Theoretical Computer Science 39 (1985) 155-169
North-Holland

A LOGICAL APPROACH OF PETRI NET LANGUAGES

Michel PARIGOT
UER de Mathematiques et Informatique, CNRS U.A. 753, Université Paris VII, 75251 Paris Cedex 05, France

Elisabeth PELZ
Laboratoire de Recherche en Informatique, CNRS U.A. 410, Université Paris-Sud, 91405 Orsay Cedex, France

Communicated by D. Perrin
Received January 1985

Abstract. For languages recognized by finite automata we dispose of two formalisms: regular expressions (Kleene, 1956) and logical formulas (Büchi, 1960). In the case of Petri net languages there is no formalism like regular expressions. In this paper we give a Büchi-like theorem which characterizes Petri net languages in terms of second-order logical formulas. This characterization has two advantages: (1) It situates exactly the power of Petri nets with respect to finite automata; roughly speaking, Petri nets are finite automata plus the ability of testing if a string of parenthesis is well formed (in this paper 'parenthesis' always means the usual one sort of parentheses). (2) Given a language, it enables us to easily prove that it is a Petri net language.

In addition we prove that Petri net languages and deadlock languages coincide.

Introduction

This paper is organized as follows: Section 1 gives basic definitions (Section 1.1), languages of Petri nets (Section 1.2), and normalization of Petri nets (Section 1.3). In Section 2, a logical characterization of Petri net languages is given. In Section 3, the main arguments of the theorems from Section 2 are proven. Section 4 concludes with some remarks on the use of the logical formalism (Section 4.1) and on the difference between Petri net languages and weak Petri net languages.

The following standard notations will be used. A natural number n also denotes the set $\{0, \ldots, n-1\}$. The cardinality of a set A is always written $|A|$. Letters with overbar indicate finite sequences; in particular, $\bar{a} \in A$ means that $\bar{a}$ is a finite sequence of elements of A. If φ is a formula, then $\varphi(\bar{X}, \bar{x})$ indicates that all free set variables of φ are among $\bar{X}$ and all free individual variables of φ are among $\bar{x}$. If L is a language over an alphabet Σ, then $CL = \Sigma^* \backslash L$ denotes the complement of L.

1. Preliminaries on Petri nets

1.1. Basic definitions

A *Petri net* is a tuple $R=(P, T, f, b)$ where $P=\{p_1, \ldots, p_r\}$ is a finite set of *places*, $T=\{t_1, \ldots, t_s\}$ a finite set of *transitions*, $f: P \times T \to \mathbb{N}$ a *forward incidence function*, and $b: P \times T \to \mathbb{N}$ a *backward incidence function.*

A *marking* of P is a mapping $M: P \to \mathbb{N}$; if $M(p)=i$, we say that place p contains i tokens. A transition t is said to be *firable* at a marking M iff for all places $p \in P$ we have $M(p) \geqslant f(p, t)$, which is denoted $M(t\rangle$. If a firable transition t fires, it takes, for each place $p \in P$, $f(p, t)$ tokens from p and adds $b(p, t)$ tokens to p. The resulting marking M' is such that, for all $p \in P$, $M'(p)=M(p)-f(p, t)+b(p, t)$, and one writes: $M(t\rangle M'$. An ordering on markings is defined by: $M_1 \leqslant M_2$ iff, for all $i \leqslant r$, $M_1(p_i) \leqslant M_2(p_i)$.

A sequence $\bar{t}=(t_1, \ldots, t_n) \in T$ is a *firing sequence* at marking M_1 if there exist markings $M_2, \ldots, M_{n+1}$ such that, for all $i \leqslant n$, $M_i(t_i\rangle M_{i+1}$; a short notation is $M_1(\bar{t}\rangle M_{n+1}$.

We call A the set of *arcs* of R, defined by $A=\{(p, t) \mid f(p, t) \geqslant 1\} \cup \{(t, p) \mid b(p, t) \geqslant 1\}$, and we write $p \to^{f(p,t)} t$ or $t \to^{b(p,t)} p$ for elements of A. When $f(p, t)$ or $b(p, t)$ is one, we omit the valuation of the arc.

For a given place $p \in P$, the set $I(p)$ of *input transitions* is $\{t \in T \mid (p, t) \in A\}$ and the set $O(p)$ of *output transitions* is $\{t \in T \mid (t, p) \in A\}$.

1.2. Languages of Petri nets

Languages over an alphabet Σ are associated to Petri nets by labelling the transitions by letters of Σ and by specifying some initial and final conditions on markings.

A *λ-free labelled Petri net* is a Petri net R together with a λ-free labelling of T in the alphabet Σ, $e: T \to \Sigma$; a marking of P, called *initial marking*, M_0, and a finite set of markings, called *final markings*, F.

In this paper, when we mention a Petri net R we always mean a λ-free labelled Petri net $R=(P, T, f, b, e, M_0, F)$ and, when we modify it in a Petri net R' or R_1, we assume $R'=(P', T', f', b', e', M_0', F')$ or $R_1=(P_1, T_1, f_1, b_1, e_1, M_{0_1}, F_1)$.

We extend the labelling to sequences of transition by setting $e(t_1, \ldots, t_n)= e(t_1)e(t_2)\ldots e(t_n)$.

The *language of* R is defined by

$$L(R)=\{w \in \Sigma^+ \mid \exists M_f \in F\ \exists \bar{t} \in T^+\ M_0(\bar{t}\rangle M_f \text{ and } e(\bar{t})=w\}.$$

The class L of all such languages is called class of Petri net languages.

The *deadlock language of* R is defined by

$$T(R)=\{w \in \Sigma^+ \mid \exists M \in \mathbb{N}^r\ \exists \bar{t} \in T^+, M_0(\bar{t}\rangle M, e(\bar{t})=w \text{ and no transition is firable at } M\}.$$

The class T of all such languages is called class of Petri net deadlock languages.

The *weak language of R* is defined by

$$G(R) = \{w \in \Sigma^* \mid \exists M \in \mathbb{N}^r\ \exists M_f \in F\ \exists \bar{t} \in T^+\ M_0(\bar{t})M,\ M \geqslant M_f \text{ and } e(\bar{t}) = w\}.$$

The class G of all such languages is called class of weak Petri net languages.

Remark. We follow Peterson [4] by designing these classes by L, T, and G; other authors (Valk and Vidal-Naquet, cf. [6]) use $\mathscr{L}_0$ instead of L.

1.3. Normalization of Petri nets

In order to obtain a clearer proof of the logical characterization we shall consider all Petri nets in a normal form. We say that a Petri net R is *normal* if it satisfies the following properties:

(P1): the forward and backward incidence functions of R take their values in $\{0, 1\}$,

(P2): the initial and final markings of R take their values in $\{0, 1\}$.

Proposition 1.1. *Let R be an arbitrary Petri net. Then there is a normal Petri net R' such that $L(R) = L(R')$ and $G(R) = G(R')$.*

Remark. It is known how to associate to an arbitrary Petri net a Petri net satisfying (P1) using λ-labelled transitions [1, 5]. But, opposed to automata, for Petri nets the addition of λ-labelled transitions modifies the associated class of languages. The proof of the next lemma gives a new construction not using λ-labelled transitions.

The proof of Proposition 1.1 follows from Lemmas 1.2 and 1.3.

Lemma 1.2. *Let R be an arbitrary Petri net. Then there is a Petri net R_1 satisfying* (P1) *such that $L(R) = L(R_1)$ and $G(R) = G(R_1)$.*

Proof. Let $Q \subseteq P$ be the set of places q for which there exists at least one transition t with $f(q, t) > 1$ or $b(q, t) > 1$. We proceed by induction on $|Q|$; thus, we just have to indicate how to decrease this number by one. Let $q \in Q$ and $n = \sup\{i \in \mathbb{N} \mid \exists t \in T\ f(q, t) = i \text{ or } b(q, t) = i\}$. We transform R into a net R' with $Q' = Q \setminus \{q\}$.

First replace q by n new places $q_0, \ldots, q_{n-1}$. For each transition $t \in I(q) \cup O(q)$ let $(D_i)_{i<n_t}$ and $(E_j)_{j<m_t}$ be enumerations of the subsets of $\{q_0, \ldots, q_{n-1}\}$ with $f(q, t)$ and $b(q, t)$ elements respectively, and replace t by $n_t m_t$ transitions $(t_{i,j})_{i<n_t, j<m_t}$ with the same label as t and such that, for all (i, j):

(i) the arcs between $t_{i,j}$ and the old places $p \in P\backslash\{q\}$ are the same as those between t and these places,

(ii) we add the arcs $r \to t_{i,j}$ and $t_{i,j} \to s$ for all $r \in D_i$, $s \in E_j$.

(Note that if $f(q, t) = 0$ or $b(q, t) = 0$, then $n_t = 1$ and $D_0 = \emptyset$ (respectively $m_t = 1$, $E_0 = \emptyset$).)

Write $M_0(q)$ as $nm + m'$ with $m, m' \in \mathbb{N}$, $m' < n$. The new initial marking M_0' is given by $M_0' \restriction_{P\backslash\{q\}} = M_0$, $M_0'(q_i) = m + 1$ for $i < m'$ and $M_0'(q_i) = m$ for $m' \leqslant i < n$.

Instead of each $M_f \in F$ we take a marking M_f' which satisfies $M_f' \restriction_{P\backslash\{q\}} = M_f$ and $\sum_{i<n} M_f'(q_i) = M_f(q)$. □

Lemma 1.3. *Let R be a Petri net satisfying* (P1). *Then there is a normal Petri net R_1 such that $L(R) = L(R_1)$ and $G(R) = G(R_1)$.*

Proof. Let $Q \subseteq P$ be the set of places p for which there exists a marking $M \in F \cup \{M_0\}$ such that $M(p) > 1$. We proceed by induction on $|Q|$; thus, we just have to indicate how to decrease this number by one. Let $q \in Q$ and $n = \sup\{M(q) \mid M \in F \cup \{M_0\}\}$.

In order to transform R into a net R' with $Q' = Q\backslash\{q\}$, we replace q by n new places $q_0, \ldots, q_{n-1}$ and each transition $t \in I(q) \cup O(q)$ by n new transitions $t_0, \ldots, t_{n-1}$ with the same label as t and such that, for all $i < n$:

(i) the arcs between q_i and the old transitions $s \in T\backslash(I(q) \cup O(q))$ are the same as those between q and s,

(ii) for all $t \in I(q) \cup O(q)$, the arcs between t_i and the old places are the same as those between t and these places and the arcs between q_i and t_i are the same as those between q and t.

We obtain the new initial marking M_0' by setting $M_0' \restriction_{P\backslash\{q\}} = M_0$, $M_0'(q_i) = 1$ for $i < M_0(q)$ and $M_0'(q_i) = 0$ for $M_0(q) \leqslant i < n$.

Instead of each $M_f \in F$ we take a marking M_f' satisfying $M_f' \restriction_{P\backslash\{q\}} = M_f$, $\sum_{i<n} M_f'(q_i) = M_f(q)$ and, for all $i < n$, $M_f'(q_i) \leqslant 1$. □

2. Logical characterization of Petri net languages

2.1. Interpretation in logic

Let Σ be an alphabet. We associate to Σ a *logical vocabulary* S_Σ, consisting of a binary relation symbol $\leqslant$ and, for each $\sigma \in \Sigma$, an unary relation symbol P_σ. A word $\sigma_0, \ldots, \sigma_{n-1} \in \Sigma^+$ will be considered as a *model for* S_Σ over the universe $n = \{0, 1, \ldots, n-1\}$ where $\leqslant$ is interpreted by the natural order on the integers and P_σ by $\{i < n \mid \sigma_i = \sigma\}$. Thus, each sentence φ of the vocabulary S_Σ defines a language over Σ: the set of words (seen as models for S_Σ) which satisfy φ. (Note that not all finite models for S_Σ are words, but there is a sentence ψ of S_Σ defining words. Hence the 'language defined by a sentence φ' is in fact the language defined by $\varphi \wedge \psi$.)

It is well known that the regular languages over Σ are exactly the languages defined by a monadic second-order sentence of the vocabulary S_Σ.

third order!

To characterize in similar manner the languages of Petri nets we add to the vocabulary two second-order relation symbols $\leqslant_g$ and $=_g$ whose interpretations are partial orders between subsets of n defined as follows:

$$X \leqslant_g Y \quad \text{iff} \quad \text{for all } m \leqslant n,\ |X \cap m| \leqslant |Y \cap m|,$$

$$X =_g Y \quad \text{iff} \quad X \leqslant_g Y \text{ and } |X| = |Y|.$$

These relations are quite natural: for $\Sigma = \{(,)\}$, $\sigma_0, \ldots, \sigma_{n-1} \in \Sigma^+$ is a well formed string of parentheses if and only if $\{i < n \mid \sigma_i =)\} =_g \{i < n \mid \sigma_i = (\}$.

More precisely, we define a *monadic second-order logic* $\mathscr{L}$ relative to Σ. The *symbols of* $\mathscr{L}$ are the following:

- individual variables $v_0, \ldots, v_n, \ldots$;
- set variables $V_0, \ldots, V_n, \ldots$;
- the connectives $\wedge$, $\vee$, $\neg$, $\rightarrow$;
- the quantifiers $\forall$, $\exists$;
- the equality symbol $=$;
- first-order relation symbols $\leqslant$ (binary) and P_σ (unary), for all $\sigma \in \Sigma$;
- second-order relation symbols $\leqslant_g$ (binary) and $=_g$ (binary).

Atomic formulas of $\mathscr{L}$ are expressions of type $x = y$, $x \leqslant y$, Ux, $U \leqslant_g V$, $U =_g V$ for individual variables x, y and set variables or unary first-order relation symbols U, V.

Formulas of $\mathscr{L}$ are obtained by application of the following rules: (i) an atomic formula is a formula; (ii) whenever φ, ψ are formulas, also $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$, and $\varphi \rightarrow \psi$ are formulas; (iii) if φ is a formula and x an individual variable, then $\forall x\varphi$ and $\exists x\varphi$ are formulas; and (iv) if φ is a formula and X a set variable, then $\forall X\varphi$ and $\exists X\varphi$ are formulas.

A *quantifier-free* $\mathscr{L}$*-formula* is a formula obtained by repeated application of rules (i) and (ii). A *first-order* $\mathscr{L}$*-formula* is a formula obtained by the application of rules (i), (ii), and (iii). A $\mathscr{L}_1$*-formula* is a $\mathscr{L}$-formula not involving second-order relation symbols.

All models we shall consider are words (in the above sense). Thus, the meaning of formulas is completely determined by the interpretation of $\leqslant_g$ and $=_g$ on words given above together with the usual rules of the *semantics of monadic-second order logic*, which we recall informally:

- Xx means "the individual x is a member of the set X";
- $\varphi \wedge \psi$ (respectively $\varphi \vee \psi$) means "φ and (respectively or) ψ";
- $\varphi \rightarrow \psi$ means "if φ then ψ";
- $\neg\varphi$ means "not φ";
- $\forall x\varphi(x)$ (respectively $\forall X\varphi(X)$) means "for all individual x (respectively subset X) of the universe, $\varphi(x)$ (respectively $\varphi(X)$)";
- $\exists x\varphi(x)$ (respectively $\exists X\varphi(X)$) means "there is some individual x (respectively subset X) of the universe, such that $\varphi(x)$ (respectively $\varphi(X)$)".

2.2. Logical characterizations

We are interested in $\mathscr{L}$-sentences which define languages of Petri nets. Since Büchi's work it is well known that $\mathscr{L}$-sentences not involving $\leqslant_g$ and $=_g$ (i.e., $\mathscr{L}_1$-sentences) define the regular languages; more precisely, we can state the following theorem.

Theorem 2.1. *The following conditions are equivalent*:
(i) *L is a regular language.*
(ii) *L is defined by a $\mathscr{L}_1$-sentence.*
(iii) *L is defined by a $\mathscr{L}_1$-sentence of type $\exists \bar{X}\varphi(\bar{X})$, where $\varphi(\bar{X})$ is first-order.*

The $\mathscr{L}$-sentences will give us a similar characterization of Petri net languages. But, we cannot hope for a condition like (ii) saying that every $\mathscr{L}$-sentence defines a Petri net language, since they are not closed under complementation (a result which easily follows from our characterization, see Section 4). However, a condition like (iii) will hold.

Let S be a set of formulas. We call *positive combinations* of elements of S the elements of the closure of S under $\wedge$ and $\vee$: clearly, all positive combinations of elements of S can be written in normal form as disjunctions (respectively conjunctions) of conjunctions (respectively disjunctions) of elements of S.

Theorem 2.2. *Let L be a language over an alphabet Σ. The following conditions are equivalent*:
(i) *L is a Petri net language.*
(ii) *L is a Petri net deadlock language.*
(iii) *L is defined by a $\mathscr{L}$-sentence of type $\exists \bar{X}\varphi(\bar{X})$, where $\varphi(\bar{X})$ is a first-order $\mathscr{L}$-formula.*
(iv) (*Normal Form* I) *L is defined by a $\mathscr{L}$-sentence of type $\exists \bar{X}\varphi(\bar{X})$, where $\varphi(\bar{X})$ is a positive combination of formulas of type $X =_g Y$ and first-order $\mathscr{L}_1$-formulas.*
(v) (*Normal Form* II) *L is defined by a $\mathscr{L}$-sentence of type $\exists \bar{X}\varphi(\bar{X})$, where $\varphi(\bar{X})$ is a boolean combination of formulas of type $X \leqslant_g Y$ and first-order $\mathscr{L}_1$-formulas.*

Theorem 2.3. *Let L be a language over an alphabet Σ. The following conditions are equivalent*:
(i) *L is a weak Petri net language.*
(ii) *L is defined by a $\mathscr{L}$-sentence of type $\exists \bar{X}\varphi(\bar{X})$, where $\varphi(\bar{X})$ is a positive combination of formulas of type $X \leqslant_g Y$ and first-order $\mathscr{L}_1$-formulas.*

These characterizations show that the difference between finite automata and Petri nets lies in the ability to test if $X =_g Y$, which is exactly what is needed to recognize well-formed strings of parentheses. Moreover, if we consider weak Petri net languages instead of Petri net languages we add $\leqslant_g$ in place of $=_g$, which is exactly what is needed to recognize prefixes of well-formed strings of parentheses.

The following implications of Theorem 2.2 will be proved:

$$\begin{tikzcd}
\text{(i)} \arrow[d, shift left] \arrow[dr, shift left] & \text{(iii)} \arrow[d] & \text{(v)} \arrow[l] \\
\text{(ii)} \arrow[u, shift left] & \text{(iv)} \arrow[ul, shift left] \arrow[ur] &
\end{tikzcd}$$

2.3. *Equivalence of the logical characterizations*

In this section we prove the equivalences between (iii), (iv), and (v) of Theorem 2.2.

Proof of 2.2(iii)$\Rightarrow$(iv). Let $\psi = \exists \bar{X}\varphi(\bar{X})$ be a $\mathscr{L}$-sentence, where $\varphi(\bar{X})$ is a first-order formula. Call $\mathscr{L}_0$ the class of positive combinations of formulas of type $X \leqslant_g Y$, $\neg X \leqslant_g Y$, $X =_g Y$, $\neg X =_g Y$. Then, $\varphi(\bar{X})$ can be written as $Q_1x_1 \ldots Q_kx_k\vartheta(\bar{X}, \bar{x})$, where $\vartheta(\bar{X}, \bar{x})$ is a positive combination of $\mathscr{L}_0$-formulas and quantifier-free first-order $\mathscr{L}_1$-formulas.

We prove by induction on $k-i$ $(0 \leqslant k-i \leqslant k)$ that $\varphi(\bar{X})$ can be written as $Q_1x_1 \ldots Q_ix_i\vartheta(\bar{X}, x_1, \ldots, x_i)$, where $\vartheta(\bar{X}, x_1, \ldots, x_i)$ is a positive combination of $\mathscr{L}_0$-formulas and first-order $\mathscr{L}_1$-formulas.

Suppose we already have $\varphi(\bar{X})$ as $Q_1x_1 \ldots Q_{j+1}x_{j+1}\vartheta(\bar{X}, x_1, \ldots, x_{j+1})$, where $j < k$. If Q_{j+1} is $\forall$, then write ϑ as $\bigwedge_i (\zeta_i(\bar{X}) \vee \eta_i(\bar{X}, x_1, \ldots, x_{j+1}))$ and if Q_{j+1} is $\exists$, then we write ϑ as $\bigvee_i (\zeta_i(\bar{X}) \wedge \eta_i(\bar{X}, x_1, \ldots, x_{j+1}))$, where ζ_i is a $\mathscr{L}_0$-formula and each η_i a first-order $\mathscr{L}_1$-formula. By the distributivity of $\bigwedge$ and $\forall$ (respectively of $\bigvee$ and $\exists$) and since x_{j+1} does not occur in any ζ_i, we obtain $\varphi(\bar{X})$ as

$$Q_1x_1 \ldots Q_jx_j \bigwedge_i (\zeta_i(\bar{X}) \vee \forall x_{j+1}\eta(\bar{X}, x_1, \ldots, x_j, x_{j+1}))$$

(respectively as

$$Q_1x_1 \ldots Q_jx_j \bigvee_i (\zeta_i(\bar{X}) \wedge \exists x_{j+1}\eta(\bar{X}, x_1, \ldots, x_j, x_{j+1}))).$$

Finally, $\varphi(\bar{X})$ appears as positive combination of $\mathscr{L}_0$-formulas and of first-order $\mathscr{L}_1$-formulas. Now we replace all subformulas of φ of type $X \leqslant_g Y$, $\neg X \leqslant_g Y$, $\neg X =_g Y$ by their equivalent formulas in which only $=_g$ appears as second-order relation symbol:

$$X \leqslant_g Y \Leftrightarrow \exists Y'[\forall x(Y'x \rightarrow Yx) \wedge X =_g Y'],$$

$$\neg X \leqslant_g Y \Leftrightarrow \exists X'\exists Y'[X' =_g Y' \wedge \exists x_1 \forall x((X'x \leftrightarrow x < x_1 \wedge Xx) \wedge (Y'x \leftrightarrow x < x_1 \wedge Yx) \wedge Xx_1 \wedge \neg Yx_1)],$$

$$\neg X =_g Y \Leftrightarrow \exists X' \exists Y' [[X' =_g Y' \wedge \exists x_1 \forall x ((X'x \leftrightarrow x < x_1 \wedge Xx) \wedge (Y'x \leftrightarrow x < x_1 \wedge Yx) \wedge Xx_1 \wedge \neg Yx_1)] \vee [X =_g Y' \wedge \exists x_2 \forall x ((Y'x \leftrightarrow Yx \wedge x < x_2) \wedge Yx_2)]].$$

By renaming the corresponding set variables if necessary, the new second-order quantifiers can be moved to the left to obtain the formula ψ in the desired form $\exists \bar{X} \exists \bar{Y} \xi(\bar{X}, \bar{Y})$, where $\xi(\bar{X}, \bar{Y})$ is a positive combination of first-order $\mathscr{L}_1$-formulas and formulas of type $X =_g Y$. □

Proof of 2.2(iv) ⇒ (v). First note that $X =_g Y$ is equivalent to a second-order formula of Normal Form II:

$$\exists X' [\forall x (Xx \leftrightarrow Yx) \vee [X \leqslant_g Y \wedge \neg X' \leqslant_g Y \wedge \exists x' \forall x ((X'x \leftrightarrow Xx \vee x = x') \wedge Yx' \wedge \neg Xx' \wedge (Yx \wedge x > x' \rightarrow Xx))]].$$

If $\psi = \exists \bar{X} \varphi(\bar{X})$ is a $\mathscr{L}$-sentence of Normal Form I we can, after having replaced all occurrences of type $X =_g Y$ by the former formula, move the new existential second-order quantifiers to the left by renaming—if necessary—the corresponding set variables. Thus, we obtain ψ as $\exists \bar{X} \exists \bar{Y} \xi(\bar{X}, \bar{Y})$, where $\xi(\bar{X}, \bar{Y})$ is a positive combination of formulas of type $X \leqslant_g Y$, $\neg X \leqslant_g Y$ and first-order $\mathscr{L}_1$-formulas. □

The proof of 2.2(v) ⇒ (iii) is trivial.

3. The main arguments

3.1. *From Petri nets to logic (proofs of 2.2(i) ⇒ (iv) and 2.3(i) ⇒ (ii))*

As seen before, a word of length n is a structure over the ordered set $\{0, \ldots, n-1\}$. The numbers 0 to $n-1$ can be considered as successive moments of time (time 0, time 1, ...) where transitions are fired (exactly one at each moment); each predicate P_σ (for $\sigma \in \Sigma$) indicates the moment where transitions labelled σ are fired.

Let $R = (P, T, f, b, e, M_0, F)$ be a normal Petri net, where we write $P = \{p_1, \ldots, p_r\}$ and $T = \{t_1, \ldots, t_s\}$. The description of the behaviour of R by logical formulas makes use of the following set variables:

- for each $t \in T$, a variable X_t which indicates the moments where t is fired,
- for each $p \in P$, variables E_p and S_p indicating respectively the moments where one token is added to p or removed from p—using the convention that the firing of a transition at time i removes tokens at time i, but adds tokens at time $i+1$,
- for each $p \in P$, a variable E'_p which takes the last token added to p and the final markings of p into account.

We call $\bar{Y}$ the sequence $X_{t_1}, \ldots, X_{t_s}, E_{p_1}, \ldots, E_{p_r}, S_{p_1}, \ldots, S_{p_r}, E'_{p_1}, \ldots, E'_{p_r}$.

Consider the following $\mathscr{L}$-formulas:

$$\varphi_1(\bar{Y}) = \forall x\left[\bigvee_{t\in T}\left(X_t x \wedge P_{e(t)}x \wedge \bigwedge_{t'\in T, t\neq t'} \neg X_{t'}x\right)\right],$$

$$\varphi_2(\bar{Y}) = \forall x\left[\bigwedge_{p\in P}\left(S_p x \leftrightarrow \bigvee_{t\in T,(p,t)\in A} X_t x\right)\right],$$

$$\varphi_3(\bar{Y}) = \exists y\left[\forall x\, x\geqslant y \wedge \left[\forall u\, \forall v\, (y\leqslant u \wedge u<v \wedge \forall z \neg(u<z \wedge z<v)) \rightarrow \bigwedge_{p\in P}\left(E_p v \leftrightarrow \bigvee_{t\in T,(t,p)\in A} X_t u\right)\right] \wedge \left(\bigwedge_{p\in P, M_0(p)=1} E_p y\right) \wedge \left(\bigwedge_{p\in P, M_0(p)=0} \neg E_p y\right)\right],$$

$$\varphi_4(\bar{Y}) = \exists y\left[\forall x\left[(x\leqslant y) \wedge \bigvee_{M_f\in F}\left[\forall x\left(\bigvee_{p\in P, M_f(p)=1}\left[\bigvee_{t\in T,(t,p)\in A} (X_t y \wedge (E'_p x \leftrightarrow E_p x)) \vee \bigvee_{t\in T,(t,p)\notin A} (X_t y \wedge \exists z((E_p x \rightarrow x\leqslant z) \wedge E_p z \wedge (E'_p x \leftrightarrow E_p x \wedge x\neq z)))\right] \wedge \bigwedge_{p\in P, M_f(p)=0} (E'_p x \leftrightarrow E_p x)\right)\right]\right]\right],$$

$$\varphi_5(\bar{Y}) = \bigwedge_{p\in P} S_p \leqslant_g E'_p,$$

$$\varphi'_5(\bar{Y}) = \bigwedge_{p\in P} S_p =_g E'_p.$$

Meaning of these formulas

Formula φ_1 says that $\{X_t\}_{t\in T}$ is a partition of the domain and, for each $t\in T$, $P_{e(t)}$ contains X_t, thus, to each word w a sequence $\bar{t} = t_0, \ldots, t_{n-1}$ whose label is w is associated.

Formulas φ_2 and φ_3 say that, for all $p\in P$,

$$0\in E_p \quad \text{iff} \quad M_0(p)=1,$$

$$i+1\in E_p \quad \text{iff} \quad (t_i, p)\in A \ \text{for}\ 0\leqslant i\leqslant n-1,$$

$$i\in S_p \quad \text{iff} \quad (p, t_i)\in A \ \text{for}\ 0\leqslant i\leqslant n-1.$$

At each moment i $(1\leqslant i\leqslant n-1)$ we have to insure that transition t_i can be fired, i.e., that, for all $p\in P$, $|E_p\cap i+1|\geqslant|S_p\cap i+1|$. This requirement can be expressed by $S_p\leqslant_g E_p$. But we also have to insure that after firing of t_{n-1} we obtain (respectively surpass) a final marking; thus, instead of E_p we consider E'_p which depends of $M_f(p)$ and of the existence or absence of the arc (t_{n-1}, p) not considered in the definition of E_p.

Formula φ_4 says that E'_p is equal to E_p less its last element if and only if $M_f(p)=1$ and $(t_{n-1}, p)\notin A$; otherwise, E'_p and E_p are equal.

Formula φ_5 gives the condition to surpass a final marking, $S_p \leqslant_g E'_p$.

Formula φ'_5 gives the condition to obtain a final marking, $S_p \leqslant_g E'_p$ and $|S_p| = |E'_p|$.

It follows from this choice of formulas that

$$\psi_L = \exists \bar{Y}\left[\left(\bigwedge_{1\leqslant i\leqslant 4} \varphi_i(\bar{Y})\right) \wedge \varphi'_5(\bar{Y})\right] \text{ defines } L(R)$$

and that

$$\psi_G = \exists \bar{Y}\left[\bigwedge_{1\leqslant i\leqslant 5} \varphi_i(\bar{Y})\right] \text{ defines } G(R).$$

Since we have seen in Section 1 that for each Petri net there is a normal Petri net with the same language and the same weak language, the implications 2.2(i) $\Rightarrow$ (iv) and 2.3(i) $\Rightarrow$ (ii) are proved.

3.2. From logic to Petri nets (proofs of 2.2(iv)$\Rightarrow$(i) and 2.3(ii)$\Rightarrow$(i))

In order to prove the results by induction we need to work on formulas and not only on sentences. Thus, we have to explain how to associate a language to a formula.

Let $\varphi = \varphi(X_1, \ldots, X_k, x_1, \ldots, x_l)$ be a $\mathscr{L}$-formula; all free variables of φ are among $X_1, \ldots, X_k$, $x_1, \ldots, x_l$. A *model of* φ of length n is a structure $(w, A_1, \ldots, A_k, a_1, \ldots, a_l)$, where w is a word on Σ of length n, A_i a subset of n for $1 \leqslant i \leqslant k$, a_j an element of n for $1 \leqslant j \leqslant l$, such that $\varphi(A_1, \ldots, A_k, a_1, \ldots, a_l)$ is true in w.

Such a structure can be seen as a word of length n on $\Sigma \times \{0, 1\}^{k+l}$, or as a matrix with n columns and $k+l+1$ lines, where line 0 denotes w, line i $(1 \leqslant i \leqslant k)$ denotes the characteristic function of A_i and line $k+j$ $(1 \leqslant j \leqslant l)$ denotes the characteristic function of $\{a_j\}$.

Thus, a formula $\varphi(X_1, \ldots, X_k, x_1, \ldots, x_l)$ defines a language over $\Sigma \times \{0, 1\}^{k+l}$: the *set of models of* φ of finite length, which we call Mod $\varphi(X_1, \ldots, X_k, x_1, \ldots, x_l)$. (Note that if φ has no free variables, then this definition fits together with the definition of Section 2.1.)

Our notion of language defined by a formula $\varphi(\bar{X}, \bar{x})$ depends on the choice of the sequence $\bar{X}, \bar{x}$ among which the free variables of φ are taken. But, the membership to the class of Petri net languages does not depend on this choice. In particular, if Mod $\varphi(\bar{X}, \bar{x}) = L(R)$ for some Petri net R, and Y (respectively y) is a new variable, then there is a Petri net R' (respectively R'') such that Mod $\varphi(\bar{X}, Y, \bar{x}) = L(R')$ (respectively Mod $\varphi(\bar{X}, \bar{x}, y) = L(R'')$).

To obtain the Petri net R' from R, we just have to replace each transition t of T by two copies t_0, t_1 (i.e., the arcs between t_0 (respectively t_1) and $p \in P$ are the same as those between t and these places), respectively labeled $\binom{e(t)}{0}$, $\binom{e(t)}{1}$.

To obtain R'', first we can construct R' and then add one place q, initially marked by one and finally marked by zero and arcs $q \to t_1$, for all $t \in T$.

Remark. The same property holds for weak Petri net languages.

Now we have to prove that each $\mathscr{L}$-formula of type $\exists \bar{X}\, \varphi(\bar{X}, \bar{Y}, \bar{y})$, where $\varphi(\bar{X}, \bar{Y}, \bar{y})$ is a positive combination of formulas of type $X =_g Y$ (respectively $X \leq_g Y$) and first-order formulas of $\mathscr{L}_1$, defines a language $L(R)$ (respectively $G(R)$) for some Petri net R.

First, we associate to a first-order formula $\varphi(\bar{X}, \bar{x})$ of $\mathscr{L}_1$ a Petri net R—which is in fact a finite state machine—such that Mod $\varphi = L(R) = G(R)$. The proof is obtained by induction on the construction of formulas, as follows.

For the atomic formulas

$$\varphi_1(x, y) := x \leq y; \qquad \varphi_2(Y, x) := Yx; \qquad \varphi_3(x) := P_\tau x$$

the associated Petri nets R_1, R_2, R_3 are represented in Figs. 1, 2, and 3, respectively. We write next to each transition t its label with the following convention:

(i) a transition t with more than one label means that for each label there is a copy of t with this label,

(ii) a label containing σ means that for each $\sigma \in \Sigma$ there is such a label.

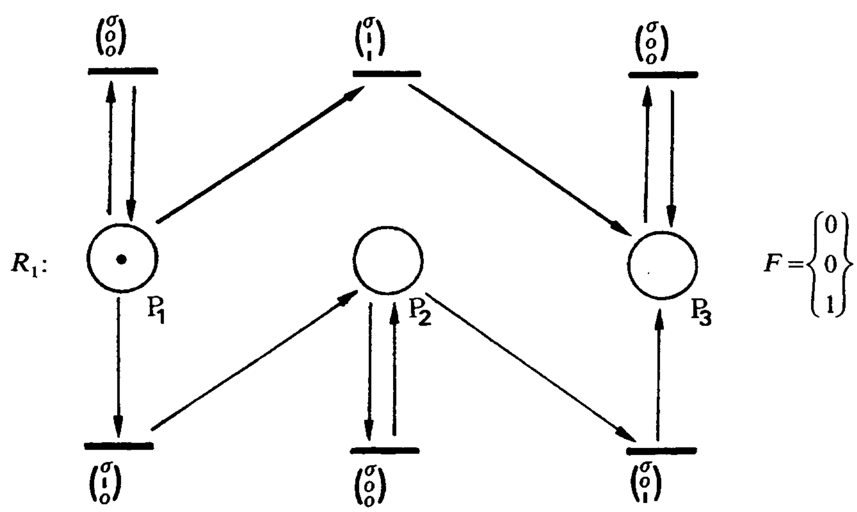


Fig. 1.

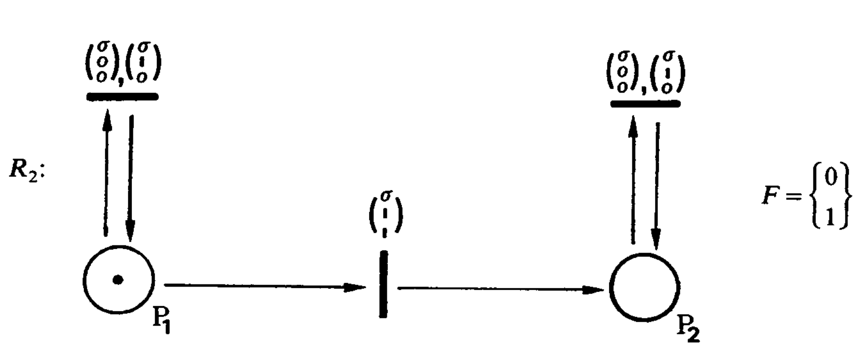
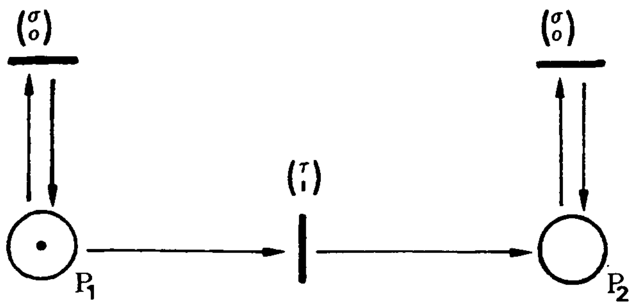


Fig. 2.

R_3:



$F = \left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right\}$

Fig. 3.

The initial marking is represented by tokens (dots) in the places, the final marking F is given by the set of vectors $M_f = (m_1, \ldots, m_r)^T$, indicating that, for all $i \leq r$, $M_f(p_i) = m_i$.

Since

$$\mathrm{Mod}(\varphi_1 \wedge \varphi_2)(\bar{X}, \bar{x}) = \mathrm{Mod}\ \varphi_1(\bar{X}, \bar{x}) \cap \mathrm{Mod}\ \varphi_2(\bar{X}, \bar{x}),$$

$$\mathrm{Mod}(\varphi_1 \vee \varphi_2)(\bar{X}, \bar{x}) = \mathrm{Mod}\ \varphi_1(\bar{X}, \bar{x}) \cup \mathrm{Mod}\ \varphi_2(\bar{X}, \bar{x}),$$

$$\mathrm{Mod}(\neg\varphi_1)(\bar{X}, \bar{x}) = \mathrm{C}\ \mathrm{Mod}\ \varphi_1(\bar{X}, \bar{x})$$

and since regular languages are closed by complement, union, and intersection, there are finite automata and also Petri nets which recognize these languages.

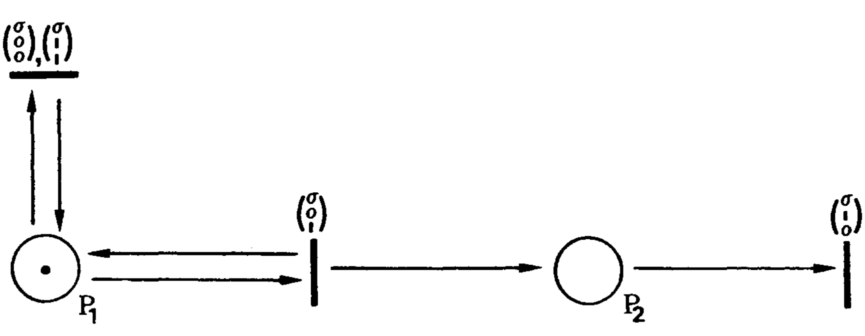
In order to obtain an automaton or a Petri net which recognize Mod $\exists y\, \varphi(\bar{X}, \bar{x}, y)$, we just take the automaton or the Petri net whose language is Mod $\varphi(\bar{X}, \bar{x}, y)$ and forget the line which corresponds to y in the labelling of arcs (respectively transitions).

Now we associate Petri nets to the atomic formulas (see Figs. 4 and 5, respectively)

$$\varphi_4(X, Y) := X =_g Y \quad \text{and} \quad \varphi_5(X, Y) := X \leq_g Y.$$

Note that the language and the weak language of R_4 are not the same: $L(R_4) =$ Mod $\varphi_4(X, Y)$, $G(R_4) =$ Mod $\varphi_5(X, Y)$; for R_5 these languages coincide: $L(R_5) = G(R_5) =$ Mod $\varphi_5(X, Y)$.

R_4:



$F = \left\{ \begin{matrix} 1 \\ 0 \end{matrix} \right\}$

Fig. 4.

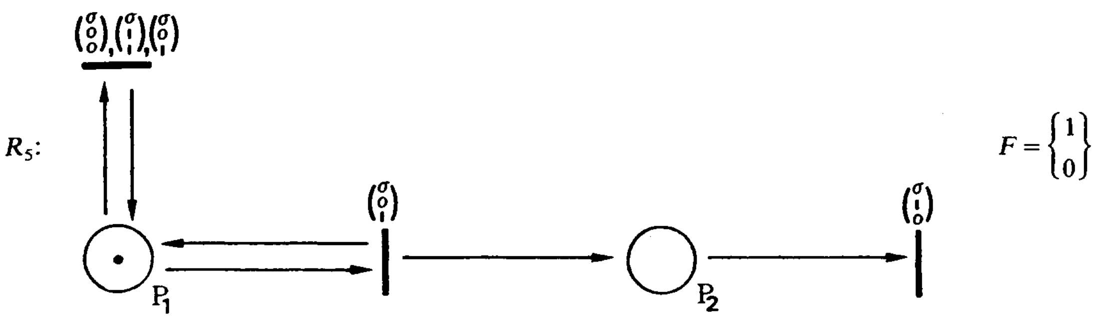


Fig. 5.

Since Petri net languages (respectively weak languages) are closed by union and intersection, each positive combination of formulas of type $X =_g Y$ (respectively $X \leqslant_g Y$) and first-order formulas of $\mathscr{L}_1$ defines a Petri net language (respectively weak language).

These classes of languages are also closed by existential second-order quantification: the Petri net recognizing Mod $\exists Y \varphi(\bar{X}, Y, \bar{x})$ is just the Petri net whose language is Mod $\varphi(\bar{X}, Y, \bar{x})$ and where the line designating Y in the labelling of transitions is forgotten.

3.3. Petri net languages and deadlock languages (proof of 2.2(i) $\Leftrightarrow$ (ii))

It is a well-known result [4] that every Petri net deadlock language is also a Petri net language. (This result can also be obtained by expressing such a language in the logical formalism, like the proof of 2.2(i) $\Rightarrow$ (iv).) Thus, we just have to prove that Petri net languages are also Petri net deadlock languages.

Let R be a Petri net. We first modify R into a Petri net R' whose final marking is the zero marking (i.e., zero token at each place). In order to do this we add

(i) a place q and, for all $t \in T$, the arcs $q \to t$ and $t \to q$,

(ii) for all $t \in T$ and $M \in \mathbb{N}^r$ such that $M(t\rangle M_f$ for some $M_f \in F$, a transition t' with $e(t') = e(t)$ and the arcs $q \to t'$, $p \to^{M(p)} t'$, for all p, such that $M(p) \neq 0$.

Further, we set $M_0'(r) = 1$, $M_0'(p) = M_0(p)$, for all $p \in P$. Hence, for all $w \in \Sigma^+$, $w \in L(R)$ iff $w \in L(R')$.

Now we have to modify R' into a Petri net R'' which avoids all deadlock situations other than in the zero marking. We attain this aim by adding for each place $p \in P'$ a place p' and two new transitions t_p and t_p', arbitrary labelled in Σ, and the arcs $p \to t_p$, $t_p \to p'$, $p' \to t_p'$, and $t_p' \to p'$. We set $M_0''(p) = M_0'(p)$ and $M_0''(p') = 0$, for all $p \in P'$.

This new net R'' has the following properties:

(i) for every nonzero marking on the places of P', a new transition can fire,

(ii) if a new transition has fired, then R'' never deadlocks.

Therefore, for all sequences $\bar{t}$ of T'' the following holds: R'' deadlocks after having fired $\bar{t}$ if and only if $\bar{t}$ is a sequence of T' and R' deadlocks in the zero marking after having fired $\bar{t}$.

4. Concluding remarks

4.1. *The use of the logical formalism*

In order to prove that a language is a Petri net language (or a weak Petri net language) it is sufficient to find a logical formula which defines these languages. This is much easier than explicitly giving the Petri net associated with the language.

For instance, $L_1 = \{a^n b^n \mid n \geq 1\}$ is defined by

$$\varphi_1 := P_b =_g P_a \wedge \forall x\, \forall y\, ((P_a x \wedge P_b y) \rightarrow x < y),$$

$L_2 = \{a^n b^{2n} c^n \mid n \geq 1\}$ is defined by

$$\varphi_2 := \exists X\, \exists Y\, [P_c =_g Y \wedge Y =_g X \wedge X =_g P_a \wedge \forall x\, (Xx \vee Yx) \leftrightarrow P_b x)$$
$$\wedge \forall u\, \forall x\, \forall y\, \forall z\, ((P_a u \wedge Xx \wedge Yy \wedge P_c z)$$
$$\rightarrow (u < x \wedge x < y \wedge y < z))],$$

$L_3 = \{w \in \{a, b\}^+ \mid |w|_a = |w|_b\}$ is defined by

$$\varphi_3 := \exists Z\, [\forall x\, \forall y\, ((Zx \wedge y < x) \rightarrow Zy) \wedge P_a =_g Z \wedge P_b =_g Z],$$

$L_4 = \{w \in \Sigma^+ \mid w \neq w^R\}$ is defined by

$$\varphi_4 := \exists X\, \exists Y\, \exists x\, \exists y [\forall z\, ((Xz \leftrightarrow z \leqslant x) \wedge (Yz \leftrightarrow z \geqslant y))$$
$$\wedge Y =_g X \wedge \bigvee_{\sigma \in \Sigma} (P_\sigma x \wedge \neg P_\sigma y)].$$

Since palindroms are not Petri net languages, the last mentioned example gives the logical proof that Petri net languages are not closed under complementation and shows that we cannot hope to obtain a complete analogy to Büchi's theorem.

4.2. *The difference between Petri languages and weak Petri net languages*

In order to justify the difference between logical formalisms for Petri net languages and weak Petri net languages, we have to prove that the two notions do not coincide (i.e., $L \subsetneq G$), and, in particular, that the language defined by $X =_g Y$ is not a weak Petri net language. This follows from the next proposition.

Proposition 4.1. *$L_1 = \{a^n b^n \mid n \geq 1\}$ is not a weak Petri net language.*

The proof uses the following combinatoric fact on vectors (known as the Lemma of Dickson), which can be proved by applying the Ramsey Theorem.

Fact. *Let $E \subsetneq \mathbb{N}^k$, $k \geqslant 1$, be an infinite set of vectors of length k and $B \subsetneq E$ the set of minimal vectors of E for the ordering $\leqslant$ defined by: $e \leqslant f$ iff, for all $i \leqslant k$, $e(i) \leqslant f(i)$. Then B is finite.*

Proof of Proposition 4.1. Assume that R is a Petri net whose weak language is L_1. For all $i \geq 1$, let $\bar{t}_i$, $\bar{s}_i$ be a firing sequence with $e(\bar{t}_i) = a^i$, $e(\bar{s}_i) = b^i$, for which there exist markings M_i, M'_i and a final marking $M_{f_i} \in F$ such that $M_0(\bar{t}_i > M_i(\bar{s}_i > M'_i$ and $M'_i \geq M_{f_i}$.

The set of minimal vectors of $\{M_i \mid i \geq 1\}$ being finite, there are distinct numbers m and n such that $M_n \leq M_m$. Since $M_n(\bar{s}_n > M'_n$, we have $M_m(\bar{s}_n > M'$ for some $M \geq M'_n$, and therefore $a^m b^n \in L_1$, contrary to the definition of this language. □

Proposition 4.2. *The language defined by* $\neg X \leq_g Y$ *is not a weak Petri net language.*

Corollary 4.3. *The language defined by* $X =_g Y$ *is not a weak Petri net language.*

Proof. Assume that $\neg X \leq_g Y$ defines a weak Petri net language. By the closure properties of weak Petri net languages, the formula $X =_g Y$, which is equivalent to a formula involving only subformulas with second-order relation symbols of type $U \leq_g V$ and $\neg U \leq_g V$ (see the proof of 2.2(iv)$\Rightarrow$(v)), defines a weak Petri net language, too.

It follows that the classes of Petri net languages and weak Petri net languages are the same, which is a contradiction to Proposition 4.1. □

References

[1] G.W. Brams, *Réseaux de Petri, Théorie et Pratique* (Masson, Paris, 1983).

[2] J.R. Büchi, Weak second-order arithmetic and finite automata, *Z. Math. Logik und Grundlagen der Math.* **6** (1960) 66-92.

[3] M. Kleene, Representation of events in nerve nets and finite automata, in: Shannon and McCarthy, eds., *Automata Studies* (Princeton University Press, 1956).

[4] J.L. Peterson, *Petri Net Theory and the Modeling of Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1981).

[5] J. Sifakis, Le contrôle des systèmes asynchrones: Concepts, propriétés, analyse statique, Thèse USM Grenoble, 1979.

[6] R. Valk and G. Vidal-Naquet, Petri nets and regular languages, *J. Comput. System Sci.* **23** (3) (1981) 299-325.