Bounded Self-Stabilizing Petri Nets

Ludmila Cherkasova
Hewlett-Packard Laboratories
1501 Page Mill Road, Bldg. 1U-14
P.O. Box 10490
Palo Alto, CA 94303, USA

Rodney R. Howell
Dept. of Computing and Information Sciences
Kansas State University
Manhattan, KS 66506, USA

Louis E. Rosier[†]
Dept. of Computer Sciences
The University of Texas at Austin
Austin, TX 78712, USA

ABSTRACT: We investigate the property of self-stabilization in bounded Petri nets. We give characterizations for both self-stabilizing bounded ordinary Petri nets (i.e., Petri nets without multiple arcs) and self-stabilizing bounded general Petri nets (i.e., Petri nets with multiple arcs). These characterizations allow us to determine the complexity of deciding self-stabilization for each of these classes. In particular, we show the self-stabilization problem to be PTIME-complete for bounded ordinary Petri nets and PSPACE-complete for bounded general Petri nets.

Keywords: Self-stabilization, bounded Petri nets, computational complexity.

[†]Louis Rosier passed away on May 6, 1991, before the final version of this paper was complete.

CONTENTS

0. Introduction

1. Basic definitions and notation

2. Self-stabilizing Petri nets: definition and elementary properties

3. A characterization of self-stabilizing bounded ordinary Petri nets

4. A characterization of self-stabilizing bounded general Petri nets

5. Conclusion

0 Introduction

A system is said to be self-stabilizing if starting from any configuration the system is guaranteed to reach a "legal" configuration. The motivation behind this concept is that if, due to some unpredictable error, the system were to reach an "illegal" configuration, it would eventually correct itself, returning to some "legal" configuration. Thus, self-stabilizing systems are in some sense more robust than those that are not self-stabilizing, due to their inherent fault-tolerant nature. Self-stabilization has gained much attention in computer science research, particularly in the area of fault-tolerant distributed computing; see, e.g., [Dij73, Dij74, Lam86, BYC88, BP89, BGW89, Mul89, Gou90, GHR90].

In [GHR90], a number of curious properties were shown regarding simulations of self-stabilizing systems. For example, it was shown that there can be no self-stabilization-preserving simulation of vector addition systems with states (VASSs) by Petri nets.[1] This result is quite surprising, since most properties, such as deadlock freedom, liveness, and fair nontermination, can be preserved by simulations in either direction (see, e.g., [HP79]). The proof of this result relies on a rather interesting property of self-stabilizing Petri nets also shown in [GHR90]: if a self-stabilizing Petri net has an infinite computation, then all places are simultaneously unbounded.

The fact that any unbounded self-stabilizing Petri net must be unbounded in all places simultaneously suggests that unbounded self-stabilizing Petri nets have rather unusual properties. Indeed,

[1]We should point out that the definition of self-stabilization used in [GHR90] is a somewhat restricted definition in which the "legal" states are defined to be those reachable from the initial marking. It is often the case in self-stabilizing systems that the set of legal states is exactly the set of reachable states; see, e.g., [Dij73]. In order to avoid problems in specifying the set of legal states, we adopt this definition in this paper. See [GHR90] for a more detailed discussion.

we are as yet unable to give a good characterization of unbounded self-stabilizing Petri nets. In fact, the best bounds we have been able to derive for deciding whether an arbitrary Petri net is self-stabilizing are an upper bound of Π_2 (the second level of the arithmetic hierarchy) and a lower bound of exponential space. As a contrast, it is not too difficult to show that the same problem for Turing machines with empty input is complete for Π_2. Such a wide disparity between the known upper and lower bounds suggests that there is much to be learned concerning the properties of self-stabilizing Petri nets.

In this paper, we restrict our attention to self-stabilizing bounded Petri nets. We also consider separately a proper subclass: self-stabilizing bounded ordinary Petri nets (self-stabilizing bounded Petri nets in which no multiple arcs are allowed). Normally, not much distinction is made between ordinary Petri nets and general Petri nets, since there is a straightforward simulation of general Petri nets by ordinary Petri nets; however, this simulation does not preserve self-stabilization. In fact, we show in this paper that self-stabilizing bounded ordinary Petri nets have a much simpler characterization and are much easier to recognize than self-stabilizing bounded general Petri nets. Specifically, we give for each of these classes a characterization that allows us to determine the complexity of deciding self-stabilization. Furthermore, our algorithms for deciding self-stabilization are such that they reject all unbounded Petri nets (along with those that are not self-stabilizing), so boundedness does not need to be decided first. This last fact is important, since the boundedness problem for Petri nets requires exponential space [Lip76, Rac78].

The characterization we give for self-stabilizing bounded ordinary Petri nets is a structural characterization; i.e., it is independent of the initial marking. Thus, self-stabilization is a structural property of bounded ordinary Petri nets. This fact is important because Petri nets that can be characterized in terms of structure are typically easier to analyze than those that cannot (see, e.g., [JLL77, HRY87, HR88, HRY89, HJR91]). Furthermore, our characterization gives a close relationship between self-stabilizing bounded ordinary Petri nets and structurally bounded Petri nets — a class that has already been studied extensively (see, e.g., [Sil85, Mur89]). We use our characterization to give a polynomial time algorithm for determining self-stabilization, then show the problem to be PTIME-complete with respect to logspace many-one reductions.

For bounded general Petri nets, we show that self-stabilization is not a structural property; i.e., it is dependent upon the initial marking. Thus, a structural characterization is clearly impossible for this class. Therefore, let us consider what type of characterization would be more useful than the definition. Somewhat informally, the definition of self-stabilization states that every firing sequence occurring from each marking must eventually reach some marking reachable from the initial marking. Notice that this definition involves a universal quantifier over an infinite set. On the other hand, we give a characterization consisting of two properties, one of which is structural (i.e., independent of the initial marking), and the other of which gives a finite set of markings that must be reachable. This characterization is clearly more useful than the definition; however, reachability analyses are typically very costly. In particular, even for bounded Petri nets, the reachability problem requires at least exponential space [Lip76]; furthermore no primitive recursive algorithm is known (cf. [May84, Kos82, Lam88]). To overcome this obstacle, we show that for Petri nets satisfying the structural part of our characterization, reachability can be decided in PSPACE. Using this fact, we show the problem of determining self-stabilization for bounded general Petri nets to be PSPACE-complete.

The remainder of this paper is organized as follows. In Section 2, we give the basic definitions and notation to be used throughout the paper. In Section 3, we present some elementary properties of self-stabilizing Petri nets. In Section 4, we give a characterization of self-stabilizing bounded ordinary Petri nets and show the problem of deciding whether a given bounded ordinary Petri net is self-stabilizing to be PTIME-complete. In Section 5, we give a characterization of self-stabilizing bounded general Petri nets and show that the problem of deciding self-stabilization for this class is PSPACE-complete. We conclude in Section 6 with a discussion of possible extensions to this work.

1 Basic definitions and notation

A net is a tuple: $N = (P, T, F, W)$, where

- P is a finite set of places,
- T is a finite set of transitions,
- $F \subseteq (P \times T) \cup (T \times P)$ is an incidence relation, and
- $W : F \rightarrow \mathbf{N} \backslash \{0\} = \{1, 2, 3, \ldots\}$ is an arc weight function.

Graphically, the places are represented by circles, the transitions by boxes or bars, the incidence relation F by arcs connecting places with transitions and transitions with places, and the arc weight function by the multiplicity of the arcs. A Petri net is a pair (N, M_0) where

- $N = (P, T, F, W)$ is a net, and
- $M_0 : P \rightarrow \mathbf{N} = \{0, 1, 2, \ldots\}$ is an initial marking.

$\mathbf{0}$ will denote the zero marking; i.e., $\mathbf{0}(p) = 0$ for all $p \in P$. Graphically, the initial marking is represented by tokens (dots) inside the circles. A Petri net with multiplicity of arcs all equal to 1 is called an ordinary Petri net. We will use the term general Petri net to explicitly indicate that multiple arcs are allowed.

On the basis of incidence relation F and weight function W, we define an incidence function $F : (P \times T) \cup (T \times P) \to \mathbf{N}$ as follows:

$$F(x, y) = \begin{cases} n & \text{if } xFy \land (W(x, y) = n) \\ 0 & \text{if } \neg(xFy). \end{cases}$$

For any net element x ($x \in P \cup T$) we will denote:

- ${}^{\bullet}x = \{y \mid yFx\}$ — the set of its input elements,
- $x^{\bullet} = \{y \mid xFy\}$ — the set of its output elements.

A transition t is enabled in the net N at marking M (written as $M[t\rangle$) iff $\forall p \in {}^{\bullet}t : M(p) \geq F(p, t)$. For ordinary Petri nets this condition means that each input place of transition t has at least one token inside, i.e., a non-zero marking. If a transition t is enabled at M, then t may occur (or can fire), changing the marking M to the marking M' by the following rule:

$$\forall p \in P : M'(p) = M(p) - F(p, t) + F(t, p);$$

we then write: $M[t\rangle M'$. A marking M' is reachable from the marking M in the net N iff there exists a sequence of transitions (firing sequence) $\tau = t_1 t_2 \ldots t_k$ and a sequence of markings $M_1, \ldots, M_{k-1}$ such that

$$M[t_1\rangle M_1[t_2\rangle M_2 \ldots [t_k\rangle M'.$$

In this case, we will write: $M[\tau\rangle M'$, or $M[\ \rangle M'$ if τ is not essential. To simply indicate that τ can fire from M, we will write $M[\tau\rangle$. If $\tau = t_1 t_2 ...$ is an infinite firing sequence and there exists an infinite sequence of markings $M_1, M_2, ...$ such that

$$M[t_1\rangle M_1[t_2\rangle M_2 ...,$$

then we also write $M[\tau\rangle$. The reachability set of (N, M_0), denoted $R(N, M_0)$, is the set $\{M \mid M_0[\ \rangle M\}$, or the set of all its reachable markings. For a Petri net (N, M_0) we define the reachability tree $Tr(N, M_0)$ as follows. The root of the tree is labeled M_0. Let v be any node of the tree, and let the label of v be some marking M. Suppose k different transitions are enabled at M. Then v has k children, one for each transition enabled at M, and each child is labeled by the marking resulting from the firing of the respective transition at M. For the Petri net (N_1, M_{01}) shown in Figure 1, its reachability tree is represented in Figure 2.

A marking M is called a dead marking for a net N iff there are no enabled transitions in (N, M). We will denote by $R_d(N)$ the set of all dead markings for a net N, and by $R_d(N, M_0)$ the set of all dead markings in $R(N, M_0)$. For example, the markings $M_d^1 = (2, 0, 0, 1, 0)$, $M_d^3 = (1, 0, 0, 1, 1)$ and $M_d^5 = (0, 0, 0, 1, 2)$ (see reachability tree in Figure 2) are dead markings for the net (N_1, M_{01})

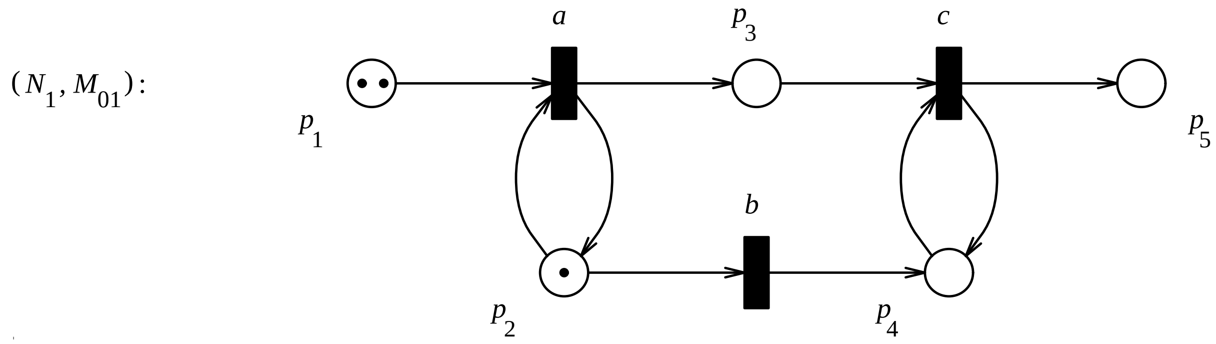


Figure 1.

shown in Figure 1. Moreover, $R_d(N_1, M_{01}) = \{M_d^1, M_d^3, M_d^5\}$. The set of all dead markings (i.e., independent of the initial marking) for the net N_1 shown in Figure 1 is given by:

$$R_d(N_1) = \{(k_1, 0, 0, k_2, k_3) \mid k_1, k_2, k_3 \geq 0\} \cup \{(k_1, 0, k_2, 0, k_3) \mid k_1, k_2, k_3 \geq 0\}.$$

Let τ be a firing sequence, and let $|\tau|$ denote its length. If τ is infinite then $|\tau| = \infty$. Let us consider the markings which are reachable in the net (N, M_0) via firing sequences τ with $|\tau| = k$ together with dead markings reachable with $|\tau| \leq k$. More precisely, let

$$\begin{aligned} R^k(N, M_0) &= \{M \mid (M_0[\tau\rangle M) \wedge (|\tau| = k)\} \cup \\ &\quad \{M \mid (M_0[\tau\rangle M) \wedge (M \in R_d(N, M_0)) \wedge (|\tau| \leq k)\} \end{aligned}$$

We will call $R^k(N, M_0)$ the reachability set of level k for (N, M_0). The utility of this definition will become apparent when we introduce self-stabilizing Petri nets.

In graphical representation, the markings from $R^k(N, M_0)$ are exactly the nodes in the reachability tree for (N, M_0) at level k (i.e., nodes which are reachable by paths with length equal to k), plus all leaves above level k (i.e., dead markings which are reachable by paths with length no more than k). For example, for Petri net (N_1, M_{01}) shown in Figure 1 with reachability tree shown in Figure 2 its reachability sets of levels 2 and 4 are the following:

$$R^2(N_1, M_{01}) = \{M_1^2, M_2^2, M_d^1\}, \qquad R^4(N_1, M_{01}) = \{M_1^4, M_d^3, M_d^1\}.$$

Let us define $R^{\geq k}(N, M_0) = \{M' \mid (M[\tau\rangle M') \wedge (M \in R^k(N, M_0))\}$. In graphical representation, the markings from $R^{\geq k}(N, M_0)$ are the nodes in the reachability tree for (N, M_0) at or below level k (i.e., the nodes reachable from the root by paths with length $\geq k$; the root is considered to be the topmost node), plus all leaves. For example, for Petri net (N_1, M_{01}) in Figure 1 with reachability tree in Figure 2, its reachability trees of level 2 and 4 are presented in Figure 3a, b. Let us denote by $R^{<k}(N, M_0) = \{M \mid (M_0[\tau\rangle M) \wedge (M \notin R_d(N, M_0)) \wedge |\tau| < k\}$; i.e., the markings

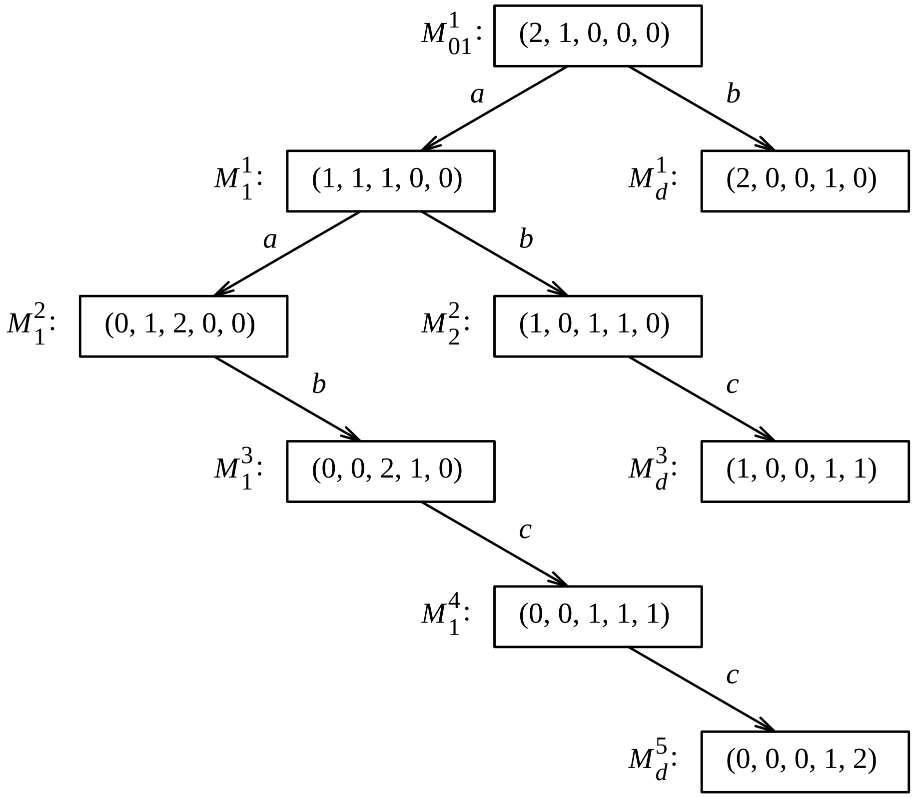


Figure 2.

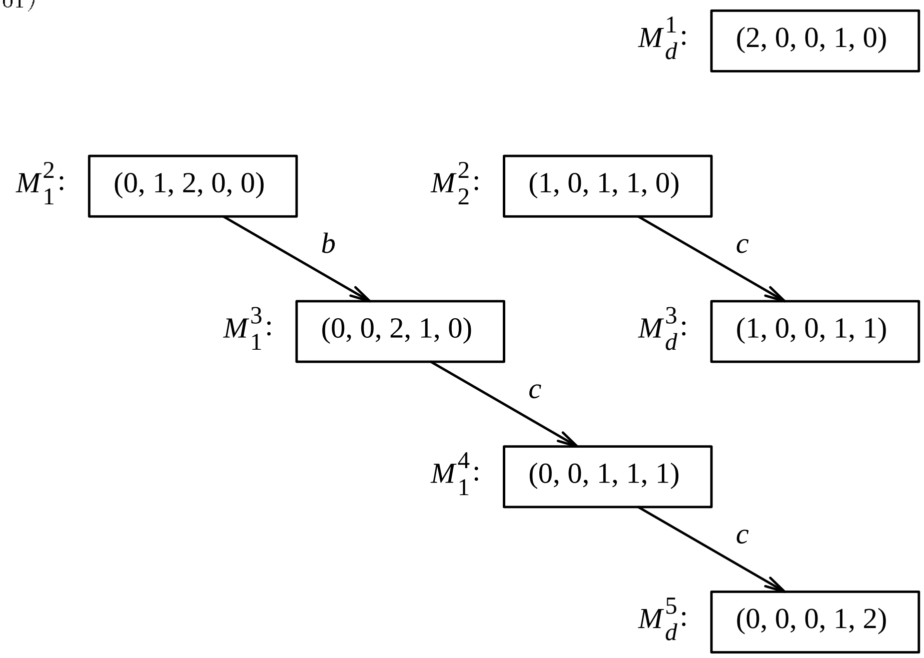
of $R^{<k}(N, M_0)$ belonging to the reachability tree strictly above level k, minus all leaves. It is clear that $R(N, M_0) = R^{<k}(N, M_0) \cup R^{\geq k}(N, M_0)$.

Petri net (N_1, M_{01}) shown in Figure 1 has a finite set of finite firing sequences (i.e., it defines finite behavior). As a result, the reachability set and reachability tree are finite. For Petri net (N_2, M_{02}) shown in Figure 4a the reachability set $R(N_2, M_{02})$ is finite: $R(N_2, M_{02}) = \{(1,0),(0,1)\}$, but its reachability tree (shown in Figure 4b) is infinite (because of the infinite firing sequence $\tau = aa\ldots a\ldots$). For Petri net (N_3, M_{03}) shown in Figure 5a, both the reachability set and the reachability tree (shown in Figure 5b) are infinite due to the fact that place p_3 is unbounded (i.e., can get arbitrarily many tokens as a marking). A place $p \in P$ in Petri net (N, M_0) is <u>bounded</u> iff $(\exists k \in \mathbf{N})(\forall M \in R(N, M_0)): \quad M(p) \leq k$. Otherwise, place p is called <u>unbounded</u>. A Petri net is <u>bounded</u> iff all its places are bounded.

It is clear that a bounded Petri net has a finite reachability set, and conversely, if a Petri net is unbounded, then its reachability set is infinite. It is well-known that the boundedness problem for Petri nets is decidable [KM69]; in fact, the problem is complete for DSPACE($2^{poly(n)}$) with respect to logspace many-one reductions [Lip76, Rac78] (see also [HR87]).

The set $\mathcal{L}(N, M_0) = \{\tau \in T^* \mid M_0[\tau\rangle\}$ is called the <u>free language</u> of Petri net (N, M_0), or the set of all its firing sequences.

a) $R^{\geq 2}(N_1, M_{01})$



b) $R^{\geq 4}(N_1, M_{01})$

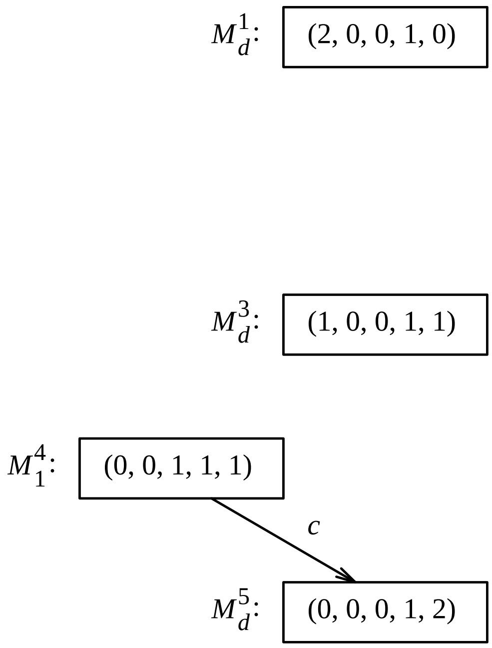


Figure 3.

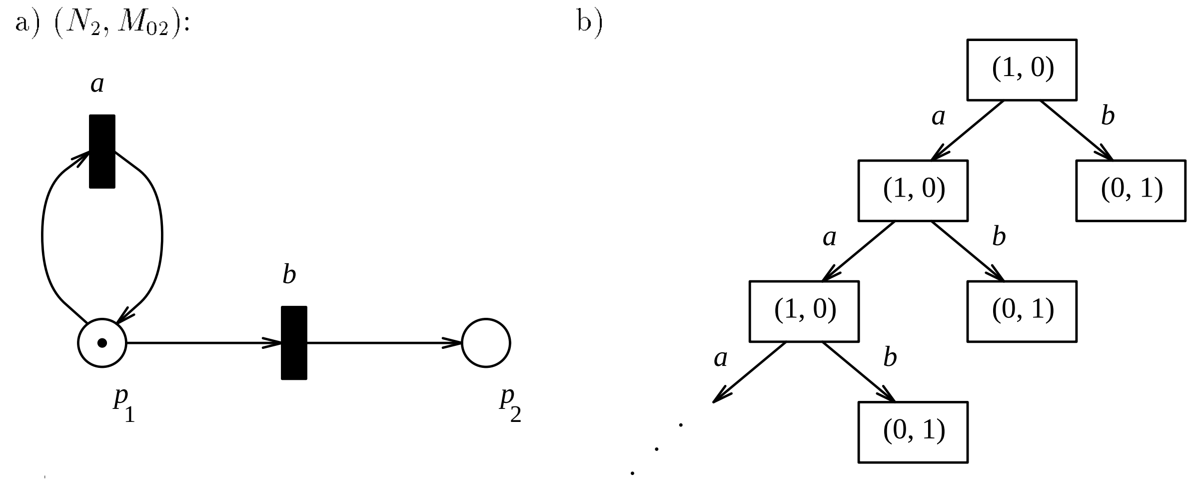


Figure 4.

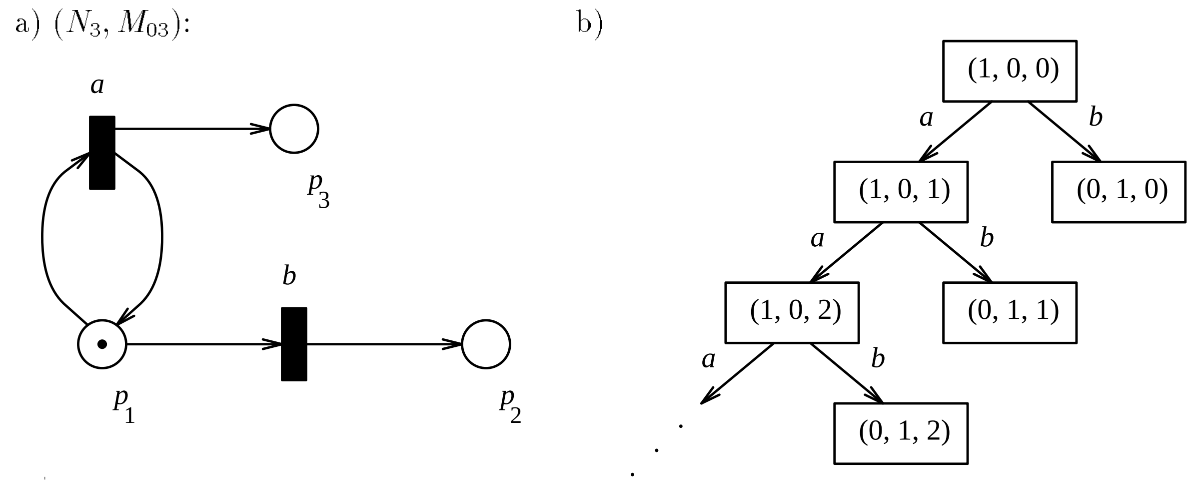


Figure 5.

2 Self-stabilizing Petri nets: definition and elementary properties

In this section, we introduce self-stabilizing Petri nets and give some of their elementary properties. We then use these properties to motivate the study of self-stabilizing bounded Petri nets.

A Petri net (N, M_0) is self-stabilizing (abbreviation: ss) iff $\forall M \exists k : R^k(N, M) \subseteq R(N, M_0)$. This definition is equivalent to the following: $\forall M \; \exists k : R^{\geq k}(N, M) \subseteq R(N, M_0)$. Thus, if we consider the net N with an arbitrary initial marking M, then sooner or later all the firing sequences of the net N will reach a marking in the reachability set of (N, M_0). In this sense, the behavior of the net N under an arbitrary initial marking will always stabilize to the behavior of the net N with initial marking M_0. The self-stabilization problem is to determine, for a given Petri net (N, M_0), whether (N, M_0) is self-stabilizing. The Petri nets shown in Figure 6a, b are self-stabilizing, whereas the Petri nets in Figures 1, 4a, 5a are not ss. The following lemma holds for self-stabilizing systems in general.

a) (N_4, M_{04}) : b) (N_5, M_{05}) :

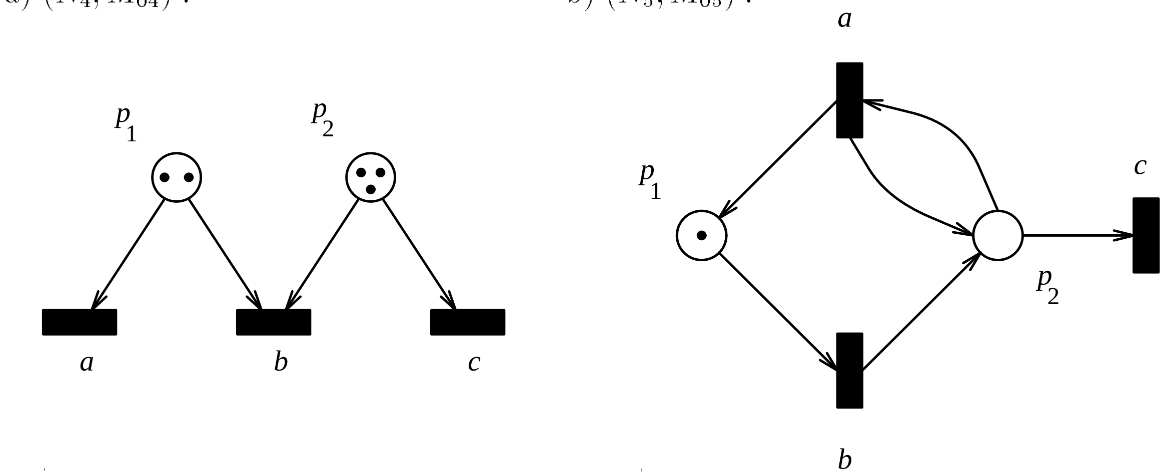


Figure 6.

Lemma 3.1 (from [GHR90]): If (N, M_0) is a ss Petri net then $R_d(N) \subseteq R(N, M_0)$; i.e., all dead markings for N are reachable in (N, M_0).

Proof. Since (N, M_0) is a ss Petri net, then $\forall M \exists k : R^{\geq k}(N, M) \subseteq R(N, M_0)$. By definition, $R^{\geq k}(N, M) = \{M' \mid (M[\tau\rangle M') \wedge |\tau| \geq k\} \cup R_d(N, M)$. Hence, $\forall M : R_d(N, M) \subseteq R(N, M_0)$. Thus $R_d(N) \subseteq R(N, M_0)$. ∎

The following lemma gives an interesting property of self-stabilizing Petri nets with infinite firing sequences.

Lemma 3.2 (from [GHR90]): Let (N, M_0) be a *ss* Petri net. If there exists an infinite sequence τ in (N, M_0) (i.e., $\tau \in \mathcal{L}(N, M_0)$) then all places of the net (N, M_0) are unbounded.

Proof. Let us suppose, to the contrary, that some place p in (N, M_0) is bounded. By definition, $\exists k \in \mathbf{N}(\forall M \in R(N, M_0) : M(p) < k)$. Let us consider the same net N with the following initial marking $M_0 + K$, where

$$K(p_i) = \begin{cases} k & \text{if } p_i = p \\ 0 & \text{otherwise.} \end{cases}$$

Now let us compare the markings reachable in (N, M_0) and $(N, M_0 + K)$ through the same infinite firing sequence $\tau = t_1 t_2 t_3 \ldots t_n \ldots$. (By the monotonicity property of Petri nets, if $\tau \in \mathcal{L}(N, M_0)$, then $\forall M \geq M_0 : \tau \in \mathcal{L}(N, M)$). So, if $M_0[t_1\rangle M_1[t_2\rangle M_2 \ldots [t_n\rangle M_n \ldots$, then $(M_0 + K)[t_1\rangle(M_1 + K)[t_2\rangle(M_2 + K) \ldots [t_n\rangle(M_n + K) \ldots$. For any marking $M_i + K$ in $(N, M_0 + K)$, place p has at least k tokens. However, $\forall M \in R(N, M_0) : M(p) < k$. Hence $\neg\exists n : R^n(N, M_0 + K) \subseteq R(N, M_0)$. This contradicts the lemma condition that (N, M_0) is a *ss* Petri net; hence the initial assumption is false, and all the places of the net (N, M_0) are unbounded. ∎

Since the existence of one unbounded place immediately implies the existence of an infinite firing sequence, we have the following corollary.

Corollary 3.3: Let (N, M_0) be a *ss* Petri net. If there exists a place p in (N, M_0) which is unbounded, then all the places of (N, M_0) are unbounded.

Lemma 3.2 can be restated as follows:

If (N, M_0) is a *ss* Petri net then

(1) either all its places are bounded and there are no infinite firing sequences in it, i.e., $\forall \tau \in \mathcal{L}(N, M_0) : |\tau| < \infty$, or

(2) all places of the net (N, M_0) are unbounded.

The remainder of the paper is devoted to investigation of the first case: *ss* bounded Petri nets. We will consider separately the properties of *ss* bounded ordinary Petri nets and *ss* bounded general Petri nets. First, however, we give an important result that holds for all *ss* bounded Petri nets.

Lemma 3.4: Let (N, M_0) be a *ss* bounded Petri net. Then for arbitrary marking M,

(1) (N, M) is bounded, and

(2) $\forall \tau \in \mathcal{L}(N, M) : \ |\tau| < \infty$.

Proof. Since (N, M_0) is a bounded Petri net, the reachability set $R(N, M_0)$ is finite. Since (N, M_0) is a *ss* Petri net, $\forall M \exists k : R^{\geq k}(N, M) \subseteq R(N, M_0)$. Hence for arbitrary marking M the reachability set $R(N, M)$ is finite. Therefore for arbitrary marking M the net (N, M) is bounded. By using an argument similar to the proof of Lemma 3.2, we can now conclude that $\forall \tau \in \mathcal{L}(N, M)$: $|\tau| < \infty$. ■

Part (1) of Lemma 3.4 states that any *ss* bounded Petri net is structurally bounded (see, e.g., [Sil85, Mur89]). Part (2) gives a stronger property, which is important enough to give a name. We say that a net N is structurally terminating if for every marking M, $\forall \tau \in \mathcal{L}(N, M) : |\tau| < \infty$. Thus, every *ss* bounded Petri net is structurally terminating.

3 A characterization of self-stabilizing bounded ordinary Petri nets

In this section, we examine self-stabilizing bounded ordinary Petri nets. We first characterize the set by showing that a bounded ordinary Petri net is *ss* iff it is structurally terminating and for every place there is a transition whose set of input places is exactly that place. Thus, we show self-stabilization to be a structural property of bounded ordinary Petri nets, independent of their initial marking (recall from Lemma 3.4 that bounded *ss* Petri nets must be structurally bounded). We then use our characterization to give a polynomial-time algorithm for deciding self-stabilization for bounded ordinary Petri nets. We conclude the section by showing this problem to be PTIME-complete with respect to logspace many-one reductions.

Lemma 4.1: Let (N, M_0) be a *ss* bounded ordinary Petri net. Then $R_d(N) = R_d(N, M_0) = \{\mathbf{0}\}$.

Proof. First, let us notice that if $M \in R_d(N)$ then $\forall \tilde{M} : \tilde{M} \leq M$ we have $\tilde{M} \in R_d(N)$. Since (N, M_0) is a *ss* bounded Petri net, there are no infinite firing sequences in it (see Lemma 3.2). Hence, the reachability tree for (N, M_0) is finite, and the set of dead markings for (N, M_0) is not empty. Therefore, $\mathbf{0} \in R_d(N)$. Let us suppose that there exists a dead marking M of N which is not $\mathbf{0}$; i.e., $M \in R_d(N)$ and $\exists \tilde{p} \in P : M(\tilde{p}) = k > 0$. Since (N, M_0) is bounded, the place $\tilde{p}$ is bounded, and $\exists \tilde{k} \forall \tilde{M} \in R(N, M_0) : \tilde{M}(\tilde{p}) < \tilde{k}$. Let us consider the net N with the following marking $M + \tilde{K}$, where

$$\tilde{K}(p_i) = \begin{cases} \tilde{k} \text{ if } p_i = \tilde{p}, \\ 0 \text{ otherwise.} \end{cases}$$

Note that for each place p_i, $M(p_i) = 0$ iff $(M + \tilde{K})(p_i) = 0$, so since (N, M_0) is an ordinary Petri net, any transition enabled at $M + \tilde{K}$ is also enabled at M. Hence, $M + \tilde{K}$ is also a dead marking for N. But $M + \tilde{K} \notin R(N, M_0)$. Then by Lemma 3.1, (N, M_0) is not a *ss* Petri net — a contradiction.

Hence, $R_d(N) = R_d(N, M_0) = \{\mathbf{0}\}$. ■

Corollary 4.2: Let (N, M_0) be a bounded ordinary Petri net. (N, M_0) is *ss* iff

(1) N is structurally terminating; and

(2) $R_d(N) = \{\mathbf{0}\}$.

Proof.

$\Rightarrow$

Immediately follows from Lemma 3.4 and Lemma 4.1.

$\Leftarrow$

Since for any marking M there are no infinite firing sequences in (N, M), i.e., $\forall \tau \in \mathcal{L}(N, M) : |\tau| < \infty$, it means that reachability tree for any (N, M) is finite. Since $R_d(N) = \{\mathbf{0}\}$, $\forall M : R_d(N, M) = \{\mathbf{0}\}$; i.e., all the leaves in the reachability tree for any (N, M) are the $\mathbf{0}$ marking. Thus, at least at the last level all the reachability trees are the same. Hence, (N, M_0) is a *ss* Petri net. ■

Since the conditions of Corollary 4.2 are both independent of the initial marking, we also have the following corollary.

Corollary 4.3: Let (N, M_0) be a *ss* bounded ordinary Petri net. Then for any marking M the net (N, M) is *ss*.

In what follows, we will show how to test the conditions of Corollary 4.2 in polynomial time. Condition (1) can be tested by determining the existence of an integer solution to a system of homogeneous linear inequalities, which may be done using (rational) linear programming techniques. In order to give an efficient means of testing condition (2), we now show that if condition (1) holds, then condition (2) is equivalent to a simple syntactic property of the net.

Lemma 4.4: Let N be a structurally terminating ordinary net. Then $R_d(N) = \{\mathbf{0}\}$ iff for each net place p there exists a transition t such that ${}^{\bullet}t = \{p\}$.

Proof.

$\Leftarrow$

Since N is structurally terminating, there cannot be any transition enabled at $\mathbf{0}$. On the other hand, if some place p has a nonzero marking, then the transition t such that ${}^{\bullet}t = \{p\}$ is enabled. Therefore, $R_d(N) = \{\mathbf{0}\}$.

$\Rightarrow$

Suppose, to the contrary, that $R_d(N) = \{\mathbf{0}\}$ and for some place $\tilde{p}$ there is no transition t such that ${}^{\bullet}t = \{\tilde{p}\}$; i.e., $(\forall t \in \tilde{p}^{\bullet})(\exists p \neq \tilde{p}) : p \in {}^{\bullet}t$. Let us consider the following marking $\tilde{M}$:

$$\tilde{M}(p_i) = \begin{cases} 1 & \text{if } p_i = \tilde{p} \\ 0 & \text{if } p_i \neq \tilde{p} \end{cases}$$

The marking $\tilde{M}$ is a dead marking for the net N, i.e., $\tilde{M} \in R_d(N)$. However, $\tilde{M} \neq \mathbf{0}$ — a contradiction. ■

The following theorem now follows immediately from Corollary 4.2 and Lemma 4.4.

Theorem 4.5: Let (N, M_0) be a bounded ordinary Petri net. (N, M_0) is *ss* iff

(1) N is structurally terminating, and

(2) $\forall p \in P \exists t \in T : {}^{\bullet}t = \{p\}$.

We can use Theorem 4.5 to show that the self-stabilization problem for bounded ordinary Petri nets is solvable in polynomial time. First, we need to introduce some notation. Let $N = (P, T, F, W)$ be a net. The incidence matrix C for the net N is defined as $C(p,t) = F(t,p) - F(p,t)$. For example, the net N_5 in Figure 6b is characterized by the following incidence matrix:

p \ t	a	b	c
p_1	1	-1	0
p_2	0	1	-1

Theorem 4.6: The self-stabilization problem for bounded ordinary Petri nets is solvable in polynomial time.

Proof. We will show that conditions (1) and (2) from Theorem 4.5 can be verified in polynomial time. Clearly, (2) can be verified in polynomial time. We verify (1) as follows. Let N be an arbitrary bounded ordinary net, and let C be the incidence matrix of N. It is easily seen that N is structurally terminating iff there is no nonnegative, nonzero integer solution to the inequality $Cx \geq \mathbf{0}$. We first note that any rational solution to this inequality can be transformed into an integer solution by multiplying by a common denominator of the components of the solution vector. By using either Khachian's algorithm [Kha79] or Karmarkar's algorithm [Kar84] to determine whether there is a nonnegative, nonzero solution to the above inequality, we can therefore decide self-stabilization in polynomial time. ■

Note that if the algorithm in the above proof is given an unbounded ordinary Petri net, it will be rejected because it is not structurally terminating; thus, given any ordinary Petri net, the algorithm will determine, in polynomial time, whether the Petri net is both bounded and self-stabilizing. We also note that as an alternative algorithm, we could first convert the net N into a net N' that is structurally bounded iff N is structurally terminating; Silva [Sil85] gives a polynomial-time algorithm for deciding whether a net is structurally bounded.

We conclude this section by showing the self-stabilization problem for bounded ordinary Petri nets to be PTIME-complete. In order to show this problem to be PTIME-hard, we use a reduction from the complement of the path system problem, which is known to be PTIME-complete [JL77]. A path system is a 4-tuple $P = (X, R, S, T)$, where X is a finite set of nodes, $S(\subseteq X)$ is the set of starting nodes, $T(\subseteq X)$ is the set of terminal nodes, and $R(\subseteq X \times X \times X)$ is the set of rules. A sequence of rules ρ is said to be a path if

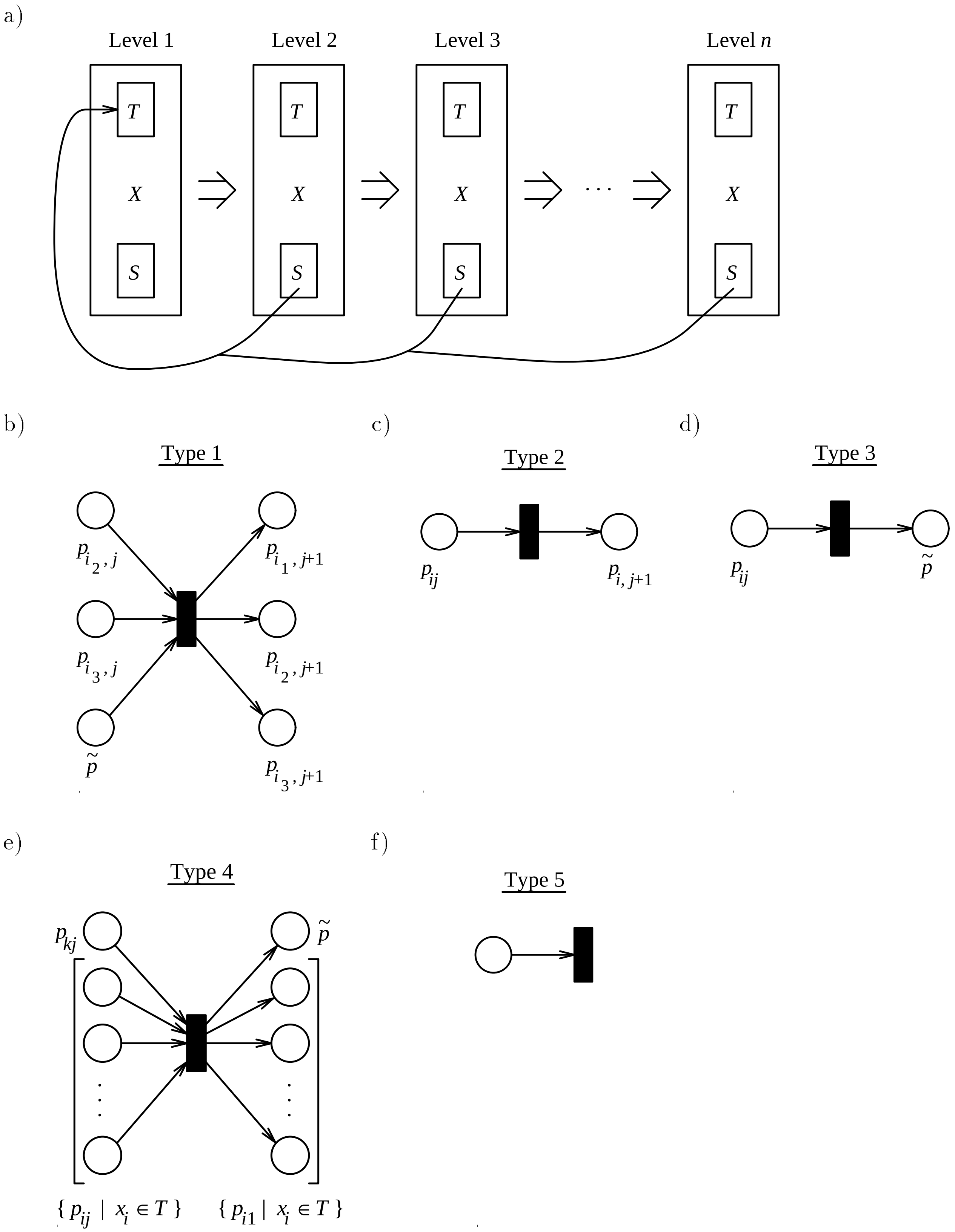
(1) $\rho = \epsilon$; or

(2) $\rho = \rho'(x, y, z)$, where ρ' is a path such that for every $w \in \{y, z\} \backslash T$, some rule (w, y', z') occurs in ρ'.

ρ is said to admit $x \in X$ if either $x \in T$ or x occurs in some rule in ρ. A node x is said to be admissible if there is some path admitting x. We say that P has a solution if there is an admissible node in S.

Theorem 4.7: The self-stabilization problem for bounded ordinary Petri nets is complete for PTIME with respect to logspace many-one reductions.

Proof. We will show that the complement of the path system problem logspace reduces to the self-stabilization problem for bounded ordinary Petri nets; the theorem will then follow from Theorem 4.6. Let $P = (X, R, S, T)$ be an arbitrary path system. Without loss of generality, we assume $S \cap T = \emptyset$. We will construct a structurally bounded net N such that conditions (1) and (2) of Theorem 4.5 hold iff P does not have a solution. (More precisely, condition (2) will always hold, and condition (1) will hold iff P does not have a solution). Let $X = \{x_1, x_2, ..., x_n\}$. We will construct N to consist of n "levels," each of which will contain places representing the nodes in X. A somewhat informal flow of control for N is shown in Figure 7a. Intuitively, when a new node is admitted, control passes to the next level. Once a node in S is admitted, control will be allowed to return to Level 1. Thus, N will have a structural loop iff P has a solution.



a)
Level 1
Level 2
Level 3
Level n
T
X
S
b)
Type 1
c)
Type 2
d)
Type 3
e)
Type 4
f)
Type 5

Figure 7.

More formally, for each $x_i \in X$ and $1 \leq j \leq n$, we define a place p_{ij}; we will say that p_{ij} is at Level j. In order to assure that the net is structurally bounded, we will construct N so that no transition increases the total number of tokens in the net. In order to be able to enforce this behavior, we need an additional place $\tilde{p}$ to be used as a pool of extra tokens. As we describe the transitions, it will be helpful to imagine an initial marking M_0 such that $M_0(p_{i1}) = 1$ for all $x_i \in T$, $M_0(\tilde{p}) = n$, and for all other places p, $M_0(p) = 0$. From this initial marking, p_{ij} will be able to receive a token if x_i is admitted by a path of length j, and only if x_i is admissible. We will classify the transitions of N into five types (see Figures 7b–7f). For each rule $(x_{i_1}, x_{i_2}, x_{i_3})$ and $1 \leq j \leq n-1$, we have the Type 1 transition shown in Figure 7b. The firing of this transition corresponds to the inclusion of $(x_{i_1}, x_{i_2}, x_{i_3})$ in a path. For each $x_i \in X$ and $1 \leq j \leq n-1$, we have the Types 2 and 3 transitions shown in Figures 7c and 7d. These transitions allow tokens from Level j to be either transferred to Level $j+1$ or returned to the pool of extra tokens. For each $x_k \in S$ and $2 \leq j \leq n-1$, we have the Type 4 transition shown in Figure 7e. Once a node in S has been admitted, these transitions allow tokens to be returned to the places in Level 1 corresponding to nodes in T. Finally, for each place in N, we have the Type 5 transition shown in Figure 7f. These transitions allow condition (2) of Theorem 4.5 to be satisfied. Clearly, this construction can be done using logarithmic space. We will now show that P has a solution iff N is not structurally terminating.

$\Rightarrow$

Suppose P has a solution. We will show how to construct an infinite firing sequence from M_0 (defined above). Let ρ be a shortest path admitting some node $x_k \in S$. Clearly, ρ contains no more than $n-1$ rules, and since $S \cap T = \emptyset$, ρ contains at least one rule. Let ρ_j, $0 \leq j \leq |\rho|$, denote the prefix of ρ of length j. We will now construct a sequence of markings $M_0[\ \rangle M_1[\ \rangle \cdots [\ \rangle M_{|\rho|}$ of N such that for $0 \leq j \leq |p|$:

(1) $M_j(p_{i,j+1}) = 1$ if x_i is admitted by ρ_j;

(2) $M_j(\tilde{p}) = n - j$; and

(3) for all other places p, $M_j(p) = 0$.

Clearly, all three conditions hold for $j = 0$. Suppose for some $j > 0$, M_{j-1} satisfies the above conditions. Let $(x_{i_1}, x_{i_2}, x_{i_3})$ be the jth rule in ρ. Then x_{i_2} and x_{i_3} are admitted by ρ_{j-1}. Therefore, from condition (1), $M_{j-1}(p_{i_2,j}) = M_{j-1}(p_{i_3,j}) = 1$. We can therefore fire a Type 1 transition removing the tokens from these places and placing tokens on $p_{i_1,j+1}$, $p_{i_2,j+1}$, and $p_{i_3,j+1}$. Note that this firing leaves $n-j$ tokens on $\tilde{p}$, and the places that now need tokens (to produce M_j) are exactly those place $p_{i,j+1}$ such that p_{ij} has a token. Therefore, by firing Type 2 transitions, we can reach M_j. Proceeding thus, we eventually arrive at $M_{|\rho|}$. In $M_{|\rho|}$, from condition (1), there is a token on $p_{k,|\rho|+1}$. Therefore, we may fire Types 3 and 4 transitions to return to M_0. We therefore have $M_0[\tau\rangle M_0$ for a nonempty firing sequence τ, so N is not structurally terminating.

$\Leftarrow$

Suppose N is not structurally terminating; i.e., there is a marking at which an infinite firing sequence can fire. From [KM69], there must exist markings M_1 and M_2 and a nonempty firing sequence τ such that $M_1[\tau\rangle M_2$ and $M_1 \leq M_2$. Since no transition in N increases the number of tokens in the net, $M_1 = M_2$, and there can be no Type 5 transitions in τ. Since $M_1 = M_2$, for every transition $t \in \tau$ and every place p, if $p \in {}^{\bullet}t$, there must be a transition $t' \in \tau$ with $p \in t'^{\bullet}$. Notice that all transitions of Types 1–4 have at least one input place other than $\tilde{p}$. Let j_0 be the lowest level such that some transition in τ has an input place in Level j_0. Then in τ, only Type 4 transitions can have output places Level j_0 (i.e., $j_0 = 1$). We conclude that τ must contain a Type 4 transition. This transition has as an input place p_{kj}, where $x_k \in S$. We will now show by induction on j that there is a transition in τ with p_{ij} as an input place only if x_i is admissible; it will then follow that x_k is admissible, and, hence, P has a solution. For $j = 1$, the only places p_{ij} that are output places are those such that $x_i \in T$; therefore, if a transition in τ has p_{i1} as an input place, x_i is admissible. We now inductively assume that for some $j > 1$, if there is a transition in τ with $p_{i,j-1}$ as an input place, then x_i is admissible. Suppose there is some transition t_1 in τ with $p_{ij} \in {}^{\bullet}t_1$. Then there must be some transition t_2 in τ with $p_{ij} \in t_2^{\bullet}$. Since $j > 1$, t_2 must be either of Type 1 or Type 2. Suppose t_2 is of Type 2. Then $p_{i,j-1} \in {}^{\bullet}t_2$, and from the induction hypothesis, x_i is admissible. On the other hand, suppose t_2 is of Type 1. Then $\{p_{i_2,j-1}, p_{i_3,j-1}\} \subseteq {}^{\bullet}t_2$, and $\{p_{i_1,j}, p_{i_2,j}, p_{i_3,j}\} = t_2^{\bullet}$, where $(x_{i_1}, x_{i_2}, x_{i_3}) \in R$ and $x_i \in \{x_{i_1}, x_{i_2}, x_{i_3}\}$. From the induction hypothesis, both x_{i_2} and x_{i_3} are admissible, so x_{i_1} is also admissible. Therefore, x_i is admissible. This completes the induction and the proof that P has a solution.

We have shown that P has a solution iff N is not structurally terminating. From Theorem 4.5, we now conclude that $(N, \mathbf{0})$ is ss iff P does not have a solution. ■

4 A characterization of self-stabilizing bounded general Petri nets

In this section, we examine self-stabilizing bounded general Petri nets. We first demonstrate that the characterization given for bounded ordinary Petri nets no longer holds, and that self-stabilization is not even a structural property when we allow multiple arcs. We then characterize this more general class by showing that a bounded Petri net is ss iff it is structurally terminating and every dead marking is reachable. We conclude by showing the self-stabilization problem for bounded general Petri nets to be PSPACE-complete; thus, allowing multiple arcs increases the complexity of the problem from PTIME to PSPACE, assuming these two complexity classes are not identical.

If we consider bounded general Petri nets, then Lemma 4.1, Corollary 4.2, and Theorem 4.5 no longer hold. For example, the net (N_7, M_{07}) in Figure 8a is *ss*, but $R_d(N_7) = R_d(N_7, M_{07}) = \{\mathbf{0}, (0,1), (0,2)\} \neq \{\mathbf{0}\}$. For the net (N_7, M'_{07}) in Figure 8b, the conditions of Theorem 4.5 hold, but (N_7, M'_{07}) is not *ss*. In particular, $R_d(N_7, M'_{07}) = \{\mathbf{0}, (0,1)\} \not\supseteq R_d(N_7)$; i.e., not all dead markings for N_7 are reachable in (N_7, M'_{07}). Thus, we can see that the conditions of Theorem 4.5 are not sufficient to guarantee self-stabilization for bounded general Petri nets; however, as we show in the lemma and theorem below, both conditions are still necessary. Furthermore, Figure 8 illustrates that self-stabilization for bounded general Petri nets is not a structural property; rather, it depends on the initial marking.

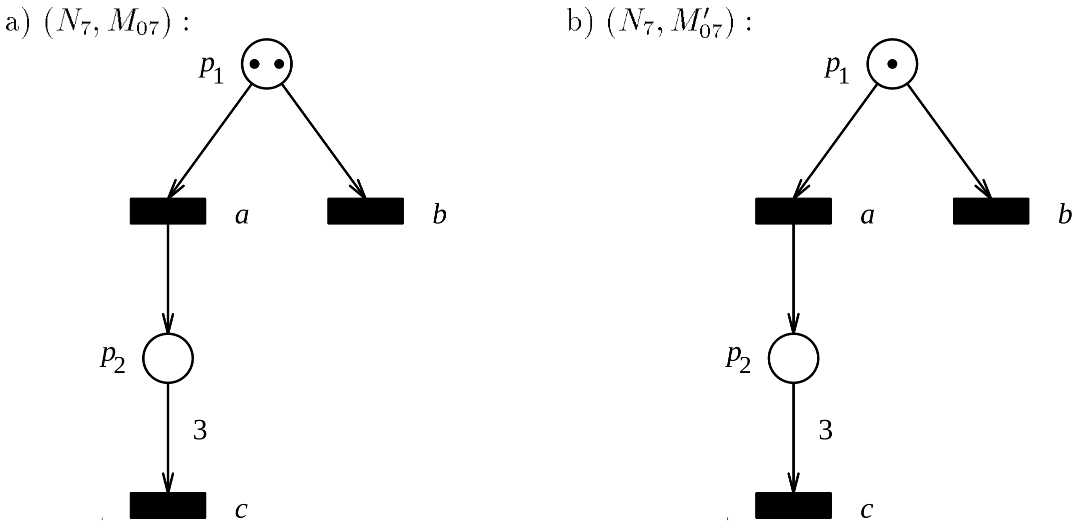


Figure 8.

Lemma 5.1: Let (N, M_0) be a *ss* bounded Petri net, $N = (P, T, F, W)$. Then $\forall p \in P\ \exists t \in T : {}^\bullet t = \{p\}$.

Proof. Let us suppose the contrary, that for some place $\tilde{p}$ there is no transition t such that ${}^\bullet t = \{\tilde{p}\}$, i.e., $\forall t \in \tilde{p}^\bullet\ \exists p \neq \tilde{p} : p \in {}^\bullet t$. Let k be such that $\forall M \in R(N, M_0) : \ M(\tilde{p}) < k$. Let us consider the following marking $\tilde{M}$.

$$\tilde{M}(p_i) = \begin{cases} k & \text{if } p_i = \tilde{p} \\ 0 & \text{if } p_i \neq \tilde{p} \end{cases}$$

The marking $\tilde{M}$ is a dead marking for the net N, i.e., $\tilde{M} \in R_d(N)$. However, $\tilde{M} \notin R(N, M_0)$; this contradicts Lemma 3.1. ∎

We now give a characterization for *ss* bounded general Petri nets.

Theorem 5.2: Let (N, M_0) be a bounded Petri net. (N, M_0) is *ss* iff

(1) N is structurally terminating, and

(2) $R_d(N) \subseteq R_d(N, M_0)$.

Proof.

$\Rightarrow$

Immediately follows from Lemma 3.1 and Lemma 3.4.

$\Leftarrow$

Let M be an arbitrary marking of N. Since N is structurally terminating, the Petri net (N, M) has no infinite firing sequences. Hence, the reachability tree for (N, M) is finite. The leaves in this tree are dead markings for the net N. But by condition (2), $R_d(N) \subseteq R_d(N, M_0)$. Hence, to satisfy the *ss* property:

$$\forall M \exists k \quad : \quad R^k(N, M) \subseteq R(N, M_0)$$

we can choose k which defines the last level in the reachability tree. ∎

Condition (1) of Theorem 5.2 may be tested in polynomial time as in Theorem 4.6 (that algorithm works for arbitrary nets). However, condition (2) appears to require a reachability analysis, and therefore seems to be much more difficult to decide. Recall that even for bounded Petri nets, the reachability problem requires at least exponential space [Lip76]; in fact, there is no primitive recursive algorithm known for the problem (cf. [May84, Kos82, Lam88]). On the other hand, from Theorem 5.2, we may restrict our attention to structurally terminating Petri nets. In what follows, we will show that if the necessary condition given in Lemma 5.1 holds, then each dead marking can be generated in PSPACE, and that if the Petri net is structurally terminating, reachability can be decided in PSPACE. From these two facts, we conclude that the self-stabilization problem for bounded Petri nets is in PSPACE. We then give strong evidence that this bound cannot be tightened: we show the problem to be PSPACE-complete.

Lemma 5.3: Let $N = (P, T, F, W)$ be a net such that $\forall p \in P \; \exists t \in T : {}^{\bullet}t = \{p\}$. Then $R_d(N) \subseteq \{M \mid M \leq M'\}$, where $M'(p) = \min\{W(p, t) \mid {}^{\bullet}t = \{p\}\} - 1$.

Proof. Let M_1 be any marking in $R_d(N)$. Suppose that for some place p, $M_1(p) > M'(p)$. Then there is a transition t such that ${}^{\bullet}t = \{p\}$ and $W(p, t) \leq M_1(p)$. Then t can fire at M_1 — a contradiction. ∎

Corollary 5.4: Let $N = (P, T, F, W)$ be a net such that $\forall p \in P \; \exists t \in T : {}^{\bullet}t = \{p\}$. Then each marking in $R_d(N)$ can be generated and written down in turn using only a polynomial amount of space.

It was shown in [How91] that the reachability problem for structurally bounded Petri nets is PSPACE-complete. Since any structurally terminating net is clearly structurally bounded, the following lemma is immediate.

Lemma 5.5: Let (N, M_0) be a structurally terminating Petri net, and let M be a marking of N. It can be decided in PSPACE whether $M \in R(N, M_0)$.

From Corollary 5.4 and Lemma 5.5, we can test condition (2) of Theorem 5.2 in PSPACE. We therefore have the following Theorem.

Theorem 5.6: The self-stabilization problem for bounded Petri nets is in PSPACE.

Again, as in the previous section, if we are given an arbitrary Petri net without knowing whether it is bounded, we can decide in PSPACE whether it is both bounded and self-stabilizing: any unbounded Petri net will be rejected because it is not structurally terminating.

We conclude by showing the self-stabilization problem for bounded Petri nets to be PSPACE-complete.

Theorem 5.7: The self-stabilization problem for bounded Petri nets is PSPACE-complete with respect to logspace many-one reductions.

Proof. We will show that the LBA acceptance problem (see [Kar72]) logspace reduces to the self-stabilization problem for bounded general Petri nets; the theorem will then follow from Theorem 5.6. Let L be an arbitrary linearly bounded automaton (LBA), and w an arbitrary input. Jones, Landweber, and Lien [JLL77] show how to construct, using logarithmic space, a Petri net that simulates L on w. It is a straightforward matter to modify this construction such that the resulting Petri net (N_1, M_1) can reach some marking in which a designated place p_f has no tokens iff L accepts w. Furthermore, (N_1, M_1) can be constructed such that each move of L is simulated by the firing of one transition. Thus, if (N_1, M_1) can set the place p_f to 0, it can do so via a firing sequence of length at most $p(|L|)^{|w|} = 2^{|w|\lg(p(|L|))}$, where p is some polynomial independent of L and w, and lg denotes the base-2 logarithm. Using $|w|\lg(p(|L|)) + 1$ places and $|w|\lg(p(|L|))$ transitions, we can construct a Petri net (N_2, M_2), as shown in Figure 9, that can generate $2^{|w|\lg(p(|L|))}$ tokens. We then couple the two Petri nets as shown in Figure 9 so that the net is structurally terminating, and p_f can still be emptied iff L accepts w. Finally, we add additional transitions so that from any marking, all places except p_f can be emptied, and p_f may always be reduced to at most one token (see Figure 9). The resulting net (N, M_0) has two dead markings — $\mathbf{0}$ and the marking in which all places are 0 except p_f, which is 1. The latter of these markings is clearly in $R_d(N, M_0)$ and $\mathbf{0} \in R_d(N, M_0)$ iff p_f can be emptied iff L accepts w. Thus, from Theorem 5.2, (N, M_0) is self-stabilizing iff L accepts w.

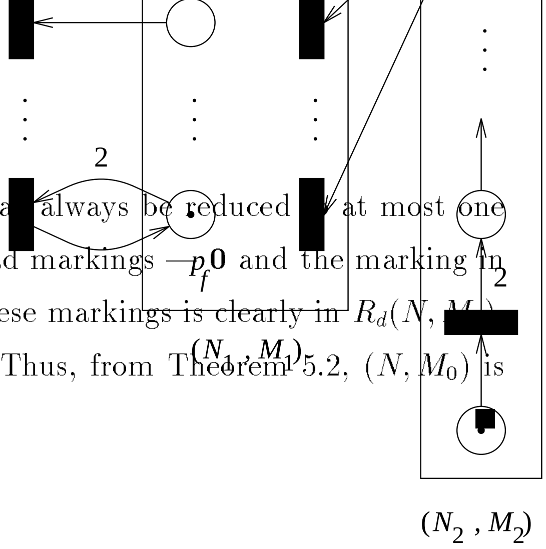


Figure 9.

Note that the above reduction also shows that the reachability problem for structurally terminating Petri nets is PSPACE-hard. Thus, from Lemma 5.5 and Theorem 5.7, we have the following corollary.

Corollary 5.8: The reachability problem for structurally terminating Petri nets is PSPACE-complete with respect to logspace many-one reductions.

5 Conclusion

We have given characterizations for self-stabilizing bounded Petri nets, both for ordinary Petri nets (no multiple arcs) and general Petri nets (multiple arcs allowed). Our characterizations have enabled us to give completeness results for the problems of determining whether a bounded ordinary Petri net is self-stabilizing and whether a bounded general Petri net is self-stabilizing. The complexities of these two problems were shown to be PTIME-complete and PSPACE-complete, respectively.

The precise complexities of the above problems for potentially unbounded Petri nets are unknown at this time; in fact, it is not currently known whether these problems are decidable, or even recursively enumerable. The best bounds we can achieve at the present time are an upper bound of Π_2 (the second level of the arithmetic hierarchy) and a lower bound of exponential space. Hence, we do not yet have as helpful a characterization for unbounded self-stabilizing Petri nets as we do for bounded self-stabilizing Petri nets.

Acknowledgments. We would like to thank the three anonymous referees for their helpful comments.

References

[BGW89] G. Brown, M. Gouda, and C. Wu. Token systems that self-stabilize. *IEEE Trans. on Computers*, 38:845–852, 1989.

[BP89] J. Burns and J. Pachl. Uniform self-stabilizing rings. *ACM Trans. on Programming Languages and Systems*, 11:330–344, 1989.

[BYC88] F. Bastani, I. Yen, and I. Chen. A class of inherently fault tolerant distributed programs. *IEEE Trans. on Software Engineering*, 14:1432–1442, 1988.

[Dij73] E. Dijkstra. EWD391 Self-stabilization in spite of distributed control, 1973. Reprinted in *Selected Writings on Computing: A Personal Perspective*, Springer-Verlag, Berlin, 1982, pp. 41-46.

[Dij74] E. Dijkstra. Self stabilizing systems in spite of distributed control. *Communications of the ACM*, 17:643–644, 1974.

[GHR90] M. Gouda, R. Howell, and L. Rosier. The instability of self-stabilization. *Acta Informatica*, 27:697–724, 1990.

[Gou90] M. Gouda. The stabilizing philosopher: Asymmetry by memory and by action, 1990. To appear in *Science of Computer Programming*.

[HJR91] R. Howell, P. Jančar, and L. Rosier. Single-path Petri nets. In *Proceedings of the 16 International Symposium on Mathematical Foundations of Computer Science*, pages 202–210, 1991. LNCS 520. To appear in *Information and Computation*.

[How91] R. Howell. The complexity of problems concerning structurally bounded and conservative Petri nets. *Information Processing Letters*, 39:309–315, 1991.

[HP79] J. Hopcroft and J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoret. Comput. Sci.*, 8:135–159, 1979.

[HR87] R. Howell and L. Rosier. Recent results on the complexity of problems related to petri nets. In G. Rozenberg, editor, *Advances in Petri Nets 1987*, pages 45–72. Springer, 1987. LNCS 266.

[HR88] R. Howell and L. Rosier. Completeness results for conflict-free vector replacement systems. *J. Comput. System Sci.*, 37:349–366, 1988.

[HRY87] R. Howell, L. Rosier, and H. Yen. An $O(n^{1.5})$ algorithm to decide boundedness for conflict-free vector replacement systems. *Inform. Process. Lett.*, 25:27–33, 1987.

[HRY89] R. Howell, L. Rosier, and H. Yen. Normal and sinkless Petri nets. In *Proceedings, 7th International Conference on Fundamentals of Computation Theory*, pages 234–243, 1989. LNCS 380. To appear in *J. Comput. System Sci.*

[JL77] N. Jones and W. Laaser. Complete problems for deterministic polynomial time. *Theoret. Comp. Sci.*, 3:105–117, 1977.

[JLL77] N. Jones, L. Landweber, and Y. Lien. Complexity of some problems in Petri nets. *Theoret. Comput. Sci.*, 4:277–299, 1977.

[Kar72] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[Kha79] L. Khachian. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk. SSSR*, 244:1093–1096, 1979. In Russian. English translation in *Soviet Math. Dokl.*, 20:191-194.

[KM69] R. Karp and R. Miller. Parallel program schemata. *J. Comput. System Sci.*, 3:147–195, 1969.

[Kos82] R. Kosaraju. Decidability of reachability in vector addition systems. In *Proceedings, 14th ACM Symposium on Theory of Computing*, pages 267–280, 1982.

[Lam86] L. Lamport. The mutual exclusion problem: Part II — Statement and solutions. *JACM*, 33:327–348, 1986.

[Lam88] J. Lambert. Consequences of the decidability of the reachability problem for Petri nets. In G. Rozenberg, editor, *Advances in Petri Nets 1988*, pages 266–282. Springer, Berlin, 1988. LNCS 340. An expanded version to appear in *Theoret. Comput. Sci.*

[Lip76] R. Lipton. The reachability problem requires exponential space. Technical Report 62, Yale University, Dept. of CS., Jan. 1976.

[May84] E. Mayr. An algorithm for the general Petri net reachability problem. *SIAM J. Comput.*, 13:441–460, 1984. A preliminary version of this paper was presented at the "13th ACM Symposium on Theory of Computing," 1981.

[Mul89] N. Multari. *Self-stabilizing Protocols.* PhD thesis, Dept. of Computer Sciences, University of Texas at Austin, 1989.

[Mur89] T. Murata. Petri nets: properties, analysis, and applications. *Proc. of the IEEE*, 77:541–580, 1989.

[Rac78] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoret. Comput. Sci.*, 6:223–231, 1978.

[Sil85] M. Silva. *Las Redes de Petri: en la Automatica y la Informatica.* Editorial AC, Madrid, 1985. ISBN 84 7288 045 1.