Automated Synthesis of Liveness Enforcing Supervisors Using Petri Nets

Technical Report of the ISIS Group
at the University of Notre Dame
ISIS-00-004
October, 2000
Revised in January 2001 and May 2002

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 LIVENESS ENFORCING 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 liveness enforcement 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 live for all initial markings that satisfy the supervision constraints. Also the supervision is proved to be maximally permissive for a large class of Petri nets, which includes the fully controllable and observable Petri nets. Moreover, the supervisor supports specifications requiring only some of the Petri net transitions to be live. The maximal permissivity typically applies also for this case. The procedure allows automated synthesis of the supervisors. The sufficient conditions for which our theoretical results are guaranteed to apply can be automatically verified.

Contents

1 Introduction **1**
- 1.1 Scope of the Paper . . . 1
- 1.2 Related Work . . . 2
- 1.3 Paper Structure . . . 3

2 Review of Some Petri Net Basic Properties **4**

3 Deadlock and Liveness Properties of Petri Nets **5**
- 3.1 Intrinsic Properties . . . 5
- 3.2 Conditions for Deadlock Prevention and Liveness Enforcement . . . 6
- 3.3 A Characterization of Petri Nets Based on Subnets which Can Be Made Live, in View of Deadlock Prevention and Liveness Enforcement . . . 10

4 Preliminaries to the Liveness Enforcing Method **13**
- 4.1 A Transformation of Petri Nets to PT-ordinary Petri Nets . . . 13
- 4.2 Transformation of Petri nets to asymmetric choice Petri nets . . . 14
- 4.3 Petri Net Supervisors Based on Place Invariants . . . 15
  - 4.3.1 Fully Controllable and Observable Petri Nets . . . 15
  - 4.3.2 Petri Nets with Uncontrollable and Unobservable Transitions . . . 16

*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)

4.4 Siphon Control Based on Place Invariants . . . 17
4.4.1 Case 1: All Transitions are Controllable and Observable . . . 17
4.4.2 Case 2: Transitions Uncontrollable and/or Unobservable are Present . . . 18

5 The Liveness Enforcing Method 19
5.1 Introduction to the Method . . . 19
5.2 Generating Marking Constraints . . . 20
5.2.1 The enforced place invariants . . . 20
5.2.2 Constraints which do need control place enforcement . . . 21
5.2.3 Constructing the constraints of (L, b) and (L_0, b_0) . . . 21
5.2.4 Implicitly controlled siphons . . . 22
5.2.5 Initial constraint transformation . . . 22
5.2.6 Transforming Constraints to Admissible Constraints . . . 22
5.3 The Computation of a T-minimal Active Subnet . . . 25
5.4 The Liveness Enforcing Procedure . . . 26
5.5 Remarks . . . 28
5.6 Illustrative Examples . . . 28

6 Properties 32
6.1 Basic Properties of the Method . . . 32
6.1.1 Introduction and Notations . . . 32
6.1.2 Properties . . . 34
6.2 Main Results . . . 43
6.2.1 Success and Permissivity Results . . . 43
6.2.2 Extending Permissivity . . . 47
6.2.3 Termination Results . . . 48
6.3 Final Remarks and Directions for Further Research . . . 50
6.3.1 Additional Constraints . . . 50
6.3.2 Finite Capacity Petri Nets . . . 50
6.3.3 The Termination Problem . . . 51

7 Summary of Results 52

1 Introduction

1.1 Scope of the Paper

Liveness is a desirable quality of concurrent systems. Due to mutual interdependencies, such systems may reach states of local or total deadlock. Deadlock means that some actions (or all, for total deadlock) are impossible to pursue. A system is live when deadlock (both local and global) is impossible. Rather than providing a method to verify whether a system is live, we provide a method which synthesizes a supervisor such that the supervised system is live. We consider discrete event systems modeled as Petri nets. We note that it is more natural to model concurrent systems as Petri nets rather than as finite automata. Further on, we do not restrict the Petri net models. They are allowed to be unbounded, generalized and

with uncontrollable and unobservable transitions. The approach is not dependent on the initial marking. Instead, the set of initial markings for which liveness is enforced is characterized as the feasible set of a system of linear marking inequalities. Thus the liveness supervisor produced by our approach is defined for a set of initial markings, rather than for a single initial marking. Moreover, when the supervisor is maximally permissive, enforcing liveness is impossible for all initial markings for which the supervisor is not defined. Thus the method can also be used for liveness verification.

In some applications, some Petri net transitions may model undesirable behavior. Such transitions should not be live. Thus, rather than supervising for liveness, we supervise for T-*liveness*, which is a concept generalizing liveness: instead of requiring all transitions to be live, only the transitions in the set T are to be live. Note that liveness is a special case of T-liveness. The liveness enforcing procedure can be described as follows. Given a fully controllable and observable Petri net and a set T, the procedure will provide a supervisor for T-liveness which typically is maximally permissive, or it will detect that T-liveness is impossible, in which case the supervisor will enforce T_x-liveness, for $T_x \subseteq T$. Given a Petri net with uncontrollable and/or unobservable transitions and a set T, the procedure will provide a supervisor enforcing T_x-liveness, $T_x \subseteq T$; if $T_x \neq T$ then the procedure is unable to enforce T-liveness. A sufficient condition which guarantees the supervisor to be maximally permissive can be easily tested. In particular, in the case of fully controllable and observable Petri nets, the supervisor is guaranteed to be maximally permissive in the case of liveness enforcement.

The disadvantages of our procedure are that termination is not guaranteed and that when the procedure terminates the computations may be complex. However all computations are performed offline. Thus the supervisor generated by the procedure is appropriate for real-time problems. It is possible to guarantee termination, but this may come at the expense of permissivity. We give two variants of the procedure with guaranteed termination. However these two variants are only useful for bounded Petri nets.

The method presents the conditions necessary to insure liveness enforcement as a set of linear marking inequalities. This feature can be used directly in optimization problems, e.g., a linear program can be used to determine the minimum number of resources a system requires such that deadlock can be avoided.

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 approaches which consider a given initial marking and a bounded Petri net. Applications which may benefit from considering the initial marking unknown are in the area of Flexible Manufacturing, as the initial marking corresponds to the number of available resources.

1.2 Related Work

Previous results about enforcing liveness in Petri nets usually consider restricted classes of Petri nets. A necessary and sufficient condition for the existence of liveness supervisors appears in [18]. A method for liveness enforcement in a class of conservative ordinary Petri nets has been given in [5]; the approach is not maximally permissive. The approach of [5] has been recently extended to generalized Petri nets in [15]. Polynomial complexity has been proved, however the considered Petri nets are conservative and the approach is not maximally permissive. A liveness enforcing approach for a restricted class of ordinary Petri nets is given in [19]. Another liveness enforcing approach appears in [20]; it is based on the coverability graph, and hence the initial marking is required. In [7] the authors consider enforcing liveness based on the unfolding of a Petri net. Unfolding is an efficient technique of searching the reachability graph. The approach of [7]

is limited to bounded Petri nets and the initial marking must be known. Our approach is most related to the deadlock prevention procedure we presented in [10, 8], and its improvement in [9]. While our former procedure prevented deadlock but was not guaranteed to enforce liveness, the procedure of this paper is guaranteed to enforce liveness.

The liveness enforcement procedure of this report is iterative, at every iteration correcting new deadlock situations. Using iterations to correct deadlock situations has also been used in [12]. In our procedure we employ supervisory control based on place invariants [13, 22], which is an established method in the supervisory control of Petri nets. We also use a transformation to almost ordinary Petri nets and a transformation to asymmetric choice nets. The first transformation was inspired by a similar transformation in [12]. A transformation to free choice nets, which is a particular class of asymmetric choice nets, has been used in [17]. In [17] it is shown that liveness enforcing policies of a free choice equivalent of a Petri net can be used to enforce liveness in the original Petri net. Our interest for asymmetric choice nets stems from a generalization of the Commoner's Theorem for asymmetric choice nets [2]. Thus liveness in an asymmetric choice Petri nets can be related to siphon control. Our approach involves the computation of a special class of minimal siphons. Methods of siphon computation have been given for instance in [11, 3, 6].

1.3 Paper Structure

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 we derive Corollary 3.2 which is the basis for better deadlock tests, such as Proposition 3.4 and Proposition 3.5. In Theorem 3.2 we prove a fundamental result for our method. A consequence of Theorem 3.2 is Proposition 3.6, which gives a necessary condition and a sufficient condition for T-liveness in a class of Petri nets. In section 4 we present preliminaries to our methodology. The supervisory technique used by our method, supervisory control based on place invariants [13, 22], is outlined in section 4.3. Transformations from generalized Petri nets to ordinary Petri nets and to asymmetric choice Petri nets are presented in sections 4.1 and 4.2. The siphon control approach (largely a particular case of the supervision based on place invariants) is given in section 4.4. Section 5 defines the liveness enforcement procedure and the operations which are involved. The procedure for liveness enforcement is stated 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 properties characterizing the procedure or the operations involved by it. These properties are also used to derive our main results. The main results are given in section 6.2. Theorem 6.2 proves that the procedure does enforce liveness and Theorem 6.3 proves that for a large class of Petri nets the procedure is not more restrictive than any other liveness enforcing supervisor. Section 6.2.3 contains results that show that by (possibly) compromising the performance of the procedure, termination can be guaranteed. We conclude with some significant special cases and remarks in section 6.3.

This is an almost self-contained report. The liveness enforcement procedure of this report is essentially a modification of the deadlock prevention method presented in our previous technical report [9]. Thus a significant part of the material of [9] resembles or is included in this report. The new material of this report is included in the following sections. Section 3.3 includes Theorem 3.2, which is essential for our liveness enforcement approach. Section 4.2 describes the asymmetric choice transformation. Section 5 is the adaptation for liveness enforcement of the similar section in [9], except for section 5.3, which contains new

material. Compared to [9], some results of section 6 needed new proofs or restatements due to the new form of the procedure; also, new technical results have been added. The main theoretical results for the liveness enforcing procedure are given in section 6.2, where the termination results are essentially the same as in [9].

2 Review of Some Petri Net Basic Properties

We assume the reader to be familiar with Petri net fundamentals. Petri net surveys may be found in [16], [14] and [4]. In this section we introduce our conventions and 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 $W : F \to \mathbb{N} \setminus \{0\}$ is a **weight function**. A **marking** μ of the Petri net structure is a map $\mu : P \to \mathbb{N}$. A Petri net structure $\mathcal{N}$ with **initial marking** μ_0 is called a **Petri net**, and will be denoted by $(\mathcal{N}, \mu_0)$. 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 vectors can be seen as maps defined on an arbitrary finite set domain [16], 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 outcoming 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 $\mu_1, \dots \mu_k$, $\mu_k = \mu'$, 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 $(\mathcal{N}, \mu)$ (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

$d^+_{ij} = W(t_j, p_i)$ if $(t_j, p_i) \in F$ and $d^+_{ij} = 0$ otherwise;

$d^-_{ij} = W(p_i, t_j)$ if $(p_i, t_j) \in F$ and $d^-_{ij} = 0$ otherwise.

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

$$\mu_k = \mu_{k-1} + D \cdot u_k \tag{1}$$

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:

$$\mu_k = \mu_0 + D \cdot x \tag{2}$$

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 $(\mathcal{N}, \mu_0)$ is said to be **deadlock-free** if for any reachable marking μ there is an enabled transition. $(\mathcal{N}, \mu)$ is in **deadlock** if no transition is enabled at marking μ.

Let $(\mathcal{N}, \mu_0)$ 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. $(\mathcal{N}, \mu_0)$ 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 $\forall f \in F : W(f) = 1$. We will 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 methodology of our work depends on a well known necessary condition for deadlock [16], 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.

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

An example is shown in figure 1(a). Proposition 3.1 shows that deadlock might be prevented if it can be ensured 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.

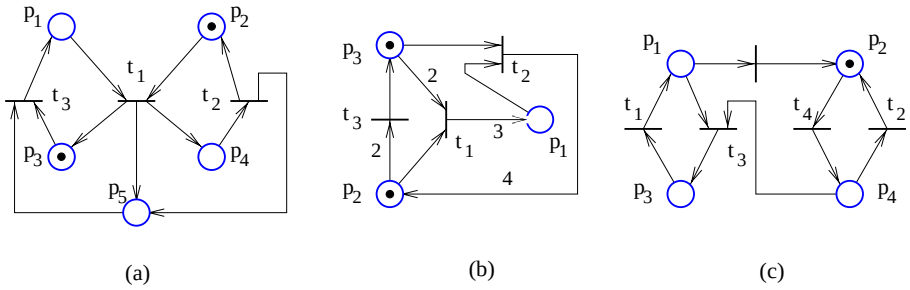

Figure 1: (a) A deadlocked PT-ordinary Petri net. An empty siphon is $\{p_1, p_4, p_5\}$. (b) A deadlocked Petri net with no empty siphon which is not PT-ordinary. (c) A deadlock-free Petri net (for the marking displayed) with an empty siphon – $\{p_1, p_3\}$.

Definition 3.2 *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 $i = 1, 2, \ldots |P|$, 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 may not be true for more general Petri nets. Proposition 3.1 has been generalized in [2] for Petri nets which are not PT-ordinary (but see also Proposition 3.2 and comments in [8]). We do not use that result. Instead we transform generalized Petri nets to PT-ordinary Petri nets (refer to section 4.1) and then use Proposition 3.1. Another drawback of Proposition 3.1 is that it is not effective for Petri nets which are not *repetitive*. We define the repetitive Petri nets in section 3.2 and then we give new results which are adequate for the Petri nets which are not repetitive.

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 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 ([16] p.103). We include later in the section a generalization to asymmetric choice nets as Theorem 3.2.

3.2 Conditions for Deadlock Prevention and Liveness Enforcement

Definition 3.3 *Let $\mathcal{N} = (P, T, F, W)$ be a Petri net, $\mathcal{M}$ the set of all markings of $\mathcal{N}$ and $U \subseteq \mathcal{M}$. 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 $\mathcal{M} \setminus U$ are called* **forbidden markings**.

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

We say that **deadlock can be prevented** in a Petri net $\mathcal{N}$ if there is an initial marking μ_0 and a supervisory policy Ξ such that $(\mathcal{N}, \mu_0)$ supervised by Ξ is deadlock-free. Similarly, we say that **liveness can be enforced** in $\mathcal{N}$ if there is an initial marking μ_0 and a supervisory policy Ξ such that $(\mathcal{N}, \mu_0)$ supervised by Ξ is live.

Proposition 3.2 *If a supervisory policy Ξ which prevents deadlock in $(\mathcal{N}, \mu_0)$ exists, then for all $\mu \geq \mu_0$ there is a supervisory policy which prevents deadlock in $(\mathcal{N}, \mu)$. The same is true for liveness enforcement.*

Proof: Let $\mu_1 \geq \mu_0$. A supervisory policy for $(\mathcal{N}, \mu_1)$ 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. □

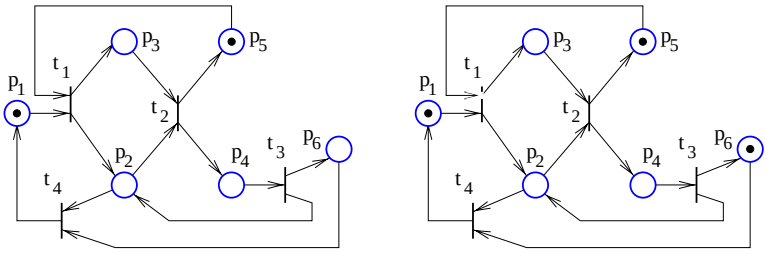

Figure 2: A Petri net which is live for the initial marking μ_0 shown in (a) and not even deadlock-free for the initial marking $\mu \geq \mu_0$ shown in (b).

Definition 3.4 [14] *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 sufficiency proof of Theorem 3.1, which is a known result. The authors are unaware of a reference in which Lemma 3.1 or the sufficiency proof of Theorem 3.1 appear. We prove the lemma as we need it in order to prove a number of other results, including Corollary 3.2. A related proof appears in [16] at page 70.

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_I which enables an infinite firing sequence σ. Let $U \subseteq T$ be the set of transitions which appear infinitely often in σ. There is a nonnegative integer vector x such that $Dx \geq 0$, $x(i) \neq 0$ $\forall t_i \in U$ and $x(i) = 0$ $\forall t_i \in T \setminus U$, where t_i denotes the transition corresponding to the i'th column of D.*

Proof: Consider firing σ and let μ_0 be the marking reached after all transitions which appear finitely often in σ have fired. Let $\sigma = \sigma_0\sigma_1\sigma_2 \ldots \sigma_k \ldots$ such that each σ_k is finite, for all $k \geq 1$ each of the transitions in U appears in σ_k, and $\mu_I[\sigma_0 > \mu_0$. Then let μ_1, μ_2, ... be defined as follows: $\mu_{k-1}[\sigma_{k-1} > \mu_k$ for all $k \geq 1$.

Let V_n be a nonempty set of the form $V_n = \{y \in \mathbb{N}^n : \nexists y_i \in V_n, y \neq y_i, y \geq y_i \text{ or } y \leq y_i\}$. 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 $C_{k,u} = \{y \in \mathbb{N}^n : y(j_k) = u, y(i_k) > y_{s,n}(i_k), \nexists y_i \in V_n, y \neq y_i, y \geq y_i \text{ or } y \leq y_i\}$, is defined for $0 \leq u < y_{s,n}(j_k)$ and $k = 1, 2 \ldots n(n-1)$ 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 $\nexists y \in \mathcal{M} : y \geq \mu_i$ 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 $\nexists i, j$, $0 \leq i < j$, 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 $y_x \not\leq u$ and $y_x \not\geq u$. 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 $\mathcal{M}^{(2)}, \ldots \mathcal{M}^{(j)}$, also subsets of $\widetilde{\mathcal{M}}$. However these operations cannot be repeated infinitely often: $j \leq |\widetilde{\mathcal{M}}|$, 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 a contradiction.

Therefore $\exists j, k$, $j < k$, such that $\mu_j \leq \mu_k$. Let q_j and q_k be the firing count vectors: $\mu_j = \mu_0 + Dq_j$ and $\mu_k = \mu_0 + Dq_k$; let $x = q_k - q_j$. Then $\mu_k - \mu_j \geq 0 \Rightarrow Dx \geq 0$, and by construction $x \geq 0$, $x(i) > 0$ $\forall t_i \in U$ and $x(i) = 0$ $\forall t_i \in T \setminus U$. □

Theorem 3.1 [14] *A Petri net is (partially) repetitive if and only if a vector x of positive (nonnegative) integers exists, such that $D \cdot x \geq 0$, $x \neq 0$.*

In general it may not be possible to enforce liveness or to prevent deadlock in an arbitrary given 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 enforcible are repetitive, and Petri nets in which deadlock is avoidable are partially repetitive. Part (b) of the corollary also appears in [18].

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 $(\mathcal{N}, \mu_0)$ if and only if $\mathcal{N}$ is partially repetitive.

(b) Initial markings μ_0 exist such that liveness can be enforced in $(\mathcal{N}, \mu_0)$ if and only if $\mathcal{N}$ is repetitive.

Proof: (a) If deadlock can be avoided in $(\mathcal{N}, \mu_0)$ 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 3.1 there is a nonnegative 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, ..., $d_{|P|}^T$, then a marking which enables σ_x is

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

At least one deadlock prevention strategy exists for μ_0: to allow only the firing sequence $\sigma_x, \sigma_x, \sigma_x, \ldots$ 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). □

Let Ξ denote a supervisory policy. Let $\mathcal{R}(\mathcal{N}, \mu_0, \Xi)$ denote the set of reachable markings from initial marking μ_0, when $(\mathcal{N}, \mu_0)$ 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 3.2 *Consider a Petri net $\mathcal{N} = (P, T, F, W)$ which is not repetitive. 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.*

Proof: There is an integer vector $x \geq 0$ with *maximum support* such that $Dx \geq 0$, which means that for all 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, 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 are in $\|x\|$. Therefore T_D is nonempty. Next, the proof shows that all transitions in $\|x\|$ can be made live, which implies that $T \setminus T_D = \|x\|$.

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 $\sigma_x, \sigma_x, \sigma_x, \ldots$. 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. □

In Corollary 3.2, T_D is nonempty. Otherwise, since all transitions from $T \setminus T_D$ could simultaneously be made live, this would imply that $\mathcal{N}$ is repetitive, which is a contradiction. A special case is $T \setminus T_D = \emptyset$, when the Petri net is not even partially repetitive, and so deadlock can not be avoided for any marking.

It was already shown that only repetitive Petri nets can be made live. The corollary above shows that the set of transitions of a partially repetitive Petri net can be uniquely divided in transitions that can be made live and transitions that cannot be made live. So the liveness property of partially repetitive Petri nets is that all transitions that can be live are live.

3.3 A Characterization of Petri Nets Based on Subnets which Can Be Made Live, in View of Deadlock Prevention and Liveness Enforcement

We denote by an *active subnet* a part of a Petri net which can be made live by supervision for appropriate markings. In the following definition we use the notations from Corollary 3.2.

Definition 3.5 *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.*

Definition 3.6 *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.)*

Proposition 3.3 *A siphon which contains places from an active subnet is an active siphon with respect to that subnet.*

Proof: Using the notations from Definition 3.5, 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 3.5. 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 $\mathcal{N}^A$. □

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 an 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 $\mathcal{N}$. □

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.

Proposition 3.4 *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 $(\mathcal{N}, \mu)$ 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: $S_0 = s$, $S_i = S_{i-1} \cup \{p \in {}^\bullet({}^\bullet S_{i-1} \setminus S_{i-1}{}^\bullet) : \mu(p) = 0\}$, 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. □

The practical significance of Proposition 3.4 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 3.5 *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. If deadlock is avoidable, after some transitions firings a marking can be reached which enables $\sigma_x \sigma_x \ldots$, where σ_x is a finite firing sequence. Let q be the firing count vector for σ_x. Then $Dq \geq 0$. If the active subnet for q is minimal, we let $x = q$, but if it is not, there is x such that $\|x\| \subset \|q\|$, $x \neq 0$, $x \geq 0$, $Dx \geq 0$ and the active subnet associated to x is minimal. But it must be an active siphon with regard to that active subnet, therefore not all of the transitions of $\|x\|$ can fire, which implies that not all of the transitions of σ_x can fire, which is a contradiction. □

The previous result supports 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. The usage of Proposition 3.5 for maximally permissive deadlock prevention has been demonstrated in section 6.4.3 of [9].

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 $p_2{}^\bullet \subseteq p_1{}^\bullet$. The following new result can be seen as the correspondent for T-liveness of a previous result for liveness in [2]. However, note that even for liveness the next result is stronger, as it relates the dead transition to an empty siphon.

Theorem 3.2 *Consider a PT-ordinary asymmetric choice Petri net $\mathcal{N}$ and a marking μ such that a transition t is dead. Then there is $\mu' \in \mathcal{R}(\mathcal{N}, \mu)$ such that S is an empty siphon for the marking μ' and $t \in S^\bullet$.*

Proof: In an asymmetric choice Petri net, ${}^\bullet p_1 \cap {}^\bullet p_2 \neq \emptyset$ implies $p_1{}^\bullet \subseteq p_2{}^\bullet$ or $p_2{}^\bullet \subseteq p_1{}^\bullet$. Therefore given n places such that $p_i{}^\bullet \cap p_j{}^\bullet \neq 0$, $\forall\ i, j \in \{1, 2, \ldots n\}$, we have $p_{i1}{}^\bullet \subseteq p_{i2}{}^\bullet \subseteq \ldots p_{in}{}^\bullet$, where $i_1, \ldots i_n$ are distinct and $i_j \in \{1, 2, \ldots n\}$ for all $j = 1 \ldots n$.

Let ${}^\bullet t = \{p_1, \ldots p_n\}$, where the notation is chosen such that $p_1{}^\bullet \subseteq p_2{}^\bullet \subseteq \ldots p_n{}^\bullet$. We prove first that $\exists \mu_1 \in \mathcal{R}(\mathcal{N}, \mu)$ and $\exists j \in \{1, \ldots n\}$ such that $\forall \mu_x \in \mathcal{R}(\mathcal{N}, \mu_1)$: $\mu_x(p_j) = 0$. Assume the contrary. Let $\mu_1 = \mu$ and i be the least number in $\{1, \ldots n\}$ such that $\exists \mu_{i,1} \in \mathcal{R}(\mathcal{N}, \mu_1)$: $\mu_{i,1}(p_i) = 0$ (i exists, for t is dead and

$\mathcal{N}$ is PT-ordinary). Then $\exists \mu_{i,2} \in \mathcal{R}(\mathcal{N}, \mu_{i,1})$: $\mu_{1,2}(p_i) \geq 1$. If $\forall \mu_{i,3} \in \mathcal{R}(\mathcal{N}, \mu_{i,2})$: $\mu_{i,3}(p_i) \geq 1$, then let $\mu_1 = \mu_{i,2}$, let i be the least integer in $\{1, \ldots n\}$ such that $\exists \mu_{i,1} \in \mathcal{R}(\mathcal{N}, \mu_1)$: $\mu_{i,1}(p_i) = 0$ and repeat the operation above. Note that i is increasing, and so after at most n such steps we find that $\exists \mu_{i,3} \in \mathcal{R}(\mathcal{N}, \mu_{i,2})$: $\mu_{i,3}(p_i) = 0$. (Otherwise we would have a reachable marking enabling t.) From $\mu_{i,2}(p_i) \geq 1$ and $\mu_{i,3}(p_i) = 0$ we infer that $\exists \mu_{i,4} \in \mathcal{R}(\mathcal{N}, \mu_{i,2})$ and $\exists t_i \in p_i\bullet$ such that $\mu_{i,4}$ enables t_i. Note that $t_i \in p_j\bullet$ $\forall j = i \ldots n$, so $\mu_{i,4}(p_j) \geq 1$ $\forall j = i \ldots n$. By the choice of i, $\mu_{i,4}(p_j) \geq 1$ $\forall j = 1 \ldots i-1$. Therefore $\mu_{i,4}$ enables t. Contradiction.

Therefore, $\exists \mu_1 \in \mathcal{R}(\mathcal{N}, \mu)$ and $\exists j \in \{1, \ldots n\}$ such that $\forall \mu_x \in \mathcal{R}(\mathcal{N}, \mu_1)$: $\mu_x(p_j) = 0$. We recursively use this property to construct S. Note that all transitions in $\bullet p_j$ are dead for μ_1. Let $S_0 = \emptyset$ and $S_1 = \{p_j\}$. We recursively construct S by generating $S_2, \ldots S_{n+1}$ and the markings $\mu_2, \ldots \mu_{n+1}$. S_i for $i \geq 1$ is such that all transitions in $\bullet S_i$ are dead for some marking μ_i. The construction in a iteration is as follows. Let $\mu_{i+1} \in \mathcal{R}(\mathcal{N}, \mu_i)$ such that $\forall t \in \bullet(S_i \setminus S_{i-1})$ $\forall \mu_x \in \mathcal{R}(\mathcal{N}, \mu_{i+1})$ $\exists p \in \bullet t$: $\mu_x(p) = 0$. Then we let $S_{i+1} = S_i \bigcup_{t_x \in \bullet(S_i \setminus S_{i-1})} \{p \in \bullet t_x : \forall \mu_x \in \mathcal{R}(\mathcal{N}, \mu_{i+1}) : \mu_x(p) = 0\}$. There is n such that $S_{n+1} = S_n$, for the Petri net has a finite number of places. We let $S = S_n$ and $\mu' = \mu_n$. Since $p_j \in S$, $t \in S\bullet$. By construction S is a siphon, S is empty for μ', and $\mu' \in \mathcal{R}(\mathcal{N}, \mu)$. □

In general we may not want all transitions to be live. For instance some transitions of a Petri net may model faults and we want to insure that some other transitions are live. This is the reason for the following definition.

Definition 3.7 *Let $(\mathcal{N}, \mu_0)$ be a Petri net and T a subset of the set of transitions. The Petri net is said to be* **T-live** *if all transitions $t \in T$ are live.*

Note that a live transition is not the opposite of a dead transition. That is, a transition may be neither live nor dead. Indeed, a transition is live if there is no reachable marking for which it is dead. Note also that T-liveness corresponds to liveness when the set T equals the set of transitions.

Definition 3.8 *Let $\mathcal{N}$ be a Petri net, T a subset of the set of transitions and $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$ an active subnet. We say that $\mathcal{N}^A$ is* **T-minimal** *if $T \subseteq T^A$ and $T^A \not\subseteq T_x^A$ for any other active subnet $\mathcal{N}_x^A = (P_x^A, T_x^A, F_x^A, W_x^A)$ such that $T \subseteq T_x^A$.*

In general the T-minimal active subnet is not unique. However, as shown in the next Proposition, any T-minimal active subnet can be used to characterize T-liveness.

Proposition 3.6 *Given a PT-ordinary asymmetric choice Petri net $\mathcal{N}$, let T be a set of transitions and $\mathcal{N}^A$ a T-minimal active subnet which contains the transitions in T. If all the minimal siphons with respect to $\mathcal{N}^A$ are controlled (i.e. they cannot become empty for any reachable marking), the Petri net is T-live (and T^A-live). If the Petri net is T-live, there is a T-minimal active subnet $\mathcal{N}^A$ such that all minimal active siphons with respect to $\mathcal{N}^A$ are controlled.*

Proof: Assume that no active siphon becomes empty. If there is a reachable marking such that a transition $t \in T^A$ is dead (and $T \subseteq T^A$), by Theorem 3.2 there is a reachable marking such that a siphon S is empty and $t \in S\bullet$. However $t \in S\bullet$ implies $S \cap P^A \neq \emptyset$, and by Proposition 3.3 S is an active siphon. Contradiction, for S is empty.

Let $\mathcal{N}_i^A$ denote a T-minimal active subnet, $i = 1 \dots k$, where k is the number of T-minimal active subnets. If there is a reachable marking μ such that an active siphon S_i is empty, let $T_i = S\bullet \cap T_i^A$, where T_i^A is the set of transitions of $\mathcal{N}_i^A$. Because S_i is active, T_i is nonempty; because S_i is empty, the transitions of T_i are dead. Assume that there is an infinite firing sequence σ_x such that all transitions of T appear infinitely often in σ_x and after a part of σ_x is fired, (let μ_x be the marking reached) all T-minimal active subnets $\mathcal{N}_i^A$ have an empty active siphon S_i. Let σ be the remaining part of σ_x which is enabled by μ. All transitions of T appear infinitely often in σ. Therefore, by Lemma 3.1, there is $x \geq 0$ such that $Dx \geq 0$ (D is the incidence matrix) and $T \subseteq \|x\|$. However, $\|x\|$ does not contain all transitions of any of the T-minimal subnets $\mathcal{N}_i^A$: $T_i \subseteq \|x\| \setminus T_i^A$, for $i = 1 \dots k$. This implies that $\|x\|$ defines another T-minimal active subnet, which is a contradiction. □

4 Preliminaries to the Liveness Enforcing Method

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

We are interested in using a transformation to PT-ordinary Petri nets because Propositions 3.1 and 3.4 in section 3 apply to PT-ordinary Petri nets. We use a modified form of the similar transformation from [12], 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** (decomposed) in several new transitions:

> The transition t_j is **split** in $m = n(t_j)$ transitions: $t_{j,0}, t_{j,1}, t_{j,2}, \dots t_{j,m-1}$, where $n(t_j) = \max\{W(p, t_j) : (p, t_j) \in F\}$. Also, $m - 1$ new places are added: $p_{j,1}, p_{j,2}, \dots p_{j,m-1}$. The connections are as follows:
>
> (i) $\bullet p_{j,i} = t_{j,i}$, $t_{j,i}\bullet = p_{j,i}$ and $p_{j,i}\bullet = t_{j,i-1}$, for $i = 1 \dots m-1$
>
> (ii) $\bullet t_{j,i} = \{p \in \bullet t_j : W(p, t_j) > i\}$, for $i = 0 \dots m-1$
>
> (iii) $t_{j,0}\bullet = t_j\bullet$
>
> Note that t_j resembles very much $t_{j,0}$: $t_{j,0}$ has all the connections of t_j plus one additional transition arc. *After the split is performed, we denote $t_{j,0}$ by t_j.*

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

$$|p_{j,i}\bullet| = |\bullet p_{j,i}| = 1 \qquad i = 1 \dots m-1 \tag{4}$$

$$|t_{j,i}\bullet| = 1 \qquad i = 1 \dots m-1 \tag{5}$$

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.

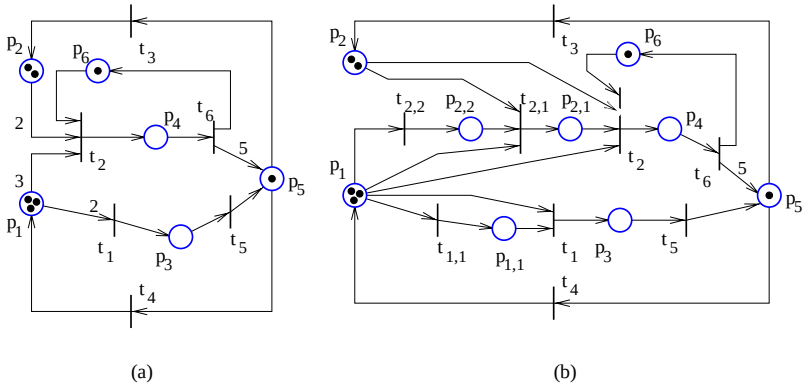

Figure 3: Illustration of the PT-transformation. (a) Original net and (b) transformed net.

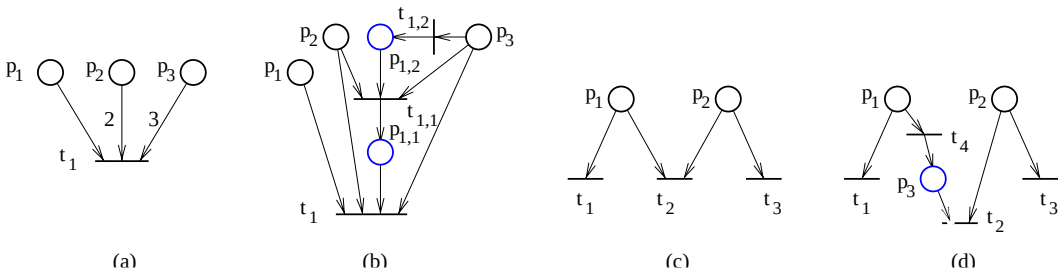

Figure 4: Illustration of the transition split: (a) initial configuration; (b) the effect of the PT-transformation; (c) initial configuration; (d) the effect of the AC-transformation.

Figure 3 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 7(a), which is changed as shown in figure 7(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 Transformation of Petri nets to asymmetric choice Petri nets

Let $\mathcal{N} = (P, T, F, W)$ be a Petri net and $\mathcal{N}' = (P', T', F', W')$ be the transformed Petri net, where $P \subseteq P'$, $T \subseteq T'$. The idea of the transformation is as follows. Given the transition t, $p_i \in \bullet t$ and $p_j \in \bullet t$ such that $p_i\bullet \not\subseteq p_j\bullet$ and $p_j\bullet \not\subseteq p_i\bullet$, remove t from either the postset of p_i or that of p_j by adding an additional place and transition. The idea is illustrated in figure 4(c-d). Note that the operations correspond to a modified form of transition split operations (section 4.1).

Algorithm of the AC-Transformation

Input: $\mathcal{N}$ and optionally $M \subseteq P$; the default value of M is $M = P$.

Output: $\mathcal{N}'$

Initialize $\mathcal{N}'$ to be identical with $\mathcal{N}$.

For every $t \in T$ with $|\bullet t| > 1$ **do**

1. Construct $U = \{(p_i, p_j) \in P \times P : p_i \in \bullet t, p_j \in \bullet t, p_i\bullet \not\subseteq p_j\bullet$ and $p_j\bullet \not\subseteq p_i\bullet\}$.
2. **if** U is empty, **then** continue with the next iteration.
3. Let $Q := \emptyset$.
4. **For** every $(p_i, p_j) \in U$
   (a) A place $p \in \{p_i, p_j\} \cap M$ is selected. If two choices are possible:
      i. $p = p_i$ (or $p = p_j$) if p_i (or p_j) has been previously selected for another element of U.
      ii. otherwise p is chosen such that p appears in other element of U. If both p_i and p_j satisfy this property, select $p \in \{p_i, p_j\}$ such that $|p\bullet| = \max\{|p_i\bullet|, |p_j\bullet|\}$.
      iii. if none of p_i and p_j appears in another element of U, select $p \in \{p_i, p_j\}$ such that $|p\bullet| = \max\{|p_i\bullet|, |p_j\bullet|\}$.
   (b) If a place p could be selected (i.e. if $\{p_i, p_j\} \cap M \neq \emptyset$) then $Q := Q \cup \{p\}$
5. **For** all $p \in Q$, delete from $\mathcal{N}'$ the transition arc (p, t) and add a new place p' and a new transition t' such that $\bullet t' = \{p\}$, $t'\bullet = \{p'\}$, $p'\bullet = \{t\}$, $W'(p, t') = W'(t', p') = 1$ and $W'(p', t) = W(p, t)$.

We call the transformation to asymmetric choice Petri nets **AC-transformation**. The operation in the step 5 of the algorithm is a **transition split**. The transition split of the AC-transformation is slightly different from the transition split of the PT-transformation in section 4.1.

The second argument of the algorithm, M, is used by the liveness enforcement procedure in order to select the transitions which the algorithm splits. Indeed, in general there are many ways in which to choose which transitions to be split such that the transformed net is with asymmetric choice. It will be seen that the liveness enforcement procedure selects M such that the place invariants created in previous iterations are not modified by the AC-transformation.

4.3 Petri Net Supervisors Based on Place Invariants

We outline here results from [13, 22] for supervisors based on linear constraints, in the particular case of fully controllable and observable Petri nets. The results of this section still apply for Petri nets with uncontrollable and unobservable transitions, if the desired constraints are *admissible*.

4.3.1 Fully Controllable and Observable Petri Nets

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

$$L \cdot \mu_p \leq b \tag{6}$$

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:

$$\mu_c = b - L \cdot \mu_p \tag{7}$$

Let μ_{c0} be the slack variables that correspond to the initial marking μ_{p0}, that is $\mu_{c0} = b - L\mu_0$. Let q be the firing vector associated with the transitions that led the Petri net from μ_{p0} to μ_p and D_p the incidence matrix, that is $\mu_p = \mu_{p0} + Dq$. So we see that $\mu_c = b - L \cdot (\mu_{p0} + D_p \cdot q)$, which also can be written as:

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

Therefore 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 $L \cdot \mu_{p0} > b$ cannot be considered, since equation (7) shows that in this case μ_{c0} will not be nonnegative. 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 elements 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 [13, 22]. □

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

4.3.2 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 [13] are true:

$$LD_{uc} \leq 0 \tag{9}$$

$$LD_{uo} = 0 \tag{10}$$

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 4.3.1. 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 [13].

4.4 Siphon Control Based on Place Invariants

Proposition 3.1 showed that in a PT-ordinary Petri net deadlock is not possible if all siphons are controlled. This suggests that all siphons should be made controlled siphons. An easy way to make a siphon controlled is to create a place invariant to control the siphon. This is done below by adding an additional place to the original Petri net. Early references of this approach for siphon control are in [1, 5]. This section presents it as a special case of the supervision method based on place invariants (section 4.3). The operations described here do not depend on the fact that the structure they are applied to is a siphon, so they are described in more general terms.

4.4.1 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, $\sum_{p \in S} \mu(p) \geq 1$ is the desired control policy. This constraint can be enforced using the methodology of invariant based supervision of [13, 22], outlined in section 4.3, which yields an additional place C, called **control place**. The place invariant 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 closed loop incidence matrix. This invariant corresponds to the equation

$$\mu(C) = \sum_{p_k \in S} \mu(p_k) - 1 \tag{11}$$

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

(a) $\bullet C = \emptyset$ 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 [13], p.87-88).

(b) $C\bullet \subseteq \bullet S$ (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 $\sum_{p \in S_0} \mu_0(p) \geq 1$.

(c) $\bullet C = \emptyset$ and $C\bullet = \emptyset$: the marking of S cannot vary, and so there is a place invariant x such that $x(i) = 1$ for all $p_i \in S$ and $x(i) = 0$ otherwise.

Case (a) detects transitions that cannot be made live when S is a siphon (Corollary 3.2). Case (b) shows the case when S does not need control. Note that the method depends only on structural properties of the Petri net. That is, it does not detect whether S does not need control for some initial markings, but it detects only the case when S does not need control for all initial markings μ_0 such that $\sum_{p \in S} \mu_0(p) \geq 1$. Therefore the

method when applied to a siphon that is not a trap, but includes a trap, always produces a control place. The reason that this is correct is that there are nonzero initial markings of the siphon such that the included trap has null marking; hence the siphon is not trap controlled for such markings.

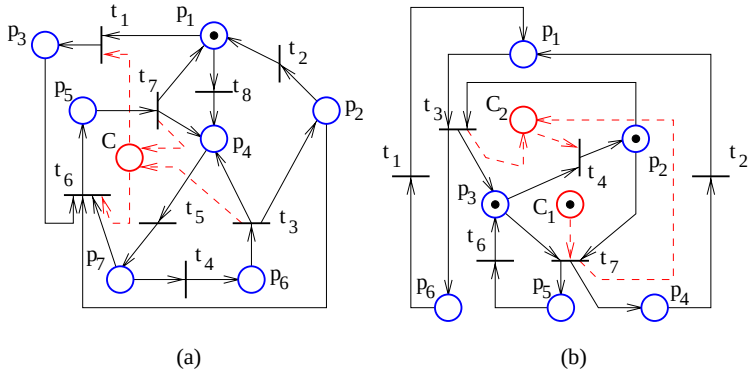

Figure 5: Siphon Control Examples. Connections to control places are dashed.

In figure 5(a) there is a single minimal siphon, $\{p_1, p_2, p_4, p_5, p_6, p_7\}$. The siphon includes a trap $\{p_4, p_5, p_6, p_7\}$, but it is not trap controlled because the marking of the trap is 0. The control place C prevents firing t_1, which would empty the siphon. In figure 5(b) the original Petri net has two minimal siphons, $\{p_3, p_2, p_5\}$ 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 that results by controlling the minimal siphon $\{p_2, C_2\}$ satisfies $\bullet C = \emptyset$ and $C\bullet = \emptyset$.

By Theorem 4.1, the way in which the constraint $\sum_{p \in S} \mu_0(p) \geq 1$ was enforced is maximally permissive. Therefore, because the enforcement of this constraint on a siphon by definition 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 closed loop remains a Petri net.

4.4.2 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 [13]. Yet we will choose to use a different method in section 5.2.6. 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 section 5.2.6 and [13], $b = 1$ (in [13] 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 section 5.2.6 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 4.4.1, 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 section 5.2.6 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.

Note also that this is not the exact way we do the control of siphons in the case of uncontrollable and unobservable transitions. The liveness enforcement procedure needs to take in account more than just admissibility constraints. Refer to section 5.2.6 for the details.

5 The Liveness Enforcing Method

5.1 Introduction to the Method

The method we introduce in this paper produces supervisors enforcing liveness. In some cases it might not be desirable to enforce that all transitions of a Petri net are live, but rather that only some of them are live (for instance transitions modeling system faults are not desired to be live). Our method solves a problem more general than liveness, in which the objective is to insure that the transitions in a given set T are live. In this context we have introduced in Definition 3.7 the concept of *T-liveness*.

Given a target Petri net $\mathcal{N}_0$, the liveness enforcing procedure generates a sequence of asymmetric choice PT-ordinary Petri nets, $\mathcal{N}_1$, $\mathcal{N}_2$, ... $\mathcal{N}_k$, increasingly enhanced for liveness. $\mathcal{N}_1$ is $\mathcal{N}_0$ transformed to be PT-ordinary and with asymmetric choice. 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 for the PT and/or the AC transformation. Thus the iteration i produces the asymmetric choice PT-ordinary net $\mathcal{N}_{i+1}$. The active siphons (see Definition 3.6) 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, in view of Proposition 3.6, the transitions of the active subnet of $\mathcal{N}_k$ are live 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 liveness enforcement 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 used, the constraints $[l, -l]^T\mu \geq [c, -c]^T$ are part of the 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 $L_I\mu \ngeq b_I$, (b) less complex supervisors can be obtained if the procedure takes in account that markings such that $L_I\mu \ngeq b_I$ are never reached for all initial markings for which the liveness enforcement supervisor will be used, and (c) convergence help is needed. Case (a) occurs when the liveness enforcing procedure is applied to the closed loop Petri net resulted from a supervision based on place invariants; in this case the initial constraints specify the equations which the markings of the control places of the target Petri net must satisfy.

The liveness enforcement procedure is defined in section 5.4. The sections preceding section 5.4 define

in detail operations performed by the procedure. Sections 4.1 and 4.2 have shown how the Petri nets are transformed to be PT-ordinary and with asymmetric choice. Sections 4.3 and 4.4 have shown 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 a T-minimal 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 Generating Marking Constraints

The liveness enforcement procedure gradually restricts the sets of acceptable markings. To each minimal active siphon corresponds a linear inequality expressing 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 the initial constraints on the target net are changed by transition splits. 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 4.4 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 $\alpha_p \geq 0$. When all transitions of S are controllable and observable: $\alpha_p = 1\ \forall p \in S$. 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 Petri net considered in an iteration is PT-ordinary and with asymmetric choice. However, by adding control places, the net may no longer be PT-ordinary or with asymmetric choice. Therefore the liveness procedure PT-transforms the Petri net in order that the next iteration will work on a PT-ordinary net. The control places and the PT-transformation may cause the Petri net not to be with asymmetric choice. Therefore the AC-transformation is applied. The PT-transformation and the AC-transformation may change the place invariant of a control place. Proposition 6.8 proves that a place invariant $\mu(C) = \sum_{p\in S} \alpha_p \mu(p) - 1$ is transformed to

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

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 $p_{i,j}$. The places p_z are the places resulting from the

AC-transformation which satisfy $\bullet\bullet p_z = C$. In particular, if $\forall t_i \in C\bullet$: $m_i = 1$ and the AC-transformation does not generate places p_z such that $\bullet\bullet p_z = C$ (for instance, this may happen when the Petri net remains with asymmetric choice after adding C) the place invariant is not changed

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

Assume that a control place C is added to $\mathcal{N}_i$, and so at the end of the iteration i the enforced place invariant is (12). By Proposition 6.9(c), the form of the place invariant stays the same in all $\mathcal{N}_{i+1}, \mathcal{N}_{i+2}, \ldots$. Therefore no update is necessary in a iteration j for constraints added in previous iterations.

5.2.2 Constraints which do 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. Also the form of the constraint is not changed to the form (12). Instead it remains the same in all further iterations, by Proposition 6.9(c). In order to reduce the complexity of the supervisor, the case when a desired constraint does not need a control place is identified (see section 4.4.1). No control place is added in such a case. 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 liveness enforcement 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 8. Example 5.3 also shows how the constraints (L_0, b_0) are obtained.

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

Equation (12) shows that control places enforce inequalities of the form

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

Section 5.2.2 showed that the inequalities which do not need control place enforcement have the same form. Consider an inequality of the form (14) which is added in the iteration number i. Note that S in (14) may contain control places added in the previous iterations $i-1$, $i-2$, ..., 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 of the expressions giving the marking of a control place, 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).) Then, for convenience, we drop the null columns of l which correspond to control places and take μ with respect to the places which are not control places. This is how the inequalities $L\mu \geq b$ are obtained. The inequalities $L_0\mu \geq b_0$ are similarly obtained from the inequalities which do not need control place enforcement.

The Petri nets $\mathcal{N}_1, \mathcal{N}_2, \ldots$ have an increasing number of places. So the dimension of the marking vector μ is also increasing. The new places which are added in an 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 liveness enforcing procedure is to provide constraints in terms of the marking of the target net $\mathcal{N}_0$. The constraints of the net $\mathcal{N}_k$, denoting the 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 liveness enforcement procedure and they exist when the procedure is started. This section considers the way initial constraints should be 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 6.9(c). However this property is not always true for the initial constraints, since $\mathcal{N}_0$ may not be PT-ordinary and with asymmetric choice (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. By Proposition 6.9(c), if some constraint $L'\mu \geq b'$ is enforced in $\mathcal{N}_1$ for all valid initial markings of some set $\mathcal{M}'$, it stays enforced in all other nets $\mathcal{N}_i$ obtained in the following iterations, for all valid initial markings with restriction to the places of $\mathcal{N}_1$ in $\mathcal{M}'$. However, because of the PT and AC transformations, it may not be true that $L_0\mu \geq b_0$ is enforced in $\mathcal{N}_1$ for all 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 transformation appears in Proposition 6.9(a) and (b). (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.)

Let t_1, t_2, ... t_k be the transitions of $\mathcal{N}_0$ which are split to obtain $\mathcal{N}_1$. 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 $\mu(p) + \sum_{z=1}^{r} \mu(p_z) + \sum_{i=1}^{k} \sum_{j=1}^{m_i-1} j\mu(p_{i,m_i-j})$ 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; p_z are the places created by the AC-transformation such that ${}^\bullet{}^\bullet p_z = p$. We see, the substitution of $\mu(p)$ is simply $\mu(p)$ when no transitions in the postset of p are split and no places p_z appear. In particular, 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 4.4.2. We are to find the nonnegative integers α_p of the inequality

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

such that the constraint satisfies a number of requirements, which we specify in this section. The admissibility requirement appears because the final constraints $L\mu \geq b$ which define the supervisor of $\mathcal{N}_0$ are to be admissible. Thus we are to obtain the coefficients α_p such that the constraint of $L\mu \geq b$ which reflects (15) is admissible in $\mathcal{N}_0$. This operation is necessary in order to insure that all constraints of $L\mu \geq b$ are admissible in $\mathcal{N}_0$, while admissibility is needed to enforce $L\mu \geq b$ with the invariant based approach of section 4.3.1.

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:

$$a^T d > 0 \tag{16}$$

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 4.4.1 is used (that is, when no transformation to an admissible constraint is necessary.) We also require that at least two of the coefficients α_p are nonzero.

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. An additional constraint is

$$a^T D_s \leq 0 \tag{17}$$

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 4.4.1 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: $\mu_c = U\mu_p - g$, where U is a matrix and g an integer vector. $[\mu_c^T, \mu_p^T]^T$ can be obtained from μ by applying to μ a permutation π; let a_z be a after applying the permutation π and let $a_z = [a_c^T, a_p^T]^T$ (where a_c is the restriction of a_z to μ_c). Equation (15) can be written as

$$a_z^T [U^T, I]^T \mu_p \geq 1 + a_c^T g \tag{18}$$

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 4.3.2):

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

where N_r is obtained from $[U^T, I]^T$ as follows. Let V be $[U^T, I]^T$ with the rows permuted according to π^{-1}. Then N_r is the restriction of V 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). (This can be seen from the fact that a_z^T times the restriction of $[U^T, I]^T$ to the columns corresponding to the places in $\mathcal{N}_0$ does not depend on a_n.) Then we can choose a_n such that:

$$a^T D_{sn} = 0 \tag{20}$$

The advantage of doing this would be that the control place C will result with less connections, and so perhaps less siphons in the next iteration. The following algorithm finds the coefficients α_p. The algorithm does not fail if a solution of the form (15) exists.

Input: $\mathcal{N}_\prime = (P_0, T_0, F_0, W_0)$, P - the set of places at the current iteration, $P^A \subset P$ the set of places of the active subnet at the current iteration, $L\mu \geq b$ and $L_0\mu \geq b_0$ - the current constraints restricted to the markings of $\mathcal{N}_0$, and the siphon S.

Output: An admissible constraint (15).

1. Let α be the restriction of a to the places $p \in S$.
2. Initialize α to $\alpha_p = 1\ \forall p \in S$.
3. **If** (19) is satisfied **then** exit and declare $\sum_{p \in S} \mu(p) \geq 1$ admissible constraint.
4. Let R be the set of transitions which correspond to the constraints of (19) not satisfied by α.
5. **If** initial constraints have been given **then**[1]
   (a) **For** each $t \in R$
      i. **If** the system of inequalities $\mu(p) \geq W_0(p,t)\ \forall p \in {}^\bullet t$, $L\mu \geq b$, $L_0\mu \geq b_0$, $\mu \geq 0$ and μ integer vector is infeasible, **then** $R = R \setminus \{t\}$
   (b) **If** $R = \emptyset$ **then** exit and declare $\sum_{p \in S} \mu(p) \geq 1$ admissible.
6. Keep in (19) only the constraints which correspond to transitions in R. Then write (17), (19) and (20) as $Za \geq 0$ and then as $V\alpha \geq 0$, where V is the restriction of Z to the columns corresponding to the places $p \in S$.
7. Let $f = TRUE$ and $A = \emptyset$.
8. **While** f is $TRUE$
   (a) Check[2] the feasibility of $\sum_{i \notin A} x(i) \geq 1$ for $x \geq 0$ and $Vx \geq 0$.
   (b) **If** infeasible, $f = FALSE$.
   (c) **Else** let $A = A \cup \{p \in P : x(p) \neq 0\}$
9. **If** $A \cap P^A = \emptyset$ or $|A| < 2$ **then** declare siphon control failure and exit.[3]
10. Let $Y\alpha \geq b$ be the constraints $V\alpha \geq 0$ and $\alpha(i) \geq 1\ \forall i \in A$.
11. Solve the linear integer program $\min_\alpha \sum \alpha(i)$ subject to $Y\alpha \geq b$ and return α.

[1] without initial constraints the step below will not reduce R
[2] The feasibility check involves solving a linear program
[3] $|A|$ denotes the number of elements of A

5.3 The Computation of a T-minimal Active Subnet

The active subnet of a Petri net is defined in Definition 3.5. It 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 3.2 shows that for all rational 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 $\mathcal{N}_0 = (P_0, T_0, F_0, W_0)$ and its incidence matrix D; a nonempty set of transitions $T \subseteq T_0$; an optional set Z (default is $Z = \emptyset$) of 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. Check the feasibility of $Dx \geq 0$ subject to $x \geq 0$, $x(i) \geq 1$ $\forall t_i \in T$ and $x(i) = 0$ $\forall t_i \in Z$.

   If feasible **then** let x_0 be a solution; $T^A =$ *minactn(T_0, x_0, D, T)*

   else $T^A =$ *maxactn(T_0, D, T, Z)* (no T-minimal solution exists, and so an approximation is constructed)

2. The active subnet is $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$, $P^A = T^A\bullet$, $F^A = F_0 \cap \{(T^A \times P^A) \cup (P^A \times T^A)\}$ and W^A is the restriction of W_0 to F^A.

minactn(T_0, x_0, D, T)

Let $M = \|x_0\|$ and $x_s = x_0$.

For $t_i \in M \setminus T$ **do**

Check feasibility of $Dx \geq 0$ subject to $x \geq 0$, $x(i) = 0$, $x(j) = 0$ $\forall t_j \in T_0 \setminus M$ and $x(j) \geq 1$ $\forall t_j \in T$.

If feasible **then** let x^* be a solution; $M = \|x^*\|$ and $x_s = x^*$.

Return $\|x_s\|$

maxactn(T_0, D, T, Z)

Let $M = T$ and $x_s = \mathbf{0}_{|T_0| \times 1}$

While $M \neq \emptyset$ **do**

Check feasibility of $Dx \geq 0$ subject to $x \geq 0$, $\sum_{t_i \in M} x(i) \geq 1$ and $x(i) = 0$ $\forall t_i \in Z$.

If feasible **then** let x^* be a solution; $M = M \setminus \|x^*\|$ and $x_s = x^* + x_s$.

Else $M = \emptyset$.

$N =$ *minactn(T_0, x_s, D, $T \cap \|x_s\|$)*

Return N

In the cases when the 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 liveness enforcement procedure, let Z be the set of their indices in D, the incidence matrix. Then the active subnet is computed by using Z as input for the algorithm above. Using a nonempty set Z adds to the feasibility problems of the algorithm above the additional constraints that $x(j) = 0\ \forall j \in Z$. The set Z may also be used to specify transitions which are not desired to be live (for instance transitions modeling system faults.)

Because of the iterative nature of the liveness enforcement procedure, the active subnet needs to be reevaluated in every iteration. In general, the algorithm above needs to be used only once, to compute $\mathcal{N}_0^A$. The active subnet $\mathcal{N}_1^A$ and usually the other active subnets $\mathcal{N}_2^A$, $\mathcal{N}_3^A$, ... can be computed by simply repeating the changes done to $\mathcal{N}_{i-1}$ in $\mathcal{N}_{i-1}^A$, $i = 1, 2, \ldots$. (The procedure changes $\mathcal{N}_{i-1}$ by adding control places and splitting transitions.) Such an update of the active subnets is summarized by the following algorithm.

Input: $\mathcal{N}_{i-1}^A = (P_{i-1}^A, T_{i-1}^A, F_{i-1}^A, W_{i-1}^A)$, $\mathcal{N}_i = (P_i, T_i, F_i, W_i)$ and the sets $\Sigma(t)$, denoting for each $t \in T_{i-1}$ which has been split the set of the new transitions in $T_i \setminus T_{i-1}$ which appeared by splitting t.

Output: $\mathcal{N}_i^A = (P_i^A, T_i^A, F_i^A, W_i^A)$.

1. $T_i^A = T_{i-1}^A \cup \{t \in T_i : \exists t_u \in T_{i-1}^A \text{ and } t \in \Sigma(t_u)\}$
2. The active subnet is $\mathcal{N}_i^A = (P_i^A, T_i^A, F_i^A, W_i^A)$, $P_i^A = T_i^A{}^\bullet$, $F_i^A = F_i \cap \{(T_i^A \times P_i^A) \cup (P_i^A \times T_i^A)\}$ and W_i^A is the restriction of W_i to F_i^A.

The way above of updating 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. In Proposition 6.5 we show that the result of the update is indeed an active subnet.

We include other details about the computation of active subnets in the appendix of [8].

5.4 The Liveness Enforcing Procedure

Input: The target Petri net $\mathcal{N}_0$, a nonempty set of transitions T and a possibly empty set of initial constraints (L_I, b_I).

Output: Two sets of constraints (L, b) and (L_0, b_0) (T-liveness is enforced for all initial markings μ_0 such that $L\mu_0 \geq b$, $L_0\mu_0 \geq b_0$ when $(\mathcal{N}_0, \mu_0)$ is supervised according to $L\mu \geq b$.)

Procedure:

A. (L_0, b_0) is initialized to (L_I, b_I) and (L, b) to be empty. $\mathcal{N}_0$ is transformed to be PT-ordinary, as shown in section 4.1, and then to be with asymmetric choice, as shown in section 4.2. 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. Let $i = 1$. If not previously defined, let $X = \emptyset$.

B. T-minimal active subnets of $\mathcal{N}_0$ and $\mathcal{N}_1$ are computed such that they do not contain transitions in X. If no such T-minimal active subnets exist, an approximation is taken as shown in section 5.3,

such that the active subnets contain only a subset of T and no transitions of X. If no such nonempty approximations exist, the procedure terminates: even deadlock prevention is impossible for the given $\mathcal{N}_0$ and X.

C. **For** $i \geq 1$ **do** (the initial Petri net of the iteration i is $\mathcal{N}_i$; the active subnet is $\mathcal{N}_i^A$.)

1. If no new uncontrolled minimal active siphon is found, the next step is D. (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:

   Let C be the control place which would result by controlling the siphon, and let $l^T\mu \geq c$ be the inequality $\sum_{p \in S} \mu(p) \geq 1$ written in the form derived in section 5.2.3. First, the approach of section 4.4.1 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 C.2.c.

   (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 4.4.2 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 4.4.1.

   In both cases (i) and (ii) (l, c) is included in (L, b), except when the approach of the sections 4.4.2 and 5.2.6 fails to find an admissible constraint. When this failure occurs, all transitions of $S\bullet$ are included in X, $(X \rightarrow X \cup S\bullet)$, being marked as transitions which cannot be prevented (by the supervisor) to become dead. The active subnet, when updated in step 6, 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 infeasible, all transitions of $S\bullet$ are marked as dead, that is $X \rightarrow X \cup S\bullet$, and the procedure will take it in account in step 6. 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 Petri net is PT-transformed (section 4.1.)
4. If the Petri net is no longer with asymmetric choice, the Petri net is AC-transformed according to the algorithm of section 4.2, where the second argument M is taken to be the set of the control places added in the current iteration.
5. The matrices L and L_0 are enhanced with new columns, each column corresponding to one new place resulted in the steps 3 and 4.
6. The active subnet is updated according to the changes made in the total net in the steps 2(b), 2(c), 3 and 4, such that the transitions of X do not appear in the active subnet. If the new active subnet is empty, the procedure cannot even prevent deadlock and so it terminates.
7. Let T^A be the set of transitions of the active subnet. If an infeasibility occurred at a step C.2.c of the current iteration, $X \rightarrow T_0 \setminus T^A$ and the procedure is restarted at the step A with this value of X.

8. 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.

F. The supervisor of $\mathcal{N}_0$ is built according to the constraints (L, b), as shown in section 4.3.

5.5 Remarks

1. The purpose of the procedure is to produce two sets of linear constraints on the marking of the target net and 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 $L_0\mu_0 \geq b_0$, T-liveness in the closed loop Petri net is guaranteed for the condition of Theorem 6.2. When infeasibility occurs in a step C.2.c, the supervisor enforces T'-liveness, for some $T' \subset T$.

2. 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 $L_0\mu \geq b_0$. Another usage could be to specify place invariant properties of the net.

3. The procedure assumes that the initial constraints which are given to it are already enforced in the Petri net. This is not a real restriction, as linear marking constraints can easily be enforced by using the invariant based supervision of [13, 22].

4. The difference between the constraints (L, b) and (L_0, b_0) 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 original Petri net in closed loop with the supervisor enforcing (L, b) for all initial markings μ_0 of the original Petri net that satisfy $L_0\mu_0 \geq b_0$ in addition to $L\mu \geq b$. The procedure is allowed to start with initial constraints of the type (L_0, b_0), to which it may add other constraints, as necessary. However, without reducing the generality (see section 6.3.1), no initial constraints of the form (L, b) are allowed.

5. 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 6.10, each new minimal active siphon contains at least a control place added in iteration i to $\mathcal{N}_i$ or a place from $P_i^A \setminus P_{i+1}^A$.

6. Note that liveness enforcement corresponds to $T = T_0$ and it makes sense for repetitive Petri nets. In this case the T-minimal active subnet is the maximal active subnet, which is equal to the total Petri net.

5.6 Illustrative Examples

Example 5.1 (Verification) The Petri net of figure 6(a) is live for all total markings greater than two. We use this Petri net in order to illustrate how the liveness enforcement procedure is also useful for verification purposes.

Because the Petri net is not with asymmetric choice, the AC-transformed Petri net is shown in figure 6(b). Since liveness is to be enforced and the Petri net is repetitive, the active subnet is the maximal active subnet,

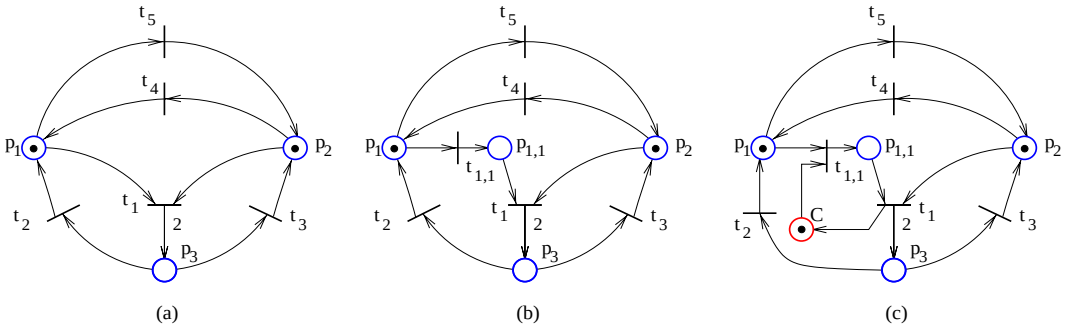

Figure 6: Example 5.1: (a) the target net $\mathcal{N}_0$, (b) $\mathcal{N}_1$ and (c) $\mathcal{N}_2$.

which equals $\mathcal{N}_1$. There is a single minimal active siphon: $\{p_1, p_2, p_3\}$. The control place C thus results. The constraint $\mu(p_1)+\mu(p_2)+\mu(p_3) \geq 1$ is added to (L, b). At the second iteration, there is a single minimal siphon: $\{C, p_{1,1}\}$. This siphon does not need control, as we are in the situation of step C.2.a of the procedure. Thus the constraint $\mu(C)+\mu(p_{1,1} \geq 1$ can be written as $\mu(p_1)+\mu(p_{1,1})+\mu(p_2)+\mu(p_3) \geq 2$, which is added to (L_0, b_0). At the third iteration no new active siphon appears, so the procedure terminates. The inequalities at the step D are: $\mu(p_1)+\mu(p_2)+\mu(p_3) \geq 1$ (in (L, b)) and $\mu(p_1)+\mu(p_2)+\mu(p_3) \geq 2$ (in (L_0, b_0)). The first inequality is redundant and so removed. Therefore the procedure terminates with empty (L, b) and with

$$\mu(p_1) + \mu(p_2) + \mu(p_3) \geq 2 \tag{21}$$

in (L_0, b_0). By Theorem 6.2, whenever the initial marking satisfies (21), the Petri net is live. □

Example 5.2 (T-liveness enforcement) Consider the Petri net of figure 7(a), which is not PT-ordinary and not with asymmetric choice. Three transitions cannot be made live, for any marking: t_1, t_2, t_3. The purpose is to enforce T-liveness, where $T = \{t_4, t_5\}$.

The first iteration begins with the PT and AC-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 7(d)). The active subnets are shown in figure 7(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 $b = 1$. Due to the subsequent AC-transformation, the invariant introduced by C_1 has the form $\mu(C_1) = \mu(p_1)+\mu(p_2)+\mu(p_3)-\mu(p_{1,2})-\mu(p_{2,2})-\mu(p_{3,2})$.

In the second iteration, $\{p_1, p_2, p_{2,1}, p_{3,1}, p_{1,2}, p_{2,2}, p_{3,2}, C_1\}$ is the only new minimal active siphon. The siphon is uncontrolled, since $\mu(p_1)+\mu(p_2)+\mu(p_{2,1})+\mu(p_{3,1})+\mu(p_{1,2})+\mu(C_1) \geq 1$, 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 the step D are:

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

There is one redundant constraint, so the final constraints are $L = [2, 2, 1]$ and $b = 2$. The supervised net is shown in figure 7(f). By Theorem 6.2 it is T-live for all initial markings μ_0 such that $L\mu_0 \geq b$. Moreover, by Theorem 6.3, the supervision is maximally permissive. □

Example 5.3 (Liveness enforcement) Consider the repetitive Petri net of figure 8(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

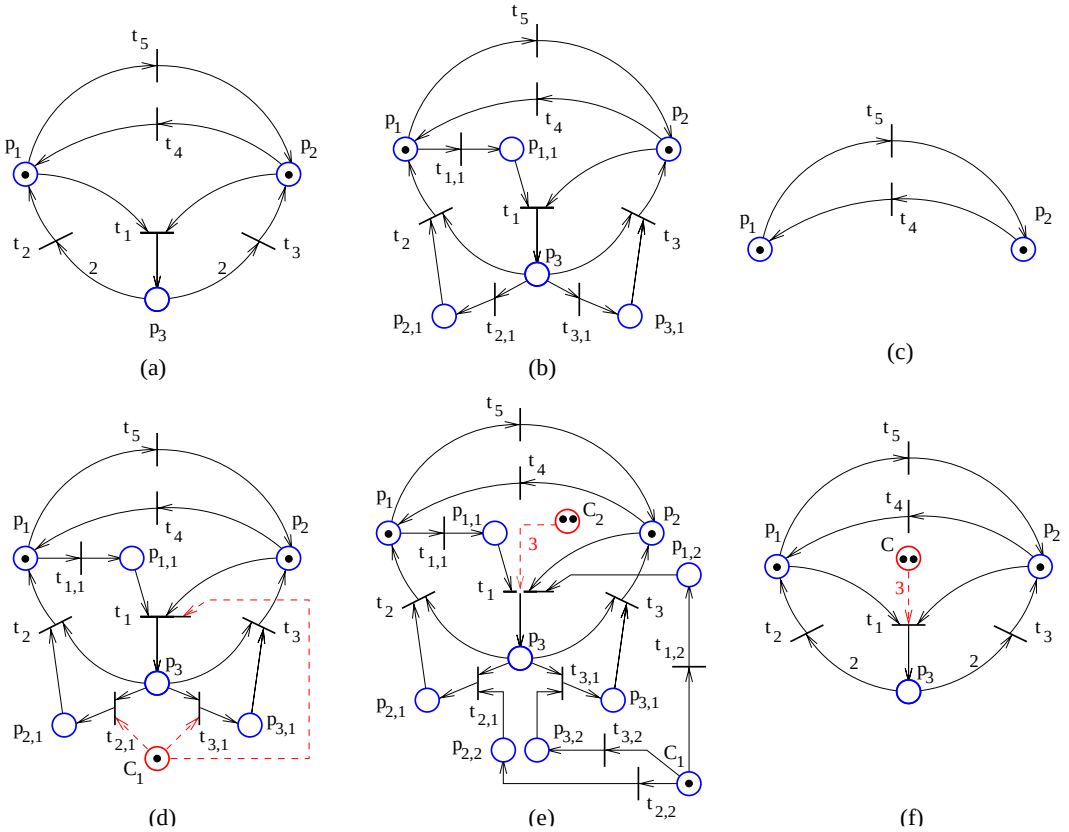

Figure 7: Example 5.2: (a) $\mathcal{N}_0$; (b) $\mathcal{N}_1$; (c) $\mathcal{N}_1^A$, the same as $\mathcal{N}_2^A$ and $\mathcal{N}_3^A$; (d) $\mathcal{N}_2$; (e) $\mathcal{N}_3$ before the split of t_1; (f) the final Petri net supervised for T-liveness

siphon $\{p_1, p_3\}$. The marking constraint $\mu(p_1)+\mu(p_3) \geq 1$ is not admissible, so the approach of section 4.4.2 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:

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

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

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

The procedure terminates, since there is no new uncontrolled siphon in the third iteration. The supervised net is shown in figure 8(b). Liveness is enforced for all initial markings such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$. Moreover, by Theorem 6.3, the supervisor is maximally permissive. □

Example 5.4 (Initial constraints) Consider the Petri net of figure 9 and assume that the initial constraint $\mu(p_1)+\mu(p_2)+\mu(p_3)+\mu(p_4)+\mu(p_5) \leq 1$ is given. This constraint could result, for instance, from the initial

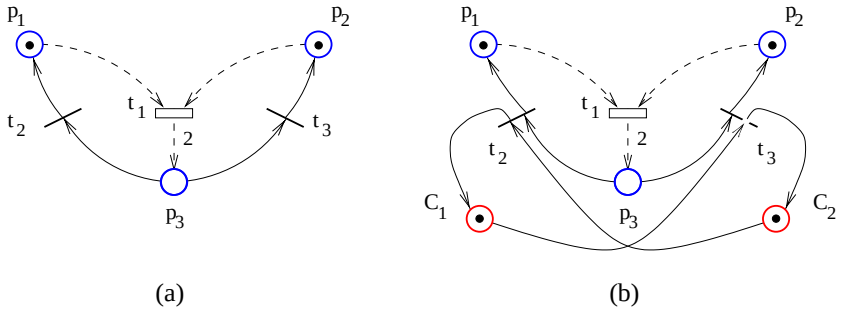

Figure 8: Example 5.3: (a) $\mathcal{N}_0$; (b) the final Petri net supervised for deadlock-freedom

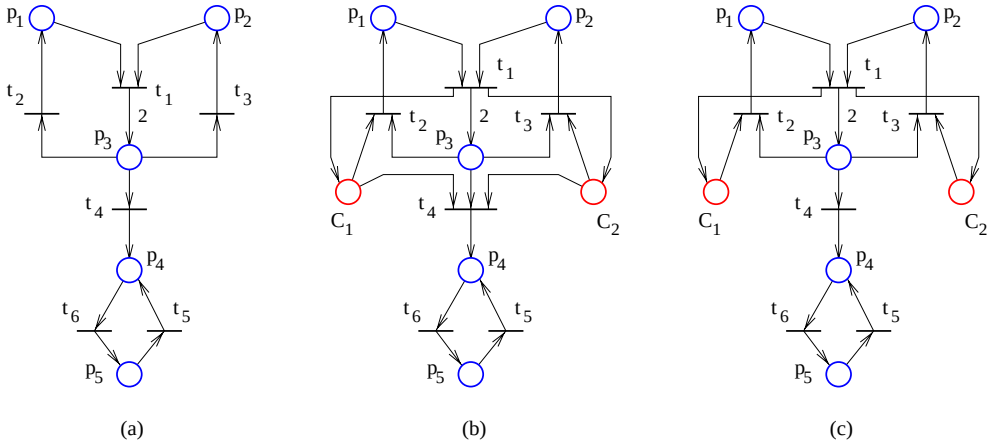

Figure 9: Example 5.3: (a) $\mathcal{N}_0$; (b) the Petri net after the first iteration at the first run; (c) after the first iteration at the second run.

markings which are of interest in some application using this Petri net. The constraint is correct, as if satisfied for μ, it is satisfied for all markings reachable from μ. At the first iteration, the active subnet equals $\mathcal{N}_0$ and there are two minimal active siphons: $\{p_1, p_3\}$ and $\{p_2, p_3\}$. The control places C_1 and C_2 are added to the net. They enforce $\mu(p_1) + \mu(p_3) \geq 1$, and $\mu(p_2) + \mu(p_3) \geq 1$, respectively.

At the second iteration there are two new minimal active siphons: $\{p_2, C_1\}$ and $\{p_1, C_2\}$. The two siphons require the same inequality for their control: $\mu(p_1) + \mu(p_2) + \mu(p_3) \geq 2$. However this conflicts with the initial constraint. Therefore we have a failure at the step C.2.c of the procedure. Thus the parameter X from the procedure become $X = \{t_1, t_2, t_3, t_4\}$, and the procedure is restarted, according to the step C.7.

At the second run of the procedure, because of X, the active subnet has the set of transitions $T_0^A = \{t_5, t_6\}$. There are two minimal active siphons: $\{p_1, p_3, p_4, p_5\}$ and $\{p_2, p_3, p_4, p_5\}$. Controlling them results in the two control places C_1 and C_2 (figure 9(c)).

At the second iteration of the second run there are no new minimal active siphons. Therefore the procedure terminates with

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

and

$$L_0 = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \end{bmatrix} \qquad b_0 = \begin{bmatrix} -1 \end{bmatrix}$$

where the constraints (L_0, b_0) reflect the initial constraints (see step A). The supervised Petri net is T-live for $T = \{t_4, t_5\}$, and is the same as the Petri net of figure 9(c). Also, the supervision is maximally permissive with respect to T-liveness enforcement. □

6 Properties

6.1 Basic Properties of the Method

6.1.1 Introduction and Notations

In the liveness enforcement procedure, we start with a Petri net $\mathcal{N}_0 = (P_0, T_0, F_0, W_0)$ that may not be PT-ordinary or with asymmetric choice. New Petri nets $\mathcal{N}_i = (P_i, T_i, F_i, W_i)$, $i \geq 1$, are derived in the iterative process. The only operations of an iteration that modify the structure of the total net are the addition of a new control place (section 4.4) and transition split (sections 4.1 and 4.2). In general the modifications done in an iteration are such that $\mathcal{N}_i$ can be regarded as a subnet of $\mathcal{N}_{i+1}$. In other words $\mathcal{N}_{i+1}$ is $\mathcal{N}_i$ enhanced with a network of new places and transitions connected to the existing transitions T_i.

The notations of Petri nets which are used 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 $\mathcal{N}_i$.

When a transition is split, one or more of its input arcs are replaced by a sequence of places and transitions. Note that a split transition is not removed from the Petri net (see sections 4.1 and 4.2.) 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 $1, 2, \ldots i$.

We also need notations to specify the transition sequences of $\mathcal{N}_i$ which resulted by successive splits of a single transition of $\mathcal{N}_0$. To state the problem, recall that our procedure recursively finds new deadlock possibilities in a Petri net $\mathcal{N}_i$, then improves it to remove them and a new Petri net $\mathcal{N}_{i+1}$ results. However we are ultimately interested in enforcing liveness in $\mathcal{N}_0$. The places which result by transition splits do not have an equivalent in $\mathcal{N}_0$, therefore we are especially interested in markings of $\mathcal{N}_i$ for which these places have zero marking. Let $\mathcal{M}_i$ denote the set of markings of $\mathcal{N}_i$ with this property. From such markings of $\mathcal{N}_i$ we may not be able to fire directly a transitions t which also appears in $\mathcal{N}_0$. If so, the reason is that t has been split in one or more of the iterations $1, 2, \ldots i-1$, and hence the new transitions created by split must fire first. Therefore we want to characterize the sequences σ such that t appears once in σ, no other transitions of $\mathcal{N}_0$ appear in σ and $\exists \mu_1, \mu_2 \in \mathcal{M}_i$ such that $\mu_1[\sigma > \mu_2$.

For instance, a transition t of $\mathcal{N}_0$ is first split in the first iteration because of the PT-transformation, and so t_1, t_2, t_3, result. In the second iteration t_1 and t_2 are split because of the PT-transformation and the new transitions are $t_{1,1}, t_{1,2}$, and $t_{2,1}, t_{2,2}$. In the third iteration t is again split but with regard to the AC-transformation; the transition $t_{0,1}$ results. 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 $\mathcal{N}_0$. 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 $\mathcal{N}_0$, but the same may be true for other orderings in a sequence of these transitions, for instance for

$t_3t_{2,2}t_{1,2}t_{2,1}t_2t_{1,1}t_1t$. Also in $\mathcal{N}_4$ firing $t_{0,1}$ must occur before firing t, but otherwise is not at all restricted by firing any of $t_3, t_{2,2}, t_{1,2}, t_{2,1}, t_2t_{1,1}, t_1$. So in $\mathcal{N}_4$ we have even more ways to define the transition sequence.

Thus, 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,

$$\{t_1,\, t_2t_3\}t_4$$

defines the sequences $t_1t_2t_3t_4$, $t_2t_1t_3t_4$ and $t_2t_3t_1t_4$; the notation denotes that t_1 and the sequence t_2t_3 can fire asynchronously, but t_4 can fire only after all of t_1 and t_2t_3 have fired.

We define $P_{R,m}$ to be the set of places which appeared by transition splits in the iterations m, $m+1$, $\ldots$ $k-1$. The following algorithm defines the set of transition sequences $\Sigma_{m,k}$ which resulted by successive splits in the iterations m, $m+1$, $\ldots$ $k-1$ of a single transition t of $\mathcal{N}_m$.

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

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 $|\, {\bullet t_x} \cap P_{R,m}| = 0$ **then**
      $I := I \setminus \{i\}$
    else if $|\, {\bullet t_x} \cap P_{R,m}| = 1$ **then**
      Let t_y be the transition such that $\{t_y\} = {\bullet}({\bullet t_x} \cap P_{R,m})$.
      $\Sigma_i := t_y\Sigma_i$
    else if $|\, {\bullet t_x} \cap P_{R,m}| = j > 1$ **then**
      Let $t_{y1}, \ldots t_{yj}$ be the transitions such that $\{t_{y1}, \ldots t_{yj}\} = {\bullet}({\bullet t_x} \cap P_{R,m})$.
      $I := (I \cup \{n+1, \ldots n+j\}) \setminus \{i\}$
      $\Sigma_{n+1} := t_{y1}$, $\ldots$ $\Sigma_{n+j} := t_{yj}$
      $\Sigma_i := \{\Sigma_{n+1}, \ldots \Sigma_{n+j}\}\Sigma_i$
      $n := n + j$
    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 a sequence $\sigma_{m,k}(t)$ is defined to contain t, and it ends with t. Important properties of the sequences $\sigma_{0,k}(t)$ are given in Propositions 6.13 and 6.14.

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, we introduce 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

Proposition 6.1 *Let $\mathcal{N}_k^A$ and $\mathcal{N}_k$ be the* the active subnet *and the* total subnet *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*, $\mathcal{N}_k^A$. Then S is controlled by C in the* active subnet *(considered as an independent net.)*

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 4.3). Because enforcing l_a ensures that S is controlled (section 4.4.2), the conclusion follows. □

Several properties also 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 4.4.2) 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 4.4.1) 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. □

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 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_{Sx} 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_{Sx}. Also, because $(\bullet\bullet P_{Sx} \setminus P_{Sx}) \cap P_S = \emptyset$ and $(P_{Sx} \bullet\bullet \setminus P_{Sx}) \cap P_S = \emptyset$ there is no cyclic structure only made up of places in P_{Sx} and other places from P_S. The same justification also applies to the active subnet. □

The algorithm of the AC-transformation in section 4.2 is obvious in the case when the second argument M satisfies $M = P$. However when $M \subset P$ the algorithm should not be in the situation that at step 4 no choice can be made, as $\{p_i, p_j\} \cap M = \emptyset$. If this happens, the final Petri net $\mathcal{N}'$ is not asymmetric choice. The next result proves that the liveness enforcement procedure uses the AC-transformation in a right way.

Proposition 6.4 *The step C:4 of the liveness enforcing procedure produces an asymmetric choice net.*

Proof: Let $\mathcal{C}$ be the set of the control places added in the current iteration and i the number of the iteration. Let P_R be the set of places resulted in the PT-transformation of the step C:3 and let $\mathcal{N}$ be the Petri net obtained after the step C:3. We consider the algorithm of the AC-transformation and show that in the step 4, $\{p_i, p_j\} \cap M \neq \emptyset$. (Note that $M = \mathcal{C}$ in the step C:4.) In order to do this we show that if $p_i \bullet \cap p_j \bullet \neq \emptyset$, $p_i\bullet \not\subseteq p_j\bullet$ and $p_j\bullet \not\subseteq p_i\bullet$, then $\{p_i, p_j\} \cap \mathcal{C} \neq \emptyset$. In this proof the $\bullet$ operator is taken with respect to $\mathcal{N}$.

Let p_l and p_j be such that $p_l \bullet \cap p_j \bullet \neq \emptyset$, $p_l\bullet \not\subseteq p_j\bullet$ and $p_j\bullet \not\subseteq p_l\bullet$. Note that $p_l, p_j \notin P_R$, because $p \in P_R \Rightarrow |p \bullet| = 1$ (see section 4.1). Since $\mathcal{N}_i$ is PT-ordinary, the postset of any place $p \in P_i$ is unchanged by the operations done in the iteration i, in particular by the PT-transformation. Therefore, since $\mathcal{N}_i$ is with asymmetric choice, it is also true in $\mathcal{N}$ that $\forall p_u, p_v \in P_i$: $p_u \bullet \cap p_v \bullet \neq \emptyset \Rightarrow p_u\bullet \subseteq p_v\bullet$ or $p_v\bullet \subseteq p_u\bullet$. Therefore

not both p_l and p_j are in P_i. Since they cannot be in P_R, it follows that $\{p_i, p_j\} \cap \mathcal{C} \neq \emptyset$. Therefore the outcome of the AC-transformation is an asymmetric choice Petri net, for there is always a possible choice at the step 4 of the AC-transformation algorithm. □

The next proposition considers the update of the active subnets as described in section 5.3. It shows that the update produces indeed an active subnet.

Proposition 6.5 *Consider the update algorithm described in section 6.5. The algorithm produces an active subnet.*

Proof: Using the notations from the algorithm, we are to prove that if $\mathcal{N}_{i-1}^A$ is an active subnet then $\mathcal{N}_i^A$ also is. Let D_{i-1} and D_i be the incidence matrices of $\mathcal{N}_{i-1}$ and $\mathcal{N}_i$. By Definition 3.5, there is $x \geq 0$ such that $D_{i-1}x \geq 0$ and $T_{i-1}^A = \|x\|$. The modifications of $\mathcal{N}_{i-1}$ are done in two stages: first the control places are added and then transitions are split. After control places are added, the new incidence matrix is $D'_{i-1} = [D_{i-1}^T, (L_i D_{i-1})^T]^T$, where L_i denotes the new inequalities $L_i\mu \geq b_i$ enforced with control places in the iteration $i-1$. The liveness procedure adds only constraints such that $L_i \geq 0$. Therefore $D'_{i-1}x \geq 0$. After the transitions to be split are split, the new incidence matrix is D_i. Let P_R be the new places which appeared by transition split and let $\mathcal{N}'_{i-1}$ be the Petri net corresponding to D'_{i-1}. There is a marking μ and a sequence σ such that all transitions of $\|x\|$ appear infinitely often in σ, no other transitions than $\|x\|$ appear in σ and μ enables σ in $\mathcal{N}'_{i-1}$. Let μ_i be a marking of $\mathcal{N}_i$ such that $\mu_i(p) = \mu(p)$ for all places p of $\mathcal{N}'_{i-1}$ and $\mu_i(p) = 0$ for all other places (i.e. those in P_R). Then μ_i enables $\sigma_{i-1,i}(\sigma)$. The transitions which appear infinitely often in $\sigma_{i-1,i}(\sigma)$ are the transitions in T_i^A. By Lemma 3.1 there is $x^* \geq 0$ such that $Dx \geq 0$ and $\|x\| = T_i^A$. Therefore $\mathcal{N}_i^A$ is an active subnet. □

Proposition 6.6 *Given a PT-ordinary Petri net, let S be (i) a (minimal) siphon, or (ii) a (minimal) active siphon.*

(a) 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).

(b) Assume that a transition t is split with regard to the AC transformation. Let p' and t' be the new place and transition obtained by splitting t and $\mathcal{N}'$ the Petri net after the split. Let $S' = S \cup \{p'\}$ if $t \in {\bullet}S$ and $t \notin S{\bullet}|_{\mathcal{N}'}$, and $S' = S$ otherwise. Then S' is a (minimal) siphon in case (i), or a (minimal) active siphon in the case (ii).

Proof: (a) See Proposition 6.4 in [9].

(b) We only consider the less obvious case when $t \in {\bullet}S$ and $t \notin S{\bullet}|_{\mathcal{N}'}$. Let $\mathcal{N}$ be the Petri net before splitting t. Note that $S'{\bullet} = S{\bullet} \cup \{t'\}$ and ${\bullet}S' = {\bullet}S \cup \{t'\}$. Therefore ${\bullet}S' \subseteq S'{\bullet}$. So S' is a siphon. Assume that $s' \subset S'$ is another siphon. There are two cases: $p' \notin s'$ and $p' \in s'$. We show that both cases lead to the conclusion that the siphon S is not minimal in $\mathcal{N}$. If $p' \notin s'$, then ${\bullet}s'$ is the same in $\mathcal{N}$ and $\mathcal{N}'$ ($t' \notin {\bullet}s'$) and $s'{\bullet}$ in $\mathcal{N}$ and $\mathcal{N}'$ may differ only by t'. Therefore s' is also a siphon of $\mathcal{N}$. Since $p' \notin s'$: $s' \subseteq S$. However $s' \subset S$, because $s' = S$ and ${\bullet}S \setminus S{\bullet}|_{\mathcal{N}'} \neq \emptyset$ contradict that s' is a siphon of $\mathcal{N}'$. If $p' \in s'$, then let $s = s' \setminus \{p'\}$; ${\bullet}s' = {\bullet}s \cup \{t'\}$ and $s'{\bullet} = s{\bullet} \cup \{t'\}$ imply ${\bullet}s \subseteq s{\bullet}$, so s is a siphon of $\mathcal{N}$ and by construction $s \subset S$. Therefore S' is minimal if S is minimal.

The same proof can be used to show that if S is a (minimal) siphon of the active subnet of $\mathcal{N}$, then S' is a (minimal) siphon of the active subnet of $\mathcal{N}'$. Now we assume that S is an active siphon. Let s be a siphon

of $\mathcal{N}^A$ (the active subnet of $\mathcal{N}$) such that $s \subseteq S$. Let s' be defined the same way as above. Then s' is also a siphon of ${\mathcal{N}'}^A$. But $s' \subseteq S'$, so S' is an active siphon of $\mathcal{N}'$. If S' is not a minimal active siphon, let $S'_1 \subset S'$ be an active siphon. If $p' \notin S'_1$, then S'_1 is also an active siphon of $\mathcal{N}$, and so S is not minimal, as $S'_1 \subseteq S$ and $S'_1 \neq S$ (S cannot be a siphon of $\mathcal{N}'$ because $\bullet S \setminus S \bullet |_{\mathcal{N}'} \neq \emptyset$). Else, if $p' \in S'_1$, then $S'_1 \setminus \{p'\}$ is an active siphon in $\mathcal{N}$, and from $S'_1 \setminus \{p'\} \subset S$ it can be seen that S is not a minimal active siphon. Therefore if S is a minimal active siphon, S' also is. □

The last result shows that the transition split operation does not change the siphons of a Petri net and their type, when done with regard to a PT-transformation. When done with regard to an AC-transformation, the transition split operation may change the siphons by adding some of the new places to them, however without changing their type. Note that during the iterations, if S is a siphon of $\mathcal{N}_i$ and $i \geq 1$, it is not possible to split $t \in \bullet S$ in view of the AC-transformation (see the proof of Proposition 6.4). Therefore we have the following result.

Corollary 6.1 *Let S be a (minimal) siphon of $\mathcal{N}_i$. Then S is a (minimal) siphon of $\mathcal{N}_{i+1}$. Furthermore, assume that no transitions are marked to be removed from the active subnet in the iteration i. Then if S is a (minimal) active siphon in $\mathcal{N}_i$, S also is a (minimal) active siphon in $\mathcal{N}_{i+1}$.*

Proof: The proof results immediately by successively applying Proposition 6.6 for every transition split and by the observation that no transition in T_i is split in view of the AC-transformation (see the proof of Proposition 6.4). □

Since the transition split operations modify the Petri net, it is important to establish how the invariants of a Petri net are changed.

Proposition 6.7 *Let $l^T\mu \geq b$ be true for all markings reachable from a set $\mathcal{M}_I$ of markings of $\mathcal{N}_i$. 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 $\mu_0 \in \mathcal{M}_I$. For all markings μ reachable from μ_0 and such that $\mu(p) = 0\ \forall p \in P_R$, $l^T\mu \geq b$ 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: This is a direct consequence of the following facts: (a) $l^T\mu \geq b$ is enforced in $\mathcal{N}_i$ for all markings reachable from markings in $\mathcal{M}_I$; (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 sections 4.1 and 4.2.) □

Proposition 6.8 *Assume that a number of constraints are enforced on a PT-ordinary Petri net $\mathcal{N}$. 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, $\ldots t_k$ 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 $\mathcal{N}'$ as shown in section 4.1. Then C enforces:*

$$\mu_e(C) + \sum_{i=1}^{k} \sum_{j=1}^{m_i - 1} j\mu_e(p_{i,m_i-j}) = l^T\mu - b \tag{22}$$

in the PT-transformed Petri net, where μ is the marking vector μ_e restricted to $\mathcal{N}$ and the usual notations of section 4.1 are used.

Furthermore the AC-transformation of section 4.2 is applied to $\mathcal{N}'$, with the parameter M equal to the set of control places added to $\mathcal{N}$. Then C enforces:

$$\mu_e(C) + \sum_{z=1}^{r} \mu_e(p_z) + \sum_{i=1}^{k} \sum_{j=1}^{m_i-1} j\mu_e(p_{i,m_i-j}) = l^T\mu - b \tag{23}$$

where $p_1, p_2 \dots p_r$ are all places created by the AC-transformation which satisfy $\bullet\bullet p_z = C$ and μ is the marking vector μ_e restricted to $\mathcal{N}$.

Proof: For the first part, see Proposition 6.6 in [9]. The second part results in a similar way, by noticing that the choice of M ensures that for all transitions split in view of the AC-transformation, all new places p_x which appear satisfy $\bullet\bullet p_x = C_y$ for some control place C_y. □

The last result shows that the PT-transformation done to an intermediary Petri net, in which only the new control places cause the Petri net not to be PT-ordinary, does not affect the enforced inequalities. Indeed, equation (22) implies that $l^T\mu \geq b$ is still true in the PT-transformed net. This is also true about the equation (23). As Proposition 6.8 applies to the PT-transformation and the AC-transformation performed by the liveness procedure in each iteration, it follows that the enforced inequalities stay enforced after these two transformations are applied.

Further on we prove a more general result. We consider an inequality or a set of inequalities to be **enforced** with respect to a set of initial markings $\mathcal{M}_I$ if they are true for all markings reachable from the markings in $\mathcal{M}_I$.

Proposition 6.9 *Assume that $l^T\mu \geq b$ (or $l^T\mu = b$) is enforced in a Petri net $\mathcal{N}$ for all initial markings in a set $\mathcal{M}_I$.*

(a) Consider that a transition t_x is split in view of the PT-transformation and let $\mathcal{N}'$ be the Petri net after t_x is split. Consider the inequality (equality) $l_1^T\mu \geq b$ (or $l_1^T\mu = b$), where $l_1^T\mu$ is obtained as follows: for all places p which in $\mathcal{N}$ satisfy $p \in \bullet t_x$ and $W(p, t_x) = m_p > 1$, replace $\mu(p)$ in $l^T\mu$ with $\mu(p) + \sum_{i=1}^{m_p-1} i\mu(p_{x,m_p-i})$, where $p_{x,i}$ are the places resulted by splitting t_x (section 4.1). Then $l_1^T\mu \geq b$ ($l_1^T\mu = b$) is enforced for all initial markings μ_0 such that $\mu_0(p_{x,i}) = 0$ for all places $p_{x,i}$ and $\mu_0|_{\mathcal{N}} \in \mathcal{M}_I$.

(b) Consider that a transition t_x is split in view of the AC-transformation. Let p' and t' be the new place and transition which result. Then the inequality (equality) becomes $l_1^T\mu \geq b$ ($l_1^T\mu = b$), where $l_1(p) = l(p)$ for all $p \neq p'$ and $l_1(p') = l(p_x)$, where $\bullet t' = \{p_x\}$ and $l_1^T\mu \geq b$ ($l_1^T\mu = b$) is enforced for all initial markings μ_0 such that $\mu_0(p') = 0$ and $\mu_0|_{\mathcal{N}} \in \mathcal{M}_I$.

(c) Let $P_l = \{p : l(p) \neq 0\}$. Assume that $\mathcal{N}$ satisfies: $\forall p \in P_l\ \forall t \in \bullet p$: $W(p, t) = 1$. Consider applying the PT-transformation and then the AC-transformation, the latter with the parameter M such that $M \cap P_l = \emptyset$. Let $\mathcal{N}'$ be the obtained Petri net. Then $l^T\mu \geq b$ ($l^T\mu = b$) is enforced in $\mathcal{N}'$ for all initial markings μ_0 such that $\mu_0(p) = 0$ if p is not in $\mathcal{N}$ and $\mu_0|_{\mathcal{N}} \in \mathcal{M}_I$.

Proof: (a) We denote by μ' any of the markings of $\mathcal{N}'$ and by μ the markings of $\mathcal{N}$. Let f be a map such that if $f(\mu') = \mu$, then $\mu(p) = \mu'(p)\ \forall p \notin \{p \in \bullet t_x : W(p, t_x) \geq 1\}$ and $\mu(p) = \mu'(p) + \sum_{i=1}^{m_p-1} i\mu(p_{x,m_p-i})$ $\forall p \in \{p \in \bullet t_x : W(p, t_x) \geq 1\}$. Note that $l_1^T\mu' = l^T f(\mu')$. Let T_R be the set of new transitions resulted through the split. Note that $\mu_1'[t > \mu_2'$ implies $f(\mu_1') = f(\mu_2')$ for $t \in T_R$, and $\mu_1'[t > \mu_2'$ for $t \notin P_R$ implies $f(\mu_1')[t > f(\mu_2')$. Let μ_0' be a marking such that $\mu_0'(p') = 0$ and $\mu_0'|_{\mathcal{N}} \in \mathcal{M}_I$. Let σ' be an arbitrary firing

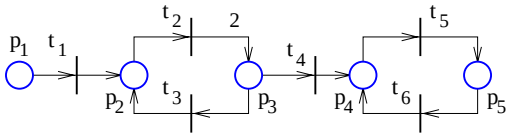

Figure 10: Example for Proposition 6.10

sequence and σ its projection to the transitions of $\mathcal{N}$. Then $\mu_0'[\sigma' > \mu_1'$ implies $f(\mu_0')[\sigma > f(\mu_1')$. But $f(\mu_0') = \mu_0'|_{\mathcal{N}}$, therefore $l^T f(\mu_1') \geq b$ and so $l_1^T \mu_1' \geq b$. Since the sequence σ' is arbitrary, the conclusion follows.

(b) Let $\mathcal{N}'$ be the transformed net. We denote by μ' any of the markings of $\mathcal{N}'$ and by μ the markings of $\mathcal{N}$. Let f be a map such that if $f(\mu') = \mu$, then $\mu(p) = \mu'(p)\ \forall p \neq p_x$ and $\mu(p_x) = \mu'(p') + \mu'(p_x)$. Note that $l_1^T \mu' = l^T f(\mu')$. Also $\mu_1'[t' > \mu_2'$ implies $f(\mu_1') = f(\mu_2')$, and $\mu_1'[t > \mu_2'$ for $t \neq t'$ implies $f(\mu_1')[t > f(\mu_2')$. Further on the proof can be done exactly as at part (a): μ_0' is chosen as at (a), in the same way an arbitrary sequence σ' is considered and then the conclusion follows.

(c) This is a consequence of (a) and (b). Indeed, by (a) the PT-transformation does not change $l^T \mu \geq b$ ($l^T \mu = b$) as no transition $t \in P_l\bullet$ is split. Also by (b) the AC-transformation does not change $l^T \mu \geq b$ ($l^T \mu = b$), as the new places p_x added by the transformation must satisfy $\bullet\bullet p_x \in M$. □

The importance of Proposition 6.9 is that it shows how the enforced constraints are modified by transition split operations. Furthermore, part (c) shows that the constraints of the form (L_0, b_0) are not changed by the iterations of the liveness procedure. They may be changed only by the first PT-transformation and the first AC-transformation used before the first iteration, where the change which may occur is described by the parts (a) and (b) of the proposition.

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: the implementation of the procedure does not need to compute all minimal active siphons (which could be computationally expensive.)

Proposition 6.10 *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: By construction (step C.4 of the procedure and the algorithm in section 4.2), if the place p resulted from a transition split of the AC-transformation of the iteration i, then there is a control place C added in the same iteration such that $\bullet\bullet p = C$. Thus any siphon which contains a place resulted from the AC-transformation of the iteration i must also contain a control place added in the iteration i. For the rest the proof is similar to that of Proposition 6.8 in [9]. □

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 $\mathcal{N}_1$, $\mathcal{N}_2$, $\mathcal{N}_3$, ... 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 the control places satisfies the invariants they 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 have empty siphons 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 6.11 *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: This is Proposition 6.9 of [9], whose proof still applies. □

Proposition 6.12 *The equivalence of markings is an equivalence relation.*

Proof: The proof is immediate by checking the symmetry, reflexivity and transitivity of the relation. □

Proposition 6.13 *Let μ be a valid marking of $\mathcal{N}_k$, σ an enabled firing sequence and $t \in T_0$. Assume that t appears in σ. Then all transitions of $\sigma_{0,k}(t)$ other than t appear in σ before the first occurrence of t in σ; let s be the sequence in which they appear before the first occurrence of t in σ. There is a subsequence s_0 of s such that the sequence $s_0 t$ equals a $\sigma_{0,k}(t)$.*

Proof: Let P_R be the set of places resulted through split operations in the iterations $1 \ldots k-1$. The marking μ is valid, so t cannot be fired unless the places $\bullet t \cap P_R$ are marked, and they cannot become marked by firing

other transitions in T_R except $\bullet(\bullet t \cap P_R)$, which are all transitions of $\sigma_{0,k}(t)$. Next, let $T_{x1} = \bullet(\bullet t \cap P_R)$. The transitions of T_{x1} cannot fire unless the places $\bullet T_{x1} \cap P_R$ are marked, which cannot happen unless the transitions in $\bullet(\bullet T_{x1} \cap P_R)$ fire before. Let $T_{x2} = \bullet(\bullet T_{x1} \cap P_R)$. We continue in the same way until we get $T_{xk} = \emptyset$. Hence all transitions of $\sigma_{0,k}(t)$ other than t appear in σ before the first occurrence of t in σ.

Given a transition t_i, we let $T_x(t_i) = \bullet(\bullet t_i \cap P_R)$. Let t_1 be the last transition from $T_x(t)$ which appears in s before t. Let t_2 be the last transition from $(T_x(t) \cup T_x(t_1)) \setminus \{t_1\}$ which appears in s before t_1. Let t_3 be the last transition from $(T_x(t) \cup T_x(t_1) \cup T_x(t_2)) \setminus \{t_1, t_2\}$ which appears in s before t_2. We continue this way until t_m such that $(T_x(t) \cup \bigcup_{i=1}^{m} T_x(t_i)) \setminus \{t_1, t_2, \ldots t_m\} = \emptyset$. Let s_0 be the sequence $t_m, t_{m-1}, \ldots t_1, t$. The construction of s_0 is in accord with the algorithm defining $\sigma_{0,k}(t)$, therefore s_0 is a sequence of the form $\sigma_{0,k}(t)$. □

Corollary 6.2 *Let s be a transition sequence which contains $t \in T_0$. Assume that s is not longer than a sequence $\sigma_{0,k}(t)$. If s is enabled by a valid marking of $\mathcal{N}_k$, s is a sequence $\sigma_{0,k}(t)$.*

Proposition 6.14 *If a valid marking of $\mathcal{N}_k$ enables a sequence $\sigma_{0,k}(t)$, then it enables all sequences $\sigma_{0,k}(t)$.*

Proof: Let σ_1 and σ_2 be two sequences $\sigma_{0,k}(t)$ such that σ_1 is enabled by the marking μ and σ_2 is not. Let $\sigma_2 = t_1 t_2 \ldots t_m t$. Consider firing σ_2 from the marking μ. As σ_2 is not enabled, there is $i < m$ such that after firing $t_1 t_2 \ldots t_{i-1}$, the transition t_i is not enabled. Let μ_i be the reached marking. Let P_R be the set of places resulted by transition split in the iterations $1, 2, \ldots k-1$. Note that $\forall p \in \bullet t_i \cap P_R$, $\mu_i(p) = 1$, for σ_2 cannot be a sequence $\sigma_{0,k}(t)$ unless the transitions in $\bullet(\bullet t_i \cap P_R)$ precede t_i in σ_2, and t_i is the only transition in the postset of $\bullet t_i \cap P_R$ (see the split transition construction.) Therefore $\exists p_i \in P_k \setminus P_R$ such that $\mu_i(p_i) = 0$. Note that by Proposition 6.2(a) and (b) firing any transition of $\sigma_{0,k}(t)$ different than t never increases the marking of a place in $P_k \setminus P_R$. Thus, if we allow the markings to go negative, we can fire the remaining transitions $t_i t_{i+1} \ldots t_m$, and so reach a marking μ_m such that $\mu_m(p_i) < 0$. Since σ_1 contains the same transitions as σ_2 but in a different order, (let $\sigma_1 = t'_1 t'_2 \ldots t'_m t$) after firing the first m transitions of σ_1 from μ, the same marking μ_m is reached. As $\mu_m(p_i) < 0$, it follows that σ_1 is not enabled by μ, which is a contradiction. □

In what follows we state a number of results which we have been derived for the deadlock prevention procedure of [9] and which still apply for the liveness procedure. The results are useful for a good understanding of the liveness procedure. Most of them are also used in the proofs of the main results in section 6.2.

Proposition 6.15 *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: This result is Proposition 6.12 in [9]; the proof does not change. □

Proposition 6.16 *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: This result is Proposition 6.13 in [9], whose proof applies without changes. □

Corollary 6.3 *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: This is Corollary 6.1 in [9]. The proof does not change. □

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: This is Theorem 6.1 in [9], whose proof applies without modification. □

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 and AC-transformed version of $\mathcal{N}_0$.

Proposition 6.17 *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: This is Proposition 6.14 in [9], whose proof still applies. □

The next proposition considers the constraints (L_0, b_0) at a iteration i. It shows that even though they are not enforced by control places, they are satisfied by the Petri nets resulted in subsequent iterations.

Proposition 6.18 *Let (L_i, b_i) and $(L_{0,i}, b_{0,i})$ be the constraints (L, b) and (L_0, b_0) at the end of the iteration i. Let (L', b) and (L_0', b_0) be (L_i, b_i) and $(L_{0,i}, b_{0,i})$ restricted to the places of $\mathcal{N}_0$. Let μ_0 be an arbitrary initial*

marking of $\mathcal{N}_0$ satisfying $L'_0\mu_0 \geq b_0$ and $L'\mu_0 \geq b$. Then, if $(\mathcal{N}_0, \mu_0)$ is supervised according to $L'\mu \geq b$, for all reachable markings μ: $L'_0\mu \geq b_0$.

Proof: Assume the contrary. Let μ_x be a reachable marking such that $L'_0\mu_x \ngeq b_0$, μ_x is reached by firing t from μ_y, and $L'_0\mu_y \geq b_0$. Let $l'_1\mu \geq b_1$ be one of the constraints of (L'_0, b_0) which are not satisfied by μ_x. Let (l_1, b_1) be the constraint which appeared in some iteration $j \leq i$ such that its restriction is (l'_1, b_1). This means that for every marking μ of $\mathcal{N}_j$ such that $l_1\mu \geq b_1$: $l_1\mu_z \geq b_1\ \forall \mu_z \in \mathcal{R}(\mathcal{N}_j, \mu)$. Let $\mu_{x,j}$ and $\mu_{y,j}$ be the valid marking of $\mathcal{N}_j$ equivalent to μ_x and μ_y. Then by Proposition 6.16: $\mu_{y,j}[\sigma_{0,j}(t) > \mu_{x,j}$. As $\mu_{x,j}$ is valid, $l_1\mu_{x,j} \geq b_1 \Rightarrow l'_1\mu_x \geq b$, which contradicts the initial assumption. □

6.2 Main Results

This section proves that the procedure enforces liveness in Theorem 6.2. Maximally permissivity (under appropriate conditions) is proved in Theorem 6.3. An extension for maximally permissivity in a case when this is not guaranteed by Theorem 6.3 is given in section 6.2.2. Termination results are provided in Theorem 6.5 and Theorem 6.6. The termination results assume certain modifications of the procedure, as described in section 6.2.3.

Throughout the section we use the notations from the description of the procedure in section 5.4 and the notations defined in section 6.1.1. That is, in every iteration i the *active subnet* is denoted by $\mathcal{N}_i^A = (P_i^A, T_i^A, F_i^A, W_i^A)$ and the *total net* $\mathcal{N}_i = (P_i, T_i, F_i, W_i)$, $\sigma_{i,j}(\sigma)$ a replacement sequence in $\mathcal{N}_j$ of the transition sequence σ of $\mathcal{N}_i$, $i < j$ and $\sigma_{i,j}(t)$ a 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 terminates in $k-1$ iterations and that $\mathcal{N}_k^A$ is nonempty. Then $\mathcal{N}_k$ is T_k^A-live for all valid initial markings.*

Proof: In every iteration, the new minimal active siphons are controlled in step 2 of the procedure. The procedure terminates when $\mathcal{N}_k$ is asymmetric choice and all minimal active siphons are controlled. Since $\mathcal{N}_k$ is PT-ordinary and $\mathcal{N}_k^A$ is T_k^A-minimal, $\mathcal{N}_k$ is T_k^A-live for all valid initial markings by Proposition 3.6. □

Theorem 6.2 *Assume that the procedure terminates. Let $\mathcal{N}_0 = (P_0, T_0, F_0, W_0)$ be the original Petri net, $T \subseteq T_0$ the parameter passed to the procedure as the set of transitions which are to be live and $\mathcal{N}_k$ the net produced by the last iteration. Let (L, b) and (L_0, b_0) denote the two sets of constraints generated by the procedure. If $\mathcal{N}_k^A$ is nonempty, then the original net $\mathcal{N}_0$ in closed loop with the supervisor enforcing $L\mu \geq b$ is T_x-live for all initial markings μ_0 of $\mathcal{N}_0$ such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$, where $T_x = T_k^A \cap T_0$. Moreover, if no siphon control failures occurred in the steps C:2(b) and (c) of the procedure, the closed loop is T_y-live for $T_y = T_0^A \cap T$ and the initial markings μ_0 satisfying $L\mu_0 \geq b$ 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 not empty. For such a marking $\mathcal{N}_k$ is T_k^A-live, by Lemma 6.1.

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 the transition $t \in T_0 \cap T_k^A$ is dead. We show that this leads to contradiction.

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 a transition sequence σ in $\mathcal{N}_k$ which includes the transitions of $\sigma_{0,k}(t)$. Let T_R be the set of transitions that appeared by split transition operations in all iterations. 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. By Proposition 6.13, σ contains a $\sigma_{0,k}(t_1)$. Since all transition of σ before t_1 are in T_R, and firing them only decrease markings of $P_0 \cup \mathcal{C}$, $\sigma_{0,k}(t_1)$ is enabled by μ_k, since it is enabled after firing the transitions that precede it in σ. Let t_2 be the next transition of σ in T_0. Similarly, $\sigma_{0,k}(t_1)\sigma_{0,k}(t_2)$ is enabled by μ_k. We continue this way and eventually find t_j in σ and in T_0 such that $t_j = t$. We have that μ_k enables $\sigma_{0,k}(t_1)\sigma_{0,k}(t_2)\ldots\sigma_{0,k}(t_j)$. But this implies that μ enables $t_1 t_2 \ldots t_j$ (by Theorem 6.1(a)), and since $t_j = t$, t is not dead in $\mathcal{N}_S$, which is a contradiction.

If no siphon control failures occur, the active subnets employed in the following iterations are not recomputed, but updated as discussed in section 5.3. Therefore $T_0^A \subseteq T_k^A$ and so $T_0^A \subseteq T_x$, which implies that $\mathcal{N}_S$ also is T_y-live, for $T_y = T \cap T_0^A$ and μ_0 satisfying $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$. □

Note that T-liveness cannot be enforced if $T \not\subseteq T_0^A$, for if this is the case, the structural properties of the net do not allow some of the transitions of T to be live. If the problem is well formulated and thus T-liveness is possible, the procedure will enforce T-liveness if no siphon control failures occur. This always is the case for Petri nets without uncontrollable and unobservable transitions, assuming that the initial constraints are not preventing the enforcement of T-liveness (that is, assuming that the liveness enforcement problem is well formulated).

The assumptions of Theorem 6.2 are that the procedure terminates and that the final active subnet $\mathcal{N}_k^A$ is not empty. The next result characterizes the cases when the procedure terminates and it does not enforce T-liveness. It shows that when there are no uncontrollable and unobservable transitions, if the procedure does not enforce T-liveness, T-liveness is impossible, given the initial constraints (if any are given). Note that $\mathcal{N}_k^A$ empty implies that even deadlock prevention is impossible, if no uncontrollable and unobservable transitions are present.

Proposition 6.19 *T-liveness is not enforcible in $\mathcal{N}_0$ for any initial marking if $T \not\subseteq T_0^A$. Furthermore, in the conditions of Theorem 6.2, if $\mathcal{N}_0$ has no uncontrollable and unobservable transitions and $T \not\subseteq T_k^A$, then T-liveness is not enforcible in $\mathcal{N}_0$ for any initial marking.*

Proof: The first part is a consequence of Corollary 3.2, as the algorithm of section 5.3 does not fail to find a T-minimal active subnet if such a subnet exists. The second part is a consequence of Lemma 6.2, as we show in what follows. For the second part we assume $T \subseteq T_0^A$ and $T \not\subseteq T_k^A$. Consider that in iteration j the first siphon control failure occurs. The failure occurs because there is an active siphon S_x of $\mathcal{N}_j$ which due to the initial constraints must be empty for all valid markings. Using the same idea as in the proof of Theorem 6.3, no transition $t \in S_x\bullet$ can be live in $\mathcal{N}_j$ for valid initial markings; so there are transitions in $T_j^A \cap T_0$ which cannot be made live in $\mathcal{N}_j$, namely the transitions t_x such that $\exists t \in \sigma_{0,j}(t_x)$: $t \in S_x\bullet$. Let $T_l \subset T_j^A$ be the set of all transitions which can be made live in $\mathcal{N}_j$. As in the proof of Theorem 6.3, $T \subseteq T_l$ is not possible, as it would imply that $\mathcal{N}_0^A$ is not T-minimal. (Indeed there is a firing sequence σ_l in which only the transitions of T_l appear infinitely often and there is μ_0 which enables σ_l in $\mathcal{N}_j$. We may assume μ_0 to be valid. If μ_0 is not valid, there is a finite sequence σ_x and the valid marking μ_0' such that

$\mu_0'[\sigma_x > \mu_0$, and we can let $\sigma_l' = \sigma_x\sigma_l$. Further on we continue as in the proof of Theorem 6.3). Therefore $T \setminus T_l \neq \emptyset$. Next we assume there is an infinite sequence σ of $\mathcal{N}_0$ including infinitely often all transitions of T and enabled by a marking μ_0, such that μ_0 and all reachable markings obtained by firing σ satisfy the initial constraints. Then $\sigma_j = \sigma_{0,j}(\sigma)$ is enabled by $\mu_{0,j}$, the marking equivalent to μ_0 in $\mathcal{N}_j$, and $\mu_{0,j}$ as well as the other markings reached by firing σ_j satisfy the initial constraints. (Indeed, this can be easily verified for the markings generated by $\sigma_1 = \sigma_{0,1}(\sigma)$ in $\mathcal{N}_1$; for σ_j it results from the facts that the initial constraints have the same form in $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_j$ (by Proposition 6.9(c)), and the marking of the places P_1 in $\mathcal{N}_j$ depends only on firing transitions in T_1.) Therefore the transitions which appear infinitely often in σ_j should be a subset of T_l. This contradicts $T \setminus T_l \neq \emptyset$. Therefore not all transitions of T can be made live in $\mathcal{N}_0$. □

As shown in section 4.4, 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 $\alpha_p \neq 0\ \forall p \in S$. 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 $1 \ldots i$ ($i \geq 1$) 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 a 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 $(\mathcal{N}_i, \mu_s)$ has at least one empty active siphon.*

Proof: Let $\mathcal{C}$ be the set of control places added to $\mathcal{N}_{i+1}$. The set of places which result through transition split is $P_R = P_{i+1} \setminus (P_i \cup \mathcal{C})$. Note that given an arbitrary firing sequence σ_x of $\mathcal{N}_i$, if $\mu_{i+1}[\sigma_{i,i+1}(\sigma_x) > \mu_x$, as μ_{i+1} is valid, $\mu_x(p) = 0\ \forall p \in P_R$. Let σ be the firing sequence that was used to reach μ: $\mu_i[\sigma > \mu$. Consider firing σ in $(\mathcal{N}_i, \mu_i)$ 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 which is controlled in $\mathcal{N}_{i+1}$ with C_1 and $\mu_1'(C_1)$ does not allow $\sigma_{i,i+1}(t_1)$ to fire. Hence $t_1 \in C_1\bullet$ was satisfied when C_1 was added to $\mathcal{N}_i$. This implies $t_1 \in S_1\bullet$. Firing $\sigma_{i,i+1}(t_1)$ in $\mathcal{N}_{i+1}$ produces the same marking change for the places in P_i as firing t_1 in $\mathcal{N}_i$. Since $\sigma_{i,i+1}(t_1)$ is not allowed by $\mu_1'(C_1)$ to fire, firing t_1 from μ_1 empties S_1. Indeed, otherwise firing $\sigma_{i,i+1}(t_1)$ would not empty S_1 and so $\mu_1'(C_1)$ would allow it. Since t_1 is fired in the sequence $\sigma = \sigma_1 t_1 \sigma_2$, S_1 is an empty active siphon of $(\mathcal{N}_i, \mu_s)$.

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 replacement sequences 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 $(\mathcal{N}_i, \mu_s)$.

(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, in view of the AC-transformation or of the PT-transformation. (This is so because for all $i \geq 1$ $\mathcal{N}_i$ is PT-ordinary and with asymmetric choice, and hence only the new added control places can affect these properties). 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 and 4.2), 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. □

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 the procedure is able to find admissible constraints of the form $\sum_{p \in S} \alpha_p \mu(p) \geq 1$ with α_p positive integers. Assume also that $\mathcal{N}_1$ has a single T-minimal active subnet. The liveness enforcement procedure provides a supervisor not more restrictive than any supervisor subject to the same initial constraints (if any initial constraints are given) which also enforces T-liveness.*

Proof: Let $\mathcal{S}$ be the set of supervisors satisfying the initial constraints and enforcing T-liveness. Note that when we compare our procedure to another 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.

Note that $(\mathcal{N}_0, \mu_0)$ cannot be made T-live if $(\mathcal{N}_1, \mu_{0,1})$ cannot be made T-live, where $\mu_{0,1}(p) = \mu_0(p)$ $\forall p \in P_0$ and $\mu_{0,1}(p) = 0$ $\forall p \in P_1 \setminus P_0$. Indeed, assume the contrary. Then μ_0 enables an infinite transition sequence σ in which all transitions of T appear infinitely often. But this implies that $\sigma_{0,1}(\sigma)$ is also enabled by $\mu_{0,1}$, and therefore $\mathcal{N}_1$ is also T-live. Next we note that if $(\mathcal{N}_i, \mu_{0,i})$ cannot be made T-live if $(\mathcal{N}_{i+1}, \mu_{0,i+1})$ cannot be made T-live, where $\mu_{i+1,0}$ is the equivalent marking of $\mu_{i,0}$. Indeed let σ be an infinite firing sequence enabled by $\mu_{i,0}$ such that all transitions of T occur infinitely often in σ. By Lemma 6.2, if $\sigma_{i,i+1}(\sigma)$ is not enabled in $\mathcal{N}_{i+1}$, firing σ in $\mathcal{N}_i$ creates an empty active siphon, which contradicts the fact that all transitions of T appear infinitely often in σ, as an empty active siphon imply that some of the transitions of T are dead. (The contradiction results as follows. An empty active siphon implies a set of dead transitions from the active subnet, T_x. Let $T_{x1} = \{t \in T_1 : \exists t_u \in \sigma_{1,i}(t) \text{ and } t_u \in T_x\}$. Using the same construction as in the proof of Theorem 6.2, the projection of σ on T_1 (let it be σ_1) is enabled by $\mu_{1,0}$, where $\mu_{1,0}$ is the restriction of $\mu_{i,0}$ to the places of P_1. Therefore we apply Lemma 3.1 for $\mathcal{N}_1$ and σ_1, and using the notation of Lemma 3.1, we let $T_x^A = \|x\|$. Note that T_x^A defines an active subnet and $T \subseteq T_x^A$, as all transitions of T appear infinitely often in σ_1. Since we are in the case with no initial constraints, the theorem assumptions

imply no siphon control failures. Then, in view of the update algorithm of section 5.3, $T_{x1} \subset T_1^A$. However $T_{x1} \cap T_x^A = \emptyset$, so T_x^A is not a subset of T_1^A. This implies that T_x^A defines another T-minimal active subnet of $\mathcal{N}_1$, which contradicts the fact that there is a single T-minimal active subnet.)

The markings forbidden at every iteration i are those for which there are empty active siphons. Therefore only some markings for which $\mathcal{N}_i$ cannot be T-live are forbidden at the iteration i. Assume that $\mathcal{N}_0$ can be made T-live for a marking μ_0 which does not satisfy all constraints $L\mu \geq b$ and $L_0\mu \geq b_0$. Let i be the first iteration in which an inequality $l_1'\mu \geq b_1$ is added such that its restriction $l_1\mu \geq b_1$ to P_0 is one of the inequalities of $L\mu \geq b$ and $L_0\mu \geq b_0$ not satisfied by μ_0. Therefore $\mathcal{N}_i$ cannot be made live for $\mu_{0,i}$, the equivalent marking of μ_0 in $\mathcal{N}_i$. By the first part of the proof this implies that $(\mathcal{N}_0, \mu_0)$ cannot be made T-live, which is a contradiction. Therefore all liveness enforcing supervisors forbid the markings such that $L\mu \ngeq b$ or $L_0\mu \ngeq b_0$.

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.2.c of the procedure are infeasible. In the case when infeasibilities at some steps C.2.c occur, consider the first occurrence. In view of the proof of the paragraph above, such infeasibilities imply that T-liveness cannot be enforced for any initial marking satisfying the initial constraints, and so there are no supervisors in $\mathcal{S}$ ($\mathcal{S}$ is empty.) Therefore, we can conclude that the supervisor generated by the procedure is more permissive than any other supervisor in $\mathcal{S}$ whenever it enforces T-liveness; when it does not, $\mathcal{S}$ is empty. □

We note that in case of liveness enforcement, there is a single T-minimal active subnet, that is the whole net, and therefore we have the following consequence.

Corollary 6.4 *Assume that for all minimal active siphons S the procedure is able to find admissible constraints of the form $\sum_{p \in S} \alpha_p \mu(p) \geq 1$ with all α_p positive integers. When the procedure is used to enforce liveness, the supervisor it provides is not more restrictive than any supervisor subject to the same initial constraints (if any) which also enforces liveness.*

Theorem 6.3 gives sufficient conditions for the T-liveness supervisor to be least restrictive. The comparison assumes that the other supervisors are subject to the same initial constraints. In particular, the assumptions of the theorem are always true for Petri nets with controllable and observable transitions. Therefore, whenever the procedure is used to enforce T-liveness, no uncontrollable and unobservable transitions exist, the target Petri net has a single T-minimal active subnet and the procedure ends successfully, the supervisor generated is maximally permissive (least restrictive). The procedure may not end successfully when initial constraints are given and the initial constraints prevent enforcing T-liveness. Also, for some Petri nets the procedure may not end at all, and therefore we consider section 6.2.3.

6.2.2 Extending Permissivity

Theorem 6.3 and Corollary 6.4 show that for a large class of Petri nets the procedure is least restrictive. The natural question whether we can use our procedure to ensure least restrictiveness for an even larger class of Petri nets has a positive answer, as we show in this section. We consider the case when the target Petri net $\mathcal{N}_0$ has the T-minimal active subnets $\mathcal{N}_0^{A,1}$, $\mathcal{N}_0^{A,2}$, ... $\mathcal{N}_0^{A,p}$. Theorem 6.3 does not apply, as we have p ($p > 1$) T-minimal subnets. However, it applies for $T_0^{A,i}$-liveness, as there is a single $T_0^{A,i}$-minimal active subnet: $\mathcal{N}_0^{A,i}$ (we denote by $T_0^{A,i}$ the set of transitions of $\mathcal{N}_0^{A,i}$ and $i = 1 \ldots p$). Assume that the procedure terminates for all $i = 1 \ldots p$ when used to enforce $T_0^{A,i}$-liveness. Let $L^{(i)}\mu \geq b^{(i)}$ and $L_0^{(i)}\mu \geq b_0^{(i)}$ be the

generated constraints. Assume that we have ordered the T-minimal active subnets such that for $1 \leq i \leq u$ the procedure had no siphon control failures when used for $T_0^{A,i}$-liveness, but for each $u+1 \leq i \leq p$ it had some siphon control failures ($0 \leq u \leq p$). Let Ξ be the supervisor defined as follows. Ξ requires the initial marking μ_0 to be in the set $\mathcal{M}$, where

$$\mathcal{M} = \bigcup_{i=1}^{u} \left\{ \mu : L^{(i)}\mu \geq b^{(i)} \bigwedge L_0^{(i)}\mu \geq b_0^{(i)} \right\}$$

Also Ξ allows a transition to fire only if the next reached marking is in $\mathcal{M}$.

Theorem 6.4 *Assume that for each $i = 1 \ldots u$, for all minimal active siphons S the procedure is able to find admissible constraints of the form $\sum_{p \in S} \alpha_p \mu(p) \geq 1$ with all α_p positive integers. Assume also that for each $i = u+1 \ldots p$ the first siphon control failure occurs at the step C.2.c and that in all iterations previous to the failure, for all minimal active siphons S the procedure is able to find admissible constraints of the form $\sum_{p \in S} \alpha_p \mu(p) \geq 1$ with all α_p positive integers. Then Ξ is the least restrictive T-liveness enforcing supervisor.*

Proof: Failures at the step C.2.c are only possible when initial constraints are given. The proof of Theorem 6.3 applies, and so for the given initial constraints $T_0^{A,i}$-liveness cannot be enforced for all $i = u+1 \ldots p$. Let $\mu_0 \notin \mathcal{M}$, and assume that μ_0 enables a firing sequence σ which includes all transitions in T infinitely often. In the notations of Lemma 3.1, let $T^A = \|x\|$. Then T^A defines an active subnet, and note that $T \subseteq T^A$. Since $\mathcal{N}_0^{A,i}$, $i = 1 \ldots p$, are all the T-minimal active subnets, there is j, $1 \leq j \leq p$, such that $T_0^{A,j} \subseteq T^A$. If $j \leq u$, we have contradiction, since by Theorem 6.3 not all transitions of $T_0^{A,j}$ can be made live for $\mu_0 \notin \mathcal{M}$, and so not all of them can appear in σ. If $j > u$ we again have contradiction, since for all initial markings satisfying the initial constraints not all transitions of $T^{A,j}$ can be made live. □

6.2.3 Termination Results

The procedure, as defined, may not terminate for any Petri net structure. By analyzing cases in which the procedure does not terminate, we considered two changes of the procedure which help termination. 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 been guaranteed. However note that the two changes we propose may help termination by themselves, that is without using initial constraints (L_0, b_0), and not only for structurally bounded Petri nets. A Petri net $\mathcal{N}$ is **structurally bounded** [14] if for all initial markings μ_0, $\mathcal{R}(\mathcal{N}, \mu_0)$ is bounded.

6.2.3.1 Modification A In every iteration, all constraints are stored only in the form restricted to the places of the target net $\mathcal{N}_0$. That is, when a new constraint $l^T\mu \geq c$ is added to (L, b) or (L_0, b_0), it is first restricted to the places of $\mathcal{N}_0$, and then added. Also, the transformation of (L_0, b_0) from the step A of the procedure is not done. Thus the constraints ignore the marking of the places resulted by transition split. A siphon is considered to be implicitly controlled if the marking inequality required for the siphon to be controlled is implied by the current set of constraints (which are 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, modification A does not change the procedure for those Petri nets in which the final Petri net $\mathcal{N}_k$ has no split transitions.

Theorem 6.5 *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. The liveness enforcement procedure with the modification A terminates if started with initial constraints(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. □

Theorem 6.5 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 modification A can be summarized as follows:

- Find a set of constrains $L_i\mu \geq b_i$ 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 modification A and initial constraints (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 modification A is that Theorem 6.2, which guarantees liveness enforcement, 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 modification A is used. (This difference does not matter in the proof of Theorem 6.2.)

6.2.3.2 Modification 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. Unlike to the deadlock prevention approach, in our liveness enforcing approach this assumption is often not satisfied, as the AC-transformation makes the procedure a lot more prone to perform transition splits.

Theorem 6.6 *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 liveness enforcement 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. □

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

- Find a set of constrains $L_i\mu \geq b_i$ 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 modification B and initial constraints (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 weak, as it may be hard to know whether the assumption on transition splits holds true, and there are many cases in which the assumption is not true.

6.3 Final Remarks and Directions for Further Research

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. The additional constraints are to be enforced first, and then the liveness enforcement procedure can be applied. Thus, enforcing (L_a, b_a) in $\mathcal{N}$ using supervision based on place invariants ([13, 22], also in section 4.3) produces the closed loop Petri net $\mathcal{N}_L$. Then the liveness enforcement procedure can be used with $\mathcal{N}_0 = \mathcal{N}_L$ and initial constraints (L_I, b_I) reflecting the invariants resulted by enforcing (L_a, b_a) to $\mathcal{N}$.

The reason why we should not first apply the liveness enforcement procedure to $\mathcal{N}$ and then enforce (L_a, b_a) is that additional constraints can effect loss of liveness. Indeed, we can easily find examples of live Petri nets which lose liveness by adding some marking constraints.

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 to each place a *capacity*. This type of Petri net is called place/transition net [16]. So, a **place/transition structure** is represented by the quintuple $\mathcal{N} = (P, T, F, W, K)$, where $K : P \to \overline{\mathbb{N}}$ is the **capacity function**. With an additional initial marking we have a **place/transition net**, denoted by $(\mathcal{N}, \mu_0)$. 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 [13, 22], outlined also in section 4.3.

If all the places have finite capacity, the equivalent Petri net is by construction structurally bounded. The liveness enforcement procedure can be started as in section 6.3.1, with $\mathcal{N}_0 = \mathcal{N}_E$ and constraints (L_a, b_a) which describe $\mu(p) \leq K(p)$ for all $p \in P$. The method can be guaranteed to terminate as shown in section 6.2.3, 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 The Termination Problem

Section 6.2.3 shows how to modify the procedure to guarantee termination for structurally bounded Petri nets. However, as shown in section 6.2.3, guaranteed termination may come to a cost. This is why in this section we consider the procedure as defined in section 5.4 and examine two divergence causes.

6.3.3.1 Converging Constraints 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(a) unless implicitly controlled siphons are not eliminated. However this operation does not guarantee termination in general. For instance this does not help the procedure to converge for the Petri net of figure 11(b). The reason is that new places generated through transition splits appear in the case (b); such places prevent the generated siphons to be implicitly controlled, and so the procedure to converge.

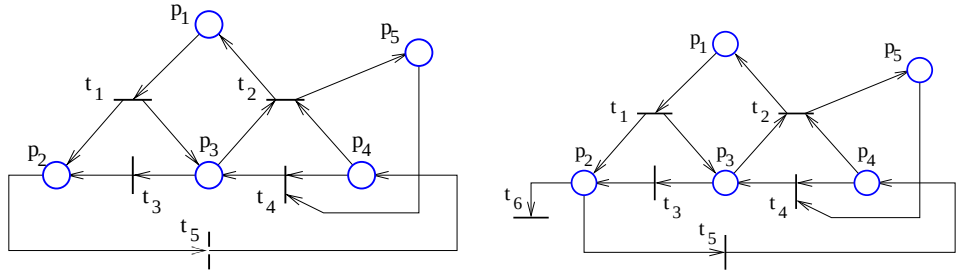

Figure 11: Example for the termination problem

Checking whether a siphon is implicitly controlled is equivalent to an integer programming feasibility problem. Solving integer programs is an *NP* type problem [21].

6.3.3.2 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 $\forall n \geq 2$, $\forall x_1, x_2, \ldots x_n \in F$ and $\forall \alpha_1, \alpha_2, \ldots \alpha_n \in \mathbb{R}_+$ 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 enforcible produce a marking for which liveness is not enforcible, 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 enforcible 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, 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.

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.

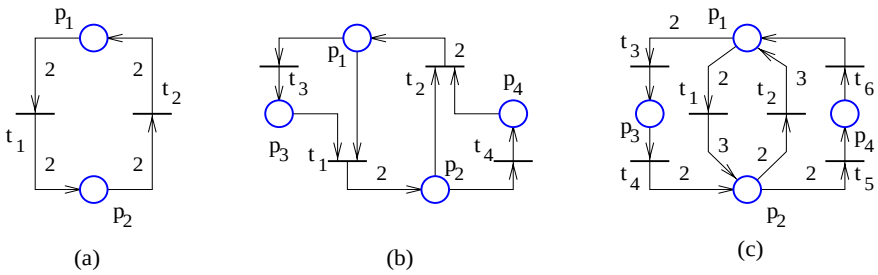

Figure 12: Examples for section 6.3.3.2

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 liveness enforcement 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 enforcible and $\mu_2 = 0.5\mu_0 + 0.5\mu_1$ is a deadlock marking, the procedure cannot terminate.

7 Summary of Results

In this section we outline the main new results of this paper. The new results of our prior work [8, 9] are not included here. In section 3:

- Theorem 3.2 is fundamental for our procedure for T-liveness enforcement. It shows the relation between siphons and dead transitions.
- Proposition 3.6 gives a necessary condition and a sufficient condition for T liveness to be enforced in a PT-ordinary asymmetric choice Petri net.

The liveness enforcing procedure has been stated in section 5.4. Variations of this procedure have been given in section 6.2.3. The procedure has the following characteristics:

- Given a Petri net structure and a set of transitions T, 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 T-live.
- 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 $L_0\mu \ngeq b_0$, (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 liveness enforcement 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, T-liveness is enforced.
- The case when the structure of $\mathcal{N}_0$ does not allow T-liveness is detected in Proposition 6.19.
- In the conditions of Theorem 6.3, the supervisor is maximally permissive (least restrictive). A situation in which the conditions of Theorem 6.3 are satisfied is when the procedure is given a Petri net with controllable and observable transitions in the case of liveness enforcement. A method to extend the permissivity is given in section 6.2.2, and the theoretical result is given in Theorem 6.4.
- Two modifications of the procedure have been proposed to guarantee termination. Theorem 6.5 and Theorem 6.6 guarantee termination for the two modifications.

References

[1] K. Barkaoui and I. Abdallah. Deadlock avoidance in fmss based on structural theory of Petri nets. In *IEEE Symposium on Emerging Technologies and Factory Automation*, 1995.

[2] K. Barkaoui and J. F. Pradat-Peyre. On liveness and controlled siphons in Petri nets. In *Lecture Notes in Computer Science: 17th International Conference in Application and Theory of Petri Nets (ICATPN'96), Osaka, Japan*, volume 1091, pages 57–72. Springer-Verlag, June 1996.

[3] E. Boer and T. Murata. Generating basis siphons and traps of Petri nets using the sign incidence matrix. *IEEE Trans. on Circuits and Systems*, 41(4), 1994.

[4] R. David and A. Hassane. Petri nets for modeling of dynamic systems - a survey. *Automatica*, 32(2):175–202, 1994.

[5] J. Ezpeleta, J. M. Colom, and J. Martínez. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. on Robotics and Automation*, 11(2):173–184, 1995.

[6] J. Ezpeleta, J. Couvreur, and M. Silva. A new technique for finding a generating family of siphons, traps and st-components. In *Advances in Petri Nets, Lecture Notes in Computer Science*. Springer-Verlag, 1993.

[7] K.X. He and M.D. Lemmon. On the existence of liveness-enforcing supervisory policies of discrete-event systems modeled by n-safe Petri nets. In *Proceedings of the 2000 IFAC Conference on Control Systems Design, Slovakia*, June 2000.

[8] M. V. Iordache, J. O. Moody, and P. J. Antsaklis. 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, July 1999.

[9] M. V. Iordache, J. O. Moody, and P. J. Antsaklis. Automated synthesis of deadlock prevention supervisors using Petri nets. Technical report of the isis group, isis-2000-003, University of Notre Dame, May 2000.

[10] M. V. Iordache, J. O. Moody, and P. J. Antsaklis. A method for the synthesis of deadlock prevention controllers in systems modeled by Petri nets. In *Proceedings of the 2000 American Control Conference*, pages 3167–3171, June 2000.

[11] K. Lautenbach. Linear algebraic calculation of deadlock and traps. *Concurrency and Nets – Advances in Petri Nets*, pages 315–336, 1987.

[12] K. Lautenbach and H. Ridder. The linear algebra of deadlock avoidance – a Petri net approach. Technical report, University of Koblenz, Institute for Computer Science, 1996.

[13] J. O. Moody and P. J. Antsaklis. *Supervisory Control of Discrete Event Systems Using Petri Nets*. Kluwer Academic Publishers, 1998.

[14] T. Murata. Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*, pages 541–580, April 1989.

[15] J. Park and S. Reveliotis. Structural control of sequential resource allocation systems with multiple resource acquisitions and flexible routings. Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology, 2000.

[16] W. Reisig. *Petri Nets*, volume 4 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.

[17] S. R. Sreenivas. On a free-choice equivalent of a Petri net. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 4092–4097, San Diego, California, December 1997.

[18] S. R. Sreenivas. On the existence of supervisory policies that enforce liveness in discrete event dynamic systems modeled by controlled Petri nets. *IEEE Transactions on Automatic Control*, 42(7):928–945, July 1997.

[19] S. R. Sreenivas. An application of independent, increasing, free-choice Petri nets to the synthesis of policies that enforce liveness in arbitrary Petri nets. *Automatica*, 44(12):1613–1615, December 1998.

[20] S. R. Sreenivas. On supervisory policies that enforce liveness in a class of completely controlled Petri nets obtained via refinement. *IEEE Transactions on Automatic Control*, 44(1):173–177, January 1999.

[21] L. Wolsey. *Integer Programming*. John Wiley & Sons, New York, 1991.

[22] E. Yamalidou, J. O. Moody, P. J. Antsaklis, and M. D. Lemmon. Feedback control of Petri nets based on place invariants. *Automatica*, 32(1):15–28, January 1996.