

Automated Synthesis of Deadlock Prevention Supervisors Using Petri Nets

Technical Report of the ISIS Group
at the University of Notre Dame

ISIS-2000-003

May, 2000

Revised in October 2000 and November 2001

Marian V. Iordache
Department of
Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
iordache.1@nd.edu

John O. Moody
Lockheed Martin
Federal Systems
1801 State Rt. 17C, MD 0210
Owego, NY 13827-3998
john.moody@lmco.com

Panos J. Antsaklis
Department of
Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
antsaklis.1@nd.edu

Interdisciplinary Studies of Intelligent Systems

AUTOMATED SYNTHESIS OF DEADLOCK PREVENTION SUPERVISORS USING PETRI NETS

Marian V. Iordache*, John O. Moody†, Panos J. Antsaklis*

Abstract

Given an arbitrary Petri net structure, which may have uncontrollable and unobservable transitions, the deadlock prevention procedure presented here determines a set of linear inequalities on the marking of a Petri net. When the Petri net is supervised so that its markings satisfy these inequalities, the supervised net is proved to be deadlock-free for all initial markings that satisfy the supervision constraints. Deadlock-freeness implies that there will always be at least one transition that is enabled in the closed loop (supervised) system. The method is not guaranteed to insure liveness, as it can be applied to systems that cannot be made live under any circumstances. However, it is shown that when the method does insure liveness, it is at least as permissive as any other liveness-insuring supervisor. Moreover, it is shown that the method is not too restrictive even for Petri nets in which not all transitions can be made live. The procedure allows automated synthesis of the supervisors. Based on this method we formulate and prove two extended methods with guaranteed termination and a method for maximally permissive deadlock prevention.

1 Introduction

Deadlock is an undesirable phenomenon that may occur in systems that contain components running in parallel and sharing common resources. A system is deadlocked when, due to mutual interdependencies and reliance on shared resources that can not be freed, no further actions can be taken by the system. *Deadlock prevention* differs from *liveness-insurance*: when the liveness of a system is guaranteed, all actions that a system can perform may be repeated infinitely often. Deadlock prevention insures that at least some subset of the system's actions may be repeated, but not necessarily all. Deadlock prevention may be applied to any system in which liveness can be guaranteed, however it is not possible to insure liveness for every system that can be made deadlock-free. The procedure presented here can be computationally expensive, however, all computations are performed off-line. This differentiates the technique from deadlock *avoidance* strategies that perform potentially expensive computations while the system is in operation. A controller resulting from our deadlock *prevention* method requires very little in terms of computational resources at run time.

Issues regarding conflict, synchronization, and concurrency naturally arise during the study of deadlock. These properties make the Petri net a particularly useful formalism for modeling systems that are susceptible to deadlock. The use of Petri nets also provides a powerful suite of algebraic and graph-theoretic tools for analyzing the nature of deadlock and performing automatic synthesis procedures. The deadlock prevention

*Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 (e-mail: iordache.1, antsaklis.1@nd.edu)

†Lockheed Martin Federal Systems, 1801 State Rt.17C, MD 0210, Owego NY 13827-3998 (e-mail: john.moody@lmco.com)

method presented here uses Petri net models for the plant and results in a Petri net model of the supervisor, providing a unified formalism for representing the closed-loop system. The unified plant/controller model allows the approach to handle timed Petri nets or plants that include uncontrollable or unobservable transitions. The method presents the conditions necessary to insure deadlock freedom as a set of linear integer inequalities. This output is important because it can be used directly in optimization problems, e.g., determining the minimum number of resources a system requires using a linear integer program. The procedure also supports initial constraints. In this way the user is allowed to prevent the procedure to generate constraints which would conflict with other requirements. The procedure is flexible enough to be able to create a deadlock prevention supervisor even in such cases.

Deadlock prevention methods rely on structural properties of the net. Deadlock in Petri nets has been related to *siphons* (see section 2 and [4]). Among deadlock prevention papers, [20] and [10] use *control places* to supervise the net, as in our approach. Control places have also been used in the deadlock avoidance method of [2]. The deadlock prevention method of [10] defines a subclass of the *ordinary* and *conservative* Petri nets and requires the target Petri net to be in that subclass. In these conditions *liveness* is enforced, rather than (total) deadlock prevention. The advantages of the method of [10] are simplicity and guaranteed success. The disadvantages are the assumptions made on the Petri net structure and that the method can be restrictive. The deadlock prevention method of [20] is intended for bounded Petri nets. A major advantage of this approach is that it can effectively handle Petri nets that are not ordinary. One disadvantage of the approach in [20] is that it is not effective for *nonrepetitive* Petri nets. Another problem is that [20] cannot *guarantee* deadlock prevention since it does not detect the case when the siphon supervision enforced by a control place is disabled by the transformation to ordinary Petri nets.

The procedure presented here is related to the approach of [20], but differs in several aspects. This new method is appropriate for use on nets that may not be structurally live, i.e., non-repetitive systems for which liveness cannot be enforced under any circumstances. When the procedure is applied to repetitive systems, complete system liveness may well be the result. We show that the resulting supervisor is at least as permissive as any liveness-enforcing supervisor, i.e., no liveness-insuring supervisor will ever allow a transition to fire that our procedure would prevent from firing. Thus, when the procedure enforces liveness, it can be said to be a “maximally permissive” liveness supervisor. With regard to our previous work ([15] and [16]), now we approach Petri nets which may have uncontrollable and unobservable transitions and allow initial constraints. Also, we include new significant theoretical results. Based on them the current deadlock prevention method could be changed for better performance, including better permissivity results and relaxed requirements for the proof of deadlock prevention. Among the new results we mention a new termination theorem and a procedure which is proved to produce maximally permissive deadlock prevention.

The method is an iterative approach that removes new potential deadlock situations at every iteration. When (and if) the procedure terminates, the control designer is presented with either a supervised net that is guaranteed to be deadlock-free or an indication that the plant cannot be made deadlock-free under any circumstances. Unlike [20], the algorithmic computations are independent of the initial conditions of the plant, in fact, the control designer is presented with a set of valid initial conditions (initial markings) for which deadlock may be successfully prevented as part of the output of the procedure.

An interesting property of our method is that it solves a problem which cannot be solved with finite automata based approaches. Indeed, by considering all possible initial markings, an automaton with an infinite number of states is obtained. Note that this is not the case for the methods which consider a given initial marking and a bounded Petri net. The applications which benefit most from considering the initial

marking to be unknown may be in the area of Flexible Manufacturing, as the initial marking corresponds to the number of available resources.

The document is organized as follows. Section 2 reviews basic Petri net properties and describes the notations which are used throughout the paper. Section 3 presents some deadlock and liveness properties. We emphasize the supervisory control aspect of enforcing liveness and preventing deadlock and we derive significant consequences of a known result. Thus Corollary 3.2(c) provides sufficient conditions that our method enforces liveness. We also derive Corollary 3.3 which is the basis for better deadlock tests, such as Proposition 3.5 and Proposition 3.6. Proposition 3.5 allows us to make our method effective for nonrepetitive Petri nets and Proposition 3.6 allows us to formulate a maximally permissive deadlock prevention approach in section 6.4.3. In section 4 we present preliminaries to our methodology. The supervisory technique used by our method is supervisory control based on place invariants ([37] and [24]), which we also outline in section 4.2. The transformation technique for nonordinary Petri nets presented in section 4.1 is a modification of that of [20]. The siphon control approach (largely a particularization of the supervision based on place invariants) is given in section 4.3. Section 5 defines the deadlock prevention procedure and the operations which are involved. The procedure is defined in section 5.4. Illustrative examples are given in section 5.6. Section 6 gives the formal characterization of the procedure. The analysis of the procedure is complex, so in section 6.1 we provide some basic results, which are used by our main results, characterizing the procedure or the operations involved by it. The main results are given in section 6.2. Theorem 6.2 proves that the procedure does prevent deadlock. Theorem 6.3 proves the permissivity quality: the procedure is not more restrictive than any supervisor which, given an initial marking μ , enforces that all transitions which can be made live (that is all transitions which appear infinitely often in some transition sequence enabled by μ) are live. In particular, this shows that our method is not more restrictive than any supervisor enforcing liveness. Section 6.2.2 contains results which show that by (possibly) compromising some aspect of the performance of the procedure, termination can be guaranteed. We end the result section with some significant special cases (section 6.3) and remarks (section 6.4). In particular, section 6.4.3 shows how to use our procedure for maximally permissive deadlock prevention.

2 Review of Some Petri Net Basic Properties

In this paper we assume that the reader knows the fundamentals of Petri nets. Good introductions to Petri nets are for instance [27], [7] and [28]. This section is meant mainly to introduce our notations.

A **Petri net structure** is a quadruple $\mathcal{N} = (P, T, F, W)$ where P is the **set of places**, T the **set of transitions**, $F \subseteq (P \times T) \cup (T \times P)$ is the set of **transition arcs** and $W : F \rightarrow \mathbb{N} \setminus \{0\}$ is a **weight function**. A **marking** μ of the Petri net structure is a map $\mu : P \rightarrow \mathbb{N}$. A Petri net structure \mathcal{N} with **initial marking** μ_0 is called a **Petri net**, and will be denoted by (\mathcal{N}, μ_0) . For simplicity, we may denote sometimes by Petri net a Petri net structure.

It is useful to consider a marking both as a map and as a vector. These requirements are not necessarily conflicting, because there are authors ([28]) that define vectors as maps defined on a set A instead of $\{1, 2, \dots, m\}$, as is customary. The **marking vector** is defined to be $[\mu(p_1), \mu(p_2), \dots, \mu(p_n)]^T$, where p_1, p_2, \dots, p_n are the places of the net enumerated in a chosen (but fixed) order and μ the current marking. The same symbol μ will denote a marking vector. The marking vector of a Petri net may be regarded as the state variable of the Petri net. An equivalent way of saying that place p has the marking $\mu(p)$ is that p has $\mu(p)$ **tokens**.

Figure 1 could be used to illustrate the graphical representation of Petri nets. A token is represented by a bullet. The marking vector in figure 1(b) is $[0, 1, 1]^T$. An arc weight is indicated near the arc when it is not one. For instance, in figure 1(b) $W(p_3, t_1) = 2$ and $W(t_2, p_2) = 4$.

The **preset** of a place p is the set of incoming transitions to p : $\bullet p = \{t \in T : (t, p) \in F\}$. The **postset** of a place p is the set of outgoing transitions from p : $p\bullet = \{t \in T : (p, t) \in F\}$. p is a **source place** if $\bullet p = \emptyset$ and a **sink place** if $p\bullet = \emptyset$. Similar definitions apply for transitions. They are also extended for sets of places or transitions; for instance, if $A \subseteq P$, $\bullet A = \bigcup_{p \in A} \bullet p$, $A\bullet = \bigcup_{p \in A} p\bullet$.

We use $\mu[t$ to denote that μ enables the transition t and $\mu[t > \mu'$ to denote that μ enables t and if t fires, then the marking becomes μ' . The marking μ' is **reachable** from μ if there is a sequence of markings μ_1, \dots, μ_k , $\mu_k = \mu'$, and a sequence of transitions t_{i_1}, \dots, t_{i_k} s.t. $\mu[t_{i_1} > \mu_1[\dots t_{i_k} > \mu'$. The **set of reachable markings** of a Petri net (\mathcal{N}, μ) (i.e. the set of markings reachable from the initial marking μ) will be denoted by $\mathcal{R}(\mathcal{N}, \mu)$.

In a Petri net $\mathcal{N} = (P, T, F, W)$ with m places and n transitions, the **incidence matrix** is an $m \times n$ matrix defined by $D = D^+ - D^-$, where the elements d_{ij}^+ and d_{ij}^- of D^+ and D^- are

$$\begin{aligned} d_{ij}^+ &= W(t_j, p_i) \text{ if } (t_j, p_i) \in F \text{ and } d_{ij}^+ = 0 \text{ otherwise;} \\ d_{ij}^- &= W(p_i, t_j) \text{ if } (p_i, t_j) \in F \text{ and } d_{ij}^- = 0 \text{ otherwise.} \end{aligned}$$

The incidence matrix allows an algebraic description of the marking change of a Petri net:

$$\mu_k = \mu_{k-1} + D \cdot u_k \quad (1)$$

where u_k is called **firing vector**, and its elements are all zero excepting $u_{k,i} = 1$, where i corresponds to the transition t_i that fired. We will denote by **firing vector** also a vector x associated with a sequence of transitions that have fired, whose entries record how often each transition appears in the sequence. If x is the firing vector of the transition sequence that led the Petri net from the marking vector μ_0 to μ_k :

$$\mu_k = \mu_0 + D \cdot x \quad (2)$$

A vector x is called **place invariant** if $x^T \cdot D = 0$. A vector x is called **transition invariant** if $D \cdot x = 0$. The **support of a transition invariant** x is $\|x\| = \{t_j \in T : x(j) \neq 0\}$.

A Petri net (\mathcal{N}, μ_0) is said to be **deadlock-free** if for any reachable marking μ there is an enabled transition. (\mathcal{N}, μ) is in **deadlock** if no transition is enabled at marking μ .

Let (\mathcal{N}, μ_0) be a Petri net. A transition t is said to be **live** if $\forall \mu \in \mathcal{R}(\mathcal{N}, \mu_0) \exists \mu' \in \mathcal{R}(\mathcal{N}, \mu)$ such that t is enabled by μ' . A transition t is **dead** at marking μ if no marking $\mu' \in \mathcal{R}(\mathcal{N}, \mu)$ enables t . (\mathcal{N}, μ_0) is said to be **live** if every transition is live.

A nonempty set of places $S \subseteq P$ is called a **siphon** if $\bullet S \subseteq S\bullet$ and **trap** if $S\bullet \subseteq \bullet S$. In particular, $S = P$ may be siphon. An **empty siphon** with respect to a Petri net marking μ is a siphon S such that $\sum_{p \in S} \mu(p) = 0$. The attribute ‘‘empty’’ refers to the fact that S has no tokens. A siphon has the property that if for some marking it is empty, it will be so for all subsequent reachable markings. A trap has the property that if at some marking it has one token, then for all subsequent reachable markings it will have at least one token. See figure 1 for siphon examples. In figure 1(a), $\{p_1, p_3\}$ and $\{p_2, p_4\}$ are traps. S is a **minimal siphon** if there is no other siphon S' (by definition, $S' \neq \emptyset$) such that $S' \subset S$.

3 Deadlock and Liveness Properties of Petri Nets

This section introduces certain liveness and deadlock properties, focusing on their relation to structural properties of Petri nets and supervision. Throughout this section all transitions are considered to be controllable and observable.

3.1 Intrinsic Properties

A Petri net $\mathcal{N} = (P, T, F, W)$ is **ordinary** if $\forall f \in F : W(f) = 1$. In the construction of our procedure we will need to refer to slightly more general Petri nets in which only the arcs from places to transitions have weights equal to one. We are going to call such Petri nets *PT-ordinary*, because all arcs (p, t) from a place p to a transition t satisfy the requirement of an ordinary Petri net that $W(p, t) = 1$.

Definition 3.1 Let $\mathcal{N} = (P, T, F, W)$ be a Petri net. We call \mathcal{N} **PT-ordinary** if $\forall p \in P, \forall t \in T$, if $(p, t) \in F$ then $W(p, t) = 1$.

The basis of the results of this paper comes from a well known necessary condition for deadlock ([28]), namely that a deadlocked ordinary Petri net contains at least one empty siphon. It can easily be seen that the proof of this result also is valid for PT-ordinary Petri nets and so the following proposition follows:

Proposition 3.1 A deadlocked PT-ordinary Petri net contains at least one empty siphon.

An example is shown in figure 1(a). A simple way to generalize this result to more general Petri nets is given in Proposition 3.2. The proof of Propositions 3.1 and 3.2 are similar.

Proposition 3.1 shows that deadlock can be prevented by ensuring in a nonblocking way that no siphon ever loses all its tokens. The condition in Proposition 3.1 is only necessary. The example of figure 1(c) illustrates that the condition of Proposition 3.1 is not sufficient and figure 1(b) that the result is not applicable to Petri nets more general than PT-ordinary.

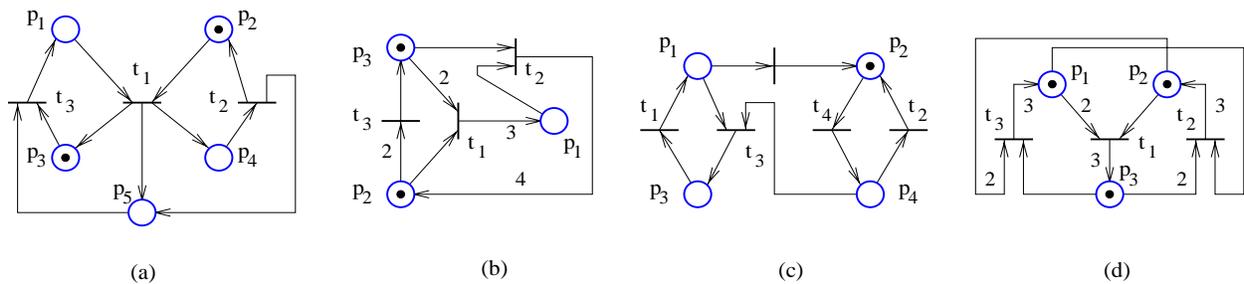


Figure 1: (a) A deadlocked PT-ordinary Petri net. An empty siphon is $\{p_1, p_4, p_5\}$. (b) A deadlocked Petri net with no empty siphon which is not PT-ordinary. (c) A deadlock-free Petri net (for the marking displayed) with an empty siphon – $\{p_1, p_3\}$. (d) Example for Proposition 3.2.

Definition 3.2 (cf. [2]) Let \mathcal{N} be a Petri net and μ a marking. \mathcal{N} is said to be **well-marked** for μ if in every siphon there is at least a token.

Definition 3.3 Let \mathcal{N} be a Petri net and \mathcal{M}_I be a set of initial markings. A siphon S is said to be **controlled** with respect to \mathcal{M}_I if $\forall \mu_0 \in \mathcal{M}_I, \forall \mu \in \mathcal{R}(\mathcal{N}, \mu_0): \sum_{p \in S} \mu(p) \geq 1$.

A controlled siphon contains for all reachable markings at least one token. A **trap controlled siphon** is a siphon that includes a trap. Recalling the trap property, for all markings such that the trap has one token, the siphon is controlled.

We define an **invariant controlled siphon** as a siphon S of a Petri net \mathcal{N} with the property that \mathcal{N} has a place invariant x such that for all $i = 1, 2, \dots, |P|$, if $x(i) > 0$ then $p_i \in S$. It is easy to show that for all initial markings μ_0 , such that $x^T \mu_0 \geq 1$, the siphon S is controlled.

In particular, a siphon which contains a controlled siphon is controlled. Therefore in a Petri net such that all minimal siphons are controlled, all siphons are controlled. Also, by Proposition 3.1, a PT-ordinary Petri net is deadlock-free if all its siphons are controlled. This is not true for more general Petri nets. The following result is also in [3].

Proposition 3.2 A deadlocked Petri net $\mathcal{N} = (P, T, F, W)$ with marking μ has at least one siphon S such that $\forall p \in S \exists t \in p \bullet$ with $W(p, t) > \mu(p)$.

Proof: Deadlock implies $\bullet P \subseteq P \bullet$. Otherwise, if $\exists t \in \bullet P \setminus P \bullet$, t can fire independently of the marking of the net, and so μ would not be a deadlock marking. Let $S = P$, and since no transition is enabled, S is a desired siphon. \square

Figure 1(b) shows a deadlocked Petri net. There are two minimal siphons: $S_1 = \{p_1, p_2\}$ and $S_2 = \{p_2, p_3\}$. The marking of p_3 does not prevent t_2 from firing but does prevent t_1 . The marking of p_2 does not prevent t_1 but prevents t_3 . For the current marking $[0, 1, 1]$, both siphons S_1 and S_2 satisfy the necessary condition of the proposition. For the deadlock the marking $[0, 0, 2]$, only one of them satisfies it. Another example is in figure 1(d), where we see that the only siphon satisfying the requirement of the proposition is the whole net. The requirement of Proposition 3.2 seems difficult to relax. For instance, it is not true that if in all minimal siphons S , if $\exists p \in S \forall t \in p \bullet \cap \bullet S, \mu(p) \geq W(p, t)$ then the Petri net is not in deadlock, as it could be checked in figure 1(b).

Loss of liveness is a less severe form of deadlock, where some actions can no longer happen while others may still be possible. Deadlock implies loss of liveness. An empty siphon in a PT-ordinary net is a necessary and not a sufficient condition for deadlock, while for loss of liveness it is a sufficient but not a necessary condition. Commoner's Theorem states that in an ordinary free choice net \mathcal{N} , if there are dead transitions for a marking μ , then there is a reachable marking $\mu' \in \mathcal{R}(\mathcal{N}, \mu)$ such that a siphon is empty ([28] p.103). Theorem 3.1 is the generalization to asymmetric choice nets. An **asymmetric choice** net is a Petri net $\mathcal{N} = (P, T, F, W)$ with the property that $\forall p_1, p_2 \in P, p_1 \bullet \cap p_2 \bullet \neq \emptyset \Rightarrow p_1 \bullet \subseteq p_2 \bullet$ or $p_2 \bullet \subseteq p_1 \bullet$.

Theorem 3.1 [3] An asymmetric choice net (\mathcal{N}, μ_0) such that $\forall p \in P \forall t \in p \bullet: W(p, t) = V(p)$ for some $V : P \rightarrow \mathbb{N}$, is live if and only if for all siphons $S, \forall \mu \in \mathcal{R}(\mathcal{N}, \mu_0) \exists p \in S$ such that $\mu(p) \geq V(p)$.

3.2 Conditions for Deadlock Prevention and Liveness Enforcement

Definition 3.4 Let $\mathcal{N} = (P, T, F, W)$ be a Petri net and $U \subseteq \mathbb{N}^{|P|}$. A **supervisory policy** Ξ is a function $\Xi : U \rightarrow 2^T$ that maps to every marking a set of transitions that the Petri net is allowed to fire. The markings in $\mathbb{N}^{|P|} \setminus U$ are called **forbidden markings**.

We denote by $\mathcal{R}(\mathcal{N}, \mu_0, \Xi)$ the set of reachable markings when (\mathcal{N}, μ_0) is supervised with Ξ . It is known that if (\mathcal{N}, μ_0) is live, then (\mathcal{N}, μ) with $\mu \geq \mu_0$ may not be live. The same is true for deadlock-freeness, as shown in figure 2. The following result shows that if liveness is enforceable at marking μ or if deadlock can be prevented at μ , then this is also true for all markings $\mu' \geq \mu$.

Proposition 3.3 *If a supervisory policy Ξ which prevents deadlock in (\mathcal{N}, μ_0) exists, then for all $\mu \geq \mu_0$ there is a supervisory policy which prevents deadlock in (\mathcal{N}, μ) . The same is true for liveness enforcement.*

Proof: Let $\mu_1 \geq \mu_0$. A supervisory policy for (\mathcal{N}, μ_1) is Ξ_1 defined as follows:

$$\Xi_1(\mu + \mu_1 - \mu_0) = \begin{cases} \Xi(\mu) \cap T_f(\mu) & \text{for } \mu \in \mathcal{R}(\mathcal{N}, \mu_0) \\ \emptyset & \text{otherwise} \end{cases}$$

where $T_f(\mu)$ denotes the transitions enabled by the marking μ , apart from the supervisor. □

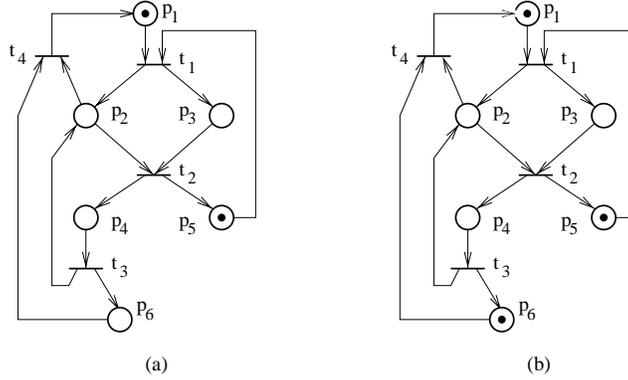


Figure 2: A Petri net that for the initial marking μ_0 shown in (a) is live, and for the initial marking $\mu \geq \mu_0$ shown in (b) is not even deadlock-free.

Definition 3.5 [27] *A Petri net is said to be (partially) repetitive if there is a marking μ_0 and a firing sequence σ from μ_0 such that every (some) transition occurs infinitely often in σ .*

The following lemma seems to be necessary for the necessity proof of Theorem 3.2, which is a known result. The authors do not know a reference where the necessity proof of Theorem 3.2 appeared. We prove the lemma as we need it in order to prove a number of other results, including Theorem 3.3 and Corollary 3.3.

Lemma 3.1 *Let $\mathcal{N} = (P, T, F, W)$ be a Petri net of incidence matrix D . Assume that there is an initial marking μ_0 which enables a firing sequence σ such that all transitions in $U \subseteq T$ appear infinitely often in σ . Then there is a nonnegative integer vector x such that $Dx \geq 0$ and $x(i) \neq 0 \forall t_i \in U$, where t_i is the transition corresponding to the i 'th column of D .*

Proof: In this proof the marking is regarded as the *marking vector*. Let U be the set of transitions which appear infinitely often in an infinite firing sequence σ enabled for some marking μ_0 . We are to prove that a

vector of nonnegative integers x , $x(i) \neq 0 \forall t_i \in U$ exists, such that $D \cdot x \geq 0$. When σ is fired, let μ_0 be the initial marking, μ_1 the first marking reached after each transition from U has fired at least once, \dots μ_k the first marking reached after each transition from U has fired at least k times.

Let V_n be a nonempty set of the form $V_n = \{y \in \mathbb{N}^n : \exists y_i \in V_n, y \neq y_i, y \geq y_i \text{ or } y \leq y_i\}$. Next it is proved by induction that V_n is finite (i.e. it cannot have infinitely many elements). Assume that any V_{n-1} is finite. Then, let $y_{s,n} \in V_n$; $V_n \subseteq \bigcup_{k,u} C_{k,u}$, where $C_{k,u} = \{y \in \mathbb{N}^n : y(j_k) = u, y(i_k) > y_{s,n}(i_k), \exists y_i \in V_n, y \neq y_i, y \geq y_i \text{ or } y \leq y_i\}$ is defined for $0 \leq u < y_{s,n}(j_k)$ and $k = 1, 2 \dots n(n-1)$ corresponds to the possibilities in which $i_k \neq j_k$, $0 \leq i_k, j_k \leq n$ can be chosen. The induction assumption implies that each $C_{k,u}$ is finite, because the component j_k of the vectors is fixed and only the remaining $n-1$ can be varied. So V_n is finite.

Let \mathcal{M} be recursively constructed as follows: initially $\mathcal{M}_0 = \{\mu_0\}$; for all i , $\mathcal{M}_i = \mathcal{M}_{i-1} \cup \{\mu_i\}$ if $\exists y \in \mathcal{M} : y \geq \mu_i$ or $y \leq \mu_i$ and else $\mathcal{M}_i = \mathcal{M}_{i-1}$. The previous paragraph showed that $\exists n_0 \in \mathbb{N} : \forall k > n_0, \mathcal{M}_k = \mathcal{M}_{n_0}$. Let $\mathcal{M} = \mathcal{M}_{n_0}$ and $\widetilde{\mathcal{M}} = \{y \in \mathbb{N}^n : \exists y_x \in \mathcal{M}, y \leq y_x\}$. Both are finite sets.

Here it is shown that $\exists i, j, 0 \leq i < j$, such that $\mu_i \leq \mu_j$ leads to contradiction. Assuming the contrary, $\forall k > 0 \exists y_x \in \mathcal{M}$ such that $\mu_{k+n_0} \leq y_x$ and $\mu_{k+n_0} \neq y_x$. If $y \in \mathbb{N}^n, y_x \in \mathcal{M}$ and $y_x \geq y$, then for u such that $u \not\geq y_x$ and $u \not\leq y_x$ either $y \leq u$ or both $y \not\leq u$ and $y \not\geq u$; for u such that $u \not\geq y$ and $u \not\leq y$ either $y_x \geq u$ or both $y_x \not\geq u$ and $y_x \not\leq u$. Let $\mathcal{M}^{(1)}$ be constructed in a similar way as \mathcal{M} , but starting from $\mathcal{M}_0^{(1)} = (\mathcal{M} \cup \{y\}) \setminus \{u \in \mathcal{M} : u \geq y\}$, where $y = \mu_{1+n_0}$, and using μ_{n_0+i} instead of μ_i for $\mathcal{M}_i^{(1)}$. For the same reason the construction ends in finitely many steps. Also, $\mathcal{M}^{(1)} \subseteq \widetilde{\mathcal{M}}$ and $\exists n_{0,1}$ such that $\forall k > 0 \exists y_x \in \mathcal{M}$ such that $\mu_{k+n_{0,1}} \leq y_x$ and $\mu_{k+n_{0,1}} \neq y_x$. So we can continue in the same way with $\mathcal{M}^{(2)}, \dots \mathcal{M}^{(j)}$, also subsets of $\widetilde{\mathcal{M}}$. However these operations cannot be repeated infinitely often: $j \leq N$, where N is the cardinality of $\widetilde{\mathcal{M}}$, because $\mathcal{M}^{(j)}$ contains at least one element from $\widetilde{\mathcal{M}} \setminus \bigcup_{i=1}^{j-1} \mathcal{M}^{(i)}$. (This is so because $y \leq u, y \neq u, u \in \mathcal{M}^{(i)} \Rightarrow y \notin \mathcal{M}^{(i)}$, also $u \in \mathcal{M}^{(i)} \setminus \mathcal{M}^{(i-1)} \Rightarrow \exists v \in \mathcal{M}^{(i-1)} : v \geq u$, hence $\exists u \in \mathcal{M}^{(i)} : y \leq u$ implies $\exists v \in \mathcal{M} : y \leq v$.) So, $\mathcal{M}^{(j+1)}$ cannot be constructed for some j , which implies $\mu_{1+n_{0,j}} \not\leq u, \forall u \in \mathcal{M}^{(j)}$, which is contradiction.

Therefore $\exists j, k, j < k$, such that $\mu_j \leq \mu_k$. Let $x = q_k - q_j$. Then $\mu_k - \mu_j \geq 0 \Rightarrow D \cdot x \geq 0$, and by construction $x \geq 0$ and $x(i) > 0 \forall t_i \in U$. \square

Theorem 3.2 [27] *A Petri net is (partially) repetitive if and only if a vector x of positive (nonnegative) integers exists, such that $D \cdot x \geq 0, x \neq 0$.*

Proof: Necessity: The proof follows immediately from Lemma 3.1. Sufficiency (cf. [27]): Consider the finite firing sequence σ_1 in which we fire $x(1)$ times t_1 , then $x(2)$ times t_2 , and so on. Let n be the dimension of x , $X = \sum_{i=1}^n x(i)$ and q_i for $i \in \overline{1, X}$ the firing vectors after each transition from σ_1 is fired (note that $q_X = x$). Then the initial marking defined by $\mu_0(k) = \max\{0, -\min_{i \in \overline{1, X}} \{(D \cdot q_i)(k)\}\}$, $k = \overline{1, n}$, enables σ_1 . Since $\mu_X = \mu_0 + D \cdot x$, and so $\mu_X \geq \mu_0$, μ_X enables σ_1 too. Now it is clear that μ_0 enables $\sigma = \sigma_1 \sigma_1 \sigma_1 \dots$, which is an infinite sequence in which each transition t_k s.t. $x(k) \neq 0$ appears infinitely often, and so the net is (partially) repetitive. \square

It is not always possible to enforce liveness or to prevent deadlock in a Petri net. This may happen because the initial marking is inappropriate or because the structure of the Petri net is incompatible with the supervision purpose. The next corollary characterizes the structure of Petri nets that allow supervision for deadlock prevention and liveness enforcement, respectively. It shows that Petri nets in which liveness is enforceable are repetitive, and Petri nets in which deadlock is avoidable are partially repetitive.

Corollary 3.1 *Let $\mathcal{N} = (P, T, F, W)$ be a Petri net.*

- (a) *Initial markings μ_0 exist such that deadlock can be prevented in (\mathcal{N}, μ_0) if and only if \mathcal{N} is partially repetitive.*
- (b) (cf. [30]) *Initial markings μ_0 exist such that liveness can be enforced in (\mathcal{N}, μ_0) if and only if \mathcal{N} is repetitive.*

Proof: (a) If deadlock can be avoided in (\mathcal{N}, μ_0) then μ_0 enables some infinite firing sequence σ , and by definition \mathcal{N} is partially repetitive.

On the other hand, if \mathcal{N} is partially repetitive, then by theorem 3.2 there is a nonnegative integer vector x , $x \neq 0$ such that $Dx \geq 0$. Let σ_x be a firing sequence associated to a firing vector $q = x$ and let q_1 denote the firing vector after the first transition of σ_x fired, q_2 after the first two fired, and so on to $q_k = q$. The incidence matrix D can be written as $D = D^+ - D^-$, where D^+ and D^- correspond to the weights $W(t, p)$ and $W(p, t)$, respectively. If the rows of the D^- are $d_1^T, d_2^T, \dots, d_{|P|}^T$, then a marking which enables σ_x is

$$\mu_0(p_i) = -\min(0, \min_{j=1 \dots k} d_j^T q_j) \quad i = 1 \dots |P| \quad (3)$$

At least one deadlock prevention strategy exists for μ_0 : to allow only the firing sequence $\sigma_x, \sigma_x, \sigma_x, \dots$ to fire. This infinite firing sequence is enabled by μ_0 because $\mu_0 + Dx \geq \mu_0$ and μ_0 enables σ_x .

(b) The proof is similar to (a). □

Given a Petri net $\mathcal{N} = (P, T, F, W)$, an initial marking μ_0 and two supervisory policies $\Xi_1 : U_1 \rightarrow 2^T$ and $\Xi_2 : U_2 \rightarrow 2^T$, where $U_1, U_2 \subseteq \mathbb{N}^{|P|}$ and $\mu_0 \in U_1 \cap U_2$, Ξ_1 is said to be more **permissive** than Ξ_2 , or equivalently Ξ_1 is less **restrictive** than Ξ_2 , if $\forall \mu \in U_1 \cap U_2: \Xi_2(\mu) \subseteq \Xi_1(\mu)$.

Corollary 3.2 *Let $\mathcal{N} = (P, T, F, W)$ be a Petri net and D its incidence matrix. Let σ_1 and σ_2 be firing sequences and $(P_1), (P_2)$ the two properties below:*

- (P_1) : $(\exists \sigma_1 \exists \mu'_1, \mu_1 \in \mathcal{R}(\mathcal{N}, \mu) \text{ s.t. } \mu_1[\sigma_1 > \mu'_1 \text{ and } \mu'_1 \geq \mu_1)$
- (P_2) : $(\exists \sigma_2 \exists \mu'_2, \mu_2 \in \mathcal{R}(\mathcal{N}, \mu) \text{ s.t. } \mu_2[\sigma_2 > \mu'_2, \mu'_2 \geq \mu_2 \text{ and all transitions of } T \text{ appear in } \sigma_2)$

- (a) *Deadlock can be prevented in (\mathcal{N}, μ) if and only if (P_1) is true.*
- (b) *Liveness can be enforced in (\mathcal{N}, μ) if and only if (P_2) is true.*
- (c) (i) *Nonzero nonnegative integer vectors x exist such that $D \cdot x \geq 0$ and all of them have no null entries if and only if all supervisory policies which prevent deadlock also enforce liveness.*
- (ii) *Consider an arbitrary initial marking μ_0 . All supervisory policies which prevent deadlock in (\mathcal{N}, μ_0) and which are more permissive than any supervisory policy which enforces liveness in (\mathcal{N}, μ_0) , enforce liveness as well if and only if for all markings $\mu \in \mathcal{R}(\mathcal{N}, \mu_0)$, if (P_1) is true then (P_2) is true.*

Proof: (a) If (P_1) is true, then a deadlock prevention strategy is to allow only a firing sequence that leads from μ to μ_1 , and then only the infinite firing sequence $\sigma_1, \sigma_1, \sigma_1, \dots$. Furthermore, if deadlock can be prevented, \mathcal{N} is partially repetitive by Corollary 3.1(a), so $x \geq 0$ exists such that $x \neq 0$ and $Dx \geq 0$, and following the proof of Corollary 3.1(a), a marking μ can be chosen as in equation (3) for the sequence σ_x . Then (P_1) is true by taking $\mu_1 = \mu$ and $\sigma_1 = \sigma_x$.

(b) The proof is similar to (a).

(c) (i) “ \Rightarrow ” Let μ_0 be the initial marking and let Ξ be an arbitrary supervisory policy which prevents deadlock in (\mathcal{N}, μ_0) . By part (a), (P_1) is true for all $\mu \in \mathcal{R}(\mathcal{N}, \mu_0, \Xi)$. Let x_1 be the firing vector associated to the firing sequence σ_1 from (P_1) for some marking μ that was reached. In (P_1) , $\mu'_1 \geq \mu_1$ implies $Dx_1 \geq 0$, so x_1 does not contain null elements. Hence σ_1 includes all transitions of the net. Because μ was arbitrary, and μ_1 reached from μ enables σ_1 , this shows that for all reachable markings μ no transition is dead. So Ξ also enforces liveness.

(i) “ \Leftarrow ” Assume the contrary. Then there is a nonnegative integer vector x such that $Dx \geq 0$ and x has some of its elements zero. Let Ξ be a deadlock prevention policy for (\mathcal{N}, μ_0) , where μ_0 is such that it enables σ_x , a transition sequence that contains $x(i)$ times each of the transitions t_i of the net. If Ξ is defined to allow only the repeated firing $\sigma_x \sigma_x \sigma_x \dots$, then deadlock is prevented but liveness is not enforced, since σ_x does not include all transitions of the net. Contradiction.

(ii) “ \Rightarrow ” Assume the contrary. Then there is a supervisory policy Ξ which prevents deadlock and $\exists \mu \in \mathcal{R}(\mathcal{N}, \mu_0, \Xi)$ such that (P_1) is true and (P_2) is not. Then by part (b), (\mathcal{N}, μ) cannot be made live, so Ξ does not enforce liveness, which is a contradiction.

(ii) “ \Leftarrow ” Let Ξ be a supervisory policy which prevents deadlock in (\mathcal{N}, μ_0) . The proof checks that for all $\mu \in \mathcal{R}(\mathcal{N}, \mu_0, \Xi)$ there is a transition sequence enabled by μ whose firing is accepted by Ξ and which includes all transitions. Let $\mu \in \mathcal{R}(\mathcal{N}, \mu_0, \Xi)$. Because deadlock is prevented, (P_2) is true since (P_1) is true. Let Ξ_L be the supervisory policy that enforces liveness in (\mathcal{N}, μ_0) by firing $\sigma \sigma' \sigma_2 \sigma_2 \sigma_2 \dots$, where $\mu_0[\sigma > \mu[\sigma' > \mu_2$, and σ_2 and μ_2 are the variables from (P_2) . Because Ξ is more permissive than any liveness enforcing policy, Ξ is more permissive than Ξ_L . Thus Ξ allows $\sigma' \sigma_2$ to fire from μ . Therefore all transitions appear in some firing sequence enabled by μ and allowed by Ξ . \square

The most important part of Corollary 3.2 is part (c), because it gives some insight about the relation between deadlock prevention and liveness enforcement. We use Corollary 3.2(c) to characterize two classes of Petri nets for which the deadlock prevention procedure introduced in section 5 also enforces liveness. Figure 3(a) shows an example for part (c)-(i), in which all nonnegative vectors x such that $Dx \geq 0$ are a linear combination with nonnegative coefficients of $x_1 = [1, 2, 1, 1]^T$ and $x_2 = [2, 3, 3, 3]^T$. Figure 3(b) shows an example for part (c)-(ii) of Corollary 3.2. Indeed, all markings μ that enable any of t_1, t_2 or t_4 satisfy (P_2) . Also, a marking that enables only t_3 either leads to deadlock or enables the sequence t_3, t_4 and hence satisfies (P_2) . For instance, the deadlock prevention policy that repeatedly fires t_2, t_1 does not enforce liveness because it does not satisfy the requirement of Corollary 3.2(c)-(ii) to be more permissive than any liveness enforcing supervisors.

Given a Petri net (\mathcal{N}, μ_0) and a supervisory policy Ξ , let $\mathcal{R}(\mathcal{N}, \mu_0, \Xi)$ denote the set of markings reachable from the initial marking μ_0 when (\mathcal{N}, μ_0) is supervised by Ξ . A vector $x \in S \subseteq \mathbb{R}^n$ has **maximum support** if no other vector in S has more nonzero entries than x . The **minimum support** is similarly defined.

Corollary 3.3 *Consider a Petri net $\mathcal{N} = (P, T, F, W)$ which is not repetitive, and let D be the incidence matrix. Then at least one transition exists such that for any given initial marking it cannot fire infinitely often. Let T_D be the set of all such transitions. There are initial markings μ_0 and a supervisory policy Ξ such that $\forall \mu \in \mathcal{R}(\mathcal{N}, \mu_0, \Xi)$, no transition in $T \setminus T_D$ is dead and there is a nonnegative integer vector x such that $Dx \geq 0$, $x(i) \neq 0 \forall t_i \in T \setminus T_D$ and $x(i) = 0 \forall t_i \in T_D$.*

Proof: There is an integer vector $x \geq 0$ with *maximum support* such that $Dx \geq 0$, which means that for all

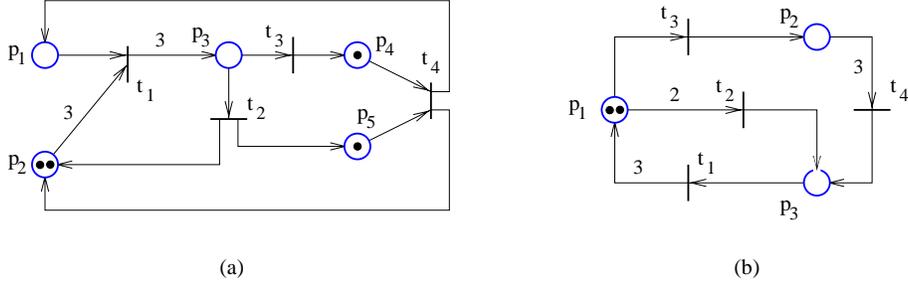


Figure 3: Examples for Corollary 3.2(c): (a) for part (i) and (b) for part (ii)

integer vectors $w \geq 0$ such that $Dw \geq 0$, $\|w\| \subseteq \|x\|$. Indeed if $y \geq 0$, $z \geq 0$ are integer vectors and $Dy \geq 0$, $Dz \geq 0$, then $D(z + y) \geq 0$ and so $y + z \geq 0$ and $\|y\|, \|z\| \subseteq \|y + z\|$.

If $t_j \in T$ can be made live for some initial marking by using an appropriate supervisory policy, there is a marking that enables an infinite firing sequence σ such that t_j appears infinitely often in σ . Therefore by Lemma 3.1 $\exists y \geq 0$ such that $Dy \geq 0$ and $y(j) > 0$. Since x has maximum support, $\|y\| \subseteq \|x\|$ and so $t_j \in \|x\|$. This proves that all transitions that can be made live under some circumstances are in $\|x\|$. Therefore the transitions in $T_D = T \setminus \|x\|$ cannot be made live under any circumstances.

Let σ_x be a firing sequence associated with x , i.e. every $t_i \in T$ appears $x(i)$ times in σ_x . Then there is a marking μ_0 given by equation (3) which enables the infinite firing sequence $\sigma_x, \sigma_x, \sigma_x, \dots$. Also, we may choose Ξ to restrict all possible firings to the former infinite firing sequence, so all transitions in $\|x\|$ can be made live.

The remaining claim to be proved is that $T_D \neq \emptyset$. Assume the contrary. Then $T = \|x\|$, so all transitions in T can simultaneously be made live for an appropriate initial marking, which contradicts the fact that \mathcal{N} is not repetitive. \square

Corollary 3.3 shows that for any Petri net structure which is not repetitive, there is a set of transitions T_D which cannot be made live under any circumstances. It also shows that all other transitions can be simultaneously made live for appropriate initial markings and supervisory policies. Note that for repetitive Petri nets $T_D = \emptyset$. Another special case is $T_D = T$, which occurs when the Petri net is not even partially repetitive, and so deadlock can not be avoided for any initial marking.

It was already shown that only repetitive Petri nets can be supervised for liveness. We are interested in the existence of a similar property for nonrepetitive Petri nets. Corollary 3.3 shows that the best a supervisory policy could do is to enforce that all transitions in $T \setminus T_D$ are live. Therefore the liveness property for partially repetitive Petri nets is that all transitions in $T \setminus T_D$ are live.

In what follows we define a class of subnets of a Petri net, which we call *active subnets*. An active subnet can be made live by supervision for appropriate initial markings.

Definition 3.6 Let $\mathcal{N} = (P, T, F, W)$ be a Petri net, D the incidence matrix and $T_D \subseteq T$ be the set of all transitions which cannot fire infinitely often given any initial marking. $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$ is an **active subnet** of \mathcal{N} if $P^A = T^A \bullet$, $F^A = F \cap \{(T^A \times P^A) \cup (P^A \times T^A)\}$, W^A is the restriction of W to F^A and T^A is the set of transitions with nonzero entry in some nonnegative vector x which satisfies $Dx \geq 0$. The **maximal active subnet** of \mathcal{N} is the active subnet $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$ such that $T^A = T \setminus T_D$. A

minimal active subnet has the property that the vector x defining it has minimum support and is nonzero.

Definition 3.7 Given an active subnet \mathcal{N}^A of a Petri net \mathcal{N} , a siphon of \mathcal{N} is said to be an **active siphon** with respect to \mathcal{N}^A if it is, or includes, a siphon of \mathcal{N}^A . An active siphon is **minimal** if it does not include another active siphon with respect to the same active subnet.

For instance, consider the Petri net in figure 4(a). The maximal active subnet has the set of transitions $T^A = \{t_2, t_3, t_6, t_7, t_9\}$ and the set of places $P^A = \{p_2, p_3, p_5, p_6, p_7, p_8\}$. There are two other active subnets, both minimal, \mathcal{N}_1^A and \mathcal{N}_2^A , which have $P_1^A = \{p_2, p_3\}$, $T_1^A = \{t_2, t_3\}$, $P_2^A = \{p_5, p_6, p_7, p_8\}$ and $T_2^A = \{t_6, t_7, t_9\}$. The maximal active subnet is shown in figure 4(b). The Petri net has four minimal siphons: $S_1 = \{p_1, p_8\}$, $S_2 = \{p_1, p_2, p_3\}$, $S_3 = \{p_1, p_4, p_5, p_6\}$ and $S_4 = \{p_5, p_6, p_7\}$. S_2 is a minimal active siphon with respect to \mathcal{N}^A and \mathcal{N}_1^A , S_3 is a minimal active siphon with respect to \mathcal{N}^A and \mathcal{N}_2^A , and S_4 is a minimal active siphon with respect to \mathcal{N}^A and \mathcal{N}_2^A . S_1 is not an active siphon with respect to any of the active subnets. Another example is in figure 4(c). The maximal active subnet is shown in figure 4(d). There is no other nonempty active subnet. The minimal active siphons are: $S_1 = \{p_1, p_4, p_7\}$, $S_2 = \{p_2, p_5, p_7\}$, $S_3 = \{p_3, p_5, p_7\}$ and $S_4 = \{p_6, p_7\}$. Among them, only S_4 is a minimal siphon of the Petri net.

Proposition 3.4 A siphon which contains places from an active subnet is an active siphon with respect to that subnet.

Proof: Using the notations from Definition 3.6, let S be a siphon such that $S \cap P^A \neq \emptyset$. $\bullet S \subseteq S \bullet$ implies that $\bullet S \cap T^A \subseteq S \bullet \cap T^A$. If $t \in T^A$ and for some $p \in P$: $t \in p \bullet$, then $p \in P^A$, by Definition 3.6. Hence $S \bullet \cap T^A \subseteq (S \cap P^A) \bullet$ and so $S \bullet \cap T^A = (S \cap P^A) \bullet \cap T^A$. Note also that $\bullet(S \cap P^A) \cap T^A \subseteq \bullet S \cap T^A$. Therefore $\bullet S \subseteq S \bullet$ implies $\bullet(S \cap P^A) \cap T^A \subseteq (S \cap P^A) \bullet \cap T^A$, which proves that $S \cap P^A$ is a siphon of \mathcal{N}^A . \square

The significance of the active subnets for deadlock prevention can be seen in the following results. First we prove a technical result.

Lemma 3.2 Let $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$ be an active subnet of \mathcal{N} . Given a marking μ of \mathcal{N} and μ^A its restriction to \mathcal{N}^A , if $t \in T^A$ is enabled in \mathcal{N}^A , then t is enabled in \mathcal{N} .

Proof: By definition, there is a nonnegative integer vector $x \geq 0$ such that $Dx \geq 0$ (D is the incidence matrix) and $x(i) > 0$ for $t_i \in T^A$ and $x(i) = 0$ for $t_i \in T \setminus T^A$. This implies that there are markings such that the transitions of T^A can fire infinitely often, without firing other transitions (see proof of Corollary 3.1.) If t is not enabled in \mathcal{N} , there is $p \in \bullet t$ such that $p \notin P^A$ (the \bullet operators are taken with respect to \mathcal{N} , not \mathcal{N}^A), since t is enabled in \mathcal{N}^A . Note that $p \notin P^A$ implies $\bullet p \cap T^A = \emptyset$. If $\bullet p = \emptyset$, t cannot fire infinitely often, which contradicts the definition of T^A , since $t \in T^A$. If $t_x \in \bullet p$, the transitions of T^A cannot fire infinitely often without firing t_x , which again contradicts the definition of T^A . Therefore t is also enabled in \mathcal{N} . \square

In general we may denote the *maximal active subnet* of a Petri net simply as the *active subnet* of the Petri net, and so the active siphons are in general taken with respect to the maximal active subnet. Considering the maximal active subnet instead of another active subnet is preferable in problems in which the objective is enforcing transition liveness rather than total deadlock prevention. Note that in a repetitive Petri net all siphons are active with respect to the maximal active subnet. The next result is a generalization of the well known Proposition 3.1.

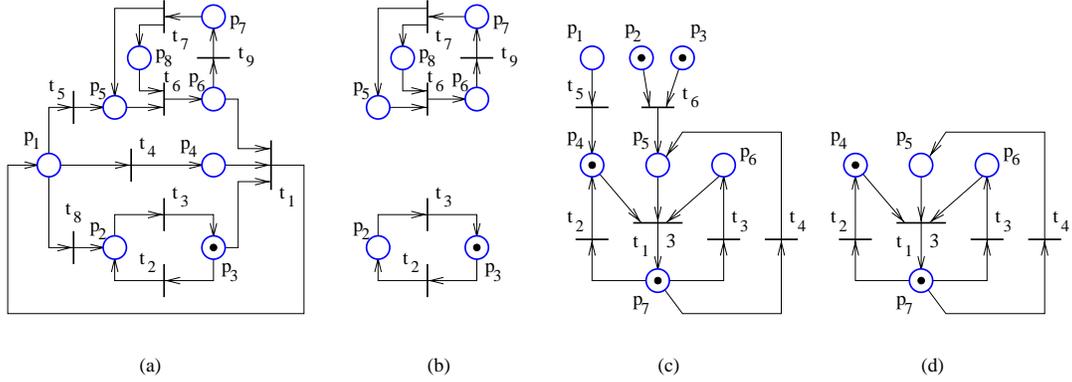


Figure 4: The maximal active subnet for the Petri net in (a) is (b), and for the Petri net in (c) is (d).

Proposition 3.5 *Let \mathcal{N}^A be an arbitrary, nonempty, active subnet of a PT-ordinary Petri net \mathcal{N} . If μ is a deadlock marking of \mathcal{N} , then there is at least one empty minimal active siphon with respect to \mathcal{N}^A .*

Proof: Since μ is a deadlock marking and $\mathcal{N} = (P, T, F, W)$ is PT-ordinary, $\forall t \in T \exists p \in \bullet t: \mu(p) = 0$. The active subnet is built in such a way that if the marking μ restricted to the active subnet enables a transition t , then μ enables t in the total net (Lemma 3.2.) Therefore, because the total net (\mathcal{N}, μ) is in deadlock, the active subnet is too. In view of Proposition 3.1, let s be an empty minimal siphon of the active subnet. Consider s in the total net. If s is a siphon of the total net, then s is also a minimal active siphon; therefore the net has a minimal active siphon which is empty. If s is not a siphon of the total net: $\bullet s \setminus T^A \neq \emptyset$. Let S be the set recursively constructed as follows: $S_0 = s$, $S_i = S_{i-1} \cup \{p \in \bullet(S_{i-1} \setminus S_{i-1} \bullet) : \mu(p) = 0\}$, where μ is the (deadlock) marking of the net. In other words S is a completion of s with places with null marking such that S is a siphon. By construction S is an active siphon and is empty for the marking μ . Hence an empty minimal active siphon exists. \square

The practical significance of Proposition 3.5 is that it provides a support for doing deadlock prevention, since deadlock is not possible when all active siphons with respect to a nonempty active subnet cannot become empty. A less restrictive condition is given in the next result.

Proposition 3.6 *Deadlock is unavoidable for the marking μ if for all minimal active subnets \mathcal{N}^A there is an empty active siphon with respect to \mathcal{N}^A .*

Proof: For any empty (active or not) siphon, all transitions in the postset of that siphon are empty. Therefore for all active minimal subnets, some of their transitions are dead. In the proof of Lemma 3.1 it is shown that if deadlock is avoidable, there is a reachable marking μ' and a finite firing sequence σ such that $\mu[\sigma > \mu'$, where $\mu' \geq \mu$. Moreover, this implies that μ enables $\sigma\sigma \dots \sigma \dots$. Let q be the firing count vector for σ . Then $Dq \geq 0$. Consider x to be a vector defined as follows. If the active subnet for q is minimal then $x = q$; else choose x such that $\|x\| \subset \|q\|$, $x \neq 0$, $x \geq 0$, $Dx \geq 0$ and the active subnet associated to x is minimal. The active subnet defined by x is minimal, and therefore has an empty siphon. This implies that some of the transitions in $\|x\|$ are dead. In view of Lemma 3.2, this contradicts the fact that all transitions of σ can fire in the Petri net. \square

The previous result gives support for maximally permissive deadlock prevention. Deadlock is avoidable in a PT-ordinary Petri net as long as it can be insured that for all allowed markings, there is a minimal active subnet such that all minimal active siphons have a token.

Consider again the examples in figure 4(a) and (c). Proposition 3.5 allows us to detect that the Petri net in figure 4(a) is not in deadlock, if we take the active siphons with respect to \mathcal{N}_1^A . Also, the Petri net in figure 4(c) is not in deadlock, as none of the active siphons is empty. Note that in both cases Proposition 3.1 cannot say whether it is or not deadlock, as some siphons are empty.

Further on we prove an existence result for supervisors which enforce linear constraints.

Theorem 3.3 *Let \mathcal{N} be a Petri net. Let Ξ be a quality like liveness, deadlock-freedom, a.o., that has the property that for any marking μ_x so that Ξ can be enforced for μ_x , Ξ can be enforced for all markings $\mu \geq \mu_x$. If Ξ can be enforced in \mathcal{N} for some markings, then \mathcal{N} can be supervised with linear constraints to enforce Ξ for some markings.*

Proof: The set of markings acceptable for the supervisory policy Σ enforcing Ξ is a subset of the set of markings such that Ξ holds in \mathcal{N} . We call μ a minimal marking accepted by Σ if there is no acceptable marking μ_i s.t. $\mu_i \leq \mu$ and $\mu_i \neq \mu$. Let \mathcal{M} be the set of minimal markings accepted by Σ . We claim that \mathcal{M} is finite. Assume the contrary. Let $\mu_k \in \mathcal{M}$. Then for all other markings $\mu_i \in \mathcal{M}$ there are $p_x, p_y \in P$ (P is the set of places of \mathcal{N}) such that $\mu_i(p_x) > \mu_k(p_x)$ and $\mu_i(p_y) < \mu_k(p_y)$. Further on, we reach contradiction by using a similar reasoning as in the proof of Lemma 3.1. Since \mathcal{M} is finite, we may find linear constraints which enforce the condition that all reachable markings μ are in the space $\mu \geq \mu_{i_1} \vee \mu \geq \mu_{i_2} \vee \dots \vee \mu \geq \mu_{i_N}$, where $\mathcal{M} = \{\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_N}\}$. For instance a rough solution is to use a single linear constraint given by the inequality $\mu \geq \mu_{max}$, where $\mu_{max}(p_i) = \max_{\mu_k \in \mathcal{M}} \mu_k(p_i) \quad \forall p_i \in P$. \square

4 Preliminaries to the Deadlock Prevention Method

4.1 A Transformation of Petri Nets to PT-ordinary Petri Nets

Because Proposition 3.1 and Proposition 3.5 apply to PT-ordinary Petri nets, we are interested in using a transformation to PT-ordinary Petri nets. In principle Proposition 3.2 could have been used instead, as it applies to generalized Petri nets, but it is difficult to express its requirement in terms of linear inequalities.

We use a modified form of the transformation from Lautenbach and Ridder (1996), and we call it the **PT-transformation**. Let $\mathcal{N} = (P, T, F, W)$ be a Petri net. Transitions $t_j \in T$ such that $W(p, t_j) > 1$ for some $p \in \bullet t_j$ may be **split** in (replaced with) several new transitions. When a transition t_j is split, m new transitions $t_{j,0}, t_{j,1}, t_{j,2}, \dots, t_{j,m-1}$ are created. Together they emulate the functioning of t_j in the original net \mathcal{N} . Let $\mathcal{N}' = (P', T', F', W')$ be the new Petri net obtained by splitting t_j . For our purposes it is very convenient to denote $t_{j,0}$ simply by t_j . In this way T' contains all transitions of T , rather than only $T \setminus \{t_j\}$. Therefore, according to this notation, when t_j is split, it is replaced with $t_j, t_{j,1}, t_{j,2}, \dots, t_{j,m-1}$. The split operation is defined as follows.

The transition t_j is **split** in m transitions, where $m = \max\{W(p, t_j) : (p, t_j) \in F\}$. The new transitions which replace t_j in \mathcal{N}' are named $t_j, t_{j,1}, t_{j,2}, \dots, t_{j,m-1}$. Also, $m - 1$ new places are added: $p_{j,1}, p_{j,2}, \dots, p_{j,m-1}$. In what follows, to avoid confusion, the preset/postset operator is denoted by \bullet for the evaluations in \mathcal{N} , and by \bullet' for the evaluations in \mathcal{N}' . The next relations define the connections of the new transitions and places.

- (i) $\bullet'p_{j,i} = t_{j,i}$ and $t_{j,i}\bullet' = p_{j,i}$ for $i = 1 \dots m - 1$, $p_{j,i}\bullet' = t_{j,i-1}$ for $i = 2 \dots m - 1$ and $p_{j,1}\bullet' = t_j$.
- (ii) $\bullet't_{j,i} = \{p \in \bullet t_j : W(p, t_j) > i\} \cup X$, for $i = 1 \dots m - 1$, where $X = \emptyset$ for $i = m - 1$ and $X = \{p_{j,i+1}\}$ otherwise.
- (iii) $\bullet't_j = \bullet t_j \cup \{p_{j,1}\}$ and $t_j\bullet' = t_j\bullet$.
- (iv) $\forall p \in \bullet't_{j,i}: W'(p, t_{j,i}) = 1$ and $W'(t_{j,i}, p_{j,i}) = 1$, for $i = 1 \dots m - 1$.
- (v) $\forall p \in \bullet't_j: W'(p, t_j) = 1$ and $\forall p \in t_j\bullet': W'(t_j, p) = W(t_j, p)$.

Note that the connections of t_j in \mathcal{N}' are the same as in \mathcal{N} , except for an additional transition arc and for the weights of the input arcs.

The **PT-transformation** consists in splitting all transitions t for which $W(p, t) > 1$ for some $p \in \bullet t$. In this way the transformed Petri net is PT-ordinary. A few properties are apparent:

$$|p_{j,i}\bullet| = |\bullet p_{j,i}| = 1 \quad i = 1 \dots m - 1 \quad (4)$$

$$|t_{j,i}\bullet| = 1 \quad i = 1 \dots m - 1 \quad (5)$$

We use the convention that a split transition t_j is also a transition of the PT-transformed net, since we denote $t_{j,0}$ by t_j .

Let P_T be the set of places of the transformed net. To a marking μ of the original net we associate in the transformed net a marking μ_T such that $\mu_T(p) = \mu(p) \forall p \in P$ and $\mu_T(p) = 0 \forall p \in P_T \setminus P$.

Firing of an unsplit transition t_j in the original net corresponds to firing the same transition in the transformed net. Firing of a split transition t_j in the original net corresponds in the transformed net to firing the sequence $t_{j,m} \dots t_{j,1}, t_j$. For similar initial markings μ and μ_T (see above) the firing sequence σ_T corresponds to a firing sequence σ , such that every split transition t_j in σ is replaced in σ_T by its components $t_{j,m} \dots t_{j,1}, t_j$, and firing σ in \mathcal{N} produces a similar marking μ' to the marking μ'_T reached by firing σ_T in the transformed net.

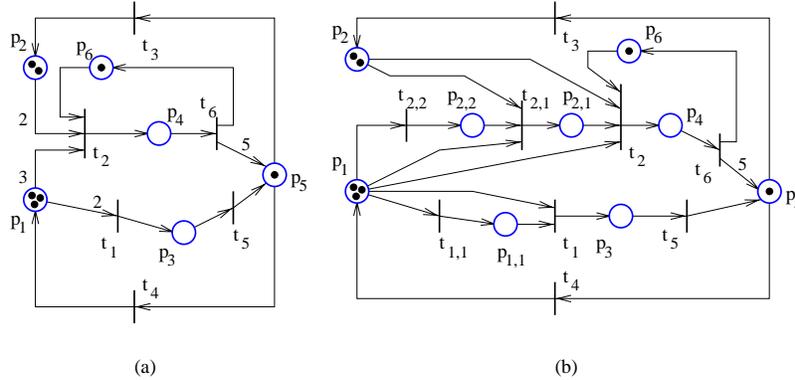


Figure 5: (a) Original net and (b) transformed net.

Figure 5 shows an example, in which the transition t_1 is split in $t_{1,1}$ and t_1 , and the transition t_2 is split in $t_{2,1}$, $t_{2,2}$ and t_2 . Firing t_1 in the original net corresponds to firing $t_{1,1}$ and t_1 in the transformed net, and firing t_2 in the original net corresponds to firing $t_{2,2}$, $t_{2,1}$ and t_2 in the transformed net. Another example is the Petri net of figure 8(a), which is changed as shown in figure 8(b) after it is PT-transformed. The transition t_2 is replaced by $t_{2,1}$ and t_2 , and t_3 by $t_{3,1}$ and t_3 .

4.2 Petri Net Supervisors Based on Place Invariants

This section outlines a method of [24] and [37]: supervisory control of Petri nets based on linear constraints. We first consider the case of fully controllable and observable Petri nets.

4.2.1 Fully Controllable and Observable Petri Nets

The control problem is to enforce a set of n_c linear constraints to prevent reaching undesired markings of a Petri net. The constraints are written in a matrix form:

$$L \cdot \mu_p \leq b \quad (6)$$

where L is an integer $n_c \times n$ matrix (n_c - the number of constraints, n - the number of places of the given Petri net), b is an integer column vector and μ_p denotes a marking vector.

Let μ_c be a vector of n_c nonnegative slack variables, defined as:

$$\mu_c = b - L \cdot \mu_p \quad (7)$$

Let μ_{c0} be the slack variables that correspond to the initial marking μ_{p0} , that is $\mu_{c0} = b - L\mu_{p0}$. Let q be the firing count vector associated to the transitions fired to change the marking from μ_{p0} to μ_p . If D_p the incidence matrix, we have: $\mu_p = \mu_{p0} + D_p q$. Therefore $\mu_c = b - L \cdot (\mu_{p0} + D_p \cdot q)$, which also can be written as:

$$\mu_c = \mu_{c0} + (-LD_p) \cdot q \quad (8)$$

In consequence μ_c may be regarded as a marking of some additional **control places**, where the extended (supervised) Petri net has a marking vector $\mu = [\mu_p^T, \mu_c^T]^T$, and an incidence matrix $D = [D_p^T, D_c^T]^T$, and where $D_c = -LD_p$.

In the supervised net, initial markings μ_{p0} such that $L \cdot \mu_{p0} \not\leq b$ cannot be considered, since equation (7) shows that in this case μ_{c0} will not be nonnegative, and so not defined. (The marking is by definition nonnegative in conventional Petri nets.) When the constraints are initially satisfied, the initial marking of the control places may be chosen according to equation (7), and therefore the constraints will remain satisfied for any reachable marking, since the D_c part of the incidence matrix prevents any firings which would attempt to make any of the variables of μ_c negative.

The way the constraints are enforced prevents only forbidden markings to be reached, so the supervisor is maximally permissive. The next theorem summarizes the construction above:

Theorem 4.1 *Let a plant Petri net with controllable and observable transitions, incidence matrix D_p and initial marking μ_{p0} be given. A set of n_c linear constraints $L\mu_p \leq b$ are to be imposed. If $b - L\mu_{p0} \geq 0$ then a Petri net controller (supervisor) with incidence matrix $D_c = -LD_p$ and initial marking $\mu_{c0} = b - L\mu_{p0}$ enforces the constraint $L\mu_p \leq b$ when included in the closed loop system $D = [D_p^T, D_c^T]^T$. Furthermore, the supervision is maximally permissive.*

Proof: See [24] and [37]. □

Because $D_c = -LD_p$, every row of $[L, I]$ is a place invariant of the incidence matrix of the closed loop system, D .

4.2.2 Petri Nets with Uncontrollable and Unobservable Transitions

Uncontrollable and/or unobservable events of the plant correspond to uncontrollable and/or unobservable transitions in the Petri net model of the plant. Uncontrollable events cannot be inhibited and unobservable events cannot be observed. As the Petri net supervisor is implemented in the form of control places connected to the plant Petri net, we need to make sure that no control place ever attempts to inhibit an uncontrollable transition enabled in the plant Petri net, and no control place marking is varied by firing unobservable transitions. The constraints $L\mu \leq b$ which satisfy this requirement are called **admissible constraints**. Note that the admissibility of a constraint may depend on the initial marking of the Petri net. (For instance, all constraints are admissible in the trivial case with null initial marking.) In this paper we are interested in constraints which are admissible for all initial markings. It can easily be seen that $L\mu \leq b$ is admissible for all initial markings if and only if the following equations of [24] are true:

$$LD_{uc} \leq 0 \quad (9)$$

$$LD_{uo} = 0 \quad (10)$$

where D_{uc} and D_{uo} denote the columns of the incidence matrix which correspond to uncontrollable and unobservable transitions, respectively. From the viewpoint of this paper all linear constraints that have matrices L that satisfy the conditions above are *admissible*. Such constraints may be enforced as in section 4.2.1. Constraints $L\mu_p \leq b$ which do not satisfy (9) and (10) may be transformed to a new set of constraints $L'\mu_p \leq b'$ such that (i) L' satisfies (9) and (10), and (ii) $\forall \mu_p \in \mathbb{N}^{n_p}: L'\mu_p \leq b' \Rightarrow L\mu_p \leq b$. Unless $\forall \mu_p \in \mathbb{N}^{n_p}: L'\mu_p \leq b' \Leftrightarrow L\mu_p \leq b$, this approach of enforcing $L\mu_p \leq b$ may not be maximally permissive. Note that enforcing linear constraints is maximally permissive in the case of fully controllable and observable Petri nets (Theorem 4.1). Algorithms which transform linear constraints to admissible linear constraints are given in [24].

4.3 Siphon Control Based on Place Invariants

Proposition 3.1 shows that in a PT-ordinary Petri net deadlock is not possible when all siphons are controlled. Also, by Proposition 3.5, deadlock is not possible when all siphons which are active with respect to an active subnet are controlled. Therefore it is important to define a method for siphon control. An easy way to control a siphon is to create a place invariant which controls the siphon. This is the approach we choose. Early references of this approach for siphon control are [2] and [10]. This section presents it as a special case of the supervision method based on place invariants (section 4.2). The operations described in this section do not rely on the fact that the structure they are applied to is a siphon.

4.3.1 Case 1: All Transitions are Controllable and Observable

Let $\mathcal{N} = (P, T, F, W)$ be a Petri net. Given a set of places S , the desired control policy is $\sum_{p \in S} \mu(p) \geq 1$. This constraint is enforced using supervision based on place invariants, as described in [24], [37], and also in section 4.2. In this way an additional place results, denoted C and called **control place**. The place invariant thus created is x , such that $x(i) = 1$ for $p_i \in S$, $x(i_C) = -1$ and $x(i) = 0$ for all other indices, where i_C is the row index of C in the incidence matrix. The invariant corresponds to the equation

$$\mu(C) = \sum_{p \in S} \mu(p) - 1 \quad (11)$$

where the constant (-1) results from the initial marking of the control place. There are several particular cases:

1. $\bullet C = \emptyset$ and $C \bullet \neq \emptyset$: no transition increases the marking of S and there are transitions which decrease the marking of S . In this case C alone makes up a minimal siphon which cannot be controlled (see also [24], p.87-88).
2. $C \bullet \subseteq \bullet S$ (in particular $C \bullet = \emptyset$): no transition can make S token free. Also, $C \bullet \subseteq \bullet S$ if and only if S is a trap. Therefore when S is also a siphon, it is (trap) controlled for all initial markings μ_0 that satisfy $\sum_{p \in S_0} \mu_0(p) \geq 1$.
3. $\bullet C = \emptyset$ and $C \bullet = \emptyset$: $\sum_{p \in S} \mu(p)$ cannot vary, and so there is a place invariant x such that $x(i) = 1 \forall p_i \in S$ and $x(i) = 0 \forall p \in P \setminus S$.

Case (a) detects transitions that cannot be made live when S is a siphon (Corollary 3.3). Case (b) shows the case when S does not need control. This is from a structural viewpoint, as we do not consider whether S does not need control for some initial markings, but rather if S does not need control for all initial markings μ_0 such that $\sum_{p \in S} \mu_0(p) \geq 1$. Therefore a control place will be produced for a siphon that is not a trap, but includes a trap. The control place is useful for all initial markings in which the trap included in the siphon has null marking; for such markings the siphon is not trap controlled.

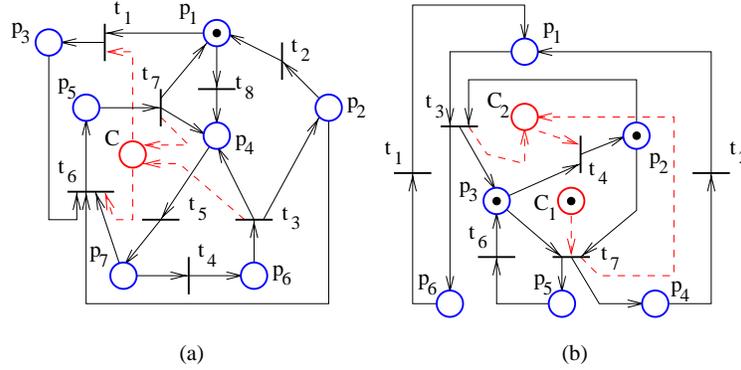


Figure 6: Siphon Control Examples. Connections to control places are dashed.

In figure 6(a) there is a single minimal siphon, $\{p_1, p_2, p_4, p_5, p_6, p_7\}$, which includes the trap $\{p_4, p_5, p_6, p_7\}$. The siphon is not trap controlled because the marking of the trap is 0. The control place C prevents firing t_1 to empty the siphon. The target Petri net of figure 6(b) has two minimal siphons, $\{p_2, p_3, p_5\}$ and $\{p_1, p_3, p_4, p_5, p_6\}$. Their control places are C_1 and C_2 , respectively. C_1 is an example of case (a). Also, the control place C which would result for the minimal siphon $\{p_2, C_2\}$ satisfies $\bullet C = \emptyset$ and $C \bullet = \emptyset$.

By Theorem 4.1, the enforcement of $\sum_{p \in S} \mu_0(p) \geq 1$ is maximally permissive. Because the enforcement of this constraint makes the siphon controlled, there is no other more permissive way to control a siphon. This is not the only way to provide maximally permissive control of a siphon; however, any other way is equivalent. An important quality of this technique is that the supervised net remains a Petri net.

4.3.2 Case 2: Transitions Uncontrollable and/or Unobservable are Present

Let D be the incidence matrix of a Petri net, and let D_{uo} and D_{uc} be D restricted to the columns of unobservable and respectively uncontrollable transitions. In order that the constraint $l^T \mu \geq b$ be admissible, the supervisor enforcing it should not need to detect unobservable transitions or inhibit enabled uncontrollable transitions, and so the constraint is required to satisfy $l^T D_{uo} = 0$ and $l^T D_{uc} \geq 0$. There are methods that allow to transform a constraint in a another constraint, in general more restrictive, which satisfies the last two requirements. Two such methods can be found in [24] and another one in the appendix of this paper. So when a desired constraint $\sum_{p \in S} \mu(p) \geq 1$ is inadmissible, it can be transformed to a constraint of the form $l^T \mu \geq b$. In both the appendix and [24], $b = 1$ (in [24] consider the construction of Lemma 4.10). Therefore the admissible form of the constraint $\sum_{p \in S} \mu(p) \geq 1$ is $\sum_{p \in S} \alpha_p \mu(p) \geq 1$. The algorithm of the appendix is guaranteed to find a solution to this problem if any of the form $l^T \mu \geq b$ exists.

Note that the transformation to admissible constraints is not always possible. There are cases when this is impossible because of limited information due to unobservable transitions and/or limited ability to control firing transitions can make impossible the task to design a supervisor which guarantees that the marking satisfies a certain constraint. Unlike the approach of the case of section 4.3.1, which corresponds to maximally permissive siphon control, this approach is suboptimal in general. Note that when the admissible constraint is obtained in the form $\sum_{p \in S} \alpha_p \mu(p) \geq 1$ with all α_p positive integers, the control of S is maximally permissive, in the sense that the only forbidden markings are the markings for which $\mu(p) = 0 \forall p \in S$. The method from the appendix finds admissible constraints of the form $\sum_{p \in S} \alpha_p \mu(p) \geq 1$, with α_p nonnegative integers, maximizing the number of coefficients α_p which are nonzero. That method is guaranteed to find a solution with all α_p positive whenever such a solution exists.

5 The Deadlock Prevention Method

5.1 Introduction to the Method

Given a target Petri net \mathcal{N}_0 , the deadlock prevention procedure generates a sequence of PT-ordinary Petri nets, $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k$, increasingly enhanced for deadlock prevention. \mathcal{N}_1 is \mathcal{N}_0 transformed to be PT-ordinary. The other Petri nets are largely obtained as follows: in each iteration i the new minimal active siphons of \mathcal{N}_i are controlled, and then, if needed, transitions are split; the resulting PT-ordinary net is \mathcal{N}_{i+1} . The active siphons (see Definition 3.7) of each \mathcal{N}_i are taken with respect to an active subnet \mathcal{N}_i^A computed for every iteration i . Recall, for each controlled siphon a linear marking inequality is enforced. Let $L_i \mu \geq b_i$ be the total set of constraints enforced in \mathcal{N}_i . Because \mathcal{N}_k is the last Petri net in the sequence, it has no uncontrolled active siphons. Therefore \mathcal{N}_k is deadlock free for all initial markings which satisfy $L_k \mu \geq b_k$. Finally, the constraints defined by (L_k, b_k) can be easily translated in constraints in terms of the markings of \mathcal{N}_0 ; these constraints define the supervisor for deadlock prevention in \mathcal{N}_0 .

The user is allowed to transfer to the procedure foreknowledge about the Petri net. This is done by using **initial constraints**. For instance, if an invariant $l^T \mu = c$ is true for all initial markings employed by the user, the constraints $[l, -l]^T \mu \geq [c, -c]^T$ may be specified as initial constraints. The usage of initial constraints $L_I \mu \geq b_I$ could benefit problems in which one of the following is true: (a) the procedure should not generate constraints which require $L_I \mu \not\geq b_I$, (b) less complex supervisors can be obtained if the procedure takes in account that markings such that $L_I \mu \not\geq b_I$ are never reached for all initial markings considered by the user

for the target Petri net, and (c) convergence help is needed.

The deadlock prevention procedure is defined in section 5.4. The sections preceding section 5.4 detail the operations performed by the procedure. Sections 4.1, 4.2 and 4.3 have shown how the Petri nets are transformed to be PT-ordinary, how constraints are enforced and how siphons are controlled. The precise way in which the constraints are generated is considered in section 5.2. In some occasions, initial constraints may be needed to help the procedure converge or to indicate place invariant constraints on the target net. Initial constraints are considered in section 5.2.5. The active subnet \mathcal{N}_i^A of the iteration i is usually the maximal active subnet, but the method may take a smaller subnet in the following cases. The initial constraints may conflict with constraints that the procedure wants to enforce. When a siphon constraint, due to the initial constraints, cannot be enforced, all transitions connected to the siphon are considered to be dead. Therefore they cannot be in the active subnet. A similar situation appears in the case of uncontrollable and unobservable transitions, when no admissible constraint can be found to control a siphon. In this case the procedure considers that all transitions connected to the siphon cannot belong to the active subnet. The computation and the updating of the active subnet is shown in section 5.3.

5.2 Implicit Inequalities

The deadlock prevention procedure gradually restricts the sets of acceptable markings. To each (minimal active) siphon corresponds a linear inequality, which expresses the requirement that the siphon is not empty. As more and more siphons are controlled, the set of acceptable markings is restricted. In section 5.2.1 we consider the form of the place invariants associated to control places. Section 5.2.2 considers the case when the control of a siphon does not require a control place. Section 5.2.3 shows the way in which the procedure constructs the sets of constraints. Section 5.2.4 defines the *implicitly controlled* siphons. In section 5.2.5 we show how initial constraints on the target net are changed by the PT-transformation. Finally, section 5.2.6 considers the details of transforming constraints to admissible constraints.

5.2.1 The Enforced Place Invariants

Consider a siphon S . When the approach of section 4.3 is used, the control place C which results enforces a constraint of the form $\sum_{p \in S} \alpha_p \mu(p) \geq 1$, where $\alpha_p \geq 0$. The most familiar case is when all transitions of S are controllable and observable, in which $\alpha_p = 1 \forall p \in S$. The supervision based on place invariants creates the following place invariant for C : $\mu(C) = \sum_{p \in S} \alpha_p \mu(p) - 1$. The deadlock prevention procedure ensures that the Petri net at the beginning of every iteration is PT-ordinary. However, by adding control places, the net may no longer be PT-ordinary towards the end of the iteration. Therefore the deadlock prevention procedure splits the transitions with arc weights greater than one, in order that the next iteration will have a PT-ordinary net. This operation may change the place invariant of a control place. Proposition 6.6 proves that a place invariant $\mu(C) = \sum_{p \in S} \alpha_p \mu(p) - 1$ is transformed to

$$\mu(C) + \sum_{i=1}^k \sum_{j=1}^{m_i-1} j \mu(p_{i,m_i-j}) = \sum_{p \in S} \alpha_p \mu(p) - 1 \quad (12)$$

The notations are as follows. k and m_i are determined before the transition split: $k = |C \bullet|$, $m_i = W(C, t_i) \forall t_i \in C \bullet$. For the places $p_{i,j}$ resulted by splitting the transitions $t_i \in C \bullet$, we use the notations of section 4.1. Note that for t_i such that $m_i = 1$ there are no places $p_{i,j}$. In particular, if $\forall t_i \in C \bullet: m_i = 1$, the place

invariant is not changed

$$\mu(C) = \sum_{p \in S} \alpha_p \mu(p) - 1 \quad (13)$$

Assume that a control place C is added to \mathcal{N}_i . One of the last operations applied in iteration i is to transform the modified \mathcal{N}_i to a PT-ordinary Petri net, which will be denoted \mathcal{N}_{i+1} . The form of the place invariant of C in \mathcal{N}_{i+1} is (12) or (13), depending on whether C introduced or not input transition arcs with weight greater than one. By Proposition 6.7, the form of the place invariant will not be further changed in the following iterations, it stays the same in all $\mathcal{N}_{i+1}, \mathcal{N}_{i+2}, \dots$. Therefore no update is necessary in a iteration j for constraints added in previous iterations.

5.2.2 Constraints which do not need control place enforcement

There are siphons S such that if $\sum_{p \in S} \mu_0(p) \geq 1$ for the initial marking μ_0 , then $\sum_{p \in S} \mu(p) \geq 1$ for all reachable markings μ . Such a siphon does not need control. In order to reduce the complexity of the supervisor, such siphons are identified (see case 2 at page 18) and no control places are added in such situations. Therefore, instead of having a single set of constraints $L\mu \geq b$ we have two: $L\mu \geq b$ and $L_0\mu \geq b_0$. The constraints $L\mu \geq b$ define the supervisor. The constraints $L_0\mu \geq b_0$ are the constraints such that whenever the initial marking satisfies them, all reachable markings do. In consequence, the supervision for deadlock prevention of the target net requires enforcing $L\mu \geq b$ and choosing an initial marking μ_0 such that $L_0\mu_0 \geq b_0$ and $L\mu_0 \geq b$.

An example of siphon which does not require control is $\{C_1, C_2\}$ in figure 9. Example 5.3 illustrates how the constraints (L_0, b_0) are obtained.

5.2.3 Constructing the constraints of (L, b) and (L_0, b_0)

From equation (12) it can be seen that control places ensure inequalities of the form

$$\sum_{p \in S} \alpha_p \mu(p) \geq 1 \quad (14)$$

Section 5.2.2 showed that the inequalities which are enforced just by an appropriate initial marking (constraints which do not need a control place enforcement) have the same form. Assume that we consider an inequality as in (14) added in iteration number i . Note that S in (14) may contain control places added in the previous iterations $i-1, i-2, \dots, 1$. To reduce the number of variables, the constraints in (L, b) and (L_0, b_0) are not specified directly in the form of (14). Instead, by repeated substitutions, the inequality is written in the form $l^T \mu \geq c$, where the entries of l corresponding to control places are null. (We substitute a control place marking by its expression of the form (12).) This is how the inequalities $L\mu \geq b$ are obtained from those associated to control places, and how $L_0\mu \geq b_0$ are obtained from the inequalities which do not need control place enforcement.

The Petri nets $\mathcal{N}_1, \mathcal{N}_2, \dots$ have an increasing number of places. So the dimension of the marking vector μ is also increasing. The new places which are added in a iteration are control places and places resulted by applying transition splits. For each new place the matrices L and L_0 need a new column. Because the columns corresponding to control places are always null, we omit them in our examples.

Finally note that the purpose of the procedure is to provide constraints in terms of the marking of the target net \mathcal{N}_0 . The constraints of the net \mathcal{N}_k , where \mathcal{N}_k denotes the Petri net of the last iteration of the

procedure, are translated to constraints of \mathcal{N}_0 by removing all columns of L and L_0 which do not correspond to places of \mathcal{N}_0 .

5.2.4 Implicitly controlled siphons

Any marking μ which does not satisfy $L\mu \geq b$ and $L_0\mu \geq b_0$ is said to be a **forbidden marking**. A marking is **valid** if not forbidden and if all control place markings satisfy the proper invariant equations (12). Consider that the current iteration of the algorithm has the number i and that currently a new siphon S of \mathcal{N}_i is considered for control. It is desired that the siphon never becomes empty, that is $\sum_{p \in S} \mu(p) \geq 1$ is always true. We say that S is **(implicitly) controlled** if the latter inequality is satisfied for all markings μ which satisfy $L\mu \geq b$ and $L_0\mu \geq b_0$. For a controlled siphon a control place is not necessary and no new constraint in (L_0, b_0) needs to be added.

5.2.5 Initial constraint transformation

The constraints which are already enforced in the target net \mathcal{N}_0 (due to the structure of \mathcal{N}_0) are called **initial constraints**, because they are not produced by the deadlock prevention procedure and they exist when the procedure is started. This section considers the way initial constraints are transformed before the first iteration. As mentioned earlier in section 5.2.1, a constraint enforced in a iteration stays enforced for the following iterations, by Proposition 6.7. However this property is not always true for the initial constraints, since \mathcal{N}_0 may not be PT-ordinary (while all \mathcal{N}_i , $i \geq 1$, are so.)

To state the problem, assume that the marking constraints $L_0\mu \geq b_0$ are always true $\forall \mu \in \mathcal{R}(\mathcal{N}_0, \mu_0)$, $\forall \mu_0 \in \mathcal{M}_I$, where \mathcal{M}_I is some set of initial markings. Let \mathcal{N}_1 be the Petri net at the beginning of iteration one, that is \mathcal{N}_1 is \mathcal{N}_0 PT-transformed. Let $L'\mu \geq b'$ be a constraint which is satisfied for all markings of \mathcal{N}_1 which are reached from all valid initial markings in some set \mathcal{M}' . In view of Proposition 6.7, the constraint stays enforced in all other nets \mathcal{N}_i obtained in the following iterations, for all markings reachable from valid initial markings with restriction to the places of \mathcal{N}_1 in \mathcal{M}' . However, because \mathcal{N}_0 may not be PT-ordinary, it may not be true that $L_0\mu \geq b_0$ is enforced in \mathcal{N}_1 for all markings reachable from valid initial markings with restriction to the places of \mathcal{N}_0 in \mathcal{M}_I . Fortunately, the constraints $L_0\mu \geq b_0$ can be transformed in a form which is true in \mathcal{N}_1 . The idea of the transformation appears in Proposition 6.6. (Note that it is not technically correct to say that $L_0\mu \geq b_0$ is enforced in both \mathcal{N}_u and \mathcal{N}_v , for some $u \neq v$, since the markings in \mathcal{N}_u and \mathcal{N}_v have different dimensions; for the sake of simplicity, we mean that μ in $L_0\mu \geq b_0$ is the marking restricted to the places of the net in which $L_0\mu \geq b_0$ has been originally written.)

Assume that k transitions are split in \mathcal{N}_0 to obtain the PT-ordinary net \mathcal{N}_1 . Let t_1, t_2, \dots, t_k be the transitions of \mathcal{N}_0 which are split. Using the notations from the section 4.1, the transformed constraints $L'_0\mu \geq b'_0$,

which are true in \mathcal{N}_1 , are obtained from $L_0\mu \geq b_0$ by substituting $\mu(p)$ with $\mu(p) + \sum_{\substack{i=1 \\ m_i > 1}}^k \sum_{j=1}^{m_i-1} j\mu(p_{i,m_i-j})$

for all places p of \mathcal{N}_0 , where $m_i = 0$ if $p \notin \bullet t_i$ and $m_i = W(p, t_i)$ otherwise. We see, the substitution of $\mu(p)$ is simply $\mu(p)$ when no transitions in the postset of p are split. Also, when no transitions of \mathcal{N}_0 are split, we have \mathcal{N}_1 equal to \mathcal{N}_0 , and the constraint $L_0\mu \geq b_0$ remains unchanged.

5.2.6 Transforming Constraints to Admissible Constraints

In this section we consider the way in which the procedure uses the approach of section 4.3.2. We are to find the nonnegative integers α_p of the inequality

$$\sum_{p \in S} \alpha_p \mu(p) \geq 1 \quad (15)$$

such that the constraint is admissible and some other requirements, which we specify in this section, are satisfied. Once the parameters α_p are found, the constraint can be enforced with the invariant based approach of section 4.2.1, as the constraint is admissible. Let a be the vector with zero elements for places not in S and α_p for the places p ; then (15) can be written as $a^T \mu \geq 1$. Let d be a column vector defined as follows $d(i) = 1$ if p_i is in the active subnet and $d(i) = 0$ otherwise. It is required that:

$$a^T d > 0 \quad (16)$$

Thus, enforcing that S is controlled, guarantees that the restriction of S to the active subnet is a controlled siphon of the active subnet whenever (16) is true. Note that this is always the case when the siphon control approach of section 4.3.1 is used (that is, when no transformation to an admissible constraint is necessary.)

As shown in section 5.2.3, the marking of the control places μ_c can be expressed only in terms of the marking of the other places, μ_p , and so we have an equation: $\mu_c = M\mu_p - g$, where M is a matrix and g an integer vector. Let $a = [a_1^T, a_2^T]^T$, where a_1 and a_2 are the restrictions of a to the control places and respectively the other places of the net. Equation (15) can be written as

$$a^T [M^T, I]^T \mu_p \geq 1 + a_1^T g \quad (17)$$

Let D_s be the restriction of the current incidence matrix D to the columns of the new transitions resulted by split operations in all previous iterations. The additional constraint is

$$a^T D_s \leq 0 \quad (18)$$

The last requirement ensures that the control place C which results by enforcing (15) satisfies $C \notin t_{j,i} \bullet$ for all transitions $t_{j,i}$ resulted by splitting some transition t_j . This requirement is necessary for Proposition 6.2(b). Note that this proposition proves that the requirement is always satisfied in the case when the siphon control approach of section 4.3.1 is used (that is, when no transformation to an admissible constraint is necessary.)

If D_{uc} and D_{uo} are the restrictions of the incidence matrix of \mathcal{N}_0 to the uncontrollable and unobservable transitions, the admissibility requirements are (see section 4.2.2):

$$\begin{aligned} a^T N_r D_{uc} &\geq 0 \\ a^T N_r D_{uo} &= 0 \end{aligned} \quad (19)$$

where N_r is the restriction of $[M^T, I]^T$ to the columns which correspond to the places of \mathcal{N}_0 . Let a_n be the restriction of a to the places which resulted through transition split, let $P_n = \{p : a_n(p) \neq 0\}$ and $T_n = \bullet P_n$. As a transition split property, each place $p \in P_n$ has exactly one input transition, which is in T_n . Let D_{sn} be the restriction of D_s to the columns which correspond to T_n . Note that a_n does not affect (19). Then we can choose a_n such that:

$$a^T D_{sn} = 0 \quad (20)$$

The advantage of doing this would be that the control place C will result with less connections, and so less siphons in the next iteration. The method from the appendix can be easily adjusted to find a vector a with nonzero elements (the elements are the parameters α_p) given the constraints (16), (18), (19) and (20). The transformation fails when no solution is found such that at least two α_p , for p in \mathcal{N}^A , are nonzero.

5.3 The Computation of the Active Subnet

The active subnets of a Petri net is defined in Definition 3.6. The procedure considers in every iteration a single active subnet, preferably the maximal active subnet. The computation of the active subnet used by the procedure in an iteration is given below. The active subnet can be easily found once all transitions which cannot be made live under any circumstances are identified. Let D be the incidence matrix and i the index of such a transition which cannot be made live. Corollary 3.3 shows that for all vectors $x \geq 0$ such that $Dx \geq 0$: $x(i) = 0$. It also shows that if $x(j) > 0$, the transition of index j can be made live. Based on this idea, a polynomial complexity algorithm which computes the active subnet is given below. The usage of the input Z , which normally is the empty set, is discussed later in this section.

Input: The Petri net $\mathcal{N} = (P, T, F, W)$ and its incidence matrix D ; an optional set Z (default is $Z = \emptyset$) of transition indices which identify transitions which cannot be made live for reasons other than structural.

Output: The active subnet $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$.

1. Transform $Dx \geq 0$ in $[D, -I] \cdot [x^T, y^T]^T = 0$, where y are the excess variables. Let n be the number of rows of x and $M = \emptyset$.
2. **For** $i = 1, 2, \dots, n$ **and** $i \notin M$ **do**
 - (a) Check feasibility of $[D, -I] \cdot [x^T, y^T]^T = 0$ subjected to $x(i) = 1$, $x(j) = 0 \forall j \in Z$, $x \geq 0$ and $y \geq 0$, with a linear programming method. If feasible, let $[x_s^T, y_s^T]^T$ be a solution.
 - (b) Add all indices in $\|x_s\|$ to M .
3. The active subnet is $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$, where T^A is identified by the set of indices M , $P^A = T^A \bullet$, $F^A = F \cap \{(T^A \times P^A) \cup (P^A \times T^A)\}$ and W^A is the restriction of W to F^A .

In the cases when the deadlock prevention procedure detects that it is unable to control certain siphons, all transitions which belong to that siphons are marked to be removed from the active subnet. Considering all such transitions marked by the deadlock prevention procedure, let Z be the set of their indices in D . Then the active subnet is computed by using Z as input for the algorithm above. Using a nonempty set Z adds to the feasibility problem of step 2a the additional constraints that $x(j) = 0 \forall j \in Z$. Note that in this case the active subnet of the procedure is not the maximal active subnet (Definition 3.6).

Because of the iterative nature of the deadlock prevention procedure, the active subnet needs to be recomputed in every iteration. In general, the algorithm above needs to be used only once, to compute \mathcal{N}_1^A . Usually the other active subnets $\mathcal{N}_2^A, \mathcal{N}_3^A, \dots$ can be computed by simply repeating the changes done to \mathcal{N}_i in \mathcal{N}_i^A . (The procedure changes \mathcal{N}_i by adding control places and splitting transitions.) This simpler way of computing the active subnets is applied for all iterations which do not mark new transitions to be removed from the active subnet. This is a very common situation. For instance this is always true for all problems with no initial constraints on the target net and no uncontrollable and unobservable transitions.

Other details about the algorithm for the computation of the active subnet can be found in the appendix of [15].

5.4 The Deadlock Prevention Procedure

Input: The target Petri net \mathcal{N}_0 and a possibly empty set of initial constraints (L_0, b_0) .

Output: Two sets of constraints (L, b) and (L_0, b_0) and a variable indicating the mode of termination. (Deadlock is prevented for all initial markings μ_0 such that $L\mu_0 \geq b$, $L_0\mu_0 \geq b_0$ when (\mathcal{N}_0, μ_0) is supervised according to $L\mu \geq b$.)

Procedure:

- A. \mathcal{N}_0 is transformed to be PT-ordinary, as shown in section 4.1; the transformed net is \mathcal{N}_1 . The initial constraints (L_0, b_0) , if any, are transformed as shown in section 5.2.5. If not previously defined, let $X = \emptyset$.
- B. The (maximal) active subnets \mathcal{N}_0^A and \mathcal{N}_1^A of \mathcal{N}_0 and \mathcal{N}_1 are computed such that they do not contain the transitions in X . If \mathcal{N}_0^A is empty, the procedure terminates and declares failure. In particular, when $X = \emptyset$ and \mathcal{N}_0^A is empty, deadlock cannot be prevented in \mathcal{N}_0 under any circumstances.
- C. **For** $i \geq 1$ **do** (the initial Petri nets of the iteration i are \mathcal{N}_i^A and \mathcal{N}_i .)

- 1. If no new uncontrolled minimal active siphon is found, the next step is D. (The active siphons are taken with respect to the current active subnet \mathcal{N}_i^A . A siphon S is *uncontrolled* if $\sum_{p \in S} \mu(p) \geq 1$ is not implied by $L\mu \geq b$ and $L_0\mu \geq b_0$)
- 2. **For** every new uncontrolled minimal active siphon S **do**

Let C be the control place which would result by controlling the siphon, and let $l\mu \geq c$ be the inequality $\sum_{p \in S} \mu(p) \geq 1$ written in the form shown in section 5.2.3, that is without reference to the marking of the control places. First, the approach of section 4.3.1 is considered for the control of S through C .

- (a) If $C \bullet \subseteq \bullet S$, then S does not need supervision and C is not added to \mathcal{N}_i . The constraint (l, c) is added to (L_0, b_0) . The next step is 2c.
- (b) If $C \bullet \not\subseteq \bullet S$ then
 - i. If (l, c) is an inadmissible constraint (because of uncontrollable and/or unobservable transitions), C is added to the net as shown in section 4.3.2 and section 5.2.6; (l, c) is set to the obtained admissible constraint, expressed without reference to the marking of the control places (section 5.2.3).
 - ii. Else, if (l, c) is admissible, C is added according to the method of section 4.3.1.

In both cases (i) and (ii) (l, c) is included in (L, b) , except when the approach of the sections 4.3.2 and 5.2.6 fails to find an admissible constraint. When this failure occurs, all transitions of $S \bullet$ are marked as transitions which cannot be prevented (by the supervisor) to become dead. The active subnet, when updated in step 4, will not include these transitions.

- (c) It is checked that the system of $L_0\mu \geq b_0$ and $L\mu \geq b$ is feasible. (This is always the case when the procedure has no initial constraints (L_0, b_0) in step A.) If the system is feasible, the procedure continues with the next new uncontrolled minimal active siphon. Else, if the system is infeasible, all transitions of $S \bullet$ are marked as dead, in view of step 4. Also, C is removed from \mathcal{N}_i and (l, c) is removed from (L_0, b_0) or (L, b) .
- 3. If the Petri net is no longer PT-ordinary, the transitions which do not comply with this requirement are *split* (section 4.1.) The matrices L and L_0 are enhanced with null columns, each column corresponding to one new place resulted by transition split.

4. The active subnet is updated according to the changes made in the total net in the steps C-2b, C-2c and C-3. If the new subnet is empty, the procedure cannot generate a deadlock prevention supervisor and so it terminates. If an infeasible system was obtained at a previous step C-2c, then X is initialized to $X = T_0 \setminus T^A$ and the procedure is restarted at the step A, where T^A is the set of transitions of the active subnet.
 5. The final nets of the iteration i are denoted by \mathcal{N}_{i+1}^A and \mathcal{N}_{i+1} . The next step is C-1.
- D. The constraints (L, b) and (L_0, b_0) are modified to be written only in terms of the marking of the target net \mathcal{N}_0 . This is done by removing the columns of L and L_0 corresponding to places not in \mathcal{N}_0 (see section 5.2.)
- E. The constraints (L, b) and (L_0, b_0) are considered for simplifications, to remove redundant constraints.

5.5 Remarks

1. The purpose of the procedure is to produce two sets of linear constraints on the marking of the target net in the form $L\mu \geq b$ and $L_0\mu \geq b_0$, where L and L_0 are integer matrices and b and b_0 are integer column vectors. For all initial markings μ_0 , such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$, the target Petri net supervised according to $L\mu \geq b$ is deadlock free by Theorem 6.2.
2. The supervisor enforcing $L\mu \geq b$ is built using supervisory control based on place invariants ([24] and [37]). The procedure generates admissible constraints, so the supervisor results by the methodology of section 4.2.1.
3. The procedure is allowed to start with initial constraints in (L_0, b_0) . The user can employ initial constraints of (L_0, b_0) to tell the procedure that in his application all reachable markings μ satisfy $L_0\mu \geq b_0$. This allows specification of equations associated to place invariant properties of the net. The procedure does not make any assumptions on the initial marking, so an invariant equation needs to be specified if the user desires the procedure to use that invariant. (Specifying the constant of an invariant equation requires information on the initial marking.)
4. The difference between the constraints (L, b) and (L_0, b_0) which are generated by the procedure is that (L, b) need to be enforced by supervision, while (L_0, b_0) need not. (L_0, b_0) are guaranteed by the structure of the target Petri net when supervised according to $L\mu \geq b$, for all initial markings μ_0 of the target Petri net which satisfy $L_0\mu_0 \geq b_0$ in addition to $L\mu \geq b$.
5. Initial constraints in the form (L_0, b_0) are allowed. Without reducing the generality (see section 6.3.1), no initial constraints of the form (L, b) are allowed.
6. In an iteration we only use one of the possible active subnets. Whenever possible, the active subnet considered by an iteration is the maximal active subnet. The active subnet which is used in the iteration i is denoted by \mathcal{N}_i^A . Therefore, in iteration i , all active siphons are taken with respect to \mathcal{N}_i^A .
7. The new minimal active siphons of \mathcal{N}_{i+1} , $i \geq 1$, can be computed without computing all minimal active siphons. As shown in Proposition 6.8, each new minimal active siphon contains at least a control place added in iteration i to \mathcal{N}_i or a place from $P_i^A \setminus P_{i+1}^A$.

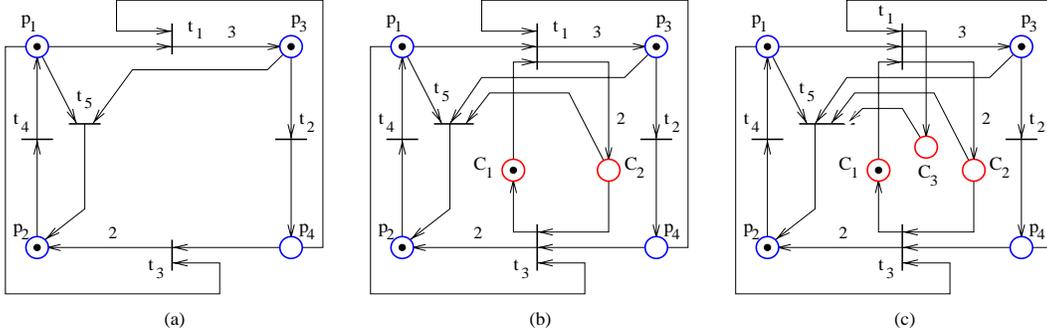


Figure 7: Example 5.1: (a) the target net, (b) after one iteration, (c) the final net. C_1 is a control place for the siphon $\{p_1, p_2\}$, C_2 for $\{p_3, p_4\}$ and C_3 for $\{C_1, C_2\}$.

8. Consider a special case: the target Petri net is repetitive, no initial constraints are given and no uncontrollable and unobservable transitions are present. Repetitive Petri nets have the property that markings exist such that liveness is enforcible via supervision. Therefore the maximal active subnet is equal to the total net and so any siphon is an active siphon. In consequence, the description of the procedure can be simplified by removing the steps 2c and 4. The procedure for this special case is similar to that of [20], where the most notable differences are the following. Our procedure does not assume the initial marking to be known. We use a different split transition operation. This makes sure that the PT-transformation cannot disable the control of siphons. In this way the failure condition mentioned in Lemma 5.1 in [15] does not appear now. We also check whether a siphon is uncontrolled. This helps termination, as shown in the example of figure 11 in section 6.4.2.
9. With regard to [15], the most significant improvements are the following. Initial constraints and uncontrollable and unobservable transitions are allowed. Redefining the active subnet allows improved permissivity in the case of nonrepetitive target nets and redefining the split transition operation allows less restrictive conditions for proving deadlock prevention.

5.6 Illustrative Examples

Example 5.1 Consider the Petri net of figure 7(a), which is repetitive. In a repetitive net any siphon is an active siphon with regard to the maximal active subnet. The original net has two minimal siphons $\{p_1, p_2\}$ and $\{p_3, p_4\}$. Therefore two control places are added, C_1 and C_2 , which enforce $\mu(P_1) + \mu(P_2) \geq 1$ and $\mu(P_3) + \mu(P_4) \geq 1$, respectively. Since C_1 and C_2 supervise the two siphons according to the invariant based approach (section 4.2 and 4.3), the following place invariants are created:

$$\mu(C_1) = \mu(P_1) + \mu(P_2) - 1 \quad (21)$$

$$\mu(C_2) = \mu(P_3) + \mu(P_4) - 1 \quad (22)$$

The current matrices L and b reflect the equations (21) and (22).

$$L = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

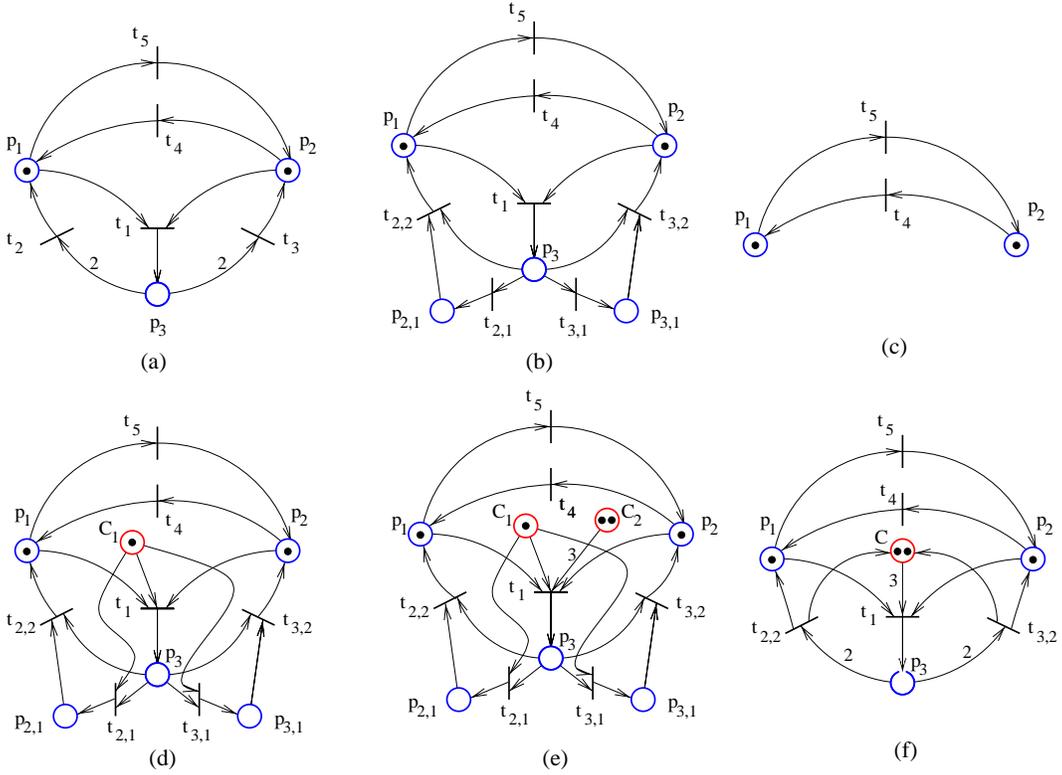


Figure 8: Example 5.2: (a) \mathcal{N}_0 ; (b) \mathcal{N}_1 ; (c) \mathcal{N}_1^A , the same as \mathcal{N}_2^A and \mathcal{N}_3^A ; (d) \mathcal{N}_2 ; (e) \mathcal{N}_3 before the split of t_1 ; (f) the final Petri net supervised for deadlock-freedom

At the second iteration the only new minimal siphon is $\{C_1, C_2\}$ and the inequality $\mu(C_1) + \mu(C_2) \geq 1$ is considered. As shown in section 5.2.3, all constraints can be written without reference to the control places. In particular, the constraint $\mu(C_1) + \mu(C_2) \geq 1$ can be written as $\mu(P_1) + \mu(P_2) + \mu(P_3) + \mu(P_4) \geq 3$ (see equations (21) and (22).) The method of section 4.3.1 is used for $\{C_1, C_2\}$, and a new control place C_3 results. The place invariant of C_3 is

$$\mu(C_3) = \mu(P_1) + \mu(P_2) + \mu(P_3) + \mu(P_4) - 3 \quad (23)$$

The resulting net (figure 7(c)) has no new minimal siphons, therefore the procedure terminates. The matrices L and b after the second iteration are:

$$L = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}$$

Because L and b cannot be simplified, the supervised net for deadlock prevention is the same as that of figure 7(c). By Theorem 6.2, the supervised Petri net is deadlock-free for all initial markings μ_0 such that $L\mu_0 \geq b$. In this example no constraints (L_0, b_0) are produced. \square

Example 5.2 Consider the Petri net of figure 8(a), which is not PT-ordinary. Three transitions cannot be made live, for any finite marking: t_1 , t_2 and t_3 .

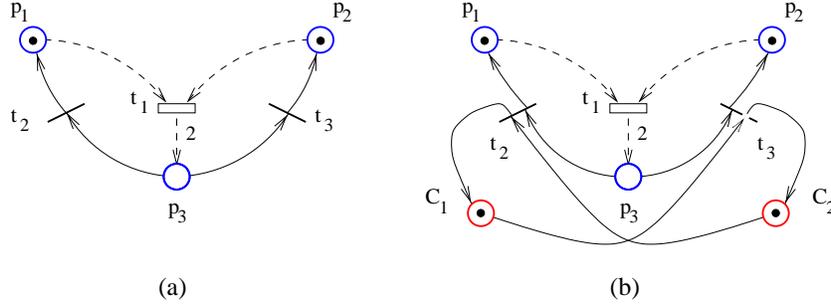


Figure 9: Example 5.3: (a) \mathcal{N}_0 ; (b) the final Petri net supervised for deadlock-freedom

The first iteration begins with the PT-transformed net \mathcal{N}_1 . There is a single minimal active siphon, $\{p_1, p_2, p_3\}$. A control place C_1 is added to the total net (figure 8(d)). The active subnets in the iterations 1, 2 and 3 are shown in figure 8(c). The inequality associated with C_1 is $\mu(p_1) + \mu(p_2) + \mu(p_3) \geq 1$, so at the end of this iteration $L = [1, 1, 1, 0, 0]$ and $b = 1$.

In the second iteration there is a single new minimal active siphon, $\{p_1, p_2, p_{2,1}, p_{3,1}, C_1\}$. The siphon is uncontrolled, since $\mu(p_1) + \mu(p_2) + \mu(p_{2,1}) + \mu(p_{3,1}) + \mu(C_1) \geq 1$, that is $2\mu(p_1) + 2\mu(p_2) + \mu(p_3) + \mu(p_{2,1}) + \mu(p_{3,1}) \geq 2$, is not implied by $\mu(p_1) + \mu(p_2) + \mu(p_3) \geq 1$. The control place C_2 which is added is also a source place. The procedure terminates, since at the third iteration there is no new minimal active siphon. The resulting matrices L and b after step D are:

$$L = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

There is one redundant constraint, so the final constraints are $L = [2, 2, 1]$ and $b = 2$. The supervised net is shown in figure 8(f). By Theorem 6.2 it is deadlock-free for all initial markings μ_0 such that $L\mu_0 \geq b$. \square

Example 5.3 Consider the repetitive Petri net of figure 9(a), where t_1 is unobservable. In the first iteration there are two minimal siphons: $\{p_1, p_3\}$ and $\{p_2, p_3\}$. Consider the siphon $\{p_1, p_3\}$. The marking constraint $\mu(p_1) + \mu(p_3) \geq 1$ is not admissible, so the approach of section 4.3.2 is used for the control. The resulting admissible constraint is $2\mu(p_1) + \mu(p_3) \geq 1$. The control place C_1 is added according to this constraint, and the place invariant $\mu(C_1) = 2\mu(p_1) + \mu(p_3) - 1$ results. Similarly C_2 enforces $2\mu(p_2) + \mu(p_3) \geq 1$ on $\{p_2, p_3\}$ and $\mu(C_2) = 2\mu(p_2) + \mu(p_3) - 1$. The matrices L and b after the first iteration are:

$$L = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

In the second iteration there is a single new minimal siphon, $\{C_1, C_2\}$. The control place which would result by enforcing $\mu(C_1) + \mu(C_2) \geq 1$ is C_3 such that $C_3 \bullet = \emptyset$. Therefore, $\{C_1, C_2\}$ does not need control, according to the step 2a of the procedure. $\mu(C_1) + \mu(C_2) \geq 1$ is written as $2\mu(p_1) + 2\mu(p_2) + 2\mu(p_3) \geq 3$, and so

$$L_0 = \begin{bmatrix} 2 & 2 & 2 \end{bmatrix} \quad b_0 = \begin{bmatrix} 3 \end{bmatrix}$$

The procedure terminates, since there is no new uncontrolled siphon in the third iteration. The supervised net is shown in figure 9(b). Deadlock is prevented for all initial markings such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$. In fact, liveness is enforced and so the supervisor is maximally permissive by Theorem 6.3. \square

6 Properties

6.1 Basic Properties of the Method

6.1.1 Introduction and Notations

In the deadlock prevention procedure, we start with a Petri net $\mathcal{N}_0 = (P_0, T_0, F_0, W_0)$ that may not be PT-ordinary. New Petri nets $\mathcal{N}_i = (P_i, T_i, F_i, W_i)$, $i \geq 1$, are derived in an iterative process. The only operations of an iteration that modify the structure of the net are the addition of a new control place (section 4.3) and transition split (section 4.1).

Adding control places does not modify the set of transitions. The set of places is increased by the set of new control places, and the set of transition arcs by the new arcs which connect the control places to already existing transitions. The old arcs have unmodified weights; new arcs connecting the new control places may have weights greater than one. If a weight of an arc entering a transition is greater than one, the Petri net is not PT-ordinary and transitions not conforming to the requirement may be split.

The Petri net notations are: $\mathcal{N}_0 = (P_0, T_0, F_0, W_0)$ – the initial Petri net, $\mathcal{N}_1 = (P_1, T_1, F_1, W_1)$ – \mathcal{N}_0 PT-transformed, $\mathcal{N}_i = (P_i, T_i, F_i, W_i)$ – the Petri net produced by iteration $i - 1$ for $i \geq 2$ and $\mathcal{N}_i^A = (P_i^A, T_i^A, F_i^A, W_i^A)$ – the active subnet of \mathcal{N}_i . (In an iteration we only use one of the possible active subnets; the active subnet, denoted by \mathcal{N}_i^A , is required in order to specify the active siphons.)

When a transition is split, one or more of its input arcs are replaced by a sequence of places and transitions. Unlike the split operations of [20] and [15], in this paper a split transition is not removed from the Petri net (see section 4.1.) Firing a split transition in the initial net is equivalent to firing it together with the sequence of replacing transitions in the transformed net. Let T_R be the set of transitions which are created by transition split. Also, let P_R be the set of places generated by transition split. Then for every \mathcal{N}_i the set of places is $P_i = P_0 \cup P_R \cup \mathcal{C}$ and the set of transitions is $T_i = T_0 \cup T_R$, where \mathcal{C} is the set of control places which were added in the iterations $1, 2, \dots, i$.

Let $\sigma_i(t)$ denote the transition sequence corresponding to the new transitions which appear when t is split in \mathcal{N}_i . That is, if t is first split in iteration $k - 1$, $\sigma_i(t) = \emptyset$ for $i < k$ and $\sigma_i(t) = \sigma_{k+1}(t) = t_k t_{k-1} \dots t_1$ for $i \geq k$, where $t_1 \dots t_k$ are the new transitions which result when t is split. We may omit the index i , meaning $\sigma_i(t)$ for $\sigma(t)$ if the current referred net is \mathcal{N}_i .

We also need notations to specify the transition sequences of \mathcal{N}_i which resulted by successive splits of a single transition of \mathcal{N}_0 . For instance, a transition t of \mathcal{N}_1 is split in the first iteration and so t_1, t_2, t_3 , result, then in the second iteration t_1 and t_2 are split and the new transitions are $t_{1,1}, t_{1,2}$, and $t_{2,1}, t_{2,2}$. We know that by firing the sequence $t_3 t_2 t_1 t$ in \mathcal{N}_2 we have the same effect as by firing t in \mathcal{N}_1 . Also, firing $t_3 t_{2,2} t_{2,1} t_2 t_{1,2} t_{1,1} t_1 t$ in \mathcal{N}_3 corresponds to firing t in \mathcal{N}_1 , but the same may be true for other ordering in a sequence of these transitions, for instance for $t_3 t_{2,2} t_{1,2} t_{2,1} t_2 t_{1,1} t_1 t$.

The problem is to specify the transition sequences $\sigma_{0,i}(t)$ of \mathcal{N}_i such that (a) $\sigma_{0,i}(t)$ enumerates the transitions (including t itself) in which t of \mathcal{N}_0 is successively split until (and including) the iteration $i - 1$, and (b) markings μ of \mathcal{N}_i exist such that $\mu(p) \neq 0$ only if $p \notin P_R$, that is p a place of \mathcal{N}_0 or a control place, and μ enables $\sigma_{0,i}(t)$. Firing the sequence $\sigma_{0,i}(t)$ in \mathcal{N}_i corresponds to firing t in \mathcal{N}_0 . If t is not split, we let $\sigma_{0,i}(t) = t$. The notation $\sigma_{j,i}(t)$ for $i > j$ and t in \mathcal{N}_i , will be similarly defined, by taking \mathcal{N}_i instead of \mathcal{N}_0 .

Because of the nature of the split operation, we need to specify sets of transition sequences, and this is done by listing sequentially sequences and groups of sequences, where in each group the sequences can fire

asynchronously. A group is included between braces. For instance, given the transitions t_1, t_2, t_3 and t_4 ,

$$\{t_1, t_2 t_3\}t_4$$

defines the sequences $t_1 t_2 t_3 t_4$, $t_2 t_1 t_3 t_4$ and $t_2 t_3 t_1 t_4$; the notation denotes that t_1 and the sequence $t_2 t_3$ can fire asynchronously, but t_4 can fire only after all of t_1 and $t_2 t_3$ have fired. We redefine P_R to be the set of places which appeared by transition splits in the iterations $m, m + 1, \dots, k - 1$. The following algorithm defines the set of transition sequences $\Sigma_{m,k}$ which resulted by successive splits in the iterations $m, m + 1, \dots, k - 1$ of a single transition t of \mathcal{N}_m .

Input: t : a transition of $\mathcal{N}_m, \mathcal{N}_m$ and P_R .

Output: $\Sigma_{m,k}$

$n = 1, I = \{1\}, \Sigma_1 = t$

While $I \neq \emptyset$ **do**

For all $i \in I$ **do**

 Let t_x be the first transition in Σ_i .

If $|\bullet t_x \cap P_R| = 0$ **then**

$I := I \setminus \{i\}$

else if $|\bullet t_x \cap P_R| = 1$ **then**

 Let t_y be the transition such that $\{t_y\} = \bullet(\bullet t_x \cap P_R)$.

$\Sigma_i := t_y \Sigma_i$

else if $|\bullet t_x \cap P_R| = j > 1$ **then**

 Let t_{y1}, \dots, t_{yj} be the transitions such that $\{t_{y1}, \dots, t_{yj}\} = \bullet(\bullet t_x \cap P_R)$.

$I := (I \cup \{n + 1, \dots, n + j\}) \setminus \{i\}$

$\Sigma_{n+1} := t_{y1}, \dots, \Sigma_{n+j} := t_{yj}$

$\Sigma_i := \{\Sigma_{n+1}, \dots, \Sigma_{n+j}\} \Sigma_i$

$n := n + j$

end if

end for

end while

$\Sigma_{m,k} = \Sigma_1$

We will denote by $\sigma_{\mathbf{m},\mathbf{k}}(\mathbf{t})$ an arbitrary transition sequence of $\Sigma_{m,k}$. In particular, $\sigma_{\mathbf{0},\mathbf{k}}(\mathbf{t})$ considers split transitions with respect to the original Petri net \mathcal{N}_0 instead of \mathcal{N}_m . Note that unlike the sequence $\sigma(t)$, a sequence $\sigma_{m,k}(t)$ is defined to contain t , and it ends with t .

The postset and the preset operations may generate confusion when we consider more Petri nets \mathcal{N}_i at the same time, as they share common transitions and places. Therefore sometimes we need to use the following notations:

1. $x \bullet_i$ is $x \bullet$ evaluated in \mathcal{N}_i , where $x \in P_i \cup T_i$.
2. $\bullet_i x$ is $\bullet x$ evaluated in \mathcal{N}_i , where $x \in P_i \cup T_i$.

6.1.2 Properties

This section introduces a number of properties which are useful in the proofs of the main results in section 6.2 and for a better understanding of the procedure.

Proposition 6.1 Let \mathcal{N}_k^A and \mathcal{N}_k be the the active subnet and the total net after iteration number $k - 1$.

- (a) $P_k \subseteq P_{k+1}$ and $T_k \subseteq T_{k+1}$ for all $k \geq 0$.
- (b) Any $p \in P_k \setminus P_k^A$ has in \mathcal{N}_k the property that $\bullet p \subseteq T_k \setminus T_k^A$.
- (c) Consider the step 2 of an iteration and let C be a control place added to the total net with regard to a minimal active siphon that contains the siphon S of the active subnet. Then S is controlled by C in the active subnet.

Proof: (a) By construction, control places are added to the total net and new places may be created by transition split. In this way $P_{k+1} = P_k \cup \mathcal{C}_k \cup P_{S,k}$, where \mathcal{C}_k is the set of control places added in iteration k and $P_{S,k}$ is the set of places resulted from transition split in iteration k . Also, by construction, when a transition t is split, it is not removed (section 4.1), but new places and transitions are added; so $T_{k+1} = T_k \cup T_s$, where T_s is the set of transitions resulted through transition split in the iteration k .

(b) Immediate consequence of the construction of the active subnet.

(c) The incidence matrix of the *active subnet* can be obtained from the *total subnet* by removing the columns and rows corresponding to transitions and places which are not in the *active subnet*. Also, the constraint matrix l_a in the *active subnet* is the restriction to the places of the *active subnet* of the constraint l . Therefore by enforcing the constraint of l in the *total net*, and then by removing the transitions which do not belong to the *active subnet*, the same connections for the control place C are obtained as in the case when l_a is enforced directly in the *active subnet* (see section 4.2). Because enforcing l_a ensures that S is controlled (section 4.3.2), the conclusion follows. \square

Several properties related to transition splitting are given in the next two propositions.

Proposition 6.2 Let \mathcal{C} be the set of control places added up to the iteration m . Then: (a) $\bullet P_0 \cap (T_m \setminus T_0) = \emptyset$, (b) $\bullet \mathcal{C} \cap (T_m \setminus T_0) = \emptyset$ and (c) $\forall t \in (T_m \setminus T_0): |t \bullet| = 1$.

Proof: (a) The property is obvious just by inspecting the transition split operation: for $m = 1$ the property is true, and for $m > 1$ it also is true since (i) transitions from a split operations are only in the preset of the new places resulted through the split and (ii) transition splits for $m > 1$ are only due to adding new control places, so the transitions connected to P_0 remain the same throughout all iterations.

(b) and (c). Note that (c) is a consequence of (b): the only way a transition can get a new place in its postset is by adding control places. Then if (b) is true, all transitions in $T_m \setminus T_0$ keep their original postset, and since the transitions t from split replacements are originally produced with $|t \bullet| = 1$ (section 4.1), (c) is verified.

The siphon control method for uncontrollable and unobservable transitions (section 4.3.2) is constructed such that property (b) is true for all controls places which are added using it. However it remains to be proved that the property is true when the more usual siphon control method (section 4.3.1) is used.

The proof is by induction. Assume that the property is true for all control places added so far, and let k be the current iteration number. Then for all transitions $t \in (T_k \setminus T_0): |t \bullet| = 1$. We assume by contradiction that adding the control place C with regard to a siphon S connects C to t such that $C \in t \bullet$ and $t \in (T_k \setminus T_0)$. This implies that t increases the marking of S when it is fired; however, before adding C , $|t \bullet| = 1$, so t cannot increase the marking of S unless $t \in S \bullet$ and $t \notin \bullet S$. But this contradicts that S is a siphon. \square

Proposition 6.3 For every iteration index i :

- (a) If $P_i^A \cap P_0 = \emptyset$ then \mathcal{N}_i^A is empty.
- (b) Let $t \in T_0$. If $t_x \in \sigma_{0,i}(t)$ and $t_x \in T_i^A$ then every transition of $\sigma_{0,i}(t)$ preceding t_x is in T_i^A , where a transition t_y of $\sigma_{0,i}(t)$ precedes t_x if $\exists t_1 \dots t_n \in \sigma_{0,i}(t)$ such that $t_x \in t_n \bullet \bullet, \dots, t_1 \in t_y \bullet \bullet$.
- (c) Let \mathcal{C} be the set of control places of \mathcal{N}_i , that is all the control places which were added in iterations $1, 2, \dots, i-1$. There is no siphon S of the total net or of the active subnet such that $S \subseteq P_i \setminus (P_0 \cup \mathcal{C})$.

Proof: (a) $P_i^A \cap P_0 = \emptyset \Rightarrow \bullet P_0 \cap T_i^A = \emptyset$, so $T_0 \cap T_i^A = \emptyset$. Recall, the transitions which are not in the active subnet cannot fire infinitely often. Note that $T_i \setminus T_0$ are transitions resulted from transition split. However, by split transition construction, there is no cycle in which only transitions from $T_i \setminus T_0$ appear and none of the transitions from $T_i \setminus T_0$ can be a source transition. Therefore the transitions in $T_i \setminus T_0$ cannot fire infinitely often. Hence, T_i^A is not a subset of $T_i \setminus T_0$, so $T_i^A = \emptyset$.

(b) A transition belongs to the active subnet if finite markings exist such that it can fire infinitely often. To prove the conclusion, it is enough to prove that $t_u \in \sigma_{0,i}(t)$ and $t_u \in \bullet \bullet t_x$ imply that t_u is in the active subnet. This can be shown as follows: $\exists p \in P_S$ (where P_S is the set of places resulted from transition split operations) such that $t_u \in \bullet p$ and $t_x \in p \bullet$. Since $|\bullet p| = 1$ (see the transition split operation) t_u must be able to fire infinitely often.

(c) Let P_S be the set of places resulted from transition split: $P_S = P_i \setminus (P_0 \cup \mathcal{C})$. The proof is a direct consequence of the splitting method (section 4.1). Thus, $p \in P_S$ cannot be a source place in the total net, while the active subnet cannot anyway have source places. Further on, if P_{S_x} is the set of places from the replacement of $t_x \in T_0$ in \mathcal{N}_i , there are no cyclic structures only made up of places in P_{S_x} . Also, because $(\bullet \bullet P_{S_x} \setminus P_{S_x}) \cap P_S = \emptyset$ and $(P_{S_x} \bullet \bullet \setminus P_{S_x}) \cap P_S = \emptyset$ there is no cyclic structure only made up of places in P_{S_x} and other places from P_S . The same justification also applies to the active subnet. \square

It is interesting to find out what happens to a siphon controlled with a control place when one or more of its transitions are split. A transition t may be split after a control place C is added in the preset of t and $W(C, t) > 1$. The following proposition shows that the siphons are not changed by transition split operations.

Proposition 6.4 *Given a PT-ordinary Petri net, let S be (i) a (minimal) siphon, or (ii) a (minimal) active siphon. Assume that after adding some control places the net is no longer PT-ordinary. If some arbitrary transition t is split, then S remains a (minimal) siphon in case (i), or a (minimal) active siphon in case (ii).*

Proof: Transition split operations do not change $S \bullet$ and $\bullet S$, so S is still a siphon. In general, for all places p but the new added control places $p \bullet$ and $\bullet p$ are not changed. So if S was a minimal siphon, it remains minimal. If t was in the active subnet, it remains so. Indeed, the split replacement sequence produces the same marking change to the original places of the net as firing t alone in the original net. If t was not in the active subnet, no siphon of the active subnet is affected by splitting t . If t was in the active subnet, all its replacing sequence is in it (Proposition 6.3(b)) and so we can apply the same reasoning to see that if s was a minimal siphon of the active subnet, it remains so. Hence if S was an active siphon, it remains so. If S is now not minimal, it includes a smaller active siphon S' which includes the siphon s' of the active subnet. Using the same reasoning, S' was a siphon in the original net and s' a siphon in the original active subnet. So S was not minimal if S is not minimal in the new net. In other words S was a minimal active siphon only if S is so in the new net. \square

Proposition 6.5 *Let C be the control place which enforces $l^T \mu \geq b$ in \mathcal{N}_i . Let P_R be the set of places resulted through transition split in iterations i through $j - 1$ and μ_0 be a marking of \mathcal{N}_j such that $\mu_0(p) = 0 \forall p \in P_R$ and $\mu_0(C) = l^T \mu_{0r} - b$. For all markings μ reachable from μ_0 and such that $\mu(p) = 0 \forall p \in P_R$, $\mu(C) = l^T \mu_r - b$ is satisfied. The notations μ_{0r} and μ_r denote the markings μ_0 and μ , respectively, restricted to the places of \mathcal{N}_i .*

Proof: This is a direct consequence of the following facts: (a) C enforces $l^T \mu \geq b$ in \mathcal{N}_i ; (b) Let $t \in T_i$, which is found split in \mathcal{N}_j . Firing the entire split replacement sequence of t in \mathcal{N}_j , modifies the marking of the places of P_i in the same way as firing the transition t in \mathcal{N}_i (see section 4.1). \square

Proposition 6.6 *Assume that a number of constraints are enforced on a PT-ordinary Petri net. Let C be the control place added to enforce the constraint $l^T \mu \geq b$, that is $\mu(C) = l^T \mu - b$. Let t_1, t_2, \dots, t_k be all transitions such that $W(C, t_i) = m_i > 1$. Next, the closed loop Petri net is transformed in a PT-ordinary Petri net as shown in section 4.1. Then C enforces:*

$$\mu_e(C) + \sum_{i=1}^k \sum_{j=1}^{m_i-1} j \mu_e(p_{i, m_i-j}) = l^T \mu - b \quad (24)$$

in the PT-transformed Petri net, where μ is the marking vector μ_e restricted to the places of the original Petri net and the usual notations of section 4.1 are used.

Proof: Before splitting the transitions, the invariant based method for enforcing constraints guaranteed that the change in marking, by firing any transition, is the same for $\mu(C)$ and $l^T \mu - b$. This is no longer true for t_1, \dots, t_k after they have been split, because after this C is not connected to them in the same way as before (now $W(C, t_i) = 1$ for all $i = 1 \dots k$). However, the other connections between C and the rest of transitions which affect $l^T \mu - b$ remain the same. Hence, among the transitions which change by firing $l^T \mu - b$, only t_1, \dots, t_k produce a different marking change in C and $l^T \mu - b$. Besides, the transitions $t_{i,j}$ ($i = 1 \dots k$ and $j = 1 \dots m_i - 1$) are the only which change the marking of C when fired, but do not affect the marking of $l^T \mu - b$ (see the split transition construction in 4.1).

Next we consider firing the transitions t_i and $t_{i,j}$, for $i = 1 \dots k$ and $j = 1 \dots m_i - 1$: the right hand side of (24) is affected only when one of t_i fires. Assume that equation (24) currently holds true. When $t_{i,j}$ fires, $j < m_i - 1$, C loses 1 token, $p_{i,j+1}$ loses 1 token and $p_{i,j}$ gets 1 token. The left side of (24) is thus changed by $-1 + (m_i - j - 1)(-1) + m_i - j = 0$, so the equality still holds. For $j = m_i - 1$, C loses one token and p_{i, m_i-1} gets one token, so the left side of (24) is not changed and the equality remains true. When t_i fires, C loses 1 token, $p_{i,1}$ also loses 1, and so the marking change of the left side is $-1 + (m_i - 1)(-1) = -m_i$. However the marking change of the right side is $-m_i$ too, because originally $W(C, t_i) = m_i$. Hence the equality also remains true when one of t_i fires. Since all possibilities have been exhausted, the conclusion is established. \square

Proposition 6.6 shows that a constraint enforced in \mathcal{N}_i stays enforced in all \mathcal{N}_j , $j > i$. Indeed, note that equation (24) implies $l^T \mu \geq b$, which is what was desired.

Proposition 6.7 *Let $l^T \mu \geq b$ be a marking inequality satisfied for all markings reachable from a set of markings \mathcal{M} of \mathcal{N}_i , $i \geq 1$. Let P_R be the set of places resulted through transition split in iterations i through $j - 1$ and μ_0 be a marking of \mathcal{N}_j such that $\mu_0(p) = 0 \forall p \in P_R$ and $l^T \mu_{0r} \geq b$. Then for all markings μ*

reachable from μ_0 , $l^T \mu_r \geq b$ is satisfied. The notations μ_{0r} and μ_r denote the markings μ_0 and μ , respectively, restricted to the places of \mathcal{N}_i .

Proof: Note that \mathcal{N}_i can be regarded as a subnet of \mathcal{N}_j : all places and transitions of \mathcal{N}_i appear in \mathcal{N}_j , with the same connections between them. The new places (the places of \mathcal{N}_j which do not appear in \mathcal{N}_i) are connected through new transitions or by new transition arcs added to transitions of \mathcal{N}_i . So the new places only restrict the possible transition firings of the \mathcal{N}_i part of \mathcal{N}_j . Therefore the reachable set of markings of the \mathcal{N}_i part of (\mathcal{N}_j, μ_0) is a subset of $\mathcal{R}(\mathcal{N}_i, \mu_i)$, which only contains markings for which S is not empty. \square

In particular, Proposition 6.7 proves that if $S \subseteq P_i$ and $\sum_{p \in S} \mu(p) \geq 1$ is ensured for all markings reachable from a set of markings \mathcal{M} of \mathcal{N}_i , then S is a controlled siphon in \mathcal{N}_j , $j > i$, for all markings μ reachable from markings μ_0 such that $\mu_0(p) = 0 \forall p \in P_R$: $\mu_0(p) = \mu_i(p) \forall p \in P_j \cap P_i$ and $\mu_i \in \mathcal{M}$.

The next proposition is significant for the efficiency of the implementation of the step C:2 of the procedure. It shows that since in each iteration i we look for new minimal active siphons, it is enough to seek only the minimal active siphons which contain the new control places added in the previous iteration and the places of $P_i^A \setminus P_{i+1}^A$. Note that $P_i^A \setminus P_{i+1}^A \neq \emptyset$ may occur due to the steps C:2b and C:2c of the procedure, when the target Petri net has uncontrollable and unobservable transitions or when (tight) initial constraints are given. The next result is useful, as the implementation of the procedure does not need to compute all minimal active siphons (which could be computationally expensive.)

Proposition 6.8 *The new minimal active siphons of \mathcal{N}_{i+1} , $i \geq 1$, contain at least one of the control places added in the iteration number i or one of the places of $P_i^A \setminus P_{i+1}^A$.*

Proof: Consider the iteration $i \geq 1$, which starts with the initial Petri net \mathcal{N}_i and generates \mathcal{N}_{i+1} . Assume that S is a siphon of \mathcal{N}_{i+1} which is a counterexample to the claim of the proposition. S could be new only if (a) S is not a siphon of \mathcal{N}_i , or if (b) S was a siphon of \mathcal{N}_i but not an active siphon, or if (c) S is not a minimal active siphon of \mathcal{N}_i .

In case (a), $T_x = \bullet S \setminus S \bullet \neq \emptyset$ in \mathcal{N}_i , but $\bullet S \setminus S \bullet = \emptyset$ in \mathcal{N}_{i+1} . But this is not possible, as $S \subseteq P_i$ and the procedure adds no new transitions in the postset or preset of $p \in P_i$ for $i \geq 1$. Indeed, \mathcal{N}_i is PT-ordinary, so only the control places added in iteration i could be connected to new transitions (resulted through the split operation.)

Case (b) isn't either possible. If S is an active siphon of \mathcal{N}_{i+1} , then $S \cap P_{i+1}^A \neq \emptyset$. By Proposition 3.4, $S \cap P_i^A = \emptyset$. Since $S \subseteq P_i$, it follows that $P_i \cap (P_{i+1}^A \setminus P_i^A) \neq \emptyset$. By definition, $P_i^A = T_i^A \bullet_i$ and $P_{i+1}^A = T_{i+1}^A \bullet_{i+1}$. Also, $T_{i+1}^A \setminus T_i^A = T_R$, where all transitions of T_R are transitions which result through the transition split operation in the iteration i . Since in the iteration $i \geq 1$ only weights of arcs in the postset of the new control places may be greater than one: $T_R \bullet \cap P_i = \emptyset$, and from $P_i \cap (P_{i+1}^A \setminus P_i^A) \neq \emptyset$ we get $P_i \cap (T_i^A \bullet_{i+1} \setminus T_i^A \bullet_i) \neq \emptyset$, which is impossible.

In case (c), let $S' \subset S$ be a minimal active siphon of \mathcal{N}_i . S' is still a siphon of \mathcal{N}_{i+1} . If S' is active in \mathcal{N}_{i+1} , then it is a minimal active siphon of \mathcal{N}_{i+1} . Indeed, assume the contrary. Then there is $S'' \subset S'$ which is minimal. Then, by the same proof as in (a), S'' is a siphon of \mathcal{N}_i and by the same proof as in (b) S'' is an active siphon of \mathcal{N}_i . But $S'' \subset S'$, so S' is not minimal in \mathcal{N}_i , which is contradiction. So S' is a minimal active siphon of \mathcal{N}_{i+1} . But this contradicts $S' \subset S$ and S minimal and active. Therefore S' cannot be an active siphon of \mathcal{N}_{i+1} . This implies that $S' \cap (P_i^A \setminus P_{i+1}^A) \neq \emptyset$ (and so $S \cap (P_i^A \setminus P_{i+1}^A) \neq \emptyset$), which contradicts the fact that S is a counterexample to the claim of the proposition. \square

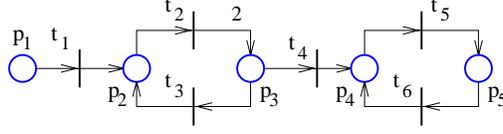


Figure 10: Example for Proposition 6.8

An example is given in figure 10. Consider that the Petri net from the figure is \mathcal{N}_0 and that t_2 and t_4 are uncontrollable. As \mathcal{N}_0 is PT-ordinary, $\mathcal{N}_1 = \mathcal{N}_0$. Note that $P_1^A = \{p_2, p_3, p_4, p_5\}$. $S = \{p_1, p_2, p_3, p_4, p_5\}$ is an active siphon, but it is not minimal. However $S' = \{p_1, p_2, p_3\}$ is minimal and active. Because of the uncontrollable transitions, the control of the siphon S' fails. Therefore $\mathcal{N}_2 = \mathcal{N}_1$, but P_2^A is reduced to $\{p_4, p_5\}$. S is a minimal active siphon in \mathcal{N}_2 , while S' is no longer active. Note that S , which is a new minimal siphon of \mathcal{N}_2 , contains the places p_2 and p_3 , which are in $P_1^A \setminus P_2^A$.

In the next definition we will denote by *valid markings* those markings in which the invariant relations associated with every control place hold and in which places obtained by transition split have the marking 0. Also we define equivalence of markings, which is an *equivalence relation* on the Petri nets $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \dots$ generated in each iteration. A class of equivalence contains the valid markings of the nets \mathcal{N}_k which have the same marking for the places $p \in P_0$.

Definition 6.1 Let $\mathcal{N}_i, (L_i, b_i)$ and (L_{i0}, b_{i0}) be the Petri net and respectively the sets of constraints, all at the beginning of iteration $i \geq 1$, or for the initial Petri net, in which case $i = 0$. Let \mathcal{C} be the set of control places that were added beginning with iteration one and $P_R = P_i \setminus (P_0 \cup \mathcal{C})$. A marking μ of \mathcal{N}_i is said to be a **valid marking** if $\mu(p) = 0 \forall p \in P_R$, $L_i \mu_e \geq b_i$ and $L_{i0} \mu_e \geq b_{i0}$, where μ_e is a marking of \mathcal{N}_0 such that $\mu_e(p) = \mu(p) \forall p \in P_0$, and the marking of every control place C of an active siphon S satisfies the equation (12) that C is to enforce.

The definition above applies also for \mathcal{N}_1 , where in case that no initial constraints exist, the remaining requirement for μ to be a valid marking of \mathcal{N}_1 is $\mu(p) = 0 \forall p \in P_R$. When we refer to a marking μ of \mathcal{N}_0 , μ is always valid when the procedure starts with no constraints in (L_0, b_0) . Otherwise, μ is valid if it satisfies the constraints stated at the beginning of the procedure.

A Petri net \mathcal{N}_i may not be *well-marked* for a marking that is valid. Indeed, the definition of valid markings does not require the *new* siphons of \mathcal{N}_i not to be empty. Previous siphons cannot be empty for a valid marking, because of the constraints $L_i \mu_e \geq b_i$ and $L_{i0} \mu_e \geq b_{i0}$ which encode this requirement for previous siphons.

Definition 6.2 Let μ_e be a valid marking of \mathcal{N}_0 and μ a valid marking of \mathcal{N}_i . If $\mu_e(p) = \mu(p) \forall p \in P_0$, then μ_e and μ are said to be **equivalent markings**. Moreover, two valid markings μ_i of \mathcal{N}_i and μ_j of \mathcal{N}_j also are called **equivalent markings** if they have the same equivalent marking in \mathcal{N}_0 .

The way in which equivalence is defined implies that if two markings are equivalent they must also be valid. Equivalence is not defined for markings that are not valid.

Proposition 6.9 Any valid marking of \mathcal{N}_i has at most an equivalent marking in \mathcal{N}_j for $0 \leq i < j$. Every valid marking of \mathcal{N}_j has a unique equivalent marking in \mathcal{N}_i when $0 \leq i < j$.

Proof: By definitions 6.1 and 6.2, for any \mathcal{N}_i a valid marking μ_i of \mathcal{N}_i has a unique equivalent marking μ in \mathcal{N}_0 . Also, μ_i is the unique equivalent marking of μ in \mathcal{N}_i . Indeed, the marking of the control places of \mathcal{N}_i are the values of the excess variables associated to $L_i\mu \geq b_i$. The marking of the other places that do not appear in the original net \mathcal{N}_0 must be zero, in order that μ_i be valid. So μ_i can have only one equivalent marking μ_j in \mathcal{N}_j . The equivalent marking μ_j may not exist if μ , the equivalent marking of μ_i in \mathcal{N}_0 , does not satisfy the additional constraints added in iterations $i, \dots, j - 1$.

Because the constraints of iteration j , (L_j, b_j) and (L_{j0}, b_{j0}) , include the constraints of iteration i , (L_i, b_i) and (L_{i0}, b_{i0}) , it is clear that $L_j\mu \geq b_j \Rightarrow L_i\mu \geq b_i$ and $L_{j0}\mu \geq b_{j0} \Rightarrow L_{i0}\mu \geq b_{i0}$. So, if μ_j is a valid marking of \mathcal{N}_j , and μ_i is μ_j restricted to the places of \mathcal{N}_i , μ_i is also valid. By definition, if the marking μ of \mathcal{N}_0 is equivalent to μ_j then μ is μ_j restricted to the places of \mathcal{N}_0 . Because μ_i and μ_j have the same equivalent marking in \mathcal{N}_0 , they are therefore equivalent. \square

Proposition 6.10 *The equivalence of markings is an equivalence relation.*

Proof: The proof follows immediately by checking the symmetry, reflexivity and transitivity of the relation. \square

In [24] it was shown that adding control places to a net results in an incidence matrix of the form

$$D_2 = \begin{bmatrix} D_1 \\ D_c \end{bmatrix} \quad (25)$$

where D_1 is the incidence matrix of the initial net.

Proposition 6.11 *Let D_i and D_j be the incidence matrices of \mathcal{N}_i and \mathcal{N}_j , $i < j$. If no transitions were split in iterations $i, \dots, j - 1$, then D_j can be written in the form:*

$$D_j = \begin{bmatrix} D_i \\ D_c \end{bmatrix} \quad (26)$$

where the lines of D_c correspond to the control places added in iterations $i, \dots, j - 1$.

Proof: Because no transitions were split, the inequalities enforced from iteration i to $j - 1$ can be written only in term of the places of \mathcal{N}_i (see section 5.2). Then, by enforcing these linear inequalities directly to \mathcal{N}_i the closed loop is the same net as \mathcal{N}_j , and so the incidence matrix can be written as in equation (26) by Theorem 4.1 of section 4.2. \square

Proposition 6.12 *Let μ_i and μ_k be two markings of \mathcal{N}_i and \mathcal{N}_k , $i < k$.*

- (a) μ_i and μ_k are equivalent markings if and only if they are valid and $\forall p \in P_i$, $\mu_i(p) = \mu_k(p)$.
- (b) Assume that μ_i and μ_k are equivalent. Let t be an arbitrary transition of \mathcal{N}_i . If $\sigma_{i,k}(t)$ is enabled in \mathcal{N}_k , then t is enabled in \mathcal{N}_i .
- (c) If S_i is an active siphon of \mathcal{N}_i and $\mu_k(p) = 0 \forall p \in S_i$, then μ_k is not a valid marking of \mathcal{N}_k . However, if $\mu_i(p) = 0 \forall p \in S_i$, μ_i may be a valid marking of \mathcal{N}_i .
- (d) Consider that the original Petri net has controllable and observable transitions. If μ_i is a valid marking and it does not have an equivalent marking in \mathcal{N}_k , j exists, such that $i \leq j < k$, \mathcal{N}_j has a marking μ_j equivalent to μ_i and \mathcal{N}_j has an empty active siphon with respect to μ_j .

(e) If μ_i and μ_k are equivalent, $t \in T_0$, $\mu_i[\sigma_{0,i}(t)] > \mu'_i$ and $\mu_k[\sigma_{0,k}(t)] > \mu'_k$ then μ'_i and μ'_k are equivalent.

Proof: (a) Two markings are equivalent if they are valid. If valid, the marking of the places from replacement sequences are zero, while equivalence implies $\mu_i(p) = \mu_k(p) \forall p \in P_0$. The marking of the common control places of \mathcal{N}_i and \mathcal{N}_j are equal, being uniquely determined by the marking of the original places, for all valid markings (see section 5.2.) Hence the conclusion follows. On the other hand, by Proposition 6.9, μ_i and μ_j have equivalent markings $\mu_{0,i}$ and $\mu_{0,j}$ in \mathcal{N}_0 . Because $P_0 \subseteq P_i$ and $\forall p \in P_i \mu_i(p) = \mu_k(p)$: $\mu_{0,i} = \mu_{0,j}$. Therefore μ_i and μ_j are equivalent.

(b) As a basic split transition property, firing a transition t of \mathcal{N} requires the same marking of places in \mathcal{N} as firing the replacing sequence $\sigma(t)t$ in \mathcal{N}' , the net obtained by splitting t (refer to section 4.1). Because successive transition split does not affect this property, firing $\sigma_{i,k}(t)$ in \mathcal{N}_k has the same marking requirements on the places of \mathcal{N}_i as that of firing t in \mathcal{N}_i .

(c) The deadlock prevention procedure adds constraints for all uncontrolled active siphons. So, the constraints (L_k, b_k) and (L_{k0}, b_{k0}) on \mathcal{N}_k include the requirement that active siphons of previous iterations be controlled. So μ_k cannot satisfy these constraints, and therefore is not a valid marking of \mathcal{N}_k . Further on, if S_i is not implicitly controlled by the constraints added in the iterations $1, 2, \dots, i-1$, there are valid markings of \mathcal{N}_i such that S_i has no tokens.

(d) Let (L_x, b_x) and (L_{x0}, b_{x0}) be the constraints associated to \mathcal{N}_x , where $x > i$ is the first index such that μ_i does not satisfy one or both of (L_x, b_x) and (L_{x0}, b_{x0}) . Because the requirements that are not satisfied only can correspond to the condition that some siphons of \mathcal{N}_{x-1} be not empty, the conclusion follows for $j = x - 1$.

(e) Because no tokens remain in split replacement places by firing the entire sequences $\sigma_{0,i}(t)$ and $\sigma_{0,k}(t)$ replacing t , both μ'_i and μ'_k are valid. Let $\mu'_{0,i}$ and $\mu'_{0,k}$ be their equivalent markings in \mathcal{N}_0 and μ_0 the equivalent marking of μ_i and μ_k in \mathcal{N}_0 . By part (b), $\mu_0[t > \mu'_{0,i}]$ and $\mu_0[t > \mu'_{0,k}]$. So $\mu_{0,i} = \mu_{0,k}$ and hence μ'_i and μ'_k are equivalent. \square

Proposition 6.13 *Let $\mu_{i,1}$ and $\mu_{j,1}$ be two equivalent markings of \mathcal{N}_i and \mathcal{N}_j , $i < j$. If $\mu_{i,2}$ and $\mu_{j,2}$ are two other equivalent markings of \mathcal{N}_i and \mathcal{N}_j and a transition t exists, such that $\mu_{i,1}[t > \mu_{i,2}]$ in \mathcal{N}_i , then $\mu_{j,1}[\sigma_{i,j}(t)] > \mu_{j,2}$ in \mathcal{N}_j .*

Proof: If $\sigma_{i,j}(t)$ is enabled by $\mu_{j,1}$ and $\mu_{j,1}[\sigma_{i,j}(t)] > \mu'_{j,2}$ then $\mu'_{j,2}(p) = \mu_{i,2}(p) \forall p \in P_i$ and $\mu'_{j,2}(p) = 0 \forall p \in P_s$ follow directly from split transition properties, where P_s is the set of the places resulted through transition splits. Therefore, since $\mu_{j,1}$ is valid, $\mu'_{j,2}$ is also, because the constraints are satisfied (see Proposition 6.5). Then by Propositions 6.9 and 6.12(a), $\mu_{j,2} = \mu'_{j,2}$.

If $\sigma_{i,j}(t)$ is not enabled by $\mu_{j,1}$, let k be the first index such that $\sigma_{i,k}(t)$ is not enabled in \mathcal{N}_k by $\mu_{k,1}$, which is the equivalent marking of $\mu_{i,1}$ in \mathcal{N}_k . Because $\sigma_{i,k-1}(t)$ is enabled in \mathcal{N}_{k-1} , there is a control place that prevents $\sigma_{i,k}(t)$ to fire, because of a constraint added in iteration $k-1$. So $\mu_{k-1,2}$ cannot satisfy one of the constraints added in iteration $k-1$, and therefore $\mu_{k-1,2}$ has no equivalent marking in \mathcal{N}_k . But this is a contradiction, because $j \geq k$ implies $\exists \mu_{k,2}$ equivalent to $\mu_{j,2}$ (Proposition 6.9), and $\mu_{j,2}$ is equivalent to $\mu_{i,2}$, which in turn is equivalent to $\mu_{k-1,2}$. (The fact that the markings $\mu_{i,2}$ and $\mu_{k-1,2}$ are equivalent follows from $\mu_{i,2}(p) = \mu_{k-1,2}(p) \forall p \in P_i$ because of the split transition construction (section 4.1), $\mu_{i,1}$ and $\mu_{k-1,1}$ are equivalent, $\mu_{i,1}[t > \mu_{i,2}]$ in \mathcal{N}_i and $\mu_{k-1,1}[t > \mu_{k-1,2}]$ in \mathcal{N}_{k-1} .) \square

Corollary 6.1 *Let $\mu^{(1)}$ and $\mu^{(2)}$ be two markings of \mathcal{N}_0 such that $\mu^{(1)}[t > \mu^{(2)}$ (where $t \in T_0$) and satisfying the constraints produced by the procedure after termination: $L\mu^{(1)} \geq b$, $L_0\mu^{(1)} \geq b_0$, $L\mu^{(2)} \geq b$, $\mu^{(1)}[t > \mu^{(2)}$.*

Then the markings $\mu_k^{(1)}$ and $\mu_k^{(2)}$ of \mathcal{N}_k equivalent to $\mu^{(1)}$ and respectively to $\mu^{(2)}$ are defined for any k , $\mu_k^{(1)}$ enables $\sigma_{0,k}(t)$ and $\mu_k^{(1)}[\sigma_{0,k}(t) > \mu_k^{(2)}]$.

Proof: Because $\mu^{(1)}$ and $\mu^{(2)}$ satisfy the constraints generated by the procedure, all the control places that were added have a well defined marking, in accord with the supervisory policy. So $\mu_k^{(1)}$ and $\mu_k^{(2)}$ are defined for all iteration indices k . Then, by Proposition 6.13, the remainder of the conclusion follows. \square

Theorem 6.1 *The following statements are true:*

- (a) *Let σ_i be an arbitrary firing sequence of \mathcal{N}_i and $\sigma_j = \sigma_{i,j}(\sigma_i)$ the corresponding firing sequence in \mathcal{N}_j , $i < j$. If μ_j is a marking of \mathcal{N}_j that enables σ_j , then the marking μ_i of \mathcal{N}_i such that $\mu_i(p) = \mu_j(p) \forall p \in P_i$ enables σ_i . Also if $\mu_i[\sigma_i > \mu'_i]$ and $\mu_j[\sigma_j > \mu'_j]$ then $\mu'_i(p) = \mu'_j(p) \forall p \in P_i$.*
- (b) *Assume that the procedure does not start with initial constraints, or if it does, all valid markings μ of \mathcal{N}_0 have the property that exists $\mu' \geq \mu$, μ' has an equivalent marking in \mathcal{N}_k . Let σ be an arbitrary transition sequence of \mathcal{N}_0 and $\sigma_k = \sigma_{0,k}(\sigma)$ the corresponding sequence in \mathcal{N}_k . If a valid marking μ of \mathcal{N}_0 exists which enables σ , a valid marking μ_k of \mathcal{N}_k exists which enables σ_k .*
- (c) *In the conditions of part (b), if some marking μ'_k of \mathcal{N}_k exists which enables σ_k , then a marking of \mathcal{N}_k exists which enables σ_k and which also is valid.*

Proof: (a) If the property is true for all σ_i finite, than it is also true for all σ_i infinite. Indeed, if the property would not be true for some σ_i infinite, then there is a partition $\sigma_i = \sigma_{i,1}\sigma_{i,2}$ such that $\sigma_{i,1}$ is finite and $\sigma_{i,1}$ does not satisfy the property. Therefore, in what follows the proof considers only the case when σ_i is finite: $\sigma_i = t_1, t_2, \dots, t_s$, where every t_k is a transition of T_i .

The set P_j is the disjoint set union $P_j = P_i \cup \mathcal{C} \cup P_R$, where \mathcal{C} is the set of control places added in the iterations i through $j - 1$ and P_R is the set of places resulted from split transition operations in the same iterations. Firing $\sigma_{i,j}(t_1)$ requires the same number of tokens from places of P_i as firing t_1 in \mathcal{N}_i , as a split transition property, and may require additional tokens from \mathcal{C} . Therefore t_1 is enabled by μ_i . Let $\mu_{i,1}$ and $\mu_{j,1}$ be the markings reached by firing t_1 and $\sigma_{i,j}(t_1)$, respectively. Again, as a split transition property, firing t_1 in \mathcal{N}_i and $\sigma_{i,j}(t_1)$ in \mathcal{N}_j modifies in the same way the marking of P_i , and firing $\sigma_{i,j}(t_1)$ does not change the marking of P_R . Hence $\mu_{j,1}(p) = \mu_{i,1}(p) \forall p \in P_i$ and $\mu_{j,1}(p) = \mu_j(p) \forall p \in P_R$. Continuing in the same way with t_2 , t_2 is enabled and the markings reached by firing t_2 and $\sigma_{i,j}(t_2)$ satisfy the same property, and by induction it follows that the markings $\mu_{i,1} \dots \mu_{i,s}$ and $\mu_{j,1} \dots \mu_{j,s}$ exist such that $\mu_i[t_1 > \mu_{i,1}[t_2 > \dots \mu_{i,s-1}[t_s > \mu_{i,s}$, $\mu_j[\sigma_{i,j}(t_1) > \mu_{j,1}[\sigma_{i,j}(t_2) > \dots \mu_{j,s-1}[\sigma_{i,j}(t_s) > \mu_{j,s}$, $\mu_{j,s}(p) = \mu_{i,s}(p) \forall p \in P_i$ and $\mu_{j,s}(p) = \mu_j(p) \forall p \in P_R$. So the conclusion follows with $\mu'_j = \mu_{j,s}$ and $\mu'_i = \mu_{i,s}$.

(b) This proof uses induction. The induction is on $i = 1, 2, \dots, k - 1$. Note that all valid markings μ of \mathcal{N}_0 have the property that there is a valid marking $\mu' \geq \mu$ such that μ' has an equivalent marking in \mathcal{N}_i , for $i = 1, 2, \dots, k$. This is so because the set of constraints a valid marking of \mathcal{N}_i must satisfy is a subset of the constraints of \mathcal{N}_k , for $i < k$. Suppose that μ_i of \mathcal{N}_i enables the sequence $\sigma_i = \sigma_{0,i}(\sigma)$. We build $\mu_{i,2}$ such that $\mu_{i,2}$ enables σ_i and $\mu_{i,2}$ has an equivalent marking μ_{i+1} in \mathcal{N}_{i+1} . By Proposition 6.13, this would show that μ_{i+1} enables $\sigma_{i+1} = \sigma_{0,i+1}(\sigma)$ in \mathcal{N}_{i+1} .

Let \mathcal{S}_0 denote the set of active siphons of \mathcal{N}_i which either are token-free under the marking μ_i , or become so by firing σ_i . By Proposition 6.3(c) each siphon $s \in \mathcal{S}_0$ includes at least an original place and/or a control place. Using the relations from section 5.2, a valid marking $\mu_{i,2} \geq \mu_i$ can be chosen such that $\forall s \in \mathcal{S}_0$,

$\sum_{p \in s} \mu_{i,2}(p) \geq \sum_{p \in s} \mu_i(p) + 1$. By construction, for the marking $\mu_{i,2}$ no siphon s is token-free, $\mu_{i,2}$ also enables σ_i and no siphon s becomes token-free when firing σ_i . Thus $\mu_{i,2}$ has an equivalent marking μ_{i+1} which enables σ_{i+1} in \mathcal{N}_{i+1} .

(c) Let P_R be the set of all places of \mathcal{N}_k that have resulted through transition split in previous iterations. Let μ''_k be defined as $\mu''_k(p) = \mu'_k(p) \forall p \in P_k \setminus P_R$ and $\mu''_k(p) = 0 \forall p \in P_R$. Then μ''_k enables σ_k . Indeed, let's assume the contrary. Then σ_k can be partitioned in the sequence $\sigma_k = \sigma_1 t_x \sigma_2$, where $t_x \in T_k$, $\mu''_k[\sigma_1 > \mu'_x$, $\mu'_k[\sigma_1 > \mu_x$, μ_x enables t_x but μ'_x does not enable t_x . The only possibility is that $P_R \cap \bullet t_x = \{p_x\}$, $\mu_x(p_x) > 0$ and $\mu'_x(p_x) = 0$ (refer also to the split transition construction in section 4.1.) Because $\sigma_k = \sigma_{0,k}(\sigma)$ and σ is a sequence of transitions of \mathcal{N}_0 , σ_1 has the form $\sigma_{0,k}(t_1)\sigma_{0,k}(t_2)\dots\sigma_{0,k}(t_n)\sigma_x$, where t_1, \dots, t_n are not necessarily distinct transitions of T_0 and σ_x is the first part of some $\sigma_{0,k}(t_{n+1})$. It follows that $\sigma_{0,k}(t_{n+1})$ has the form $\sigma_x t_x \sigma_y$. However, firing σ_x always brings a token in the replacement place p_x such that $p_x \bullet = \{t_x\}$, which contradicts $\mu'_x(p_x) = 0$.

Because μ''_k enables σ_k , we can always choose a valid marking μ_k such that $\mu_k \geq \mu''_k$ (see the form of the constraints added by the procedure in section 5.2.) Therefore μ_k is valid and enables σ_k . \square

Corollary 6.2 *Consider the assumption of Theorem 6.1(b) to be true.*

- (a) *Deadlock-freedom cannot be enforced for any finite marking in \mathcal{N}_k if and only if it also cannot be enforced in \mathcal{N}_0 .*
- (b) *Liveness cannot be enforced for any finite marking in \mathcal{N}_k if and only if it also cannot be enforced in \mathcal{N}_0 .*

Proof: Deadlock-freedom may be enforced in a net in which there is a marking allowing an infinite firing sequence. Thus necessity results directly from Theorem 6.1(b) and sufficiency from Theorem 6.1 parts (a) and (c), where part (c) is used for the case when initial constraints exists, and so not all possible markings of \mathcal{N}_0 are valid. The proof of part (b) is similar. \square

Theorem 6.1(a) showed that if $i < j$ and μ_i, μ_j are equivalent markings of \mathcal{N}_i and \mathcal{N}_j , then a firing sequence σ_i is always enabled by μ_i in \mathcal{N}_i , when its counterpart $\sigma_j = \sigma_{i,j}(\sigma)$ is enabled by μ_j in \mathcal{N}_j . The converse generally is not true. However, it is true for the particular case when $i = 0$, because \mathcal{N}_1 differs from \mathcal{N}_0 only by the fact that \mathcal{N}_1 is the PT-transformed version of \mathcal{N}_0 and no constraints were yet enforced.

Proposition 6.14 *Every valid marking μ of \mathcal{N}_0 has an equivalent marking μ' in \mathcal{N}_1 . Moreover, if μ and μ' are equivalent, σ is a transition sequence enabled by μ and $\sigma' = \sigma_{0,1}(\sigma)$, then μ' enables σ' .*

Proof: The equivalent marking μ' of μ is defined by $\mu'(p) = \mu(p) \forall p \in P_0$ and $\mu'(p) = 0 \forall p \in P_1 \setminus P_0$. The fact that $\forall t \in T_0, \mu[t > \mu_1$ implies both $\mu'[\sigma_{0,1}(t) > \mu'_1$ and μ_1 is equivalent to μ'_1 , is a property of transition split. Thus the remainder of the conclusion follows immediately. \square

6.2 Main Results

The most important results concerning the procedure are the theorems proving deadlock prevention and an important permissivity property. We include them in section 6.2.1. The termination of the procedure is considered in section 6.2.2.

In Theorem 6.2, it is shown that the procedure provides a supervisor preventing deadlock. Theorem 6.3 gives a permissivity estimate of the supervisor generated by the procedure: the supervisor is at least as permissive as any supervisor enforcing liveness for all transitions which can be made live.

We use the same notations as in the description of the procedure in section 5.4, as well as the notations from section 6.1.1. That is, in every iteration i the *active subnet* $\mathcal{N}_i^A = (P_i^A, T_i^A, F_i^A, W_i^A)$ and the *total net* $\mathcal{N}_i = (P_i, T_i, F_i, W_i)$, $\sigma_{i,j}(\sigma)$ the replacement sequence in \mathcal{N}_j of the transition sequence σ of \mathcal{N}_i , $i < j$ and $\sigma_{i,j}(t)$ the replacement sequence in \mathcal{N}_j of the transition t of \mathcal{N}_i .

6.2.1 Success and Permissivity Results

Lemma 6.1 *Assume that the procedure terminated normally (that is not at steps B or C.4) in $k-1$ iterations. Then \mathcal{N}_k is deadlock-free for all valid initial markings.*

Proof: In every iteration, the new minimal active siphons are controlled in step 2 of the procedure. The controlled siphons remain controlled (for valid markings) in the subsequent iterations (Proposition 6.7). Because the procedure terminated, all minimal active siphons of \mathcal{N}_k are controlled for valid initial markings, as the marking of the control places is defined for valid markings. Therefore, by Proposition 3.5, \mathcal{N}_k is deadlock-free for all valid initial markings. \square

Theorem 6.2 *Given a target Petri net \mathcal{N}_0 , assume that the procedure terminates and that it terminates normally (i.e. not at steps B or C.4). Let (L, b) and (L_0, b_0) denote the two sets of constraints generated by the procedure. The target net \mathcal{N}_0 in closed loop with the supervisor enforcing $L\mu \geq b$ is deadlock-free for all initial markings μ_0 of \mathcal{N}_0 such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$.*

Proof: By construction, every marking of the original Petri net \mathcal{N}_0 which satisfies the constraints has an equivalent marking in \mathcal{N}_k such that all active siphons of \mathcal{N}_k are well-marked. The proof uses the fact that for any such marking, there is an infinite transition sequence enabled in \mathcal{N}_k (Lemma 6.1). It proves by contradiction that no marking of \mathcal{N}_0 satisfying the constraints is a deadlock marking for the closed loop Petri net.

Assume that from a good initial marking μ_0 of \mathcal{N}_0 , the closed loop net (let it be \mathcal{N}_S) reaches a marking μ such that all possible firings in \mathcal{N}_0 would lead either to deadlock markings or to markings which do not comply with the enforced constraints, $L\mu \geq b$ (this is deadlock in \mathcal{N}_S .)

Let $\mu_{0,k}$ and μ_k be the equivalent markings of μ_0 and μ in \mathcal{N}_k . Because μ_k is valid, by Lemma 6.1 μ_k enables an infinite transition sequence σ in \mathcal{N}_k . Let T_R be the set of transitions that appeared by split transition operations. Let \mathcal{C} be the set of control places. Revisiting the transition split operation (section 4.1) and by Proposition 6.2(b), firing any $t \in T_R$ always reduces the marking of some places in $P_0 \cup \mathcal{C}$ and firing $t \in T_0$ (note that $T_0 = T_k \setminus T_R$) may increase the marking of some places in $P_0 \cup \mathcal{C}$. Because the total marking of $P_0 \cup \mathcal{C}$ is finite, σ must include transitions $t \in T_0$. Let t_1 be the first transition in T_0 that appears in σ . Since all transition of σ before t_1 are in T_R , firing them only decrease markings of $P_0 \cup \mathcal{C}$, and t_1 cannot fire unless all other transitions of $\sigma_{0,k}(t_1)$ fired before (as μ_k is valid), it follows that $\sigma_{0,k}(t_1)$ is enabled by μ_k . But this implies that t_1 is also enabled by μ in \mathcal{N}_S , which is a contradiction. \square

The assumptions of Theorem 6.2 are that the procedure terminates and that it terminates normally. The next result characterizes the cases when the procedure terminates with a failure at step B. It shows

that when there are no uncontrollable and unobservable transitions, this implies that deadlock prevention is impossible, given the initial constraints (if any are given.)

Proposition 6.15 *Deadlock cannot be prevented for any initial marking if the procedure terminates at step B.*

Proof: If the maximal active subnet of the target Petri net is empty, the procedure terminates at step B. An empty maximal active subnet implies that deadlock is unavoidable for any initial marking, in view of Corollary 3.3. Next, note that the procedure terminates at step B only if the maximal active subnet of \mathcal{N}_0 is empty. Indeed, assume the contrary. Then the procedure may terminate at step B after it is restarted from a step C.4 with $X \neq \emptyset$. Assuming such an occurrence, let i be the iteration number at C.4. Since the procedure did not terminate at C.4, there is a nonempty active subnet of \mathcal{N}_{i+1} excluding the transitions of X ; let it be \mathcal{N}_{i+1}^A . Next, note that $T_0 \cap T_{i+1}^A \neq \emptyset$. There is a valid marking of \mathcal{N}_{i+1} enabling an infinite firing sequence σ in which the transitions appearing infinitely often are the transitions of T_{i+1}^A . This implies that there is a marking of \mathcal{N}_0 enabling a firing sequence σ_0 in which the transitions which appear infinitely often are the transitions of $T_0 \cap T_{i+1}^A$. Then $T_0 \cap T_{i+1}^A$ defines a nonempty active subnet of \mathcal{N}_0 which does not contain the transitions of X . This contradicts the assumption that the procedure terminates at step B. \square

Given a set T of transitions, we say that a Petri net is T -live if all transitions in T are live.

Proposition 6.16 *Let T_0^A be the set of transitions of the maximal active subnet of \mathcal{N}_0 . T_0^A -liveness is not enforceable in \mathcal{N}_0 for any initial marking if the procedure terminates at step B, or if \mathcal{N}_0 has no uncontrollable and no unobservable transitions and an infeasibility occurs at a step C.2.c.*

Proof: The first part is a direct consequence of Proposition 6.15. For the second part, note that infeasibilities may occur only if initial constraints are given. Consider the occurrence of the first infeasibility: there is an active siphon which due to the initial constraints must be empty for all markings. We show in the proof of Theorem 6.3 that this implies that there is no T_0^A -liveness enforcing supervisor when the markings are restricted by the given initial constraints. \square

In the case of Petri nets without uncontrollable and unobservable transitions, termination at step C.4 may only occur only if initial constraints are given and infeasibilities occur at one or more steps C.2.c. Therefore, Proposition 6.16 has the simple consequence below:

Proposition 6.17 *When \mathcal{N}_0 has no uncontrollable and no unobservable transitions, T_0^A -liveness is not enforceable in \mathcal{N}_0 for any initial marking if the procedure terminates at step C.4.*

As shown in section 4.3, the siphon control approach used by the procedure enforces inequalities of the form $\sum_{p \in S} \alpha_p \mu(p) \geq 1$ in order to control a siphon S , where α_p are nonnegative integers. When all transitions are controllable and observable, $\alpha_p = 1 \forall p \in S$. The coefficients α_p may have other values when uncontrollable and unobservable transitions are present. The next two results are proved for the case when for all controlled siphons S , the enforced constraint satisfies $\alpha_p \neq 0$. The requirement is always satisfied for the Petri nets with controllable and observable transitions. The meaning of the requirement is that all minimal active siphons S are maximally permissive controlled (that is, only the markings μ which satisfy $\mu(p) = 0 \forall p \in S$ are forbidden.)

Lemma 6.2 *Assume that for all minimal active siphons S controlled by the procedure in the iterations $1 \dots i$ ($i \geq 1$) the enforced constraint has the form $\sum_{p \in S} \alpha_p \mu(p) \geq 1$, where α_p are positive integers. Let S be an active siphon of \mathcal{N}_{i+1} which does not appear in \mathcal{N}_i . Let μ_{i+1} be a valid marking of \mathcal{N}_{i+1} and μ_i the equivalent marking in \mathcal{N}_i . Assume that S is empty for the marking μ_{i+1} . Let t_s be an arbitrary transition of \mathcal{N}_i with the property that there is a transition $t \in S \bullet$ of \mathcal{N}_{i+1} such that $t_s = t$ or t_s is split in \mathcal{N}_{i+1} and t appears in the transition replacing sequence $\sigma_{i,i+1}(t_s)$. If $\exists \mu \in \mathcal{R}(\mu_i)$ such that $\mu[t_s > \mu_s$, then (\mathcal{N}_i, μ_s) has at least one empty active siphon.*

Proof: Let \mathcal{C} be the set of control places added to \mathcal{N}_{i+1} . Note that P_{i+1} is made up of P_i , \mathcal{C} and the set of places that result through transition split, $P_R = P_{i+1} \setminus (P_i \cup \mathcal{C})$. Let σ be the firing sequence that was used to reach μ : $\mu_i[\sigma > \mu$. Consider firing σ in (\mathcal{N}_i, μ_i) and $\sigma' = \sigma_{i,i+1}(\sigma)$ in $(\mathcal{N}_{i+1}, \mu_{i+1})$. The only reason for σ' not to be enabled in \mathcal{N}_{i+1} by the marking μ_{i+1} would be that a control place prevents it.

If σ' is not enabled, $\sigma = \sigma_1 t_1 \sigma_2$, $\mu_i[\sigma_1 > \mu_1$, $\mu_{i+1}[\sigma_{i,i+1}(\sigma_1) > \mu'_1$, μ_1 enables t_1 , but μ'_1 does not enable $\sigma_{i,i+1}(t_1)$. This corresponds to the following: \mathcal{N}_i has an active siphon S_1 , that is controlled in \mathcal{N}_{i+1} with C_1 ; when C_1 was added, $t_1 \in C_1 \bullet$, and if $W(C_1, t_1) > 1$, t_1 was split in step 3 of iteration i in $\sigma_{i,i+1}(t_1)$ or if $W(C_1, t_1) = 1$, $\sigma_{i,i+1}(t_1) = t_1$. So $t_1 \in S_1 \bullet$, and since t_1 would not be allowed by C_1 to fire from μ_1 , it means that firing it would make S_1 empty. Since t_1 is fired in the sequence $\sigma = \sigma_1 t_1 \sigma_2$, after σ is fired, S_1 is an empty active siphon in (\mathcal{N}_i, μ_s) .

If σ' is enabled by μ_{i+1} , let μ' be the marking reached: $\mu_{i+1}[\sigma' > \mu'$. Because σ' may contain only entire replacements of split transitions and μ_{i+1} is a valid marking (which implies $\mu_{i+1}(p) = 0 \forall p \in P_R$), $\mu'(p) = 0 \forall p \in P_R$. Also, μ_{i+1} and μ_i are equivalent and $\sigma' = \sigma_{i,i+1}(\sigma)$, therefore $\mu(p) = \mu'(p) \forall p \in P_i$ (Theorem 6.1(a)). Because S is a siphon, S empty for μ_{i+1} implies S empty for all reachable markings, and so for μ' too. There are two cases: (a) t_s is not split in \mathcal{N}_{i+1} and (b) t_s is split.

(a) If t_s is not split, $\bullet t_s \cap P_R = \emptyset$. Further on, μ enables t_s in \mathcal{N}_i but μ' does not enable t_s in \mathcal{N}_{i+1} , so in \mathcal{N}_{i+1} , $\bullet t_s \cap \mathcal{C} \neq \emptyset$ and there is $C \in \bullet t_s \cap \mathcal{C}$ such that $\mu'(C) = 0$. Let S_C be the active siphon of \mathcal{N}_i controlled by C . t_s was not split, so $W(C, t_s)$ was 1; t_s enabled by μ , $\mu'(C) = 0$ and $t_s \in C \bullet \Rightarrow t_s \in (S_C \bullet) \setminus (\bullet S_C)$. Since $S_C \subseteq P_i$ and $\mu'(C) = 0$, $\sum_{p \in S_C} \mu(p) = 1$. Because t_s is enabled by μ , firing t_s empties S_C , so there is an empty active siphon in (\mathcal{N}_i, μ_s) .

(b) If t_s was split, then t_s was connected to one or more of the control places C of \mathcal{C} , for only transitions connected to such places are split. (This is so because for all $i \geq 1$ \mathcal{N}_i is PT-ordinary, and hence only the new added control places can make the Petri net not to be PT-ordinary). We let \mathcal{C}_S be the set of control places added to $\bullet t_s$ in the iteration i . By recalling the split transition operation (sections 4.1), it is easy to notice that $t \in S \bullet$ implies $\exists C \in \mathcal{C}_S$ such that $C \in S$. Let S_C be the active siphon controlled by C . Since $C \in S$ and S is empty, $\sum_{p \in S_C} \mu(p) = 1$. Since before the split of t_s : $C \in \bullet t_s$, firing t_s in \mathcal{N}_i reduces the marking of S_C , and since the total marking of S_C is one, S_C becomes empty. \square

Note that Lemma 6.2 applies for $i \geq 1$. It also applies for $i = 0$ when $\mathcal{N}_1 = \mathcal{N}_0$, that is when \mathcal{N}_0 is PT-ordinary.

Theorem 6.3 *Assume that for all minimal active siphons S controlled by the procedure, the enforced constraint has the form $\sum_{p \in S} \alpha_p \mu(p) \geq 1$, where α_p are positive integers, and no failure to transform a constraint to an admissible form occurred. The deadlock prevention method provides a supervisor at least as permissive as any supervisor subject to the same initial constraints (if any initial constraints are given) and which en-*

forces that all the transitions of the target Petri net which appear in the maximal active subnet are live, if any such supervisor exists.

Proof: Let \mathcal{S} be the set of supervisors satisfying the initial constraints, which also enforce that all transitions which appear in the maximal active subnet are live in the target Petri net. Note that when we compare our procedure to other supervisor we assume an initial marking for which that supervisor is defined: we do not require the supervisors in \mathcal{S} to be defined for all initial markings for which the supervisor given by our procedure is defined.

We first consider the case when there are no initial constraints. The proof is by contradiction. It shows that any marking forbidden by the deadlock prevention method also is forbidden by any supervisor in \mathcal{S} . Recall that our procedure forbids markings which will produce an empty active siphon in an \mathcal{N}_k for some k .

Let $\mu^{(1)}$ be a marking of \mathcal{N}_0 and $\mu_k^{(1)}$ the equivalent marking in \mathcal{N}_k . Suppose that for the marking $\mu_k^{(1)}$ there is an empty active siphon S_k in \mathcal{N}_k . Because $\mu_k^{(1)}$ is valid, S_k is a new siphon which does not appear in \mathcal{N}_{k-1} ; $\mu^{(1)}$ is forbidden by iteration k , which adds the constraint that S_k be well-marked.

Assume that $\mu^{(1)}$ is not forbidden by some supervisor enforcing in \mathcal{N}_0 that all transitions of the active subnet are live and that there is an infinite firing sequence σ enabled by $\mu^{(1)}$ such that every transition of \mathcal{N}_0^A appears infinitely often in σ . According to Lemma 6.2, there is a transition t'_{k-1} of \mathcal{N}_{k-1} such that in any possible firing sequence, after t'_{k-1} fires in \mathcal{N}_{k-1} , there is an empty active siphon S_{k-1} of \mathcal{N}_{k-1} . Let $t_{k-1} \in T_0$ such that t'_{k-1} appears in $\sigma_{0,k-1}(t_{k-1})$. Let $\mu^{(2)}$ be the marking of \mathcal{N}_0 that appears while σ is fired, immediately after t_{k-1} fires for the first time. Also, let σ_1 be the subsequence of σ that was fired so far, that is $\mu^{(1)}[\sigma_1 > \mu^{(2)}$. Let $i \geq 0$ be the largest integer, such that $\mu_i^{(2)}$ is an equivalent marking of $\mu^{(2)}$ in \mathcal{N}_i . By Lemma 6.2, $i \leq k-1$. Indeed, if σ_1 is allowed to fire in \mathcal{N}_{k-1} , there is an empty siphon S_{k-1} for the marking $\mu_{k-1}^{(2)}$, but there is no valid marking of \mathcal{N}_k such that S_{k-1} is empty. Now, the fact that $\mu^{(2)}$ has an equivalent marking $\mu_i^{(2)}$ in \mathcal{N}_i but not in \mathcal{N}_{i+1} shows that there is an empty active siphon S_i in \mathcal{N}_i and that S_i does not appear in \mathcal{N}_{i-1} (Proposition 6.12(d)). Further on, the same idea as before is used, that a transition t_{i-1} with the same property as t_{k-1} exists, and following this idea, an index $j \leq i-1$ is found such that for the marking $\mu^{(3)}$ of \mathcal{N}_0 there is an empty active siphon in \mathcal{N}_{j-1} . This procedure is repeated and finally two cases may appear (Lemma 6.2 applies for $i > 0$ only) after the first n transitions of σ are fired, where n is a finite number. Let σ_p denote the sequence that enumerates the first n transitions of σ , and let $\mu^{(p)}$ be the marking reached by firing σ_p (that is, $\mu^{(1)}[\sigma_p > \mu^{(p)}$) and $\mu_1^{(p)}$ the equivalent marking in \mathcal{N}_1 . Then (a) there is an empty active siphon in $(\mathcal{N}_0, \mu^{(p)})$ or (b) there is an empty active siphon in $(\mathcal{N}_1, \mu_1^{(p)})$. Case (a) contradicts the fact that every transition appears infinitely often in σ and $\mu^{(1)}$ enables σ , since after n firings none of the transitions in the postset of the empty siphon may fire again. Case (b) leads to the same type of contradiction, because by Proposition 6.14 the sequence $\sigma' = \sigma_{0,1}(\sigma)$ is enabled by $\mu_1^{(1)}$, where $\mu_1^{(1)}$ is the equivalent marking of $\mu^{(1)}$ in \mathcal{N}_1 , and by construction every transition of \mathcal{N}_1^A appears infinitely often in σ' .

The case when there are initial constraints is similar to the case when there are no such constraints if the procedure is never in the situation that the constraints at step C:2c of the procedure are infeasible. In the case when infeasibilities at some steps C:2c occur, consider the first occurrence: there is an active siphon S which must be empty for all valid markings, in order not to have a conflict with the initial constraints. (In such a situation, being unable to control S , the procedure shrinks the active subnet such that S is no longer an active siphon.) Then, by the first part of the proof, there are no supervisors in \mathcal{S} . (\mathcal{S} is empty, as the initial constraints conflict with the requirement that the transitions of the active subnet are live.) \square

Theorem 6.3 states that the supervisor provided by the procedure is more permissive than any supervisor which enforces that all transitions of the maximal active subnet are live in the target net. Note that this is not necessarily the same thing as maximally permissive deadlock prevention (refer to section 6.4.3.) The comparison assumes that the other arbitrary supervisors are subjected to the same initial constraints. In particular, when the target Petri net is repetitive, liveness enforcing supervisors exist, and so we have the following corollary. Note also that Theorem 6.3 always applies for Petri nets with controllable and observable transitions.

Corollary 6.3 *In the conditions of Theorem 6.3, the deadlock prevention procedure provides a supervisor at least as permissive as any liveness enforcing supervisor (subject to the same initial constraints), if any such supervisor exists.*

In other words, the corollary states that the set of markings forbidden by the deadlock prevention supervisor is a subset of the set of markings forbidden by any liveness enforcing supervisor. The corollary also shows that if for some Petri net the procedure yields a supervisor which enforces liveness, the supervisor also is maximally permissive. Note that although the procedure was not designed for liveness enforcement, it is common for it to also enforce liveness (in addition to deadlock-prevention). Theorem 6.3 applies for a supervisor obtained after an arbitrary number of iterations; the proof does not assume that the procedure terminates.

6.2.2 Termination Results

The procedure, as defined, may not terminate for any Petri net structure. By analysing cases in which the procedure does not terminate, we considered two approaches to help the procedure terminate. To formally guarantee termination, we restrict the class of Petri nets to *structurally bounded* Petri nets and assume that some bounds of the reachable marking space are known. This is a reasonable assumption for Petri nets modeling real systems, because in general every quantity has some bound. For each of the two changes, if the procedure is started with initial constraints (L_0, b_0) which bound the reachable space, termination can be guaranteed. However note that the two approaches we propose may help termination by themselves, that is even when the initial constraints (L_0, b_0) do not have a bounded feasible region, and not only for structurally bounded Petri nets. A Petri net \mathcal{N} is **structurally bounded** [27] if for all initial markings μ_0 , $\mathcal{R}(\mathcal{N}, \mu_0)$ is bounded.

6.2.2.1 Approach A In this approach we change the procedure as follows: all constraints are stored only in the form in which only the marking of the places of the target net, \mathcal{N}_0 , appears. That is, the marking of the places resulted by transition split is ignored (assumed to be zero.) A siphon is implicitly controlled if the inequality associated to it, which is now written only with respect to the places of \mathcal{N}_0 , is implied by the current set of constraints, also written only with respect to the places of \mathcal{N}_0 .

The difference from the usual approach is that the contribution of the places resulted by transition split is ignored when a siphon is checked whether it is implicitly controlled. Naturally, approach A does not change the procedure for those Petri nets in which the final Petri net \mathcal{N}_k has no split transitions.

Theorem 6.4 *Let \mathcal{N} be a Petri net and (L_i, b_i) be a set of constraints $L_i\mu \geq b_i$, $\mu \geq 0$, with bounded feasible region. Then the deadlock prevention procedure of approach A terminates if started with initial constraints (L_0, b_0) which equal (L_i, b_i) .*

Proof: Let \mathcal{M}_R be the bounded feasible region of $L_i\mu \geq b_i$, with μ nonnegative integer vector. Let F_N be the set of markings forbidden by the control places added up to some point. Let S be the next siphon considered for control, and f_S the set of markings which would be forbidden in the target net \mathcal{N} by enforcing $\sum_{p \in S} \mu(p) \geq 1$. S is not implicitly controlled if $(f_S \setminus F_N) \cap \mathcal{M}_R \neq \emptyset$. Since each siphon which is not implicitly controlled adds at least a new marking $\mu_F \in \mathcal{M}_R$ to the set of forbidden markings, and since \mathcal{M}_R is finite, after we control a finite number of siphons, all new siphons are implicitly controlled and so the procedure terminates. \square

Theorem 6.4 is important because it gives a sufficient (but not necessary) condition for termination which is not very restrictive for real applications, where in general the capacity of every place is finite.

The usage of the procedure with the approach A can be summarized as follows:

- Find a set of constraints $L_i\mu \geq b_i$ with bounded feasible set F , where μ is a nonnegative integer vector, such that for all initial markings μ_0 of \mathcal{N} which are of interest: $\mathcal{R}(\mathcal{N}, \mu_0) \subseteq F$. Let \mathcal{M}_I be the set of initial markings of interest.
- Use the procedure with the approach A and with initial constraints (L_0, b_0) which equal (L_i, b_i) .
- The supervisor can be used for the initial markings $\mu_0 \in \mathcal{M}_I$ which satisfy $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$, where (L, b) and (L_0, b_0) are the two sets of constraints generated by the procedure.

The disadvantage of approach A is that Theorem 6.2, which guarantees deadlock prevention, may not apply in certain cases. Theorem 6.3 still applies, since the siphon control method is the same, and the only difference is that some siphons, which normally wouldn't be considered to be (implicitly) controlled, may be considered so when the approach A is used. (So this difference does not change the proof of Theorem 6.2.)

6.2.2.2 Approach B In this approach we simply use a set of initial constraints with bounded feasible region. This is enough to guarantee termination if transition splits can occur only in finitely many iterations (which in practice is often the case).

Theorem 6.5 *Let \mathcal{N} be a Petri net and (L_i, b_i) a set of constraints $L_i\mu \geq b_i$, $\mu \geq 0$, with bounded feasible region. Assuming that in the case of \mathcal{N} transition splits can occur only in finitely many iterations, the deadlock prevention procedure terminates if started with initial constraints (L_0, b_0) which equal (L_i, b_i) .*

Proof: Note that in step A the constraints (L_i, b_i) are transformed as in section 5.2.5 to a new form $L'_0\mu \geq b'_0$, which is true in all \mathcal{N}_j , $j \geq 1$. By construction, since the feasible set of $L_i\mu \geq b_i$ is bounded (and so finite), so is the feasible set of $L'_0\mu \geq b'_0$. The proof is by contradiction. Assume that the procedure does not terminate. After the last iteration in which transition splits occur, the size of the marking vector is no longer changed. The set of possible markings is bounded to some set \mathcal{M}_R due to the initial constraints. Thus, each time a new constraint is added to (L, b) or (L_0, b_0) , at least one new marking of \mathcal{M}_R is forbidden. Because \mathcal{M}_R is finite, after a finite number of iterations all new siphons (if any) considered in the step 2(b) of the procedure are implicitly controlled, and so the procedure terminates. \square

The usage of the procedure with the approach B can be summarized as follows:

- Find a set of constraints $L_i\mu \geq b_i$ with bounded feasible set F , where μ is a nonnegative integer vector, such that for all initial markings μ_0 of \mathcal{N} which are of interest: $\mathcal{R}(\mathcal{N}, \mu_0) \subseteq F$. Let \mathcal{M}_I be the set of initial markings of interest.

- Use the procedure with initial constraints (L_0, b_0) which equal (L_i, b_i) .
- The supervisor can be used for the initial markings $\mu_0 \in \mathcal{M}_I$ which satisfy $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$, where (L, b) and (L_0, b_0) are the two sets of constraints generated by the procedure.

Unlike in approach A, both Theorem 6.2 and Theorem 6.3 apply. The disadvantage of approach B is that the formal result which guarantees termination is somewhat weak, as it may be hard to know whether the assumption on transition splits holds true.

6.3 Special Cases

6.3.1 Additional Constraints

We consider the case when additional constraints are to be enforced. Let (L_a, b_a) be the additional constraints and \mathcal{N} the Petri net. A good way to proceed with the deadlock prevention procedure is to apply it rather to the supervised Petri net \mathcal{N}_L , which contains the additional places necessary to enforce (L_a, b_a) according to the invariant based approach ([24], also outlined in section 4.2). So the procedure would start with $\mathcal{N}_0 = \mathcal{N}_L$ and initial constraints (L_0, b_0) reflecting the invariants associated to enforcing (L_a, b_a) .

The reason why it is *not* a good idea to apply the deadlock prevention procedure first to \mathcal{N} and then to enforce (L_a, b_a) is that additional constraints can make deadlock possible. Indeed, we can easily find examples of deadlock-free Petri nets which with additional marking constraints may reach deadlock.

6.3.2 Finite Capacity Petri Nets

In many applications it is reasonable to assume that the maximum number of tokens that a place may have is bounded. In this case the Petri nets may be extended with an additional function K which maps its capacity to each place. This type of Petri net is called place/transition net [28]. So, a **place/transition structure** is represented by the quintuple $\mathcal{N} = (P, T, F, W, K)$, where $K : P \rightarrow \overline{\mathbb{N}}$ is the **capacity function**, and with an additional initial marking we have a **place/transition net**, denoted by (\mathcal{N}, μ_0) . The capacity of a place is allowed to be infinite. The firing rule of a transition in place/transition nets is the same as for conventional Petri nets, except that a transition is not enabled by a marking if firing it would cause a place to exceed its capacity.

Let $\mathcal{N} = (P, T, F, W, K)$ be a place/transition structure and $\mathcal{N}_R = (P, T, F, W)$ the corresponding Petri net structure. \mathcal{N} can be transformed in an equivalent conventional Petri net \mathcal{N}_E by enforcing in \mathcal{N}_R , to each place p with finite capacity, the linear constraint $\mu(p) \leq K(p)$. The conventional Petri net is obtained using the invariant based approach of [24], outlined also in section 4.2.

If all the places have finite capacity, the equivalent Petri net is by construction structurally bounded. The deadlock prevention procedure can be started as in section 6.3.1, with constraints (L_a, b_a) describing $\mu(p) \leq K(p)$ for all $p \in P$. The method can be guaranteed to terminate as shown in section 6.2.2, since a bound on the marking of each place is known. Indeed, the upper bound for the marking of any place $p \in P$ is the finite capacity $K(p)$ and the upper bound for the marking of a control place p_c enforcing for a place $p \in P$ the constraint $\mu(p) \leq K(p)$, is also $K(p)$.

6.3.3 Safe Petri Nets

An ordinary Petri net (\mathcal{N}, μ_0) is **safe** if for all reachable markings the marking of any place is at most 1. We consider the case when a Petri net \mathcal{N} needs to be made safe by supervision. The deadlock prevention

procedure may be used to provide such a policy which is not blocking.

Let (L_a, b_a) be the constraints associated to $\mu(p) \leq 1$, for all places of \mathcal{N} . Then we can proceed as shown in section 6.3.1.

The deadlock prevention procedure is guaranteed to terminate when one of the approaches of section 6.2.2 is used, because it is known that 1 is an upper bound of the marking of each place.

6.3.4 Some Particular Cases when Liveness is also Enforced

It is possible that if the initial Petri net is an asymmetric choice net the final Petri net still will be an asymmetric choice net. By Theorem 3.1, this is a sufficient condition for liveness for all valid initial markings.

Both parts of Corollary 3.2(c) are useful for the deadlock prevention procedure. The second part applies because of Corollary 6.3. Corollary 3.2(c) provides conditions that let us know before applying the procedure whether the supervisor also will enforce liveness. In the case of asymmetric choice net result, we need first to run the procedure, and then check whether the final result complies with Theorem 3.1.

The conditions of Corollary 3.2(c)(i) can be easily checked using a similar procedure to that which computes an active subnet. It is not clear at this time if the conditions of Corollary 3.2(c)(ii) have practical importance. It depends on whether or not there is an efficient procedure to check them.

The class of Petri nets on which the procedure enforces liveness may be larger than that resulting from Corollary 3.2(c), because the class of deadlock prevention supervisors more permissive than liveness enforcing supervisors is rather large.

Note that whenever the supervisor provided by the procedure enforces liveness, it is the maximally permissive supervisor, by Corollary 6.3.

6.4 Final Remarks and Directions for Further Research

6.4.1 Faster Convergence For Nonrepetitive Petri Nets

When a loss of permissivity in firing the transitions which cannot be made live is acceptable enough, the procedure can be applied to the active subnet \mathcal{N}_0^A rather than to the total target net \mathcal{N}_0 . Considering this modification, the constraints obtained for \mathcal{N}_0^A are used for the supervision of \mathcal{N}_0 . The benefit of this approach results from the fact that generally, the number of minimal siphons which appear by approaching \mathcal{N}_0^A may be sensibly smaller than the number of minimal active siphons considered when approaching \mathcal{N}_0 . Thus faster convergence can be obtained.

6.4.2 The Termination Problem

6.4.2.1 Converging Constraints Theorem 6.4 shows how we can guarantee the termination of the procedure in the case of structurally bounded Petri nets. The termination of the procedure is facilitated by considering only minimal siphons that are not *implicitly controlled* (see section 5.2). For instance, the procedure does not terminate for the Petri net of figure 11 if implicitly controlled siphons are not eliminated. However this operation does not guarantee termination in general. For instance, if in figure 11 we change the weight of (t_2, p_1) to 2, the procedure does not terminate, failing to generate one of the good constraints. Instead it generates a sequence of constraints converging to that constraint. When $W(t_2, p_1) = 1$ that good constraint is generated from a siphon appearing in iteration 2, which does not appear for $W(t_2, p_1) = 2$, and which allows to consider as controlled the siphon that generates the recurrent behavior.

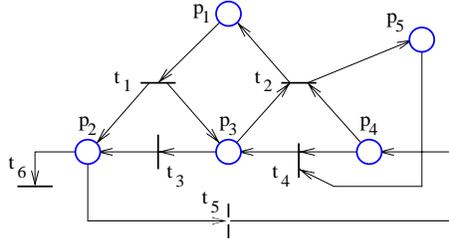


Figure 11: Example for the termination problem

Checking whether a siphon is implicitly controlled is equivalent to an integer programming feasibility problem, which is an *NP* type problem [35].

6.4.2.2 Nonconvex Feasible Sets We consider a set $F \subseteq \mathbb{N}^k$ to be convex if any convex combination of elements of F which is in \mathbb{N}^k also is in F . In other words, $F \subseteq \mathbb{N}^k$ is convex if $\forall n \geq 2, \forall x_1, x_2, \dots, x_n \in F$ and $\forall \alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{R}_+$ such that $\sum_{i=1}^n \alpha_i = 1$, if $y = \sum_{i=1}^n \alpha_i x_i$ and $y \in \mathbb{N}^k$, then $y \in F$. By using theorems 6.2 and 6.3, we can see that when a convex combination of markings for which liveness is enforceable produce a deadlock marking, the procedure cannot terminate. Indeed the feasible region of a set of linear inequalities is a convex set, so the method cannot converge to a set of constraints satisfying both theorems 6.2 and 6.3.

We show two examples in Figure 12(a) and (b). In case (a), the Petri net is live for the markings $[2, 0]$ and $[0, 2]$, but not for $[1, 1]$. The set of markings for which liveness is enforceable equals the set of markings for which the Petri net is live, which is not convex. In case (b), which corresponds to the PT-transformation of (a), deadlock can occur. Preventing deadlock is equivalent to enforcing liveness (by Theorem 3.2(c-i)), and deadlock can be prevented for the markings $[2, 0, 0, 0]$ and $[0, 2, 0, 0]$, but not for $[1, 1, 0, 0]$. In both cases (a) and (b) the method cannot terminate. For instance, in case (a), after 3 iterations (figure 13), the subnet containing $C_2, p_{1,1}, C_3$ and $p_{2,1}$ is similar to the target net (a), and this generates a cyclic behavior which leads to divergence.

A solution to avoid this type of problem is to improve the procedure as follows:

1. For all places p , let $M(p) = \{x : \exists t \in \bullet p : W(t, p) = x \text{ or } \exists t \in p \bullet : W(p, t) = x\}$.
2. Let d be the greatest common divisor of $M(p)$. If $d > 1$, then the following changes are made: (a) all weights of the arcs connected to p are divided by d ; (b) in all constraints, replace $\mu(p)$ by $\lfloor \mu(p)/d \rfloor$.

In this way we obtain more sets of linear inequalities, rather than just one for all markings. Given an initial marking μ_0 , we obtain the constraints (L, b) by replacing $\lfloor \mu(p)/d \rfloor$ with $\mu(p)/d + \lfloor \mu_0(p)/d \rfloor - \mu_0(p)/d$. We see, L does not depend on μ_0 , but b does.

However this solution is not applicable for the structurally unbounded Petri net structure shown in Figure 12(c). We can easily see that there are initial markings for which enforcing deadlock-freedom with a convex set of allowed markings conflicts with being more permissive than any liveness enforcing supervisors. Indeed, from the marking $\mu_0 = [2, 0, 0, 0]$, both $\mu_1 = [0, 2, 0, 0]$ and $\mu_2 = [1, 1, 0, 0]$ are reachable. (μ_2 is reached by firing t_1, t_5 and t_6 .) Because for μ_0 and μ_1 liveness is enforceable and $\mu_2 = 0.5\mu_0 + 0.5\mu_1$ is a deadlock marking, the procedure cannot terminate.

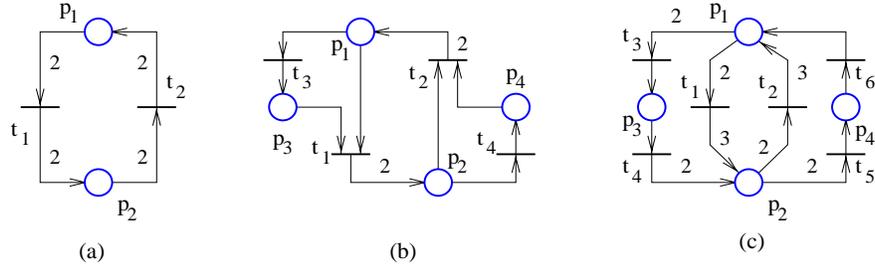


Figure 12: Examples for section 6.4.2.2

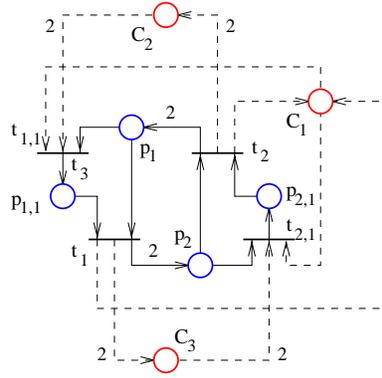


Figure 13: Example for section 6.4.2.2

6.4.3 Maximally Permissive Deadlock Prevention

Most applications do not need a maximally permissive deadlock prevention supervisor. Indeed, it is generally desired that all local deadlock is prevented, not that only part of the net is not deadlocked. An example in which our deadlock prevention procedure is not maximally permissive with regard to deadlock prevention is in figure 14(a). The constraints $L\mu \geq b$ and $L_0\mu \geq b_0$ provided by the procedure are

$$L = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (27)$$

$$L_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad b_0 = \begin{bmatrix} 2 \end{bmatrix} \quad (28)$$

Maximally permissive deadlock prevention is described by $L\mu \geq b$, where:

$$L = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \end{bmatrix} \quad (29)$$

Often a maximally permissive deadlock prevention supervisor cannot be defined just as a conjunction of linear inequalities. For instance consider the Petri net of figure 14(b). Both marking vectors $\mu_1 = [0, 0, 2, 0, 0, 0, 0]$ and $\mu_2 = [0, 0, 0, 2, 0, 0, 0]$ are acceptable for deadlock prevention, but $\mu_3 = 0.5\mu_1 + 0.5\mu_2$ is not.

Note that maximally permissive deadlock prevention is achieved with the following procedure:

Procedure MaxPDP

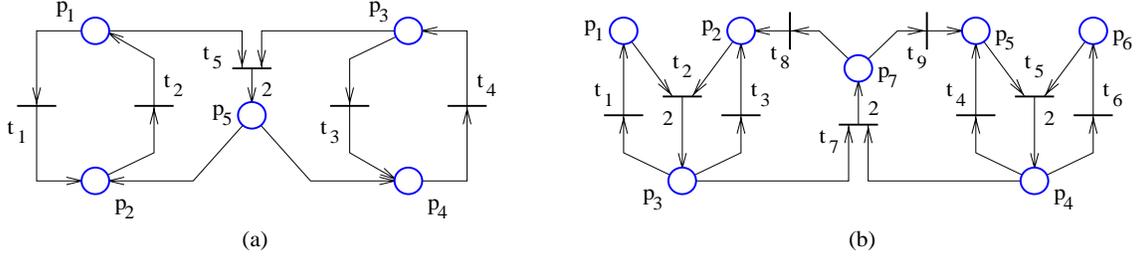


Figure 14: (a) Example in which the deadlock prevention procedure is not maximally permissive with respect to deadlock prevention; (b) a similar example in which the limitation is due to the fact that a disjunction of inequalities is required

1. Let $\mathcal{N}_{0,1}^A, \mathcal{N}_{0,2}^A \dots \mathcal{N}_{0,u}^A$ be the minimal active subnets of the target Petri net \mathcal{N}_0 .
2. Apply u times the deadlock prevention procedure (section 5.4) for \mathcal{N}_0 , each time starting with a different minimal active subnet as the initial active subnet of the procedure.
3. Let $L_{0,1}\mu \geq b_{0,1}, L_{0,2}\mu \geq b_{0,2}, \dots, L_{0,u}\mu \geq b_{0,u}$ and $L_1\mu \geq b_1, L_2\mu \geq b_2, \dots, L_u\mu \geq b_u$ be the constraints obtained.
4. Consider the following supervisor:
 - (a) An initial marking μ_0 is allowed if $I_u(\mu_0) = \{j = 1 \dots u : L_{0,j}\mu_0 \geq b_{0,j} \text{ and } L_j\mu_0 \geq b_j\}$ is nonempty.
 - (b) Given the initial marking μ_0 , let μ be the current marking, t a transition of the target net \mathcal{N}_0 which is enabled by μ and μ' the marking reached by firing t . The supervisor also enables t if $\exists j \in \{1, \dots, u\}$ such that $L_{0,j}\mu \geq b_{0,j}, L_j\mu \geq b_j$ and $L_j\mu' \geq b_j$.

Theorem 6.6 *The procedure MaxPDP enforces maximally permissive deadlock prevention if each time the usual deadlock prevention procedure is applied in step 2, the conditions of both Theorem 6.2 and Theorem 6.3 are satisfied.*

Proof: Deadlock is prevented since for all reachable markings at least one of the supervisors for $\mathcal{N}_{0,1}, \mathcal{N}_{0,2} \dots \mathcal{N}_{0,u}$ can be used, and each of them prevents deadlock. Next we need to prove that a marking unacceptable to the supervisor leads to deadlock. Assume such a marking μ . Let $x_1, x_2 \dots x_u$ be nonnegative integer vectors defining the minimal active subnets (see the definition.) The proof of Theorem 6.3 applies without change if we replace in its statement *maximal active subnet* with *initial active subnet*, where the latter refers to the case when the procedure starts with an active subnet different than the maximal active subnet. Therefore we can apply Theorem 6.3 to infer that given μ , for all $i = 1 \dots u$, not all transitions in $\|x_i\|$ can be made live. If μ is not be a deadlock marking, there is a set of transitions T_x which can fire infinitely often. Therefore there is a nonzero nonnegative integer vector x such that $Dx \geq 0$ and $\|x\| \subseteq T_x$ (by Lemma 3.1 and Theorem 3.2, where D is the incidence matrix.) It is not possible that $\|x\| \subseteq \|x_i\|$, as $\mathcal{N}_{0,i}^A$ is minimal. Moreover, because we consider all minimal active subnets, there is $j \in \{1, \dots, u\}$ such that $\|x_j\| \subseteq \|x\|$. But this contradicts that all transitions of T_x can be made live given μ . Hence only markings for which deadlock is unavoidable are forbidden. \square

7 Summary of Results

We introduce new theoretical results on the supervision of Petri nets for deadlock prevention and a new deadlock prevention procedure for Petri nets. The general theoretical results are given in section 3:

- Corollary 3.2(c) provides sufficient conditions for deadlock prevention methods to enforce liveness. It applies to the deadlock prevention procedure which we propose.
- Corollary 3.3 proves that for any nonrepetitive Petri net structure there is a set of transitions which cannot be made live under any circumstances. This result is important for defining the active subnets of a Petri net.
- Proposition 3.5 is a generalization of a classic result. The generalization is especially important for nonrepetitive Petri nets.
- Proposition 3.6 is the basis for maximally permissive deadlock prevention.

The deadlock prevention procedure has been stated in section 5.4. Variations of this procedure have been given in section 6.2.2 and 6.4.3. The procedure has the following characteristics:

- Given a Petri net structure, the procedure generates two sets of linear constraints (L_0, b_0) and (L, b) , such that for all initial markings μ_0 which satisfy $L_0\mu_0 \geq b_0$ and $L\mu_0 \geq b$, the Petri net in closed loop with the supervisor enforcing $L\mu \geq b$ is deadlock free.
- No assumptions are made on the Petri net structure. The method is effective for the Petri nets generally considered in the deadlock prevention literature, as well as for those which may be generalized, unbounded, nonrepetitive and with uncontrollable and unobservable transitions.
- The user is allowed to specify initial constraints in the form of initial constraints in (L_0, b_0) . In this way the procedure knows that only markings such that $L_0\mu \geq b_0$ are used. Using initial constraints benefits problems in which one of the following is true: (a) the procedure should not generate constraints requiring $L_0\mu \not\geq b_0$, (b) permissivity can be compromised to reduce the complexity of the supervisor (for instance by using certain place invariants in the structure of the target Petri net) (c) convergence help is needed.

The main results concerning the deadlock prevention procedure are proved in section 6.2. The fact that uncontrollable and unobservable transitions are allowed affects the permissivity related results. These results are proved for a restricted class of Petri nets with uncontrollable and/or unobservable transitions.

- In the conditions of Theorem 6.2, deadlock is prevented.
- The case when the structure of \mathcal{N}_0 does not allow deadlock to be prevented is detected in Proposition 6.15.
- Theorem 6.3 shows that the procedure is no more restrictive than any supervisor which enforces that all the transitions of the maximal active subnet are live in the target net, where the transitions which can be made live for some marking are in the maximal active subnet.
- There are particular cases in which the supervisor generated by the procedure also enforces liveness. When this is true, in the conditions of Theorem 6.3, the supervisor is a maximally permissive liveness enforcing supervisor. Such particular cases are identified by Corollary 3.2(c).

- Two approaches have been proposed to help the procedure terminate. Theorem 6.4 and Theorem 6.5 guarantee termination for the two approaches.
- A variant of the procedure is given in section 6.4.3. Theorem 6.6 guarantees maximally permissive deadlock prevention for this variant.

APPENDIX

A Bounding the reachable space with a polytope¹

We have considered using initial constraints to help the procedure terminate, and we have shown that of special interest are the initial constraints describing a polytope which bounds the reachable space of the PN. Here we consider structurally bounded PNs and present an algorithm to bound the reachable space with a polytope. Assume that the initial markings of interest are given by the polytope

$$L_z \mu_0 \leq b_z \quad (30)$$

The state equation of the PN is

$$\mu = \mu_0 + Dq \quad (31)$$

where q is the firing count vector and D the incidence matrix. The intersection of (30), $q \geq 0$, and $\mu_0 + Dq \geq 0$ is a polytope for structurally bounded PNs, and projecting it on the variable q yields a set of inequalities

$$Cq \leq d \quad (32)$$

Then, by projecting the intersection of (32), $\mu \geq 0$, and $L_z \mu - L_z Dq \leq b_z$ on μ we obtain

$$L_I \mu \geq b_I \quad (33)$$

which is the desired polytope.

B A Method to Obtain an Admissible Siphon Constraint

1. Find the maximum support of $l \geq 0$ such that $l_i = 0$ for $p_i \notin S$ and

$$\begin{aligned} l^T D_{uc} &\geq 0 \\ l^T D_{uo} &= 0 \end{aligned}$$

A possible method to find the maximum support is in the appendix of [15].

2. Let l^0 be a vector l of maximum support and e a vector of the same size such that $e_i = 1$ if $l_i^0 \neq 0$ and $e_i = 0$ otherwise. Solve the linear program

$$\begin{aligned} \min_l \quad & e^T l \\ & l \geq e \\ & l^T D_{uc} \geq 0 \\ & l^T D_{uo} = 0 \end{aligned}$$

¹Appendix A has been added to the report at the November 2001 revision.

3. Let l_x be a solution of the linear program. Any solution provided by the computer is rational, so let l_y be l_x multiplied with the least common denominator of its entries. Then $l_y \mu \geq 1$ is an admissible form of $\sum_{p_i \in S} \mu(p_i) \geq 1$.

This method might give as answer relatively large numbers, so it might be more convenient to replace steps 2 and 3 with a modified step 2 in which we have the additional constraint that l should be an integer vector (that is, we have a linear integer program instead of a linear program.)

References

- [1] Banaszak Z., B. Krogh, "Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows" in *IEEE Trans. on Robotics and Automation*, 6(6).
- [2] Barkaoui, K., I. Abdallah, (1995) "Deadlock Avoidance in FMS Based on Structural Theory of Petri Nets," *IEEE Symposium on Emerging Technologies and Factory Automation*.
- [3] Barkaoui, K., J.-F. Pradat-Peyre, (1996) "On Liveness and Controlled Siphons in Petri Nets," in *Application and Theory of Petri Nets*, Springer Verlag.
- [4] Boer E, T. Murata, (1994) "Generating Basis Siphons and Traps of Petri Nets Using the Sign Incidence Matrix," *IEEE Trans. on Circuits and Systems*, 41(4).
- [5] Coffman E., Elphick M., Shaoshani A., (1971) "System Deadlocks," *Computing Surveys*, vol. 3, pp.67-68, June 1971.
- [6] Commoner F., (1972) *Deadlocks in Petri nets*, Applied Data Research Inc., Wakefield, Massachusetts 01880, Report Nr. CA-7206-2311, 1972.
- [7] David R., A. Hassane, (1994) "Petri Nets for Modeling of Dynamic Systems – A Survey," in *Automatica*, 32(2).
- [8] Dijkstra E., (1965) "Cooperating Sequential Processes," in *Programming Languages*, Genuys F. editor, London, Academic Press, 1965.
- [9] Ezpeleta J., J. Couvreur, M. Silva, (1993), "A New Technique for Finding a Generating Family of Siphons, Traps and ST-Components. Application to Colored Petri Nets," in *Advances in Petri Nets, Lecture Notes in Computer Science*, Springer-Verlag 1993.
- [10] Ezpeleta J., J. Colom, J. Martinez, (1995) "A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems," *IEEE Trans. on Robotics and Automation*, 11(2).
- [11] Fanti M., B. Maione, S. Mascolo, B. Turchiano, (1997) "Event-Based Feedback Control for Deadlock Avoidance in Flexible Production Systems," in *IEEE Trans. on Robotics and Automation*, 13(3).
- [12] Giua A., F. DiCesare, M. Silva, (1992) "Generalized Mutual exclusion Constraints on Nets with Uncontrollable Transitions," in *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*.
- [13] Hack M., (1972) *Analysis of Production Schemata by Petri Nets*, Technical Report 94, Project MAC.

- [14] Iordache M., (1999) *Deadlock Prevention in Discrete Event Systems Using Petri Nets*, Master's Thesis, University of Notre Dame.
- [15] Iordache M., J. Moody, P. Antsaklis (1999) *A Method for Deadlock Prevention in Discrete Event Systems Using Petri Nets*, Technical Report of the ISIS Group, ISIS-99-006, University of Notre Dame, available at <http://www.nd.edu/~isis/tech.html>.
- [16] Iordache M., J. Moody, P. Antsaklis (2000) "A Method for the Synthesis of Deadlock Prevention Controllers in Systems Modeled by Petri Nets," in the *Proceedings of the 2000 American Control Conference*.
- [17] Krogh B., (1987) "Controlled Petri Nets and Maximally Permissive Feedback Logic," in *Proceedings of 25th Annual Allerton Conference*, University of Illinois, Urbana.
- [18] Lautenbach K., (1987) "Linear Algebraic Calculation of Deadlocks and Traps," in *Concurrency and Nets*, Springer-Verlag 1987.
- [19] Lautenbach K., H. Ridder, (1994) "Liveness in Bounded Petri Nets which are Covered by T-Invariants," in *Applications and Theory of Petri Nets, Lecture Notes in Computer Science*, p. 358-375, Springer-Verlag 1994.
- [20] Lautenbach K., H. Ridder, (1996) "The Linear Algebra of Deadlock Avoidance — A Petri Net Approach," Research Report at Institute for Computer Science, University of Koblenz, Germany.
- [21] Lewis F., H. Huang, D. Tacconi, A. Gürel, O. Pastravanu, (1998) "Analysis of Deadlocks and Circular Waits Using a Matrix Model for Discrete Event Systems," *Automatica*, 34(9).
- [22] Moody J., K. Yamalidou, M. Lemmon, P. Antsaklis, (1994) "Feedback Control of Petri Nets Based on Place Invariants," in *Proc. of the 33rd IEEE Conf. on Decision and Control*.
- [23] Moody, J. P. Antsaklis, (1996) "Supervisory Control of Petri Nets with Uncontrollable/Unobservable Transitions," in *Proceedings of the 35th IEEE Conference on Decision and Control*, pp. 4433-4438, Kobe, Japan, December 1996.
- [24] Moody, J., P. Antsaklis, (1998) *Supervisory Control of Discrete Event Systems Using Petri Nets*, Kluwer Academic Publishers.
- [25] Moody, J., P. Antsaklis, (1998b) "Deadlock Avoidance Using the Supervisory Enforcement of Linear State Constraints on Petri Net Plants", Technical report, Univ. of Notre Dame, May 1998.
- [26] Moody, J., P. Antsaklis, (2000) "Petri Net Supervisors for DES with Uncontrollable and Unobservable Transitions" *IEEE Trans. on Automatic Control*, 45(3).
- [27] Murata, T. (1989) "Petri Nets: Properties, Analysis and Applications," in *Proc. of the IEEE*, 77(4).
- [28] Reisig, W. (1985) *Petri Nets* Springer Verlag, 1985.
- [29] Reveliotis S., M. Lawley, (1997) "Polynomial-Complexity Deadlock Avoidance Policies for Sequential Resource Allocation Systems," *IEEE Trans. on Automatic Control*, 42(10).
- [30] Sreenivas R., (1997) "On the Existence of Supervisory Policies that Enforce Liveness in Discrete Event Systems Modeled by Controlled Petri Nets," in *IEEE Trans. on Automatic Control*, 42(7).

- [31] Sreenivas R., (1998) “An Application of Independent, Increasing, Free-Choice Petri Nets to the Synthesis of Policies that Enforce Liveness in Arbitrary Petri Nets,” in *Automatica*, 34(12), pp. 1613-1615.
- [32] Sreenivas R., (1999) “On Supervisory Policies that Enforce Liveness in in a Class of Completely Controlled Petri Nets obtained via Refinement,” in *IEEE Transactions on Automatic Control*, Vol. 44, No. 1, January, 1999.
- [33] Suraj Z., (1980) “A Resource Allocation Problem” in *Math. Found. of Comp. Sci.*, Springer Verlag.
- [34] Walukiewicz S., (1991) *Integer Programming*, Kluwer Academic Publishers.
- [35] Wolsey L., (1998) *Integer Programming*, New York: John Wiley & Sons.
- [36] Yamalidou K., J. Moody, M. Lemmon, P. Antsaklis, (1994) “Feedback control of Petri nets based on place invariants,” Technical Report of the ISIS Group ISIS-94-002.2, University of Notre Dame, May 1994.
- [37] Yamalidou K., J. Moody, M. Lemmon, P. Antsaklis, (1996) “Feedback control of Petri nets based on place invariants,” in *Automatica*, 32(1) .