JOURNAL OF COMPUTER AND SYSTEM SCIENCES 23, 299–325 (1981)

Petri Nets and Regular Languages

RÜDIGER VALK

Fachbereich Informatik, Universität Hamburg, Schlüterstrasse 70, D-2000 Hamburg 13, West Germany

AND

GUY VIDAL-NAQUET

Institut de Programmation de l'Université Paris VI, 4 Place Jussieu, F-75230 Paris Cedex 05, France

Received September 23, 1979; revised January 6, 1981

It is shown that the regularity problem for firing sequence sets of Petri nets is decidable. For the proof, new techniques to characterize unbounded places are introduced. In the class $\mathscr{L}_0$ of terminal languages of labelled Petri nets the regularity problem in undecidable. In addition some lower bounds for the undecidability of the equality problems in $\mathscr{L}_0$ and $\mathscr{L}$ are given. $\mathscr{L}_0^\lambda$ is shown to be not closed under complementation without reference to the reachability problem.

1. INTRODUCTION

Petri nets provide an important formalism for modelling and analyzing concurrent systems. In a Petri net the occurrence of an event or action of the system is represented by the firing of a transition. Such a firing changes the actual marking of the net, which describes the global state of the system. The set of all possible sequences of such firings can be interpreted as a language, in the sense of formal language theory. This approach to obtain formal properties has been used by several researchers [2, 7, 12]. A concurrent and independent firing of several transitions is represented by considering all possible sequential occurrences of single firings. A more explicit notion of concurrent firing, as introduced by Petri and as formally defined in [3, 9], is not necessary for the purpose of this paper.

We first define Petri nets and several types of languages which are associated with them. The set of all firing sequences of a Petri net starting from a specific initial marking is called the language of firing sequences of the net. If only sequences leading to a distinguished finite set of terminal markings are of interest, this set is called the terminal language of the net. When several transitions play the role of the same action under different conditions, they are labelled by a common name. Such

0022-0000/81/060299–27$02.00/0

nets will be called labelled Petri nets. Formal definitions of these languages are given in Section 2.

In Section 3 we present our main tool for investigating the language of a net. By a modified version of the coverability tree of Karp and Miller [10] we are able to analyze properties of firing sequences leading to markings of bounded and unbounded size.

Section 4 introduces new concepts that allow the description of certain aspects of unbounded behaviour of nets. The lemmas of this section relate these notions to properties of the coverability graph.

The set of reachable markings of a net may be unbounded, whereas at the same time the set of firing sequences of the net is a regular language. Such nets having a regular language are called regular nets. Regular nets properly contain the well-known class of bounded nets. In Section 5 we show that it is decidable whether a net is regular or not. This result resolves a question raised in [12]. At the end of Section 5 we also show that there is a procedure which effectively gives a finite automaton accepting the language of a regular Petri net. In addition it is proved that the space and time complexity of each such procedure cannot be bounded by a primitive recursive function.

For terminal languages of labelled Petri nets regularity is undecidable. Section 6 gives a direct proof. Although this result can be derived from earlier work in language theory [1, 5], the direct proof gives more insight into the structure of Petri nets and allows proof of additional results. Two of these theorems give lower bounds on the number of places for which the equality problem for languages is undecidable. By a similar method the class $\mathscr{L}_0^\lambda$ is not closed under complementation. It is also undecidable, whether the palindrome language is a subset of the terminal language of a net or not. This language is typical of languages accepted by pushdown automata.

Petri nets can be seen as automata with counters and partial tests for zero. A firing sequence of a net then corresponds to an accepted word on the input tape of the automaton. Under this interpretation unlabelled nets are deterministic automata, since for a given state (i.e., a marking) and a given transition at most one state transition is possible. In a labelled net, however, it can happen that two transitions with the same label can fire in the marking. This corresponds to the definition of a nondeterministic automaton.

Our results are therefore similar to the corresponding questions in the case of pushdown automata: Regularity is decidable for deterministic pushdown automata, but undecidable in the nondeterministic case.

We presented most of our results for the first time at the Third International G.I. Conference on Theoretical Computer Science in 1977 [15]. After submitting this manuscript and during the second redaction in 1980, we saw similar work using a different method for our first result [4].

2. Petri Nets and Languages

In this section we give the definition of Petri nets and the different kinds of languages associated with them. We follow the approach of [7].

2.1. *Definition of the Petri Net*

A *Petri net*, N, is a 5-tuple defined by

$$N = (P, T, B, F, M_0),$$

where

$P = \{p_1, \dots, p_r\}$	is a finite set of *places*,
$T = \{t_1, \dots, t_s\}$	is a finite set of *transitions*,
$B : P \times T \to \mathbb{N}$	is is the *backwards incidence function*,
$F : P \times T \to \mathbb{N}$	is the *forwards incidence function*,
$M_0 \in \mathbb{N}^r$	is the *initial marking*.

As usual, $\mathbb{N}$ denotes the set of nonnegative integers.

An example of a Petri net is given in Fig. 1. Usually a Petri net is represented by a bipartite graph. Both places and transitions are the nodes of the graph. Graphically we use circles for places and bars for transitions. If p_i is a place and t_j a transition, then the graph has n arcs directed from p_i to t_j when $B(p_i, t_j) = n$. If n is positive, p_i is called an *input place* of t_j and t_j an *output transition* of p_i. Similarly there are m arcs from t_j to p_i when $F(p_i, t_j) = m$. Again, for $m > 0$ the place p_i is an *output place* of t_j and t_j an *input transition* of p_i. The graphical representation of the Petri net of Fig. 1 is given in Fig. 2.

2.2. *Firing Rules for a Petri Net*

The global state of a Petri net is given by a number of tokens in each place of the net. By the firing of a transition tokens belonging to places which are connected by arcs with this transition can be moved to other places. Thus the dynamic behaviour

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6\} \qquad T = \{t_1, t_2, t_3, t_4, t_5\}$$

$$B = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Fig. 1. Example of a Petri net.

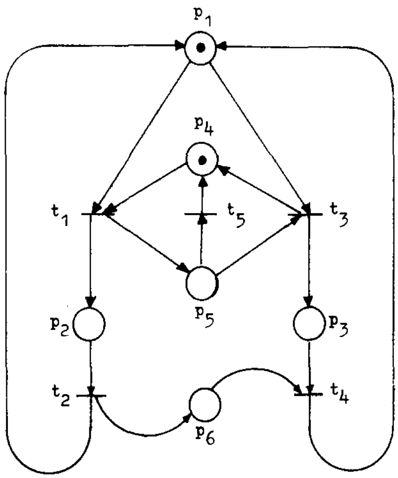

FIG. 2. Graphical representation of the Petri net of Fig. 1.

of the net can be defined by sequences of firings of such transitions. These are therefore called firing sequences.

In general the following formal conventions are supposed. The relations $=$ and $\leqslant$, and the operations $+$ and $-$ are extended from $\mathbb{N}$ to vectors in $\mathbb{N}^k$ by componentwise application. The mapping F is also interpreted as an (r, s)-matrix with r lines, s columns and components $F_{ij} := F(p_i, t_j)$ $(1 \leqslant i \leqslant r,\ 1 \leqslant j \leqslant s)$. The same conventions hold for the matrices B and $D := F - B$ (in the latter case also componentwise application of subtraction is supposed). The jth columns of these matrices are associated with the transition t_j and are denoted by $F(\cdot, t_j)$, $B(\cdot, t_j)$ and $D(\cdot, t_j)$, respectively.

DEFINITION. A *marking* is a mapping $M: P \to \mathbb{N}$, also represented by a vector $M \in \mathbb{N}^r$. $M(p)$ is the number of tokens in p for the marking M. A transition t *can fire* in a marking M if $M \geqslant B(\cdot, t)$. This relation between transitions and markings is denoted by $M(t\rangle$. If t can fire in M, t is said *to fire marking M to marking M'* when $M' = M + D(\cdot, t)$. This *firing relation* is denoted by $M(t\rangle M'$.

Intuitively, t can fire in M if the number of tokens $M(p)$ in each input place p of t is not less than $B(p, t)$. By firing M to M' the number of tokens of each input place p of t is decreased by $B(p, t)$ and each output place p' of t obtains $F(p', t)$ tokens. If p is both an input place and a output place, the number of tokens in p is decreased and increased at the same time by defining $M'(p) := M(p) - B(p, t) + F(p, t) = M(p) + D(p, t)$.

DEFINITION. The firing relation is extended to words of T^* by $M(\lambda\rangle M$ for the empty word λ and each $M \in \mathbb{N}^r$. For all $w \in T^*$, $t \in T$ and M, $M' \in \mathbb{N}^r$ we define $M(wt\rangle M'$ iff there is a marking M'' with $M(w\rangle M''$ and $M''(t\rangle M'$. M' is said to be *reachable from* M' if $\exists w : M(w\rangle M'$. This is also denoted by $M(*\rangle M'$. $R(M) := \{M' \mid M(*\rangle M'\}$ is the set of markings reachable from M. The *reachability set* of N is the set of all markings from the initial marking $R(N) := R(M_0)$. For cases where the reached marking is less important, we define $M(w\rangle$ iff $\exists M' : M(w\rangle M'$ and say that $w \in T^*$ *can fire* in M. A firing sequence $w = t_{i_1} t_{i_2} \cdots t_{i_n}$ can fire in M, if $M(p) \geqslant \sum_{j=1}^{n} B(p, t_j)$ for all $p \in P$. Therefore this sum is denoted by $B(p, w)$. As $D(p, t)$ denotes the change of tokens in p by a firing of t, we extend this notion to hold for the firing sequence w by $D(p, w) := \sum_{j=1}^{n} D(p, t_{i_j})$. For the sake of completeness these sums are defined to have the value 0 in the case of the empty word $w = \lambda$. If a word w is a prefix of w', we write $w \sqsubseteq w'$, i.e., $w \sqsubseteq w'$, iff $\exists v \in T^*$: $w' = wv$. For a set of words $L \subseteq X^*$ the set $\mathrm{Pref}(L) := \{v \in X^* \mid \exists w \in L : v \sqsubseteq w\}$ is the *prefix language* of L.

It is sometimes convenient to represent a marking $M \in \mathbb{N}^r$ with $M(p_i) = n_i$ also by a word $p_1^{n_1} p_2^{n_2} \cdots p_r^{n_r}$ over the alphabet P. Exponents $n_i = 1$ are omitted as well as letters p_i with exponent $n_i = 0$. For example by this notation the initial marking given with the net in Fig. 1 is $M_0 = p_1 p_4$. Examples for the firing relation of this net are $p_1 p_4 (t_1\rangle p_2 p_5$ and $p_2 p_5 (t_2 t_5 t_1 t_2\rangle p_1 p_5 p_6^2$. To some extent the dynamic behaviour of the net is represented by the set of all firing sequences that are firable from the initial marking.

2.3. *Languages of Petri Nets*

As usual in theoretical computer science sets of words are called languages. In this sense, sets of firing sequences are considered as languages. By allowing only certain markings as termination condition, we obtain terminal languages of nets.

DEFINITION. For a Petri net $N = (P, T, B, F, M_0)$ the *language of firing sequences* is defined by $F(N) = \{w \in T^* \mid M_0(w\rangle\}$. For a given finite set of markings $\mathfrak{M} \subseteq \mathbb{N}^r$ the pair $(N, \mathfrak{M})$ is a *Petri net with terminal marking set* and $F(N, \mathfrak{M}) := \{w \in T^* \mid \exists M \in \mathfrak{M} : M_0(w\rangle M\}$ is the *terminal language of firing sequences* of $(N, \mathfrak{M})$.

These definitions are now extended to Petri nets where transitions are labelled by letters or the empty word. This allows reference to different transitions by the same label or to ignore the firing of a transition in a firing sequence. Instead of sequences of transitions we then consider the sequences of the corresponding labels.

DEFINITION. A *labelled Petri net* N is a tuplet $N = (N_1, X, h)$, where $N_1 = (P, T, B, F, M_0)$ is a Petri net, X is a finite set of *labels* and $h : T \to X \cup \{\lambda\}$ is a *labelling function*. This function associates to each transition t a label $h(t)$. h is extended to a homomorphism from T^* into X^* in the canonical way: $h(\lambda) := \lambda$ and $h(wt) := h(w)\, h(t)$ for $w \in T^*$ and $t \in T$. If $h(t) \neq \lambda$ for all $t \in T$ the labelling function h is called *λ-free*.

DEFINITION. $L(N) := \{h(w) \mid w \in F(N_1)\}$ is the *language of the labelled net* $N = (N_1, X, h)$. $L(N, \mathfrak{M}) := \{h(w) \mid w \in F(N_1, \mathfrak{M})\}$ is the *terminal language of the* *labelled Petri net* $(N, \mathfrak{M})$. The class of all languages and the class of all terminal languages of labelled Petri nets are denoted by $\mathscr{L}^\lambda$ and $\mathscr{L}_0^\lambda$, respectively. When only λ-free labelling functions are allowed, we obtain the classes $\mathscr{L}$ and $\mathscr{L}_0$.

It has been shown in [8] that $\mathscr{L}^\lambda$ strictly includes $\mathscr{L}$ and that $\mathscr{L}_0^\lambda$ strictly includes $\mathscr{L}_0$. For an example, consider the Petri net N of Fig. 2 and $\mathfrak{M} := \{p_2 p_5\}$. Then $F(N)$ is the regular language $\text{Pref}((t_1(t_5 t_2 \cup t_2(t_5 \cup t_3 t_4)))^*)$ and $F(N, \mathfrak{M}) = (t_1 t_2 t_3 t_4)^* t_1$. Intuitively the token in p_1 can be moved to the left-hand side and to the right-hand side of the net. If x_1 is the number of moves to the left-hand side and x_2 the number of moves to the right-hand side, then p_6 plays the role of a counter for $x_1 - x_2$. Introducing a labelling function $h: T \to \{a, b, \lambda\}$ by $h(t_1) := a$, $h(t_3) = b$ and $h(t_i) = \lambda$ for $i \notin \{1, 3\}$ the language of the labelled net $N' = (N, \{a, b\}, h)$ is $L(N') = \text{Pref}((a \cup ab)^*) = (a \cup ab)^*$.

3. Unboundedness and Regular Petri Nets

To prepare Section 4 we recall the definition of a bounded net and a modified version of the decision procedure of [10] as given in [6]. Boundedness is an important property of Petri nets, since only bounded storage devices are needed. Furthermore the set of all firing sequences of a bounded net can be represented by a finite graph, but the class of Petri nets having this property properly contains the class of bounded nets. Therefore this class is defined at the end of this section as the class of regular Petri nets.

Definition. Given a Petri net $N = (P, T, B, F, M_0)$, a place $p \in P$ is *bounded* if there is an integer n, such that $M(p) \leqslant n$ for all reachable markings $M \in R(N)$. A Petri net is *bounded* when every place of the net is bounded.

Boundedness is a decidable property. The decision procedure consists in the construction of a finite graph, called the coverability graph. If a place is found to be unbounded, a special symbol ω is introduced to indicate this fact.

Definition. The set $\mathbb{N}$ is extended by a special symbol ω to the set $\mathbb{N}_\omega := \mathbb{N} \cup \{\omega\}$. $\mathbb{N}^r$ denotes the set of r-vectors over $\mathbb{N}$. The operations $+$, $-$ and the relation $\leqslant$ over $\mathbb{N}$ are extended to $\mathbb{N}_\omega$ by $\omega + \omega = \omega + n = n + \omega = \omega$ and $n \leqslant \omega$ for all $n \in \mathbb{N}$. We extend these operations and relation to $\mathbb{N}_\omega^r$ by applying them componentwise. For $Q \in \mathbb{N}_\omega^r$ and $M \in \mathbb{N}^r$ we define $Q^\omega := \{p \in P \mid Q(p) = \omega\}$ and $Q^{\text{fin}} \in \mathbb{N}^r$ by $Q^{\text{fin}}(p) = 0$ for $p \in Q^\omega$ and $Q^{\text{fin}}(p) = Q(p)$ otherwise. $Q =^{\text{fin}} M$ is the relation $\forall p \notin Q^\omega : M(p) = Q(p)$.

Now we are ready to define the coverability tree and the coverability graph.

Definition. The *coverability tree* $\text{CT}(N)$ of a Petri net $N = (P, T, B, F, M_0)$ consists in a set of nodes R labelled by elements of $\mathbb{N}_\omega^r$ $(\text{card}(P) = r)$, a set of arcs X, labelled by T, a root r_0, and is defined by the following conditions:

(a) the root is labelled by M_0,

(b) If s is a node labelled by Q, then s has no successor when either:

(1) on a path from r_0 to s there is a node $s' \neq s$, also labelled by Q, or

(2) there is no transition t such that $Q \geqslant B(\cdot, t)$.

(c) If s is labelled by Q and s does not satisfy conditions (1) or (2), then for each $t \in T$ such that $Q \geqslant B(\cdot, t)$ there is a successor s' labelled by Q' with:

(i) $Q'(p) = \omega$ for all places $p \in P$, for which there is a node s'' on the path from r_0 to s (including s) labelled by Q'' and $Q'' \leqslant Q + D(\cdot, t)$ and $Q''(p) < Q(p) + D(p, t)$.

(ii) $Q'(p) = Q(p) + D(p, t)$ otherwise.

The arc from s to s' is labelled by t.

It has been shown in [10] that this tree constructed by this procedure is finite for any net. In the following definition of the overability graph nodes of the tree having the same label, are fused.

For proving properties of the coverability graph an arc from Q to Q' labelled by t is written (Q, t, Q'). A path is a sequence of such arcs. Sometimes only the sequence of nodes of a path is given. The sequence of labels on the arcs is called the word of the path. A circuit is a path ending in the starting node. A circuit is elementary if all nodes are different with the exception of the start and end nodes.

DEFINITION. The *coverability graph* $CG(N)$ of a net N is defined by the set of nodes Q, which are the labels of the nodes in the coverability tree $CT(N)$. There is an arc from Q to Q' labelled by t iff in $CT(N)$ there is an arc labelled by t, which links a node s labelled by Q to a node s' labelled by Q'.

THEOREM 1. *Let $N = (P, T, F, B, M_0)$ be a Petri net with coverability graph $CG(N)$.*

(a) *If (Q_1, t, Q_2) is an arc in $CG(N)$, then $Q_1 \geqslant B(\cdot, t)$ and $Q_2(p)$ is either ω or $Q_1(p) + C(p, t)$ for all $p \in P$, in particular $Q_1^\omega \subseteq Q_2^\omega$.*

(b) *If Q is a node in $CG(N)$, then $\forall k \in \mathbb{N}\ \exists M_k \in R(N) : (M_k =^{\text{fin}} Q$ and $\forall p \in Q^\omega : M_k(p) \geqslant k)$.*

(c) *A place $p \in P$ is unbounded iff $p \in Q^\omega$ for some node Q of $CG(N)$.*

(d) *If $M_0(w\rangle M$ in N, then there is a unique path in $CG(N)$, labelled by w and leading from M_0 to a node Q satisfying $Q =^{\text{fin}} M$.*

(e) *If $w \in T^*$ is a word labelling a circuit in $CG(N)$, then $M(w\rangle$ for some reachable marking M of N.*

Proof. Part (a) follows immediately from the definition of $CG(N)$. Parts (b), (c) and (d) are shown in [10] for the coverability tree. For (e) let Q be the initial node of

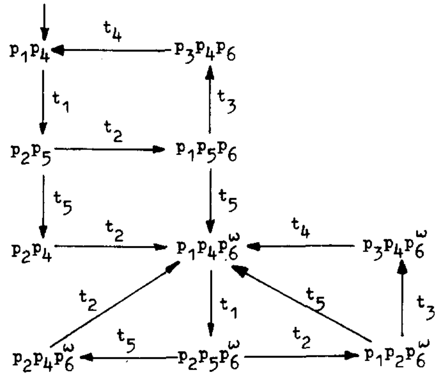

FIG. 3. The coverability graph of the net of Fig. 1.

the circuit labelled by w. From (b) for $k := \max\{B(p, w) \mid p \in Q^\omega\}$ there is reachable marking $M \in R(N)$ with $M =^{\text{fin}} Q$ and $M(p) \geqslant k$ for all $p \in Q^\omega$. If t is the first transition of w, then by (a) we have for all $p \notin Q^\omega: M(p) = Q(p) \geqslant B(p, t)$ and therefore by the definition of k also $M \geqslant B(\cdot, t)$, i.e., t can fire in M. Using the last assertion of (a) it is obvious that for all nodes Q_i of the circuit $Q_i = Q$. Therefore also the following transitions of w can fire, i.e., $M_0(*)M(w)$.

In Fig. 3 the coverability graph of the Petri net of Fig. 1 is given. The notation of markings introduced in Section 2.2 is extended to $\mathbb{N}^r_\omega$. An arrow at the top labels the initial node. The graph shows that place p_6 is unbounded, but all other places are bounded. If transition t_5 is omitted, we obtain the coverability graph of Fig. 4 showing that the new net is bounded.

If N is a bounded net, the set of nodes of the coverability graph $CG(N)$ is the set of reachable markings $R(N)$, as in the example of Fig. 4. Therefore in this case the coverability graph is called the reachability graph of the net. This graph can be seen as a finite automaton with the initial marking as initial state and all nodes as accepting states. The language accepted by this automaton is the language of firing sequences $F(N)$ of the net. Therefore the languages of bounded nets are regular, but the languages of unbounded nets can be regular, as in the case of Fig. 1, or not regular. This shows that the class of Petri nets having a regular language of firing sequences properly contains the class of bounded Petri nets.

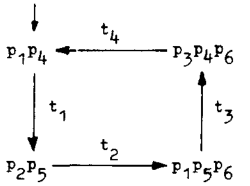

FIG. 4. The reachability graph of the Petri net of Fig. 1 with transition t_5 omitted.

Definition. A Petri net N is *regular*, if its language of firing sequences $F(N)$ is a regular language.

The net of Fig. 1 is an example of a regular unbounded net N. In Section 2.3 the language $F(N)$ of this net was given by a regular expression.

In the proof of Theorem 1(c) the following *fundamental property* of $\mathbb{N}^r_\omega$ is used: Every infinite sequence $(M_i)_{i>0}$ in $\mathbb{N}^r_\omega$ contains an infinite subsequence $(M_{i_j})_{j>0}$ such that $M_{i_j} \leqslant M_{i_{j+1}}$ for all $j > 0$.

Since this property is also essential in the proofs of the following section, we formulate the property explicitly here.

4. Properties of Unbounded Places and of Places Bounded Below

Let $P_1 = \{p_1, p_2\}$ be a set of places of a net. If there is an integer k, such that in every reachable marking at least one of p_1 and p_2 contains not more than k tokens, then the set P_1 is called bounded. In this case p_1 and p_2 can be unbounded individually. If a bound k holds for all reachable markings and for all members of P_1, then P_1 is said to be uniformly bounded.

Not only the increase of tokens, but also their decrease, is of importance. If the number of tokens that can be moved away from a set of places P_1 by firings of transitions is bounded, the set P_1 is called bounded below or similarly as before, uniformly bounded below. For this definition the places of P_1 are supplied with a sufficient amount of additional tokens. To express this in a formal way, we use the conventions that follow.

In this section we are using a fixed Petri net $N = (P, T, B, F, M_0)$, where $P = \{p_1, \ldots, p_r\}$ and $T = \{t_1, \ldots, t_s\}$. For $P_1 \subseteq P$ the vector $U_{P_1} \in \mathbb{N}^r$ is the *characteristic vector* of P_1 and is defined by $U_{P_1}(p) = 1$ if $p \in P_1$ and $U_{P_1}(p) = 0$ if $p \notin P_1$. Then for $M \in \mathbb{N}^r$ and $n \in \mathbb{N}$ the sum $M + n \cdot U_{P_1}$ is the marking M with n added in the components corresponding to elements of P_1.

4.1. *Definitions and Simple Properties*

Definition. Let $M \in \mathbb{N}^r$ be a marking and $P_1 \subseteq P$ a set of places of N.

(a) P_1 is *bounded* for M, if $\exists k \in \mathbb{N}\ \forall M_1 \in R(M)\ \exists p_1 \in P_1; M_1(p_1) \leqslant k$.

(b) P_1 is *uniformly bounded* for M, if $\exists k \in \mathbb{N}\ \forall M_1 \in R(M)\ \forall p_1 \in P_1$: $M_1(p_1) \leqslant k$.

(c) P_1 is *bounded below* for M, if $\exists k \in \mathbb{N}\ \forall n \in \mathbb{N}\ \forall M_1 \in R(M + n \cdot U_{P_1})$ $\exists p_1 \in P_1: M_1(p_1) \geqslant M(p_1) + n - k$. the drop is bounded

(d) P_1 is *uniformly bounded* below for M, if $\exists k \in \mathbb{N}\ \forall n \in \mathbb{N}$ $\forall M_1 \in R(M + n \cdot U_{P_1})\ \forall p_1 \in P_1: M_1(p_1) \geqslant M(p_1) + n - k$.

A set P_1 of places is uniformly bounded for M if every place of P_1 is bounded with

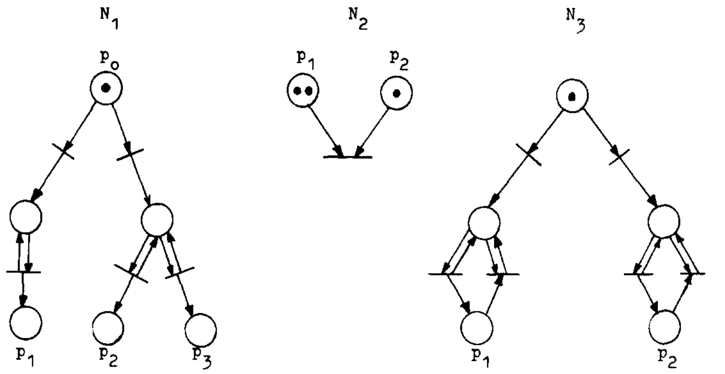

FIG. 5. Examples of Petri nets N_1, N_2 and N_3.

respect to M as initial marking. Intuitively a set P_1 is uniformly bounded below for M if the number of tokens that can be removed from places of P_1 by the firing of an arbitrary firing sequence of transitions is bounded, even if these places received additional tokens before.

In the Petri net N_1 of Fig. 5 the set $\{p_1, p_2, p_3\}$ is bounded, but not uniformly bounded for the marking $M = p_0$. The set $\{p_2, p_3\}$ is not bounded for the same marking. The sets $\{p_1\}$ and $\{p_2\}$ of the Petri net N_2 in Fig. 5 are bounded below for every marking M, but $\{p_1, p_2\}$ is not uniformly bounded below for any marking M. The set $\{p_1, p_2\}$ is bounded below and bounded in N_3 for $M = p_0$, but neither uniformly bounded nor uniformly bounded below.

The following theorem states some simple properties connecting these concepts.

THEOREM 2. *Let P_1 be a subset of places and M a marking of the net N.*

(a) *If P_1 is uniformly bounded for M then P_1 is bounded for M.*

(b) *If P_1 is uniformly bounded below for M, then P_1 is bounded below for M.*

(c) *P_1 is uniformly bounded for M, if and only if for every place $p \in P_1$ the set $\{p\}$ is bounded for M.*

(d) *If P_1 is bounded below for M, then for every place $p \in P$ the set $\{p\}$ is bounded below.*

The proof of this theorem follows immediately from the definitions. The examples given above show that the equivalence in statements (a). (b) and (d) of Theorem 2 would be false.

4.2. *Maximally Unbounded Places and Maximal Nodes*

~diagonal

If a set of places P_1 is not bounded for the initial marking M_0 of a Petri net, then for any given integer n the reachability set contains a marking, such that the number of tokens in the places of P_1 simultaneously exceed n. The content of the places not in P_1 is called the context. More precisely, we have the following definition.

DEFINITION. Let $P_1 \subseteq P$ be a nonempty subset of places and M_1 a marking of N.

(a) P_1 is *unbounded with context* M_1, if $M_1(p)=0$ for all $p \in P_1$ and $\forall k \in \mathbb{N}$ $\exists M_2 \in R(N)$ $(\forall p \in P_1 : M_2(p) \geqslant k$ and $\forall p \in P - P_1 : M_2(p) = M_1(p))$.

(b) Let $\mathfrak{C}$ be the set of couples (P', M'), such that P' is unbounded with context M'. A partial ordering relation $\leqslant$ is defined on $\mathfrak{C}$ by: $(P', M') \leqslant (P'', M'')$ iff $P' \subseteq P''$ and $\forall p \notin P'' : M'(p) \leqslant M''(p)$. As usual, the set $(P', M')^{\leqslant}$ will be set $\{(P'', M'') \mid (P', M') \leqslant (P'', M'')\}$. (P', M') is *maximal* when $(P', M')^{\leqslant} = \{(P', M')\}$, i.e., $(P', M') \leqslant (P'', M'')$ implies $P' = P''$ and $M' = M''$. In this case we also say, that P' is *maximally unbounded with context* M'.

LEMMA 1. *Every couple* $(P_1, M_1) \in \mathfrak{C}$ *has a maximal element* (P_m, M_m) *in* $(P_1, M_1)^{\leqslant}$.

Proof. Let $(P_1, M_1) \in \mathfrak{C}$. Since P is finite, there is in the set $\{P' \mid \exists M' : (P_1, M_1) \leqslant (P', M')\}$ an element P_m, which is maximal for $\subseteq$. If $(P_1, M_1)^{\leqslant}$ does not have a maximal element, then there exists a sequence $(P_m, M_i)_{i>1}$ of pairwise distinct elements in $(P_1, M_1)^{\leqslant}$. We can assume that $M_i \leqslant M_{i+1}$. This is possible due to the fundamental property of $\mathbb{N}^r$ mentioned before.

Since $M_i \neq M_{i+1}$ and from the definition of $\mathfrak{C}$, we conclude that for all $i > 1$ there exist $p \notin P_m$ with $M_i(p) < M_{i+1}(p)$. Consider the set $P' := \{p \mid \text{card}\{M_i(p) \mid i > 0\}$ is infinite$\}$. Then $P' \neq \varnothing$ and $P' \cap P_m = \varnothing$. Since $\{M_i(p) \mid p \in P - P',\ i > 0\}$ is finite, there is a subsequence $(M_{i_j})_{j>0}$ of $(M_i)_{i>0}$ such that $M_{i_j}(p) = M_{i_1}(p)$ for all $p \in P - P'$ and $j > 0$. For $P'' := P_m \cup P'$ and $M'' \in \mathbb{N}^r$ defined by $M''(p) = M_{i_1}(p)$ if $p \notin P''$ and $M''(p) = 0$ if $p \notin P''$, we obtain $(P'', M'') \in (P_1, M_1)^{\leqslant}$ and $(P'', M'') \geqslant (P_m, M_m)$. Since $P_m \subsetneq P''$ is in contradiction with the definition of P_m, $(P_1, M_1)^{\leqslant}$ has a maximal element.

In the following definition and lemma, maximal pairs are related to maximal nodes of the coverability graph.

DEFINITION. A node $Q \in \mathbb{N}_\omega^r$ of the coverability graph is *maximal* if $Q' \geqslant Q$ implies $Q' = Q$ for all nodes Q' of the graph.

LEMMA 2. *A nonempty set of places* $P_1 \subseteq P$ *is maximally unbounded with context* M, *if and only if the coverability graph contains a maximal node* Q *such that* $P_1 = Q^\omega$ *and* $M_1 = Q^{\text{fin}}$.

Proof. Assume that P_1 is unbounded with context M_1, and (P_1, M_1) is maximal. Then for any integer $k > 0$ there is a firing sequence $M_0 (w_k\rangle M_k$ such that $M_k(p) \geqslant k$ for all $p \in P_1$ and $M_k(p) = M_1(p)$ for all $p \notin P_1$.

Since $CG(N)$ is a finite graph, there is a constant k such that $Q(p) \neq \omega$ implies $Q(p) < k$ for all nodes Q and $p \in P$. For $M_0 (w_k\rangle M_k$ by Theorem 1(d) there is a node Q_k with $Q_k =^{\text{fin}} M_k$. Since $Q_k(p) > k$ implies $Q_k(p) = \omega$ for all $p \in P_1$, we have $P_1 \subseteq Q_k^\omega$ and $Q_k =^{\text{fin}} M_1$. Suppose $Q_k \neq P_1$. Then by Theorem 1(b) Q_k^ω is unbounded with context Q_k^{fin} and $(P_1, M_1) \lessneqq (Q_k, Q_k^{\text{fin}})$ holds in contradiction to the assumption that (P_1, M_1) is maximal. Therefore $Q_k = P_1$ and $Q_k^{\text{fin}} = M_1$.

To prove the maximality of Q, assume the contrary, i.e., $Q_m \gneqq Q$ for some node Q_m. Again by Theorem 1(b) Q_m^{ω} is unbounded with context Q^{fin}, but $(Q_m, Q_m^{\text{fin}}) \gneqq (P_1, M_1)$ contradicts the fact, that (P_1, M_1) is maximal.

For the proof of the inverse implication, assume that Q is a maximal node in the coverability graph and $Q^{\omega} = P_1$. By Theorem 1(b) the set P_1 is unbounded with context Q^{fin}. To prove that (P_1, M_1) is maximal, from Lemma 1 we conclude that $(P_1, M_1)^{\leqslant}$ contains a maximal couple (P_m, M_m). By the first part of this proof, for (P_m, M_m) there is also a maximal node Q_m, satisfying $Q_m^{\omega} = P_m$ and $Q_m^{\text{fin}} = M_m$. Since $Q \leqslant Q_m$ is assumed to be maximal, we have $Q = Q_m$ and $(P_1, M_1) = (P_m, M_m)$ is a maximal pair.

The next lemma characterizes those maximally unbounded places which are at the same time unbounded below, by properties of the coverability graph.

LEMMA 3. *N has a nonempty maximally unbounded set P_1 with context M_1, such that P_1 is not uniformly unbounded below, if and only if there is a circuit labelled w in the coverability graph, that starts from a maximal node Q and a place $p_1 \in Q^{\omega}$ with $D(p_1, w) < 0$.*

Proof. Suppose that there are Q, w and p_1 having the properties of the lemma. By Lemma 2. $P_1 := Q^{\omega}$ is maximally unbounded with context $M_1 := Q^{\text{fin}}$. Since there is a circuit labelled by w and starting and ending in Q, the firing sequence w can fire in M_1, if enough tokens are added to the places of P_1. This number can be taken as $d := \max\{B(p, w) \mid p \in P_1\}$. Since w labels a circuit, w can fire k time, if $k \cdot d$ is taken instead of d. The marking reached by firing of w^k is denoted by M_k:

$$\forall k: (M_1 + k \cdot d \cdot U_{P_1})(w^k\rangle M_k.$$

Since $D(p_1, w) < 0$, it follows:

$$M_k(p_1) = M_1(p_1) + k \cdot d + k \cdot D(p_1, w) < M_1(p_1) + k \cdot d - k.$$

Therefore P_1 is not uniformly bounded below.

Conversely, if P_1 is maximally unbounded with context M_1, then by Lemma 2. The coverability graph contains a maximal node Q_1 satisfying $(P_1, M_1) = (Q^{\omega}, Q_1^{\text{fin}})$.

If in addition P_1 is not uniformly unbounded below, we have $\forall k \in \mathbb{N}\ \exists n_k\ \exists M_k\ \exists w_k\ \exists p_k \in P_1$:

$$M_1 + n_k \cdot U_{P_1}(w_k\rangle M_k \text{ and } M_k(p_k) \leqslant M_1(p_k) + n_k - k.$$

Since P is finite, there is a place $p_1' \in P_1$ with $p_k = p_1'$ for infinitely many k. Therefore we can assume $p_1 = p_2 = p_3 = \cdots = p_1'$. For every w_k there is a path in $CG(N)$ starting in Q_1 and labelled by w_k. This follows from $(P_1, M_1) = (Q^{\omega}, Q^{\text{fin}})$. The greatest possible decrease of tokens in a place by firing a single transition is bounded by $d := -\min\{D(p, t) \mid p \in P, t \in T\}$. Since $D(p_1, w_k) < 0$ it follows $d > 0$.

Let γ be the number of nodes of the coverability graph and $l := \gamma \cdot d$. While firing $M_1 + n_1 \cdot U_{p_1} \langle w_l \rangle M_l$ the number of tokens in p_1 is decreased by at least $l : M_l(p_1) \leqslant M_1(p_1) + n_l - 1$. By $l = \gamma \cdot d$ this process of removing tokens from p_1 can be subdivided into γ steps $M'_1(u_1\rangle M'_2(u_2\rangle \cdots (u_\gamma\rangle M'_{\gamma+1} = M_l$, where $M'_1 = M_1 + k' \cdot U_{p_1}$ and in each step p_1 is decreased: $M'_i(p_1) < M'_{i+1}(p_1)$ for $1 \leqslant i < \gamma$.

In the coverability graph a path labelled by $w_1 = u_1 u_2 \cdots u_\gamma$ starts from Q_1. If the nodes reached by the prefixes $u_1 u_2 \cdots u_j$ $(1 \leqslant j \leqslant \gamma)$ are denoted by Q_{j+1}, then two of these nodes, let us say Q_{j_1} and Q_{j_2} $(1 \leqslant j_1 < j_2 \leqslant \gamma + 1)$, must be equal. They are connected by a path labelled by $v = u_{j_1+1} \cdots u_{j_2}$. Hence this path is a circuit starting from $Q := Q_{j_1} = Q_{j_2}$ and is labelled by v, which has the desired property: $D(p_1, v) < 0$. Since there is a path leading from Q_1 to Q, we also have $p_1 \in Q_1^\omega \subseteq Q^\omega$. To finish the proof, it is left to show that these properties of Q also hold for a maximal node Q_m.

By Lemma 1, $(Q, Q^{\mathrm{fin}})^\leqslant$ contains a maximal pair (P_m, M_m). On the other hand this pair corresponds to a maximal node Q_m with $(P_m, M_m) = (Q_m^\omega, Q_m^{\mathrm{fin}})$ by Lemma 2. By $Q_m \geqslant Q$ also Q_m is contained in a circuit labelled w.

5. Regularity is Decidable for Petri Nets

In this section we use the results of Section 4 to prove a necessary and sufficient for the regularity of $F(N)$. This property is shown to be decidable.

5.1. *Proof of the Decidability*

By the first theorem a net is regular iff for every place and every reachable marking the number of tokens that can be removed by firings of transitions is bounded by a constant k.

this characterisation fails for nondeterministic nets

THEOREM 3. *A Petri net $N = (P, T, B, F, M_0)$ is regular, if and only if there is an integer k such that*

bounded drop

$$\forall M \in R(N)\ \forall M' \in R(M)\ \forall p \in P: M'(p) \geqslant M(p) - k.$$

Proof. If N and k satisfy the conditions of the theorem, by a constant $c := k + \max\{M_0(p) \mid p \in P\} + \max\{B(p, t) \mid p \in P, t \in T\}$ we define a finite automaton $\mathfrak{A} = (E, T, \delta, e_0, E')$ accepting the language $F(N)$. $E := \{M \in \mathbb{N}^r \mid M(p) \leqslant c\} \cup \{e_g\}$, consisting in a set of markings of N and a garbage state e_g, is the set of states. $e_0 := M_0$ is the initial state. $E' := E - \{e_g\}$ is the set of final states and δ is the state transition function $\delta: E \times T \to E$. For $M \in E'$ and $t \in T$ $\delta(M, t)$ is defined by M' if $M \geqslant B(\cdot, t)$ and $M'(p) = \min\{c, M(p) + D(p, t)\}$ and $\delta(M, t) := e_g$ if $M \geqslant B(\cdot, t)$ does not hold. $\delta(e_g, t)$ is e_g for all $t \in T$. It is easy to see that $w \in T^*$ can fire in M_0 iff $\delta(e_0, w) \neq e_g$, i.e., $\mathfrak{A}$ accepts $F(N)$.

Conversely assume that the condition does not hold: $\forall k \in \mathbb{N}$ $\exists M_k \in R(N)$

pk

$\exists M'_k \in R(M_k)$ $\exists p_k : M'_k(\cancel{p}) < M_k(p_k) - k$. If $F(N)$ is accepted by a finite automaton $\mathfrak{A} = (E, T, \delta, e_0, E')$, then by the same reasoning as in the proof of Lemma 3 two words $v, w \in T^*$ can be found, satisfying $vw \in F(N)$ and $\delta(e_0, v) = \delta(e_0, vw) \in E'$ and $D(p, w) < 0$ for some place $p \in P$. Hence, $vw^* \subseteq F(N)$. But this is impossible since by $D(p, w) < 0$ the sequence w cannot fire an arbitrary number of times.

THEOREM 4. *A Petri net $N = (P, T, B, F, M_0)$ is not regular, if and only if there is a marking M and a set of places $P_1 \subseteq P$ that is maximally unbounded with context M, but not uniformly bounded below for M.*

Proof. If N is not regular, by Theorem 3 for every integer k there are markings $M_k \in R(N)$, $M'_k \in R(M_k)$, such that $M'_k(p_k) < M_k(p_k) - k$ for a place p_k. Since P is finite at least one place p' appears infinitely often. Therefore we can assume $p_k = p'$ for all $k > 0$. By the fundamental property of vectors of $\mathbb{N}^k$, mentioned in Section 3, the sequence $(M_k)_{k>0}$ contains an increasing subsequence, and we can also assume $M_k \leqslant M_{k+1}$ for all $k > 0$. p' is contained in the set P' of places p, for which $\{M_k(p) \mid k \geqslant 0\}$ is not finite. Since by this definition of P' the values $M_k(p)$ are bounded for all $p \notin P'$ and $k > 0$, there must exists a subsequence $(M_{k_i})_{i>0}$ of $(M_k)_{k>0}$, satisfying $M_{k_j}(p) = M_{k_1}(p)$ for all $j > 0$ and $p \notin P'$. In addition it is possible to choose this subsequence in such a way that $k_i > i$ for all $i > 0$. Now by defining $M'(p) = 0$ for $p \in P'$, and $M'(p) = M_{k_1}$ for $p \notin P'$, P' is unbounded with context M', i.e., $(P', M') \in \mathfrak{C}$. By Lemma 2 $(P', M')^{\leqslant}$ contains a maximal pair (P_1, M_1).

To finish the first part of the proof, it is left to show that P_1 is not uniformly bounded below for M_1. For $i > 0$ let $w_{k_i} \in T^*$ be the firing sequence leading from M_{k_i} to M'_{k_i}. Let $n_{k_i} := \max\{B(p, w_{k_i}) \mid p \in P\}$ then w_{k_i} can fire, if n_{k_i} tokens are added in each component $p \in P'$ of M'. The same holds for M_1, if P' is replaced by P_1, i.e.,

$$M_1 + n_{k_i} \cdot U_{P_1} (w_{k_i}\rangle M''_{k_i}$$

for some marking M''_{k_i}. Since $k_i > i$, for every $i > 0$ there is a number $n_{k_i} > 0$ such that $M_1 + n_{k_i} \cdot U_{P_1} (w_{k_i}\rangle M''_{k_i}$ and $M''_{k_i}(p') = n_{k_i} + D(p', w_{k_i}) = n_{k_i} + M'_{k_i} - M_{k_i} < n_{k_i} - k_i < n_{k_i} - i$. This shows that P_1 is not uniformly bounded below for M_1.

Conversely, suppose that P_1 is maximally unbounded with context M_1 and not uniformly unbounded below for M_1. We will show that for every integer $k > 0$ there are $M \in R(N)$, $M' \in R(M)$, $p \in P$ satisfying $M'(p) < M(p) - k$. Then N is not regular by Theorem 3.

Since P_1 is not uniformly unbounded below, for every $k > 0$ there are $p_k \in P$, $n_k > 0$, $w_k \in T^*$, $M_k \in \mathbb{N}^r$ for which $M_1 + n_k \cdot U_{P_1} (w_k\rangle M_k$ and $M_k(p_k) < M_1(p_k) + n_k - k$ hold. (P_1, M_1) is maximal and by Lemma 2. The coverability graph contains a maximal node Q with $(Q^\omega, Q^{\text{fin}}) = (P_1, M_1)$. By Theorem 1(b) we obtain from Q a reachable marking $M \in R(N)$ with $M(p) \geqslant n_k$ for $p \in P_1$ and $M(p) = M_1(p)$ for $p \notin P_1$. Hence for every $k > 0$ the sequence w_k can fire in M and the marking obtained by such a firing of w_k in M has the property we wanted to prove: $M'(p_k) = M(p_k) + D(p_k, w_k) = M(p_k) + M_k(p_k) - (M_1 + n_k \cdot U_{P_1})(p_k) = M(p_k) + M_k(p_k) - n_k < M(p_k) + M_1(p_k) - k = M(p_k) - k$.

The net is not regular iff there exists an unbounded place which can have arbitrary drops. → bounded places cannot have unbounded drops!

THEOREM 5. *The regularity of a Petri net N is decidable.*

Proof. In Theorem 4 and Lemma 3 we have shown that N is not regular iff there is a circuit starting in some node Q of $CG(N)$, which is labelled by a word $w \in T^*$, such that $D(p, w) < 0$ for some $p \in Q^\omega$.

We now show that it is sufficient to consider this property for elementary circuits only. Suppose that the circuit is not elementary. Then w can be decomposed into $w = v_1 v_2 v_3$, $v_2 \neq \lambda$, where v_1 labels a path leading from Q to a node Q_1 and v_2 labels an elementary circuit in Q_1. From Theorem 1(a) we have $Q^\omega = Q_1^\omega$.

There are two possible cases:

(a) If $D(p, v_2) \geqslant 0$ then $D(p, v_1 v_3) \leqslant D(p, w)$ and $v_1 v_3$ labels a shorter circuit in Q with $D(p, v_1 v_3) < 0$. This construction can be repeated until case (b) holds.

(b) If $D(p, v_2) < 0$ then v_2 also labels a maximal node $Q_m \geqslant Q_1$. To test regularity of a net N, it is therefore sufficient to check whether an elementary circuit, labelled by some word $w \in T^*$ and starting in a maximal node Q of the coverability graph, satisfies $D(p, w) < 0$ for some place $p \in Q^\omega$.

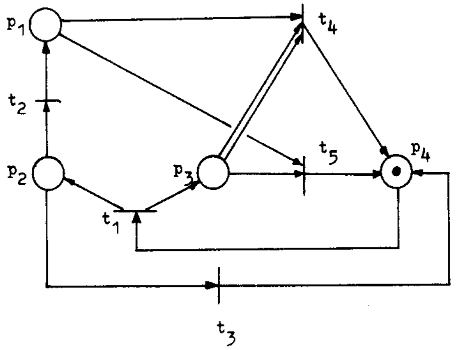

FIG. 6. A nonregular Petri net.

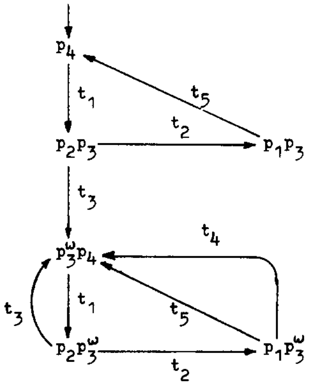

FIG. 7. To coverability graph of the Petri net of Fig. 6.

To illustrate this procedure, consider the net of Fig. 6. In the coverability graph of this net (Fig. 7) we find an elementary circuit labelled by $w = t_1 t_2 t_4$. $Q = p_3^\omega p_4$ is a maximal node in this circuit and $p_3 \in Q^\omega$ satisfies $D(p_3, w) = -1 < 0$. Therefore, the net is not regular.

Since the regularity of a net, as defined in this paper, depends on a given intial marking, this property is not necessarily preserved if the initial marking changes. In a structurally regular net, however, this is not the case.

DEFINITION. A Petri net $N = (P, T, B, F, M_0)$ is called *structurally regular*, if the net $N_i = (P, T, B, F, M_i)$ is regular for arbitrary initial marking $M_i \in \mathbb{N}^r$.

THEOREM 6. *It is decidable, whether a Petri net N is structurally regular or not.*

Proof. By the proof of Theorem 4 a net $N = (P, T, B, F, M_0)$ is not regular iff $M_0(*\rangle M_1(*\rangle M_2(*\rangle M_3(*\rangle M_4$ holds for markings $M_1, ..., M_4$ satisfying:

(a) $M_1 \leqslant M_2$ and $M_1 \neq M_2$ and

(b) $M_1(p) \geqslant M_2(p)$ implies $M_3(p) \leqslant M_4(p)$ for every $p \in P$ and

(c) $M_3(p) > M_4(p)$ for some $p \in P$.

Now, for vectors $X, Y \in \mathbb{Z}^r$, where $r = \text{card}(P)$, define $X[*\rangle Y :\Leftrightarrow \exists v \in T^* : Y = X + D(\cdot, v)$. If N is not regular, then $O^r[*\rangle X_1[*\rangle X_2[*\rangle X_3[*\rangle X_4$ holds for vectors $X_1, ..., X_4 \in \mathbb{Z}^r$ and the null-vector O^r, which satisfy conditions (a), (b) and (c) when $M_1, ..., M_4$ are replaced by $X_1, ..., X_4$. This condition is not only necessary but also sufficient for the net to be not structurally regular, since a sufficient large initial marking M_i can be chosen.

Since it is decidable whether there are X, $Y \in \mathbb{Z}^r$ and $Z \in \mathbb{Z}^s$ such that $Y = X + D \cdot Z$, this condition is decidable

5.2. *Effectiveness of the Decision Procedure*

There is not only a procedure which decides whether a given net is regular, but also one which effectively gives the automaton accepting the regular language. However, every procedure has a very high complexity, as shown in Theorem 8.

THEOREM 7. *There is an effective procedure that associates to every regular Petri net N a finite automaton accepting F(N).*

Proof. If N is regular, then for every word w labelling a circuit of $CG(N)$ and every node of the circuit and $p \in Q^\omega$, we have $D(p, w) \geqslant 0$. In order to find the maximum number k of tokens that can be removed from places, it is sufficient to check all circuit free paths. With $k := \max\{Q, -D(p, w) \mid p \in P$ and w labels a circuit-free path$\}$ the construction of the automaton is given by the proof of Theorem 3.

The algorithm is not primitive recursive, since it uses a construction of a graph similar to the one described in [10]. One can ask then, if there is a better procedure that is primitive recursive. This is a natural question, especially in view of the results of [14], where a procedure for the decision of boundedness is given, that works in exponential space.

DEFINITION. The *size* $s(N)$ of a Petri net $N=(P,T,B,F,M_0)$ is defined by $s(N):=\sum_{p\in P, t\in T}(\log(F(p,t+1)+\log(B(p,t)+1))+\sum_{p\in P}(\log(M_0(p)+1)$.

The definition of $s(N)$ is natural because it gives the amount of tape needed on a Turing machine to encode the net N.

THEOREM 8. *For any primitive recursive function f, there is an infinity of regular nets N, such that the number of states of any automaton accepting $F(N)$ is greater than $f(s(N))$.*

Proof. The proof goes along the sames lines as that used in [7] to show that the set of reachable markings grows in a nonprimitive recursive way. Let $N_0=(P_0,T_0,B_0,F_0,M_0)$ be the net of Fig. 8, and $N_0^n=(P_0,T_0,B_0,F_0,M_0^n)$ with $M_0^n:=M_0+n\cdot U_{\{c_0\}}$, i.e., one token in b_0 and n tokens in c_0. It is easy to see that N_0^n is bounded for every $n\in\mathbb{N}$ and that the maximum number of tokens in c_0 is $2\cdot n$. There is a reachable marking M with $M(c_0)=2\cdot n$. Consider the sequence of nets $(N_i)_{i\geqslant 0}$, where N_i is obtained from N_{i-1} in the way described in Fig. 9. If (P_i,T_i,B_i,F_i,M_{0i}) is the net N_i, then we define for $n\in\mathbb{N}$ the net $N_i^n=(P_i,T_i,B_i,F_i,M_{0i}^n)$ by a new initial marking $M_{0i}^n:=M_{0i}+n\cdot U_{\{c_i\}}$.

Intuitively, the net N_i^n works in the following way. If n tokens are placed in place $\mathbf{c}_i$, then by firing transition $\mathbf{b}$ n-times, we obtain n tokens in $\mathbf{c}_{i-1}$ and $\mathbf{c}_i'$. Now by transition $\mathbf{a}$ the subnet N_{i-1} can be started, which is assumed to compute a maximum number of $f(n)$ tokens in $\mathbf{c}_{i-1}$. By the construction of the net N_i and by the number of tokens in $\mathbf{c}_i'$ the subnet N_{i-1} can be started at most n-times. After the execution of all n repetitions $f(f(\dots f(n)\dots))=f^n(n)$ tokens can be collected in $\mathbf{c}_{i-1}$. Then all these tokens can be transported to $\mathbf{c}_i$. Hence, the maximum number of tokens to be reached in place $\mathbf{c}_i$ of the net N_i is given by a function f_i, which is recursively defined by

$$f_0(n)=2\cdot n,$$
$$f_i(n)=f_{i-1}^n(n) \qquad \text{for} \quad i\geqslant 1.$$

(f^n is the nth application of f.)

It is well known that $(f_i)_{i\geqslant 0}$ is a sequence of primitive recursive functions and that every primitive recursive function is majorized by a function from this sequence.

Consider now the net $\hat{N}_i^n$ constructed from N_i^n for every $n\geqslant 0$ as in Fig. 10. For this net the language is $F(\hat{N}_i^n)=\{wt^m \mid \exists w\in F(N_i^n): M_{0i}^n(w\rangle M \text{ and } m\leqslant M(c_i)\}$. This language is regular, since N_i^n is bounded. The values of m have $f_i(n)$ as smallest upper bound. Therefore any automaton accepting $F(\hat{N}_i^n)$ has at least $f_i(n)$ states.

The size of the net $\hat{N}_i^n$ is $s(\hat{N}_i^n)=k\cdot i+\log(n+2)$, where k is a constant independent of i. Therefore there is an integer n_1 such that $f_i(n)>f_i(s(\hat{N}_i^n))$ for all $n\geqslant n_1$. This proves the theorem.

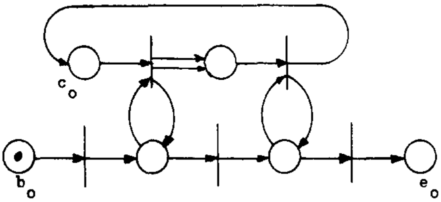

FIG. 8. The Petri net N_0.

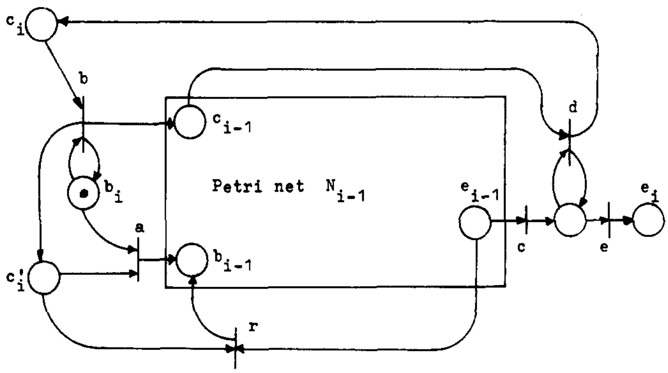

FIG. 9. Construction of N_i from N_{i-1}.

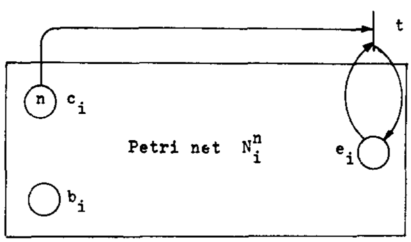

FIG. 10. Construction of $\hat{N}_i^n$ from N_i^n.

Remark. For a bounded net, the coverability graph is isomorphic to an automaton accepting the set of firing sequences. This property does not hold for unbounded nets. For an example, consider the net N with its coverability graph of Fig. 11. This net is regular, but the coverability graph is not isomorphic to any automaton accepting $F(N)$, otherwise the word $t_2 t_1 t_1$ would be in $F(n)$, which is not the case. The maximum number of tokens that can be removed from a place is two. The automaton that accepts $F(N)$ and is constructed by the procedure described in Theorem 3 is given in Fig. 12. As can be seen, it is not minimal, but Theorem 8 ensures that it is not much bigger than the minimal automaton accepting $F(N)$.

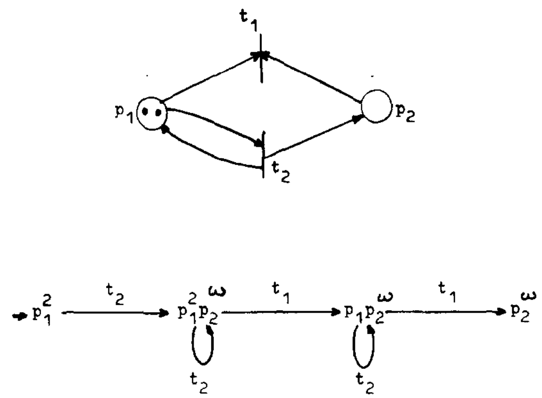

FIG. 11. Petri net N and its coverability graph.

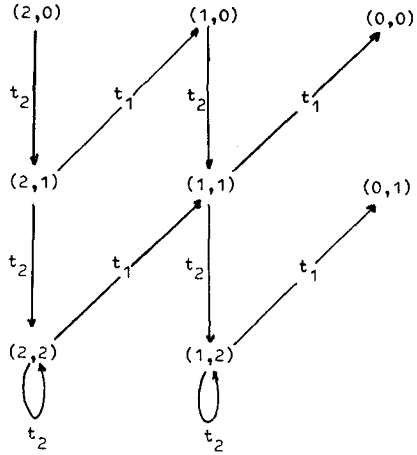

FIG. 12. The automaton accepting the set of firing sequences of the net of Fig. 11, from the proofs of Theorems 3 and 4. The garbage state, and the arcs leading to it, are not represented.

5.3. *Decidable Properties of Regular Nets*

An interesting feature of regular nets is that important properties are decidable.

THEOREM 9. *For regular Petri nets, the following problems are decidable*:

> *Reachability (given a net N and a marking M does $M \in R(N)$?). Inclusion (given N and N', is $R(N)$ included in $R(N')$?). Liveness of a transition (given a net N and a transition t, $\forall M \in R(N)$ $\exists M' \in R(M)\, M'(t\rangle$?).*

Proof. If N is regular, then the language of firing sequences is semi-linear, and the set of reachable markings is semi-linear in an effective way, i.e., there is a procedure giving its description in the form of a semi-linear set. This implies the decidability of reachability and inclusion.

For liveness, t is live for any final state s of the finite automaton accepting $F(N)$, there is a word w such that $w \cdot t$ leads from s to a final state s'. This last property is decidable since only words of length less than the number of states have to be checked.

Remarks. It is shown in [6] that reachability and liveness are recursively equivalent, but the reductions used destroy regularity of nets. Therefore two distinct proof were needed in Theorem 9.

In general properties that can be expressed in the strong monadic theory of two successors (see [13]) will be decidable for regular Petri nets. Such properties are: fairness, and the finite delay property.

6. Regularity is Undecidable for Languages in $\mathscr{L}_0$

In this section we show that regularity is undecidable in $\mathscr{L}_0$. The method used in the proof also applies to other undecidability proofs, such as the equality problem in $\mathscr{L}_0$ and $\mathscr{L}$. In addition we are able to give some lower bounds on the number of unbounded places for which these results still hold.

6.1. *Simulation of Program Machines*

Using a simulation of program machines by Petri nets, we show that regularity is undecidable for terminal labelled Petri net languages. Therefore we first recall the definition of a program machine as given in [11].

Definition. A *program machine* PM is given by a finite set $R = \{r_1, ..., r_p\}$ of *registers*, a finite set $Q = \{q_0, ..., q_r\}$ of *labels* and a finite set I of *instructions*. Each instruction is labelled by a element of Q and no two instructions have the same label. PM has exactly one start-instruction

$$q_0 : \textbf{start goto } q_1;$$

and exactly one halt-instruction

$$q_f : \textbf{halt}.$$

The other instructions are of the type

increment register r_i: $\quad q_s : r_i := r_i + 1 \textbf{ goto } q_m;$

or

test and decrement register r_i: $\quad q_s : \textbf{if } r_i = 0 \textbf{ then goto } q_m$
$\textbf{else } r_i := r_i - 1 \textbf{ goto } q_l;$

The registers have nonnegative integers as values. First the start-instruction is executed. Then the flow of control is determined by the goto contained in each instruction, until the halt-instruction is reached.

A *computation* is a sequence: q_0, $(k_1^1,..., k_1^p, q_1)$, $(k_2^1,..., k_2^p, q_{i_2}),..., (k_j^1,..., k_j^p, q_{i_j}),...,$ $(k_r^1,..., k_r^p, q_{i_r})$ with $q_{i_r} = q_f$, i.e., the halt-statement is reached. q_{i_j} is the actual instruction to be executed in step j and k_j^s is the content of register r_s just before the execution of instruction q_{i_j} in step j. In computations arbitrary intitial values $(k_1^1,..., k_1^p) \in \mathbb{N}^p$ are allowed.

The effect of each instruction should be clear now, and is therefore not formally defined. For illustration we give the following example defined. For illustration we give the following example of a program machine *PM*.

PM: q_0: **start goto** q_1;

q_1: **if** $r_1 = 0$ **then goto** q_f **else** $r_1 := r_i - 1$ **goto** q_1;

q_f: **halt.**

For the initial value $(3, 0)$ of (r_1, r_2) the program machine has the computation: q_0, $(3, 0, q_1)$, $(2, 0, q_1)$, $(1, 0, q_1)$, $(0, 0, q_1)$, $(0, 0, q_f)$.

Notice that a program machine is deterministic. By the definition of the start instruction every computation contains at least q_0, q_1, and q_f.

universality for labelled Petri nets is undecidable

THEOREM 10. *It is undecidable whether the terminal language $L(N, \mathfrak{M})$ of a λ-free labelled Petri net $N = (N_1, h, X)$ is equal to X^* or not.*

(Minsky machine)

Proof. Consider an arbitrary program machine *PM* with two registers. A computation of *PM* can be written as a word over the alphabet $X = Q \cup \{0, 1\}$ of labels of *PM*, augmented by 0 and 1 in the following way:

$$w = q_0 1^{k_1^1} 0 1^{k_1^2} q_1 1^{k_2^1} 0 1^{k_2^2} q_{i_2} \cdots 1^{k_r^1} 0 1^{k_r^2} q_{i_r}, \qquad \text{where} \quad q_{i_r} = q_f.$$

For the example we have $w = q_0 1^3 0 q_1 1^2 0 q_1 1 0 q_1 0 q_1 0 q_f$. The set of all such computations defines the language $L(PM) := (w \in X^* \mid w$ describes a computation of $PM\}$.

We now show that there is a labelled net $N = (N_1, h, X)$, which has exactly those words in X^* in its terminal language $L(N, \mathfrak{M})$, that are not computations of *PM*, i.e., $L(N, \mathfrak{M}) = X^* - L(PM)$. Having shown this, the proof ends as follows: $L(N, \mathfrak{M})$ equals X^*, if and only if $L(PM) = \varnothing$. If $L(N, \mathfrak{M}) = X^*$ would be decidable, then the emptiness problem for $L(PM)$ would be decidable. This is known not to be the case, even if only program machines with two registers are considered.

Now, the net $N = (N_1, h, X)$ will be constructed. A word $w \in X^*$ is not in $L(PM)$, iff:

(a) w is not in the regular set $q_0 1^* 0 1^* q_1 (1^* 0 1^* Q)^* 1^* 0 1^* q_f$ or when w contains a subword $v = 1^{k_i} 0 1^{k_i'} q_i 1^{k_j} 0 1^{k_j'} q_j$, where

(b) q_j is not the correct label following q_i with k_i and k'_i in the registers, or

(c) k_j and k'_j are not the correct contents of registers r_1 and r_2, respectively,

after the execution of the instruction labelled q_i, with k_i and k'_i in registers r_1 and r_2, respectively.

Condition (a) and (b) can be checked by a finite automaton, and therefore also by a labelled Petri net.

To show how a labelled Petri net can check condition (c), consider the net in Fig. 13. This net corresponds to that subcase of condition (c), where q_i labels an instruction of the type

$$q_i : r_1 := r_1 + 1 \textbf{ goto } q';$$

By this subnet all words $w \in X^*$ belonging to $L(N, \mathfrak{M})$, contain a subword like v, but with $k_j > k_i + 1$ in contradiction to the instruction q_i.

To explain this, we say that a word $u \in X^*$ is "read" by the net, if u are the labels of a sequence of transitions that have fired. In Fig. 13 the labels of the transitions are given in place of their names. The net works as follows: When the token in p_1 has reached p_2, an arbitrary subword of the type of v is selected, i.e., all letters in w before v are read. Then in p_2 letters 1 are read. If the token has reached p_3, the place

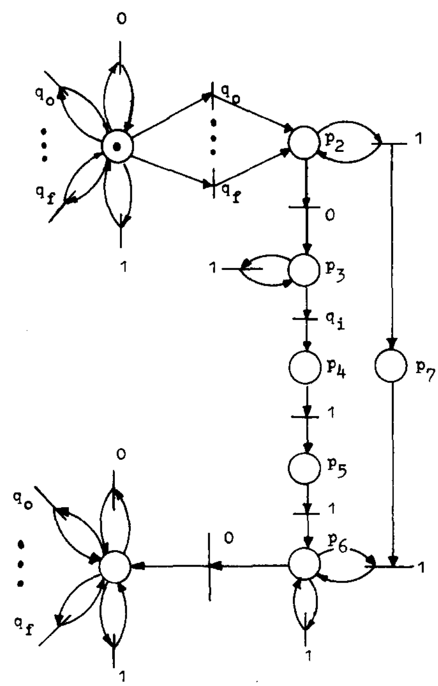

FIG. 13. Petri net checking a subcase of condition (c).

p_7 contains k_i tokens. Now the next subword of k'_i letters 1 and $q_i \in Q$ is read. Before reaching p_6, where p_7 is decreased again, two additional letters 1 are read. Also in p_6 a number $m \geqslant 0$ of letters 1 is read without decreasing p_7. Then the rest of the word is read. The set of terminal markings $\mathfrak{M}$ contains only one marking M, where $M(p_8) = 1$ and all other places are empty. Therefore, since p_7 is also empty for the number k_j we have $k_j = 2 + k_i + m$ and $m \geqslant 0$, i.e., $k_j > k_i + 1$.

For the other cases the construction of the corresponding subnet is similar. All these subnets are combined together in such a way that by starting the whole net, exactly one subnet is selected and executed in a nondeterministic way.

All subnets have at most one unbounded place (p_7 in the example). In the composed net these unbounded places can be substituted by one common unbounded place. So, the net N has only a single unbounded place.

6.2. *Application to Proofs of Undecidability and Nonclosure for Complementation*

In the following corollary we give an application of this theorem by showing that equality is undecidable for terminal languages of labelled nets. Moreover this result holds true, if only one of the two nets has one unbounded place. This strengthens a result found in [7].

COROLLARY 1. *Given two λ-free labelled Petri nets, one of them having one unbounded place and the other none, it is undecidable whether their terminal languages are equal.*

Proof. For any alphabet X there is clearly a λ-free labelled and bounded net having X^* as terminal language. Therefore, if the problem of the corollary were decidable, then the problem of Theorem 10 would be decidable also. This has been shown to be not the case. It is important to remember that the net constructed in the proof of Theorem 10 has only one unbounded place.

This result is optimal, since a net having no unbounded places is regular. Then the equality problem is decidable. Next, we prove that regularity is also undecidable for this class of nets.

reduction from universality of labelled PN languages

COROLLARY 2. *The regularity problem for languages of $\mathscr{L}_0$ is undecidable.*

Proof. We first show that the language $L(PM)$ of a program machine is regular, if and only if it is finite. If $L(PM)$ is finite, it is clearly regular. Conversely assume that $L(PM)$ is infinite, i.e. there are infinitely many computations of PM. Since PM is a deterministic machine, there are also infinitely many initial values $(k_1^1, k_1^2, \ldots, k_1^p)$ in these computations. Therefore the initial values of at least one register r_i are unbounded. In the second step, which always exists, k_2^i differs from k_1^i at most by one. By the pumping lemma for regular languages it easily follows that $L(PM)$ cannot be regular. Hence, $L(PM)$ is regular iff it is finite.

To prove the corollary, let $L(PM)$ and $L(N, \mathfrak{M})$ be as in the proof of Theorem 10, i.e., $L(N, \mathfrak{M}) = X^* - L(PM)$. Now $L(N, \mathfrak{M})$ is regular, if and only if $L(PM)$ is finite, which is known to be not decidable.

From [7] we know that $\mathscr{L}_0^\lambda$ is not closed by effective complementation, if the reachability problem is decidable. We now prove this property without referring to the reachability problem. Since the proof uses the same idea as the proof of Theorem 10, we included it in this paper.

THEOREM 11. *$\mathscr{L}_0^\lambda$ is not closed by complementation.*

Proof. Let be $X = \{a, b\}$ and w^R the reversal of a word $w \in X^*$. If $L_0 := \{ww^R \mid w \in X^*\}$ then we first show that $L_1 := X^* - L_0$ is in $\mathscr{L}_0^\lambda$. Let $lg(w)$ denote the length of the word w.

A word $w \in X^*$ is in L_1, iff

(a) $lg(w)$ is odd or

(b) $w = w_1 w_2$ with $lg(w_1) = lg(w_2)$ and $w_1 = u_1 x u_2$, $w_2 = v_1 y v_2$ with $x, y \in X$ and

(b1) $lg(u_2) = lg(v_2)$ and

(b2) $x \neq y$.

Condition (a) can be checked by a labelled terminal net. To test condition (b) consider the labelled net in Fig. 14 with the single terminal marking M, where $M(p) = 1$ if $p = p_6$ and $M(p) = 0$ otherwise. In this net condition (b1) is tested by $M(p_5) = 0$ in the terminal marking and places p_3 and p_4 ensure the holding of condition (b2).

Assume that $\mathscr{L}_0^\lambda$ is closed under complementation. Then also $L_0 \in \mathscr{L}_0^\lambda$. Since $\mathscr{L}_0^\lambda$ is an intersection-closed full semi-AFL (see [7], [12]), by Corollary 3 of [1]. $\mathscr{L}_0^\lambda$ is the set r.e. of all recursively enumerable languages. But r.e. is not closed by complementation in contradiction to the assumption.

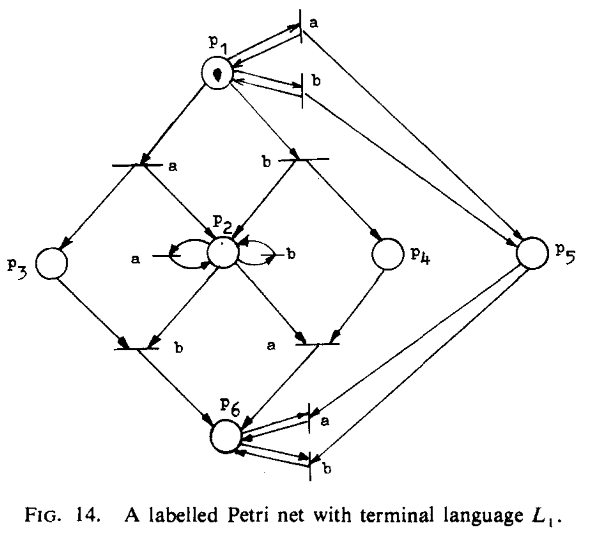

FIG. 14. A labelled Petri net with terminal language L_1.

The use of words describing computations enables us to improve the result in [7] about the undecidability for equivalence in $\mathscr{L}$.

THEOREM 12. *The equality problem for languages of two λ-free labelled Petri nets, one of them having four and the other having five unbounded places, is undecidable.*

Proof. Let PM be a program machine with two registers.

Two functions f_1 and f_2 are defined by $f_i: I \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $f_i(q,m,n) = n-1$ when q labels an instruction $r_i := r_i + 1$ **goto** q' and $f_i(q, m, n) = n+1$ when q labels an instruction **if** $r_i = 0$ **then goto** q' **else** $r_i := r_i - 1$ **goto** q''; and $m \neq 0$, and $f_i(q, m, n) = n$ otherwise.

A word $w = q_{i_1} 1^{k_1^1} 0 1^{k_1^2} q_{i_2} \cdots q_{i_r} 1^{k_r^1} 0 1^{k_r^2} q_{i_{r+1}}$ satisfies the following condition (C), if for all $n \in \mathbb{N}$ with $2n \leqslant r$

$$\alpha_n^i := \sum_{l=1}^{n} (k_{2l-1}^i - f_i(q_{2l}, k_{2l-1}^i, k_{2l}^i)) \geqslant 0$$

and for all m with $2m + 1 \leqslant r$

$$\beta_n^i := \sum_{l=1}^{n} (k_{2l}^i - f_i(q_{2l-1}, k_{2l}^i, k_{2l+1}^i)) \geqslant 0.$$

For the example of PM in Section 6.1 consider the word $w = q_0 1^3 0 q_1 1^2 0 q_1 1 0 q_1 0 q_1 0 q_f$. w satisfies condition (C), since $\alpha_1^1 = \alpha_1^2 = \alpha_2^1 = \alpha_2^2 = \beta_1^1 = \beta_1^2 = 0$, the word $w' = q_0 1^4 0 q_1 1^4 0 q_1$ does not since $\alpha_1^1 = -1$.

A word w does not describe a computation through inferiority, when it satisfies condition (a) or (b) given in the proof of Theorem 9, or the condition (c′) obtained in replacing "is not the correct content" by "is smaller than the correct content" in condition (c).

Let $L'(PM)$ and $L''(PM)$ be the following languages:

> $L(PM) := \{w¢ \mid w$ does not describe a computation through inferiority and w satisfies condition $(C)\} \cup (Q \cup \{1,0\})^*$,
> $L''(PM) := \{w¢ \mid w$ satisfies condition $(C)\} \cup (Q \cup \{1,0\})^*$.

We want to show that $L'(PM) = L''(PM)$, if and only if $L(PM) = \varnothing$. If $w \in L(PM)$ then $w \notin L'(PM)$ and $w \in L''(PM)$. Conversely, by $L'(PM) \subseteq L''(PM)$ we have to prove that $w \in L''(PM)$ and $w \notin L'(PM)$ imply $w \in L(PM)$. Since $w \notin L'(PM)$, we have $w = c_{i_1} 1^{k_1^1} 0 1^{k_1^2} q_{i_2} \cdots 0 1^{k_r^2} q_{i_{r+1}}$ with $q_{i_1} = q_0$, $q_{i_2} = q_1$ and $q_{i_{r+1}} = q_f$.

Therefore for $l = 1$ the word $q_{i_1} 1^{k_1^1} 0 1^{k_1^2} q_{i_{l+1}}$ represents the contents of registers r_1 and r_2 and the next instruction at step number 1. Suppose this is true for all $l < n$. From the definition of f_1 and f_2, for all r with $2r < n$ and all s with $2s + 1 < n$ we have $\alpha_r^1 = \alpha_r^2 = \beta_s^1 = \beta_s^2 = 0$.

Let us examine the case where n is even and where q_{i_n} labels the instruction

$r_1 := r_1 + 1$ **goto** q'. The other cases are quite similar. Since $w \notin L'(PM)$, we have $q' = q_{i_{n+1}}$ and since w satisfies condition (C) also $\alpha^1_{n'-1} = k^1_{n-1} - (k^1_n - 1) = \alpha^1_{n'} \geqslant 0$ with $2n' = n$. Therefore $k^1_{n-1} + 1 \geqslant k^1_n$. Since w does not satisfy conditions c′), we have $k^1_n = k^1_{n-1} + 1$.

We have shown by recurrence that for all n at step number n the content of the register r_1 is k^1_n. Since $q_{i_{r+1}} = q^f$, w describes a computation of PM.

One concludes the proof by checking that one can define, given a program machine PM two λ-free labelled Petri nets, without terminal markings N_1 and N_2', such that $L(N_1) = L'(PM)$ and $L(N_2) = L''(PM)$. To check condition (C) one needs two unbounded places for each register, and one needs one more unbounded place to check condition (c′).

As a last application, consider the language $L_2 := \{w \not\subset w^R \mid w \in \{a, b\}^*\}$, where w^R is the reversal of w. This language is typical of languages not in $\mathscr{L}_0^\lambda$.

THEOREM 12. *It is undecidable, whether the terminal language $L(N, \mathfrak{M})$ of a given λ-free labelled Petri net N contains L_2.*

Proof. A word does not describe a computation through superiority, when its satisfies condition (a) or (b) of the proof of Theorem 9, or condition (c″) obtained from condition (c) by replacing "is not the correct content" by "is greater than the correct content."

For a program machine PM the language $\{w_1 \not\subset w_2 \mid w_1$ does not describe a computation through inferiority or w_2^R does not describe a computation through superiority$\}$ contains L_2, if and only if $L(PM) = \varnothing$. It is easy to see that this language is in $\mathscr{L}_0$.

ACKNOWLEDGMENTS

The authors are grateful to the referees for finding some errors.

REFERENCES

1. B. BAKER AND R. BOOK, Reversal-bounded multipushdown machines, *J. Comput. System Sci.* **8** (1974), 315–332.
2. S. CRESPI-REGHIZZI AND D. MANDRIOLI, Petri nets and Szilard languages, *Inform. Contr.* **33** (1977), 177–192.
3. H. J. GENRICH, K. LAUTENBACH, AND P. S. THIAGARAJAN, Elements of general net theory, *in* "Net Theory and Appliations," Lecture Notes in Computer Sci. No. 84, pp. 23–163, Springer-Verlag, Berlin 1980.
4. A. GINZBURG AND M. YOELI, Vector addition systems and regular languages, *J. Comput. System Sci.* **20** (1980), 277–284.
5. S. A. GREIBACH, An infinite hierarchy of context-free languages, *J. Assoc. Comput. Mach.* **16** (1969), 91–106.
6. M. HACK, "Decidability Questions for Petri Nets," Lab. for Comp. Sci., Techn. Rep. 161, 1976.
7. M. HACK, "Petri Net Languages," Comp. Struct. Group Memo 124, Project MAC, M.I.T., 1975.

8. M. Jantzen, On the hierarchy of Petri net languages, *RAIRO Inform. Théor.* **13** (1979), 19–30.
9. M. Jantzen and R. Valk, Formal properties of place/transition nets, *in* "Net Theory and Applications," Lecture Notes in Computer Sci. No. 84, pp. 166–212, Springer-Verlag, Berlin, 1980.
10. R. M. Karp and R. E. Miller, Parallel program schemata, *J. Comput. System Sci.* **3** (4) (1969), 167–195.
11. M. Minsky, "Computation: Finite and Infinite Machines," Prentice–Hall, Englewood Cliffs, N. J., 1967.
12. J. L. Peterson, Computation sequence set, *J. Comput. System Sci.* **13** (1976), 1–24.
13. M. O. Rabin, Decidability of second order theories and automata on infinite trees, *Trans. Amer. Soc.* **141** (1969), 1–35.
14. C. Rackoff, The covering and boundedness problems for vector addition systems, *Theoret. Comput. Sci.* **6** (1978), 223–231.
15. R. Valk and G. Vidal-Naquet, On the rationality of Petri net languages, *in* "Theoretical Computer Science," Lecture Notes in Comput. Sci. No. 48, pp. 319–328, Springer-Verlag, Berlin, 1977.