitions that are enabled after some immediate steps.

Let m be a marking of a P/T-system Σ and $k \in \mathbb{N}$. Then a transition $t \in T$ is called k-enabled in m , if there is some $v \in T^*$ of length at most k such that vt is enabled in m . t is called ∞-enabled in m , if v can be of arbitrary length. An infinite occurrence sequence $v \in F_\omega(\Sigma)$ is called k-productive and k-fair for $k \in \mathbb{N}_\infty := \mathbb{N} \cup \{\infty\}$ if in the definitions of productive and fair, respt., the word "enabled" is replaced by "k-enabled" [Best 83].

Theorem 5.4

For any P/T-system Σ a number $k_o \in \mathbb{N}$ can be effectively computed such that for all occurrence sequences $v \in F_\omega(\Sigma)$: (v is ∞-fair) $\Leftrightarrow$ (v is k_o-productive) .

Corollary 5.5

For any P/T-system Σ and $v \in F_\omega(\Sigma)$: (v is ∞-fair) $\Leftrightarrow$ ($\forall k \in \mathbb{N}$: v is k-fair) .

The corollary is from [Best 83] whereas the stronger theorem is proven in [Carstensen 86] again by using the theory of residue sets. Note that the sets of all markings for which a transition is k-enabled is right-closed and that the finite residue set R_t can be computed.

By the following program P from [Best 83]

```
program P =                 s := 1
              do  x: true  →  s := s+2
                  y: true  →  s := s-1
                  t: (s=0) →  s := s   od.
```

corollary 5.5 is not true for general models of concurrent systems having the power of Turing machines: the infinite occurrence sequence $(xy)^\omega$ is k-fair for any $k \in \mathbb{N}$ but not ∞-fair.

In [Merceron 86] a notion of fairness for occurrence nets is discussed. A process of a P/T-net is considered to be fair if all associated occurrence sequences are fair. Though this may be a useful notion, it can have the drawback that (unfair) occurrence sequences are taken into account which actually do not appear for a participant of the system (e.g. a philosopher). The reason lies in the fact that processes of P/T-systems do not reflect enabled transitions that do not occur. This is another hint that occurrence sequences are more adequate to study fairness problems. The notion of 'objective' case in [Reisig 86] might be able to combine both aspects, but makes sense for some subclasses of nets only.

A further notion of fairness is based on synchronic distances [Lautenbach 77]. Two transitions are in a fair relation if there exists a positive integer k such that neither of them can fire more then k times without firing the other. This approach is further studied in [Murata/Wu 85].

6. References

[Best 83] E. Best: Fairness and Conspiracies, Information Processing Letters 18 (1983, 215-220

[Boasson/Nivat 80] L. Boasson and M. Nivat: Adherences of Languages, J. Comput. System Sci. 20 (1980), 285-309

[Brinch Hansen 73] P. Brinch Hansen: Operating System Principles, Englewood Cliffs: Prentice-Hall 1973

[Carstensen 82] H. Carstensen: Fairness bei Petrinetzen mit unendlichem Verhalten, Univ. Hamburg, FB Informatik, Bericht Nr. 93, IFI-HH-B-93/82, 1982

[Carstensen 84] H. Carstensen: Fairness in Deadlockfree Petri Nets with the Finite Delay Property, Proc. 5th European Workshop on Applications and Theory of Petri Nets, Arhus 1984, pp 234-253

[Carstensen/Valk 85] H. Carstensen, R. Valk: Infinite Behaviour and Fairness in Petri Nets, in G. Rozenberg (ed.): Advances in Petri Nets 1984, Lect. Notes in Computer Sci. 188, Springer, Berlin, 1985, pp 83-100

[Carstensen 86] H. Carstensen: Fairness bei nebenläufigen Systemen, eine Untersuchung am Modell der Petrinetze, Bericht des FB Informatik, Univ. Hamburg, 1986

[Cohen/Gold 77] R.S. Cohen, A.Y. Gold: Theory of ω-languages, Journ. Computer Syst. Sciences 15 (1977), 169-208

[Dijkstra 68] E.W. Dijkstra: Co-operating sequential processes, in F. Genuys (ed.): Programming Languages, Academic Press, London, 1968, pp 43-112

[Dijkstra 71] E.W. Dijkstra: Hierarchical Ordering of Sequential Processes, Acta Informatica 1 (1971),pp 115-138

[Eilenberg 74] S. Eilenberg: Automata, Languages and Machines, Vol. A., Academic Press, New York 1974

[Hauschildt/Valk 86] D. Hauschildt, R. Valk: Safe States in Banker like Resource Allocation Problems, in: G. Rozenberg (ed.), Advances in Petri Nets 1985, Lecture Notes in Computer Science 222, Springer, Berlin, 1986

[Jessen/Valk 86] E. Jessen, R. Valk: Rechensysteme, Springer-Verlag, Berlin, 1986

[Jantzen 86] M. Jantzen: Language Theory of Petri Nets (this volume)

[Landweber 69] L.H. Landweber: Decision problems for ω-automata, Math. Systems Theory 3 (1969), 376-384

[Lautenbach 77] K. Lautenbach: Ein kombinatorischer Ansatz zur Erreichung von Fairness in Scheduling Problemen, Applied Computer Sci. 8, Verlag C. Hauser, München, 1977

[Merceron 86] A. Merceron: Fair Processes, Proc. 7th Europ. Workshop on Application and Theory of Petri Nets, Oxford, 1986

[Murata, Wu 85] T. Murata, Z. Wu: Fair Relation and Modified Synchronic Distances in a Petri Net, Journ. Franklin Institute, 320 (1985) 2, 68-82

[Pelz 86] E. Pelz: ω-languages of Petri Nets and logical sentences, Proc. 7th Europ. Workshop on Application and Theory of Petri Nets, Oxford, 1986

[Reisig 86] W. Reisig: Coexistence: A Refinement of Concurrency, Proc. 7th Europ. Workshop on Application and Theory of Petri Nets, Oxford, 1986

[Raynal 85] M. Raynal: Algorithmes Distribuês et Protocoles, Eyrolles, Paris, 1985

[Ricart/Agrawala 81] An Optimal Algorithm for Mutual Exclusion in Computer Networks, Comm. ACM, 24 (1981), 9-17

[Roucairol 86] G. Roucairol: FIFO-Nets (this volume)

[Valk 83] R. Valk: Infinite Behaviour of Petri Nets, Theor. Comput. Sci. 25, (1983), 311-341

[Valk/Jantzen 85] R. Valk, M. Jantzen: The Residue of Vector Sets with Applications to Decidability Problems in Petri Nets, Acta Informatica 21 (1985), 643-674