ON THE MODELLING POWER OF PETRI NETS

Erik Meiling

TR79-403

December, 1979

Computer Science Department
Cornell University
Ithaca, New York 14853

Permanent address:

Computer Science Department
University of Copenhagen
41 Sigurdsgade
DK-2200 Copenhagen N
Denmark

This work was supported in part by the Danish Natural Science Research Counsil.

On the Modelling Power of Petri Nets

Abstract

The behavior of a Petri net is expressed as a formal language. Certain families of Petri net languages are characterized by propositions similar to the classical pumping theorems. The results are used to give examples of behaviors that cannot be expressed by languages in these families.

Introduction

Petri nets (or equivalently vector addition systems) have been studied as models of asynchronous processes. The language defined by a Petri net (Hack (1975), Peterson (1976), Meiling (1980)) can be viewed as the set of all possible behaviors of the Petri net, and the study of properties of such languages can therefore increase our understanding of what kind of systems it is possible to model with Petri nets. Petri nets are very similar to multicounter automata (Fisher et al. (1968)) with the important difference that a Petri net cannot in general test for zero in its counters. The limitation is referred to as "partial blindness" in Greibach (1978), and in this paper we investigate the consequences of partial blindness.

In Section 2 the basics of Petri nets and Petri net language theory are introduced. In Section 3 we study languages generated by Petri net schemata where the set of final markings forms a principal arcone. Such languages can be "pumped" in a way very similar to the pumping of regular languages. In Section 4 we investigate languages generated by Petri net schemata where the event function is one-to-one. The consequences of partial blindness is expressed in the "postponement" theorem, showing that for certain words in such languages the generation of some subwords may be postponed. The theorem is used to show $\{a^n b^n | n \geq 1\}^*$ cannot be generated by a Petri net schemata with a one-to-one event function. Languages generated by Petri net schemata where the initial and final marking are the same are studied in Section 5, and it is shown that such languages also allow postponement in the sense of the previous section.

2. Preliminaries

The definition of Petri nets and Petri net languages follows Meiling (1980). A *petri net* is a bipartite graph, i.e. a directed graph in which the set of nodes is partitioned into two disjoint subsets called the set of places (drawn as circles) and the set of transitions (drawn as bars), each arc connecting nodes of different type.

In the following we will consider a Petri net Q in which $P = \{P_1, P_2, \ldots, P_r\}$ is the set of places and $T = \{t_1, t_2, \ldots, t_s\}$ is the set of transitions. A *marking* of Q is a mapping $M: P \to \mathbb{N}$ ($\mathbb{N}$ is the set of nonnegative integers). A marking is often represented by a vector in $\mathbb{N}^r$, and graphically it is represented by $M(P_i)$ tokens (dots) in the place P_i, $i = 1, 2, \ldots, r$.

The transition function δ_Q: $T \to \mathbb{Z}^r$ ($\mathbb{Z}$ is the set of integers), or δ where Q is understood, is defined as follows:

$$(\delta_Q(t))_i = \text{(number of arcs from } t \text{ to } P_i) - \text{(number of arcs from } P_i \text{ to } t)$$

Elements of T^* (λ denotes the empty word) are called *firing sequences* The transition function is extended to a mapping

$T^* \to \mathbb{Z}^r$ as follows:

$$\delta_Q(\lambda) = 0,$$

$$\delta_Q(\sigma_1\sigma_2\ldots\sigma_n) = \sum_{i=1}^{n} \delta_Q(\sigma_i),\ \sigma_i \in T,\ i = 1, 2, \ldots, r.$$

A transition t is *firable* at the marking M if

$$M(P_i) \geq (\text{number of arcs from } P_i \text{ to } t) \text{ for } i = 1, 2, \ldots, r.$$

The marking M' resulting from firing the (firable) transition t at the marking M is defined as $M' = M * \delta_Q(t)$. A firing sequence $\sigma_1\sigma_2\ldots\sigma_n$ is valid at the marking M if σ_i if firable at $M \times \delta_Q(\sigma_1\sigma_2\ldots\sigma_{i-1})$, $i = 1, 2, \ldots, n$. We will occasionally use δ_Q as the *next state function*, a partial function from $N^r \times T^* \to N^r$ defined as follows:

$$\delta_Q(M, \sigma) = \begin{cases} M \times \delta_Q(\sigma) & \text{if } \sigma \text{ is valid at } M, \\ \text{undefined otherwise.} \end{cases}$$

Let P' be a subset of P. The *restriction* Q' of Q to P' is the Petri net obtained from Q by deleting all places from $P \backslash P'$ and deleting all arcs leading to or from a place in $P \backslash P'$. A marking M' of Q' is said to be the restriction of the marking M of Q if $M'(P_i) = M(P_i)$ for all $P_i \in P'$. Note that if $\sigma \in T^*$ is valid at M in Q, then σ is also valid at M' in Q' and $\delta_{Q'}(M', \sigma)$ is the restriction of $\delta_Q(M, \sigma)$.

Let Σ be an alphabet called *the alphabet of events*. We will often introduce a homomorphism h: $T^* \to \Sigma^*$ called the *event function*of Q. h(t) is called the *label* of the transition t, and a firing sequence σ is said to *spell* $h(\sigma)$.

A *Petri net schemata* over Σ is a quadruple (Q, M_0, h, F) where

Q is a Petri net,
M_0 is a marking (the *initial* marking),
h is an event function,
F is a set of markings (the set of *final* markings).

(Q, M_0) is called a *marked Petri net*. The language generated by the Petri net schemata is denoted

$$L(Q, M_0 h, F)$$

and it is defined as

$$\{h(\sigma) \mid \delta(M_0, \sigma) \in F\}.$$

Let M, N be markings. The partial ordering is defined as

$$M \le N \iff \forall 1 \le i \le r : M(i) \le N(i).$$

Notice that if a transition is firable at a marking M, then it is also firable at any marking M' for which $M' \ge M$. We need the following lemmas:

<u>Lemma 1</u>. (Karp and Miller (1969))

Let $M_1, M_2, \ldots, M_i, \ldots$ be an infinite sequence of markings.

There exists an infinite subsequence $M_{i_1}, M_{i_2}, \ldots, M_{i_j}, \ldots$ such that

$$M_{i_1} \le M_{i_2} \le \ldots \le M_{i_j} \le \ldots .$$

Lemma 2.

Let $M_1, M_2, \ldots, M_i, \ldots$ be an infinite sequence of elements of $\mathbb{N}^r$. There exists an infinite subsequence $M_{i_1}, M_{i_2}, \ldots, M_{i_j}, \ldots$ such that

$$\forall 1 \le k \le r : (M_{i_1}(k) = M_{i_2}(k) = \ldots = M_{i_j}(k) = \ldots \quad \text{or}$$
$$M_{i_1}(k) < M_{i_2}(k) < \ldots < M_{i_j}(k) < \ldots \;).$$

Proof.

Extract an infinite subsequence either constant or strictly increasing in the first coordinate. Extract from this an infinite subsequence either constant or strictly increasing in the second coordinate, and so forth.

3. Pumping of Petri Net Languages.

In this section we will show that languages generated by Petri net schemata in which the set of final markings has a certain property satisfies a pumping theorem very similar to the pumping theorem for the regular languages. The analysis is based on the reachability tree for marked Petri nets.

Definition.

Let $M_f \in \mathbb{N}^r$. The set $\{M \in \mathbb{N}^r \mid M \geq M_f\}$ is called the *principal arcone* defined by M_f.

It can be shown (Meiling (1980)) that if $B = L(Q, M_0, h, F)$, where F is a set having the property:

$$(N \geq M \text{ and } M \in F) \text{ implies } N \in F$$

then B can be generated by a Petri net schemata in which the set of final markings is a principal arcone.

Let G denote the family of languages generated by Petri net schemata in which the set of final markings is a principal arcone. Since $\mathbb{N}^r$ is the principal arcone defined by $(0, 0, \ldots, 0)$, languages generated by Petri net schemata in which all states are considered final, are members of G.

Definition. (Karp and Miller (1969))

Let (Q, M_0) be a marked Petri net and let P be the set of places in Q. We associate a rooted tree called the *reachability tree* with (Q, M_0). Let ω be a symbol such that for any integer a, $a < \omega$ and $a + \omega = \omega + a = \omega - a = \omega$. A *quasi-marking* is a mapping $P \to \mathbb{N} \cup \{\omega\}$, and it is represented by a vector similar to a marking. The partial ordering $\leq$ and the transition function δ are extended to quasi-markings. The reachability tree is constructed in the following way: each node is labelled by a quasi-marking and is either marked or unmarked. Each edge is labelled by a transition. Consider the following algorithm:

```
create the root as an unmarked node labelled M_0;
while there are unmarked nodes do
begin
  Let α be an unmarked node labelled M and
  let M_0, M_1, ..., M_k = M be the labels of the nodes of the
  path from the root to α.
  for each transition t firable at M do
  begin
    N := δ(M, t);
    for j := 0 to k do
      if N ≥ M_j then N(i) := ω for any i such that N(i) > M_j(i);
    create a new unmarked node β labelled N;
    create an edge from α to β labelled t;
    if N = (ω, ω, ..., ω) or N = M_j for any 0 ≤ j ≤ k then mark(β)
  end;
  mark(α)
end
```

<u>Theorem 3</u>. (Karp and Miller (1969))

The reachability tree if finite for all marked Petri nets.

As a consequence of this theorem we can prove the following lemma:

Lemma 4.

Let (Q, M_0) be a marked Petri net. There is a constant k such that if v is a firing sequence valid at M_0 of length greater than k then we may write $v = rst$, $s \neq \lambda$, $|rs| \leq k$ in such a way that $\delta_Q(s) \geq 0$.

Proof.

Let k be the height of the reachability tree for (Q, M_0). Assume that there exists a firing sequence v valid at M_0 of length greater than k. Some prefix of v spells a path from the root to a leaf α in the reachability tree. Let this prefix be v' and let $v = v't$, $t \neq \lambda$. Since α is a terminal node for which the first transition of t is firable, it must be the case that the label of α is greater than or equal to the label of some node β on the path from the root to α. Let r be the labels of the edges on the path from the root to β and let s be the labels of the edges on the path from β to α.

r

β k

s

α

Now $v = v't = rst$ and $\delta_Q(s) \geq 0$. Furthermore, $s \neq \lambda$ and $|rs| \leq k$.

Definition.

A language B is said to satify *(1, 1)-pumping* if there is a constant k such that if $w \in B$ is of length greater than k then we may write $w = xyz$, $y \neq \lambda$ in such a way that $xy^nz \in B$ for all $n \geq 1$. (The term (1, 1)-pumping refers to the fact that one subword is pumped, and that this subword has to be present at least once.) If there is a constant c such that $|xy|$ can always be chosen less than c, we say that B satisfies *initial bounded (1, 1)-pumping*.

Theorem 5.

All languages in G satisfies initial bounded (1, 1)-pumping.

Proof.

Let B be generated by (Q, M_0, h, F) where F is a principal ar-cone. We first consider the case where h is λ-free. Since the family of languages satisfying initial bounded (1, 1)-pumping is closed under λ-free homomorphisms, we may assume that h is the identity. By lemma 4, there is a constant k such that if $v \in B$ is of length greater than k then we may write $v = rst$, $s \neq \lambda$, $|rs| \leq k$ in such a way that $\delta_Q(s) \geq 0$. Since s is valid at $\delta_Q(M_0, r)$ and $\delta_Q(M_0, rs) \geq \delta_Q(M_0, r)$, s is also valid at $\delta_Q(M_0, rs)$

and

$$\delta_Q(M_0, rs^2) = \delta_Q(M_0, r) \quad 2\ \delta_Q(s) \geq \delta_Q(M_0, r).$$

By repeating this argument we see that s is valid at $\delta_Q(M_0, rs^{n-1})$ and

$$\delta_Q(M_0, rs^n) = \delta_Q(M_0, r) \quad n\ \delta_Q(M_0, s),\ n = 1, 2, \ldots,$$

t is valid at $\delta_Q(M_0, rs^n)$ for all n = 1, 2, ... and

$$\delta_Q(M_0, rs^n t) \geq \delta_Q(M_0, rst) \in F.$$

Since F is a principal arcone this implies that $\delta_Q(M_0, rs^n t) \in F$ and thus $rs^n t \in B$ for all $n \geq 1$.

We now turn to the case where h is not necessarily λ-free. The proof above cannot be used when the event function is not λ-free, since h(s) may be λ. Instead we construct a "λ-compact" reachability tree in which paths spelling λ do not occur.

Definition.

Let (Q, M_0) be a marked Petri net and let h be an event function. Let M be a marking. The tree $\Lambda_Q(M)$ is defined as the reachability tree for (Q, M) using only transitions spelling λ. By König's theorem (König (1936)) and lemma 1, $\Lambda_Q(M)$ is finite. The λ-compact reachability tree is now constructed as follows: Each node is labelled by a quasi-marking and is either marked or unmarked. Each edge is labelled by a word in Σ^+. Consider the following algorithm:

```
create the root as an unmarked node labelled M_0;
while there are unmarked nodes do
begin
  Let α be an unmarked node labelled M and
  Let M_0, M_1, ..., M_k = M be the labels of the nodes of the
  path from the roof to α.
  for each path p in Λ_Q(M) starting at the root do
    for each transition t firable at δ_Q(M, p) do
    if h(t) ≠ λ then
    begin
      N := δ(M, pt);
      for j := 0 to k do
        if N > M_j then N(i) := ω for any i such that N(i) > M_j(i);
      create a new unmarked node β labelled N;
      created an edge from α to β labelled h(pt) = h(t);
      if N = (ω, ω, ..., ω) or N = M_j for any 0 ≤ j ≤ k then mark(β)
    end;
  mark(α)
end
```

Lemma 6.

The λ-compact reachability tree is finite for all marked Petri nets and all event functions.

Proof.

The proof in Karp and Miller (1969) for the finiteness of reachability trees can be adapted to λ-compact reachability trees. Note, however, that the outdegree of nodes in the λ-compact reachability tree is not bounded (but still finite).

Lemma 7.

Let (Q, M_0) be a marked Petri net and let h be an event function. There is a constant k such that if v is a firing sequence valid at M_0 for which $|h(v)| > k$ then we may write v = rst in a way such that

$$h(s) \neq \lambda,$$
$$|h(r)h(s)| \leq k,$$
$$\delta_Q(s) \geq 0.$$

Proof.

The lemma can be proved in a way similar to the proof of lemma 4, using the λ-compact reachability tree instead of the reachability tree.

The proof of theorem 5 can now be complemented using the technique for the case where h is λ-free together with lemma 7.

We now give some examples of languages not satisfying initial bounded (1, 1)-pumping.

Consider the language

$$L_0 = \{a^n cb^n \mid n \geq 0\}$$

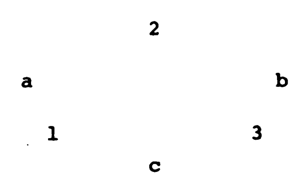
L_0 does not satisfy (1, 1)-pumping and hence $L_0 \notin G$. So languages in G cannot perform counting. The language

$$L_1 = \{a^n cb^m \mid m \geq n \geq 0\}$$

is not in G. Let $w = xyz$ be a word in L_1 such that $xy^n z \in L$ for all $n \geq 1$. y must consist of b's only, so xy is of the form $a^n cb^p$. The length of words of this form cannot be bounded, showing that L_1 does not satisfy initial bounded (1, 1)-pumping. Hence $L_1 \notin G$. Consider the language

$$L_2 = \{a^n cb^m \mid n \geq m \geq 0\}.$$

If w is a word from L_2 starting with an a, this a can always be pumped. If w starts with c, the first b can be pumped. In both cases the pumping is initial bounded. L_2 is generated by the Petri net



with $\{M \mid M \geq (0, 0, 1)\}$ as the set of final markings. The regular languages satisfy a pumping condition where the length of both x and z is bounded. The length of z cannot be bounded for all languages in G as it can be seen from the language L_2.

Let D_1 be the Dyck language over $\{(,)\}$. Since

$$w \in D_1 \text{ implies } w^n \in D_1 \text{ for all } n \geq 1$$

the language D_1 satisfies (1, 1)-pumping. However, by considering words of the form $(^n)^n$ it is seen that D_1 does not satisfy initial bounded (1, 1)-pumping, so D_1 is not in G.

The pumping theorem in this section is also valid for certain generalizations of Petri nets for which can be constructed trees similar to the λ-compact reachability tree. This is the case for resetting Petri nets (Araki and Kasami (1977)) in which a transition can reset a place to zero and also for the more general post-self-modifying nets in Volk (1978).

4. The Postponement Theorem

In this section we investigate another consequence of the "partial blindness" of Petri nets. In a Petri net a certain marking can be ensured at the end by choosing it as the final marking, but it is in general not possible to check for specific markings during the generation of a word. Expressed in terms of language it is not possible to check if subwords are generated by the net with some specific marking as the final marking. This limitation of the power of a Petri net is expressed in the "postponement" theorem. The theorem can be used much in the same manner as a pumping theorem.

Let in the following (Q, M_0) be a marked Petri net and $P = \{p_1, p_2, \ldots, p_r\}$ resp. $T = \{t_1, t_2, \ldots, t_s\}$ be the places resp. the transitions of Q.

Definition.

Let $U_1, U_2, \ldots$ be an infinite sequence of subsets of T^*. The sequence is said to be *double infintite* for (Q, M_0) if

(i) U_i is infinite for all $i = 1, 2, \ldots,$

(ii) the set of all firing seqences valid at M_0 includes $U_1U_2\ldots U_k$ for all $k = 1, 2, \ldots.$

Definition.

A pair of firing sequences (x', y') is said to be a *postponement* of the pair of firing sequences (x, y) if

$$x = x_1x_2x_3, \qquad x' = x_1x_3,$$
$$y = y_1y_2, \qquad y' = y_1x_2y_2,$$

for some $x_1, x_2, x_3, y_1, y_2 \in T^+$.

Theorem 8. (the postponement theorem for firing sequences)
Let $U_1, U_2, \ldots$ be double infinite for (Q, M_0). For infinitely many values of $u_i \in U_i$, $i = 1, 2, \ldots$ there exists integers j, k such that

$$u_1 u_2 \ldots u_j \ldots u_k \quad \text{and}$$
$$u_1 u_2 \ldots u_j' \ldots u_k'$$

are firing sequences valid at M_0 and (u_j', u_k') is a postponement of (u_j, u_k).

Proof.
The idea of the proof is as follows: Notice that if a place p contains d tokens, the validity of a firing sequence of length d/(outdegree of d) is independant of the place p. This leads us to consider places which simultaneously can contain an arbitrary large number of tokens and to consider the restriction of the Petri net to the remaining places. For the Petri net obtained in this way we will prove the existence of a marking M and firing sequences u and v such that $\delta(u) \le 0$ and $\delta(M, uv) \ge M$. This condition ensures that it is possible to postpone the generation of u until after the generation of v. Going back to the original Petri net, the content of the simultaneous unbounded places have to be chosen large enough to permit the postponement of u.

The theorem is proved by induction on r, the number of places in Q. Assume the theorem to be true for all Petri nets with fewer than r places. Let $U_1 = \{u_{11}, u_{12}, \ldots, u_{1i}, \ldots\}$. The infinite sequence of markings

$$\delta_Q(M_0, u_{11}), \delta_Q(M_0, u_{12}), \ldots, \delta_Q(M_0, u_{1i}), \ldots$$

contains by lemma 2 an infinite subsequence $K_1, K_2, \ldots, K_i, \ldots$ in which every coordinate is either strictly increasing or constant. Assume without loss of generality that the first p coordinates $(0 \le p \le r)$ are strictly increasing, while the remaining (r-p) coordinates are constant. The proof is now divided into the follwoing two cases:

case 1: $p > 0$.

Let Q' be the restriction of Q to the places represented by the (r-p) last coordinates and let π be the mapping of a marking in Q to its restriction to Q'. Let M_1 be the common value of $\pi(K_i)$. $U_2, U_3, \ldots$ is a double infinite sequence for (Q', M_1) and Q' has fewer than r places. Hence, by induction, for infinitely many values of $u_i \in U_i$, i = 2, 3, ... there exists integers j, k such that

$$u_2 u_3 \ldots u_j \ldots u_k \quad \text{and}$$
$$u_2 u_3 \ldots u_j' \ldots u_k'$$

are firing sequences valid at M_1 and (u_j', u_k') is a postponement of (u_j, u_k). Consider the sequence $u_2u_3...u_j'...u_k'$ as a firing sequence in Q at some marking M for which $\pi(M = M_1$. The sequence might not be valid at M_1 since for some initial segment u' of the sequence some of the first p coordinates of $M + \delta_Q(u')$ could be negative. However, this is not the case if the first p coordinates of M is larger than the length of u times the maximal indegree of any transition. Since the sequence $K_1, K_2, ..., K_i, ...$ is strictly increasing in the first p coordinates this condition is fulfilled for all but finitely many K_i. Let $u_1 \varepsilon U_1$ be any element such that $\delta_Q(M_0, u_1)$ fulfills the condition. Then

$$u_1u_2...u_j...u_k \quad \text{and}$$
$$u_1u_2...u_j'...u_k'$$

are valid at M_0 completing the proof for case 1.

<u>case 2: p = 0</u>.

In case 2 we need the following Lemma:

<u>Lemma 9</u>.

Let Q be a Petri net and let M be a marking. There is a constant k such that if v is a firing sequence of length greater than k and $\delta_Q(M', v) = M$ for some marking M', then we may write $v = rst$, $s, \neq \lambda$, $|st| \leq k$ in such a way that $\delta_Q(s) \leq 0$.

Proof.

Consider the marked Petri net (reverse(Q), M), where reverse(Q) is the Petri net obtained from Q by reversing the direction of all edges. The set of all firing sequences in reverse(Q) valid at M equals the reverse of the set of all valid firing sequences in Q ending in M. Furthermore, $\delta_{reverse(Q)} = -\delta_Q$. The lemma is now proved by applying lemma 4 to (reverse(Q), M).

When $p = 0$ all the markings in the sequence $K_1, K_2, \ldots, K_i, \ldots$ are identical. Let $\hat{K}_1$ be their common value. Let u_1 be any element of U_1 for which

$$\delta_Q(M_0, u_1) = \hat{K}_1.$$

By lemma 9, if u_1 is sufficiently long, we may write

$$u_1 = r_1 s_1 t_1, \; s_1 \neq \lambda, \; \delta_Q(s_1) \leq 0.$$

The sequence $U_2, U_3, \ldots$ is double infinite for $(Q, \hat{K}_1)$. We now repeat the argument of the first part of the proof using this double infinite sequence. If we find ourself in case 1 we are through, otherwise we find a sufficiently long $u_2 \in U_2$

and a marking $\hat{K}_2$ such that

$$\hat{K}_2 = \delta_Q(\hat{K}_1, u_2),$$
$$U_3, U_4, \ldots \text{ is double infinite for } (Q, \hat{K}_2),$$
$$u_2 = r_2 s_2 t_2, \ s_2 \neq \lambda, \ \delta_Q(s_2) \leq 0.$$

By repeating this argument we now only have to deal with the situation where case 1 never occurs. In this case we have an infinite sequence $u_1, u_2, \ldots, u_i, \ldots$ such that

$$u_i \in U_i, \ i = 1, 2, \ldots,$$
$$u_1 u_2 \ldots u_k \text{ is valid at } M_0, \ k = 1, 2, \ldots,$$
$$u_i = r_i s_i t_i, \ s_i \neq \lambda, \ \delta_Q(s_2) \leq 0.$$

Consider the infinite sequence of states:

$$N_i = \delta_Q(M_0, u_1 u_2 \ldots u_{i-1} r_i), \ i = 1, 2, \ldots.$$

By lemma 1 there exist integers $j < k$ such that $N_j \leq N_k$. The firing sequence

$$u_1 u_2 \ldots u_j \ldots u_k$$

is valid at M_0. We will prove that the firing sequence

$$u_1 u_2 \ldots u_j' \ldots u_k'$$

where

$$u_j' = r_j t_j,$$
$$u_k' = r_k s_j s_k t_k$$

is also valid at M_0. Since (u_j', u_k') is a postponement of (u_j, u_k) this will complete the proof. Let

$$x = u_1 u_2 \ldots u_{j-1} r_j,$$
$$y = t_j u_{j+1} \ldots u_{k-1} r_k,$$
$$z = s_k t_k.$$

By definition, $u_1 u_2 \ldots u_j \ldots u_k$ equals xyz. Any prefix of a firing sequence valid at M_0 is valid at M_0, so x is valid at M_0. Since

$$\delta_Q(M_0, x) \geq \delta_Q(M_0, x) \quad \delta_Q(M_0, xs_j)$$

every firing sequence valid at $\delta_Q(M_0, xsj)$ is also valid at $\delta_Q(M_0, x)$. In particular, y is valid at $\delta_Q(M_0, x)$. Furthermore, s_j is valid at $\delta_Q(M_0, xy)$ since s_j is valid at $\delta_Q(M_0, x)$ and

$$\delta_Q(M_0, xy) \geq \delta_Q(M_0, xs_j y) = N_i \geq N_j = \delta_Q(M_0, x).$$

Finally, z is valid at $\delta_Q(M_0, xys_j)$ since z is valid at $\delta_Q(M_0, xs_j y)$ and $\delta_Q(M_0, xys_j) = \delta_Q(M_0, xs_j y)$. This completes the proof.

We will now consider the consequences of this theorem for the family of languages generated by Petri net schamata in which the event function is one-to-one. Notice that we do not make any assumptions about the set of final markings (indeed, the set might not even be decidable).

Definition.

Let $B \subseteq \Sigma^*$ and let $U_1, U_2, \ldots, U_i, \ldots$ be an infinite sequence of subsets of Σ^*. The sequence is said to be *double infinite* for B if

$$U_i \text{ if infinite for all } i = 1, 2, \ldots,$$
$$\forall k \forall (u_1, u_2, \ldots, u_k) \in U_1 \times U_2 \times\times\ldots\times\times U_k \; \exists v: u_1 u_2 \ldots u_k v \in B$$

Notice the connection with the definition of double infinite sequences for marked Petri nets: If (Q, M_0) is a marked Petri net where Q has r places and $U_1, U_2, \ldots, U_i, \ldots$ is double infinite for (Q, M_0) then $U_1, U_2, \ldots, U_i, \ldots$ is double infinite for $L(Q, M_0, t \to t, \mathbb{N}^r)$.

Definition.

A language B is said to *allow postponement* for the double infinite sequence $U_1, U_2, \ldots, U_i, \ldots$ for B if for infinitely many values of $u_i \in U_i$, $i = 1, 2, \ldots$ there exists integers j and k and a word v such that

$$u_1 u_2 \ldots u_j \ldots u_k v \in B \quad \text{and}$$
$$u_1 u_2 \ldots u_j \ldots u_k' v \in B$$

for some postponement (u_j', u_k') of (u_k, u_k). (Postponement for words is defined analogous to postponement for firing sequences.)

B is said to allow postponement if B allows postponement for all double infinite sequences for B.

Theorem 10.

All languages in H allow postponement.

Proof.

Let B be a language generated by a Petri net schemata (Q, M_0, h, F) where h is one-to-one and let $V_1, V_2, \ldots, V_i, \ldots$ be double infinite for B. Define the sequence $U_1, U_2, \ldots, U_i, \ldots$ where $U_i = h^{-1}(V_i)$, $i = 1, 2, \ldots$. THis sequence is double infinite for (Q, M_0). By theorem 8, for infinitely many values of $u_i \in U_i$, $i = 1, 2, \ldots$ there exists integers j and k such that

$$u_1u_2\ldots u_j\ldots u_k \quad \text{and}$$
$$u_1u_2\ldots u_j'\ldots u_k'$$

are valid at M_0 and (u_j', u_k') is a postponement of (u_j, u_k). Let $v_i = h(u_i)$, $i = 1, 2, \ldots, k$ and let $v_j' = h(u_j')$, $v_k = h(u_k')$. Since $V_1, V_2, \ldots, V_i, \ldots$ is double infinite for B we get

$$\exists\ v \in \Sigma^* : v_1v_2\ldots v_j\ldots v_kv \in B \quad \Longleftrightarrow$$
$$\exists\ u \in T^* : \delta_Q(M_0, u_1u_2\ldots u_j\ldots u_ku) \in F \quad \Longleftrightarrow$$
$$\exists\ u \in T^* : \delta_Q(M_0, u_1u_2\ldots u_j'\ldots u_k'u) \in F \quad \Longleftrightarrow$$
$$\exists\ v \in \Sigma^* : v_1v_2\ldots v_j'\ldots v_k'v \in B \quad \Longleftrightarrow$$

Since (v_j', v_k') is a postponement of (v_j, v_k) this completes the proof.

Remark.

Theorem 10 can be strengthened in two ways: First we may assume k to be arbitrarily larger than j and thus the distance a subword can be postponed is arbitrarily long. This can be seen from the construction in the second part of the proof of theorem 8. The integers j and k are indices in an infinite sequence of markings chosen such that the j'th marking is less than or equal to the k'th marking. Since every infinite sequence of markings contains an infinite increasing subsequence, k can be arbitrarily larger than j. Another important property can be deduced from this construction: Let $u_j = r_j s_j t_j$. The size of $s_j t_j$ can--given $u_1, u_2, \ldots, u_{j-1}, u_{j+1}, \ldots, u_k$--by Lemma 9 be chosen to be less than some integer c. We will call this type of postponement *final bounded postponement*.

We will now give some examples of languages not allowing postponement. By Theorem 10 such a language cannot be in H.

Consider the language $L_3 = \{a^n cb^n | n \geq 1\}^*$. Let $U = \{a^n cb^n | n \geq 1\}$. The sequence U, U, ..., U, ... is double infinite for L_3. It is seen that L_3 does not allow postponement for this sequence (the generation of a subword containing c cannot be postponed, and postponing the generation of a subword containing only a's or b's will destroy the balance), so $L_3 \notin \mathcal{H}$.

Consider the Dyck language D_1 over $\{(,)\}$. D_1 allows postponement since it is always possible to postpone the generation of closing parenthesis. D_1 is generated by

with initial marking (0) and $\{(0)\}$ as the set of final markings. Now consider the Dyck language D_2 over $\{(,),[,]\}$. Let $U = \{[(^n)^n] \mid n \geq 0\}$. The sequence U, U, ..., U, ... is double infinite for D_2. Notice the similarity between this sequence and the infinite sequence for $\{a^n b^n \mid n \geq 1\}^*$. Because of the absence of c's it is possible to postpone any subword of the form $(^p)^p$. However, this postponement is not final bounded, and since no other postponement is possible, we conclude that $D_2 \notin \mathcal{H}$.

5. Cyclic Languages

In this section we will consider other Petri net languages allowing postponement. We will look at Petri nets in which some marking can occur arbitrarily often. Such nets are of importance when modelling systems which in principle never stop, e.g. operating systems. In such systems there will often be a neutral state to which you always return. This situation can be modelled by Petri net schemata in which the initial marking is the only final marking. Such languages were first mentioned in Hack (1975).

Definition.

A Petri net schemata of the form

$$(Q,\ M_0,\ h,\ \{M_0\})$$

is called cyclic. The language

$$\{h(w)\ |\ \delta_Q(M_0,\ w) = M_0 \quad \text{and} \quad \delta_Q(M_0,\ w') \neq M_0 \text{ for any nontrivial subword } w' \text{ of } w\}$$

is called the *atomic cycle* of $(Q,\ M_0,\ h,\ \{M_0\})$.

Theorem 11.

Let B be the language generated by a cyclic Petri net schema $(Q,\ M_0,\ h,\ \{M_0\})$ and let C be the atomic cycle of $(Q,\ M_0,\ h,\ \{M_0\})$. If C is infinite then C, C, ..., C, ... is a double infinite sequences for B allowing final bounded postponement.

Proof.

Since $B = C^*$ the sequence C, C, ..., C, ... is double infinite for B if C is infinite.

Consider now reverse(Q), the Petri net obtained from Q by reversing the direction of all edges. By applying lemma 7 to reverse(Q), M_0 and h in the same way as in the proof of lemma 9 we see that there is a constant k such that if $c \in C$ is of length greater than k and u is a firing sequence spelling c such that $\delta_Q(M_0, u) = M_0$ then we may write u = rst in a way such that

$$h(s) \neq \lambda,$$
$$|h(s)h(t)| \leq k,$$
$$\delta_Q(s) \leq 0.$$

Let $c_1, c_2, \ldots, c_i, \ldots$ be a sequence of elements of C, $|c_i| > k$ for all i = 1, 2, ..., and let $u_i = r_i s_i t_i$, i = 1, 2, ... be firing sequences satisfying the requirements above. Consider the sequence

$$\delta_Q(r_1), \delta_Q(r_2), \ldots, \delta_Q(r_i), \ldots.$$

By lemma 1 there exists integers j < k such that $\delta_Q(r_j) \leq \delta_Q(r_k)$. The firing sequence $u_j u_k$ is valid at M_0 and $h(u_j)h(u_k) \in B$. As in the proof of theorem 8 it is seen that the firing sequence

$$(r_j t_j)(r_k s_j s_k t_k)$$

is valid at M_0. Furthermore, $(h(r_j)h(t_j), h(r_k)h(s_j)h(s_k t_k))$ is a postponement of $(h(u_j), h(u_k))$ and $|h(s_j)h(t_j)| \leq k$. This completes the proof.

Conclusion.

The properties of Petri nets exhibited in this paper demonstrates that even a very simple arithmetic operation like counting is not easily modelled by Petri nets. However, pumping and postponement do not seem to effect the ability to model flow of control. An open problem is whether languages generated by arbitrary Petri net schemas allow postponement. If this is the case the language $\{a^n b^n | n \geq 1\}^*$ could not be generated by a Petri net. If this language can be generated by a Petri net schema in which the set of final markings is a singleton, the reachability problem for vector addition systems is undecidable (Greibach (1978)).

References.

T. Araki and T. Kasami: "Some Decision Problems Related to the Reachability Problem for Petri Nets," Theoretical Computer Science 3 (1977), 85-104.

P. Fisher, A. Meyer and A. Rosenberg: "Counter Machines and Counter Languages," Mathematical System Theory 2 (1968), 265-283.

S. A. Greibach: "Remarks on Blind and Partially Blind One-Way Multicounter Machines," Theoretical Computer Science 7 (1978), 311-324.

M. Hack: "Petri Net Languages," Computation Structures Group Memo 124, Project MAC, Cambridge, Mass. (1975).

E. Meiling: "On the Definition of Languages from Petri Nets," Department of Computer Science, University of Copenhagen, 802 (1980).

J.L. Peterson: "Computation Sequence Sets," Journal of Computer and Systems Sciences 13 (1976), 1-24.

R.M. Karp and R.E. Miller: "Parallel Program Schemata," Journal of Computer and Systems Science 3 (1969), 167-195.

K. König: "Theorie des Endlicher und Unendlicher Graphen," Academische Verlagsgesellschaft, Leipzig (1936).

R. Valk: "Self-Modifying Nets, a Natural Extension of Petri Nets," in "Automata, Languages and Programming Proceedings 1978," Lecture Notes in Computer Science Vol. 62, Springer-Verlag (1978).