INFORMATION AND COMPUTATION 96, 119–137 (1992)

A Unified Approach for Deciding the Existence of Certain Petri Net Paths

HSU-CHUN YEN

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, Republic of China

In this paper, we develop a unified approach for deriving complexity results for a number of Petri net problems. We first define a class of formulas for paths in Petri nets. We then show that the satisfiability problem for our formulas is EXPSPACE complete. Since a wide range of Petri net problems can be reduced to the satisfiability problem in a straightforward manner, our approach offers an umbrella under which many Petri net problems can be shown to be solvable in EXPSPACE.

1. INTRODUCTION

Petri nets provide an elegant and useful tool for modeling concurrent systems. In many applications, however, modeling by itself is of little practical use if one cannot analyze the modeled system. Despite the efforts made by many researchers, many analytical questions concerning Petri nets remain unanswered. In fact, the decidability of one of the most important questions concerning Petri nets, the reachability problem, was left unsolved for a long period of time until Mayr (1984) finally provided an affirmative answer to the question just a few years ago. (The precise complexity of the reachability problem, however, has not yet been established.) Due to the degree of difficulty involved in many of the Petri net problems, it is desirable to have unified and systematic approaches to reason about Petri nets. Developing such a mechanism is exactly the goal of this paper.

We first define a class of formulas for paths in Petri nets. What makes this class of formulas useful is that it is powerful enough to express many Petri net properties. We also show that the satisfiability problem, i.e., the problem of determining whether there exists a path in a given Petri net satisfying a given formula, is complete for EXPSPACE. By reducing to the satisfiability problem, we are able to derive EXPSPACE upper bounds for a number of Petri net problems (some of which were previously unsolved) in a straightforward and unified manner. In what follows, we first review

0890-5401/92 $3.00

some of the existing strategies known to be useful for examining the decidability and complexity of Petri net problems.

A useful tool for showing the decidability of Petri net problems is based on Karp–Miller coverability graph analysis (Karp and Miller, 1969). A coverability graph is a generalized reachability graph in which each potentially unbounded place is represented by a special symbol "ω." It has been shown by Karp and Miller (1969) that the coverability graph of any vector addition system (equivalently Petri net) is finite. As a result, a vector addition system (Petri net) is unbounded iff an ω occurs in its coverability graph. This technique has subsequently been used in a number of places for showing the decidability of Petri net problems. (See, e.g., Finkel, 1987; Ginzburg and Yoeli, 1980; Suzuki and Kasami, 1983; Valk and Jantzen, 1985; Valk and Vidal–Naquet, 1982.) The key deficiency of the coverability graph approach lies in its inability to produce complexity bounds. This is because the size of the coverability graph of a Petri net, in general, is not primitive recursive. As a result, the whole strategy is based on an unbounded search.

Semilinearity has played a crucial role in the analysis of Petri nets. It is a common belief that the difficulties involved in obtaining results for general Petri nets lie in the fact that the reachability sets for Petri nets of dimension six or more are not in general semilinear (Hopcroft and Pansiot, 1979). However, there are restricted subclasses of Petri nets for which the reachability sets are semilinear. Utilizing results concerning semilinear sets and Presburger arithmetic, one can prove the decidability of the containment, equivalence, and reachability problems for a restricted subclass of Petri nets by demonstrating that the subclass exhibits semilinearity. In fact, a large body of results appearing in the literature rely exactly upon this. These include the decidability proofs of the containment, equivalence, and reachability problems for conflict-free (Crespi–Reghizzi and Mandrioli, (1975), persistent (Landweber and Robertson, 1978; Mayr, 1981; Muller, 1980), weakly persistent (Yamasaki, 1981), normal (Yamasaki, 1984), sinkless (Yamasaki, 1984), 3-dimensional (van Leeuwen, 1974), 5-dimensional (Hopcroft and Pansiot, 1979), symmetric (Araki and Kasami, 1977), and regular (Ginzburg and Yoeli, 1980; Valk and Vidal–Naquet, 1982) Petri nets. A drawback in using this approach is that the sizes of the semilinear set representations for some of the above classes were not known. Hence, no complexity bounds could be derive for those classes as a result of their being semilinear. Recently, however, some efforts have been successfully made toward analyzing the size of the semilinear set representation for some of the above subclasses of Petri nets, thus yielding complexities for equivalence, containment, and reachability. (See, e.g., Howell and Rosier, 1988; Howell, Rosier, and Yen, 1989; Howell, Rosier, Huynh, and Yen, 1986; Huynh, 1985.) A somewhat related approach, based on the notion of "residue sets," has been used by Valk and Jantzen (1985) to show the decidability of some Petri net problems. (See Valk and Jantzen (1985) for more details.) This approach, again, reveals no complexity bounds.

An important cornerstone in the complexity analysis of Petri nets is the EXPSPACE[1] upper bound of the boundedness problem shown by Rackoff (1978). This together with an earlier result by Lipton (1976) that the problem requires EXPSPACE, provides a near optimal solution for one of the most important Petri net problems. Rackoff's paper is significant not only because of the result itself, but also and perhaps more importantly because it provides a technique for deriving complexities for other Petri net related problems. Such applications include the proofs of the complexity of the equivalence problem for commutative semigroups and symmetric vector addition systems (Huynh, 1985), the equivalence between 2-way nondeterministic multihead finite automata and 2-way nondeterministic weak counter machines as well as a hierarchy theorem for 2-way nondeterministic weak counter machines (Chan, 1987), a detailed multiparameter analysis of the boundedness problem for vector addition systems (Rosier and Yen, 1986), and the complexity of the model checking problem for systems with many identical processes (Sistla and German, 1987).

Since reachability and boundedness are known to be decidable, a natural way to show the decidability of an unknown problem is to reduce the unknown to reachability (or boundedness). In practice, however, this approach suffers from a drawback that many such reductions rely on rather complicated (and *ad hoc*) Petri net constructions which are hard to understand. This is perhaps because the notion of boundedness (or reachability), as a reduction tool, is too weak to "model" or "simulate" other properties of Petri nets. In an attempt to avoid using "brute force" reductions, a temporal logic for Petri nets was introduced by Howell, Rosier, and Yen (1991) for which the model checking problem has been shown to be equivalent to reachability. In addition, a number of fairness related problems (fair nontermination problems, more precisely) concerning Petri nets have been reduced to the model checking problem. As a result, temporal logic provides an "umbrella" under which certain fair nontermination problems can be shown to be decidable. (See (Jancar, 1989) for a similar approach for showing the decidability of the nontermination problem with respect to finite-delay property.) In view of the above, the following question naturally arises: Is there a similar umbrella under which certain Petri net problems can be shown to be equivalent to boundedness?

[1] $\text{EXPSPACE} = \bigcup_{c>0} \text{NSPACE}(2^{n^c})$, where $\text{NSPACE}(f(n))$ is the class of languages accepted by nondeterministic Turing machines using at most $f(n)$ space.

As an attempt to answer the above question, in this paper we first define a class of Petri net path formulas, each of which is of the form

$$\exists \mu_1, \mu_2, ..., \mu_k \exists \sigma_1, \sigma_2, ..., \sigma_k (\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2 \cdots \xrightarrow{\sigma_k} \mu_k)$$
$$\wedge F(\mu_1, ..., \mu_k, \sigma_1, ..., \sigma_k),$$

meaning that marking μ_i can be reached from μ_{i-1} $(1 \leqslant i \leqslant k)$ through the firing of transition sequence σ_i and predicate $F(\mu_1, ..., \mu_k, \sigma_1, ..., \sigma_k)$ holds. What makes this class of formulas useful is that it is powerful enough to express many Petri net properties. For example, the unboundedness condition can be expressed as $\exists \mu_1, \mu_2 \exists \sigma_1, \sigma_2 (\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge (\mu_2 > \mu_1)$. By augmenting the proof of Rackoff (1978), we then show that if a formula is satisfiable, then there exists a short "witness" (path) whose length is bounded by $O(2^{2^{c*n*\log n}})$, for some constant c. As a result, the satisfiability problem can be solved in $O(2^{d*n*\log n})$ space, for some constant d. Using this result, we are able to provide simple and unified proofs for many known results, including the complexities of the boundedness, covering, and some fair nontermination problems, which would otherwise require more complicated arguments. More importantly, it also allows us to show EXPSPACE upper bounds for many unsolved problems—the regularity detection problem (of Ginzburg and Yoeli (1980) and Valk and Vidal-Naquet (1982)), for example. Another contribution of our result is that it explains why so many Petri net problems, even though bearing little similarity on the surface, possess the same complexity bound.

The remainder of this paper is organized as follows. In Section 2, we give the basic definitions of Petri nets and a class of predicates for Petri nets. In Section 3, we show that the satisfiability problem is solvable in $O(2^{d*n*\log n})$ space, for some constant d. In Section 4, we show that many Petri net problems, some of which were previously unsolved, can be reduced to the satisfiability problem. Thus, our main result may therefore be viewed as an umbrella under which the subsequent results in this paper are derived.

2. A Class of Predicates for Paths in Petri Nets

Let N (respectively, Z and R) be the set of nonnegative integers (respectively, integers and rational numbers), and let N^k (respectively, Z^k) be the set of vectors of k nonnegative integers (respectively, integers). A *Petri net* is a tuple (P, T, φ, μ_0), where P is a finite set of *places*, T is a finite set of *transitions*, φ is a *flow function* $\varphi: (P \times T) \cup (T \times P) \to N$, and μ_0 is the *initial marking* $\mu_0: P \to N$. A *marking* is a mapping $\mu: P \to N$. By establishing an order on P and T, i.e., $P = \{p_1, ..., p_k\}$ and $T = \{t_1, ..., t_r\}$, we can think of a marking μ as a vector in N^k, where the ith component represents $\mu(p_i)$. (For convenience, $\mu(p_i)$ will be abbreviated as $\mu(i)$.) We also define the *transition vector* of a transition t, denoted by $\bar{t}$, to be a k-dimensional vector in Z^k such that $\bar{t}(i) = \varphi(t, p_i) - \varphi(p_i, t)$, and the set of transition vectors, denoted by $\bar{T}$, to be $\{\bar{t} \mid t \in T\}$. 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 $\mu \xrightarrow{t} \mu'$, where $\mu'(p) = \mu(p) - \varphi(p, t) + \varphi(t, p)$ for all $p \in P$. A sequence of transitions $\sigma = t_1 \cdots t_n$, $n > 0$, is a *firing sequence* from μ_0 iff $\mu_0 \xrightarrow{t_1} \mu_1 \xrightarrow{t_2} \cdots \xrightarrow{t_n} \mu_n$ for some sequence of markings $\mu_1, ..., \mu_n$. We also write $\mu_0 \xrightarrow{\sigma} \mu_n$. (We sometimes write $\mu_0 \xrightarrow{\sigma}$ if μ_n is not important.) A marking μ_n is said to be *reachable* from μ_0, denoted by $\mu_0 \xrightarrow{*} \mu_n$, iff $\mu_n = \mu_0$ or $\exists \sigma$ such that $\mu_0 \xrightarrow{\sigma} \mu_n$. We let $R(P, T, \varphi, \mu_0) = \{\mu \mid \mu_0 \xrightarrow{*} \mu\}$ denote the set of all reachable markings. Let σ be a sequence of transitions. We define $\#_\sigma$ to be a mapping from T to N such that $\#_\sigma(t_i)$ = the number of occurrences of t_i in σ. $\#_\sigma$ can be viewed as a vector in N^r whose ith component is $\#_\sigma(t_i)$.

To deal with the complexity issue, it is necessary to define the *size* of a Petri net in a precise manner. Throughout this paper, each integer will be represented by its binary representation. The *length* of an integer is the number of bits of its binary representation. The size of a set (or vector) of integers is defined to be the sum of the lengths of the components. Consider a Petri net $\mathscr{P} = (P, T, \varphi, \mu_0)$, where $P = \{p_1, ..., p_k\}$ and $T = \{t_1, ..., t_r\}$. Each transition $\varphi(p_i, t_j) = m$ ($\varphi(t_j, p_i) = m$) can be thought of as a four-tuple $(0, i, j, m)$ ($(1, j, i, m)$). (The first component (0 or 1) is to indicate the flow direction (0: from a place to a transition; 1: from a transition to a place). In this way, φ can be treated as a set of four tuples. Now the size of Petri net $\mathscr{P}$ can be defined as $\lceil \log k \rceil + \lceil \log r \rceil +$ the sum of the sizes of elements in $\varphi +$ the size of μ_0. Since the binary representation is used, the firing of a transition may result in removing (or adding) 2^n tokens from (to) a place, where n is the size of the Petri net.

In what follows, we present a class of Petri net path formulas. In Section 3, the satisfiability problem is shown to be solvable in EXPSPACE. Let (P, T, φ, μ_0) be a k-place r-transition Petri net. 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) $\in Z^r$, $c \in N$, and $\odot$ denotes the inner product (i.e., $(a_1, a_2, ..., a_k) \odot (b_1, b_2, ..., b_k) = \sum_{i=1}^{k} a_i * b_i$). It is worth mentioning that predicates of the form $\#_{\sigma_i}(t_j)/\#_{\sigma_i}(t_l) = a/b$, where $a, b \in N$ $(b \neq 0)$, $i > 1$ and $t_j, t_l \in T$, can be reduced to the predicates described above. To see this, let $c = 0$ and let y be a vector such that $y(j) = b$, $y(l) = -a$, $y(p) = 0$, $\forall p \notin \{j, l\}$. Then $y \odot \#_{\sigma_i} = c$ iff $b * \#_{\sigma_i}(t_j) - a * \#_{\sigma_i}(t_l) = 0$, which is equivalent to $\#_{\sigma_i}(t_j)/\#_{\sigma_i}(t_l) = a/b$.

• $\#_{\sigma_1}(t_j) \leqslant c$ 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 is equal to, less than, and 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. Let F be a predicate and D a set of positive integers. We define $F^{[D]}$ to be the predicate resulting from removing type 1 atomic marking predicates of the form $\mu(i) \geqslant c$ and $\mu(i) > c$ from F, for all $i \notin D$.

In this paper, we deal with formulas f of the following form (with respect to Petri net (P, T, φ, μ_0)):

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

Given a Petri net $\mathscr{P}$ and a formula f, we use $\mathscr{P} \models f$ to denote that f is true in $\mathscr{P}$. The *satisfiability problem* (for $F(\mu_1, ..., \mu_m, \sigma_1, ..., \sigma_m)$) is the problem of determining, given a Petri net $\mathscr{P}$ and a formula f, whether $\mathscr{P} \models f$. If $p = \mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \cdots \mu_{m-1} \xrightarrow{\sigma_m} \mu_m$ is a path, such that $F(\mu_1, ..., \mu_m, \sigma_1, ..., \sigma_m)$ is true in $\mathscr{P}$, we say p satisfies F.

3. The Main Result

In this section, we show that the satisfiability problem is solvable in $O(2^{d * n * \log n})$ space in the size of the Petri net and the formula (i.e., n), for some constant d. We first show that, given a Petri net $\mathscr{P}$ and a formula f, the length of the shortest path satisfying F, if exists, is bounded by $O(2^{2^{c*n*\log n}})$, for some constant c. The upper bound then follows immediately from a nondeterministic search. In our proof, we generalize an induction strategy originated by Rackoff (1978). In order to do so, we need a few lemmas first.

We begin by showing that our predicates satisfy the so-called "monotonic" property, which is essential in the proof of the main theorem.

LEMMA 3.1. *Let* Δ $(\in Z^k)$ *be a vector whose first* i *components are nonnegative (i.e.,* $\forall j,\ 1 \leqslant j \leqslant i,\ \Delta(j) \geqslant 0$*). For every predicate* F*, if* $F(\mu_1, ..., \mu_m, \sigma_1, ..., \sigma_m) = true$*, then* $F^{[\{1,2,...,i\}]}(\mu_1 + \Delta, ..., \mu_m + \Delta, \sigma_1, ..., \sigma_m) = true$*.*

Proof. Clearly, it is sufficient to consider only marking predicates. It should also be noted that $F^{[1,2,...,i]}$ does not contain type 1 marking predicates of the form $\mu(j) \geqslant c$ and $\mu(j) > c$, for $j > i$. Hence, adding Δ (whose first i components are nonnegative) has no effect on any type 1 marking predicate in $F^{[1,2,...,i]}$. Also note that each type 2 marking predicate is built from terms of the form $\mu_j - \mu_i$ or constants. As a result, adding Δ to all marking variables has no effect on any term. Thus, $F(\mu_1, ..., \mu_m, \sigma_1, ..., \sigma_m) \Rightarrow F^{[\{1,2,...,i\}]}(\mu_1 + \Delta, ..., \mu_m + \Delta, \sigma_1, ..., \sigma_m)$. ∎

The following lemma indicates that in deciding satisfiability, it is sufficient to consider formulas consisting of marking predicates only.

LEMMA 3.2. *Given a Petri net* $\mathscr{P} = (P, T, \varphi, \mu_0)$ *and a formula* f*, we can construct, in polynomial time, a Petri net* $\mathscr{P}' = (P', T', \varphi', \mu_0')$ *and a formula* f' *containing no transition predicates such that* $\mathscr{P} \models f$ *iff* $\mathscr{P}' \models f'$*.*

Proof. In what follows, we show how each transition predicate can be transformed into a marking predicate, while the same satisfiability status is maintained. Let $\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \cdots \mu_{m-1} \xrightarrow{\sigma_m} \mu_m$ be a path in $\mathscr{P}$ satisfying F.

1. $y \odot \#_{\sigma_i} < (=, >) c$, where $i > 1$. $\mathscr{P}'$ is constructed from $\mathscr{P}$ in the following way. Let $\mathscr{P}'$ contain all transitions and places of $\mathscr{P}$. In addition, two new places s^+ and s^- are added to $\mathscr{P}'$ such that $\varphi(t_j, s^+) = y(j)$ if $y(j) > 0$ and $\varphi(t_j, s^-) = -y(j)$ if $y(j) < 0$. Initially, s^+ and s^- are empty. Now consider $\mu_0' \xrightarrow{\sigma_1} \mu_1' \xrightarrow{\sigma_2} \cdots \mu_{m-1}' \xrightarrow{\sigma_m} \mu_m'$ in $\mathscr{P}'$. It is reasonably easy to see that $y \odot \#_{\sigma_i} < (=, >) c$ iff $(\mu_i' - \mu_{i-1}')(s^+) - (\mu_i' - \mu_{i-1}')(s^-) < (=, >) c$.

2. $\#_{\sigma_1}(t_j) \geqslant (\leqslant) c$. The "$\geqslant$" part can easily be shown by adding a new place s_j such that $\varphi(t_j, s_j) = 1$ and $\mu_0'(s_j) = 0$. Clearly, $\#_{\sigma_1}(t_j) \geqslant c$ iff $\mu_1'(s_j) \geqslant c$. The $\#_{\sigma_1}(t_j) \leqslant c$ case is a bit involved. $\mathscr{P}'$ is constructed from $\mathscr{P}$

in a way described in Fig. 1. In σ'_1, t'_j is used to simulate t_j. Initially, s_j and s'_j, and s''_j contain c, 1, and 0 tokens, respectively. Note that t'_j is controlled by s_j and s'_j, while t_j is controlled by s''_j. Since σ_1 consists of at most c occurrences of t_j, this can be simulated by t'_j in $\mathscr{P}'$. (s_j assures that t'_j will fire at most c times.) Now consider $\mu'_0 \xrightarrow{\sigma'_1} \mu'_1 \xrightarrow{r\sigma_2} \cdots \mu'_{m-1} \xrightarrow{\sigma_m} \mu'_m$ in $\mathscr{P}'$. (σ'_1 is obtained from σ_1 by replacing each occurrence of t_j by a t'_j.) (Note that after the completion of σ'_1, a token is moved from s'_j to s''_j.) It is not hard to see that $\#_{\sigma_1}(t_j) \leqslant c$ in $\mathscr{P}$ iff $\mu'_1(s'_j) \geqslant 1$ in $\mathscr{P}'$.

In view of the above, transition predicates can always be "simulated" by marking predicates. This completes the proof. ▌

In what follows, we show that given a Petri net $\mathscr{P} = (P, T, \varphi, \mu_0)$ and a formula f, the length of the shortest path satisfying F, if exists, is bounded by $O(2^{2^{d \cdot n \cdot \log n}})$, for some constant d. Without loss of generality, we assume that F contains only marking predicates (Lemma 3.2). Furthermore, because $F_1 \vee F_2$ is satisfiable iff F_1 is satisfiable or F_2 is satisfiable we can assume that F is of the normal form $F = F_1 \wedge F_2$, where $F_1 = h_1 \wedge \cdots \wedge h_p$ and $F_2 = g_1 \wedge \cdots \wedge g_q$ and h_i and g_i are atomic predicates of types 1 and 2, respectively. Before going into details, we require some definitions. Most of them are the same as in (Rackoff, 1978) (see also (Rosier and Yen, 1986). A *generalized marking* is a mapping $\mu: P \to Z$. A *generalized firing sequence* is any sequence of transitions. A finite sequence of vectors $w_1, w_2, ..., w_m \in Z^k$ is said to be a *path* (of length $m-1$) in P if $w_1 = \mu_0$ and $w_{i+1} - w_i \in \bar{T}$ (the set of transition vectors), for all i, $1 \leqslant i < m$. (We sometimes use $w_1 \xrightarrow{t_1} w_2 \cdots \xrightarrow{t_{m-1}} w_m$, where $\bar{t}_i = w_{i+1} - w_i$, to represent a path if the associated (generalized) firing sequence is important.) Let $w \in Z^k$ and $0 \leqslant i \leqslant k$. The vector w is *i bounded* if $w(j) \geqslant 0$ for $1 \leqslant j \leqslant i$. If $r \in N^+$ is such that $0 \leqslant w(j) < r$ for $1 \leqslant j \leqslant i$, the w is called *i–r bounded.* Let $p = w_1, w_2, ..., w_m$ be a sequence of vectors, we say p is *i bounded* (*i–r*

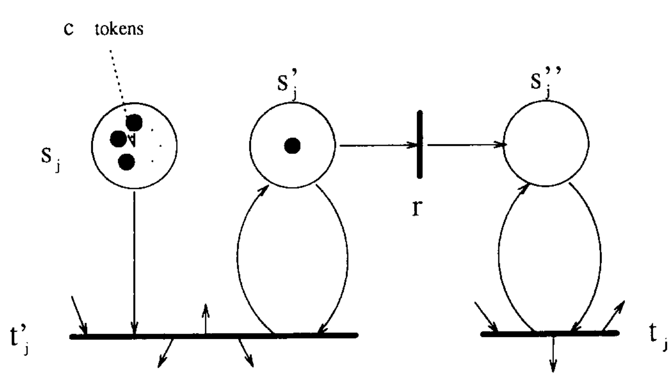

FIG. 1. A Petri net for simulating $\#_{\sigma_1}(t_j) \leqslant c$.

bounded) if every member in p is i bounded (i–r bounded). p is called an *i loop* if in the first i places $w_1 = w_m$ and $w_{j1} \neq w_{j2}$ for all $1 \leqslant j_1 < j_2 \leqslant m$; i.e., p is a path such that the start and end vectors have their first i components identical and no other intermediate points have this property. The *loop value* of p is defined to be $w_m - w_1$. Given a predicate $F(\mu_1, \mu_2, ..., \mu_p)$, an i bounded (i–r bounded) path $w_1, w_2, ..., w_m$ is called an i bounded (i–r bounded) *F-path* if $\exists 1 \leqslant j_1 \leqslant j_2 \leqslant \cdots \leqslant j_p \leqslant m$ such that $F^{[\{1, 2, ..., i\}]}(w_{j_1}, w_{j_2}, ..., w_{j_p})$ is true. Let $m'(i, \mu, F)$ be the length of the shortest i bounded F-path whose initial generalized marking is μ. (If no such path exists, then $m'(i, \mu, F) = 0$.) Let $g(i, F) = \max\{m'(i, \mu, F) \mid \mu \in Z^k\}$. In what follows, we argue that $g(i, F) \in N$. First note that function m' is *monotonic with respect to* μ in the sense that if an i bounded path p satisfying F exists for a marking μ, then p is guaranteed to satisfy F for the marking $\mu + \Delta$, for any $\Delta \geqslant 0$ (Lemma 3.1). This implies $m'(i, \mu + \Delta, F) \leqslant m'(i, \mu, F)$, for any $\Delta \geqslant 0$. As a result, if we let $S(i, F) = \{\mu \mid m'(i, \mu, F) > 0\}$ (i.e., the set of all markings from which i bounded paths satisfying F exist), then $S(i, F)$ is *right-closed.* (A set $M \subseteq N^k$ is right-closed iff $v \in M \Rightarrow \forall v' \geqslant v,\ v' \in M$. See (Valk and Jantzen, 1985) for more about basic properties of right-closed sets.) It is well-known that given a right-closed set, its set of minimal elements is finite. Let $S'(i, F)$ be the set of minimal elements of $S(i, F)$. Then $g(i, F) = \max\{m'(i, \mu, F) \mid \mu \in Z^k\} = \max\{m'(i, \mu, F) \mid \mu \in S'(i, F)\}$, which is finite. (Recall that $m'(i, \mu + \Delta, F) \leqslant m'(i, \mu, F)$, for any $\Delta \geqslant 0$.) It is worth mentioning here that $g(i, F)$ does not depend on the starting marking.

Given a vector y, we let $\|y\|$ be the maximum absolute value of y's components. To prove our upper bound result, we need the following lemma concerning the bounds of solutions of linear equations. The lemma is from (Rackoff, 1978). (The proof is essentially from (Borosh and Treybis, 1976).)

LEMMA 3.3. *Let $d_1, d_2 \in N^+$, let B be a $d_1 \times d_2$ integer matrix, and let b be a $d_1 \times 1$ matrix. Let $d \geqslant d_2$ be an upper bound on the absolute values of the integers in B and d. If there exists a vector $v \in N^{d_2}$ which is a solution to $Bv \geqslant b$, then for some constant c independent of d, d_1, d_2, there exists a vector $v \in N^{d_2}$ such that $Bv \geqslant b$ and $v(i) < d^{c * d_1}$ for all i, $1 \leqslant i \leqslant d_2$.*

We are now in a position to show that, given a Petri net $\mathscr{P}$ and a formula F, if there is a path satisfying formula F, then there must exist a short "witness" whose length is bounded by $O(2^{2^{d * n * \log n}})$. The proof closely parallels the induction strategy used by Rackoff (1978).

LEMMA 3.4. *Let $\mu_{j,i}$, $j > i$, denote $\mu_j - \mu_i$. Then each type 2 atomic marking predicate can be expressed as $n_2\mu_{2,1}(i) + n_3\mu_{3,2}(i) + \cdots +$*

$n_m\mu_{m,m-1}(i)+n'_2\mu_{2,1}(j)+n'_3\mu_{3,2}(j)+\cdots+n'_m\mu_{m,m-1}(j)<(=,>)c$ for some constants $n_2, ..., n_m, n'_2, ..., n'_m$, c ($\in Z$) whose absolute values are $\leqslant 2^{O(n)}$ *(n is the combined size of the Petri net and the formula).*

Proof. In what follows, we only consider the "<" case; the other two cases are analogous. First note that each type 2 marking predicate of the form $T_1(i)<T_2(j)$, by definition, can be expressed as

$$\left(\sum_{l=1}^{f} d_l(\mu_{p(l)}-\mu_{q(l)})+d\right)(i)<\left(\sum_{t=1}^{g} e_t(\mu_{r(t)}-\mu_{s(t)})+e\right)(j),$$

where $p(l)>q(l)$, $r(t)>s(t)$, $0\leqslant f, g\leqslant n$, and $d, d_1, ..., d_f, e, e_1, ..., e_g$ are constants (each of which has its absolute value $\leqslant 2^{O(n)}$ since binary representation is used for integers). When each occurrence of $\mu_j-\mu_i$ is replaced by its equivalence $\mu_{j,j-1}+\mu_{j-1,j-2}+\cdots+\mu_{i+1,i}$, for $j>i$, the above inequality can further be simplified as

$$\left(\sum_{l=1}^{m-1} n_{l+1}\mu_{(l+1),l}\right)(i)+\left(\sum_{t=1}^{m-1} n'_{t+1}\mu_{(t+1),t}\right)(j)<c,$$

where $n_2, ..., n_m, n'_2, ..., n'_m, c$ are (possibly negative) constants whose absolute values are $\leqslant 2^{O(n)}$. This completes the proof. ∎

LEMMA 3.5. *If there is an i–r bounded F-path in Petri net (P, T, φ, μ), then there is an i–r bounded F-path of length $\leqslant r^{n^c}$, for some constant c independent of r and n.*

Proof. The proof is similar to (but more involved than) the corresponding one in (Rackoff, 1978). Let $\mu \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \cdots \mu_{m-1} \xrightarrow{\sigma_m} \mu_m$ be an i–r bounded F-path. First note that $\mu \xrightarrow{\sigma_1} \mu_1$ need not be longer than r^k; otherwise, there must exist an i loop which can be removed without affecting the validity of F. (Such a loop can conceptually be removed by adding $-\Delta$ (Δ = loop value) to $\mu_1, ..., \mu_m$. According to Lemma 3.1, the validity of F remains intact.)

Now consider $\mu_1 \xrightarrow{\sigma_2} \cdots \mu_{m-1} \xrightarrow{\sigma_m} \mu_m$. In segment δ_h: $\mu_{h-1} \xrightarrow{\sigma_h} \mu_h$, $2\leqslant h\leqslant m$, suppose Q is an i loop in marking q (i.e., starting in q and ending in q). We have the following important property:

- the new path resulting from repeating Q t times, for an arbitrary $t\geqslant 0$, in q is still an i–r bounded path satisfying $F_1^{[\{1,2,...,i\}]}$. (Recall that F_1 consists of only type 1 predicates.) Note that "$t=0$" corresponds to the removal of i loop Q from marking q.

The above property ensures that i loops can be removed or repeated (an arbitrary number of times) in any marking without affecting the satisfiability status. In what follows, we will show how to remove, duplicate, and rearrange existing i loops so as to shorten the existing path, while preserving the validity of $F_2^{[\{1, 2, ..., i\}]}$.

We decompose δ_h into a path s_h and some i loops such that

- the length of s_h is $\leqslant (r^k + 1)^2$,
- the length of each loop is $\leqslant r^k$,
- each place of a loop value is $\leqslant 2^n * r^k$, and
- the total number of distinct loop values is $\leqslant (2 * (2^n * r^k) + 1)^k$.

(The reason that such a decomposition exists can be found in (Rackoff, 1978). (See also (Rosier and Yen, 1986).))

Let $l_1^h, ..., l_{p_h}^h$ $(2 \leqslant h \leqslant m,\ p_h \leqslant (2 * (2^n * r^k) + 1)^k)$ be the distinct loop values in segment δ_h. Note that $\mu_h - \mu_{h-1} = ([s_h] + a_1^h l_1^h + \cdots + a_{p_h}^h l_{p_h}^h)$, for some $a_1^h, ..., a_{p_h}^h$ $(\in N)$, and $[s_h]$ $(\in Z^k)$ is the difference of the end and start points of the path s_h. Now according to Lemma 3.4, each atomic predicate of type 2 can be represented as

$$\begin{aligned} & n_2 * ([s_2] + a_1^2 l_1^2 + \cdots + a_{p_2}^2 l_{p_2}^2)(i) + \cdots \\ & + n_m * ([s_m] + a_1^m l_1^m + \cdots + a_{p_m}^m l_{p_m}^m)(i) \\ & + n_2' * ([s_2] + a_1^2 l_1^2 + \cdots + a_{p_2}^2 l_{p_2}^2)(j) + \cdots \\ & + n_m' * ([s_m] + a_1^m l_1^m + \cdots + a_{p_m}^m l_{p_m}^m)(j) < (=, >) e, \end{aligned}$$

where $e, n_h, n_h' \leqslant 2^n$, $\|[s_h]\|$, $\|l_q^h\| \leqslant d * 2^n * (r^k + 1)^2$, $p_h \leqslant (2 * (2^n * r^k) + 1)^k$ $(2 \leqslant h \leqslant m)$, and d is a constant. Consequently, $F_2^{[\{1, ..., i\}]}$ can be expressed as (no more than) n such (un)equations. To find a shorter path satisfying $F_2^{[\{1, ..., i\}]}$, it suffices to solve the associated system of (un)equations (with respect to variables $a_1^h, ..., a_{p_h}^h$, $2 \leqslant h \leqslant m$, $p_h \leqslant (2 * (2^n * r^k) + 1)^k$). By letting $d_1 = n$ and $d = r^{3n^2}$ and using Lemma 3.3, we are able to find solutions whose values are $\leqslant r^{n^{c'}}$, for some constant c'. As a result, there exists a "short" i–r bounded F-path whose length is no more than r^{n^c}, for some constant c independent of r and n. ∎

In what follows, we derive $g(i, F)$ recursively.

LEMMA 3.6. *$g(0, F) \leqslant 2^{n^c}$, for some constant c independent of n.*

Proof. Let $\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_m} \mu_m$ be a 0-bounded path satisfying F. Clearly, it is sufficient to consider F_2. Suppose $F_2 = f_1 \wedge f_2 \wedge \cdots \wedge f_z$. Let

$\bar{T} = \{v_1, ..., v_q\}$ be the set of transition vectors in the Petri net. According to Lemma 3.4, each f_r, $1 \leqslant r \leqslant z$, can be represented as

(1) $n_2^r \mu_{2,1}(i) + n_3^r \mu_{3,2}(i) + \cdots + n_m^r \mu_{m,m-1}(i) < (=, >) e^r$, for some constants $n_2^r, ..., n_m^r, e^r$ whose absolute values are $\leqslant 2^n$. Furthermore,

(2) $\mu_{l,l-1} = p_1^l v_1 + p_2^l v_2 + \cdots + p_q^l v_q$, for some $p_1^l, ..., p_q^l \in N$.

Substituting each $\mu_{l,l-1}$ in (1) by (2) will yield a linear equation with respect to variables $p_1^l, ..., p_q^l$, $\forall 1 \leqslant l \leqslant m$. As a result, F_2 can be expressed as a system of at most n such equations. Choose $d_1 = n$ and $d = 2^{2*n}$. Using Lemma 3.4, we can find a solution with each $p_j^l \leqslant 2^{n^{c''}}$, for some constant c''. Clearly then the lemma follows. ▮

LEMMA 3.7. *$g(i+1, F) \leqslant (2^n(g(i, F)+1))^{n^c}$ for all $i < k$, where c is a constant independent of n.*

Proof. Case 1. If there is an $(i+1) - (2^n(g(i, F)+1))$ bounded F path, then from Lemma 3.5, there exists a short one with length $\leqslant (2^n(g(i, F)+1))^{n^c}$.

Case 2. Otherwise, let $v_1, ..., v_{m0}, v_{m0+1}, ..., v_m$ be the path such that v_{m0} is the first one not $(i+1) - 2^n(g(i, F)+1)$ bounded. Without loss of generality, we assume that $v_{m0}(i+1) > 2^n(g(i, F)+1)$. Note that no two of $v_1, ..., v_{m0}$ can agree on the first $i+1$ positions; otherwise, the path could be made even shorter. Therefore, $m_0 \leqslant (2^n(g(i, F)+1))^{i+1}$. Let p be the shortest i bounded F path in Petri net (P, T, φ, v_{m0}). Clearly, the length of p is $\leqslant g(i, F)$. Since $v_{m0}(i+1) > 2^n(g(i, F)+1)$ and each place of each transition vector in the Petri net is at most 2^n in absolute value, p must also be $(i+1)$ bounded and the $(i+1)$ position will never fall below 2^n in p (so that type 1 marking predicates of the form $\mu(i+1) \geqslant e$ and $\mu(i+1) > e$ will still hold in p). As a result, the sequence $v_1, ..., v_{m0-1}, p$ is an $(i+1)$ bounded F path of length $(2^n(g(i, F)+1))^{i+1} + g(i, F) < (2^n(g(i, F)+1))^{n^c}$. ▮

THEOREM 3.8. *The satisfiability problem can be decided in $O(2^{d*n*\log n})$ space, for some constant d independent of n.*

Proof. By recursive application of Lemmas 3.3 and 3.4, it is fairly easy to see that $g(k, F) < 2^{n^{d*n}}$, for some constant d. This means that if a formula F is satisfiable, then there must exist a path of length $< g(k, F) < 2^{n^{d*n}}$ satisfying F. A nondeterministic search procedure will yield an $O(2^{d*n*\log n})$ space upper bound. ▮

It is known that the boundedness problem for Petri nets is EXPSPACE-hard (Lipton, 1976). Since unboundedness can be expressed using our path formulas, the following result follows:

COROLLARY 3.9. *The satisfiability problem is EXPSPACE complete.*

4. SOME APPLICATIONS

In this section, we demonstrate the usefulness of Theorem 3.8 by first showing that many known complexity results concerning Petri nets follow immediately from our main result. We then derive some new results. Let $\mathscr{P} = (P, T, \varphi, \mu_0)$ be a k-place r-transition Petri net.

1. *Boundedness Problem.* The boundedness problem is the problem of determining whether $R(P, T, \varphi, \mu_0)$ is finite. This problem has been shown to be solvable in EXPSPACE in (Rackoff, 1978). (See also (Rosier and Yen, 1986) for a detailed multiparameter analysis.) Clearly, unboundedness can be formulated as $\exists \mu_1, \mu_2 \exists \sigma_1, \sigma_2((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge (\mu_2 > \mu_1))$.

2. *Coverability Problem.* The coverability problem is to determine, given a vector $v \in N^k$, whether there exists a reachable marking μ such that $\mu \geqslant v$. This problem has also been shown to be solvable in EXPSPACE in (Rackoff, 1978). Clearly, coverability can be formulated as $\exists \mu_1 \exists \sigma_1((\mu_0 \xrightarrow{\sigma_1} \mu_1) \wedge (\mu_1 \geqslant v))$.

3. *(Strict) Self-Coverability Problem.* In (Huynh, 1985), the notions of *self-coverability* and *strict self-coverability* were introduced for solving the equivalence problem of commutative semigroups and symmetric vector addition systems. It was observed in (Huynh, 1985) that the problem is solvable in EXPSPACE. Given a set of places I $(\subseteq P)$, a path $\mu_0 \xrightarrow{*} \mu_1 \xrightarrow{*} \mu_2$ is *I-self-covering* if $\forall s \in I$, $\mu_2(s) \geqslant \mu_1(s)$, and $\forall s' \notin I$, $\mu_2(s') = \mu_1(s')$. It is strict if "$\geqslant$" in the above definition is replaced by "$>$." The *(strict) self-coverability problem* is to determine, given an I, whether there is a (strict) I-self-covering path. It is easy to see that the existence of an I-self-covering path can be expressed as $\exists \mu_1, \mu_2 \exists \sigma_1, \sigma_2((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge ((\bigwedge_{s \in I} \mu_2(s) \geqslant \mu_1(s)) \wedge (\bigwedge_{s' \notin I} \mu_2(s') = \mu_1(s'))))$. (For strict self-coverability, simply replace "$\geqslant$" by "$>$."

4. *u-Self-Coverability Problem.* This problem was also defined and solved by Huynh (1985). Given a $u \in N^k$, a path $\mu_0 \xrightarrow{*} \mu_1 \xrightarrow{*} \mu_2$ is *u-self-covering* if $\mu_2 - \mu_1 = u$. The *u-self-coverability problem* is to determine, given a u, whether a u-self-covering path exists. This property can be expressed as $\exists \mu_1, \mu_2 \exists \sigma_1, \sigma_2((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge (\mu_2 - \mu_1 = u))$.

5. *Final-State Self-Coverability Problem.* This problem was defined and solved by Sistla and German (1987) to study the complexity of the

model checking problem for a special type of systems with many identical processes. It was originally defined in the context of vector addition systems with states. Here we consider an equivalent version of the problem in the context of Petri nets. Let F be a set of places (corresponding to the set of final states in a vector addition system with states). A *final-state self-covering* path (with respect to F) is $\mu_0 \xrightarrow{*} \mu_1 \xrightarrow{*} \mu_2 \xrightarrow{*} \mu_3$ where $\mu_3 \geqslant \mu_1$ and $\exists s \in F, \mu_2(s) > 0$. The *final-state self-coverability problem* is to determine whether a such path exists. This problem can easily be formulated as $\exists \mu_1, \mu_2, \mu_3 \, \exists \sigma_1, \sigma_2, \sigma_3((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2 \xrightarrow{\sigma_3} \mu_3) \wedge ((\mu_3 \geqslant \mu_1) \wedge (\bigvee_{s \in F} \mu_2(s) > 0)))$.

6. *Fair Nontermination Problems.* Let $\mathscr{A}$ be a finite set of nonempty subsets of transitions. Given an infinite sequence of transitions $\sigma = t_1, t_2, \ldots$, let $\inf^T(\sigma)$ be the set of transitions occurring infinitely often in σ. In (Howell, Rosier, and Yen, 1991) the following 6 types of fairness were defined: σ is said to be

- T1-fair iff $\exists A \in \mathscr{A}, \exists i \geqslant 1, t_i \in A$.
- T1′-fair iff $\exists A \in \mathscr{A}, \forall i \geqslant 1, t_i \in A$.
- T2-fair iff $\exists A \in \mathscr{A}, \inf^T(\sigma) \cap A \neq \emptyset$.
- T2′-fair iff $\exists A \in \mathscr{A}, \inf^T(\sigma) \subseteq A$.
- T3-fair iff $\exists A \in \mathscr{A}, \inf^T(\sigma) = A$.
- T3′-fair iff $\exists A \in \mathscr{A}, A \subseteq \inf^T(\sigma)$.

The *fair nontermination problem* with respect to T1 (T1′, T2, T2′, T3, T3′, respectively) fairness is the problem of determining whether a given Petri net has an infinite type T1- (T1′-, T2-, T2′-, T3-, T3′-, respectively) fair computation. In (Howell, Rosier, and Yen, 1991), the nontermination problem was shown to be equivalent to the boundedness problem (and hence, solvable in exponential space) for all six types of fairness. In what follows, we show how to formulate the fair nontermination problem for all six notions of fairness. This will immediately yield the upper bounds. Given a subset of transitions A, let v_A be a vector in N^r such that $v_A(i) = 1 (0)$ iff $t_i \in (\notin) A$. In what follows, let $\sigma \neq \Lambda$ denote the predicate $\bigvee_{t \in T} \#_\sigma(t) > 0$, meaning that σ contains at least one transition. A Petri net is X-fair nonterminating iff

- $X = \mathrm{T1}$

$$\Leftrightarrow \exists \mu_1, \mu_2 \, \exists \sigma_1, \sigma_2 \Bigg((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge \Bigg(\Bigg(\bigvee_{A \in \mathscr{A}} \bigvee_{t_i \in A} 1 \leqslant \#_{\sigma_1}(t_i) \Bigg) \wedge (\mu_2 \geqslant \mu_1) \wedge (\sigma_2 \neq \Lambda) \Bigg) \Bigg).$$

- $X = \mathrm{T1}'$

$$\Leftrightarrow \exists \mu_1, \mu_2 \, \exists \sigma_1, \sigma_2 \Bigg((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge \Bigg(\Bigg(\bigvee_{A \in \mathscr{A}} \bigwedge_{t_i \notin A} \#_{\sigma_1}(t_i) \leqslant 0 \wedge \#_{\sigma_2}(t_i) \leqslant 0 \Bigg) \wedge (\mu_2 \geqslant \mu_1) \wedge (\sigma_2 \neq \Lambda) \Bigg) \Bigg).$$

- $X = \mathrm{T2}$

$$\Leftrightarrow \exists \mu_1, \mu_2 \, \exists \sigma_1, \sigma_2 \Bigg((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge \Bigg(\Bigg(\bigvee_{A \in \mathscr{A}} \bigvee_{t_i \in A} 1 \leqslant \#_{\sigma_2}(t_i) \Bigg) \wedge (\mu_2 \geqslant \mu_1) \wedge (\sigma_2 \neq \Lambda) \Bigg) \Bigg).$$

- $X = \mathrm{T2}'$

$$\Leftrightarrow \exists \mu_1, \mu_2 \, \exists \sigma_1, \sigma_2 \Bigg((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge \Bigg(\Bigg(\bigvee_{A \in \mathscr{A}} \bigwedge_{t_i \notin A} \#_{\sigma_2}(t_i) \leqslant 0 \Bigg) \wedge (\mu_2 \geqslant \mu_1) \wedge (\sigma_2 \neq \Lambda) \Bigg) \Bigg).$$

- $X = \mathrm{T3}$

$$\Leftrightarrow \exists \mu_1, \mu_2 \, \exists \sigma_1, \sigma_2 \Bigg((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge \Bigg(\Bigg(\bigvee_{A \in \mathscr{A}} \Bigg(\bigwedge_{t_i \notin A} \#_{\sigma_2}(t_i) \leqslant 0 \Bigg) \wedge (v_A \leqslant \#_{\sigma_2}) \Bigg) \wedge (\mu_2 \geqslant \mu_1) \wedge (\sigma_2 \neq \Lambda) \Bigg) \Bigg).$$

- $X = \mathrm{T3}'$

$$\Leftrightarrow \exists \mu_1, \mu_2 \, \exists \sigma_1, \sigma_2 \Bigg((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge \Bigg(\Bigg(\bigvee_{A \in \mathscr{A}} v_A \leqslant \#_{\sigma_2} \Bigg) \wedge (\mu_2 \geqslant \mu_1) \wedge (\sigma_2 \neq \Lambda) \Bigg) \Bigg).$$

In the remainder of this section, we show how to use Theorem 3.8 to derive new results. By reduction to the satisfiability problem, each of the following problems will be shown to be solvable in EXPSPACE.

1. *Regularity Detection Problem.* In (Ginzburg and Yoeli, 1980; Valk and Vidal–Naquet, 1982), a subclass of Petri nets called *regular* Petri nets was defined. A Petri net is regular iff the set of all (finite) fireable sequences of transitions defines a regular language (over T). An algorithm was given by Ginzburg and Yoeli (1980), and Valk and Vidal–Naquet (1982) to determine whether a given Petri net is regular. A Petri net is not regular iff there exist markings $\mu_1, \mu_2, \mu_3, \mu_4$ such that μ_i, $1 \leqslant i \leqslant 4$, is reachable from μ_{i-1} and the following conditions are met (see (Valk and Vidal–Naquet, 1982, p. 314):

(a) $\mu_1 \leqslant \mu_2$ and $\mu_1 \neq \mu_2$,
(b) $\forall p \in P, (\mu_1(p) \geqslant \mu_2(p)) \Rightarrow (\mu_3(p) \leqslant \mu_4(p))$, and
(c) $\exists p \in P, \mu_3(p) > \mu_4(p)$.

However, no complexity analysis was given regarding that particular algorithm. Using our path formulas, we have that a k-place Petri net is not regular iff $\exists \mu_1, \mu_2, \mu_3, \mu_4 \exists \sigma_1, \sigma_2, \sigma_3, \sigma_4 (\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2 \xrightarrow{\sigma_3} \mu_3 \xrightarrow{\sigma_4} \mu_4)$ 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 result, the regularity detection problem is solvable in exponential space.

2. (*Potential*) *Determinism Detection Problem.* In (Howell and Rosier, 1989), a subclass of Petri nets called *deterministic*[2] Petri nets was defined. A Petri net is deterministic iff for any reachable marking μ, there is at most one enabled transition in μ. It has also been shown by Howell and Rosier (1989) that the boundedness, reachability, containment, and equivalence problems can all be solved in EXPSPACE. Let v_t be the minimum vector for which transition t is enabled. Clearly, a Petri net is not deterministic iff $\exists \mu_1 \exists \sigma_1 ((\mu_0 \xrightarrow{\sigma_1} \mu_1) \wedge (\bigvee_{t,t',t \neq t'} (\mu_1 \geqslant v_t) \wedge (\mu_1 \geqslant v_{t'})))$. A Petri net is *potentially deterministic* iff there does not exist an infinite path along which two or more transitions are enabled in infinitely many markings. (I.e., the Petri net can only be nondeterministic for a finite period of time.) It is fairly easy to see that a Petri net is not potentially deterministic iff $\exists \mu_1, \mu_2 \exists \sigma_1, \sigma_2 ((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge ((\bigvee_{t,t',t \neq t'} (\mu_1 \geqslant v_t) \wedge (\mu_1 \geqslant v_{t'})) \wedge (\mu_2 \geqslant \mu_1)))$. Consequently, the (potential) determinism detection problem can be solved in EXPSPACE.

[2] The reader should notice that the definition in (Howell and Rosier, 1989) differs from the original definition of deterministic Petri nets given in (Ramchandani, 1974).

3. *Frozen Token Detection Problem.* The concept of *frozen token* was first introduced by Best and Merceron (1985) as a way to study the behavior of a concurrent system modeled by a Petri net. A Petri net has a frozen token iff there exist an infinite computation $\mu_0 \xrightarrow{*} \mu_1 \xrightarrow{t_{a_1}} \mu_2 \xrightarrow{t_{a_2}} \mu_3 \cdots$ and a place s such that $\forall i \geqslant 1$, $\mu_i(s) \geqslant 1 + \varphi(s, t_{a_i})$. (I.e., there exists a token that will never be removed from a place.) To detect a frozen token, we first modify the Petri net as follows. Let P be the set of places. Let p be a new place. For every $q \in P$, we introduce a transition t_q such that $\varphi(q, t_q) = \varphi(t_q, p) = 1$. Let $\mu'_0(p) = 0$, and let μ'_i be the extension of μ_i by taking place p into consideration. Then we claim that the new Petri net has a frozen token iff $\exists \mu'_1, \mu'_2 \, \exists \sigma'_1, \sigma'_2((\mu'_0 \xrightarrow{\sigma'_1} \mu'_1 \xrightarrow{\sigma'_2} \mu'_2) \wedge (\mu'_1(p) > 0 \wedge \mu'_2 \geqslant \mu'_1) \wedge (\sigma'_2 \neq \Lambda))$. To see this, suppose the above formula is satisfiable. Then there exists a path $\mu'_0 \xrightarrow{\sigma'_1} \mu'_1 \xrightarrow{\sigma'_2} \mu'_2$ satisfying (1) $\mu'_1(p) > 0$ (2) $\mu'_2 \geqslant \mu'_1$ and (3) $\sigma'_2 \neq \Lambda$. (2) and (3) guarantee that σ'_2 can be repeated an arbitrary number of times. (1) ensures that at least one token has been deposited in p when reaching μ'_1. Since p is a sink, once reaching p a token cannot contribute to further computation. As a result, one can think of any token in p as a frozen token in the original Petri net. Let σ_1, σ_2 be the resulting sequences of transitions by removing all added transitions (i.e., transitions of the form t_q) from σ'_1, σ'_2, respectively. It is reasonably easy to see that $\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2 \xrightarrow{\sigma_2} \cdots$ is an infinite path with at least one token frozen. The other direction (i.e., every Petri net with a frozen token can be converted into a Petri net satisfying the conditions described above) is similar. Again, p can be thought of as a sink for storing those frozen tokens. Condition (1) ensures that we may freeze at least one token during the course of an infinite computation of a Petri net. In view of the above, the frozen token detection problem is solvable in EXPSPACE.

4. (*Strong*) *Promptness Detection.* The concept of (*strong*) *promptness* was introduced by Valk and Jantzen (1985) as a way to deal with systems communicating with the environment. Let T_{I} and T_{E} be two disjoint sets of transitions such that $T_{\mathrm{I}} \cup T_{\mathrm{E}} = T$. ($T_{\mathrm{I}}$ and T_{E} can be viewed as the sets of internal and external transitions, respectively.) A Petri net (P, T, φ, μ_0) is said to be

(a) *Strongly prompt* (with respect to $(T_{\mathrm{I}}, T_{\mathrm{E}})$) iff $\exists k \in N$, $\forall \mu \in R(P, T, \varphi, \mu_0)$, $\forall w \in T_{\mathrm{I}}^*: \mu \xrightarrow{w} \Rightarrow |w| < k$, meaning that for every reachable marking μ, the longest sequence of internal transitions firable in μ is of length $< k$, for some k.

(b) *Prompt* (with respect to $(T_{\mathrm{I}}, T_{\mathrm{E}})$) iff $\forall \mu \in R(P, T, \varphi, \mu_0)$, $\exists k \in N$, $\forall w \in T_{\mathrm{I}}^*: \mu \xrightarrow{w} \Rightarrow |w| < k$, meaning that for every reachable marking μ, there is no infinite sequence of internal transitions firable in μ.

It is not hard to see that a Petri net is not prompt (strongly prompt) iff $\exists \mu_1, \mu_2 \exists \sigma_1, \sigma_2((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge ((\bigwedge_{t \notin T_1} \#_{\sigma_2}(t) \leqslant 0) \wedge (\mu_2 \geqslant \mu_1) \wedge (\sigma_2 \neq \Lambda)))$. As a result, the (strong) promptness detection problem is solvable in exponential space.

5. *y-Synchronization Problem.* In (Suzuki and Kasami, 1983), the notion of *y-distance* was introduced. Given a Petri net $\mathscr{P}$ and a $y \in Z^r$, let $D(\mathscr{P}, y) = \sup_{\mu_0 \xrightarrow{\sigma}} (|y \odot \#_\sigma|)$. $\mathscr{P}$ is said to be *y-synchronized* if $D(\mathscr{P}, y)$ is finite. The *y-synchronization problem* is that of determining, given a Petri net $\mathscr{P}$ and a y, whether $\mathscr{P}$ is y-synchronized. This problem has been shown to be decidable and EXPSPACE-hard by Suzuki and Kasami (1983). However, the precise complexity was left unanswered there. Now, we show that the problem is in fact EXPSPACE-complete. As mentioned by Suzuki and Kasami (1983), $D(\mathscr{P}, y)$ is finite iff for every path $\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2$, if $\mu_2 \geqslant \mu_1$, then $|y \odot \#_{\sigma_2}| = 0$. Using our path formulas, $\mathscr{P}$ is not y-synchronized iff $\exists \mu_1, \mu_2 \exists \sigma_1, \sigma_2((\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2) \wedge (((y \odot \#_{\sigma_2}) > 0 \vee (y \odot \#_{\sigma_2} < 0)) \wedge (\mu_2 \geqslant \mu_1)))$. As a consequence, the y-synchronization problem can be solved in EXPSPACE.

Acknowledgment

The author thanks the anonymous referees for numerous suggestions that greatly improved the presentation as well as the correctness of these results. The author also thanks Professor L. Rosier for his suggestions and encouraging comments.

Received August 23, 1989; final manuscript received June 27, 1990

References

Araki, T., and Kasami, T. (1977), Decidable problems on the strong connectivity of Petri net reachability sets, *Theoret. Comput. Sci.* **4**, 97.

Best, E., and Merceron, A. (1985), Frozen tokens and *d*-continuity: A study in relating system properties to process properties, *in* "Lecture Notes in Computer Science," Vol. 188, pp. 48–61, Springer-Verlag, New York/Berlin.

Borosh, I., and Treybis, L. (1976), Bounds on positive integral solutions of linear diophantine equations, *Proc. Amer. Math. Soc.* **55**, 299.

Chan, T. (1987), On two way weak counter machines, *Math. Systems Theory* **20**, 31.

Crespi-Reghizzi, S., and Mandrioli, D. (1975), A decidability theorem for a class of vector addition systems, *Inform. Process. Lett.* **3**, 78.

Finkel, A. (1987), A generalization of the procedure of Karp and Miller to well structured transition systems, *in* "Proceedings, 13th International Colloquium on Automata, Languages and Programming," pp. 499–508; *Inform. Comput.*, to appear.

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

HOPCROFT, J., AND PANSIOT, J. (1979), On the reachability problem for 5-dimensional vector addition systems, *Theoret. Comput. Sci.* **8**, 135.

HOWELL, R., AND ROSIER, L. (1988), Completeness results for conflict-free vector replacement systems, *J. Comput. System Sci.* **37**, 349.

HOWELL, R., AND ROSIER, L. (1989), personal communication.

HOWELL, R., ROSIER, L., AND YEN, H. (1991), A taxonomy of fairness and temporal logic problems for Petri nets, *Theoret. Comput. Sci.* **82**, 341.

HOWELL, R., ROSIER, L., AND YEN, H. (1989), Normal and sinkless Petri nets, *in* "Proceedings, 7th International Conference on the Fundamentals of Computation Theory," pp. 234–243; *J. Comput. System Sci.*, to appear.

HOWELL, R., ROSIER, L., HUYNH, D., AND YEN, H. (1986), Some complexity bounds for problems concerning finite and 2-dimensional vector addition systems, *Theoret. Comput. Sci.* **46**, 107.

HUYNH, D. (1985), The complexity of the equivalence problem for commutative semigroups and symmetric vector addition systems, *in* "Proceedings, 17th ACM Symposium on Theory of Computation," 405.

JANCAR, P. (1989), Decidability of weak fairness in Petri nets, *in* "Proceedings, 6th Symposium on Theoretical Aspects of Computer Science," 446.

KARP, R., AND MILLER, R. (1969), Parallel program schemata, *J. Comput. System Sci.* **3**, 147.

LANDWEBER, L., AND ROBERTSON, E. (1978), Properties of conflict-free and persistent Petri nets, *J. Assoc. Comput. Mach.* **25**, 352.

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

MAYR, E. (1981), Persistence of vector replacement systems is decidable, *Acta Inform.* **15**, 309.

MAYR, E. (1984), An algorithm for the general Petri net reachability problem, *SIAM J. Comput.* **13**, 441.

MULLER, H. (1980), Decidability of reachability in persistent vector replacement systems, *in* "Proceedings, 9th Symposium on Mathematical Foundations of Computer Science," 426.

RACKOFF, C. (1978), The covering and boundedness problems for vector addition systems, *Theoret. Comput. Sci.* **6**, 223.

RAMCHANDANI, C. (1974), Analysis of asynchronous concurrent systems by Petri nets, Project MAC TR-120, MIT.

ROSIER, L., AND YEN, H. (1986), A multiparameter analysis of the boundedness problem for vector addition systems, *J. Comput. System Sci.* **32**, 105.

SISTLA, A., AND GERMAN, S. (1987), Reasoning with many processes, *in* "Proceedings, IEEE Symposium on Logic in Computer Science," 138.

SUZUKI, I., AND KASAMI, T. (1983), Three measures for synchronic dependence in Petri nets, *Acta Inform.* **19**, 325.

VALK, R., AND JANTZEN, M. (1985), The residue of vector sets with applications to decidability problems in Petri nets, *Acta Inform.* **21**, 643.

VALK, R., AND VIDAL-NAQUET, G. (1982), Petri nets and regular languages, *J. Comput. System Sci.* **23**, 299.

VAN LEEUWEN, J. (1974), A partial solution to the reachability problem for vector addition systems, *in* "Proceedings, ACM Symposium on Theory of Computation," 303.

YAMASAKI, H. (1981), On weak persistency of Petri nets, *Inform. Process. Lett.* **13**, 94.

YAMASAKI, H. (1984), Normal Petri nets, *Theoret. Comput. Sci.* **31**, 307.

Printed in Belgium