Ordered Firing in Petri Nets

By Hans-Dieter Burkhard

Abstract. Two simple rules for solving conflicts and for ordering the firings of transitions in a Petri net are studied:
1. The "Maximum Strategy" (Salwicker, Müldner) whereby maximal sets of simultaneously firable transitions are fired.
2. Firing in the order of enabling of transitions by some queue regimes.
In both cases the computational power of Petri nets is extended up to the power of counter machines. As a consequence, the reachability, boundedness and liveness problems are all undecidable if the firing of transitions is ordered by the Maximum Strategy or by the considered queue regimes.

Some extensions of Petri nets have been studied ([1], [4], [5], [6], [7], [9]) which lead to more computational power, especially to the power of counter machines. We are not sure whether more power is a really desirable thing because it can make problems like reachability, boundedness and liveness undecidable. But, nevertheless we want to know for which cases such effects could arise.

Inhibitor arcs ([1], [4]) and fixed priorities for transitions ([4]) should be mentioned at first. Fixed delays are considered in [5] and [7]: transitions can only fire at certain time points (given by fixed intervals) after they become enabled. In [9] the number of transported tokens may be modified by contents of other places. The concept in [6] is the nearest one to our Maximum Strategy. But, while in [6] the synchronization is given by "external events", maximal sets of simultaneously firable transitions are chosen under the Maximum Strategy.

In this paper we show that some very natural assumptions about the firing of transitions give the Petri nets the power of counter machines:
1. The Maximum Strategy: Maximal sets of simultaneously firable transitions are chosen such that the transitions of those sets have to fire "together".
2. Queue regimes: The transitions are fired in that order in which they became enabled.

The Maximum Strategy is related to maximal steps in condition-event-systems. Its study was suggested by the MAX-semantics for concurrent computations by Salwicker and Müldner [8]. The extended computational power under the Maximum Strategy has consequences for this theory: There may be certain concurrent computations which cannot faithfully be represented by single-processor-computations.

Firing in the order of enabling is a concept for ensuring fairness conditions. Additional troubles arise from those transitions which are disabled by firings of other transitions. This is reflected by different kinds of queue regimes.

Our concepts may be considered under the point of view of temporary priorities which depend on the actions of the net. We don’t make use of external invariable priorities (like in [4]).
Such concepts suggest themselves and seem to be a very useful method for organizing the work of concurrent systems. But, as we shall see, this usefulness could be restricted by the undecidability of liveness, reachability and boundedness.

1. Preliminaries

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

A (generalized initial) Petri net is given by
\[ \mathcal{N} = (P, T, F, m_0), \]
where $P$ and $T$ are the finite disjoint sets of places and transitions, respectively. $F : (P \times T) \cup (T \times P) \rightarrow N$ is the flow relation, $m_0 \in N^P$ is the initial marking.

For a transition $t \in T$ we define the vectors $t^-, t^+ \in N^P$ by $t^-(p) := F(p, t)$, $t^+(p) := F(t, p)$ $(p \in P)$. Then a transition $t \in T$ is fireable (under the common firing rule) in a marking $m \in N^P$ iff $t^- \leq m$. After its firing the new marking is $m + At$, where $At := t^+ - t^-.$

A sequence $u = t_1 \ldots t_n \in T^*$ is a firing sequence (under the common firing rule) iff each transition $t_i$ ($i = 1, \ldots, n$) is fireable at $m_0 + \sum_{j=1}^n At_j$, it leads to the new marking $m + \Delta u$, where $\Delta u := \sum_{j=1}^n \Delta t_j$. The set of all firing sequences is denoted by $L_{\mathcal{N}}$.

A marking $m \in N^P$ is reachable (under the common firing rule) iff there is a firing sequence $u \in L_{\mathcal{N}}$ with $m = m_0 + \Delta u$. The set of all reachable markings (the reachability set) is denoted by $R_{\mathcal{N}}$.

The nonterminal Petri net language of $\mathcal{N}$ (under the common firing rule) with respect to a transition labelling function $h : T \rightarrow \Sigma \cup \{ \epsilon \}$ ($\Sigma$ a finite alphabet, $h$ used as a monoid homomorphism) is given by
\[ L_{\mathcal{N}} = \{ h(t_1) \ldots h(t_n) \mid t_1 \ldots t_n \in R_{\mathcal{N}} \}. \]

The terminal Petri net language of $\mathcal{N}$ (under the common firing rule) with respect to a transition labelling function $h$ and a final submarking $y \in N^X$ on the places of a given set $Y \subseteq P$ is defined by
\[ L_{\mathcal{N}, h, y} := \{ h(u) \mid u \in L_{\mathcal{N}} \land (m_0 + \Delta u) (p) = y(p) \text{ for all } p \in Y \}. \]

By $\mathcal{L}$ ($\mathcal{L}_0$) we denote the class of all nonterminal (terminal) Petri net languages. The nonterminal Petri net predicate of $\mathcal{N}$ (under the common firing rule) with respect to a given set $X \subseteq P$ is the projection from $R_{\mathcal{N}}$ on the places of $X$:
\[ M_{\mathcal{N}, X} := \{ x \in N^X \mid \exists m \in R_{\mathcal{N}} : m(p) = x(p) \text{ for all } p \in X \}. \]

The terminal Petri net predicate of $\mathcal{N}$ (under the common firing rule) with respect to $X \subseteq P$ and with respect to a final submarking $y \in N^P$ on a set $Y = P \setminus X$ is given by
\[ M_{\mathcal{N}, X, y} := \{ x \in N^X \mid \exists m \in R_{\mathcal{N}} : m(p) = x(p) \text{ for all } p \in X \text{ and } m(q) = y(q) \text{ for all } q \in Y \}. \]

By $\mathcal{M}$ ($\mathcal{M}_0$) we denote the class of all nonterminal (terminal) Petri net predicates.

An idea of the computational power of Petri nets under the common firing rule without termination is given by the following "pumping lemmata" [3]:

1. There are numbers $k, l$ for each language $L_{\mathcal{N}} \in \mathcal{L}$ 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 $u = u_1u_2u_3$ such that $1 \leq \text{length of } u_2 \leq l$ and $u_4u_5^{n+1}u_6 \in L_{\mathcal{N}}$ for all $n \in N$. 
2. There are vectors \( y', y'' \in N^X \) for each set \( M_{x',x} \in \mathcal{M} \) such that the following holds: If a vector \( x \in M_{x',x} \) covers \( y' \) (i.e. \( x \supseteq y' \)) then there exists a vector \( z \in (N \setminus \{0\})^X \) such that \( z \subseteq y \) and \( x + n \cdot z \in M_{x',x} \) for all \( n \in N \).

Let \( \mathcal{L}_{r.e.}, \mathcal{M}_{r.e.} \) denote the set of all recursively enumerable languages (predicates). Clearly, \( \mathcal{L}_0 \subseteq \mathcal{L}_{r.e.} \), and \( \mathcal{M}_0 \subseteq \mathcal{M}_{r.e.} \), but it is unknown whether these inclusions are proper or not.

We have \( \mathcal{L} \subseteq \mathcal{L}_0 \) and \( \mathcal{M} \subseteq \mathcal{M}_0 \), where the problem whether \( \mathcal{M} = \mathcal{M}_0 \setminus \{\emptyset\} \) is open. This problem might be very hard since the reachability problem is decidable with the help of a constructive proof for \( \mathcal{M} = \mathcal{M}_0 \setminus \{\emptyset\} \) (to each net \( N \) terminally generating a non-empty set \( M_{x',x,y} \), one had to construct another net \( N' \) which non-terminally generates \( M_{x',x,y} \) [3]).

2. Firing under the Maximum Strategy

Differing from the usual firing rule we demand from all firable transitions that they are to fire “together” as far as it is possible (limitations may arise by conflicts). Thus we make use of the maximal possible parallelism. This is related to maximal steps in condition event systems and also to the MAX-semantics for concurrent computations introduced by Salwicki and Müldner [8].

This concept is formalized by a new firing rule which we call the Maximum Strategy:

In a marking \( m \) we choose a maximal set \( T' \) of simultaneously firable transitions, i.e.

\[
\sum_{t \in T'} t^- \leq m \quad \text{and} \quad \sum_{t \in T''} t^- \equiv m \quad \text{for all} \ T'' \supseteq T'.
\]

Then the transitions of \( T' \) are fired (each \( t \in T' \) exactly once). After these firings a new set \( T'_1 \) is chosen for the marking \( m + \sum_{t \in T'} At \) and so on.

From the point of view of the maximal possible parallelism the transitions of a maximal set \( T' \) of simultaneously firable transitions should fire in parallel. But, although the commonly used firing sequences are artificial with respect to concurrency, the introduction of such an artificial sequentializing is useful for theoretical studies of the behaviour of nets. Hence we make the following (theoretical) conventions: After a maximal set \( T' \) of simultaneously firable transitions is chosen, the transitions of \( T' \) are fired in an arbitrary sequential order (each transition exactly once before the next set \( T' \) is chosen).

Thus a sequence \( u \in T^* \) is a firing sequence under the Maximum Strategy iff it can be fired under these conventions. The set of all firing sequences under the Maximum Strategy is denoted by \( L^\text{MAX}_{x'} \). (It is the least set containing (1) the empty word \( e \) and (2) a sequence \( ut \in T^* \) if (a) \( u \in L^\text{MAX}_{x'} \) and (b) \( t \) is chosen from the actually chosen maximal set \( T' \) of simultaneously firable transitions, whereby \( t \) has not already fired after the choice of \( T' \). — In the beginning the set \( T' \) had to be chosen with respect to the initial marking \( m_0 \).)

A marking \( m \) is reachable under the Maximum Strategy iff there is a firing sequence \( u \in L^\text{MAX}_{x'} \) with \( m = m_0 + Au \). Thus the set \( R^\text{MAX}_{x'} \) of all reachable markings under the Maximum Strategy contains all markings \( m + \sum_{t \in T''} At \) for sets \( T'' \subseteq T' \), where \( T' \) is a maximal set of simultaneously firable transitions chosen in the marking \( m \).

It means that \( L^\text{MAX}_{x'} \) contains all markings which may be eventually reached when some transitions of a set \( T' \) have fired while others have not. From a practical point of view it might be better to consider markings \( m - \sum_{t \in T'} t^- + \sum_{t \in T'} t^+ \), or even only
the markings \( m + \sum_{t \in T'} \Delta t \), which are reached after performing the actions on the set \( T' \). To have the correspondence to the sets \( L_{\nu}' \) we prefer the definition of \( R_{\nu}' \) as given above. Nevertheless, the main results of this paper concerning (sub-)markings remain valid if we consider only those markings that are reached when all transitions of a set \( T' \) have fired.

Now the nonterminal/terminal Petri net languages/predicates \( L_{\nu}' \), \( L_{\nu}', h, y \), \( M_{\nu}', x \), \( M_{\nu}', x, y \) under the Maximum Strategy are defined in the same way as \( L_{\nu}, h \), \( L_{\nu}, h, y \), \( M_{\nu}, x \), \( M_{\nu}, x, y \): In the related definitions \( L_{\nu}' \) and \( R_{\nu}' \) have to be replaced by \( L_{\nu}' \) and \( R_{\nu}' \).

By \( \mathcal{L}_{\nu}' \), \( \mathcal{M}_{\nu}' \), \( \mathcal{W}_{\nu}' \), \( \mathcal{W}_{\nu}' \) we denote the classes of all nonterminal/terminal Petri net languages/predicates under the Maximum Strategy, respectively.

To show that the Maximum Strategy is more powerful with respect to computations than the common firing rule we \\examine the example of Fig. 1 (a modified version of the well-known net for the weak computation of \( 2^i \)). Another example which computes squares was examined in [2].

\[ \text{In this net a while-loop is realized under the Maximum Strategy: For } m(y_1) = m(y_2) = 1 \text{ and } m(p_1) > 0 \text{ we have to choose } T' = \{ t_2, t_3 \} \text{ and then we have to fire } t_1, t_2 \text{ or } t_3, t_2. \text{ Hence the transition } t_4 \text{ cannot fire as long as there are tokens in the place } p_1. \text{ Thus we have under the Maximum Strategy:} \\
\text{while } m(p_1) > 0 \text{ do begin } m(p_1) := m(p_1) - 1; \\
m(p_2) := m(p_2) + 2; \\
m(x_2) := m(x_2) + 1 \text{ end.}
\]

Another while-loop is realized by the transitions \( t_5, t_6 \).

Now it is not difficult to verify that we have for \( X := \{ x_1, x_2 \}, Y := \{ y_1, \ldots, y_4 \}, y := (0, 0, 0, 2) \) and \( h(t_i) := A, h(t_5) := B, h(t_6) := e \) for \( i = 3, \ldots, 6 \):

- \( L_{\nu}', h = \{ u \mid u \subseteq \text{ABAB}^2\text{AB}^4\text{AB}^8\text{AB}^{16} \ldots \} \),
- \( L_{\nu}', h, y = \{ e, A, \text{ABAB}^2, \text{ABAB}^2\text{AB}^4, \text{ABAB}^2\text{AB}^4\text{AB}^8, \ldots \} \),
- \( M_{\nu}', x = \{ (i, j) \mid i \geq 0 \land 2^{i-1} \leq j \leq 2^i \} \),
- \( M_{\nu}', x, y = \{ (i, j) \mid i \geq 0 \land j = 2^i \} \).

Since \( L_{\nu}', h \) and \( M_{\nu}', x \) do not satisfy the related pumping lemma we have here

\( L_{\nu}', h \notin \mathcal{L}_{\nu}' \) and \( M_{\nu}', x \notin \mathcal{W}_{\nu}' \).
This shows that there are computations under the Maximum Strategy which cannot faithfully be represented by any computation under the common firing rule. Hence there are systems realizing parallelism by several processors in a related meaning such that their actions cannot exactly be simulated by a system with only one processor. In this interpretation the one-processor-systems do in general "too much" since we always have

\[
L^\text{MAX}_{\mathcal{N}^\prime, h} \subseteq L_{\mathcal{N}^\prime, h}, \quad L^\text{MAX}_{\mathcal{N}^\prime, h,y} \subseteq L_{\mathcal{N}^\prime, h,y}, \\
M^\text{MAX}_{\mathcal{N}^\prime, x} \subseteq M_{\mathcal{N}^\prime, x}, \quad M^\text{MAX}_{\mathcal{N}^\prime, x,y} \subseteq M_{\mathcal{N}^\prime, x,y}.
\]

Up to now it is not known if termination extends the computational power of Petri nets under the common firing rule such that \(L^\text{MAX}_{\mathcal{N}^\prime, h}, L^\text{MAX}_{\mathcal{N}^\prime, h,y} \in \mathcal{L}_0\) and \(M^\text{MAX}_{\mathcal{N}^\prime, x}, M^\text{MAX}_{\mathcal{N}^\prime, x,y} \in \mathcal{M}_0\), respectively. It is also unknown if a \textbf{while}-loop may be realized in a Petri net using termination under the common firing rule. The example shows that it cannot be possible without termination under the common firing rule.

The next theorems and their proofs give some impressions about the extensions of the computational power under the Maximum Strategy.

**Theorem 1.** \(\mathcal{M} \subseteq \mathcal{M}^\text{MAX} \subseteq \mathcal{M}^\text{MAX}_0 = \mathcal{M}_\text{r.e.}\).

(The position of \(\mathcal{M}^\text{MAX}_0\) is unknown: \(\mathcal{M}^\text{MAX}_0 \not\equiv \mathcal{M} \cup \{\emptyset\}\) and \(\mathcal{M}^\text{MAX}_0 \not\equiv \mathcal{M}_\text{r.e.}\), respectively, are both open problems.)

**Proof.**

a) \(\mathcal{M} \subseteq \mathcal{M}^\text{MAX}\): To each net \(\mathcal{N} = (P, T, F, m_0)\) we can add a run loop as in Fig. 2. In the new net \(\mathcal{N}'\) with the run loop at most one transition is firable each time. Hence both, the common firing rule and the Maximum Strategy, do the same with respect to the firings in the part of the original net \(\mathcal{N}\). Thus we have \(M^\text{MAX}_{\mathcal{N}', x} = M_{\mathcal{N'}, x}\)

for arbitrary sets \(X \subseteq P\).

b) \(\mathcal{M} \equiv \mathcal{M}^\text{MAX}\) by the example.

c) \(\mathcal{M}^\text{MAX} \subseteq \mathcal{M}^\text{MAX}_0\) since \(\mathcal{M}^\text{MAX}_0 = \mathcal{M}_\text{r.e.}\) (see under e)).

We can also use a construction like in [4] for clearing places after switching off all computations of the original net by a stop transition in the way as shown in Fig. 3.

d) \(\mathcal{M}^\text{MAX} = \mathcal{M}^\text{MAX}_0\): The set \(\{i, 2^i\} \mid i \in \mathcal{N}\) is contained in \(\mathcal{M}^\text{MAX}_0\) by the example, but not in \(\mathcal{M}^\text{MAX}\): A transition leading from \(i, 2^i\) to \((i + k, 2^{i+k})\) cannot be applicable to another pair \((j, 2^j)\) since it would lead to a submarking \((j + k, l)\) with \(l = 2^{i+k}\).

Thus we would need infinitely many transitions. — By this argument no set \(\{(i, f(i)) \mid i \in \mathcal{N}\}\) with a not linearly majorizable function \(f\) can be in the class \(\mathcal{M}^\text{MAX}\).

**Fig. 2**

**Fig. 3**
e) \( \mathcal{W}_{t.e.}^{MAX} = \mathcal{W}_{r.e.}^{MAX} \): We show that we can simulate non-deterministic counter machines \( \mathcal{C} = (S, C, I) \) (cf. [4]) by Petri nets under the Maximum Strategy. A state \( s \in S \) is simulated by a token in a related place \( p_s \). The counters \( c \in C \) are simulated by (unbounded) places \( p_c \). The instructions of the set \( I \) are simulated as in Fig. 4. Then we have a one-to-one correspondence between the situations \( (s, m^*) \), \( s \in S \), \( m^* \in \mathcal{N}^C \), of the counter machine \( \mathcal{C} \) and the markings \( m \) of the simulating net where the place \( p_s \) is marked and we have \( m^*(c) = m(p_c) \) for all \( c \in C \). Since each set \( M \in \mathcal{W}_{t.e.}^{MAX} \) can be computed with termination ("halt") in a non-deterministic counter machine on the counters of a set \( C' \subseteq C \), we can compute \( M = M_{MAX}^{N}, \mathcal{X}, \gamma \) in the simulating net \( \mathcal{N} \) with \( X = \{ p_c | c \in C' \}, \gamma = \{ p_{HALT} \}, \gamma(p_{HALT}) = 1 \). □

\[
\begin{align*}
\text{Fig. 4}

\text{Theorem 2.} \quad \mathcal{L} \sqsubseteq \mathcal{L}_{MAX} \sqsubseteq \mathcal{L}_{t.e.} \sqsubseteq \mathcal{L}_{r.e.}. \\
(\text{The position of } \mathcal{L}_{0} \text{ is unknown except that } \mathcal{L} \sqsubseteq \mathcal{L}_{0} \sqsubseteq \mathcal{L}_{t.e.})
\]

\text{Proof. For } \mathcal{L} \sqsubseteq \mathcal{L}_{MAX} \sqsubseteq \mathcal{L}_{MAX} \text{ the proof is the same as for Theorem 1 (a), (b), (c)).}

\text{d) } \mathcal{L}_{MAX} \sqsubsetneq \mathcal{L}_{MAX}^{t.e.}: \text{The language } \{ e, AB, ABAB^2, ABAB^3AB^4, \ldots \} \text{ is contained in } \mathcal{L}_{MAX}^{t.e.} \text{ by the example, but not in } \mathcal{L}_{MAX}^{t.e.}: \text{Otherwise it would be implied that } \{ (i, 2^i) \mid i \in \mathcal{N} \} \in \mathcal{L}_{MAX}^{t.e.} \text{ (cf. the proof of Theorem 1, d)) by counting the firings of all transitions labelled with } A \text{ and } B \text{ in a related net on two places } p_A \text{ and } p_B, \text{ respectively.}

\text{e) } \mathcal{L}_{MAX}^{t.e.} = \mathcal{L}_{r.e.} \text{ follows from the simulation of non-deterministic counter machines with corresponding labelled transitions for print statements. All the other transitions are labelled by the empty word. □}

According to the modified firing rule, the reachability problem in Petri nets under the Maximum Strategy is the problem if a given (sub-)marking is contained in the set \( R_{MAX}^{t.e.}(M_{MAX}, \mathcal{X}) \). Similarly, the coverability problem is to be understood as the problem...
Ordered Firing in Petri Nets

if a given (sub-)marking may be covered by some marking in $R_{\mathcal{N},X}^{\text{MAX}} (M_{\mathcal{N},X}^{\text{MAX}})$. The boundedness problem for some set $X \subseteq P$ is the problem if the set $M_{\mathcal{N},X}^{\text{MAX}}$ is bounded. Finally, the liveness problem for a set of transitions of a Petri net under the Maximum Strategy is the problem if for every marking $m \in R_{\mathcal{N},X}^{\text{MAX}}$ these transitions may always become firable sometime later (a transition $t$ is live under the Maximum Strategy iff for every $u \in \mathcal{L}_{\mathcal{N}}^{\text{MAX}}$ there exists some $u'$ such that $uu't \in \mathcal{L}_{\mathcal{N}}^{\text{MAX}}$).

Theorem 3. The reachability, boundedness, coverability and liveness problems are all undecidable for Petri nets under the Maximum Strategy.

Proof. By the given constructions we can also simulate deterministic counter machines by Petri nets under the Maximum Strategy. Since the halting property is not decidable for deterministic counter machines (with counters initially 0), it is not decidable for our simulating nets if a token reaches the place $p_{\text{HALT}}$.

We consider extensions of such nets as given in Fig. 5 (cf. the similar constructions in [5]). Then we have:

![Diagram](image)

Fig. 5

The deterministic counter machine simulated by $\mathcal{N}$ halts iff the place $p_{\text{HALT}}$ can be marked by one token in $\mathcal{N}$.

a) — iff the marking $m$ with $m(p_{\text{HALT}}) = 1$ and $m(p) = 0$ for all $p \in P \setminus \{p_{\text{HALT}}\}$ is reachable and coverable in the net of Fig. 5a);

b) — iff the place $p^*$ (which counts the number of firings of all transitions) is bounded in the net of Fig. 5b),

— iff all places are bounded (i.e. the net is bounded) in the net of Fig. 5b);

c) — iff the transition $t^*$ is live in the net of Fig. 5c),

— iff all transitions are live (i.e. the net is live) in the net of Fig. 5c), whereby the numbers $K$ of transported tokens must be large enough.
Hence the undecidability of the halting property for deterministic counter machines implies the undecidability of reachability, coverability, boundedness and liveness for Petri nets working under the Maximum Strategy. Note that by Theorem 1 there are single nets for which the reachability of markings is undecidable if we use the Maximum Strategy. □

Since firings under the Maximum Strategy are restricted with respect to the common firing rule, it is only possible that a place which was unbounded under the common firing rule becomes bounded under the Maximum Strategy (or it remains unbounded). Hence the undecidability of boundedness under the Maximum Strategy (while boundedness is decidable under the common firing rule) is connected with those places which are no longer unbounded if we restrict the firings by the Maximum Strategy.

Similarly we have $R_{\mathcal{C}}^{\text{MAX}} \subseteq R_{\mathcal{C}}$. Hence, if we claim that the reachability problem is decidable under the common firing rule, the undecidability result under the Maximum Strategy arises from the markings $m \in R_{\mathcal{C}} \setminus R_{\mathcal{C}}^{\text{MAX}}$ (note that the set $R_{\mathcal{C}}^{\text{MAX}}$ is recursively enumerable).

Those markings are also responsible for the undecidability of the boundedness and coverability problems. (We could also say that the suppression of the sequences from $L_{\mathcal{C}} \setminus L_{\mathcal{C}}^{\text{MAX}}$ leads to these results.)

A transition which was live under the common firing rule may become not live under the Maximum Strategy and vice versa as in the example of Fig. 6 where the transitions $t_2$ and $t_3$ are live and the transition $t_4$ is not live under the Maximum Strategy while $t_4$ is live and $t_1$, $t_2$ are not live under the common firing rule.

Another possibility to simulate zero-testing under the Maximum Strategy is given by the construction shown in Fig. 7. If we use appropriate different numbers $m$ and $n$ and different corresponding transition sets, we can simulate all zero-tests of a net by this construction. Moreover, the both unnamed places can also be used for describing the states of a counter machine. Hence we need only two bounded places. On the other hand, the halting property of deterministic counter machines is undecidable even for machines with only two counters. Thus, the reachability, boundedness and liveness problems are undecidable even for Petri nets with four places under the Maximum Strategy.

3. Firing by Queue Regimes

The Maximum Strategy may be considered under the point of view that temporary priorities are given to the transitions of the chosen maximal set of simultaneously
firable transitions. Another concept of temporary priorities is that of queues for realizing firings in the order of enabling (in some different meanings).

A queue \( q \) is a word over \( T \) where each transition may occur at most once (hence the length of the queue is bounded by the cardinality of \( T \)). By \( T(q) \) we denote the set of all transitions belonging to the queue. For \( q = t_1 \ldots t_n \) the transition \( t_i \) is at the top and the transition \( t_n \) is at the end of the queue. Initially the queue is formed from all transitions that are firable at \( m_0 \). The order in the initial queue is arbitrary. In each situation \( (m, q) \), where \( m \) is the actual marking and \( q \) is the actual queue, only that transition \( t \) can be fired which is firable at \( m \) and which is the nearest one to the top of the queue. After the firing of \( t \) the queue must be reorganized with respect to the new marking \( m + \Delta t \):

a) The transition \( t \) is deleted from the queue.

b) All transitions which are firable at \( m + \Delta t \) and which are not already in the queue (not in \( T(q) \setminus \{t\} \)) have to be added to the end of the queue in an arbitrary order (the fired transition \( t \) is among them if it is firable at \( m + \Delta t \)). — Note that the information about \( T(q) \) may be stored outside the queue such that the queue must not be checked for this point of reorganization.

c) With respect to those transitions in the queue which are not firable at \( m + \Delta t \) we have different possibilities. It is a special property of Petri nets that enabled transitions (processes) may loose their concession by firings of other transitions before they could work. Usually in the considerations of fairness problems in concurrent systems such effects are not regarded.

We propose three kinds of queue regimes which are different with respect to their handling and their effects (firing in the order of enabling):

Queue 1. Those transitions (the transitions of \( T(q) \setminus \{t\} \)) which are not firable at \( m + \Delta t \) are deleted from the queue.

— In the actual queue (after reorganization) we have exactly all transitions which are firable at the actual marking. The reorganization needs a checking of the whole queue for deleting the disabled transitions.

Queue 2. Those transitions from the top up to the first enabled transition are deleted.

— This queue is easier to handle. The transitions at the top are checked one by one: If the top transition is not firable, then it is deleted. Otherwise it is fired and the queue is reorganized as under a) and b).

Queue 3. The queue is not changed with respect to those transitions.

— A transition remains in the queue at its relative position until it becomes the first firable one. Then it is fired and deleted from the queue (by a) (but it can be among those transitions which are added to the end of the queue (by b)). The reorganization is easy, but checking of the queue for the first firable transition is needed: There may be transitions at the top which are disabled for a long time (or even forever).

A sequence \( u \in T^* \) is a firing sequence under Queue \( i \) \( (i = 1, 2, 3) \) if it may be fired starting in \( m_0 \) under the regime of Queue \( i \). The set of all firing sequences under this queue regime is denoted by \( L^i_\psi \). The reachability set \( R^i_\psi \) and the nonterminal/terminal Petri net languages/predicates \( L^{Q^i_\psi}_{\psi, h}, L^{Q^i_\psi}_{\psi, h, y}, M^{Q^i_\psi}_{\psi, x}, M^{Q^i_\psi}_{\psi, x, y} \) under Queue \( i \) are defined in the same way as the related sets under the common firing rule by replacing \( L^\psi \setminus R^\psi \) by \( L^i_\psi \setminus R^i_\psi \). Similarly the classes of all nonterminal/terminal Petri net languages/predicates under Queue \( i \) are denoted by \( \mathcal{L}^Q, \mathcal{L}^{Q^i_\psi}, \mathcal{M}^Q, \mathcal{M}^{Q^i_\psi} \), respectively.

The "queue-marking-graph" \( G^Q_\psi \) of the net \( \mathcal{N} = (P, T, F, m_0) \) under Queue \( i \) is a directed graph, where the nodes are the situations \( (m, q) \) which may be reached from
one of the initial situations \((m_0, q_0)\) (remember that the initial queue may be formed from all initially firable transitions in an arbitrary order). An arc labelled by \(t\) leads from \((m, q)\) to \((m + At, q')\) iff \(t\) is the firable transition at \(m\) under the actual queue \(q\) and \(q'\) is a reorganized queue after the firing of \(t\). In general those graphs are infinite and different for different queue regimes working over the same net.

For the net \(\mathcal{N}\) of Fig. 8 where \(q_0 = AB\) and \(q_0 = BA\) are the possible initial queues we have the queue-marking-graphs of Fig. 9 \((G_{\mathcal{N}}^Q = G_{\mathcal{N}}^Q)\) and Fig. 10 \((G_{\mathcal{N}}^Q\), disabled transitions in an actual queue are enclosed in brackets).

Note that none of our queue regimes can ensure that an enabled transition has to fire sometime later (or even after a fixed delay), since an enabled transition may lose its concession by the firings of other transitions. In general those requirements are not satisfiable by firings in arbitrary Petri nets. Only Queue 3 ensures that a transition which is enabled sufficiently often has to fire (at the latest if it has been enabled \(k\) times, where \(k\) is the cardinality of \(T\)). By Queue 1 and Queue 2 it is possible that a transition can never fire even if it is enabled infinitely often as in the example given above. But, as we shall see, for each Queue \(i\) \((i = 1, 2, 3)\) there are certain transitions in certain nets such that these transitions are live under Queue \(i\) and not live under the other queues. Hence it may depend on the nets which queue is to be used.

Since firing under a queue regime is again a restriction of firing under the common firing rule, we have

\[
\begin{align*}
L^Q_{i, h, x} &\subseteq L_{\mathcal{N}, h, x}, & L^Q_{i, h, y} &\subseteq L_{\mathcal{N}, h, y}, & I^Q_i &\subseteq I_{\mathcal{N}}, \\
M^Q_{i, X} &\subseteq M_{\mathcal{N}, X}, & M^Q_{i, x, y} &\subseteq M_{\mathcal{N}, x, y}, & R^Q_i &\subseteq R_{\mathcal{N}}.
\end{align*}
\]
Ordered Firing in Petri Nets

![Diagram of Petri Net]

Fig. 11

![Expanded States and Transitions]

Fig. 12
The sets generated by different queues in the same net are in general incomparable as in the example of Fig. 11.

We shall also use this example in the proof of Theorem 6: There it is connected to the place \( p_{\text{HALT}} \) of a deterministic counter machine whereby the place \( p_5 \) has to be unmarked.

Now, for each possible initial queue \( A_2A_1A_0A_n \) formed from the transitions \( A_1, A_2, A_3, A_4 \), we have in the queue-marking-graphs the parts as shown in Fig. 12 whereby

\[
m = (m(p_k), m(p_1), m(p_m), m(p_n), m(p_3), m(p_0), m(p_f))
\]

Under Queue \( i \) we have for an initial queue \( q_0 \) the firing sequence \( q_0BC_i \) such that

\[
F_{\varphi_i}^{q_0} = \{ q_0 \mid q_0 \text{ is an initial queue} \} \cdot (B) \cdot \{ C_i \}
\]

Hence the sets \( L_{\varphi_i}^{q_0} \) of firing sequences under the different queue regimes are in comparable in this net. The reachability sets and related Petri net languages and predicates are also incomparable.

By connecting appropriate nets \( N_i \) to the transitions \( C_i \) (cf. the related constructions in the proof of Theorem 3) we can show that also the behaviour of different queue regimes may be incomparable with respect to liveness and boundedness in the same net.

Obviously the behaviour of nets under queue regimes may depend on the chosen initial queue. Now, by constructions as in the example, it is possible to construct nets such that a certain ordering of firings is derivable independent of the initially chosen queue \( q_0 \).

The computational power of Petri nets under the queue regimes is again extended with respect to the common firing rule:

**Theorem 4.**

\[
\mathfrak{M} \subseteq \mathfrak{M}_{\varphi_i}^{q_0} \subseteq \mathfrak{M}_{\varphi_i}^{q_0} = \mathfrak{M}_{\text{r.e.}}
\]

\[
F \subseteq F_{\varphi_i}^{q_0} \subseteq F_{\varphi_i}^{q_0} = F_{\text{r.e.}} \quad (i = 1, 2, 3)
\]

**Proof.** For \( i = 1, 2 \) the proof is the same as for the Maximum Strategy (Theorem 1, 2). Only the additional construction (not needed for the proof) in part c) of the proof of Theorem 1 does not remain valid.

For \( i = 3 \) we need several new constructions. We start with the devices for choice and for zero-testing in order to simulate non-deterministic counter machines. The former construction for choice is not useful for simulation under the regime of Queue 3 since the transition not chosen in one run (since it was the second in the queue) is stored at the top of the queue up to the next application of this construction — and then it has to fire absolutely.

We construct the new devices in such a way that all transitions that become enabled during one run have to be fired. Then no transition can remain in the queue, and we have always the same initial situation with respect to the transitions before we want to make use of such a device.

For choice we use the construction shown in Fig. 13 where the choice between \( p_{1i} \) and \( p_{2i} \) depends on the choice of queuing \( t_{1j} \) or \( t_{2j} \) after \( t_1 \) has fired. A choice of \( p_{1i} \) for instance consists of firing the sequence \( t_{1k}t_{2j}t_{1k} \), where we have the successive queues \( t_1, t_{2j} \) (here we could have the other queue \( t_{1k} \) resulting in the choice of \( p_{2i} \)), \( t_{1k}, t_{2j}, t_{1j} \).

The construction shown in Fig. 14 can be used for zero-testing under Queue 3. If there are tokens in the place \( p_0 \), then all transitions except for \( t_3 \) have to fire while \( t_5 \) will not become enabled.

Note that only enabled transitions may be in the queues if we use these constructions for the simulation of choice and zero-testing, respectively, since no enabled
transition can lose its concession by firings of other transitions (under queue regimes).
Hence the firings are the same under all of our queue regimes.

By the same arguments as in the proof of Theorem 1 (e), (e)) we have now \( M^{Q3} \subseteq \subseteq M^{Q3} = M_{r.e.} \) (and similarly for languages).

The construction with run loops (a)) does also not remain valid to prove \( M \subseteq M^{Q3} \).
These difficulties can be overcome by using choice constructions and additional places to the transitions in the way as shown in Fig. 15.

To prove \( M = M^{Q3} + M^{Q3} \) we have to modify the example by other while-loop constructions: We have to replace the transitions \( t_j, j = 2, 3, 5, 6 \), each by two transitions as in Fig. 16.

As another possibility we could use the zero-testing device for the construction of the **while**-loop device under Queue 3. The proof is completed by the same arguments as for Theorem 1 (b), (d)) and for Theorem 2. □

Again we are able to simulate deterministic counter machines. Using the same constructions as for Theorem 3 we can show:

**Theorem 5.** The reachability, coverability boundedness and liveness problems are all undecidable for Petri nets under the regime of Queue \( i \) \( (i = 1, 2, 3) \).
(Here reachability, coverability, boundedness and liveness are defined with respect to firings under Queue \( i \).)

Similar to the results for the Maximum Strategy, the markings in \( R_{V} \setminus R_{V}^{Q3} \) are responsible for the undecidability results. Transitions which are live under the common firing rule may become not live under the queue regimes and vice versa (cf. the example, given for the Maximum Strategy, a related example for Queue 3 can be constructed from the example for the incomparability of the sets \( L_{V}^{Q3} \)). □
The following theorem shows that it is not possible to decide effectively if a queue regime for a given net may be replaced by another queue regime such that the behaviour is preserved in some sense.

**Theorem 6. The containment problems**

\[ L_{Q_i}^N \subseteq L_{Q_j}^N ? \]

and the equivalence problems

\[ L_{Q_i}^N = L_{Q_j}^N ? \]

are all undecidable.

Furthermore the related containment and equivalence problems for nonterminal/terminal Petri net Languages/predicates are all undecidable.

**Proof.** Let \( N \) be a net which simulates a deterministic counter machine by the construction for the zero-testing device given for Queue 3. Then we have \( L_{Q_i}^N = L_{Q_i}^{Q_3} = L_{Q_i}^{Q_3} \) since in such nets each of our queues contains only firable transitions. Now for \( i, j \in \{1, 2, 3\}, i \neq j \), we connect the place \( p_{\text{HALT}} \) to our net \( N' \), where \( L_{Q_i}^{N'} = L_{Q_j}^{N'} \) are incomparable. In the so constructed net \( N'' \) we have \( L_{Q_i}^{N''} = L_{Q_j}^{N''} \) iff no token can reach the place \( p_{\text{HALT}} \), i.e., iff the simulated deterministic counter machine does not halt. Otherwise the sets \( L_{Q_i}^{N''} \) and \( L_{Q_j}^{N''} \) are incomparable. In the same way the undecidability results are proved for the other sets. \( \square \)

By the same methods it can be shown that it is not decidable if a queue regime may be replaced by another queue regime such that certain properties (for instance: liveness, boundedness — cf. the proof for the incomparability of \( L_{Q_i}^{N'}, L_{Q_j}^{N'} \)) are preserved.

There are further possibilities of queue regimes (addition of new transitions only if the queue is empty, remaining of fired transitions in the queue at their position until they are disabled . . . ). But, as far as we have examined it, all possibilities extend the power of the Petri nets up to the power of at least deterministic counter machines. This implies the unsolvability of the liveness, reachability, coverability and boundedness problems.

We remark that all of our queues are finite since each transition occurs at most once. Therefore, only a finite state control is needed, but it must receive total information about each transition whether it is firable or not, and it must be able to control the firings (especially: to paralyze unwanted firings).

**Conclusions**

The firing rule is very sensitive with respect to the extension of the computational power of the Petri nets. Deterministic counter machines can be simulated only with the additional possibility of zero-testing. This results in the undecidability of the liveness, reachability, coverability and boundedness problems. The additional simulation of choice together with termination extends the computational power up to the power of Turing machines.

As it can be seen by the given constructions, the zero-testing needs only a small priority of parallel executions (by only two processors) or a small priority of some fairness conditions with respect to firings in the order of enabling. These effects have to be taken into consideration if one uses such (firing) rules.

The differences to the common firing rule arise from the fact that the modified firing rules are more selective with respect to firable sequences. Especially, if \( m \) covers
a marking $m'$, it is not implied that under the modified rules all fireable sequences at $m'$ are fireable at $m$, too.

Stating that by the modified firing rules fewer sequences are fireable, the difficulties to the decision problems can arise from those sequences which are not fireable any more. If we suppose these sequences to be more complicated to fire (since the priorities given by the modified firing rules seem to be natural), we get an understanding of the high complexity of the known decision procedures.

Acknowledgement

The author is greatly indebted to A. Salwicki who has suggested the study of the Maximum Strategy.

References


Kurzfassung

Zwei einfache Vorschriften zur Entscheidung von Konflikten und zur Einführung einer Reihenfolge für das Feuern der Transitionen eines Petri-Netzes werden untersucht:

1) Die „Maximum Strategie“ (Salwicki, Mülnder), bei der jeweils maximale Mengen von gleichzeitig feuerbaren Transitionen geschaltet werden.

2) Feuern in der Reihenfolge, in der die Transitionen Konzession erhalten haben (Einführung entsprechender Warteschlangen-Regimes).

In beiden Fällen werden Zählerautomaten durch Petri-Netze simulierbar, und demzufolge sind Erreichbarkeit, Beschränktheit und Lebendigkeit nicht mehr entscheidbar.

Резюме

Исследуются два простые правила решения конфликтов и порядка включения переходов в сетях Петри:

1) „стратегия максимума“ [8], при которой на каждом шаге включается максимальное количество переходов, которые могут отрабатывать одновременно;

2) переходы отрабатывают в том порядке, в каком они получают разрешение (режим очереди).
В обоих случаях оказывается, что с помощью сетей Петри можно моделировать счетчиковые машины, следовательно, для них проблемы достижимости, ограниченчности и живучести неразрешимы.

Author's address:
Dr. H.-D. Burkhard
Sektion Mathematik der
Humboldt-Universität zu Berlin
1086 Berlin
PSF 1297
German Democratic Republic

(Received: first version May 28, 1980, revised version October 21, 1980)