International Journal of Foundations of Computer Science

Some Complexity Results for Rings of Petri Nets*

Hsu-Chun Yen, Bow-Yaw Wang, and Ming-Shang Yang
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 106, Republic of China

Received (received date)
Revised (revised date)
Communicated by Editor's name

ABSTRACT

We define a subclass of Petri nets called $m-state\ n-cycle$ Petri nets, each of which can be thought of as a ring of n bounded (by m states) Petri nets using n potentially unbounded places as *joins*. Let $Ring(n, l, m)$ be the class of $m-state\ n-cycle$ Petri nets in which the largest integer mentioned can be represented in l bits (when the standard binary encoding scheme is used). As it turns out, both the *reachability* problem and the *boundedness* problem can be decided in $O(n(l + log\ m))$ nondeterministic space. Our results provide a slight improvement over previous results for the so-called *cyclic communicating finite state machines.* We also compare and contrast our results with that of $VASS(n, l, s)$, which represents the class of n-dimensional s-state VASSs where the largest integer mentioned can be described in l bits.

Keywords: Communicating finite state machine, complexity, boundedness, Petri net, reachability, vector addition system with states.

1. Introduction

Models proposed in the literature for concurrent and/or distributed systems have included *Petri nets* (or equivalently, *Vector Addition Systems* (VASs), *Vector Addition Systems with States* (VASSs)), and more recently networks of *Communicating Finite State Machines* (CFSMs). Of problems that are of interest and importance to the study of the above computational models, *reachability* and *boundedness* are two that have received considerable attention in the computer science community. For the Petri net model, the reachability problem is known to be decidable (and is EXPSPACE-hard[a]) [1,2,3], whereas the boundedness problem is EXPSPACE-complete [2,4]. For networks of CFSMs, however, both the reachability and the boundedness problems are undecidable in general [5].

*This work was supported in part by the National Science Council of the Republic of China under NSC-83-0408-E-002-055.

This is an abstract footnote

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

One way to ease the high degree of complexity associated with deciding reachability and boundedness is to lower our expectations by considering *restricted* versions of the above computational models. For instance, research along the line of focusing on *restricted* networks of CFSMs can be found in [6,7,8,9,10,11]. In a recent article [8], a class of CFSMs called *cyclic* CFSMs has been introduced. Basically, the network topology is such that CFSMs form a ring along which neighboring CFSMs can exchange a single type of message in one direction. In [8], it has been shown that the *deadlock detection* problem (equivalently, the reachability problem), and the *unboundedness* problem are solvable in EXPTIME[b]. For practical motivation behind the study of such CFSMs, the reader is referred to [8].

In this paper, we study the complexities of the reachability and the boundedness problems for Petri nets with a ring-structured topology as depicted in Figure 1. As Figure 1 indicates, such a Petri net consists of n bounded Petri net modules $T_0, T_1, ...,$ and T_{n-1}. Each T_i, $0 \leq i \leq n-1$, is capable of 'communicating' with its two neighbors by adding tokens to place $p_{(i+1) mod\ n}$ and/or subtracting tokens from place p_n, respectively. Let $Ring(n, l, m)$ denote the class of ring-structured Petri nets consisting of n modules each of which can be in one of at most m states (configurations), and the largest integer mentioned, in any instance, can be represented in l bits (when the standard binary encoding scheme is used). Our model can be thought of as a generalization of that of [8] as the firing of each of Petri net's transitions is capable of depositing (consuming) 2^l tokens to (from) a place, as opposed to the cyclic CFSM model in which at most one message (token) can be sent to (received from) a channel.

As it turns out, both the reachability and the boundedness problems are solvable in $O((l + log\ m)n)$ nondeterministic space for $Ring(n, l, m)$. Our result provides a slight improvement over that of [8] provided that PSPACE[c] $\subsetneq$ EXPTIME. From the Petri net standpoint, our analysis is also of interest in its own right. Let $VASS(n, l, s)$ be the class of n-dimensional s-state VASSs where the largest integer mentioned can be described in l bits. Consider the work of [12] in which reachability for $VASS(2, l, s)$ has been shown to be decidable in DTIME($2^{2^{cls}}$), where c is a constant. Now for $Ring(2, l, m)$ ($\subseteq VASS(2, l, m^2)$), our result yields $O(l + log\ m)$ nondeterministic space for deciding reachability, which, again, is an improvement over that of [12] provided that PSPACE $\subsetneq$ EXPTIME. Notice that both $VASS(2, l, s)$ and $Ring(2, l, m)$ allow transitions to add to or subtract from two potentially unbounded places (or equivalently, vector components). The reason for the disparity in complexity appears to be that for $VASS(2, l, s)$, the actions of the two potentially unbounded places are controlled by a single finite-state control, whereas for $Ring(2, l, m)$, the finite-state control is splitted into two separate units which operate in an asynchronous fashion. For the boundedness problem, the reader should compare and contrast our $O((l + log\ m)n)$ nondeterministic space result for

[b] EXPTIME=$\bigcup_{c>0} DTIME(2^{n^c})$, where $DTIME(f(n))$ is the class of languages accepted by deterministic Turing machines using at most $f(n)$ time.

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

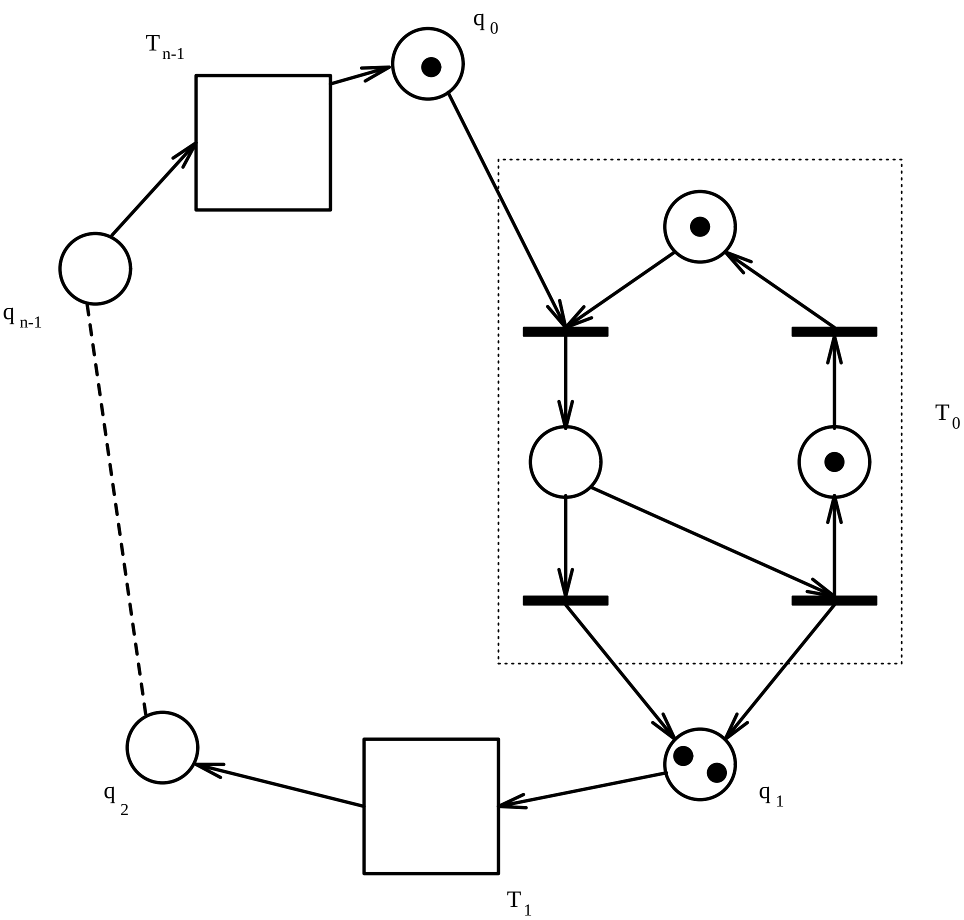

Fig. 1. Ring Structure.

$Ring(n,l,m)$ ($\subseteq VASS(n,l,m^n)$) with the EXPSPACE-completeness result (more precisely, $O(2^{c*n*log\ n}(l + log\ s))$ nondeterministic space) for $VASS(n,l,s)$ [13]. Finally, the reachability problem for general Petri nets is known to be decidable; the best decision procedure, however, is not even primitive recursive. Again, we feel that the separation of the finite-state control, in conjunction of the uni-direction flow of tokens, has make both reachability and boundedness easier to analyze, in comparison with their general counter-parts.

2. Definitions

Let N denote the set of nonnegative integers, Z the set of integers, and N^k (Z^k) the set of vectors of k nonnegative integers (integers, respectively). For a k-dimensional vector v, let v(i), $1 \le i \le k$, denote the *i*th component of v. Throughout the rest of this paper, we let $a \oplus_n b$ denote $(a+b)\ mod\ n$.

A *Petri net* (PN, for short) 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* φ : (P $\times$ T) $\cup$ (T $\times$ P) $\rightarrow$ N, and μ_0, the *initial marking*, is a mapping from P to N. A *marking* is a mapping μ : P $\rightarrow$ N. Suppose P' is a subset of P and μ is a marking. We define $\mu(P')$ to be the restriction of φ on P'. We define the *transition vector* of a transition t, denoted by $\bar{t}$, to be a k-dimensional vector in Z^k (where $k = |P|$) such that $\bar{t}(i) = \varphi(t,p_i) - \varphi(p_i,t)$.

A transition t $\in$ T is *enabled* at a marking μ iff for every p $\in$ P, φ(p,t) $\le$ μ(p). A transition t may *fire* at a marking μ if t is enabled at μ. We then write $\mu \stackrel{t}{\rightarrow} \mu'$, where μ'(p) = μ(p) $-$ φ(p,t) + φ(t,p) for all p $\in$ P. (Equivalently, $\mu' = \mu + \bar{t}$.) A sequence of transitions $\sigma = t_1 ... t_n$ is a *firing sequence* from μ_0 (or a firing sequence of (P,T,φ,μ_0)) iff $\mu_0 \stackrel{t_1}{\rightarrow} \mu_1 \stackrel{t_2}{\rightarrow} \cdots \stackrel{t_n}{\rightarrow} \mu_n$ for some sequence of markings μ_1,...,μ_n. The *reachability set* of $\mathcal{P}$ is the set R($\mathcal{P}$) = $\{\mu \mid \mu_0 \stackrel{\sigma}{\longmapsto} \mu$ for some $\sigma\}$. The *reachability problem* is that of determining, given a Petri net $\mathcal{P}$ and a marking μ, whether $\mu \in$ R($\mathcal{P}$). The *boundedness problem* is to determine whether there exists a $k \in N$ such that $\forall \mu \in$ R($\mathcal{P}$) $\forall p \in P$, $\mu(p) \le k$. (In other words, the size of R($\mathcal{P}$) is finite.) Given a set of transitions $\tilde{T}$, we define ${}^\bullet\tilde{T} = \{q | \varphi(q,t) > 0, \exists t \in \tilde{T}\}$ and $\tilde{T}^\bullet = \{q | \varphi(t,q) > 0, \exists t \in \tilde{T}\}$.

A Petri net (P,T,φ,μ_0) is said to be an *n-cycle* Petri net if P and T can be partitioned into disjoint subsets $P_0, P_1, ..., P_{n-1}, \{q_0\}, \{q_1\}, ..., \{q_{n-1}\}$ and $T_0, T_1, ..., T_{n-1}$, respectively, such that $\forall i, 0 \le i \le n-1$, ${}^\bullet T_i = \{q_i\} \cup P_i$, and $T_i^\bullet = \{q_{i \oplus_n 1}\} \cup P_i$. Figure 1 displys such a Petri net, which can be viewed as the composition of n Petri nets $(P_0 \cup \{q_0,q_1\}, T_0, \varphi_0)$, $(P_1 \cup \{q_1,q_2\}, T_1, \varphi_1)$, ..., $(P_{n-1} \cup \{q_{n-1}, q_0\}, T_{n-1}, \varphi_{n-1})$ in a circular fashion using places $q_0, q_1, ..., q_{n-1}$ as *joins*. If $\forall i, 0 \le i \le n-1$, $|\{\mu(q) | \forall \mu \in R(P,T,\varphi,\mu_0), \forall q \in P_i\}| \le m$, then the Petri net is said to be an *m-state* Petri net. We let $Ring(n,l,m)$ denote the class of m-state n-cycle Petri nets in which the largest integer mentioned can be represented in l bits. For ease of expression, such a Petri net will be written as $\{q_0\} \rightarrow (P_0,T_0,\varphi_0) \rightarrow \{q_1\} \rightarrow (P_1,T_1,\varphi_1) \rightarrow \cdots \rightarrow \{q_{n-1}\} \rightarrow (P_{n-1},T_{n-1},\varphi_{n-1}) \rightarrow \{q_0\}$, dubbed the *description* of the Petri net. Notice that in our setting, the firing of a transition may result in removing (or adding) 2^l tokens from (to) a place, for integers are represented in binary.

3. Complexity Analysis

In this section, we analyze the complexities of detecting reachability and boundedness for $Ring(n,l,m)$. For the reachability problem, our proof relies on showing that if a marking is reachable, then there exists a witnessing path along which all but one place are bounded by 2^l. Viewing such bounded places together with their transitions as a finite state control gives rise to a computation which resembles that of a *one-counter automaton*. We then take advantage of techniques known to be useful for analyzing problems concerning one-counter automata. A similar technique can be applied to the boundedness problem.

To show our main results, a few lemmas are in order.

Lemma 1 *Given a Petri net* (P,T,φ,μ_0) *(*$= \{q_0\} \to (P_0,T_0,\varphi_0) \to \{q_1\} \to (P_1,T_1,\varphi_1) \to \cdots \to \{q_{n-1}\} \to (P_{n-1},T_{n-1},\varphi_{n-1}) \to \{q_0\}$*) in* $Ring(n,l,m)$ *and a marking* μ*, we can construct in polynomial time a Petri net* (P',T',φ',μ'_0) *(*$= \{q_0\} \to (P'_0,T'_0,\varphi'_0) \to \{q_1\} \to (P'_1,T'_1,\varphi'_1) \to \cdots \to \{q_{n-1}\} \to (P'_{n-1},T'_{n-1},\varphi'_{n-1}) \to \{q_0\}$*) in* $Ring(n,l,m')$ *and a marking* μ' *such that*

1. $\mu \in R(P,T,\varphi,\mu_0)$ *iff* $\mu' \in R(P',T',\varphi',\mu'_0)$,
2. $m' = 2^l * m$*, and*
3. $\forall 0 \le i \le n-1$, $\mu'_0(q_i) = 0$.

Proof. Let $N' = (P',T',\varphi',\mu'_0)$ be a Petri net whose description is $\{q_0\} \to (P_0 \cup \{q'_0\}, T_0 \cup \{t_0\}, \varphi'_0) \to \{q_1\} \to \cdots \to \{q_{n-1}\} \to (P_{n-1} \cup \{q'_{n-1}\}, T_{n-1} \cup \{t_{n-1}\}, \varphi'_{n-1}) \to \{q_0\}$, where

- $\varphi'_i(t,p) = \varphi_i(t,p), \forall t \in T_i, p \in P_i \cup \{q_{i \oplus_n 1}\}$,
- $\varphi'_i(p,t) = \varphi_i(p,t), \forall t \in T_i, p \in P_i \cup \{q_i\}$,
- $\varphi'_i(q'_i,t_i) = \varphi'_i(t_i,q_{i\oplus_n 1}) = 1$,
- $\mu'_0(q'_i) = \mu_0(q_{i\oplus_n 1})$, $\mu'_0(q_i) = 0, \forall i$, and $\mu'_0(p) = \mu_0(p), \forall p \in P_i$. (Notice that (3) holds.).

Intuitively, places $q'_0, q'_1, ..., q'_{n-1}$ serve for the purpose of 'hiding' the initial tokens of $q_1, q_2, ..., q_0$, respectively. By letting μ' be such that $\forall q \in P, \mu'(q) = \mu(q)$ and $\forall i, \mu'(q'_i) = 0$ (indicating that all the tokens originally hidden in q'_i must be released before reaching the designated marking), it is reasonably easy to see that $\mu \in R(P,T,\varphi,\mu_0)$ iff $\mu' \in R(P',T',\varphi',\mu'_0)$ – (1) holds. Finally, since the number of internal states of (P_i,T_i) is bounded by m, and the initial number of tokens in place q'_i (i.e., $\mu_0(q'_i)$) is bounded by 2^l, the number of internal states of (P'_i,T'_i) is therefore bounded by $2^l \cdot m$. Hence, (2) holds. □

Lemma 2 *Given a Petri net* (P,T,φ,μ_0) *(*$= \{q_0\} \to (P_0,T_0,\varphi_0) \to \{q_1\} \to (P_1,T_1,\varphi_1) \to \cdots \to \{q_{n-1}\} \to (P_{n-1},T_{n-1},\varphi_{n-1}) \to \{q_0\}$*) in* $Ring(n,l,m)$*, if* $\mu_0(q_i) = 0, \forall 0 \le i \le n-1$ *and* $\mu_0 \xrightarrow{\sigma} \mu$*, then* σ *can be rearranged into* $\sigma' = \sigma_1\sigma_2\cdots\sigma_n$ *satisfying the following conditions: (Let* $\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2 \cdots \xrightarrow{\sigma_n} \mu_n = \mu$*.)*

1. $\forall 1 \le i \le n-1, \forall 1 \le j \le n-i, \forall$ *marking* M *in* $\mu_{i-1} \xrightarrow{\sigma_i} \mu_i$, $M(q_j) \le 2^l - 1$,

2. $\forall 2 \le i \le n$, $\forall n-(i-1) \le j \le n-1$, $\forall t \in T_j$, $\#_{\sigma_i}(t) = 0$ *(i.e., transitions belonging to* $T_{n-1}, ..., T_{n-(i-1)}$ *do not take part in the execution of* σ_i*).*

Proof. In what follows, we show how to rearrange σ into a canonical sequence σ' satisfying conditions (1) and (2). Before presenting the details, we first give the intuition so as to allow the reader to have a better feel for how the procedure works. Let $\gamma_0, \gamma_1, \ldots, \gamma_{n-1}$ be the projections of σ on $T_0, T_1, \ldots, T_{n-1}$, respectively. The idea is as follows. The rearrangement is carried out in phases. Consider the first phase (i.e., transition sequence σ_1). Recall that all $q_i, 0 \le i \le n-1$, are empty initially. We start with firing as many transitions in γ_{n-1} (according to their original order in γ_{n-1}) as possible until a transition, say t, whose execution requires additional tokens to be placed in place q_{n-1} is reached. Clearly in order for t to proceed, a prefix of γ_{n-2} has to be executed until enough tokens have been deposited in q_{n-1}. Likewise, in order for the above prefix of γ_{n-2} to be firable, a number of tokens might have to be consumed from place q_{n-2}. This, in turn, requires portion of γ_{n-3} to be executed beforehand. By repeating the above argument, one is easy to see that portion of γ_0 may have to be fired so as to supply enough tokens along the 'pipeline' $q_1, q_2, ..., q_{n-1}$, resulting in the enabledness of t. Repeat the above until the entire γ_{n-1} is exhausted. This completes the first phase. The second phase is similar to the previous phase, except that we begin with the remaining sequence of γ_{n-2} instead. Notice that transitions belonging to T_{n-1} do not take part in the course of the second phase, The second phase ends immediately after the entire γ_{n-2} is exhausted. The remaining phases are similar.

Let $\gamma_0, \gamma_1, \ldots, \gamma_{n-1}$ be the projections of σ on $T_0, T_1, \ldots, T_{n-1}$, respectively. Consider the following procedure **GetToken(i)**, returning a firing sequence whose execution will result in depositing at least a token in place $q_{i \oplus_n 1}$. Sequences $\gamma_0, \gamma_1, \ldots, \gamma_{n-1}$ as well as the current marking μ are global to GetToken. For convenience, given a sequence $x = ty$ (where $t \in T$ and $x, y \in T^*$) we define $first(x)$ to be a procedure returning the head of x (i.e., t) while assigning y to x as a side-effect. In what follows, '$\cdot$' denotes the concatenation operator of strings.

procedure GetToken(i)
$\delta := \lambda$; (* Initialize δ to an empty string *)
repeat
  $t := first(\gamma_i)$;
  while $\varphi(q_i, t) > \mu(q_i)$ **do** $\delta := \delta\cdot$ GetToken(i-1); (* q_i has insufficient number of tokens *)
  $\delta := \delta \cdot t$;
  $\mu := \mu + \bar{t}$;
until $\varphi(t, q_{i \oplus_n 1}) > 0$ (* Tokens have been deposited in $q_{i \oplus_n 1}$ *)
GetToken := δ; (* Return string δ *)
end procedure

Using procedure GetToken, firing sequence σ' can be constructed from σ in the following way.

procedure Rearrange(σ)
σ_1, σ_2, $\cdots$, σ_n:=λ; (* Initialize σ_1, σ_2, $\cdots$, σ_n to empty strings *)
$\mu := \mu_0$; (* Initialize the current marking to the initial marking *)

for $i := n-1$ **downto** 0 **do**
    begin
    $x := \gamma_i$;
    while $x \neq \lambda$ **do**
        begin
        $t := first(x)$;
        while $\varphi(q_i, t) > \mu(q_i)$ **do** $\sigma_{n-i} := \sigma_{n-i} \cdot$ GetToken(i-1);
        $\sigma_{n-i} := \sigma_{n-i} \cdot t$;
        $\mu := \mu + \bar{t}$;
        end
    end
Rearrange $:= \sigma_1 \sigma_2 \cdots \sigma_n$;
end procedure

The validity of the above algorithm is straightforward. To give the reader a better feel for how the algorithm works, the profiles of the token distributions of places $q_0, q_1, \cdots, q_{n-1}$ against the canonical computation $\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2 \cdots \xrightarrow{\sigma_n} \mu_n = \mu$ are depicted in Figure 2. Intuitively, our strategy is to 'run' different modules at different speeds (according to the following order $T_{n-1} > T_{n-2} > T_1 > T_0$). By doing so, the majority of tokens during the course of the execution will pile up in place q_0 while the remaining places are bounded by 2^l at, almost, all times (except, perhaps, the final stage at which the designated marking is to be reached). □

In the following lemma, we prove that if μ is a reachable marking, there must be a 'short' path leading to μ. We basically apply a technique which was previously used in [14] (see also [10]) for analyzing counter machine computations.

Lemma 3 *Consider a Petri net (P, T, φ, μ_0) in Ring(n, l, m) and a reachable marking μ satisfying the conditions stated in Lemma 1. then there exists a 'short' path (reaching μ) along which no place possesses more than $2k^6$ tokens, where $k = m^n 2^{l(n-1)}$.*

Proof. Let $\mu_0 \xrightarrow{\sigma_1} \mu_1 \xrightarrow{\sigma_2} \mu_2 \cdots \xrightarrow{\sigma_n} \mu_n = \mu$ be the canonical computation guaranteed by Lemma 2. In what follows, we show how portion of $\sigma_1 \sigma_2 \cdots \sigma_n$ can be short-circuited if its length exceeds $2k^6$.

The number of tokens in q_i along the computation of σ is illustrated in Figure 2. Clearly, the number of tokens in $q_i, \forall 1 \leq i \leq n-1$, is bounded by $max\{\mu(q_i), 2^l\}$ $(= 2^l)$ at all times. This leaves q_0 as the sole place that is capable of possessing 'a large amount of tokens' along σ. In view of this, the crux of the remaining proof is to view subnet $(P_0, T_0, \varphi_0) \rightarrow \{q_1\} \rightarrow (P_1, T_1, \varphi_1) \rightarrow \cdots \rightarrow \{q_{n-1}\} \rightarrow (P_{n-1}, T_{n-1}, \varphi_{n-1})$ (i.e., the original Petri net less place q_0 and its neighboring arcs) as a finite-state control with no more than $m^n (2^l)^{n-1}$ states. In what follows, we let $k = m^n 2^{l(n-1)}$. Clearly, the number of tokens added to or subtracted from q_0 by the above integrated subnet is bounded by $2^l (\leq k)$.

Let σ' be a canonical path reaching μ, and suppose at some point, say b, the number of tokens in q_0 exceeds $2k^6$. Let a (resp., c) be the last (resp., first) point before (resp., after) b in which the number of tokens is k^6. Define intervals (a_i, b_i)

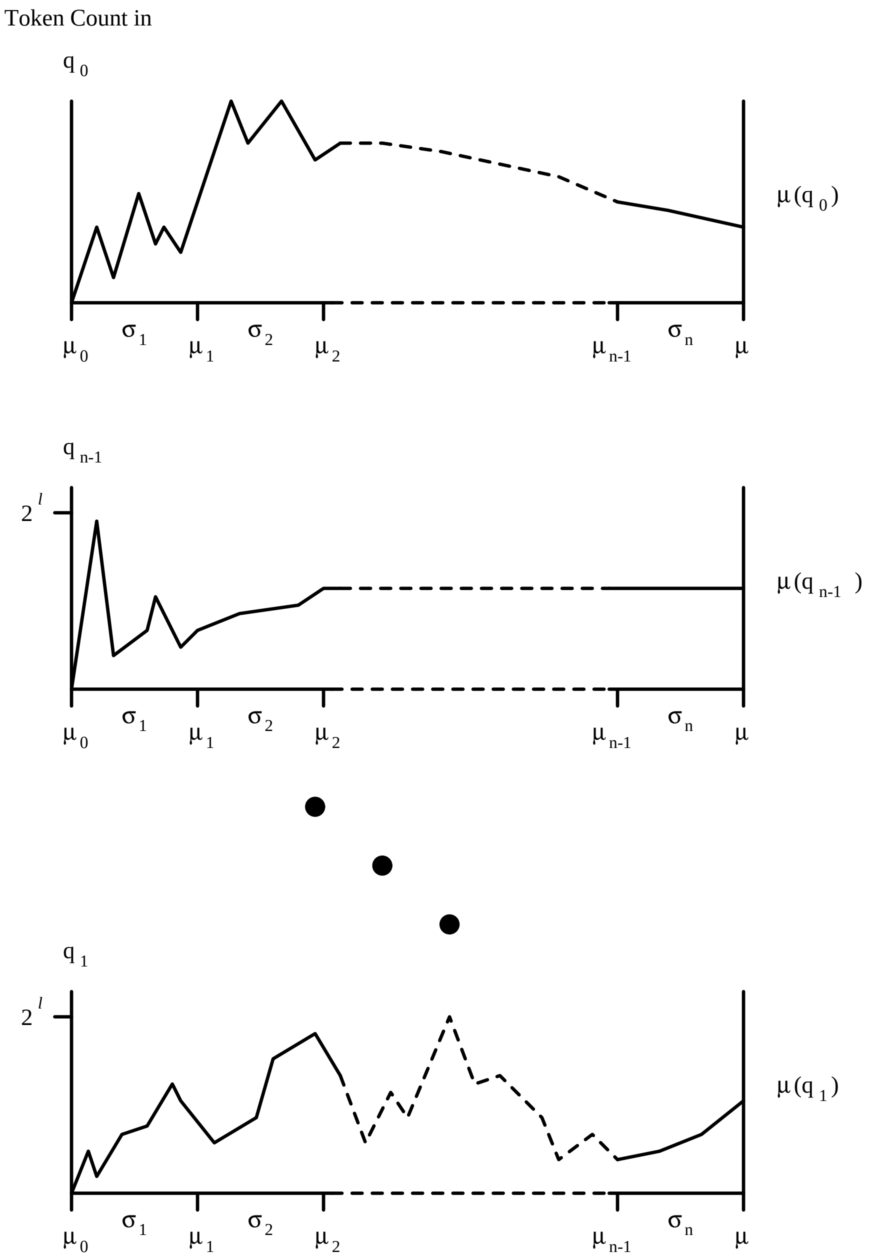

Fig. 2. token in $q_0, q_1, q_2, \ldots$

and (b'_i, c_i), $0 \leq i < k^4$ as follows: $a_0 = a$, a_{j+1} is the first point after a_j at which the number of tokens is $k^6 + (j+1) \cdot k^2$ for all $1 < j < k^4 - 1$. b_j is the first point after a_j at which the number of tokens is $k^6 + (j+2) \cdot k^2$. $c_0 = c$, c_{j+1} is the last point before c_j at which the number of tokens is $k^6 + (j+1) \cdot k^2$ for all $1 < j < k^4 - 1$. b'_j is the last point before c_j at which the number of tokens is $k^6 + (j+2) \cdot k^2$. Since the total number of states is bounded by k, there must exist at least one loop in every interval (a_i, b_i) (resp., (b'_i, c_i)) with a gain (resp., loss) in the token value of between 1 and k^2. See Figure 3.

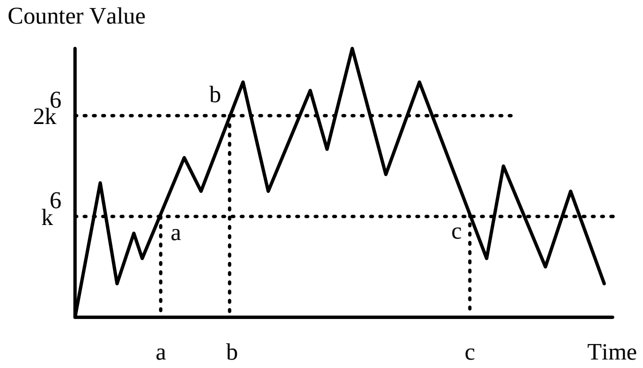

Fig. 3. Counter variation.

Now there are k^4 short loops between a and b (resp., b and c), among which at least k^2 loops must have the same gain (resp., loss) of say x (resp., y) tokens, where $1 \leq x, y \leq k^2$. Let u and v be such that $x * u = y * v =$ the least common multiple of x and y, $1 \leq u, v \leq k^2$. Now from σ' we can create a shorter path reaching μ by cutting off u short loops between a and b and v short loops between b and c, respectively, since the total increase in token number by those u loops is exactly offset by the total loss by those v loops. Hence, we a shorter compuation reaching the same marking μ. The above procedure can be repeated until there is no time in which the number of tokens in q_0 exceeds $2k^6$. □

Theorem 1 *The reachability problem for Petri nets in Ring(n, l, m) is solvable in $O(n(l + log\ m))$ nondeterministic space.*

Proof. According to Lemmas 1 and 3, the upper bound follows immediately from a nondeterministic search procedure using at most $O(log(2k^6))$ nondeterministic space, where $k = (m2^l)^n 2^{l(n-1)}$. This amounts to $O(n(l + log\ m))$ nondeterministic space. □

In what follows, we consider the boundedness problem.

Lemma 4 *A Petri net (P, T, φ, μ_0) $(= \{q_0\} \to (P_0, T_0, \varphi_0) \to \{q_1\} \to (P_1, T_1, \varphi_1) \to \cdots \to \{q_{n-1}\} \to (P_{n-1}, T_{n-1}, \varphi_{n-1}) \to \{q_0\})$ in Ring(n, l, m) is unbounded iff there exists a computation $\mu_0 \xmapsto{\sigma_1} \mu_1 \xmapsto{\sigma_2} \mu_2$ and an i, $0 \leq i \leq n-1$, such that $\mu_2(q_i) > \mu_1(q_i)$ and $\forall p \neq q_i$ $\mu_2(p) = \mu_1(p) \leq 2^l$. In other words, the sequence $\mu_1 \xmapsto{\sigma_2} \mu_2$ yields a gain in token in place q_i, while tokens in the remaining places remain unchanged (and bounded by 2^l).*

Proof. Without loss of generality, we assume q_0 to be a potentialy unbounded place. Given an arbitrary $k \in N$, let $\mu_0 \stackrel{\sigma}{\longmapsto} \mu$ be a computation such that $\mu(q_0) > k$. Now we show that from σ we can construct a σ' such that $\mu_0 \stackrel{\sigma'}{\longmapsto} \mu'$, $\mu'(q_0) \geq \mu(q_0) > k$, and $\forall q \neq q_0, \forall$ marking M in $\mu_0 \stackrel{\sigma'}{\longmapsto} \mu'$, $M(q) < 2^l$. The proof idea parallels that of Lemma 2. Let $\gamma_0, \gamma_1, \ldots, \gamma_{n-1}$ be the projections of σ on $T_0, T_1, \ldots, T_{n-1}$, respectively. The following procedure generates such a sequence σ'.

procedure Seq(σ)
$\sigma_1 := \lambda$; (* Initialize σ_1 to an empty string *)
$\mu := \mu_0$; (* Initialize the current marking to the initial marking *)
$x := \gamma_{n-1}$;
while $x \neq \lambda$ **do**
  begin
  $t := first(x)$;
  while $\varphi(q_{n-1}, t) > \mu(q_{n-1})$ **do** $\sigma_1 := \sigma_1 \cdot$ GetToken(n-2);
  $\sigma_1 := \sigma_1 \cdot t$;
  $\mu := \mu + \bar{t}$;
  end
Seq := σ_1;
end procedure

Based on the above idea, there exists a path witnessing unboundedness with place q_0 as the only one whose token number can exceed 2^l. The lemma follows. □

Lemma 5 *Suppose (P, T, φ, μ_0) is a Petri net in Ring(n, l, m) and $\mu_0 \stackrel{\sigma_1}{\longmapsto} \mu_1 \stackrel{\sigma_2}{\longmapsto} \mu_2$ is one of the shortest paths satisfying the conditions stated in Lemma 4. Then along the above path no place possesses more than k^c tokens, where $k = m^n 2^{l(n-1)}$, and c is a constant.*

Proof. First consider the sequence σ_2. Suppose, at some point, q_0 exceeds $\mu_1(q_0) + k^2$ tokens. Then a short positive loop must exist – a contradiction (see the proof of Lemma 2). Therefore, the length of σ_2 is bounded by k^3 (= # of distinct states * # of tokens) regardless of what μ_1 is. Now for the prefix σ_1, consider the following two cases:

1. $\mu_1(q_0) \leq k^2$.
   An almost identical argument of the proof of Lemma 2 will yield our result.

2. $\mu_1(q_0) > k^2$.
   In this case, there must exist a short positive loop – a contradiction.

□

Thus, we immediately have:

Theorem 2 *The boundedness problem for Petri nets in Ring(n, l, m) is solvable in $O(n(l + \log m))$ nondeterministic space.*

Acknowledgements

We would like to thank the referee for suggestions which improved the content as well as the presentation of our results.

References

1. R. Kosaraju, Decidability of Reachability in Vector Addition Systems. *Proc. 14th Annu. ACM Symp. on Theory of Computing*, 1982, 267-280.
2. R. Lipton, The Reachability Problem Requires Exponential Space. Technical Report 62, Yale University, Dept. of CS., 1976.
3. E. Mayr, An Algorithm for the General Petri Net Reachability Problem. *SIAM J. Comput.*, Vol. 13, 1984, 441-460.
4. C. Rackoff, The Covering and Boundedness Problems for Vector Addition Systems. *Theoretical Computer Science*, Vol. 6, 1978, 223-231.
5. D. Brand and P. Zafiropulo, On Communicating Finite-State Machines. *JACM*, Vol. 30, No. 2, 1983, 323-342.
6. M. Gouda, E. Gurari, T. Lai and L. Rosier, On Deadlock Detection in Systems of Communicating Finite State Machines. *Computers and Artificial Intelligence*, Vol. 6, No. 3, 1987.
7. M. Gouda, E. Manning and Y. Yu, On the Progress of Communication between Two Finite State Machines. *Information and Control*, Vol. 63, No. 3, 1984, 308-320.
8. W. Peng and S. Purushothaman, Analysis of a Class of Communicating Finite States Machines. *Acta Informatica*, Vol. 29, 1992, 499-522.
9. W. Peng and S. Puroshothaman, Data Flow Analysis of Communicating Finite State Machines. *ACM Transaction on Programming Languages and Systems*, Vol. 13, No. 3, 1991, 339-442.
10. L. Rosier and M. Gouda, Deciding Progress for a Class of Communicating Finite State Machines. In *"Proceedings, 18th Annual Conference on Information Sciences and Systems"*, Princeton, N.J., Mar. 1984, 663-667.
11. L. Rosier and H. Yen, Boundedness, Empty Channel Detection, and Synchronization for Communicating Finite Automata. *Theoretical Computer Science*, Vol. 44, No. 1, 1986, 69-105.
12. R. Howell, L. Rosier, D. Huynh and H. Yen, Some Complexity Bounds for Problems Concerning Finite and 2-Dimensional Vector Addition Systems with States. *Theoretical Computer Science*, Vol. 46, 1986, 107-140.
13. L. Rosier and H. Yen, A Multiparameter Analysis of the Boundedness Problem for Vector Addition Systems. *J. of Computer and System Sciences*, Vol. 32, No. 1, 1986, 105-135.
14. L. Rosier and H. Yen, Logspace Hierarchies, Polynomial Time and the Complexity of Fairness Problems Concerning ω-Machines. *SIAM J. Computing*, Vol. 16, No. 5, (1987).
15. N. Jones, L. Landweber, and Y. Lien, Complexity of some problems in Petri nets. *Theoretical Computer Science*, Vol. 4, (1977), 277-299.