# SELF-MODIFYING NETS, A NATURAL EXTENSION OF PETRI NETS

Rüdiger Valk
Fachbereich Informatik, Universität Hamburg

## 1. Introduction

In this paper we study a natural extension of Petri nets and discuss the usefullness of this approach. Like ordinary Petri nets self-modifying nets are defined as multigraphs having edges of the form $\bigcirc \xrightarrow{q} |$ and $| \xrightarrow{q} \bigcirc$ . Circles denote places and bars represent transitions. But in opposition to ordinary Petri nets q may be either the number 1 or an arbitrary place p of the net. If q = 1, as in the case of a Petri net, firing the transition means, that one token has to be moved from or to the place. But if q is a place p, then the number of tokens to be moved equals the actual number of tokens in p and thus depends on the actual marking. If for an actual marking M we define n := IF $q \in P$ THEN M(q) ELSE q and substitute q by n, we obtain a Petri net and the firing rule for that marking is defined as usually. Therefore a self-modifying net is a Petri net, which is able to modify its own firing rules. Bounded models of such nets have been used to simulate national economic systems /Fu/.

The aim of this paper is threefold :

1. Self-modifying nets have a greater computational power than Petri nets. This is shown using the notation of net languages of Petri nets.

2. Self-modifying nets can be used to increase the understanding of Petri nets. Some questions open for Petri nets can be answered for self-modifying nets and counterexamples to disprove incorrect results on Petri nets can be constructed.

3. Problems in the design and analysis of concurrent processes can be solved by using self-modifying nets in a way which is natural with respect to the problem. This is shown by a first result applicable for different kinds of synchronisation problems.

## 2. Definitions

Throughout this paper $\mathbb{N}$ denotes the set of nonnegative integers. A <u>$\mathbb{N}$-subset of a set X</u> is a function $A : X \longrightarrow \mathbb{N}$ /Ei/. For each $x \in X$ the

element $A(x) \in \mathbb{N}$ is called the multiplicity with which x belongs to A.
A <u>self-modifying net ( SM-net )</u> $N = (P,T,pre,post,M_o)$ is defined by a <u>set of places</u> $P = \{p_1,\dots,p_a\}$ ,
a <u>set of transitions</u> $T = \{t_1,\dots,t_b\}$ , disjoint with P,
a $\mathbb{N}$-subset $M_o$ of P, called <u>initial marking</u> of N, and
two $\mathbb{N}$-subsets pre and post of $P \times P_1 \times T$, where $P_1 := P \cup \{1\}$ and $1 \notin P$.
N is called a <u>post-self-modifying net ( PSM-net )</u> if pre is a $\mathbb{N}$-subset of $P \times \{1\} \times T$ and N is a <u>Petri net ( P-net )</u> if both pre and post are $\mathbb{N}$-subsets of $P \times \{1\} \times T$.
A graphical representation of N as a bipartite multigraph is obtained by representing each $(p,q,t) \in P \times P_1 \times T$ such that $Pre(p,q,t) = n \neq 0$ by n copies of an edge $\underset{p}{\bigcirc}\!\xrightarrow{\;q\;}\!\underset{t}{|}$ and each such tuple with $post(p,q,t) = m \neq 0$ by m copies of an edge $\underset{t}{|}\!\xrightarrow{\;q\;}\!\underset{p}{\bigcirc}$ . A label $q = 1$ may be omitted. Examples can be found in figures 1, 2 and 3. A <u>marking</u> of the SM-net N is a $\mathbb{N}$-subset of P.

Sometimes a marking is written as a vector $M = \begin{pmatrix} M(p_1) \\ \vdots \\ M(p_a) \end{pmatrix} \in \mathbb{N}^a$ or as a word $M \in P^*$ defined by $p_1^{M(p_1)} p_2^{M(p_2)} \dots p_a^{M(p_a)}$ . In the last case exponents identical to one and letters having zero exponents are omitted.

Given a marking $M \in \mathbb{N}^a$ we define $v_M : P_1 \longrightarrow \mathbb{N}$ by $v_M(q) :=$ IF $q \in P$ THEN $M(q)$ ELSE 1 . A transition $t \in T$ is <u>firable</u> for a given marking $M \in \mathbb{N}^a$ if for all $p \in P$ : $M(p) \geqslant \sum_{q \in P_1} pre(p,q,t) \cdot v_M(q)$. A transition $t \in T$ <u>fires a marking M to a marking M'</u> : $M \xrightarrow{\;t\;} M' :\Longleftrightarrow$ t is firable in M and
$\forall p \in P : M'(p) = M(p) - \sum_{q \in P_1} pre(p,q,t) \cdot v_M(q) + \sum_{q \in P_1} post(p,q,t) \cdot v_M(q)$.

We now give an equivalent definition of SM-nets similar to the 'matrix-definition' of P-nets /Ha/.

Let be $\sum := \{n_o + \sum_{i=1}^{a} n_i p_i \mid n_i \in \mathbb{N}\}$ a set of formal sums. For a marking $M \in \mathbb{N}^a$ and $\delta = n_o + \sum_{i=1}^{a} n_i p_i$ define $\delta(M) := n_o + \sum_{i=1}^{a} n_i \cdot M(p_i)$ . Then a SM-net N can be defined by $N = (P,T,B,F,M_o)$, where P, T and $M_o$ are defined as before. $B : P \times T \longrightarrow \sum$ is the <u>backwards incidence function</u> and $F : P \times T \longrightarrow \sum$ is the <u>forwards incidence function</u>. For a given marking M $B_M$ and $F_M$ are functions from $P \times T$ to $\mathbb{N}$ defined by $B_M(p,t) := B(p,t)(M)$ and $F_M(p,t) := F(p,t)(M)$. $B_M$ and $F_M$ are matrices over $\mathbb{N}$ and $\Delta_M := F_M - B_M$ is called the <u>incidence matrix of N in the marking M</u>.

We now have : $M \xrightarrow{t} M' \iff M \geqslant B_M t \wedge M' = M + \Delta_M t$ .

In this notation $t \in T$ is identified with the corresponding unit vector $t \in \mathbb{N}^b$ defined by $t(i) :=$ IF $t=t_i$ THEN 1 ELSE 0 .

We now come to the definition of a net language. The net language is the set of all finite sequences of transitions firable from the initial marking, and therefore represents the set of all possible sequences of actions of a concurrent process. If a set of terminal conditions is given, the language is defined as the set of all firing sequences ending with a terminal condition. Terminal conditions are represented by finite sets of terminal markings.

In applications several transitions can play the role of the same action under different conditions. Therefore all these transitions can be labelled by a single letter. This will be described by a labelling function. The appearence of the empty word $\lambda$ as a label can be interpreted as a transition which is not important from an external point of view.

For any finite sequence $w = t_{i_1} t_{i_2} \dots t_{i_n} \in T^*$ of transitions and for markings M, M' the <u>firing relation</u> $M \xrightarrow{w} M'$ is recursively defined by : $M \xrightarrow{\lambda} M$ ( $\lambda$ is the empty word ) and $\forall w \in T^* \forall t \in T : M \xrightarrow{tw} M' \iff \exists M'' : M \xrightarrow{t} M'' \wedge M'' \xrightarrow{w} M'$ . Let be $(P,T,B,F,M_o)$ a SM-net, $\mathfrak{M}_t \subseteq \mathbb{N}^a$ a finite <u>set of terminal markings</u> and $h : T \longrightarrow X \cup \{\lambda\}$ a <u>labelling function.</u> Then $N = (P,T,B,F,M_o, \mathfrak{M}_t, h, X)$ is called a <u>labelled net with terminal marking set</u>. For a labelled SM-net N with terminal marking set we define the <u>terminal language of N</u> by :

$$L_o(N) := \{ h^*(w) \mid \exists M \in \mathfrak{M}_t : M_o \xrightarrow{w} M \}$$

( $h^* : T^* \longrightarrow X^*$ is the monoid homomorphism generated by h. h is called $\lambda$-free if $h(t) \neq \lambda$ for all $t \in T$.)

From this the following families of languages are defined :

<u>the family of terminal SM-languages</u> :

$$\mathcal{L}_o^\lambda(SM) := \{ L_o(N) \mid N \text{ is a SM-net} \} \quad ,$$

<u>the family of $\lambda$-free terminal SM-languages</u> :

$$\mathcal{L}_o(SM) := \{ L_o(N) \mid N \text{ is a SM-net with } \lambda\text{-free } h \}$$

By restricting SM-nets to PSM-nets and P-nets we obtain the families $\mathcal{L}_o^\lambda(PSM)$, $\mathcal{L}_o^\lambda(P)$ and $\mathcal{L}_o(PSM)$, $\mathcal{L}_o(P)$, respectively.

Sometimes we refer to the well-known reachability problem for P-nets. Since we do not know at this time, whether the proof of /ST/ showing its decidability is true, we sometimes refer to the following assumption :

(R) : "the reachability problem for P-nets is solvable".

## 3. Languages generated by SM-nets

We now prove that languages generated by SM-nets, PSM-nets and P-nets are different, which shows that their computational power is different.

Let be $X := \{x_1, x_2\}$ an alphabet and $w^R$ the reversal of a word w. The following languages will be used :

$L_R := \{ww^R \mid w \in X^*\}$ and $L_{Ry} := \{ww^R y^{f(w)} \mid w \in X^*\}$ where $f(w) :=$ IF $lg(w) \geqslant 3$ THEN $3^{lg(w)-2}$ ELSE 0 and $lg(w)$ denotes the length of the word w. It is known from /Pe/ that $L_R$ cannot be generated by $\lambda$-free P-nets.

Lemma 1 : $L_R \in \mathcal{L}_o(SM)$

proof : Consider the SM-net $N = (P,T,B,F,M_o)$ given by the graph in figure 1. (A label 2p stands for two edges, both labelled by p.) Let be $M_o = p_o p''$ the initial marking and $\mathfrak{M}_t = \{M_t\} = \{p_t\}$ the set of terminal markings. The $\lambda$-free labelling function $h : T \to X$ is given in the graph. We shall prove that $L_o(N) = L_R' := \{ww^R \mid w \in XXXX^*\}$. Since $\mathcal{L}_o(SM)$ is closed by union and contains all finite sets, it then follows $L_R \in \mathcal{L}_o(SM)$. Let be $w = xvyy'v'x'$ with $v = x_{i_1} x_{i_2} \ldots x_{i_n}$, $v' = x_{j_1} x_{j_2} \ldots x_{j_m}$, and $x, x', y, y', x_{i_k}, x_{j_k} \in X$. Then after firing $xvyy'v'$ in the places p and p' are $c_v = \sum_{k=0}^{n-1} 3^k \cdot i_{n-k}$ and $c_{v'} = \sum_{k=0}^{m-1} 3^k \cdot j_{k+1}$ tokens, respectively. The terminal marking $p_t$ can be reached only if by a firing the places p and p' become empty. Thus $w \in L_o(N)$ iff $c_v = c_v'$ and $x = x'$ and $y = y'$, i.e. $y'v'x' = (xvy)^R$. QED

Lemma 2 : $L_R \in \mathcal{L}_o^{\lambda}(PSM)$ and $L_{Ry} \in \mathcal{L}_o(PSM)$

proof : The net of fig. 1 is not a PSM-net since it includes the arrows starting in the places p, p' and p'', the task of which was to perform the test. All other arrows are also allowed for PSM-nets. But this test can also be done by a repeated firing of a transition which removes one token simultanously from p and p'. This transition must be labelled by the empty word $\lambda$. Furthermore place p'' must be emptied by a $\lambda$-transition. Thus we get $L_R \in \mathcal{L}_o^{\lambda}(PSM)$. To prove $L_{Ry} \in \mathcal{L}_o(PSM)$ the labels of transitions counting down p, p' and p'' are supposed to be the letter y. QED

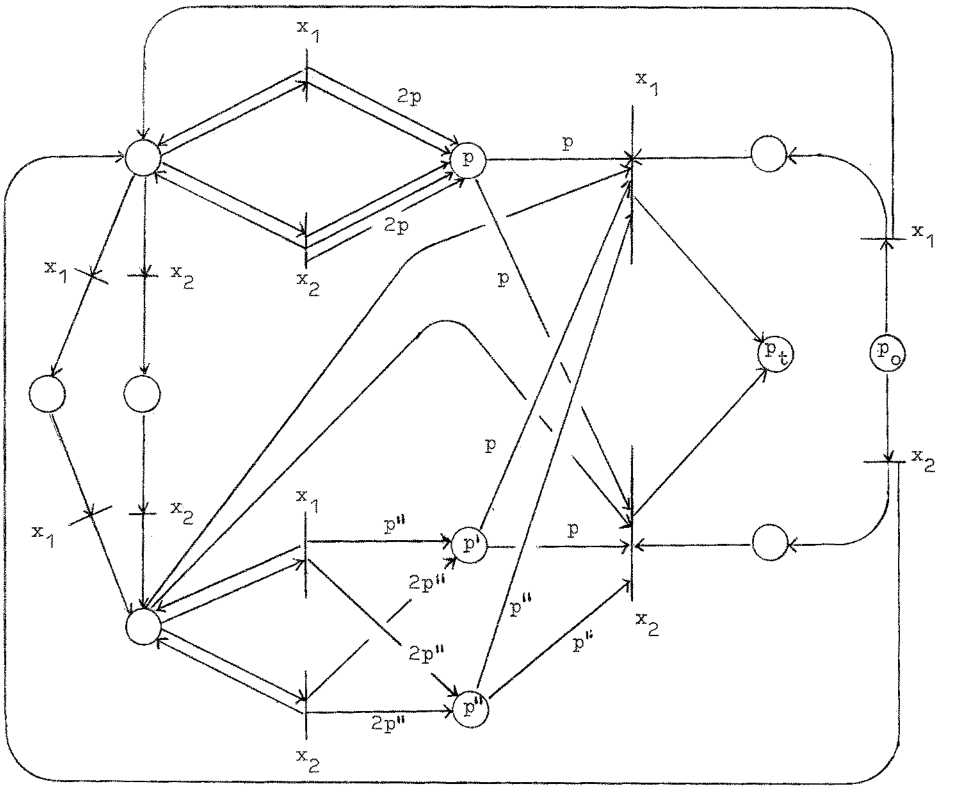



figure 1

<u>Lemma 3</u> : $L_R \notin \mathcal{L}_o(\text{PSM})$

proof : Suppose $L_R = L_o(N)$ for a PSM-net N having card(P) = a places, initial marking $M_o = p_o$ and $\mathcal{M}_t = \{M_t\} = \{\underline{0}\}$ as a set of terminal markings ( $\underline{0}$ denotes the zero marking). Since there are $2^l$ different words of length l over the alphabet $X = \{x_1, x_2\}$ , the net must be able to reach at least $2^l$ different markings after firing l times. Let be $k := \max\{B(p,t) \mid p \in P,\ t \in T\}$ k is well-defined for PSM-nets. We choose a number $l \in \mathbb{N}$ such that

$$2^l > (k \cdot l + 1)^a \qquad \text{(I)}$$

Having a number of a places and assuming at most q tokens in any place there are $(q + 1)^a$ different markings. Since by the assumption $2^l$ different markings can be reached after $l = lg(w)$ firings, at least one place $\hat{p}$ must contain

$\hat{q} > 2^{1/a}-1$ tokens. But by any transition at most k tokens can be removed from a single place, i.e. after firing a sequence of transitions labelled by the letters of the sequence $w^R$ at most $k \cdot lg(w) = k \cdot l$ tokens can be removed from $\hat{p}$. As a consequence of (I) it follows :

$$\hat{q} - k \cdot l > 2^{1/a} - 1 - k \cdot l > 0$$

i.e. the terminal marking $M_t$ cannot be reached. This is a contradiction. If the net has arbitrary initial and terminal markings, similar arguments can be applied. QED

The last proof is a simplified version of the proof that $L_R$ is not in $\mathcal{L}_o(P)$ in /Pe/. By the same method the following lemma can be proved.

<u>Lemma 4</u> : $L_{Ry} \notin \mathcal{L}_o(P)$

The preceeding lemmas establish a strict hierarchie for $\lambda$-free nets :

<u>Theorem 1</u> : $\mathcal{L}_o(P) \subsetneqq \mathcal{L}_o(PSM) \subsetneqq \mathcal{L}_o(SM)$

proof : $L_{Ry} \in \mathcal{L}_o(PSM) - \mathcal{L}_o(P)$ (by lemmas 2 and 4)

$L_R \in \mathcal{L}_o(SM) - \mathcal{L}_o(PSM)$ (by lemmas 1 and 3) QED

We now answer the question whether the introduction of $\lambda$-transitions increases the family of SM-languages as in the case of P-nets /Ja/.

<u>Lemma 5</u> : $\mathcal{L}_o^\lambda(PSM)$ is an intersection closed full semi-AFL, i.e. closed under intersection, union, intersection with regular sets, homomorphisms and inverse homomorphic images.

The proof is essentially the same as the proof for the same assertion for the family $\mathcal{L}_o^\lambda(P)$ /Ha2,Pe/. This shows the structural similarity of SM-nets and P-nets.

As a corollary we obtain the following theorem.

<u>Theorem 2</u> : $\mathcal{L}_o^\lambda(PSM) = \mathcal{L}_o^\lambda(SM)$ is the family RE of recursively enumerable languages.

proof : Obviously we have $\mathcal{L}_o^\lambda(PSM) \subseteq \mathcal{L}_o^\lambda(SM) \subseteq RE$ . RE is the smallest intersection-closed full semi-AFL containing the language $L_R$ /BB/, hence by lemmas 2 and 5 : $\mathcal{L}_o^\lambda(PSM) \supseteq RE$. QED

$R(N) := \{ M \in \mathbb{N}^a \mid \exists w \in T^* : M_o \xrightarrow{w} M \}$ is the set of reachable markings or the reachability set of a SM-net. N is called deadlockfree, if for every reachable marking $M \in R(N)$ at least one transition is firable. The reachability problem for SM-nets is the question whether for a given SM-net N a given marking $M_t$ belongs to R(N).

Corollary 1 : The reachability problem is undecidable for both, PSM-nets and SM-nets.

proof : By theorem 2 the reachability problem for PSM-nets is equivalent to the emptyness problem in RE. QED

Remark : By similar arguments as above $\mathcal{L}_o(\mathrm{PSM})$ can be shown to be a semi-AFL. Furthermore it can be proved that $\mathcal{L}_o(\mathrm{PSM})$ is closed under the Kleene star operator. Therefore $\mathcal{L}_o(\mathrm{PSM})$ is an intersection-closed AFL. By the argument used in the proof of theorem 2 and corollary 1 from (R) it follows also $L_R \notin \mathcal{L}_o^\lambda(P)$ /Ja/.

Theorem 3 :

$$\begin{array}{ccccccccc} \mathcal{L}_o(P) & \subsetneqq & \mathcal{L}_o(\mathrm{PSM}) & \subsetneqq & \mathcal{L}_o(\mathrm{SM}) & & \\ \cap\# & & \cap\# & & \cap\# & & \\ \mathcal{L}_o^\lambda(P) & \subseteq & \mathcal{L}_o^\lambda(\mathrm{PSM}) & = & \mathcal{L}_o^\lambda(\mathrm{SM}) & = & \mathrm{RE} \end{array}$$

and $\mathcal{L}_o^\lambda(P) \subsetneqq \mathcal{L}_o^\lambda(\mathrm{PSM})$ if (R) is true.

proof : By lemmas 2 and 3 we have $L_R \in \mathcal{L}_o^\lambda(\mathrm{PSM}) - \mathcal{L}_o(\mathrm{PSM})$. $\mathcal{L}_o(\mathrm{SM})$ is a subset of the family of decidable languages, hence $\mathcal{L}_o(\mathrm{SM}) \neq \mathrm{RE}$. From the remark it follows $L_R \in \mathcal{L}_o^\lambda(\mathrm{PSM}) - \mathcal{L}_o^\lambda(P)$ if (R) is true. $\mathcal{L}_o(P) \neq \mathcal{L}_o^\lambda(P)$ is shown in /Ja/.

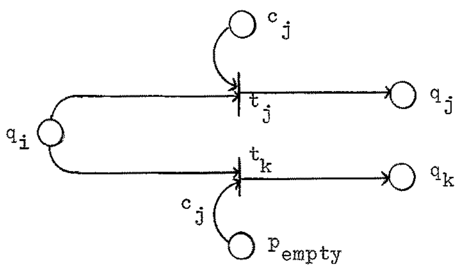
## 4. Decidability of boundedness

Decidability of boundedness is one of the most characteristic properties of Petri nets. In this section it will be shown, that boundedness remains decidable for PSM-nets, but becomes undecidable for SM-nets. Thus the class of PSM-nets is a finite device generating all recursively enumerable languages, for which boundedness of memory requirements is decidable.

A SM-net $N = (P,T,B,F,M_o)$ is bounded, if there is an integer n such that $M(p) \leq n$ for all places $p \in P$ and all reachable markings $M \in R(N)$.

Theorem 4 : Given a SM-net, it is undecidable whether it is bounded.

proof : Since for a counter automaton with at least two counters boundedness is undecidable, it suffices to show that every counter automaton can be simulated by a SM-net. Every counter can be simulated by a place. The finite control can be trivially represented by a (safe) SM-net. As in the counter automaton the simulating SM-net is able to increase the content of a specified counter in some state. The essential point, however, is to show how a test $q_i$ : IF $c_j > 0$ THEN $c_j \leftarrow c_j - 1$ GOTO $q_j$ ELSE GOTO $q_k$ of the counter automaton can be simulated by a net. $q_i$ is the actual state, $q_j$ and $q_k$ are possible next states and $c_j$ is the content of a counter j. This simulation can be done in the following way, where $p_{empty}$ is a place, which never contains a token.




Then $t_j$ can be fired iff $c_j$ is not empty and $t_k$ is firable iff $c_j$ is empty.

QED

Theorem 5 : Given a PSM-net, it is decidable whether it is bounded.

proof : We first prove the following property (I) of PSM-nets :

Given a PSM-net $N = (P,T,B,F,M_o)$ and markings $M_1$, $M_2$, $M_1' \in \mathbb{N}^a$ and a transition $t \in T$ such that $M_1 \xrightarrow{t} M_2$ and $M_1' \geqslant M_1$ and $M_1' \neq M_1$, then t is firable for $M_1'$ to a marking $M_2'$ and $M_1' \xrightarrow{t} M_2'$ and $M_2' \geqslant M_2$ and $M_2' \neq M_1$.

Since t is firable for $M_1$ we have $M_1 \geqslant B_{M_1} t$. But in the case of a PSM-net we have $B_{M_1} = B_M$ for all $M \in \mathbb{N}^a$ and therefore $M_1' \geqslant M_1 \geqslant B_{M_1'} t$ and t is firable for $M_1'$. $M_1' \geqslant M_1$ implies $F_{M_1'} \geqslant F_{M_1}$. Therefore it follows $\Delta_{M_1'} = F_{M_1'} - B_{M_1'} = F_{M_1'} - B_{M_1} \geqslant F_{M_1} - B_{M_1} = \Delta_{M_1}$ and $M_2' = M_1' + \Delta_{M_1'} t \geqslant M_1 + \Delta_{M_1} t = M_2$ and $M_2' = M_2$

would imply $M_1 = M_1'$.

Clearly property (I) remains true if transition t is substituted by a finite sequence $w \in T^*$ of transitions. In /Ha1/ a decision procedure for boundedness of P-nets is given using the 'coverability tree'. This proof is due to Karp and Miller. Observing property (I) this proof can be rewritten for PSM-nets.

QED

Remark : By theorem 5 the proof of $\mathcal{L}_0^\lambda(P) \neq RE$ in /Ha2/ also applies to $\mathcal{L}_0^\lambda(PSM)$ in contradiction to theorem 2. Therefore the proof in /Ha2/ is false.

The decision procedure for regularity of P-nets as given in /VV/ can also be applied to PSM-nets.

Let be RMa the class of nondeterministic register machines which have conditional goto-instructions and instructions allowing to increment the content of one register by the content of a second register. Then the family of $\lambda$-free terminal PSM-languages equals the family of quasi-real-time languages accepted by those register machines. The proof is long and technical, but shows how to work with PSM-nets /Va/.

We now summarize some of the properties stated about SM-nets :

| | P-nets | PSM-nets | SM-nets |
|---|---|---|---|
| $L_R \in \mathcal{L}_0$ | no /Pe/ | no | yes |
| $L_R \in \mathcal{L}_0^\lambda$ | no, if (R) is true /Ja/ | yes | yes |
| $\mathcal{L}_0^\lambda = RE$ | no, if (R) is true | yes | yes |
| reachability decidable | yes, if (R) is true | no | no |
| boundedness decidable | yes | yes | no |
| $\mathcal{L}_0 = \mathcal{L}_0^\lambda$ | no /Ja/ | no | no |

## 5. Deadlock-freeness of concurrent processes

In this chapter we show that SM-nets can be used to find elegant descriptions of synchronisation problems. Since unrestricted Petri nets are too powerful to guarantee wellstructured solutions, also the use of SM-nets must be strongly restricted.

A SM-net is called to be spontaneous, if for every reachable marking $M = p_{i_1}p_{i_2}\cdots p_{i_m}$ and every $j \in \{1,..,m\}$ there is a transition t, which is firable for M and which moves at least one token from the place $p_{i_j}$. If $B(p,q,1) = 1$ is true for a SM-net, we speak of the edge $(p,q,t)$ of the net. It is called a SM-edge, if q is a place of the net. A spontaneous process system with constraints ( SPSC ) is a spontaneous P-net $N = (P,T,B,F,M_o)$, to which for edges $(p,1,t)$ new SM-edges $(p,p',t)$ may be added. Then the P-net N is called the underlying P-net of the SPSC.

Although the definition of a SPSC seems to be very restricted, it is rather usefull for practical synchronisation problems.

If $N = (P,T,B,F,M_o)$ is a SPSC, then we define connected places by $p \sim p' \;:\Longleftrightarrow\; \exists\, t \in T : B(p,1,t) = B(p',1,t) = 1$. Every sequence of SM-edges $(p_o,p_o,t_o), (p_1,p_1',t_1),..,(p_n,p_n',t_n)$ such that $p_i' \sim p_{i+1}$ for all $0 \leq i \leq n-1$ and $p_n' \sim p_o$ is called to be a critical circuit. (For n = 0 the SM-edge $(p_o,p_o',t_o)$ with $p_o \sim p_o'$ is a critical sequence.)

Lemma 6 : Let be M a marking in a SPSC and $p_o \in M$. If the token in $p_o$ cannot be fired away in the SPSC, but can be fired away in the underlying P-net, then there exist $p_o' \in M$, $p_1 \in M$, $p_o' \sim p_o$ and $(p_o',p_1,t_o)$ is an edge of the SPSC.

Theorem 6 : A spontanous process system with constraints is deadlockfree, if it contains no critical circuit.

proof : Let be $N = (P,T,B,F,M_o)$ a SPSC without critical circuits and $M = p_{i_1}p_{i_2}\cdots p_{i_m}$ a reachable marking. We prove that there is a transition, which is firable in M. Let be $p_o$ a place with a token in M. If this token can be fired away, the proof ends. If this not the case, by the lemma there are $p'_o$ and $p_1$ containing a token with $p'_o \sim p_o$ and $(p'_o, p_1, t_o)$ is an edge. Suppose that starting in this way a sequence $(p'_i, p_{i+1}, t_i)_{o \leq i \leq n}$ with $p_i \sim p'_i$, and all $p_i$, $p'_i$ contain a token $(o \leq i \leq n)$, has been constructed. If the token in $p_{n+1}$ can be fired away, the proof ends, if not then by the lemma the sequence can be continued for $o \leq i \leq n+1$. Finally, since $n > \mathrm{card}(P)$ would imply, that in contradiction to the assumption the net contains a critical circuit. Therefore the construction must stop, and the net is deadlockfree. QED

As an example of a SPSC consider a SM-net representation of the second reader writer problem of /CHP/ in figure 2. Places rc and wc represent the variables 'readcount' and 'writecount' of /CHP/, respectively. It is obvious, that the underlying P-net is spontanous. Since there is no critical sequence, the net is deadlockfree.
Remark : From (R) it can be deduced, that this problem cannot be modelled by Petri-nets assuming an unbounded number of writers.

As a second example consider the SM-net representation of an incorrect solution for the five-philosophers-problem in figure 3. A token in place $p_i$ means "philosopher i is thinking", $l_i$ means "philosopher i took the left fork", $e_i$ means "philosopher i took both forks and is eating". The net is a SPSC and contains $(l_i, l_{i+1}, t_i)_{1 \leq i \leq 5}$ with $l_6 = l_1$ as a critical circuit. It is deadlocked in the marking $M = l_1 l_2 l_3 l_4 l_5$. Other examples of deadlockfree SPSC, where the theorem is applicable, are the first reader-writer-problem in /CHP/, the correct solution of the five-philosophers-problem and the SM-net representation of the airline reservation system in /As/. In the last case the structure of the underlying P-net is essentially the structure of a state machine net and therefore trivially spontanous.

For more complex systems similar results have been found using underlying P-nets, where the behaviour of the tokens can be described as a flow.

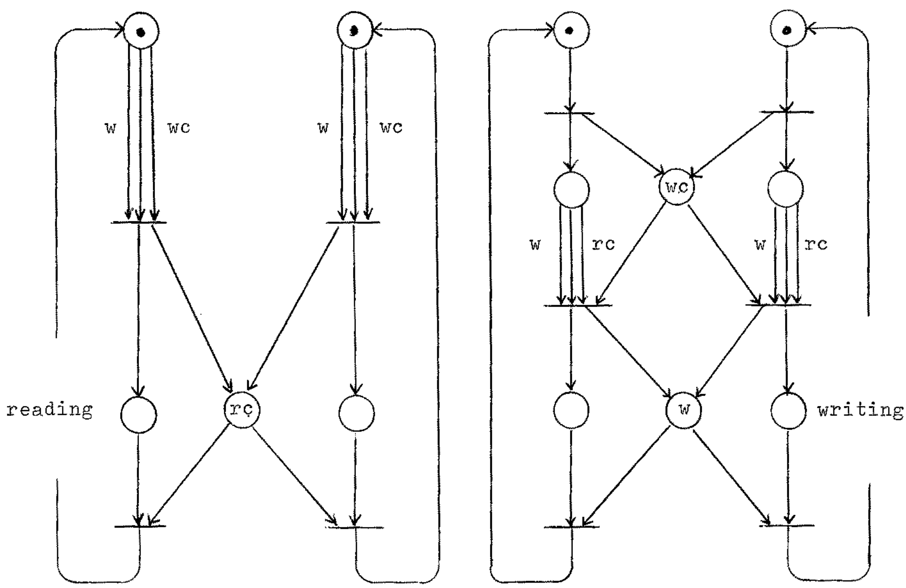



figure 2

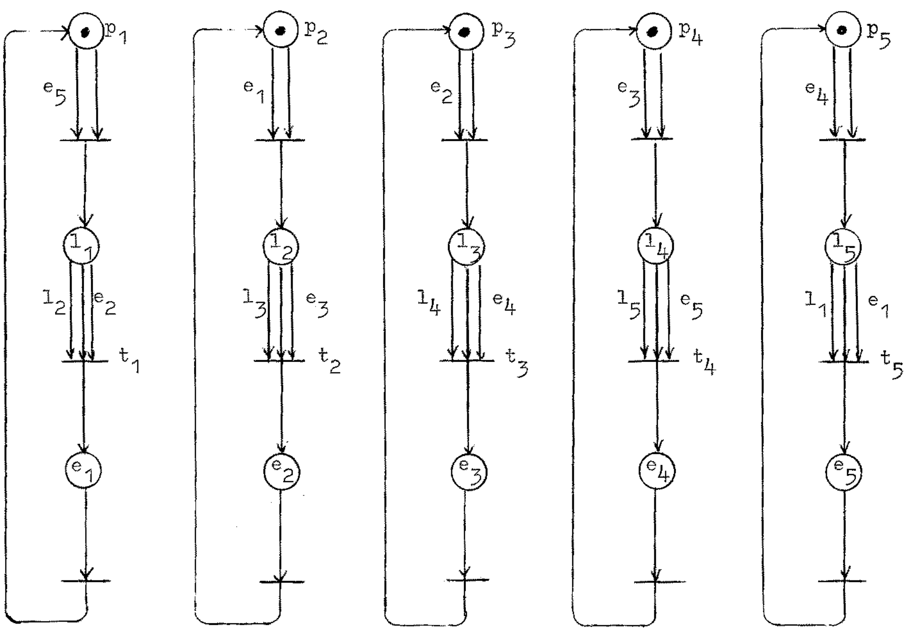



figure 3

## 6. References


/As/ ASHCROFT,E.A., Proving Assertions about Parallel Programs, Journ. Computer Syst. Scie. 10(1975),110-135

/BB/ BAKER,B.S., BOOK,V.B., Reversal-Bounded Multipushdown Machines, Journ. Computer Syst. Scie. 8(1974),315-332

/CHP/ COURTOIS,P.J., HEYMANS,F., PARNAS,D.L., Concurrent Control with "Readers" and "Writers", Comm.ACM 14(1971)10,667-668

/Ei/ EILENBERG,S., Automata, Languages and Machines, Vol A, Academic Press, New York, 1974

/Fu/ FUSS,H.E., AFMG - Ein asynchroner Fluss-Modell-Generator, Gesellschaft für Mathematik und Datenverarbeitung, Bericht Nr. 100, Bonn 1975

/Ha1/ HACK,M., Decision Problems for Petri Nets and Vector Addition Systems, MAC Techn. Mem. 59, MIT, 1975

/Ha2/ HACK,M., Petri Net Languages, MIT Comp. Struct. Group, Memo 124, 1975

/Ja/ JANTZEN, M., On the Hierarchy of Petri Net Languages, submitted to Rev. Française Automat. Informat. Recherche Opérationnelle

/Pe/ PETERSON,J.L., Computation Sequence Sets, Journ. Computer Syst. Scie. 13(1976),1-24

/ST/ SACERDOTE,G.S., TENNEY,R.L., The Decidability of the Reachability Problem for Vector Addition Systems, (Preliminary Version), Proc. 9th. ACM-Sympos. Theory of Computing, New York 1977

/Va/ VALK, R., Self-modifying Nets, Institut für Informatik, Universität Hamburg, Bericht IFI-HH-B-34/77

/VV/ VALK,R., VIDAL-NAQUET,G., On the Rationality of Petri Net Languages, Theoretical Computer Scie., 3rd GI-Conf. 1977, Lec. Notes in Computer Scie., No 48, Springer, Berlin 1977