Vietnam Journal of Mathematics **28**:3 (2000) 209-216

Vietnam Journal
of
MATHEMATICS

On an Infinite Hierarchy of Petri Net Languages

Pham Tra An and Pham Van Thao
Institute of Mathematics, P. O. Box 631, Bo Ho, Hanoi, Vietnam

Received May 11, 1999
Revised February 22, 2000

Abstract. In this paper we show the existence of an infinite hierarchy of Petri net languages on the number of transitions and places of their recognizing nets.

1. Preliminaries

As is well known, the Petri net is a mathematical model of parallel and distributed computing systems. In the last ten years, the theory of Petri nets and its applications have been investigated extensively by many authors (see, for example, [8–11]).

Let $\mathcal{N}$ be a Petri net with m transitions and n places, and $k = \min\{m, n\}$. For any integer $n \geq 1$ we denote by $\mathcal{L}(n)$ the class of all Petri net languages acceptable by a Petri net $\mathcal{N}$ with $k \leq n$.

Our aim in this paper is to prove that there exists an increasing infinite sequence of integers n_i,

$$1 \leq n_1 < n_2 < \cdots < n_i < n_{i+1} < \cdots,$$

such that

$$\mathcal{L}(n_1) \subset \mathcal{L}(n_2) \subset \cdots \subset \mathcal{L}(n_i) \subset \mathcal{L}(n_{i+1}) \subset \cdots$$

The proof of the result is based on a complexity characteristic of Petri net languages, obtained earlier by the first author [6].

Analogous hierarchies for some other classes of languages were earlier considered by several authors, for instance, by Cole for languages recognizable by iterative arrays of finite automata [1], by P.D. Dieu and P.T. An for languages recognizable by probabilistic automata and those with time-variant-structure [3, 4].

Definitions of Petri nets and Petri net languages are recalled in this section. In Sec. 2 a complexity characteristic of languages is considered. Using this characteristic a necessary condition for the Petri net languages is given. However, as it will be shown, this condition is not sufficient. In Sec. 3, we show the existence of an infinite hierarchy of Petri net languages on the number of transitions and places of their recognizing nets. For any finite alphabet Σ, we denote by Σ^*, (resp. Σ^r, $\Sigma^{\leq r}$) the set of all words (resp. of all the words of length r, of all the words of length at most r) on the alphabet Σ, Λ denotes the empty word. For any word $\omega \in \Sigma^*$, $l(\omega)$ denotes the length of ω. Every subset $L \subseteq \Sigma^*$ is called a language over the alphabet Σ. Let N be the set of all non-negative integers and $N^+ = N\backslash\{0\}$.

A *(labeled) Petri net* $\mathcal{N}$ is given by a list:

$$\mathcal{N} = (P, T, I, O, \sigma, \mu_0, M_f),$$

where
$P = \{p_1, ..., p_n\}$ is a finite set of *places*;
$T = \{t_1, ..., t_m\}$ is a finite set of *transitions*, $P \cap T = \emptyset$;
$I : P \times T \to N$ is the *input function*;
$O: T \times P \to N$ is the *output function*;
$\sigma : T \to \Sigma$ is the *labeling function*, where Σ is a finite *output alphabet*;
$\mu_0 : P \to N$ is the *initial marking*;
$M_f = \{\mu_{f_1}, ..., \mu_{f_k}\}$ is a finite set of *final markings.*

We can extend the labeling function σ for the words in T^* as follows:

$$\text{if } t = t_1 t_2 \dots t_n, \text{ then } \sigma(t) = \sigma(t_1)\sigma(t_2)\dots\sigma(t_n).$$

A marking μ (global configuration) of the Petri net $\mathcal{N}$ is a function $\mu : P \to N$ from the set of places P into N. The marking μ can also be represented as an n-vector $\mu = (\mu_1, ..., \mu_n)$ where $\mu_i = \mu(p_i)$ and $n = |P|$. A transition t of $\mathcal{N}$ is said to be *firable at the marking* μ if

$$\forall p \in P : \mu(p) \geq I(p,t).$$

If t is firable at μ, then when t fires, the Petri net $\mathcal{N}$ will go into a new marking μ' given by

$$\forall p \in P : \mu'(p) = \mu(p) - I(p,t) + O(t,p).$$

We write then $\delta(\mu, t) = \mu'$ and call δ *the state changing function of the net* $\mathcal{N}$.

A *firing sequence of* $\mathcal{N}$ can be defined as a sequence of transitions such that the firing of each of its prefixes will lead $\mathcal{N}$ into a marking at which the next transition is firable. The set of all firing sequences of $\mathcal{N}$ is denoted by $\mathcal{F}_\mathcal{N}$.

The function δ can be extended for firing sequences by induction as follows:

$$\begin{cases} \delta(\mu, \Lambda) = \mu, \\ \delta(\mu, tt_j) = \delta(\delta(\mu, t), t_j), \end{cases}$$

where $t \in T^*$, $t_j \in T$, and μ is a marking at which tt_j is a firing sequence.

We call *language acceptable by a (labeled) Petri net* $\mathcal{N}$ the set:

$$L(\mathcal{N}) = \{x \in \Sigma^* \mid \exists t \in T^* : (x = \sigma(t)) \wedge (t \in \mathcal{F}_\mathcal{N}) \wedge (\delta(\mu_0, t) \in M_f)\},$$

The set of all (labeled) Petri net languages is denoted by $\mathcal{L}$.

2. On a Criterion for Petri Net Languages

In this section we recall a necessary condition for Petri net languages introduced in [7] (Theorem 2.1) and show that the condition is not sufficient (Theorem 2.3). This condition is based on a complexity characteristic for languages defined as follows:

With every language $L \subseteq \Sigma^*$ we associate an equivalence relation on $\Sigma^{\leq r}$, denoted by $E_{\leq r}(\mathrm{mod}L)$, and an equivalence relation on Σ^r, denoted by $E_r(\mathrm{mod}L)$, which are defined, respectively, as follows:

$$\forall x_1, x_2 \in \Sigma^{\leq r}, x_1 E_{\leq r} x_2(\mathrm{mod}L) \Leftrightarrow \forall \omega \in \Sigma^* : x_1\omega \in L \leftrightarrow x_2\omega \in L;$$

$$\forall x_1, x_2 \in \Sigma^r, x_1 E_r x_2(\mathrm{mod}L) \Leftrightarrow \forall \omega \in \Sigma^* : x_1\omega \in L \leftrightarrow x_2\omega \in L.$$

Then we define

$$G_L(r) = \mathrm{Rank}E_{\leq r}(\mathrm{mod}L),$$

$$H_L(r) = \mathrm{Rank}E_r(\mathrm{mod}L).$$

They are used as complexity characteristics of the language L. It is easy to see that, for any $r \in N$,

$$1 \leq H_L(r) \leq G_L(r) \leq \mathrm{Exp}(r),$$

where $\mathrm{Exp}(r)$ denotes some exponential function of r.

Let us take some examples.

Example 1. Let $\Sigma = \{a, b\}$ and $L_1 = \{a^m b^n | m, n \in N^+\}$. Consider the subsets

$$W_1 = \{a^m | 1 \leq m \leq r\};$$

$$W_2 = \{a^m b^k | m + k \leq r; \ k \geq 1\};$$

$$W_3 = \{\omega \in \Sigma^{\leq r} | \omega \notin W_1 \cup W_2\}.$$

Obviously, $\Sigma^{\leq r} = W_1 \cup W_2 \cup W_3$ and $W_i \cap W_j = \emptyset$, $i \neq j$.

It is easy to prove that any two words in every W_i, $i = 1, 2, 3$, are equivalent. However, any two words in different sets W_i are not equivalent by the relation $E_{\leq r}(\mathrm{mod}L_1)$. Therefore, $G_{L_1}(r) = 3$.

Example 2. Let $|\Sigma| = k \geq 2$ and $L_2 = \{xx^R | x \in \Sigma^*\}$, where x^R is the inverse image of x.

It is easy to show that if $x_1, x_2 \in \Sigma^r$, $x_1 \neq x_2$, then $x_1 \overline{E}_r x_2(\mathrm{mod}L_2)$, thereby $H_{L_2}(r) = |\Sigma^r| = k^r$.

Example 3. Let $|\Sigma| = k \geq 2$, $c \notin \Sigma$ and $L_3 = \{xcx \mid x \in \Sigma^*\}$.

It can be verified that if $x_1, x_2 \in \Sigma^{\leq r}$, $x_1 \neq x_2$, then $x_1 \overline{E}_{\leq r} x_2(\mathrm{mod}L_3)$. Therefore, $G_{L_3}(r) = |\Sigma^{\leq r}| = k(k^r - 1)/(k - 1)$.

The following result has been established in [7].

Theorem 2.1. *Let L be accepted by a Petri net with m transitions and n places and $k = \min\{m, n\}$. There exists a polynomial P_k of degree k such that, for any integer $r \geq 1$,*

$$H_L(r) \leq P_k(r),$$
$$G_L(r) \leq P_k(r).$$

Using Theorem 2.1, we can show a series of rather simple languages not being acceptable by any Petri net.

Example 4. Let $|\Sigma| = k \geq 2$ and $c \notin \Sigma$. Consider the languages $L_2 = \{xx^R \mid x \in \Sigma^*\}$, $L_3 = \{xcx \mid x \in \Sigma^*\}$, where x^R is the inverse image of x.

We have proved in Examples 2 and 3 that $H_{L_2}(r) = k^r$ and $G_{L_3}(r) = k(k^r - 1)/(k-1)$. By Theorem 2.1 we have $L_2 \notin \mathcal{L}$ and $L_3 \notin \mathcal{L}$.

Now we shall show that the necessary condition in Theorem 2.1 is not sufficient. For this we need some notions in the theory of codes (see [14]) .

A language $L \subseteq \Sigma^*$ is *a code over* Σ if for all $n, m \geq 1$ and $x_1, \ldots x_n, x'_1, \ldots, x'_m \in L$, the equality

$$x_1 x_2 \ldots x_n = x'_1 x'_2 \ldots x'_m$$

implies $n = m$ and $x_i = x'_i$ for $i = 1, \ldots, n$.

In other words, a set L is a code if any word in L^* can be written uniquely as a product of words in L, that is, it has a unique factorization on words of L.

A subset L of Σ^* is a *prefix set* if no word in L is a proper left of another word in L. Evidently every prefix set L with $L \neq \{\Lambda\}$ is a code called a prefix code.

It is not difficult to check that if L is a prefix code, then every word $x \in \Sigma^+$ can be written uniquely in the form of $x = \tilde{x}x_0$, where $\tilde{x} \in L^*$ and x_0 has no left factor in L.

Using the above fact we obtain

Lemma 2.2. *If L is a prefix code, then, for any $r \in N^+$,*

$$G_{L^+}(r) \leq G_L(r),$$

where $L^+ = L^* \backslash \{\Lambda\}$.

Proof. It suffices to prove that $\forall x, y \in \Sigma^{\leq r}$, $\exists x_0, y_0 \in \Sigma^{\leq r}$ such that

$$x_0 E_{\leq r} y_0 (\text{mod} L) \longrightarrow x E_{\leq r} y\, (\text{mod} L^+),$$

which implies

$$\text{Rank}\, E_{\leq r}(\text{mod} L^+) \leq \text{Rank}\, E_{\leq r}(\text{mod} L),$$

i.e., $G_{L^+}(r) \leq G_L(r)$.

Since L is a prefix code, x, y can be written uniquely as $x = \tilde{x}x_0$, $y = \tilde{y}y_0$, where $\tilde{x}, \tilde{y} \in L^*$; $x_0, y_0 \in \Sigma^{\leq r}$ and x_0, y_0 have no left factors in L.

If $x_0 E_{\leq r} y_0(\mathrm{mod} L)$, then $\forall \omega \in \Sigma^*, x_0\omega \in L \longleftrightarrow y_0\omega \in L$. Two cases are possible:

Case 1. $x_0\omega \in L$ and $y_0\omega \in L$.

From $x_0\omega \in L$ and $y_0\omega \in L$, it follows that $\tilde{x}x_0\omega \in L^+$ and $\tilde{y}y_0\omega \in L^+$, i.e. $x\omega \in L^+$ and $y\omega \in L^+$, therefore, $xE_{\leq r}y\,(\mathrm{mod}L^+)$.

Case 2. $x_0\omega \notin L$ and $y_0\omega \notin L$.

If $x_0\omega \in L^+$, $y_0\omega \in L^+$, then $\tilde{x}x_0\omega \in L^+$ and $\tilde{y}y_0\omega \in L^+$, i.e., $x\omega \in L^+$ and $y\omega \in L^+$.

If $x_0\omega \notin L^+, y_0\omega \notin L^+$, then $\tilde{x}x_0\omega \notin L^+$ and $\tilde{y}y_0\omega \notin L^+$, i.e., $x\omega \notin L^+$ and $y\omega \notin L^+$.

Let $x_0\omega \in L^+$, $y_0\omega \notin L^+$. We have $x_0\omega = (x_0\omega_0)\tilde{\omega} \in L^+$, where $x_0\omega_0 \in L$, $\tilde{\omega} \in L^+$. On the other hand, $y_0\omega = (y_0\omega_0)\tilde{\omega} \notin L^+$, L is prefix and $\tilde{\omega} \in L^+$; it follows that $y_0\omega_0 \notin L$, i.e., there exists ω_0 such that $x_0\omega_0 \in L$ and $y_0\omega_0 \notin L$. This contradicts the hypothesis $x_0E_{\leq r}y_0(\mathrm{mod}L)$.

Thus, in both cases, we have proved that $xE_{\leq r}y\,(\mathrm{mod}L^+)$. This completes the proof. ∎

Now we can establish the main result of this section.

Theorem 2.3. *There exists a language L with $G_L(r) \leq P_5(r)$, which cannot be accepted by any Petri net. In other words the necessary condition in Theorem 2.1 is not sufficient.*

Proof. We consider the language

$$L' = \{a^n b^n \mid n > 1\}.$$

This language L' is easily verified to be accepted by the Petri net $\mathcal{N}$, depicted in Fig. 1, with $k = \min\{|T|, |P|\} = 5$. By Theorm 2.1 it follows that $G_{L'}(r) \leq P_5(r)$.

On the other hand, L' is obviously a prefix code.

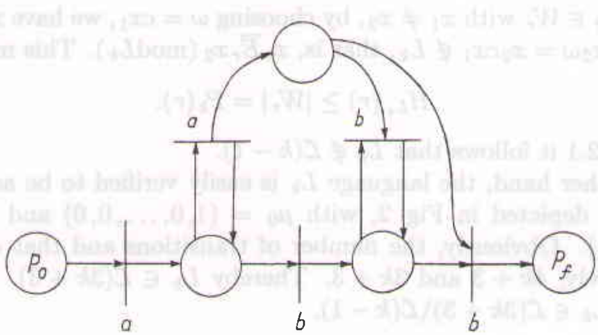

Fig. 1.

Put $L = (L')^+$. By Lemma 2.2 we have

$$G_L(r) = G_{(L')^+}(r) \leq G_{L'}(r) \leq P_5(r).$$

As shown in [13] by Peterson the language $L = (L')^+$ is not a Petri net language. The theorem is proved. ■

3. An Infinite Hierarchy of Petri Net Languages

Based on Theorem 2.1, we can obtain the solution of problem on the infinite hierarchy of Petri net languages.

Theorem 3.1. *There exists an increasing infinite sequence of integers n_i,*

$$1 \leq n_1 < n_2 < \cdots < n_i < n_{i+1} < \cdots,$$

with $n_{i+1} = 3n_i + 6$, such that

$$\mathcal{L}(n_1) \subset \mathcal{L}(n_2) \subset \cdots \subset \mathcal{L}(n_i) \subset \mathcal{L}(n_{i+1}) \subset \cdots$$

Proof. Let $\Sigma = \{0, 1\}$, $c \notin \Sigma$, $k \geq 2$. Consider the language

$$L_k = \{xcx \mid x \in \Sigma^*, \, |x|_1 = k\},$$

where $|x|_1$ denotes the number of occurences of 1 in x.

We now prove the two following propositions:

(i) For any $r \geq k$: $H_{L_k}(r) \geq P_k(r)$; therefore, $L_k \notin \mathcal{L}(k-1)$.

(ii) $L_k = L(\mathcal{N})$, where $\mathcal{N}$ is a Petri net with $\min\{|T|, |P|\} = 3k+3$; therefore, $L_k \in \mathcal{L}(3k+3)$.

Put

$$W_r = \{x \mid x \in \Sigma^*; \, l(x) = r; \, |x|_1 = k\},$$

where $l(x)$ denotes the length of x. It is easy to verify that

$$|W_r| = C_r^k = r!/k!(r-k)! = r(r-1)\dots(r-k+1)/k! = P_k(r).$$

For any $x_1, x_2 \in W_r$ with $x_1 \neq x_2$, by choosing $\omega = cx_1$, we have $x_1\omega = x_1cx_1 \in L_k$, whereas $x_2\omega = x_2cx_1 \notin L_k$, that is, $x_1 \overline{E}_r x_2 \,(\mathrm{mod} L_k)$. This means that

$$H_{L_k}(r) \geq |W_r| = P_k(r).$$

By Theorem 2.1 it follows that $L_k \notin \mathcal{L}(k-1)$.

On the other hand, the language L_k is easily verified to be accepted by the Petri net $\mathcal{N}$, depicted in Fig. 2, with $\mu_0 = (1, 0, \dots, 0, 0)$ and $M_f = \{\mu_f = (0, 0, \dots, 0, 1)\}$. Obviously, the number of transitions and that of places of $\mathcal{N}$ are, respectively, $4k+3$ and $3k+3$. Thereby $L_k \in \mathcal{L}(3k+3)$. Thus we have proved that $L_k \in \mathcal{L}(3k+3) \backslash \mathcal{L}(k-1)$.

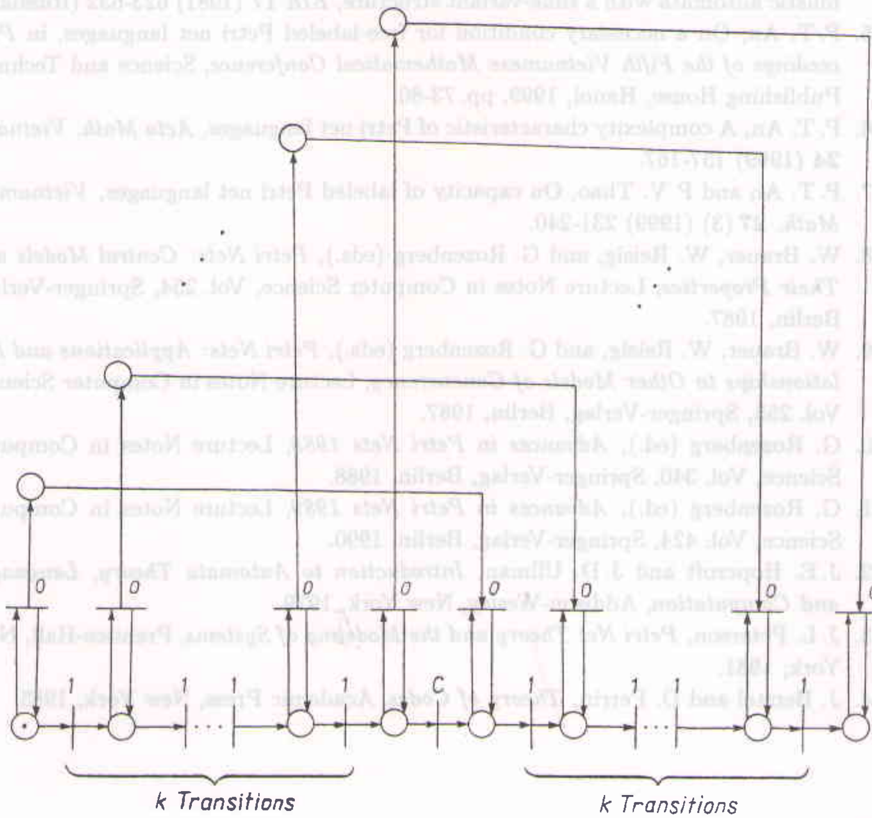

Fig. 2.

To obtain the sequence n_i of integers, it suffices to fix a $k \geq 2$ and put $n_1 = k - 1$, $n_{i+1} = 3n_i + 6$ for all $i \geq 1$.

The theorem is proved. ∎

Acknowledgement. The authors would like to thank the referee for making some valuable suggestions for improving the presentation of the paper.

References

1. S. N. Cole, Real-time computation by n-dimensional iterative arrays of finite state machines, *IEEE Trans. Comp.* **C-18** (1969) 349-365.
2. P. D. Dieu, On a complexity characteristic of languages, *EIK* **8** (1972) 447-460.
3. P. D. Dieu and P. T. An, Probabilistic automata with a time-variant structure, *EIK* **12** (1976) 3-27.

4. P. T. An, Some necessary conditions for the class of languages accepted by probabilistic automata with a time-variant structure, *EIK* **17** (1981) 623-632 (Russian).
5. P. T. An, On a necessary condition for free-labeled Petri net languages, in *Proceedings of the Fifth Vietnamese Mathematical Conference*, Science and Technics Publishing House, Hanoi, 1999, pp. 73-80.
6. P. T. An, A complexity characteristic of Petri net languages, *Acta Math. Vietnam.* **24** (1999) 157-167.
7. P. T. An and P. V. Thao, On capacity of labeled Petri net languages, *Vietnam J. Math.* **27** (3) (1999) 231-240.
8. W. Brauer, W. Reisig, and G. Rozenberg (eds.), *Petri Nets: Central Models and Their Properties*, Lecture Notes in Computer Science, Vol. 254, Springer-Verlag, Berlin, 1987.
9. W. Brauer, W. Reisig, and G. Rozenberg (eds.), *Petri Nets: Applications and Relationships to Other Models of Concurrency*, Lecture Notes in Computer Science, Vol. 255, Springer-Verlag, Berlin, 1987.
10. G. Rozenberg (ed.), *Advances in Petri Nets 1988*, Lecture Notes in Computer Science, Vol. 340, Springer-Verlag, Berlin, 1988.
11. G. Rozenberg (ed.), *Advances in Petri Nets 1989*, Lecture Notes in Computer Science, Vol. 424, Springer-Verlag, Berlin, 1990.
12. J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, New York, 1979.
13. J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, New York, 1981.
14. J. Berstel and D. Perrin, *Theory of Codes*, Academic Press, New York, 1985.