INFORMATION AND COMPUTATION **124**, 168–181 (1996)
ARTICLE NO. 0013

On the Regularity of Petri Net Languages

HSU-CHUN YEN

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, Republic of China
E-mail: yen@cc.ee.ntu.edu.tw

Petri nets are known to be useful for modeling concurrent systems. Once modeled by a Petri net, the behavior of a concurrent system can be characterized by the set of all executable transition sequences, which in turn can be viewed as a language over an alphabet of symbols corresponding to the transitions of the underlying Petri net. In this paper, we study the language issue of Petri nets from a computational complexity viewpoint. We analyze the complexity of the *regularity problem* (i.e., the problem of determining whether a given Petri net defines an irregular language or not) for a variety of classes of Petri nets, including *conflict-free*, *trap-circuit*, *normal*, *sinkless*, *extended trap-circuit*, *BPP*, and *general* Petri nets. (Extended trap-circuit Petri nets are trap-circuit Petri nets augmented with a specific type of *circuits.*) As it turns out, the complexities for these Petri net classes range from NL (nondeterministic logspace), PTIME (polynomial time), and NP (nondeterministic polynomial time), to EXPSPACE (exponential space). In the process of deriving the complexity results, we develop a *decomposition approach* which, we feel, is interesting in its own right, and might have other applications to the analysis of Petri nets as well. As a by-product, an NP upper bound of the reachability problem for the class of extended trap-circuit Petri nets (which properly contains that of trap-circuit (and hence, conflict-free) and BPP-nets, and is incomparable with that of normal and sinkless Petri nets) is derived.

1. INTRODUCTION

Petri nets (or equivalently, vector addition systems) represent a formalism useful for modeling concurrent systems. Once modeled by a Petri net, the behavior of a system can be characterized by the set of all executable transition sequences, which in turn can be viewed as a language over an alphabet of symbols corresponding to the transitions of the underlying Petri net. As a result, studying Petri nets from the aspect of Formal Language Theory has long been recognized as an important branch of research in Petri net theory. For results along this line of research, see, e.g., Ginzburg and Yoeli, 1980; Jantzen and Petersen, 1994; Peterson, 1981; Schwer, 1986; Schwer, 1992a; Schwer, 1992b; Valk and Vidal-Naquet, 1981. Among them, it has been shown that all Petri net languages are context-sensitive, assuming that a transition's symbol cannot be λ. Also, Petri net languages and context-free languages are incomparable, i.e., there are Petri net languages that are not context-free and vice versa. One particular application stemming from the study of Petri net languages has to do with determining the modeling power of Petri nets. For example, it has been shown in (Agerwala and Flynn, 1973; Kosaraju, 1973) that Petri nets, in general, cannot model problems involving "priority." A more recent work (of Howell, Rosier and Yen, 1993) focuses on the modeling powers of subclasses of Petri nets, including conflict-free, persistent, normal, and sinkless Petri nets. More precisely, persistent (and also conflict-free) Petri nets were shown to be unable to model the well-known *producer-consumer* and *mutual exclusion* problems in concurrent systems. Normal and sinkless Petri nets, although capable of modeling the producer-consumer problem, lack the capability to model unrestricted mutual exclusion. (A restricted version of mutual exclusion in which the total number of exclusions in any computation is bounded by a fixed constant can be modeled by normal and sinkless Petri nets, even though it still cannot be modeled by persistent Petri nets.) See (Howell, Rosier, and Yen, 1993) for more details. The interested reader is referred to (Peterson, 1981) for more motivations about the study of Petri net languages.

This paper deals with the complexity analysis of determining whether a Petri net defines a regular language or not. Such a problem will be referred to as the *regularity problem* throughout the rest of this paper. For general Petri nets, the regularity problem was first shown to be decidable in (Ginzburg and Yoeli, 1980; Valk and Vidal-Naquet, 1981). (Recently, the decidability result has been generalized to testing context-freeness of Petri net languages by Schwer (1992b).) In particular, the work of Valk and Vidal-Naquet (1981) yields a necessary and sufficient condition for a Petri net language to be irregular. More precisely, the language associated with a Petri net is *not* regular iff there exists a computation $\mu_0 \xmapsto{\sigma_1} \mu_1 \xmapsto{\sigma_2} \mu_2 \xmapsto{\sigma_3} \mu_3 \xmapsto{\sigma_4} \mu_4$ (where μ_0 is the initial marking), for some transition sequences $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ and markings $\mu_1, \mu_2, \mu_3, \mu_4$, such that

(1) $\mu_1 \leqslant \mu_2$ and $\mu_1 \neq \mu_2$,

(2) $\mu_1(p) = \mu_2(p)$ implies $\mu_3(p) \leqslant \mu_4(p)$, for every place p, and

(3) $\mu_3(p) > \mu_4(p)$, for some place p.

0890-5401/96 $18.00

TABLE 1

Complexities of the Regularity Problem for Various Petri Net Classes

Petri net class	Complexity result
Conflict-free	PTIME-complete
BPP	NL-complete
Trap-circuit	NP-complete
Normal	NP-complete
Extended Trap-circuit	NP-complete
Sinkless	NP-complete
General	EXPSPACE-complete

(Here $\mu_i(p)$ denotes the number of tokens in place p in marking μ_i. Intuitively, σ_4 constitutes a "pumpable loop" which can be fired an arbitrary number of times provided that a sufficient number of non-negative "loops" σ_2's are fired in advance. Furthermore, the firing of σ_4 results in some place losing tokens.) With the help of the above conditions, it has subsequently been shown by Yen (1992) that if a Petri net defines an irregular language, there must exist a "short" path (whose length is at worst double-exponential in the size of the Petri net) that witnesses the above conditions. As a consequence, an EXPSPACE upper bound follows. In this paper, the complexity of the regularity problem for a number of subclasses of Petri nets is investigated. Our results are summarized in Table 1. The containment relationships among these Petri nets are depicted in Fig. 1. We assume familiarity with basic definitions in complexity theory. The reader is referred to (Hopcroft and Ullman, 1979) for details. With the exception of the EXPSPACE result for general Petri nets, all the remaining complexity results are new.

Conflict-free, trap-circuit, normal, sinkless, BPP, and extended trap-circuit Petri nets have one thing in common: they are subclasses of Petri nets with constraints imposed on their *circuits*. (A *circuit* of a Petri net is simply a closed path (i.e., a cycle) in the Petri net graph.) By and large, the presence of complex circuits is troublesome in Petri net analysis. In fact, strong evidence has suggested that circuits constitute the major stumbling block in the analysis of Petri nets. To get a feel for why this is the case, it is well known that in a Petri net $\mathscr{P}$ with initial marking μ_0, a marking μ is reachable (from μ_0) in $\mathscr{P}$ *only if* there exists a column vector $x \in \mathbf{N}^m$ such that $\mu_0 + A \cdot x = \mu$, where m is the number of transitions in $\mathscr{P}$ and A is the *addition matrix* of $\mathscr{P}$. The converse, however, does not necessarily hold. In fact, lacking a simple necessary and sufficient condition for reachability in general has been blamed for the high degree of complexity in the analysis of Petri nets. (Otherwise, one could tie the reachability analysis of Petri nets to the *integer linear programming* problem, which is relatively well understood.) There are restricted classes of Petri nets for which necessary and sufficient conditions for reachability are available. Most notable, of course, is the class of *circuit-free* Petri nets (i.e., Petri nets without circuits) for which the equation $\mu_0 + A \cdot x = \mu$ is sufficient and necessary to capture reachability. A slight relaxation of the circuit-freedom constraint yields the same necessary and sufficient condition for the class of Petri nets without *token-free* circuits in every reachable marking (Yamasaki, 1984). Conflict-free, normal, and sinkless Petri nets have been extensively studied in the literature; see, e.g., Esparza, 1992; Howell, and Rosier, 1988; Howell, Rosier, and Yen, 1987; Howell, Rosier, and Yen, 1993; Landweber, and Robertson, 1978; Yamasaki, 1984. BPP-nets, defined and studied by Esparza (1994), provide an alternative view of the so-called *commutative context-free grammars* (Huynh, 1983), and are also strongly related to the model of *Basic Parallel Processes* (see, e.g., Christensen, Hirshfeld, and Moller, 1993). To the best of our knowledge, the class of extended trap-circuit Petri nets defined in this paper is new. Basically they are trap-circuit Petri nets "augmented" with a simple type of circuits, which will be referred to as $\oplus$-circuits throughout the rest of this paper. (Let c: $p_1 t_1 p_2 \cdots p_n t_n p_1$ (where $p_1, ..., p_n$ are places and $t_1, ..., t_n$ are transitions) be a circuit. Circuit c is a $\oplus$-circuit if for every i, p_i is t_i's sole input place, and the firing of t_i

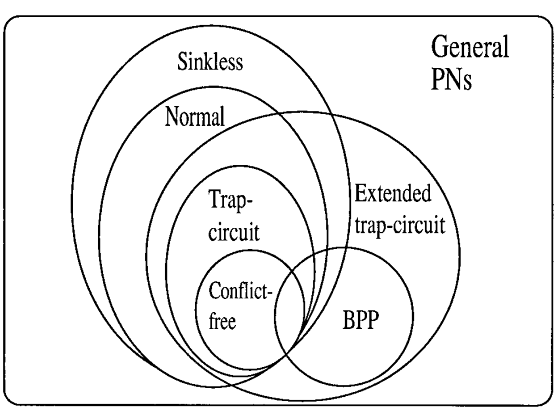

FIG. 1. Containment relationships among various Petri net classes.

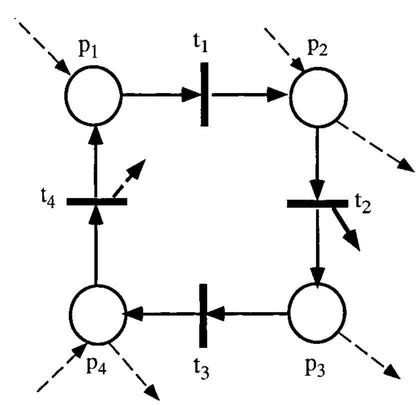

FIG. 2. A $\oplus$-circuit.

removes exactly one token from p_i. See Fig. 2.) Simply speaking, in an extended trap-circuit Petri net every "non-trap" circuit must be a $\oplus$-circuit. With respect to extended trap-circuit Petri nets, our analysis yields NP-completeness for both the reachability and the regularity problems. We feel that broadening the set of computationally analyzable Petri net classes is also one of the contributions of this paper.

Our strategy of proving the NP upper bounds listed in Table 1 relies on characterizing the reachability problem for the respective class of Petri nets by integer linear programming, utilizing a decomposition approach which will be developed in Section 3. Such a strategy is applied to conflict-free, trap-circuit, normal, sinkless, and extended trap-circuit Petri nets. By taking advantage of several nice properties uniquely offered by conflict-free Petri nets and utilizing the notions and results of the so-called *iterable factors* defined by Schwer (1992a), we are able to come up with an integer-preserving transformation from integer linear programming to linear programming, giving rise to a PTIME upper bound of the regularity problem for conflict-free Petri nets. In the process of doing so, we also yield a simplified sufficient and necessary condition under which conflict-free Petri nets define irregular languages. Such a result is interesting in its own right. For BPP-nets, the regularity problem will be solved by exploring the Petri net graph to see whether a path (from a graph-theoretic viewpoint) meeting certain conditions exists. As it turns out, such a test can be carried out in nondeterministic logspace. Finally, all the complexities mentioned in Table 1 are tight. The lower bound proofs are easy modifications of the boundedness (or reachability) problem's one for the respective classes of Petri nets.

The remainder of this paper is organized as follows. In Section 2, we define the basic notations and definitions of Petri nets. In Section 3, we develop a decomposition approach through which integer linear programming will be applied to solving the regularity problem. Section 4 concerns itself with the complexity analysis of the regularity problem for various Petri net classes described in Fig. 1.

2. PRELIMINARIES

Let $\mathbf{Z}$ ($\mathbf{N}$) denote the set of (nonnegative) integers, and $\mathbf{Z}^k$ ($\mathbf{N}^k$) the set of vectors of k (nonnegative) integers. For a k-dimensional vector v, let $v(i)$, $1 \leqslant i \leqslant k$, denote the ith component of v. For a $k \times m$ matrix A, let $a_{i,j}$, $1 \leqslant i \leqslant k$, $1 \leqslant j \leqslant m$, denote the element in the ith row and the jth column of A, and let a_j denote the jth column of A. For a given value of k, let $\mathbf{0}$ (resp. $\mathbf{1}$) denote the vector of k zeros (resp. ones) (i.e., $\mathbf{0}(i)=0$ (resp. $\mathbf{1}(i)=1$) for $i=1,..., k$). We let $|S|$ be the number of elements in set S. Given a column vector x, we let x^{T} denote the *transpose* of x (which is a row vector).

A *Petri net* (*PN*, for short) is a 3-tuple (P, T, φ), where P is a finite set of *places*, T is a finite set of *transitions*, and φ is a *flow function* $\varphi\colon (P \times T) \cup (T \times P) \to \{0, 1\}$. *In this paper, k and m will be reserved for $|P|$ (the number of places in P) and $|T|$ (the number of transitions in T), respectively.* A *marking* is a mapping $\mu\colon P \to N$. A transition $t \in T$ is *enabled* at a marking μ iff for every $p \in P$, $\varphi(p, t) \leqslant \mu(p)$. A transition t may *fire* at a marking μ if t is enabled at μ. We then write $\mu \xmapsto{t} \mu'$, where $\mu'(p) = \mu(p) - \varphi(p, t) + \varphi(t, p)$ for all $p \in P$. A sequence of transitions $\sigma = t_1, ..., t_n$ is a *firing sequence* from μ_0 iff $\mu_0 \xmapsto{t_1} \mu_1 \xmapsto{t_2} \cdots \xmapsto{t_n} \mu_n$ for some sequence of markings $\mu_1, ..., \mu_n$. (We also write "$\mu_0 \xmapsto{\sigma} \mu_n$.") We write "$\mu_0 \xmapsto{\sigma}$" to denote that σ is enabled and can be fired from μ_0, i.e., $\mu_0 \xmapsto{\sigma}$ iff there exists a marking μ such that $\mu_0 \xmapsto{\sigma} \mu$. A *marked* PN is a pair $((P, T, \varphi), \mu_0)$, where (P, T, φ) is a PN, and μ_0 is a marking called the *initial marking*. Throughout the rest of this paper, the word 'marked' will be omitted if it is clear from the context.

Given a PN (P, T, φ) and a set of transitions $H \subseteq T$, we define the *restriction* of φ to H, written as $\varphi|_H$, to be a mapping $\varphi|_H\colon (P \times H) \cup (H \times P) \to \{0, 1\}$ such that $\varphi|_H(p, t) = \varphi(p, t)$ and $\varphi|_H(t, p) = \varphi(t, p)$, for every $p \in P$, and $t \in H$. A PN (P, T', φ') is said to be a *sub-PN* of PN (P, T, φ) if $T' \subseteq T$, and $\varphi' = \varphi|_{T'}$. By establishing an ordering on the elements of P and T (i.e., $P = \{p_1, ..., p_k\}$ and $T = \{t_1, ..., t_m\}$), we define the $k \times m$ *addition matrix* A of (P, T, φ) so that $a_{i,j} = \varphi(t_j, p_i) - \varphi(p_i, t_j)$. Thus, if we view a marking μ as a k-dimensional column vector in which the ith component is $\mu(p_i)$, each column a_j of A is then a k-dimensional vector such that if $\mu \xmapsto{t_j} \mu'$, then $\mu' = \mu + a_j$. Let $\mathscr{P} = ((P, T, \varphi), \mu_0)$ be a PN. The *reachability set* of $\mathscr{P}$ is the set $R(\mathscr{P}) = \mu \mid \mu_0 \xmapsto{\sigma} \mu$ for some $\sigma\}$.

For ease of expression, the following notations will be used extensively throughout the rest of this paper. (Let σ, σ' be transition sequences, p be a place, t be a transition, Q be a set of places, and H be a set of transitions.)

- $\#_\sigma(t)$ represents the number of occurrences of t in σ. (For convenience, we sometimes treat $\#_\sigma$ as an m-dimensional vector assuming that an ordering on T is established ($|T| = m$).)
- $\Delta(\sigma) = A \cdot \#_\sigma$ defines the *displacement* of σ. (Notice that if $\mu \xmapsto{\sigma} \mu'$, the $\Delta(\sigma) = \mu' - \mu$.)
- $\mathrm{Tr}(\sigma) = \{t \mid t \in T, \#_\sigma(t) > 0\}$, denoting the set of transitions used in σ.
- $\|\sigma\|^+ = \{p \mid p \in P, \Delta(\sigma)(p) > 0\}$ is the *positive support* of σ.
- $\|\sigma\|^- = \{p \mid p \in P, \Delta(\sigma)(p) < 0\}$ is the *negative support* of σ.
- $\|\sigma\|^0 = \{p \mid p \in P, \Delta(\sigma)(p) = 0\}$ is the *zero support* of σ.
- $\sigma \mathbin{\dot-} \sigma'$ is defined inductively as follows. Suppose $\sigma' = t_1, ..., t_n$. Let σ_0 be σ. If t_i is in σ_{i-1}, let σ_i be σ_{i-1} with

the leftmost occurrence of t_i deleted; otherwise, let $\sigma_i = \sigma_{i-1}$. Finally, let $\sigma \dot{-} \sigma' = \sigma_n$. For example, if $\sigma = t_1 t_2 t_3 t_4 t_5$ and $\sigma' = t_4 t_3 t_1$, then $\sigma \dot{-} \sigma' = t_2 t_5$.

- $p^\bullet = \{t \mid \varphi(p, t) \geqslant 1,\ t \in T\}$ is the set of output transitions of p;

  $t^\bullet = \{p \mid \varphi(t, p) \geqslant 1,\ p \in P\}$ is the set of output places of t;

  $Q^\bullet = \bigcup_{p \in Q} p^\bullet$; $H^\bullet = \bigcup_{t \in H} t^\bullet$.

- ${}^\bullet p = \{t \mid \varphi(t, p) \geqslant 1,\ t \in T\}$ is the set of input transitions of p;

  ${}^\bullet t = \{p \mid \varphi(p, t) \geqslant 1,\ p \in P\}$ is the set of input places of t;

  ${}^\bullet Q = \bigcup_{p \in Q} {}^\bullet p$; ${}^\bullet H = \bigcup_{t \in H} {}^\bullet t$.

If $\mu_0 \stackrel{\sigma}{\longmapsto} \mu$, then $\mu_0 + A \cdot \#_\sigma = \mu$. (Note that the converse does not necessarily hold.) Given a path $\mu \stackrel{\sigma}{\longmapsto} \mu'$, a sequence σ' is said to be a *rearrangement* of σ if $\#_\sigma = \#_{\sigma'}$ and $\mu \stackrel{\sigma'}{\longmapsto} \mu'$.

A *circuit* of a PN is a "simple" closed path in the PN graph. (By "simple" we mean all nodes are distinct along the closed path.) Given a PN $\mathscr{P}$, let $c = p_1 t_1 p_2 t_2 \cdots p_n t_n p_1$ be a circuit and let μ be a marking. Let $P_c = \{p_1, p_2, \cdots, p_n\}$ denote the set of places in c. With a slight abuse of notation, we also use c to denote $t_1 t_2 \cdots t_n$ when the exact order is not important. We define the token count of circuit c in marking μ to be $\mu(c) = \sum_{p \in P_c} \mu(p)$. A circuit c is said to be *token-free* in μ iff $\mu(c) = 0$. We say c is *minimal* iff P_c does not properly include the set of places in any other circuit. Circuit c is said to have a *sink* iff for some $\mu \in R(\mathscr{P})$ and some σ and μ' such that $\mu \stackrel{\sigma}{\longmapsto} \mu'$, $\mu(\mathrm{c}) > 0$, but $\mu'(c) = 0$. Circuit c is said to be *sinkless* iff it does not have a sink. (See Yamasaki (1984) for more details.) Circuit c is said to be a $\oplus$-circuit iff for every i, $1 \leqslant i \leqslant n$, ${}^\bullet t_i = \{p_i\}$. A set of places Q is called a *trap* iff $(\forall t \in \mathrm{T})((\exists p \in Q,\ t \in p^\bullet) \Rightarrow (\exists q \in Q,\ t \in {}^\bullet q))$, i.e., any transition which has an input place in Q must also have an output place in Q. A set of $\oplus$-circuits $\mathscr{C} = \{c_1, c_2, ..., c_n\}$ is said to be *connected* iff for every i, j, $1 \leqslant i$, $j \leqslant n$, there exist $1 \leqslant h_1, h_2, ..., h_r \leqslant n$, for some r, such that $h_1 = i$, $h_r = j$, and for every $1 \leqslant l < r$, $P_{c_{h_l}} \cap P_{c_{h_{l+1}}} \neq \emptyset$. Given a path $\mu \stackrel{\sigma}{\longmapsto}$, σ is said to *cover* $\oplus$-circuit c if $\#_c \leqslant \#_\sigma$, i.e., every transition of c appears in σ.

Given an alphabet A, we write A^* to denote the set of all finite-length strings (including the empty string λ) using symbols from A. We write A^+ to denote $A^* - \{\lambda\}$. (See Hopcroft and Ullman (1979) for more details.) For a language L (over an alphabet A) and a word $u \in A^+$, u is said to be an *iterable factor* (of L) iff $\forall n \geqslant 0$, $A^* u^n A^* \cap L \neq \emptyset$. For a *prefix-closed* language L, u is an *iterable factor* (of L) iff $\forall n \geqslant 0$, $A^* u^n \cap L \neq \emptyset$. (A language L is prefix-closed if $w \in L$ implies every prefix of w is also in L.) Given a PN $\mathscr{P} = ((P, T, \varphi), \mu_0)$, the *language* associated with $\mathscr{P}$ over alphabet T, denoted as $L(\mathscr{P})$, is the set $\{\sigma \mid \mu_0 \stackrel{\sigma}{\longmapsto}\}$. Clearly, Petri net languages are prefix-closed. The *regularity problem* is that of determining whether $L(\mathscr{P})$ defines an irregular language or not.

In this paper, we mainly focus on the following subclasses of Petri nets. Their containment relationships are depicted in Fig. 1.

- *Conflict-free Petri nets*: A PN $\mathscr{P} = (P, T, \varphi)$ is said to be *conflict-free* iff for every place p, either

1. $|p^\bullet| \leqslant 1$, *or*
2. $\forall t \in p\bullet$, t and p are on a self-loop.

In words, a PN is conflict-free if every place which is an input of more than one transition is on a self-loop with each such transition (Jones, Landweber, and Lien, 1977; Landweber, and Robertson, 1978). In a conflict-free PN, once a transition becomes enabled, the only way to disable the transition is to fire the transition itself. (That is, $\forall t$, $t' \in T$, $t \neq t'$, $\mu \stackrel{t}{\longmapsto} \mu'$ and $\mu \stackrel{t'}{\longmapsto}$ implies $\mu' \stackrel{t'}{\longmapsto}$.)

- *Normal Petri nets*: A PN is *normal* (Yamasaki, 1984) iff for every minimal circuit c and transition t_j, $\sum_{p_i \in P_c} a_{i,j} \geqslant 0$. (Recall that $a_{i,j} = \varphi(t_j, p_i) - \varphi(p_i, t_j)$.) Hence, for every minimal circuit c and transition t in a normal PN, if one of t's input places is in c, then one of t's output places must be in c as well. Intuitively, a Petri net is normal iff no transition can decrease the token count of a minimal circuit by firing at any marking.

- *Sinkless Petri nets*: A PN $\mathscr{P}$ is said to be *sinkless* (Yamasaki, 1984) iff each minimal circuit of $\mathscr{P}$ is sinkless.

- *BPP nets*: A PN (P, T, φ) is said to be a *BPP*-net (Esparza, 1994) if $\forall t \in T$, $|{}^\bullet t| = 1$, i.e., every transition has exactly one input place. (Notice that every arc going from a place to a transition has weight 1.)

- *Trap-circuit Petri nets*: A PN $\mathscr{P}$ is a *trap-circuit* PN (Ichikawa and Hiraishi, 1987) iff for every circuit c in $\mathscr{P}$, P_c is a trap.

- *Extended Trap-circuit Petri nets*: A PN $\mathscr{P}$ is an *extended trap-circuit* PN iff for every circuit c in $\mathscr{P}$, either P_c is a trap or c is a $\oplus$-circuit.

The interested reader is referred to (Murata, 1989; Peterson, 1981; Reisig, 1985) for more about Petri nets and their related problems.

3. A DECOMPOSITION APPROACH FOR TESTING REACHABILITY

In (Valk and Vidal-Naquet, 1981), a necessary and sufficient condition has been derived for checking whether the language defined by a PN is regular or not. More precisely:

THEOREM 3.1 (from Valk and Vidal-Naquet, 1981)). *The language of a* PN $\mathscr{P} = ((P, T, \varphi), \mu_0)$ *is not regular iff there exists a path* $\mu \stackrel{*}{\mapsto} \mu_1 \stackrel{*}{\mapsto} \mu_2 \stackrel{*}{\mapsto} \mu_3 \stackrel{*}{\mapsto} \mu_4$ *in* $\mathscr{P}$, *for some markings* μ_1, μ_2, μ_3 *and* μ_4, *such that*

(a) $\mu_1 \leqslant \mu_2$ *and* $\mu_1 \neq \mu_2$,

(b) $\mu_1(p) = \mu_2(p)$ *implies* $\mu_3(p) \leqslant \mu_4(p)$, *for every* $p \in P$, *and*

(c) $\mu_3(p) > \mu_4(p)$, *for some* $p \in P$.

In words, the sequence from μ_3 to μ_4 constitutes an iterable factor with at least one place losing tokens.

In this paper, the above result is going to serve as the core around which our complexity analysis will be built. Another important ingredient of our complexity analysis lies in the ability to model reachability as integer linear programming. Given a PN $\mathscr{P} = ((P, T, \varphi), \mu_0)$ and two arbitrary markings μ and μ', suppose testing whether $\mu \overset{*}{\mapsto} \mu'$ is equivalent to solving a system of linear inequalities, say $ILP(\mathscr{P}, \mu, \mu')$, then the regularity problem for PN $\mathscr{P}$ can be answered by solving the following system of linear inequalities (in which x_0, x_1, x_2, x_3 and x_4 are variables of dimension k (i.e., the number of places in $\mathscr{P}$)) over the integers:

(1) $x_0 = \mu_0$;

(2) $ILP(\mathscr{P}, x_0, x_1)$; $ILP(\mathscr{P}, x_1, x_2)$; $ILP(\mathscr{P}, x_2, x_3)$; $ILP(\mathscr{P}, x_3, x_4)$;

(3) $(x_2 \geqslant x_1) \wedge (\bigvee_{i=1}^{k} (x_2(i) > x_1(i)))$;

(4) $\bigwedge_{i=1}^{k} ((x_1(i) < x_2(i)) \vee (x_3(i) \leqslant x_4(i)))$;

(5) $\bigvee_{i=1}^{k} (x_3(i) > x_4(i))$.

Conditions (3), (4), and (5) capture the essence of conditions (a), (b), and (c) of Theorem 3.1, respectively. Since integer linear programming is known to be solvable in NP, the upper bound follows provided that the above system of linear inequalities is of size polynomial in $\mathscr{P}$. (Because NP is what we are aiming for, the "$\bigvee_{i=1}^{k}$" in (3) and (5) above can be dealt with by guessing an i first and then setting up the system of linear inequalities accordingly.)

Over the past years, considerable effort has been spent on finding necessary and sufficient conditions for reachability for restricted classes of Petri nets. See, e.g., Murata (1989). Among such results, the following lemma will be used later in this paper to derive some of our results.

LEMMA 3.2 (from Yamasaki, 1984). *If a PN* $\mathscr{P} = ((P, T, \varphi), \mu_0)$ *has no token-free circuits in every reachable marking, then* $R(\mathscr{P}) = \{\mu \mid \mu = \mu_0 + A \cdot x \geqslant 0$, *for some* $x \in N^m\}$, *where m is the number of transitions in T.*

Despite the fact that necessary and sufficient conditions for reachability are hard to come by in general, integer linear programming has long been recognized as a powerful tool for analyzing PNs. A notable example concerns the classes of *normal* and *sinkless* PNs (Howell, Rosier, and Yen, 1993; Yamasaki, 1984). The idea behind the analysis of normal and sinkless PNs lies in constructing the reachability set in a greedy fashion. To do so, we build a sequence of small sub-PNs, each of which has its reachability set characterized by an integer linear programming instance. Furthermore, the number of sub-PNs as well as the size of each integer linear programming instance are polynomial in the size of the original PN. As a consequence, testing reachability for a normal (sinkless) PN can be equated with solving a system of linear inequalities (in the integer domain). In what follows, we generalize the idea employed by Howell, Rosier, and Yen (1993) to come up with what we call a "*decomposition approach*" to analyze PNs.

Given a PN $\mathscr{P} = (P, T, \varphi)$, a *decomposition* of $\mathscr{P}$ is a sequence of PNs $\mathscr{P}_1, \mathscr{P}_2, ..., \mathscr{P}_d$, for some integer $d \geqslant 1$, such that $\mathscr{P}_i = (P, T_i, \varphi_i)$, $T_i \subseteq T$, and φ_i is the restriction of φ to T_i. Let A_i be the addition matrix of PN $\mathscr{P}_i$. (It should be noted that the T_i, $1 \leqslant i \leqslant d$, are not in general disjoint.)

The crux of testing whether μ is reachable from μ_0 relies on setting up a system of linear inequalities

$$\begin{cases} x_{i-1} + A_i y_i = x_i & (1) \\ F_i(P, T_i, \varphi_i) & (2) \\ x_0 = \mu_0 \text{ and } x_d = \mu & (3) \end{cases}$$

$(1 \leqslant i \leqslant d)$ in such a way that (1) is a necessary condition for marking x_i to be reachable from x_{i-1}, F_i in (2) captures extra constraints with which (1) becomes sufficient as well, and (3) describes the initial and final markings. (Here $x_i \in N^k$ and $y_i \in N^{m_i}$ are variables, where $k = |P|$ and $m_i = |T_i|$.) Now the strategy for setting up $ILP(\mathscr{P}, \mu_0, \mu)$, i.e., a system of linear inequalities for checking whether μ is reachable or not, consists of the following steps:

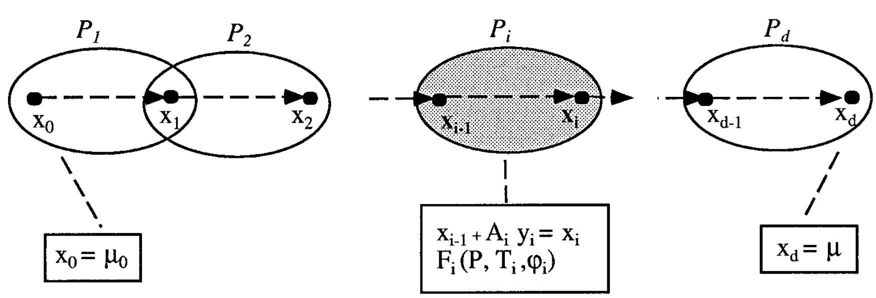

Fig. 3. Decomposition of a PN.

(i) Guess a decomposition of $\mathscr{P}$, say $\mathscr{P}_1, \mathscr{P}_2, ..., \mathscr{P}_d$, where $\mathscr{P}_i = (P, T_i, \varphi_i)$.

(ii) Set up the linear inequalities listed in (1), (2), and (3) above.

(iii) Prove that if $\mu_0 \overset{*}{\mapsto} \mu$ in $\mathscr{P}$, then there exists a path $\mu_0 \xmapsto{\sigma_1} \mu_1 \xmapsto{\sigma_2} \cdots \xmapsto{\sigma_d} \mu_d$ such that $(\forall 1 \leqslant i \leqslant d)$ $(Tr(\sigma_i) \subseteq \mathrm{T}_i)$.

Obviously, (ii) and (iii) imply the reachability of μ from μ_0. What (iii) says is that if μ is reachable, then there must *exist* a "canonical" path reaching μ such that the path can be decomposed into a sequence of subpaths coinciding with the PN decomposition (see Fig. 3). In the following section, we demonstrate that for a number of classes of PNs, reachability can be determined with the help of the above decomposition approach.

4. COMPLEXITY ANALYSIS OF THE REGULARITY PROBLEM

The main theme of this section is to investigate the regularity problem from a computational complexity viewpoint for various PN classes shown in Fig. 1. Our results are summarized in Table 1. With the exception of the EXPSPACE result for general PNs, all of our complexity results are new. Our approach of proving the NP upper bound relies on characterizing the reachability problem by integer linear programming using the decomposition approach discussed in Section 3. Such a strategy is applied to conflict-free, normal, sinkless, trap-circuit, and extended trap-circuit PNs. For conflict-free PNs, we are able to come up with an integer-preserving transformation from integer linear programming to linear programming, yielding a PTIME upper bound. For BPP-nets, a nondeterministic logspace procedure will be developed to solve the regularity problem. Finally, all the complexities mentioned in Table 1 will be shown to be tight. The lower bound proofs are easy modifications of the boundedness (or reachability) problem's one for the respective classes of PNs.

We begin with trap-circuit, normal and sinkless PNs. The interested reader is referred to (Howell, Rosier, and Yen, 1993; Ichikawa, and Hiraishi, 1987; Yamasaki, 1984) for more about these three classes of PNs.

4.1. *Trap-Circuit, Normal and Sinkless Petri Nets*

Let $x_0 \xmapsto{t_{j_1}} \cdots x_1 \xmapsto{t_{j_2}} \cdots x_{i-1} \xmapsto{t_{j_i}} \cdots x_{n-1} \xmapsto{t_{j_n}} \cdots x_n$ be a path in a sinkless PN reaching μ such that x_{i-1}, $1 \leqslant i \leqslant n$, marks the first time at which transition t_{j_i} fires. ($t_{j_1}, t_{j_2}, \cdots, t_{j_n}$ are distinct.) Now consider a sequence of PNs $\mathscr{P}_1, ..., \mathscr{P}_n$ ($\mathscr{P}_i = (P, T_i, \varphi_i)$) such that $T_0 = \varnothing$, $T_i = \{t_{j_1}, ..., t_{j_i}\}$, and φ_i is the restriction of φ to T_i, for $1 \leqslant i \leqslant n$. It has been shown by Howell, Rosier, and Yen (1993) that for every i, the following system of linear inequalities exactly characterizes the reachability set of PN ($\mathscr{P}_i, x_{i-1}$):

$$S_i = \begin{cases} x_{i-1} + A_i y_i = x_i & (1) \\ x_{i-1}(l) \geqslant \varphi(p_l, t_{j_i}),\ 1 \leqslant l \leqslant k & (2) \end{cases}$$

That is, marking x_i is reachable in ($\mathscr{P}_i$, x_{i-1}) iff there exists a solution y_i in the integer domain. ((2) is to ensure that transition t_{j_i} is enabled in x_{i-1}.) The validity of the above argument is based upon the following lemma and Lemma 3.2:

LEMMA 4.1 (from Howell, Rosier, and Yen, 1993). *Let $\mathscr{P} = ((P, T, \varphi), \mu_0)$ be a sinkless* PN, *and let $\mathscr{P}' = ((P, T', \varphi'), \mu)$ be such that $\mu_0 \xmapsto{\sigma} \mu$ in $\mathscr{P}$ for some σ, $T' \subseteq T$ such that each $t \in T'$ is enabled at some point in the firing of σ from μ_0, and φ' is the restriction of φ to T'. Then $\mathscr{P}'$ has no token-free circuits in any reachable marking.*

As a result, μ is reachable iff $\{x_0 = \mu_0\} \cup \{x_n = \mu\} \cup \bigcup_{1 \leqslant i \leqslant n} \{S_i\}$ has an integer solution. Hence, we have:

LEMMA 4.2. (from Howell, Rosier, and Yen, 1993). *Given a trap-circuit (normal, or sinkless)* PN *$\mathscr{P} = ((P, T, \varphi), \mu_0)$ and a marking μ, we can construct, in nondeterministic polynomial time, a system of linear inequalities $ILP(\mathscr{P}, \mu_0, \mu)$ in such a way that μ is reachable from μ_0 iff $ILP(\mathscr{P}, \mu_0, \mu)$ has an integer solution.*

Using Lemma 4.2 and the decomposition approach proposed in Section 3, we immediately have:

THEOREM 4.3. *The regularity problem for trap-circuit (normal, and sinkless) PNs is solvable in NP.*

4.2. *Extended Trap-Circuit Petri Nets*

In what follows, we show that the decomposition approach discussed in Section 3 can be applied to solving

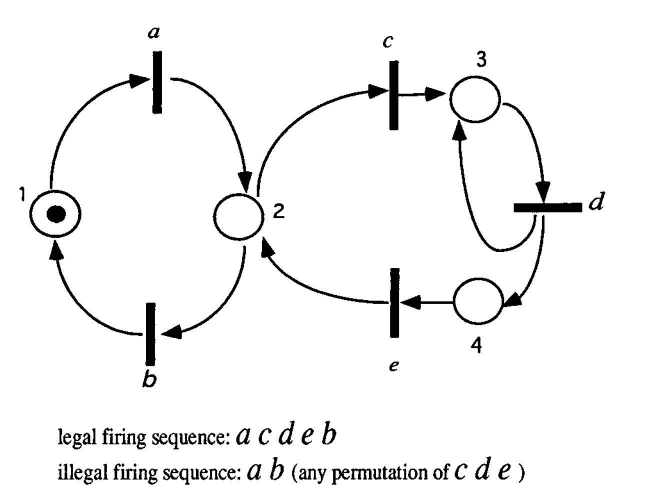

FIG. 4. An unsuccessful attempt of restructuring a path with $\oplus$-circuits.

the reachability problem as well as the regularity problem for extended trap-circuit PNs. In the literature, one of the few techniques proven to be useful for analyzing PNs relies on the ability to rearrange PN paths into some "canonical" form. As one might expect, the nature of $\oplus$-circuits, in particular, the ability to repeat a $\oplus$-circuit an arbitrary number of times at any marking at which the circuit is marked suggests a good starting point for devising a rearrangement technique. The first attempt, perhaps, is to fire a $\oplus$-circuit immediately when one of its transitions becomes enabled, even though the transitions of the $\oplus$-circuit are interleaved with others in the original path. Unfortunately, such an attempt does not work, as Fig. 4 indicates. To remedy such a difficulty, we first present a nice property concerning any set of connected $\oplus$-circuits.

LEMMA 4.4. *Given a set of connected $\oplus$-circuits $\mathscr{C}=\{c_1, c_2, ..., c_n\}$ in a PN $\mathscr{P}$ and a marking μ with $\mu(c_i)>0$, for some i, then for arbitrary integers $a_1, a_2, ..., a_n>0$, there exists a sequence σ such that $\mu \overset{\sigma}{\longmapsto}$ and $\#_\sigma=\sum_{j=1}^{n} a_j(\#_{c_j})$. (In words, from μ there exists a firable sequence σ utilizing circuit c_j exactly a_j times, for every j.)*

Proof. Without loss of generality, we assume $i=1$, and let p_1 be a place in c_1 such that $\mu(p_1)>0$. The proof is done by induction on the number of circuits in $\mathscr{C}$.

(Induction Basis) For $n=1$, the result is trivial.

(Induction Hypothesis) Assume that the assertion is true for $n \leqslant h$.

(Induction Step) Consider $n=h+1$. Starting from place p_1, let $p_2, .., p_r$, for some r, be places along c_1 that are shared with other circuits in $\mathscr{C}$. Let $\mathscr{C}_j$ $(1 \leqslant j \leqslant r)$ be the largest connected subset of $\mathscr{C}-\{c_1\}-(\bigcup_{l \leqslant j-1} \mathscr{C}_l)$ for which one of its circuits contains place p_j. By induction hypothesis, all circuits in $\mathscr{C}_j$ can be fired arbitrarily, provided that p_j is marked. Let $\alpha_i \in T^*$ $(1 \leqslant i \leqslant r)$ be the transition sequence from place p_i to p_{i+1} along circuit c_1 (assuming that $p_{r+1}=p_1$). Then the desired sequence σ is the following: (sequence guaranteed by induction hypothesis for $\mathscr{C}_1$) α_1 (sequence guaranteed by induction hypothesis for $\mathscr{C}_2$) $\cdots \alpha_{r-1}$ (sequence guaranteed by induction hypothesis for $\mathscr{C}_r$) α_r $(\alpha_1 \cdots \alpha_r)^{a_1-1}$. ∎

The idea of rearranging an arbitrary path in an extended trap-circuit PN into a "canonical" one is as follows. Suppose $\mu \overset{\sigma}{\longmapsto}$ is a path, and c is a $\oplus$-circuit covered by σ such that $\mu(c)>0$. Then we use c as a "seed" to grow the largest collection of connected $\oplus$-circuits that are covered by σ. We then follow a transition sequence of the remaining path until we reach a marking in which a non-token-free $\oplus$-circuit (with respect to the current marking) which is covered by the subsequent path exists. Using such a newly found circuit as a new seed and repeating the above procedure, we are able to arrange an arbitrary path of an extended trap-circuit PN into a "canonical" one as the following theorem indicates. Notice that the above procedure need not be repeated for more than m times, because for each of the circuits collected in a marking, at least one of its transitions must be absent from the remaining path. See Fig. 5 for a pictorial description of such a rearrangement strategy.

We are now ready to present one key lemma on which our decomposition approach for extended trap-circuit PNs relies.

LEMMA 4.5. *Consider a path $\mu_1 \overset{\sigma}{\longmapsto} \mu_2$ in an extended trap-circuit PN $\mathscr{P}=(P, T, \varphi)$. Let $\mathscr{C}=\{c_1, c_2, ..., c_n\}$ be a set of connected $\oplus$-circuits and $a_1, a_2, ..., a_n$ be positive integers such that:*

(1) $(\exists i, 1 \leqslant i \leqslant n)$ $(\mu_1(c_i)>0)$ *(i.e., c_i is not token-free in marking μ_1)*

(2) $\sigma \dot{-} ((c_1)^{a_1} \cdots (c_n)^{a_n})$ *does not cover any $\oplus$-circuit that shares some place with circuits in $\mathscr{C}$.*

Then there exist δ_1 and δ_2 such that

(a) $\#_{\delta_1}=\sum_{j=1}^{n} a_j(\#_{c_j})$,

(b) $\#_{\delta_2}=\#_{\sigma \dot{-} \delta_1}$, *and*

(c) $\mu_1 \overset{\delta_1}{\longmapsto} \mu_3 \overset{\delta_2}{\longmapsto} \mu_2$, *for some μ_3.*

Proof. First notice that the existence of a δ_1 witnessing $\mu_1 \overset{\delta_1}{\longmapsto} \mu_3$ is guaranteed by Lemma 4.4; it suffices to prove that $\mu_3 \overset{\delta_2}{\longmapsto} \mu_2$, for some δ_2 which is a rearrangement of

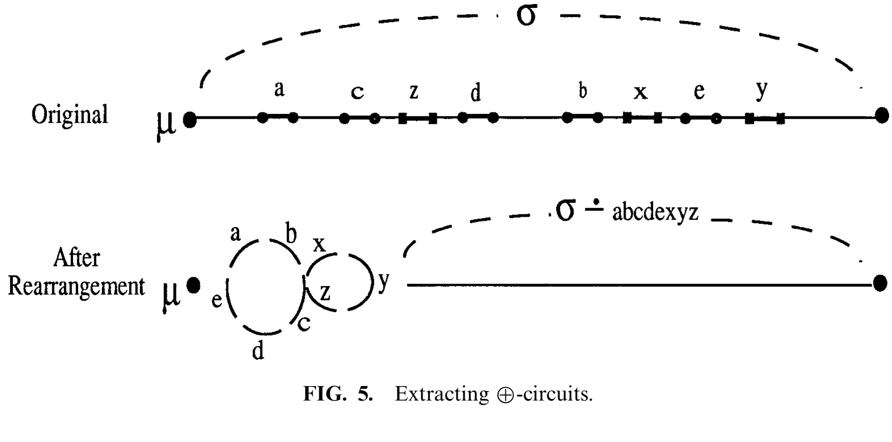

FIG. 5. Extracting $\oplus$-circuits.

$\sigma \dot{-} \delta_1$. Suppose, to the contrary, that none of the permutations of $\sigma \dot{-} \delta_1$ is firable in μ_3. We let α be a longest sequence such that $\#_\alpha < \#_{\sigma \dot{-} \delta_1}$ and $\mu_3 \xmapsto{\alpha} \mu_4$, for some μ_4. (By "longest" we mean that for all α' with $\#_{\alpha'} < \#_{\sigma \dot{-} \delta_1}$ and $\mu_3 \xmapsto{\alpha'}$, it must be the case that $|\alpha'| \leqslant |\alpha|$.) Let $\beta = (\sigma \dot{-} \delta_1) \dot{-} \alpha$. Clearly, in μ_4 every transition in $\mathrm{Tr}(\beta)$ must have at least one of its input places empty. (Otherwise, α could be extended–violating the assumption about α being longest.) See Fig. 6. We let X be $\{p \mid \mu_4(p) = 0,\ p \in {}^\bullet t,\ t \in \mathrm{Tr}(\beta)\}$, i.e., X consists of all the input places (which are token-free in μ_4) of transitions in $\mathrm{Tr}(\beta)$. We now make the following observations:

1. $\forall p \in X$, $\exists t' \in \mathrm{Tr}(\beta)$, such that $p \in t'^\bullet$. (This is because $\mu_4(p) + \Delta(\beta)(p) = \mu_2(p) \geqslant 0$ and $\mu_4(p) = 0$.)

2. There must be some place r in X such that either (i) $\mu_1(r) > 0$, *or* (ii) $(\exists t_1 \in \mathrm{Tr}(\delta_1 \alpha))\ (r \in t_1^\bullet)$. And for each such r, $\exists t_2 \in \mathrm{Tr}(\delta_1 \alpha)$ such that $r \in {}^\bullet t_2$. (Assume, to the contrary, that neither (i) nor (ii) holds. In σ, let f be the *first* transition depositing a token into some place in X. Since $f \notin \mathrm{Tr}(\delta_1 \alpha)$, f's input place, say g, must be in X. In this case, place g could never have possessed a token along the path from μ_1 to the marking at which f is fired—a contradiction. The existence of a t_2 results from $\mu_4(r) = 0$.)

Let R be the set of all places r satisfying Observation 2(i) or (ii) above. What we need next is to show that at least one place in R must be along a circuit consisting of some places in X and some transitions in $\mathrm{Tr}(\beta)$. Suppose, to the contrary, that none of R is on a circuit; then there must be an $s \in R$ such that s cannot be reached from the remaining places in R through places in X and transitions in $\mathrm{Tr}(\beta)$. For s, let t_3 be a transition guaranteed by Observation 1 above. Due to the selection of s, t_3 could never have been fired in σ since its input place would never possess a token (because the input place of t_3 (i.e., ${}^\bullet t_3$) is not in R, and none of R is capable of supplying a token to ${}^\bullet t_3$ directly or indirectly)—a contradiction. Intuitively, one can think of R as places through which tokens are "pumped" into the sub-PN consisting of places in X and transitions in $\mathrm{Tr}(\beta)$.

Let $r \in R$ be a place on a circuit, say c, and t_2 (whose existence is guaranteed by Observation 2) be a transition in $\delta_1 \alpha$ removing a token from r. (Note that c is token-free in μ_4.) Clearly P_c is not a trap; otherwise, c would not have become token-free in μ_4. Hence, c is a $\oplus$-circuit. If t_2 is in δ_1 (which comprises only circuits from $\mathscr{C}$), then c must have shared some place with one of the circuits in $\mathscr{C}$—violating Assumption (2) of the lemma. If t_2 is in α, then r is marked during the course of the path α, which implies that c should have been added to α—violating the assumption about α being longest. This completes the proof of the lemma. ▮

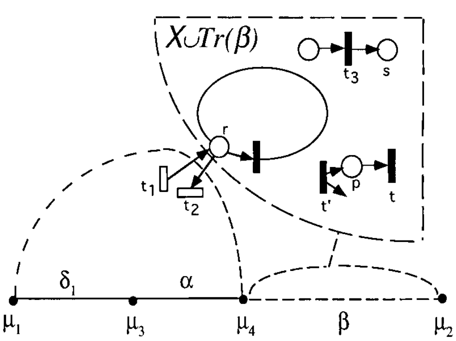

FIG. 6. A picture illustrating the concept used in the proof of Lemma 4.5.

With the help of the above lemma, we have:

THEOREM 4.6. *Let μ be a reachable marking in an extended trap-circuit* PN $\mathscr{P} = ((P, T, \varphi), \mu_0)$. *Then there exist a decomposition $\mathscr{P}_1, \mathscr{P}_2, ..., \mathscr{P}_{2h}$ (where $\mathscr{P}_i = (P, T_i, \varphi_i)$, $1 \leqslant i \leqslant 2h$) and a sequence $\pi_1 \alpha_1 \cdots \pi_h \alpha_h$ which witnesses $\mu_0 \xmapsto{\pi_1 \alpha_1 \cdots \pi_h \alpha_h} \mu$ such that*

(1) $1 \leqslant h \leqslant m$ *(m is the number of transitions),*

(2) $T_{2i-1} = \mathrm{Tr}(\pi_i)$, *and* $T_{2i} = \mathrm{Tr}(\alpha_i)$,

(3) $\forall i$, $1 \leqslant i \leqslant h$, *there exists a set $\mathscr{C}_i = \{c_1^i, ..., c_{r_i}^i\}$ of connected $\oplus$-circuits, where $r_i \leqslant m$, such that $\Delta(\pi_i) = \sum_{j=1}^{r_i} a_j^i \Delta(c_j^i)$ for some positive integers $a_1^i, ..., a_{r_i}^i > 0$. Furthermore, the remaining sequence $\alpha_i \cdots \pi_n \alpha_n$ does not cover any $\oplus$-circuit which shares some place with circuits in $\mathscr{C}_i$.*

(4) $\forall i$, $1 \leqslant i \leqslant h$, $\alpha_i \in T^+$, $(P, \mathrm{Tr}(\alpha_i), \varphi|_{\mathrm{Tr}(\alpha_i)})$ *(i.e., $\mathscr{P}_{2i}$) forms a trap-circuit sub-PN.*

Proof. Given a sequence σ witnessing $\mu_0 \xmapsto{\sigma} \mu$, if $(P, \mathrm{Tr}(\sigma), \varphi|_{\mathrm{Tr}(\sigma)})$ is not a trap-circuit sub-PN, then the following procedure can be used for constructing the desired rearrangement $\pi_1 \alpha_1 \pi_2 \alpha_2 \cdots \pi_h \alpha_h$.

(1) **Procedure** *decompose*$(((P, T, \varphi), \mu_0), \sigma)$
(2) $i := 1$; $\delta := \sigma$; $\mu := \mu_0$
(3) **while** $\delta \neq \lambda$ (i.e., the empty string) **do**
(4) $\quad \mathscr{C}_i := \varnothing$; $j := 1$
(5) $\quad$ let c_1^i be a $\oplus$-circuit covered by δ and $\mu(c_1^i) > 0$
(6) $\quad$ **do forever**
(7) $\quad\quad \mathscr{C}_i := \mathscr{C}_i \cup \{c_j^i\}$
(8) $\quad\quad \delta := \delta \dot{-} \overbrace{c_j^i \cdots c_j^i}^{a_j^i}$, where $a_j^i \cdot \#_{c_j^i} \leqslant \#_\sigma$ but $a \cdot \#_{c_j^i} \nleqslant \#_\sigma$, $\forall a > a_j^i$
(9) $\quad\quad$ **if** there exists a $\oplus$-circuit covered by δ and connected to $\mathscr{C}_i$
(10) $\quad\quad\quad$ **then** $j := j + 1$; let c_j^i be such a $\oplus$-circuit
(11) $\quad\quad\quad$ **else EXIT**
(12) $\quad$ **end do**
(13) $\quad$ let π_i be such that $\mu \xmapsto{\pi_i}$ and $\Delta(\pi_i) = \sum_{l=1}^{j} a_l^i \Delta(c_l^i)$ (* guaranteed by Lemma 4.4.*)
(14) $\quad$ rearrange δ so that $\mu \xmapsto{\pi_i \delta}$ (* guaranteed by Lemma 4.5*)

(15) **if** Tr(δ), together with its associated places, is a trap-circuit sub-PN

(16) **then EXIT**

(17) **else** let α_i be the *shortest* prefix of δ such that

(18) $\delta \dot{-} \alpha_i$ covers a $\oplus$-circuit which is not token-free in μ', where $\mu \xmapsto{\pi_i \alpha_i} \mu'$;

(19) $\delta := \delta \dot{-} \alpha_i$; $\mu := \mu'$; $i := i+1$

(20) **end while**

(21) **end procedure**

In the above procedure, variable δ keeps track of the remaining sequence as the construction proceeds. Lines (3)–(12) constitute the extraction of connected $\oplus$-circuits in a greedy fashion. (Line (8), in particular, is used for extracting the maximum number of occurrences of c_j^i in δ.) The "do" loop continues until no more connected $\oplus$-circuits can be found in the remaining sequence. The existence of a π_i satisfying the conditions listed in Line (13) is guaranteed by Lemma 4.4. In addition, the remaining δ can be rearranged into a firable sequence as Line (14) indicates. (See also Lemma 4.5) Lines (17)–(19) find the shortest prefix α_i along which all the $\oplus$-circuits covered by the remaining sequence (i.e., δ) remain token-free. Clearly (P, Tr(α_i), $\varphi|_{\mathrm{Tr}(\alpha_i)}$) is a trap-circuit sub-PN. The "while" loop then repeats anew. ▮

LEMMA 4.7. *Given an extended trap-circuit* PN $\mathscr{P}$ $(=(P, T, \varphi))$, *detecting each of the following can be done in polynomial time.*

(1) *There exist a $\oplus$-circuit c and a transition t such that* $t \notin \mathrm{Tr}(c)$, ${}^{\bullet}t \cap P_c \neq \varnothing$, *and* $t^{\bullet} \cap P_c = \varnothing$ *(i.e., P_c is not a trap)*

(2) *Given a set of $\oplus$-circuits $\mathscr{C}$, there exists a $\oplus$-circuit c' such that $c' \notin \mathscr{C}$ and c' shares a place with some circuit in $\mathscr{C}$.*

Proof. For each transition t and one of its input places p, we check whether there exists a circuit $pt_1 p_1, ..., p_{r-1} t_r p$, for some r, such that $\forall 1 \leqslant i \leqslant r$, $|{}^{\bullet}t_i| = 1$ and $t^{\bullet} \cap \{p, p_1, ..., p_r\} = \varnothing$. Clearly, checking the existence of such a circuit can be done in NL (and hence, PTIME) using a nondeterministic search procedure. Hence, (1) follows. To prove (2), pick a place p in $\mathscr{C}$, and a place q not in $\mathscr{C}$. (2) holds iff there exists a $\oplus$-circuit passing through p and q. Clearly, such a test can be done in polynomial time. ▮

The interested reader should contrast the above with the NP-completeness result of checking whether a given PN is normal or not (see (Howell, Rosier, and Yen, 1993)).

THEOREM 4.8. *The regularity problem for extended trap-circuit PNs is solvable in NP.*

Proof. Given a PN $(\mathscr{P}, \mu_0)$ (where $\mathscr{P} = (P, T, \varphi)$), $\mu \in R(\mathscr{P}, \mu_0)$ iff there exists a decomposition $\mathscr{P}_1, \mathscr{P}_2, ..., \mathscr{P}_{2h}$ satisfying the conditions stated in Theorem 4.6. The system of linear inequalities associated with the reachability problem can be set up as follows:

(1) Guess $\mathscr{P}_1, \mathscr{P}_2, ..., \mathscr{P}_{2h}$, where $0 \leqslant h \leqslant m$,

(2) Verify conditions (3) and (4) of Theorem 4.6, which can be done in PTIME (see Lemma 4.7). For all even i, set up ILP($\mathscr{P}_i, x_i, x_{i+1}$), which is the system of linear inequalities (guaranteed by Lemma 4.2) for testing the reachability of x_{i+1} from x_i in trap-circuit PN $\mathscr{P}_i$,

(3) For all odd i, let $\{c_1^i, ..., c_{r_i}^i\}$ be the set of connected $\oplus$-circuits guaranteed by Theorem 4.6. Set up linear inequalities $\{x_i(c_j^i) > 0$, for some $j\} \cup \{x_i + \sum_{j=1}^{r_i} (z_j * \Delta(c_j^i)) = x_{i+1}\}$. (Notice that $x_i(c_j^i) > 0$ is to ensure that one of the $\oplus$-circuits is marked. $(z_j * \Delta(c_j^i))$ denotes executing $\oplus$-circuit c_j^i z_j times, where z_j is a scalar variable.)

Based on the decomposition strategy of Section 3, our result follows. ▮

4.3. *BPP-Nets*

Using the concept of *siphons*, it was shown by Esparza (1994) that the reachability problem for BPP-nets is solvable in NP, and the reachability set of a BPP-net is always semilinear. What makes BPP-nets interesting, as pointed out by Esparza (1994), is that BPP-nets provide an alternative view of the so-called *commutative context-free grammars* (Huynh, 1983), and are also strongly related to the model of *Basic Parallel Processes* (see, e.g., Christensen, Hirshfeld, and Moller, 1993) which has received much attention in concurrency theory recently.

With respect to the regularity problem, we have the following simple necessary and sufficient condition for BPP-nets.

LEMMA 4.9. *Given a BPP-net* $P = ((P, T, \varphi), \mu_0)$, *the language defined by $\mathscr{P}$ is not regular iff*

(1) $\exists$ *places p, p' and a $\oplus$-circuit $c : p_1 t_1 p_2 \cdots t_n p_1$ such that $\mu_0(p) > 0$, $p \rightsquigarrow p_1$ and $\Delta(c)(p') > 0$, and*

(2) $\exists$ *a place q and a transition t such that $p' \rightsquigarrow q$, and $t \in q^{\bullet} - {}^{\bullet}q$.*

(See Fig. 7 for a pictorial description of the above two conditions.)

Proof. Recall from Theorem 3.1 that $L(\mathscr{P})$ is not regular iff $\mu_0 \overset{*}{\mapsto} \mu_1 \overset{*}{\mapsto} \mu_2 \overset{*}{\mapsto} \mu_3 \overset{*}{\mapsto} \mu_4$, for some markings μ_1, μ_2, μ_3 and μ_4, such that the path from μ_1 to μ_2 constitutes a pumpable loop which supplies tokens to the iterable factor from μ_3 to μ_4 which has at least one place losing tokens. Due to the structure of BPP-nets, it is not hard to see that Theorem 3.1 can be simplified as our lemma indicates. The detail is left to the reader. ▮

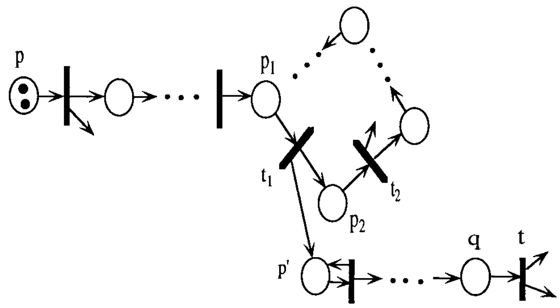

FIG. 7. A path in a BPP-net graph witnessing irregularity.

The two conditions stated in the above lemma can easily be checked in NL using a nondeterministic search procedure; hence, we have

THEOREM 4.10. *The regularity problem for BPP-nets is in NL.*

4.4. *Conflict-Free Petri Nets*

Before deriving our PTIME upper bound for conflict-free PNs, we require a few known results.

LEMMA 4.11 (from Howell, Rosier, and Yen, 1993). *Given a conflict-free* PN $\mathscr{P}=((P, T, \varphi), \mu_0)$, *we can construct in polynomial time a sequence* π *in which no transition in* $\mathscr{P}$ *is used more than once, such that if some transition t is not used in* π, *then there is no path in which t is used.*

In words, the above lemma guarantees the existence of a "short" sequence, i.e., π, which collects all the potentially firable transitions in a given conflict-free PN.

LEMMA 4.12 (from Yen, 1991). *Let* $\mu \xmapsto{\sigma} \mu'$ *be a path in a conflict-free* PN $\mathscr{P}=(P, T, \varphi)$. *Then there exist* σ_1 *and* σ_2 *such that*

(1) $\#_\sigma = \#_{\sigma_1\sigma_2}$,

(2) $\mu \xmapsto{\sigma_1\sigma_2} \mu'$,

(3) $\mathrm{Tr}(\sigma_2) \subseteq \mathrm{Tr}(\sigma_1)$, *and*

(4) $(\forall\, t \in T)\,(\#_{\sigma_1}(t) \leqslant 1)$.

In words, $\sigma_1\sigma_2$ is a rearrangement of σ such that if a transition occurs in σ, it can also be found in σ_1; in addition, no transition in σ_1 appears more than once in σ_1. It is important to note that the result holds for arbitrary μ. By repeatedly applying Lemma 4.12, we have:

COROLLARY 4.13. *For an arbitrary path* $\mu \xmapsto{\sigma} \bar{\mu}$ *in a conflict-free* PN $\mathscr{P}=(P, T, \varphi)$, σ *can be rearranged into* $\overbrace{\sigma_1 \cdots \sigma_1}^{l_1}\overbrace{\sigma_2 \cdots \sigma_2}^{l_2} \cdots \overbrace{\sigma_d \cdots \sigma_d}^{l_d}$, *for some sequences* $\sigma_1, \sigma_2, ..., \sigma_d$ *and integers* $l_1, l_2, ..., l_d$, $1 \leqslant d \leqslant m$ (*m is the number of transitions*), *such that*

(1) $(\forall 1 \leqslant i \leqslant d)\,(\forall\, t \in \mathrm{T})\,(\#_{\sigma_i}(t) \leqslant 1)$, *and*

(2) $(\forall 1 \leqslant i \leqslant d-1)\,(\mathrm{Tr}(\sigma_{i+1}) \subsetneq \mathrm{Tr}(\sigma_i))$.

What the above corollary says is that σ can be rearranged into a "canonical" sequence consisting of pieces of short segments. Furthermore, the sequence $\mathrm{Tr}(\sigma_1)$, $\mathrm{Tr}(\sigma_2)$, ..., $\mathrm{Tr}(\sigma_d)$, $1 \leqslant d \leqslant m$, forms a "shrinking" sequence of sets. A direct and important consequence of such a shrinking sequence (in conjunction with the PN being conflict-free) is stated as follows, which governs the pattern of sign change regarding $\Delta(\sigma_1)(p)$, $\Delta(\sigma_2)(p)$, ..., $\Delta(\sigma_d)(p)$ for a place p.

LEMMA 4.14. *Let* $\sigma_1, \sigma_2, ..., \sigma_d$, $1 \leqslant d \leqslant m$, *be transition sequences as stated in Corollary* 4.13. *Then*

(1) $(\forall p)(\forall i)(-1 \leqslant \Delta(\sigma_i)(p) \leqslant m)$,

(2) $(\forall p)(\forall i)(\Delta(\sigma_i)(p) < 0 \Rightarrow (\forall i < l \leqslant d)(\Delta(\sigma_l)(p) \leqslant 0))$,

(3) $(\forall p)(\forall j)((\exists i > j)(\Delta(\sigma_j)(p) > 0 \wedge \Delta(\sigma_i)(p) < 0) \Rightarrow (\forall l < j)(\Delta(\sigma_l)(p) > 0))$, *and*

(4) $(\forall p)(\forall g)((\exists i > g)(\Delta(\sigma_g)(p) = 0 \wedge \Delta(\sigma_i)(p) < 0) \Rightarrow (\forall l \geqslant g)(\Delta(\sigma_l)(p) \leqslant 0))$.

Proof. First notice that for an arbitrary transition t and an arbitrary place p in a conflict-free PN, if $\Delta(t)(p) = -1$, then t is the sole transition that removes a token from p. Using the above fact and $|\sigma_i| \leqslant m$, for all i, (1) is rather obvious. For a place p and a segment σ_i, if p loses tokens as a result of firing σ_i, then $(\forall t \in {}^\bullet p, t \notin \mathrm{Tr}(\sigma_i))$. This implies for every subsequent segment σ_h, $h > i$, $(\forall t \in {}^\bullet p, t \notin \mathrm{Tr}(\sigma_h))$ (due to the shrinking property); hence, (2) follows. In addition, if one of the σ_i's preceding segment, say σ_j, has a positive gain in p, then every segment preceding σ_j must have a positive gain in p as well, yielding (3). On the other hand, if one of σ_i's preceding segment, say σ_g, has a zero gain in p, then none of the subsequent segments of σ_g can have a positive gain. Hence, (4) holds. ∎

We are now ready to embark for the regularity problem. To begin with, we show that if the language of a PN is not regular, then there exists a "short" witnessing path with "good" properties.

LEMMA 4.15. *Given a conflict-free* PN $\mathscr{P}=((P, T, \varphi), \mu_0)$, $L(\mathscr{P})$ *is not regular iff there exists a path* $\mu_0 \xmapsto{\pi} \mu_1 \xmapsto{\delta} \mu_2 \xmapsto{\tau} \mu_3$, *for some sequences* π, δ, τ *and markings* μ_1, μ_2, μ_3, *such that*

(1) π *is the sequence guaranteed by Lemma* 4.11,

(2) $\|\delta\|^- = \varnothing$ (*i.e.*, $\Delta(\delta) \geqslant \mathbf{0}$),

(3) $\|\tau\|^- \subseteq \|\delta\|^+$ (*i.e.*, $\forall p \in P$, $\Delta(\delta)(p) = 0 \Rightarrow \Delta(\tau)(p) \geqslant 0$),

(4) $\#_\tau \leqslant \mathbf{1}$ (*i.e.*, $\forall t \in T$, *t occurs at most once in* τ), *and*

(5) $|\delta|$ (*i.e., the length of* δ) $\leqslant 3m^2$, *where m is the number of transitions in T.*

Proof. The *if* part follows from Theorem 3.1; in what follows, we consider the *only if* part.

According to Theorem 3.1, if $L(\mathscr{P})$ is not regular, then there exists a path $\mu_0 \xmapsto{\alpha} M_1 \xmapsto{\beta} M_2 \xmapsto{\gamma} M_3 \xmapsto{\omega} M_4$, for some transition sequences α, β, γ, ω and markings M_1, M_2, M_3, M_4, such that $\|\beta\|^- = \emptyset$, $\|\omega\|^- \neq \emptyset$, and $\|\omega\|^- \subseteq \|\beta\|^+$. Using the result of Corollary 4.13, ω can be rearranged into $\overbrace{\sigma_1 \cdots \sigma_1}^{l_1}\overbrace{\sigma_2 \cdots \sigma_2}^{l_2} \cdots \overbrace{\sigma_d \cdots \sigma_d}^{l_d}$, for some integers $l_1, l_2, ..., l_d$ and sequences $\sigma_1, \sigma_2, ..., \sigma_d$ satisfying the conditions stated in Corollary 4.13. Let i, $1 \leqslant i \leqslant d$, be the *smallest* index such that $\|\overbrace{\sigma_1 \cdots \sigma_1}^{l_1} \cdots \overbrace{\sigma_i \cdots \sigma_i}^{l_i}\|^- \neq \emptyset$. (Since $\|\omega\|^- \neq \emptyset$, such an i must exist.) Clearly, $\Delta(\overbrace{\sigma_1 \cdots \sigma_1}^{l_1} \cdots \overbrace{\sigma_{i-1} \cdots \sigma_{i-1}}^{l_{i-1}}) \geqslant \mathbf{0}$ (if $i > 1$), and $\|\sigma_i\|^- \neq \emptyset$, for i is smallest. Let $\varepsilon = \overbrace{\sigma_1 \cdots \sigma_1}^{l_1} \cdots \overbrace{\sigma_{i-1} \cdots \sigma_{i-1}}^{l_{i-1}}$, if $i > 1$. We claim that for every $p \in \|\sigma_i\|^-$, either $p \in \|\varepsilon\|^+$, or $p \in \|\beta\|^+$. To see this, observe that if $p \notin \|\varepsilon\|^+$, then for all j, $i < j \leqslant d$, $\Delta(\sigma_j)(p) \leqslant 0$ (Condition 4 of Lemma 4.14), indicating that $\Delta(\omega)(p) < 0$, and, hence, $p \in \|\beta\|^+$ (because $\|\omega\|^- \subseteq \|\beta\|^+$).

In view of the above, we have $\Delta(\beta) \geqslant \mathbf{0}$, $\Delta(\varepsilon) \geqslant \mathbf{0}$, and $\|\sigma_i\|^- \subseteq \|\beta\|^+ \cup \|\varepsilon\|^+$. Let π be the sequence guaranteed by Lemma 4.11 and $\mu_0 \xmapsto{\pi} \mu_1$. By Lemma 4.1, there is no token-free circuit reachable from μ_1. As a result, $\mu_1 + \Delta(\beta\varepsilon) \geqslant \mathbf{0}$ implies the existence of a rearrangement δ' of $\beta\varepsilon$ such that $\mu_0 \xmapsto{\pi} \mu_1 \xmapsto{\delta'} \mu_2'$, for some marking μ_2'. In addition, $\|\sigma_i\|^- \subseteq (\|\beta\|^+ \cup \|\varepsilon\|^+)(= \|\delta'\|^+)$ and $\#_{\sigma_i} \leqslant \mathbf{1}$ (i.e., every transition occurs at most once in σ_i) imply $\mu_2' + \Delta(\sigma_i) \geqslant \mathbf{0}$. Hence, there is a rearrangement τ of σ_i such that $\mu_0 \xmapsto{\pi} \mu_1 \xmapsto{\delta'} \mu_2' \xmapsto{\tau} \mu_3'$, for some marking μ_3'.

It remains to show that δ' can be made "short." To this end, it suffices to come up with a "short" δ such that $\Delta(\delta) \geqslant \mathbf{0}$, and $\|\delta'\|^+ \subseteq \|\delta\|^+$. By Corollary 4.13, δ' can be rearranged into $\overbrace{\delta_1 \cdots \delta_1}^{n_1}\overbrace{\delta_2 \cdots \delta_2}^{n_2} \cdots \overbrace{\delta_h \cdots \delta_h}^{n_h}$, for some integers h $(1 \leqslant h \leqslant m)$, $n_1, n_2, ..., n_h$ and sequences $\delta_1, \delta_2, ..., \delta_h$ satisfying the conditions stated in Corollary 4.13. We view $\Delta(\delta_1)\ \Delta(\delta_2) \cdots \Delta(\delta_h)$ as an $k \times h$ matrix. By Lemma 4.14, the following properties hold:

1. whenever a row contains a negative number, this number is -1, and the first column (i.e., the one corresponding to $\Delta(\delta_1)$) has a positive number in this row;
2. the sign sequence of any row without a negative entry is of the form $0^* +^* 0^*$.

To ensure $\|\delta'\|^+ \subseteq \|\delta\|^+$, we take a copy of δ_i, $i > 1$, for all those rows where a positive total change is required but $\Delta(\delta_1)$ is zero, plus enough copies (at most $2m$) of δ_1. In all, at most $m + 2m$ copies of $\delta_1, ..., \delta_h$ are needed, and each of which is of length at most m. According to Lemma 4.1, there exists a rearrangement δ of the above constructed sequence such that $\mu_0 \xmapsto{\pi} \mu_1 \xmapsto{\delta} \mu_2 \xmapsto{\tau} \mu_3$ (for some markings μ_2, μ_3) and Conditions (1), (2), (3), (4), and (5) of the lemma hold. This completes the proof of the lemma. ∎

Suppose $\sigma: \mu_0 \xmapsto{\pi} \mu_1 \xmapsto{\delta} \mu_2 \xmapsto{\tau} \mu_3$ is a path satisfying the conditions given in Lemma 4.15. Without loss of generality, we let $\pi = t_1 t_2 \cdots t_r$ be the sequence guaranteed by Lemma 4.11. In our subsequent discussion, we will restrict our attention to PN $\mathscr{P} = ((\mathrm{P}, \{t_1, t_2, ..., t_r\}, \varphi'), \mu_0)$, where φ' is the restriction of φ to $\{t_1, t_2, ..., t_r\}$. For convenience, we also label the set of places as $P = \{p_1, p_2, ..., p_s\}$. Now we are ready to set up a set of instances of linear programming $\{ILP(\mathscr{P}, p_1), ILP(\mathscr{P}, p_2), ..., ILP(\mathscr{P}, p_s)\}$ to capture the essence of the above path. Since π can be found in polynomial time (Lemma 4.11), segment $\mu_0 \xmapsto{\pi} \mu_1$ (more accurately, μ_1) will be computed in the beginning. As a result, only the suffix path starting at μ_1 needs to be expressed as a set of linear inequalities.

The construction of $ILP(\mathscr{P}, p_i)$, $1 \leqslant i \leqslant s$, is done as follows: (Let $\{t_{i_1}\} = p_i^\bullet$, and $\{t_{i_2}, ..., t_{i_a}\} = {}^\bullet p_i$ for some $a \geqslant 2$. Notice that for p_i to be in $\|\tau\|^-$, p_i cannot be on any self-loop.)

1. $\mu_1 \xmapsto{\delta} \mu_2$. Let variable x_i represent the number of occurrences of transition t_i, $1 \leqslant i \leqslant r$, in δ. For ease of expression, we let $x = (x_1, x_2, ..., x_r)^T$. Then we include the following inequalities:

(A1) $A \cdot x \geqslant \mathbf{0}$,

(A2) $x \geqslant \mathbf{0}$,

(A1) is sufficient to guarantee that μ_2 is reachable. (A2) is trivial.

2. $\mu_2 \xmapsto{\tau} \mu_3$. Let y_i be the number of occurrences of transition t_i, $1 \leqslant i \leqslant r$, in $\delta\tau$. For ease of expression, we let $y = (y_1, y_2, ..., y_r)^T$. Then we include the following inequalities:

(A3) $A \cdot y \geqslant \mathbf{0}$,

(A4) $y \geqslant x$,

(A5) $y_{i_1} = x_{i_1} + 1$, and $y_{i_j} = x_{i_j}$, $\forall j$, $2 \leqslant j \leqslant i_a$,

(A6) $y \leqslant x + \mathbf{1}$.

(A3) and (A4) are sufficient to guarantee that μ_3 is reachable through the firing of sequence $\delta\tau$. In addition, (A3) guarantees that $\|\tau\|^- \subseteq \|\delta\|^+$. (A5) ensures that τ is not empty as well as that $\|\tau\|^- \neq \emptyset$ (more precisely, $p_i \in \|\tau\|^-$). Finally, (A6) is to ensure that every transition in τ occurs at most once.

For every p_i, it is not hard to see that $ILP(\mathscr{P}, p_i)$ can be constructed in polynomial time. Now we are ready to present the following important theorem, which serves as the foundation upon which our polynomial time algorithm for detecting regularity relies. Based on the above discussion, the proof of the theorem should be straightforward.

THEOREM 4.16. *Given a conflict-free* PN $\mathscr{P}$, $\mathscr{P}$ *is not regular iff there exists a* $p_i \in P$ *such that* $ILP(\mathscr{P}, p_i)$ *has an integer solution.*

Given the fact that integer linear programming is NP-complete, tractability of the regularity problem does not come free of charge, even with the help of Theorem 4.16. What makes a speed-up possible is an integer-preserving transformation from integer linear programming to linear programming. (Our strategy was motivated by the work of Esparza (1992).) This is made possible by the result of Lemma 4.15 (in particular, Condition (4)), in conjunction with the unique feature of conflict-free PNs. Notice that being conflict-free alone is not sufficient, for the reachability problem for conflict-free PNs is known to be NP-complete (Howell and Rosier, 1988). The crux of this approach is that by adding additional constraints to a system of linear inequalities modeling the regularity problem of a conflict-free PN, if a solution (over the reals) exists, then the ceiling of that solution is itself a solution. The reader is referred to (Lenstra, 1983) for more general treatment of subclasses of integer linear programming (such as integer linear programming with a fixed number of variables) for which PTIME algorithms are available. The result of Lenstra (1983), however, cannot be applied directly to our analysis, for the number of variables in our case is not fixed.

To be more precise, we have:

THEOREM 4.17. *Given a conflict-free PN* $\mathscr{P}$ *and a place* p_i, $ILP(\mathscr{P}, p_i)$ *has an integer solution iff the following optimization problem has a solution* (*not necessarily over the integers*).

$$\textit{Maximize} \sum_{j=1}^{r} (x_j + y_j).$$

$$\textit{subject to} \begin{cases} ILP(\mathscr{P}, p_i) \\ 0 \leqslant x_j,\ y_j \leqslant 3m^2 + m,\ \forall 1 \leqslant j \leqslant r \end{cases}$$

where x_j *and* $y_j (1 \leqslant j \leqslant r)$ *are those variables used in expressing* δ *and* $\delta\tau$, *respectively*, *in* $ILP(\mathscr{P}, p_i)$.

Proof. Let $LP(\mathscr{P}, p_i)$ denote the above set of linear inequalities. Intuitively, $\sum_{j=1}^{r} x_j$ and $\sum_{j=1}^{r} y_j$ amount to the lengths of δ and $\delta\tau$, respectively. According to Lemma 4.15, there exists a short witness such that $|\delta| \leqslant 3m^2$. (Also recall that $|\tau| \leqslant m$.) As a result, $LP(\mathscr{P}, p_i)$ has a solution (which maximizes the given function) if $ILP(\mathscr{P}, p_i)$ has an integer solution.

To prove the converse, it suffices to show that the optimal solution of $LP(\mathscr{P}, p_i)$ is an integer solution. Let $(x_1, x_2, ..., y_1, y_2, ...)$ be the solution of $LP(\mathscr{P}, p_i)$. In what follows, we show that $(\lceil x_1 \rceil, \lceil x_2 \rceil, ..., \lceil y_1 \rceil, \lceil y_2 \rceil, ...)$ is a solution as well. To do so, recall that each inequality in $LP(\mathscr{P}, p_i)$ is of one of the following forms:

(1) $x_j (y_j) \geqslant 0$, for some j,

(2) $y_j \geqslant (=) x_j$, for some j,

(3) $y_j \leqslant (=) x_j + 1$, for some j,

(4) $\sum_{i=1}^{r} a_{j,i} x_i \geqslant 0$, or $\sum_{i=1}^{r} a_{j,i} y_i \geqslant 0$, where x_is, y_is are variables, and for each j, $a_{j,i}$s, $1 \leqslant i \leqslant r$, are the components of the jth row of the addition matrix A. (This type of inequality comes from (A1) and (A3).)

Clearly, (1), (2), and (3) remain after each variable being replaced by its ceiling. For case (4), first notice that due to the conflict-freedom property of $\mathscr{P}$, for each j, at most one component, say $a_{j,h}$, can be negative (and, if so, $= -1$). Hence, (4) can be rewritten as $\sum_{i=1..r, i \neq h} a_{j,i} x_i \geqslant (-a_{j,h}) x_h$, where $a_{j,h} = -1$. Clearly, $\sum_{i=1..r, i \neq h} a_{j,i} \lceil x_i \rceil \geqslant \lceil (\sum_{i=1..r, i \neq h} a_{j,i} x_i) \rceil \geqslant \lceil (-a_{j,h}) x_h \rceil = (-a_{j,h}) \lceil x_h \rceil$. Hence, $(\lceil x_1 \rceil, \lceil x_2 \rceil, ..., \lceil x_r \rceil)$ satisfies (4) as well. The case for y is similar. Finally, it is also obvious that $0 \leqslant \lceil x_j \rceil, \lceil y_j \rceil \leqslant 3m^2 + m$, if $0 \leqslant x_j,\ y_j \leqslant 3m^2 + m$, $\forall 1 \leqslant j \leqslant r$.

In light of the above, the optimal solution of $LP(\mathscr{P}, p_i)$ must be an integer solution. This completes the proof of our theorem. ∎

Since *Linear Programming* is well-known to be in PTIME (Khachian, 1979), we have:

THEOREM 4.18. *The regularity problem for conflict-free PNs is in PTIME.*

4.5. *General Petri Nets*

In (Yen, 1992), a class of path formulas has been defined for which the satisfiability problem has been shown to be solvable in EXPSPACE. As it turns out, the regularity problem is a special case of the satisfiability problem; hence, the EXPSPACE upper bound for the regularity problem follows (for general Petri nets). For the sake of completeness, we now briefly state the definition of the path formulas defined by Yen (1992), and show how regularity detection is related to satisfiability for general Petri nets. Let $((P, T, \varphi), \mu_0)$ be a k-place m-transition PN. Each path formula consists of the following elements:

1. Variables: There are two types of variables, namely, *marking variables* $\mu_1, \mu_2, ...$ and *variables for transition sequences* $\sigma_1, \sigma_2, ...$, where each μ_i denotes a vector in Z^k and each σ_i denotes a finite sequence of transitions.

2. Terms: Terms are defined recursively as follows:

(a) $\forall$ constant $c \in N^k$, c is a term.

(b) $\forall j > i$, $\mu_j - \mu_i$ is a term, where μ_i and μ_j are marking variables.

(c) $T_1 + T_2$ and $T_1 - T_2$ are terms if T_1 and T_2 are terms.

3. Atomic predicates: There are two types of atomic predicates, namely, *transition predicates* and *marking predicates.*

(a) *Transition predicates*:

• $y \odot \#_{\sigma_i} < c$, $y \odot \#_{\sigma_i} = c$ and $y \odot \#_{\sigma_i} > c$ are predicates, where $i > 1$, y (a constant) $\in Z^m$, $c \in N$ and $\odot$ denotes the inner product

• $\#_{\sigma_1}(t_j) \leqslant c$ and $\#_{\sigma_1}(t_j) \geqslant c$ are predicates, where $c \in N$ and $t_j \in T$.

(b) *Marking predicates*:

• *Type* 1: $\mu(i) \geqslant c$ and $\mu(i) > c$ are predicates, where μ is a marking variable and $c(\in Z)$ is a constant.

• *Type* 2: $T_1(i) = T_2(j)$, $T_1(i) < T_2(j)$ and $T_1(i) > T_2(j)$ are predicates, where T_1, T_2 are terms and $1 \leqslant i, j \leqslant k$, meaning that the ith component of T_1 equals, is less than, resp. is greater than the jth component of T_2, respectively.

$F_1 \vee F_2$ and $F_1 \wedge F_2$ are predicates if F_1 and F_2 are predicates.

In (Yen, 1992), the *satisfiability problem* for the following class of formulas has been shown to be solvable in EXPSPACE:

$$\exists \mu_1, ..., \mu_m \exists \sigma_1, ..., \sigma_m((\mu_0 \stackrel{\sigma_1}{\longmapsto} \mu_1 \stackrel{\sigma_2}{\longmapsto} \cdots \mu_{m-1} \stackrel{\sigma_m}{\longmapsto} \mu_m) \wedge F(\mu_1, ..., \mu_m, \sigma_1, ..., \sigma_m))$$

As it turns out, conditions (a), (b), and (c) stated in Theorem 3.1 can be expressed using the above class of path formulas as $\exists \mu_1, \mu_2, \mu_3, \mu_4 \exists \sigma_1, \sigma_2, \sigma_3, \sigma_4(\mu_0 \stackrel{\sigma_1}{\longmapsto} \mu_1 \stackrel{\sigma_2}{\longmapsto} \mu_2 \stackrel{\sigma_3}{\longmapsto} \mu_3 \stackrel{\sigma_4}{\longmapsto} \mu_4)$ and

(a′) $(\mu_2 \geqslant \mu_1) \wedge (\bigvee_{i=1}^{k} (\mu_2(i) > \mu_1(i)))$,

(b′) $\bigwedge_{i=1}^{k} ((\mu_1(i) < \mu_2(i)) \vee (\mu_3(i) \leqslant \mu_4(i)))$ and

(c′) $\bigvee_{i=1}^{k} (\mu_3(i) > \mu_4(i))$.

As a consequence, the following theorem holds.

THEOREM 4.19. *The regularity problem for general PNs is in EXPSPACE.*

4.6. *Lower Bounds*

This section is devoted to the derivation of the lower bounds of the regularity problem for those Petri net classes listed in Fig. 1. All the lower bound proofs are easy modifications of the boundedness (or reachability) problem's one for the respective classes of PNs. As a result, we only provide references and proof sketches.

THEOREM 4.20. *The regularity problem for conflict-free PNs is PTIME-hard.*

Proof (Sketch). The proof is done along the same line as that of showing the boundedness problem for conflict-free PNs to be PTIME-hard (Howell, Rosier, and Yen, 1987). The proof in (Howell, Rosier, and Yen, 1987) involves showing how the *path system problem* (which is well-known to be PTIME-complete) can be reduced to the boundedness problem for conflict-free PNs. Given an instance of the path system problem, we can construct a bounded conflict-free PN with a distinguished place p such that the path system instance has a solution iff a marking μ with $\mu(p) > 0$ is reachable in the constructed PN. Now by slightly modifying place p, one can force the new PN to be irregular iff the path system has a solution. Hence, the PTIME-hardness result follows for the regularity problem. The reader is referred to (Howell, Rosier, and Yen, 1987) for details. ∎

THEOREM 4.21. *The regularity problem for trap-circuit (extended trap-circuit, normal, and sinkless) PNs is NP-hard.*

Proof (Sketch). In (Howell, Rosier, and Yen, 1987), 3-SAT (a known NP-complete problem) was shown to be reducible to the reachability problem for trap-circuit, (normal, and sinkless) PNs. Again, the constructed PN (from a given 3-SAT instance) has a place p such that the 3-SAT instance has a solution iff a marking μ with $\mu(p) > 0$ is reachable. The rest of the proof is similar to that of Theorem 4.20. ∎

THEOREM 4.22. *The regularity problem for BPP-nets is NL-hard.*

Proof (Sketch). This can be done by reducing from the *graph accessibility problem*, which is known to be NL-complete (Hopcroft and Ullman, 1979). ∎

THEOREM 4.23. *The regularity problem for general PNs is EXPSPACE-hard.*

Proof (Sketch). Similar to the lower bound proof of the reachability problem for general PNs (Lipton, 1976). ∎

ACKNOWLEDGMENTS

The author thanks the anonymous referees for their comments and suggestions, which greatly improved the correctness as well as the presentation of this paper.

Received August 24, 1993; final manuscript received September 7, 1995

REFERENCES

Agerwala, T., and Flynn, M. (1973), Comments on capabilities, limitations and 'correctness' of Petri nets, *in* "Proceedings, 1st Annual Symposium on Computer Architecture," pp. 81–86.

Christensen, S., Hirshfeld, Y., and Moller, F. (1993), Bisimulation equivalence is decidable for basic parallel processes, *in* "Proceedings, 4th International Conference on Concurrency Theory," pp. 143–157.

Esparza, J. (1992), A solution to the covering problem for 1-bounded conflict-free Petri nets, *Inform. Process. Lett.* **41**, 313–319.

Esparza, J. (1994), "On the Uniform Word Problem for Commutative Context-Free Grammars," Technical Report, Laboratory of Foundations of Computer Science, University of Edinburgh.

Ginzburg, A., and Yoeli, M. (1980), Vector addition systems and regular languages, *J. Comput. System Sci.* **20**, 277–284.

Hopcroft, J., and Ullman, J. (1979), "Introduction to Automata Theory, Languages, and Computation," Addison–Wesley, Reading, MA.

Howell, R., and Rosier, L. (1988), Completeness results for conflict-free vector replacement systems, *J. Comput. System Sci.* **37**, 349–366.

Howell, R., Rosier, L., and Yen, H. (1987), An $O(n^{1.5})$ algorithm to decide boundedness for conflict-free vector replacement systems, *Inform. Process. Lett.* **25**, 27–33.

Howell, R., Rosier, L., and Yen, H. (1993), Normal and sinkless Petri nets, *J. Comput. System Sci.* **46**, 1–26.

Huynh, D. (1983), Commutative grammars: The complexity of uniform word problems, *Inform. and Control* **57**, 21–39.

Ichikawa, A., and Hiraishi, K. (1987), Analysis and control of discrete event systems represented by Petri nets, *in* "Lecture Notes in Control and Information Science," Vol. 103, pp. 115–134, Springer-Verlag, Berlin/New York.

Jantzen, M., and Petersen, H. (1994), Cancellation in context-free languages: Enrichment by induction, *Theoret. Comput. Sci.* **127**, 149–170.

Jones, N., Landweber, L., and Lien, Y. (1977), Complexity of some problems in Petri nets, *Theoret. Comput. Sci.* **4**, 277–299.

Khachian, L. (1979), A polynomial algorithm for linear programming, *Dokl. Akad. Nauk USSR* **244**, 1093–1096, translated in *Sov. Math. Dokl.* **20**, 191–194.

Kosaraju, R. (1973), Limitations of Dijkstra's semaphore primitives and Petri nets, *Oper. Systems Rev.* **7**, 122–126.

Landweber, L., and Robertson, E. (1978), Properties of conflict-free and persistent Petri nets, *J. Assoc. Comput. Mach.* **25**, 352–364.

Lenstra, H., Jr. (1983), Integer programming with a fixed number of variables, *Math. Oper. Res.* **8**, 538–548.

Lipton, R. (1976), "The Reachability Problem Requires Exponential Space," Technical Report 62, Dept. of Computer Science, Yale University.

Murata, T. (1989), Petri nets: Properties, analysis and applications, *Proc. IEEE* **77**, 541–580.

Peterson, J. (1981), "Petri Net Theory and the Modeling of Systems," Prentice–Hall International, Englewood Cliffs, NJ.

Reisig, W. (1985), "Petri Nets: An Introduction," Springer-Verlag, Heidelberg.

Schwer, S. (1986), On the rationality of Petri net languages, *Inform. Process. Lett.* **22**, 145–146.

Schwer, S. (1992a), Fine covers of a VAS language, *Theoret. Comput. Sci.* **95**, 159–168.

Schwer, S. (1992b), The context-freeness of the languages associated with vector addition systems is decidable, *Theoret. Comput. Sci.* **98**, 199–247.

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

Yamasaki, H. (1984), Normal Petri nets, *Theoret. Comput. Sci.* **31**, 307–315.

Yen, H. (1991), A polynomial time algorithm to decide pairwise concurrency of transitions for 1-bounded conflict-free Petri nets, *Inform. Process. Lett.* **38**, 71–76.

Yen, H. (1992), A unified approach for deciding the existence of certain Petri net paths, *Inform. and Comput.* **96**, 119–137.