REVERSIBILITY ENFORCEMENT FOR UNBOUNDED PETRI NETS

Hanife Apaydın Özkan and Aydın Aybar
Department of Electrical and Electronics Engineering, Anadolu University
26470 Eskişehir, TURKEY

Keywords: Petri nets, partially reversibility, reversibility, algorithms.

Abstract: In this paper, partially reversibility property and reversibility enforcement are studied for unbounded Petri nets. A method which tests partial reversibility, and also finds a bound vector guaranting reversibility for unbounded Petri nets is developed and an algorithm of the method is generated. Furthermore a controller design approach which enforces reversibility for unbounded Petri nets is introduced.

1 INTRODUCTION

Petri net model is a common tool for discrete event systems. Some properties and definitions are used to describe this model. Properties of Petri nets are decomposed into two types such as behavioral and structural properties (Desrochers and Al-Jaar, 1995; Proth and Xie, 1996). In this work, we consider reversibility and partially reversibility which are two of important behavioral properties of Petri nets.

Some approaches have been presented to analyse reversibility and partially reversibility of Petri nets. The most favor is constructing reachability set. But it is not efficient for unbounded Petri nets because of infinite number of reachable marking vectors (Peterson, 1981). If a Petri net is partially reversible for at least one initial state, that is proven by using a structural analysis method named T-invariant (Desrochers and Al-Jaar, 1995). The method which was developed in (Jeng et al., 2002) verifies reversibility for 1-place unbounded Petri nets. Since these approaches give sufficient but not necessary conditions for partially reversibility, they do not propose a way to test partially reversibility of all unbounded Petri nets.

In this work, reversibility enforcement is considered for unbounded Petri nets. It is possible to enforce reversibility for a Petri net, if the net is partially reversible. Hence, testing partially reversibility is very important for our work. We present a method to test partially reversibility for unbounded Petri nets. If the net is partially reversible the method proposes a bound vector covering all reachable markings in a reversible subset of the reachability set. Moreover, we explain the controller presented in (Aybar et al., 2005) which enforces reversibility at each times it is used with the bound vector proposed by our method.

2 PRELIMINARIES

2.1 Notations of Petri Nets

A *Petri net* is denoted by five tuple $G(P,T,N,O,m_0)$, where P is the set of *places*, T is the set of transitions, $N : P \times T \to \mathcal{Z}$ is the *input matrix* that specifies the weights of arcs directed from places to transitions, $O : P \times T \to \mathcal{Z}$ is the *output matrix* that specifies the weights of arcs directed from transitions to places, where $\mathcal{Z}$ is the set of non-negative integer numbers, and m_0 is the *initial marking*.

$M : P \to \mathcal{Z}$ is a *marking vector* in other words *marking*, $M(p)$ indicates the number of *tokens* assigned by marking M to place p. A transition $t \in T$ is *enabled* if and only if $M(p) \geq N(p,t)$ for all $p \in P$. A *firing sequence* g is a sequence of enabled transitions $t_1 t_2 \ldots t_k$, where $t_1, t_2, \ldots, t_k \in T$. A marking M' is said to be *reachable* from M if there exists a firing sequence starting from M (i.e., the first transition of the sequence fires at M) and yielding M' (i.e., the final transition of the sequence yields M'). The set denoted by $R(G,M)$ is the set of all marking vectors reachable from M. $R(G,m_0)$ is

Apaydın Özkan H. and Aybar A. (2005).
REVERSIBILITY ENFORCEMENT FOR UNBOUNDED PETRI NETS.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics - Robotics and Automation*, pages 181-186
DOI: 10.5220/0001180501810186

called as *reachability set* of the Petri net. We let $\mathcal{E}(G, M)$ to denote the set of transitions which are enabled at $M \in R(G, m_0)$. For a Petri net, we also let $\rho(M, g)$ to denote the *transition function*, which gives the yielded marking when the sequence g fires starting from M (ρ is in fact a partial function, since it is not defined if g contains transitions which are not enabled) (Aybar and İftar, 2003). If $M' = \rho(M, g)$, then $M' = M + (O - N)U = M + AU$. Here; M and M' are markings, N and O are input and output matrices respectively, $A := O - N$ is *incidence matrix* and $U : T \rightarrow \mathcal{Z}$ is *firing count vector* whose jth element indicates how many times t_j is fired in g.

Let us remember some behavioral properties related to the discussion of this work. G is said to be *K-bounded*, if $M(p) \leq K(p)$, $\forall p \in P$, $\forall M \in R(G, m_0)$ ($K : P \rightarrow \mathcal{Z}$), G is said to be *bounded* if it is K-bounded for some $K : P \rightarrow \mathcal{Z}$. Otherwise G is *unbounded*. G is said to be *reversible* if $m_0 \in R(G, M)$, $\forall M \in R(G, m_0)$. G is said to be *partially reversible* if $m_0 \in R(G, M)$ for at least one $M \in R(G, m_0)$ such that $M \neq m_0$. Note that, if a Petri net is partially reversible, there exists a reversible subset of $R(G, m_0)$ and it is possible to find a bound vector covering all markings in this subset. It is possible to enforce reversibility of the net by using this bound vector with the controller in (Aybar et al., 2005). Therefore, we say that this bound vector guarantees reversibility for the considered net. *Deadlock* is said to occur in a Petri net if there exists $M \in R(G, m_0)$ such that no transition $t \in T$ can fire at M (Desrochers and Al-Jaar, 1995). A marking $\tilde{M}$ *covers* a marking M if $\tilde{M}(p) \geq M(p)$, $\forall p \in P$. A marking $\tilde{M}$ *dominates* a marking M, if $\tilde{M}$ covers M and $M \neq \tilde{M}$. That is denoted by $\tilde{M} >_d M$. If $\tilde{M} >_d M$ and $\mathcal{E}(G, M) = \mathcal{E}(G, \tilde{M})$, then $\rho(\tilde{M}, t) >_d \rho(M, t)$, $\forall t \in \mathcal{E}(G, M)$ (Cassandras and Lafortune, 1999).

The behavioral properties of Petri nets are commonly explained by using reachability set. Since unbounded Petri nets have infinite number of reachable markings, the coverability tree (CT) is used to analyse some behavioral properties instead of the reachability set. CT is drawn as a tree, where each node of tree either explicitly represents a reachable marking of m_0 or covers a reachable marking of m_0 through w notation. If there exists a w notation at a place of a marking in the CT, this place is unbounded place and this Petri net is unbounded (Desrochers and Al-Jaar, 1995). Note that since the representation of an infinite set is finite, an infinite number of markings must be mapped onto the same representation in the CT.

In this paper an algorithm in (Zhou and DiCesare, 1993) (Algorithm 5.1 on page 104) is used to construct CT.

3 REVERSIBILITY FOR UNBOUNDED PETRI NETS

Some works have been presented about reversibility of Petri nets (Peterson, 1981; Desrochers and Al-Jaar, 1995; Jeng et al., 2002). But none of them has ability of testing partially reversibility of unbounded Petri nets.

In this section we will explain a method, called *Partially Reversibility Testing Method* (PRTM). PRTM tests partially reversibility of unbounded Petri nets and yields a bound vector guaranting reversibility for the considered net. To facilitate discussion of the method, we first give the following explanations and Lemma 1.

Let R' be a set of marking vectors such that $R' := \{M \in R(G, m_0) \mid \rho(M, t) = m_0,\ t \in T\}$ then $M(p) = m_0(p) \pm a$, $a \in \{0,\ 1 ... \nu_p\}, \forall p \in P$. This means $\forall M \in R'$, $M(p) \leq m_0(p) + \nu_p$, $\forall p \in P$. Here ν_p denotes the maximum number of token variation in place p by firing any enabled transition. It can be determined for each place $p \in P$ by the following way:

$$\nu_p = \max_{t \in T}(N(p, t), O(p, t)) \quad (1)$$

Note that, if $M(p) > m_0(p) + \nu_p$ for at least one $p \in P$, then $M \notin R'$.

Lemma 1: If there exists a marking $M \in R'$ such that $M \neq m_0$, Petri net is partially reversible. Otherwise, Petri net is not partially reversible.

Proof: In a Petri net, each of the marking vector M satisfiying $\rho(M, t) = m_0$ are the members of the set R'. So, if there exists $M \in R'$ such that $M \neq m_0$, Petri net is partially reversible. If there exists no $M \in R(G, m_0)$ such that $\rho(M, t) = m_0$ and $M \neq m_0$ (there exists no marking $M \in R'$ such that $M \neq m_0$), then there exists no $\tilde{M} \in R(G, m_0)$ such that $\rho(\tilde{M}, g) = m_0$. Hence Petri net is not partially reversible. ◇

Although we do not know the reachability set for unbounded Petri nets; as a result of Lemma 1, we know that R' must have a marking vector other than m_0 for the presence of partially reversibility. Therefore, it is efficient to determine only R' set of the Petri net to test partially reversibility and one does not need to construct all reachability set for this test. PRTM tests partially reversibility of the considered unbounded Petri net by using this fact. It is possible to find a bound vector guaranting reversibility for a Petri net, iff the net is partially reversible. Hence, PRTM proposes a bound vector guaranting reversibility, if the considered net is partially reversible.

At the first step, PRTM determines unbounded places of the Petri net by using the CT and begins obtaining reachable markings from m_0 by firing transitions. In fact, for any unbounded Petri net it is possible to construct sets of reachable markings such that markings in each set are obtained by firing transitions or transition sequences from markings in the one previous generated set and each marking in each set dominates one of the marking in the one previous generated set (Cassandras and Lafortune, 1999). The method determines these sets ($R_1,\ R_2,\ R_3$...) step by step. When it finds a set R_i such that $\forall M \in R_i$ there exists a $\tilde{p} \in \tilde{P}$ ($\tilde{P}$ denotes the set of unbounded places) such that $M(\tilde{p}) > m_0(\tilde{p}) + \nu_{\tilde{p}}$, this means $R_i \cap R' = \emptyset$. Then, the method obtains a set $\tilde{R}$ including all of the markings obtained from m_0 to that point, i.e $\tilde{R} = \bigcup_{j=0}^{i-1} R_j$. Since $\forall i \in \{1,\ 2, ...\}$, each of the markings in R_{i+1} will dominate one of the marking in R_i, R' of the Petri net is a subset of $\tilde{R}$ and it is determined by searching the markings $M \in \tilde{R}$ such that $\rho(M,t) = m_0$. If there exists a marking $M \in R'$ such that $M \neq m_0$, this Petri net is partially reversible (see, Lemma 1) and PRTM proposes a bound vector covering all of the markings in $\tilde{R}$ for guaranting reversibility of the Petri net. Otherwise, Petri net is not partially reversible (see, Lemma 1) and it is impossible to guarantee reversibility. Hence, the method does not propose any bound vector.

3.1 Algorithms

In this section, the algorithm for PRTM, which is explained in Section 3, is presented with the help of a motivation example.

The main algorithm for PRTM is named $\mathcal{M}ain[G,M]$ (see, Appendix A). In this algorithm; first the set of unbounded places of a Petri net is determined by the function $\tilde{P} = C_T(G)$ which finds the set of unbounded places ($\tilde{P}$) of Petri net by constructing CT; for the Petri net shown in Figure 1, the set of vectors in the CT is $\{[2\ 1\ 0]^T,\ [1\ 2\ 1]^T,\ [3\ 0\ 0]^T,\ [0\ 3\ 2]^T,\ [2\ 1\ w]^T,\ [1\ 2\ w]^T,\ [3\ 0\ w]^T,\ [0\ 3\ w]^T\}$ and $\tilde{P} = \{p_3\}$. Then the set R' of Petri net is determined by the algorithm $\mathcal{R}prime[G,\ M,\ \tilde{P}]$. If there exists $M \in R'$ such that $M \neq m_0$, Petri net is partially reversible and a bound vector guaranting reversibility of the net is determined. Otherwise Petri net is not partially reversible (see, Lemma 1) and the algorithm is halted.

$\mathcal{M}ain[G,M]$ algorithm calls $\mathcal{R}prime[G,\ M,\ \tilde{P}]$ to construct R' set of Petri net (see, Appendix A). At this algorithm, the sets $R_1,\ R_2, ...$ are the sets of markings and each of these sets are obtained by firing transitions or transition sequences from markings in the previous set. Each marking in each set dominates one of the marking in the one previous set, i.e, R_{i+1} is obtained by firing transitions from R_i, and each of the markings in R_{i+1} dominates one of the marking in R_i, $i \in \mathcal{Z}$. For the construction of these sets;

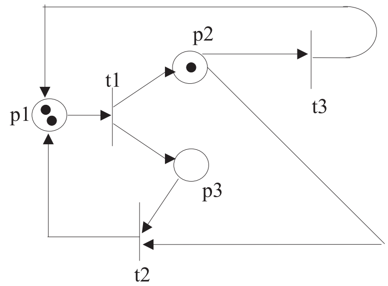

Figure 1: Motivation example (Proth and Xie, 1996)

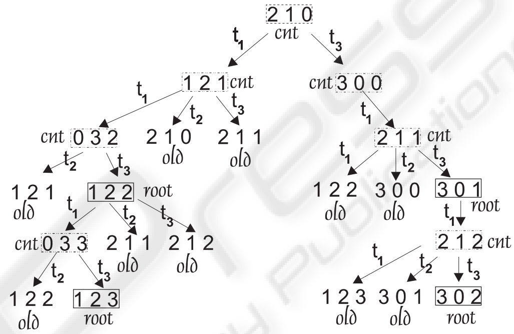

Figure 2: Obtained markings of Petri net in Figure 1

first, all enabled transitions are fired from m_0. This leads new markings. The new markings which are previously generated are labeled as old. The new markings which are deadlock are labeled as $dead$. If on the path $\tilde{M} = \rho(m_0, g)$ (path from m_0 to a new marking $\tilde{M}$), there exists a marking M such that, $\tilde{M} >_d M$, and $\mathcal{E}(G,M) = \mathcal{E}(G,\tilde{M})$, then $\tilde{M}$ is labeled as $root$ ($\rho(\tilde{M},t) >_d \rho(M,t),\ \forall t \in \mathcal{E}(G,M)$). From markings which are not labeled as old, $dead$ or $root$, we continue firing transitions and obtain new markings. If all of the enabled transitions of a marking are fired, this marking is labeled as cnt. This process is proceed until there exists no nolabeled markings. Then, except old labeled markings, all of the markings obtained from m_0 at this process construct the set R_0 and the $root$ labeled markings in R_0 construct the set SM_0. Then a new cycle begins by firing enable transitions of each markings in SM_0. As before, until there exists no nolabeled marking, new markings are obtained and they are labeled. But after that point rule of labeling as $root$ changes: a new marking $\tilde{M}$ is labeled as $root$ if there exists a marking M in the set SM_0 such that, $\tilde{M} >_d M$ and $\mathcal{E}(G,M) = \mathcal{E}(G,\tilde{M})$. Then, except old labeled markings, all the markings obtained from SM_0 at this process construct the set R_1 and the $root$ labeled markings in R_1 construct the set SM_1. If $\forall M \in R_1$, there exists a $\tilde{p} \in \tilde{P}$ such that,

$$M(\tilde{p}) > m_0(\tilde{p}) + \nu_{\tilde{p}},\ (R_1 \cap R' = \emptyset) \qquad (2)$$

all of the markings in the set R_i ($i > 1$) also satisfy equation (2). This means, none of the markings which will be obtained can not be a new member of the set R' of Petri net. Then a set $\tilde{R}$ ($R' \subset \tilde{R}$) is obtained, i.e. $\tilde{R} = R_0$, and the set R' of Petri net is determined by searching the markings in $\tilde{R}$ satisfying $\rho(M,t) = m_0,\ t \in T$. For the motivation example, the set R_0 is determined as $\{[2\ 1\ 0]^T,\ [1\ 2\ 1]^T,\ [3\ 0\ 0]^T,\ [0\ 3\ 2]^T,\ [2\ 1\ 1]^T,\ [1\ 2\ 2]^T,\ [3\ 0\ 1]^T\}$ (see, Figure 2). Since some markings in the set R_0 do not satisfy equation (2), i.e $[2\ 1\ 1]^T$; from markings in the set R_0, a new set is constructed as $R_1 = \{[0\ 3\ 3]^T,\ [1\ 2\ 3]^T,\ [3\ 0\ 2]^T,\ [2\ 1\ 2]^T\}$ (see, Figure 2). Note that, each of the markings in R_1 dominates one of the marking in R_0. Since all markings in R_1 satisfy equation (2), $\tilde{R} = R_0$. By searching the markings M in $\tilde{R}$ such that $\rho(M,t) = m_0$, the set R' is obtained as $R' = \{[1\ 2\ 1]^T\}$ for the Petri net shown in Figure 1.

If some of the markings of R_1 of a Petri net do not satisfy equation (2), from each markings in SM_1 their enabled transitions are fired. By this way, new markings are obtained and the sets SM_2 and R_2 are constructed (except old labeled markings, all of the markings obtained from markings in SM_1 at this process construct the set R_2 and the $root$ labeled markings in R_1 construct the set SM_2). This process is proceed until a set R_i satisfying equation (2) is obtained. Then a set $\tilde{R}$ ($R' \subset \tilde{R}$) is obtained, i.e. $\tilde{R} = \bigcup_{j=0}^{i-1} R_j$, and the set R' of Petri net is determined by searching the markings in $\tilde{R}$ such that $\rho(M,t) = m_0$.

If there exists $M \in R'$ such that $M \neq m_0$, Petri net is partially reversible and $\mathcal{M}ain[G,M]$ algorithm determines a bound vector K covering all of the markings in $\tilde{R}$ and guaranting reversibility of Petri net. For the motivation example, the set R' is determined as $\{[1\ 2\ 1]^T\}$. Since $[1\ 2\ 1]^T \neq m_0$, Petri net is partially reversible. The set of markings including all of the markings on the path $\rho(m_0,g) = [1\ 2\ 1]^T$ is a reversible subset of the reachability set. Since $\tilde{R}$ includes this set, the bound vector $[3\ 3\ 2]^T$ covering all markings in $\tilde{R}$ guarantees reversibility of this net.

If there exists no $M \in R'$ such that $M \neq m_0$, Petri net is not partially reversible and any bound vector is not determined by the algorithm $\mathcal{M}ain[G,M]$. Because, there exists no $M \in R(G,m_0)$ such that $\rho(M,g) = m_0$, if a Petri net is not partially reversible. Therefore, any K can not guarantee reversibility of the net.

Theorem 1: The bound vector K obtained by the algorithm $\mathcal{M}ain[G,M]$ guarantees reversibility for unbounded Petri net G.

Proof: If a marking M such that $M \neq m_0$ is a member of R', then Petri net is partially reversible and a bound vector K is determined. Since K covers not only all of the markings in R' but also all of the markings on the path from m_0 to each markings in R' (K covers all of the markings in $\tilde{R}$), it covers all of the markings in a reversible subset of the reachability set of the Petri net and it guarantees reversibility. $\diamond$

4 A CONTROLLER TO ENFORCE REVERSIBILITY

In (Aybar et al., 2005), some algorithms and a controller have been presented to enforce boundedness, reversibility and liveness. In that work; initially, with an arbitrarily chosen bound vector, a bounded reachability set of an unbounded Petri net has been determined; then, the reversible subset of that bounded set is constructed by using developed algorithms. Since obtained reversible set may be empty, reversibility can not be enforced by the controller everytimes.

If the considered unbounded Petri net is partially reversible, PRTM presented in the Section 3 obtains a bound vector, K, which guarantees reversibility for unbounded Petri nets. By running the developed algorithms in (Aybar et al., 2005) with the obtained bound vector gives a reversible subset of the reachability set of considered unbounded Petri net. So it is possible to enforce reversibility for this net by using controller presented in (Aybar et al., 2005). In this section that controller will be explained.

If a bound vector K is obtained by PRTM, Petri net is partially reversible and it is possible to enforce reversibility for this net by a controller. K bounded reachability set is found by the algorithm named Bounded-Set (Aybar et al., 2005), i.e. RB=Bounded-Set(G,K). Here, K and G are the inputs of the algorithm and represent the bound vector and definition of the Petri net, respectively; RB is the output of the algorithm and represents K bounded reachability set of G. Reversible subset of RB is found by the algorithm named Reversible-Set, i.e. Rr=Reversible-Set(RB) where R_r is the reversible subset of RB. Note that, since K guarantees reversibility, $R_r \neq \emptyset$. Then, it is known that if a Petri net is partially reversible, the controller $c(M,t)$ below enforces boundedness and reversibility of the net (Aybar et al., 2005).

$$c(M,t) = \begin{cases} 1, & \text{if } \rho(M,t) \in R_r \\ 0, & \text{otherwise} \end{cases} \qquad (3)$$

where, $M \in R(G,m_0),\ t \in \mathcal{E}(G,M)$. If $c(M,t) = 1$, then $\rho(M,t) \in R_r$ and firing transition t from marking M is allowed. If $c(M,t) = 0$, then $\rho(M,t) \notin R_r$ and firing transition t from marking M is forbidden.

5 EXAMPLE

Let us consider a Petri net modelled manufacturing system borrowed from (Proth and Xie, 1996) as an example. The Petri net model of this system is presented in Figure 3. Since the weights of arcs are unity, $\nu_p = 1, \forall p \in P$. For this Petri net, the set of places $P = \{p_1,\ p_2,\ p_3\ , r_1,\ r_2,\ r_3,\ p_4,\ p_5,\ p_6\}$, the set of transitions $T = \{t_1,\ t_2,\ t_3,\ t_4,\ t_5,\ t_6\}$, and the initial marking is $m_0 = [1\ 0\ 0\ 2\ 1\ 1\ 1\ 0\ 0]^T$. At the first step,

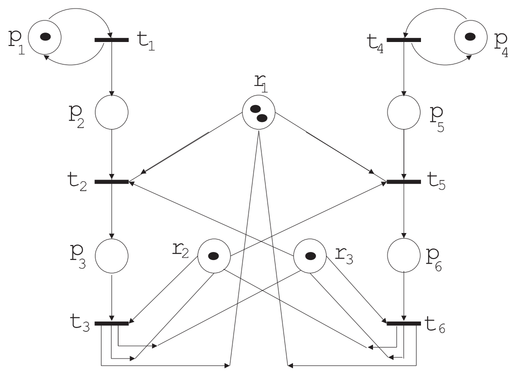

Figure 3: Example Petri net.

the algorithm $\mathcal{M}ain[G, M]$ calls the function $C_T(G)$ to find the unbounded places. Since there exists w notation at some places (p_2, p_5) of some markings at CT of this Petri net (Apaydın-Özkan, 2005), the set of unbounded places of this net is $\tilde{P} = \{p_2,\ p_5\}$. Then $\mathcal{M}ain[G, M]$ calls the algorithm $\mathcal{R}prime[G,\ M,\ \tilde{P}]$ to obtain a set including R' set of Petri net and R' itself. For this purpose first R_0 then R_1 are obtained ($|R_0| = 16,\ |R_1| = 41$, here $|*|$ denotes the number of elements of set "$*$"). Since R_1 does not satisfy equation (2), process continues and R_2 is obtained ($|R_2| = 63$). $\forall M \in R_2$, R_2 satisfies equation (2). This means, $R_2 \cap R' = \emptyset$ and all the members of R' are obtained ($R_i \cap R' = \emptyset,\ i \in \{2,\ 3,\ 4,\ 5...\}$). Then a set $\tilde{R}$ including R' of Petri net is obtained as $\tilde{R} = R_0 \cup R_1$ ($|\tilde{R}| = 57$). Through the markings in the set $\tilde{R}$, only markings $M_1 = [1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0]^T$ and $M_2 = [1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1]^T$ reach to m_0 by firing only one transition, i.e. $\rho(M_1, t_3) = m_0,\ \rho(M_2, t_6) = m_0$. Hence, R' is determined as $\{M_1,\ M_2\}$. Since there exist some $M \in R'$ such that $M \neq m_0$ ($M_1 \neq m_0,\ M_2 \neq m_0$), Petri net is partially reversible and $\mathcal{M}ain[G, M]$ determines the vector $K = [1\ 4\ 1\ 2\ 1\ 1\ 1\ 4\ 1]^T$ covering all of the markings in the set $\tilde{R}$. The bounded set RB, which is obtained by RB=Bounded-Set(G, K), is partially reversible and the reversible set R_r, which is obtained by Rr=Reversible-Set(RB), is not empty ($|RB| = 100,\ |R_r| = 75$)[1]. As a result, obtained K guarantees reversibility of the Petri net. Additionally the controller $c(M, t)$ enforces not only boundedness but also reversibility for this net. If we were to give two specific cases as an example, $c([1\ 2\ 0\ 2\ 1\ 1\ 1\ 2\ 0]^T,\ t_1) = 1$ since $\rho([1\ 2\ 0\ 2\ 1\ 1\ 1\ 2\ 0]^T, t_1) = [1\ 3\ 0\ 2\ 1\ 1\ 1\ 2\ 0]^T \in R_r$; $c([1\ 4\ 0\ 1\ 0\ 1\ 1\ 4\ 1]^T,\ t_4) = 0$ since $\rho([1\ 4\ 0\ 1\ 0\ 1\ 1\ 4\ 1]^T, t_4) = [1\ 4\ 0\ 1\ 0\ 1\ 1\ 5\ 1]^T \notin R_r$.

6 CONCLUSION

In this work, we consider partially reversibility and reversibility enforcement for unbounded Petri nets. For this purpose, a method and its corresponding algorithm is developed. By using the algorithm, it is determined whether considered unbounded Petri net is partially reversible or not. If it is partially reversible, the algorithm determines a bound vector guaranting reversibility and the controller $c(M, t)$ enforces boundedness and reversibility of this net. If the Petri net is not partially reversible, algorithm does not find a bound vector and reversibility can not be enforced for this Petri net.

In this work, a Matlab program is also developed to simulate the presented algorithm.

Further research is underway to use T-invariants (see, section 5.6 in (Desrochers and Al-Jaar, 1995)) for testing partially reversibility and reversibility enforcement of Petri nets. Only the Petri nets with controllable transitions are the subjects under the discuss in this work, this approach may be extended to Petri nets with controllable and uncontrollable transitions.

APPENDICES

A) Algorithms for PRTM

$\mathcal{M}ain$ [G,M]
$\tilde{P}$ = $\mathcal{C}_T[G]$;
$< \tilde{R},\ R' >$=$\mathcal{R}prime[G,\ p_i]$;
If ($\nexists M \in R'$ *such that* $M \neq m_0$**) Then**
  "Petri net is not partially reversible"
  Exit $\mathcal{M}ain$
Else
  "Petri net is partially reversible"
  For ($i = 1 : |P|$**)**
    $K(i) = \max_{M \in \tilde{R}}(M([P]_i))$;
  End

[1]The sets $R_0,\ R_1,\ R_2,\ RB$ and R_r of the example Petri net are not given here due to space limitations. But one can see them in (Apaydın-Özkan, 2005).

End
Return K

$\underline{\mathcal{R}prime[G,\ M,\ \tilde{P}]}$
$i = 0$; $SM_0 = \emptyset$;
For each $\tilde{p} \in \tilde{P}$ determine $\nu_{\tilde{p}}$;
Do loop $Rtilde$
  If (i=0**) Then**
    R_i=m_0;
  Else
    R_i=SM_{i-1};
  End
  Do loop R_xset
    Select a nolabeled marking M from R_i;
    If (M *is previously generated***) Then**
      Label M as old; $R_i = R_i \backslash \{M\}$;
    Else If ($\mathcal{E}(G, M) = \emptyset$**) Then**
      Label M as $dead$;
    Else If ($i = 0$ && $\exists \tilde{M}$ *on the path from* m_0 *to* M*, such that* $M >_d \tilde{M}$*,* $\mathcal{E}(G, M) = \mathcal{E}(G, \tilde{M})$**)Then**
      Label M as $root$; $SM_i = SM_i \cup \{M\}$;
    Else If ($i > 0$ && $\exists \tilde{M} \in SM_{i-1}$ *such that* $M >_d \tilde{M}$ $\mathcal{E}(G, M) = \mathcal{E}(G, \tilde{M})$**) Then**
      Label M as $root$; $SM_i = SM_i \cup \{M\}$;
    Else
      Fire each transition in $\mathcal{E}(G, M)$ from M;
      Add each obtained marking vector to set R_i;
      Label M as cnt;
    End
    If ($\nexists$ *nolabeled marking in* R_i**) Then**
      If ($i \neq 0$**)Then**
        $R_i = R_i \backslash SM_{i-1}$;
      End
      Exit R_xset
    End
  LoopR_xset
  If ($\forall M \in R_i$ $\exists \tilde{p}$ *in* $\tilde{P}$ *such that* $M(\tilde{p}) > m_0(\tilde{p}) + \nu_{\tilde{p}}$ **) Then**
    $\tilde{R} = \bigcup_{j=0}^{i-1} R_j$;
    Exit Rtilde
  End
  $i = i + 1$;
  $R_i = \emptyset$; $SM_i = \emptyset$;
Loop $Rtilde$
For ($i = 1 : |\tilde{R}|$**)**
  If ($\exists\ t \in T$ *such that* $\rho([\tilde{R}]_i, t) = m_0$**) Then**
    $R' = R' \cup [\tilde{R}]_i$;
  End
End
Return R' $\tilde{R}$

B) Notation used in the presentation of algoritms:

For a set X, $|X|$ denotes the number of elements of set X and $[X]_i$ denotes the ith element of X ($i = 1, 2, ..., |X|$). All the sets are assumed to be ordered. If a new element is added to a set size m, the new element is taken as the $(m + 1)$th element of the set. $\cup$ is used to set *union*. If a vector X dominates a vector Y, $X >_d Y$ denotes this situation. $M(p_i)$, denotes the ith place of marking M. The logic "*and*" operation is represented by && in the algorithms.

REFERENCES

Apaydın-Özkan, H. (2005). Determination of bound vectors to guarantee reversibility for unbounded Petri nets. M.S Thesis. (in Turkish). Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, Turkey.

Aybar, A. and İftar, A. (2003). Decentralized controller design to enforce boundedness, liveness and reversibility in Petri nets. In *Proceeding CD-ROM of the European Control Conference*, Cambridge, UK.

Aybar, A., İftar, A., and Apaydın-Özkan, H. (2005). Centralized and decentralized supervisory controller design to enforce boundedness, liveness and reversibility in Petri nets. *International Journal of Control*, 78:537–553.

Cassandras, C. G. and Lafortune, S. (1999). *Introduction to Discrete Event Systems*. Kluwer Academics, Norwell, MA.

Desrochers, A. A. and Al-Jaar, R. Y. (1995). *Applications of Petri Nets in Manufacturing Systems*. The Institute of Electrical and Electronics Engineers Inc., New York.

Jeng, M., Xie, X., and Peng, M. (2002). Process nets with resources for manufacturing modeling and their analysis. *IEEE Transactions on Robotics and Automation*, 18:875–889.

Peterson, J. (1981). *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, NJ : Prentice-Hall, New Jersey.

Proth, J. and Xie, X. (1996). *Petri Nets: A Tool for Design and Management of Manufacturing Systems*. John Wiley & Sons, West Sussex.

Zhou, M. and DiCesare, F. (1993). *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic, Norwell, MA.