ON PRIORITIES OF PARALLELISM: PETRI NETS UNDER THE MAXIMUM FIRING STRATEGY

Hans-Dieter Burkhard

Sektion Mathematik
Humboldt-Universität
DDR-1086 Berlin, PSF 1297

Abstract: The computational power of Petri nets is extended up to the power of counter machines by realizing certain priorities of parallelism. Hence certain concurrent computations can not exactly be reflected by the sets of all sequentialized computations in related systems. Moreover, the reachability, boundedness and liveness problems are undecidable under the modified firing rule.

0. Introduction

The states and the processed sequences in concurrent systems may be heavily affected by the assumptions about the occurences of parallelism. To show this we consider concurrent computations using the Petri net model where we claim that maximal sets of simultaneously firable transitions have to fire in parallel ("Maximum Firing Strategy" 2.1). Petri nets under this firing rule are of more computational power than the nets under the common firing rule (3.2).

While the common firing rule (1.2) for Petri nets corresponds to all possible sequentialized computations (executable by one processor) (1.3), the Maximum Firing Strategy allows only those concurrent computations which make use of the maximally possible parallelism (with a related number of processors). This concept is related to the strategy MAX for concurrent computations, which was introduced by Salwicki and Müldner /SM/ . The extended computational power under the Maximum Firing Strategy implies that there are concurrent computations which can not be faithfully represented by the set of sequentialized runs.

Furthermore, the Petri nets working under the Maximum Firing Strategy are able to simulate counter machines (3.2). As a consequence the boundedness, reachability and liveness problems are undecidable (4.1). This result may be unpleasant with respect to practical use.

But, as it can be seen by the used constructions, these results already hold for parallelism of two processors: If at least two transitions have to fire simultaneously whenever this is possible, then the computational power is again extended up to the power of counter machines (4.4). In this sense it can be stated that the use of parallelism must be paid by undecidability results.

1. Preliminaries. The common firing rule.

1.1 $\mathbb{N}$ is the set of all non-negative integers. For a finite alphabet A, A^* is the free monoid with the empty word e . Operations and relations on vectors are understood componentwise.

A (generalized initial) Petri net is given by $\mathcal{N} = (P, T, F, m_\alpha)$, where P and T are the finite sets of places and transitions, respectively. $F: (P \times T) \cup (T \times P) \longrightarrow \mathbb{N}$ is the flow function, $m_\alpha \in \mathbb{N}^P$ is the initial marking. For a transition $t \in T$ we define the vectors $t^-, t^+ \in \mathbb{N}^P$ by $t^-(p) := F(p,t)$, $t^+(p) := F(t,p)$ $(p \in P)$.

1.2 The transition $t \in T$ is firable under the common firing rule at a marking $m \in \mathbb{N}^P$ iff $t^- \leqq m$. After its firing the new marking is $m + \Delta t$, where $\Delta t := t^+ - t^-$.

A sequence $u = t_1 \dots t_n \in T^*$ is a firing sequence under the common firing rule iff each transition t_i $(i=1,\dots,n)$ is firable at the marking $m_\alpha + \Delta t_1 + \dots + \Delta t_{i-1}$ under the common firing rule, it leads to the new marking $m_\alpha + \Delta u$, where $\Delta u := m_\alpha + \Delta t_1 + \dots + \Delta t_n$. If only one processor is working, then the firing sequences may be considered as the computational sequences which can be processed by this processor.

1.3 The set of all firing sequences under the common firing rule of a Petri net $\mathcal{N}$ is denoted by $L_{\mathcal{N}}$. The following pumping lemma /B2/ holds:

There are numbers k, l for each language $L_{\mathcal{N}}$ such that the following holds:
If the length of a sequence $u \in L_{\mathcal{N}}$ is greater than k ,
then there is a decomposition $u = u_1 u_2 u_3$ such that
$1 \leqq$ length of $u_2 \leqq l$ and $u_1 u_2^{n+1} u_3 \in L_{\mathcal{N}}$ for all $n \in \mathbb{N}$.

By the modified firing rule, which we shall define later on (2.1), we get sets of firing sequences which are subsets of $L_{\mathcal{N}}$. In general, such a pumping lemma is not valid for these sets.

1.4 The set of all reachable markings under the common firing rule is defined by $R_{\mathcal{N}} := \{ m_\alpha + \Delta u \ / \ u \in L_{\mathcal{N}} \}$. For a given subset $X \subseteq P$

of places the (non-terminal) Petri net predicate $M_{\mathcal{N},X}$ is defined as the projection of $R_{\mathcal{N}}$ on the places of X :

$M_{\mathcal{N},X} := \{ x \in \mathbb{N}^X \; / \; \exists m \in R_{\mathcal{N}} : m(p) = x(p) \text{ for all } p \in X \}$.

There is again a pumping lemma /B2/:

There are vectors $y', y'' \in \mathbb{N}^X$ for each set $M_{\mathcal{N},X}$ such that the following holds:
If $x \in M_{\mathcal{N},X}$ covers y' (i.e. $x \geqq y'$), then there exists a vector $z \in (\mathbb{N} \setminus \{0\})^X$ such that $z \leqq y''$ and $x + n \cdot z \in M_{\mathcal{N},X}$ for all $n \in \mathbb{N}$.

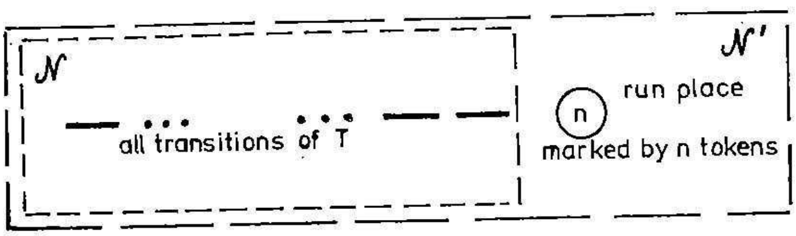
In general, the Petri net predicates computable by the modified firing rule do not satisfy such a pumping lemma.

2. Firing under the Maximum Strategy.

2.1 The strategy "MAX" for concurrent computations was introduced by Salwicki and Müldner /SM/ : As many processes as possible (limitations may arise by conflicts) have to work concurrently. Thus we want to make use of maximal parallelism. This can be represented in Petri nets by the following firing rule called the Maximum Firing Strategy:

In a marking m we choose a maximal set T' of simultaneously firable transitions, i.e. $\sum_{t \in T'} t^- \leqq m$ and $\sum_{t \in T''} t^- \nleq m$ for all $T'' \supsetneq T'$.
Then the transitions of T' are fired simultaneously. After this firing the new marking is $m + \sum_{t \in T'} \Delta t$. For that marking a new set T' is chosen ...

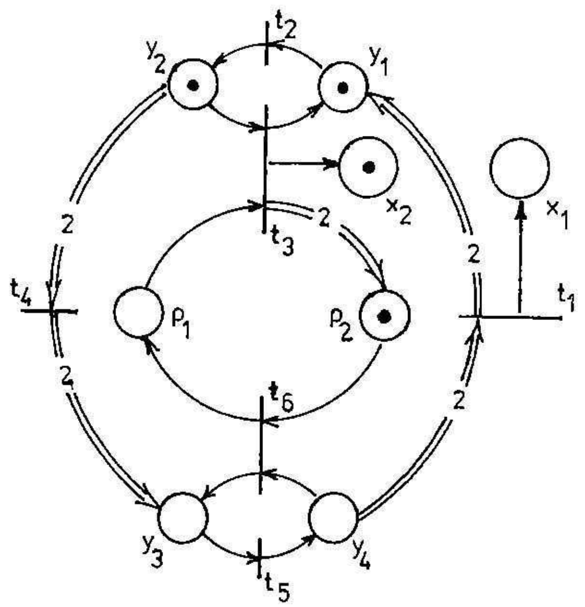
2.2 The number of simultaneously firable transitions is bounded by the number of transitions in the net. Additionally it can be bounded by the structure of the net. By adding a "run place" it is possible to change the net (thereby preserving the internal structure) such that not more than a given number n of transitions may fire simultaneously:



2.3 The set $R_{\mathcal{N}}^{MAX}$ of all reachable markings under the Maximum Firing Strategy contains all those markings which can be reached from m_o by firing the maximal sets T' of simultaneously firable transitions, i.e., only those markings are valid which are reached when all transitions of a set T' have fired. However, the results presented in this paper remains true if we consider the sets additionally containing the intermediate markings (where some transitions of T' have fired — this would be related to the languages as in 3.4).

The Petri net predicate $M_{\mathcal{N},X}^{MAX}$ under the Maximum Firing Strategy is the projection of $R_{\mathcal{N}}^{MAX}$ on the places of the set $X \subseteq P$. For each net $\mathcal{N}$ we have: $R_{\mathcal{N}}^{MAX} \subseteq R_{\mathcal{N}}$ and $M_{\mathcal{N},X}^{MAX} \subseteq M_{\mathcal{N},X}$.

2.4 As an example we consider the following net (a modified version of Hack's example for the weak computation of 2^i):



We have $M_{\mathcal{N},X} = \{ (i,j) \ / \ i \in \mathbb{N} \ \wedge \ 1 \leq j \leq 2^i \}$ under the common firing rule for $X = \{ x_1, x_2 \}$.

The computations under the Maximum Firing Strategy lead to the following reachability graph, whereby

$m = (m(x_1), m(x_2), m(p_1), m(p_2), m(y_1), \dots, m(y_4))$:

$$
\begin{array}{c}
(0,1,0,1,1,1,0,0) \\
\downarrow \{t_2\} \\
(0,1,0,1,0,2,0,0) \\
\downarrow \{t_4\} \\
(0,1,0,1,0,0,2,0) \\
\downarrow \{t_5\} \\
(0,1,0,1,0,0,1,1) \\
\downarrow \{t_5, t_6\} \\
(0,1,1,0,0,0,1,1) \\
\downarrow \{t_5\} \\
(0,1,1,0,0,0,0,2) \\
\downarrow \{t_1\} \\
(1,1,1,0,2,0,0,0) \\
\downarrow \{t_2\} \\
(1,1,1,0,1,1,0,0) \\
\downarrow \{t_2, t_3\} \\
(1,2,0,2,1,1,0,0)
\end{array}
\quad \longrightarrow \quad
\begin{array}{c}
(i,2^i,0,2^i,1,1,0,0) \\
\downarrow \{t_2\} \\
(i,2^i,0,2^i,0,2,0,0) \\
\downarrow \{t_4\} \\
(i,2^i,0,2^i,0,0,2,0) \\
\downarrow \{t_5\} \\
(i,2^i,0,2^i,0,0,1,1) \\
\downarrow \{t_5, t_6\} \\
(i,2^i,1,2^i-1,0,0,1,1) \\
\downarrow \{t_5, t_6\} \\
\vdots \\
\downarrow \{t_5, t_6\} \\
(i,2^i,2^i,0,0,0,1,1) \\
\vdots \\
\downarrow \\
(i+1,2^i,2^i,0,1,1,0,0) \\
\downarrow \{t_2, t_3\} \\
\vdots \\
\downarrow \{t_2, t_3\} \\
(i+1,2^{i+1},0,2^{i+1},1,1,0,0) \\
\vdots \\
\downarrow
\end{array}
$$

Here we have $M^{MAX}_{\mathcal{N}, X} = \{ (i,j) \;/\; i \in \mathbb{N} \wedge 2^{i-1} \leq j \leq 2^i \}$. Since this set contains no infinite linear subset, it does not satisfy the conditions of the pumping lemma in 1.4 . Hence it can not be computed in any Petri net under the common firing rule.

2.5 . In the example a <u>while-loop</u> is realized: If the places y_1 and y are each marked by one token, then the transitions t_2 and t_3 have to fire simultaneously as long as there are tokens in place p_1. Thus we have under the Maximum Firing Strategy:

while $m(p_1) > 0$ **do** **begin** $m(p_1) := m(p_1) - 1$;
$m(p_2) := m(p_2) + 2$;
$m(x_2) := m(x_2) + 1$ **end** .

Another while-loop is realized by the transitions t_5 and t_6 . Under the common firing rule it can not be possible to realize while-loops in Petri nets. Otherwise the set $M^{MAX}_{\mathcal{N}, X}$ of our example would also be computable under the common firing rule.

2.6 The reachability graphs under the Maximum Firing Strategy (as well as under the common firing rule) may be infinite as in our example. In general they may also have branchings (if there are two or more maximal sets of simultaneously firable transitions). The reachability graph is finite iff the net is bounded (iff all reachable markings are bounded).

2.7 Certain properties of Petri nets - especially boundedness - under the common firing rule can be examined with help of the well-known construction of the coverability tree /KM/, /K1/, /H1/. An important fact used for this construction is the following one (which is also related to the pumping lemma 1.4):

If a marking m is reachable from m' by firing of a sequence u , then m + a is reachable from m' + a by firing of u under the common firing rule for each $a \in \mathbb{N}^P$.

This is not true for the Maximum Firing Strategy as it can be seen by the example. Hence a coverability tree with respect to the Maximum Firing Strategy can not be constructed. Furthermore, boundedness is not decidable in Petri nets working under this firing rule (4.1) .

3. The computational power of Petri nets under the Maximum Firing Strategy.

3.1 The Maximum Firing Strategy is more "selective" than the common firing rule, thus we have $M^{MAX}_{\mathcal{N}, X} \subseteq M_{\mathcal{N}, X}$. By this selecting, the Maximum Firing Strategy is more powerful with respect to computations:

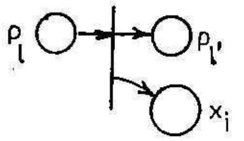
Let $\mathfrak{M}$ and $\mathfrak{M}^{MAX}$ be the classes of all Petri net predicates $M_{\mathcal{N}, X}$ and $M^{MAX}_{\mathcal{N}, X}$, respectively. Then we have

$$\mathfrak{M} \subsetneq \mathfrak{M}^{MAX} .$$

For the proof we refer to 2.2 and 2.4 : If $n = 1$, then we are able to fire in the net $\mathcal{N}'$ constructed in 2.2 exactly all sequences of $L_{\mathcal{N}}$ even under the Maximum Firing Strategy, and hence $M^{MAX}_{\mathcal{N}', X} = M_{\mathcal{N}, X}$. For the example 2.4 we have $M^{MAX}_{\mathcal{N}, X} \in \mathfrak{M}^{MAX} \setminus \mathfrak{M}$.

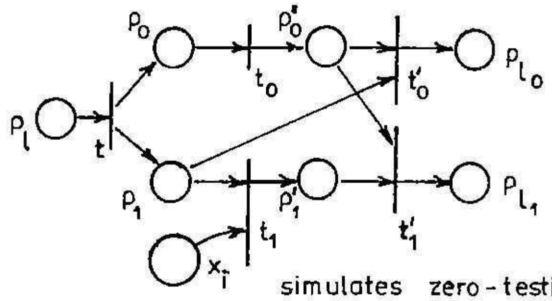
3.2 Petri nets under the Maximum Firing Strategy are able to simulate deterministic counter machines. Other possibilities to simulate counter machines by modified Petri nets were given by several authors (cf. 3.5) . The crucial point is the simulation of zero-testing, which is not possible in Petri nets working under the common firing rule /K2 The consequence of the ability to simulate counter machines are the undecidability results given in 4.1 . The instructions of a deterministic counter machine can be simulated by Petri nets working under the Maximum Firing Strategy in the following way (for more details the reader is referred to the literature):

simulates "start in state l"



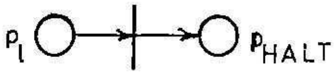
simulates "l: $x_i := x_i + 1$; goto l' ;"

(counters are simulated by the places x_i)



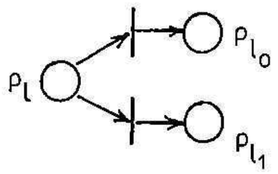
simulates zero-testing :

"l: if $x_i = 0$ then goto l_0 else $x_i := x_i - 1$; goto l_1;"



simulates "l: halt"

Non-deterministic counter machines may also be simulated if we make use of the additional choice-construction:



simulates "l: goto l_0 or l_1 ;"

3.3 As it was shown in /B1/, the set $\{ (i,2^i) \ / \ i \in \mathbb{N} \}$ is not in the class $\mathcal{M}^{MAX}$, and hence it is not possible to compute all recursively enumerable predicates in the sense of $\mathcal{M}^{MAX}$. To do this, termination is needed: Only those computations (markings) are valid for which a given submarking $y \in \mathbb{N}^{P \setminus X}$ is reached on the places of the set $P \setminus X$ (where X denotes the places on which the predicate is computed as before). By such predicates

$$M^{MAX}_{\mathcal{N},X,y} := \{ x \in \mathbb{N}^X \ / \ \exists \, m \in R^{MAX}_{\mathcal{N}} \ \forall p \in X \ \forall p' \in P \setminus X : \ m(p)=x(p) \wedge m(p')=y(p') \}$$

all recursively enumerable predicates can be represented /B1/.

Remark: It is an open problem which predicates can be represented using termination in Petri nets under the common firing rule. But it is conjectured that not all recursively enumerable predicates can be generated in this way.

3.4 The order of transitions in a firing sequence of $L_{\mathcal{N}}$ may be artificial in the case of concurrently firable transitions. For reasons of comparing results we can also introduce such an artificial order for the firings under the Maximum Firing Strategy: For each maximal set T' of simultaneously firable transitions the transitions of T' may fire in an arbitrary order (each transition exactly once before the next set T' is chosen). Then we obtain that the Maximum Firing Strategy is more powerful also with respect to the representation of languages by Petri nets. Using termination and a transition labelling function (homomorphism) $h: \ T \longrightarrow \Sigma \cup \{e\}$ we can generate all recursively enumerable languages over the alphabet Σ /B1/.

3.5 The power of counter machines is also met by the modified Petri net versions given by several authors. In /H2/ inhibitor arcs and priorities for transitions, respectively, are used. In the nets defined in /JLL/ and /MATTK/ the transitions have to fire during fixed (individual) time intervals after their enabling. Concepts of firing in the order of enabling (realized by certain queue regimes) have thos effects, too /B1/. In /V/ the numbers of transported tokens are modifie by the markings on certain places. The concept in /MPS/ is the closest one to our Maximum Firing Strategy: There the firings of transitions are synchronized by external events such that all enabled transitions connected to the actual event have to fire. The construction for the simulation of inhibitor arcs given in /MPS/ would also work under the Maximum Firing Strategy. But, on the other hand, the construction give there is quite opposite to parallelism since all concurrent firings of the essential transitions are suppressed by a run loop (similar to the construction in 2.2).

4. "The price of parallelism"

4.1 It is well established in the literature that the ability of Petri nets to simulate deterministic counter machines (whereby the Petri nets are modified in some sense) results in the undecidability of the boundedness (are all reachable markings bounded with respect to certain places), the reachability (is a given marking/submarking reachable) and the liveness (can certain transitions always become firable sometime later) problems. Since the halting problem is not decidable for deterministic counter machines, it is not decidable if a token can arrive at the place p_{HALT} (cf.3.2) and hence the reachability problem for submarkings is undecidable. By connecting certain simple subnets to the place p_{HALT} the undecidability of the reachability, boundedness and liveness problems can be proved (cf. for instance /H1/, /JLL/, /B1/).

4.2 In the constructions for the simulation of the counter machines all places excluding the counter-place x_i may only be marked by 0 or 1. It is known from the theory that the halting problem is undecidable even for counter machines with two counters. Hence the undecidability results hold for Petri nets under the Maximum Firing Strategy where the nets have only two unbounded places. By a construction given in /B1/ the number of bounded places can also be limited by two. Hence the total number of places need not be greater than four. On the other hand, the reachability sets R_N of Petri nets with 4 places under the common firing rule are always semilinear /HP/. This illustrates the difference between the firing rules once more.

4.3 Since we always have $M^{MAX}_{\mathcal{N}, X} \subseteq M_{\mathcal{N}, X}$, a place which is bounded under the common firing rule must also be bounded under the Maximum Firing Strategy. Hence it is possible that a place which was formerly unbounded becomes bounded under the Maximum Firing Strategy. But it is not decidable in general if this happens. Still more important could be the fact that a transition which was live under the common firing rule may become not live with respect to the Maximum Firing Strategy and vice versa /B1/. Here the undecidability results are very strongly affecting the practical use.

4.4 In the Maximum Firing Strategy we make use of maximal parallelism. But for simulating the counter machines the parallelism which is used may also be restricted: Only the parallelism of two transitions is needed for the zero-testing device (3.2): If the transition t_1 is firable (if there are tokens on the place x_i), then the transition t'_o must not become firable (the places p_1 and p'_o must not be marked at the same time). That can be ensured if the transition t_1 must start working (with taking the token from p_1) before the transition t_o has ended its actions (has given the token to p'_o). This condition can be satisfied if we claim that in a net simulating a deterministic counter machine (3.2) at least two transitions have to fire simultaneously whenever this is possible. Moreover, both transitions t_o and t_1 become firable at the same time. Hence it should be reasonably accepted that under the assumptions of a parallel system both transitions are simultaneously acting in reality. Thus deterministic counter machines can be simulated. In this sense we can state that the use of parallelism (the priority of parallelism) must be paid by the undecidability of the reachability, boundedness and liveness problems.

4.5 Of course, the constraints of firings by parallelism of at least two transitions (as far as it is possible) lead also to more computational power (in comparison to the common firing rule as in 3.1). The consequence of those extensions is the impossibility of faithful simulations by all one-processor-computations: There are concurrent computations by nets working under constraints by parallelism such that no net working under the common firing rule can exactly simulate them.

4.6 As it was pointed out in 3.5, all known related extensions of Petri nets (together with termination, cf. 3.3, 3.4) give the nets the power of Turing machines. What we can say now is that already the use of parallelism can give the nets this computational power. The restrictions of this power under the common firing rule result from this point of view from irresolution with respect to parallelism. On the other hand, firing by fair scheduling (sequentializing instead of parallelism results in the power of Turing machines, too /B1/ (consider the zero-testing device in 3.2 under the assumption that a firable transition has to fire which was enabled the longest time).

Thus the restricted computational power (and the decidability of the boundedness problem, for instance) of Petri nets under the common firing rule can be understood as the consequence of allowing "too much": If there is made a decision concerning parallelism (or fair scheduling or one of the modifications mentioned in 3.5), than these restrictions may be overcome.

5. Conclusions.

The restrictions of firability by the use of parallelism extends the computational power of Petri nets. It is not possible to simulate all these computations by nets working under the common firing rule: The nets under the common firing rule are in general computing "too much". Hence there are concurrent computations executed by several processors using the possibilities of parallel working which cannot be exactly reflected by all computations which one processor could execute in the same system or even in any other system of the same kind.

Under the aspects of practical use the power of Turing machines (or at least of deterministic counter machines) may not be welcome. The decidability of liveness, for instance, is desirable. On the other hand, the use of parallelism as far as it is possible is desirable with respect to efficiency, too. Now the question arises for which classes of nets the mentioned problems (or at least some of them) are decidable with respect to the modified firing rules. A positive answer can trivially be given for the class of bounded nets.

Acknowledgement

The inspiration for the study of the Maximum Firing Strategy was given by Prof. A. Salwicki. I would like to thank him for his suggestions.

References

/B1/ Burkhard, H.D., Ordered Firing in Petri Nets, Elektron.Informationsverarb.Kybernetik 17(1981)2/3,71-86.

/B2/ Burkhard, H.D., Two Pumping Lemmata for Petri Nets. To appear in Elektron.Informationsverarb.Kybernetik.

/H1/ Hack, M., Decision Problems for Petri Nets and Vector Addition Systems, MAC-TM 59, Proj.MAC, M.I.T. 1975.

/H2/ Hack, M., Petri net languages, CSG Memo 124, .Proj.MAC, M.I.T. 1975.

/HP/ Hopcroft, J., Pansiot, J.J., On the reachability problem for 5-dimensional vector addition systems, Theor.Comp.Science 8(1979), 135-159.

/JLL/ Jones, N., Landweber, L., Lien, E., Complexity of some problems in Petri nets, Theor.Comp.Science 4(1977),277-299.

/KM/ Karp, R.M., Miller, R.E., Parallel Program Schemata, Journ. Comp. and System Sciences 3(1969), 147-195.

/K1/ Keller, R.M., Vector replacement systems: a formalism for modelling asynchronous systems, Tech.Rep.117, Comp.Science Lab., Princeton Univ., 1972/74.

/K2/ Keller, R.M., Generalized Petri nets as models for system verification, Tech.Rep.202, Dept.Electrical Eng., Princeton Univ., 1975.

/MATTK/ Mori, M., Araki, T., Taniguchi, K., Tokura, N., Kasami, T., Some decision problems for time Petri nets and applications to the verification of communication protocols, Trans.IECE'77/10 Vol.J60-D, No.10, 822-829.

/MPS/ Moalla, M., Pulou, J., Sifakis, J., Synchronized Petri nets: A model for the description of non-autonomous systems, Lecture Notes in Computer Science 64(1978), 374-384.

/SM/ Salwicki, A., Müldner, T., On algorithmic properties of concurrent programs, Manuscript, Warsaw 1979.

/V/ Valk, R., Self-modifying nets, a natural extension of Petri nets, Lecture Notes in Computer Science 62(1978), 464-476.