

Synthesis of T -Liveness Enforcing Supervisors in Petri Nets

Marian V. Iordache and Panos J. Antsaklis*

Abstract

We consider the problem of enforcing via supervision that the transitions in some set T of a Petri net are live. We call this T -liveness enforcement. The procedure presented in this paper synthesizes T -liveness enforcing supervisors for arbitrary Petri nets, which may have uncontrollable and unobservable transitions. It is shown that for a large class of Petri nets the synthesized supervisor is least restrictive. The procedure is based on the structural properties of the Petri net, and so the supervisor synthesis is independent of the initial marking.

1 Introduction

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 total) is impossible. Rather than providing a method for liveness verification, we provide a method which synthesizes a supervisor such that the supervised system is live. When not all events of a system are desirable, enforcing liveness with respect to the set of desirable events is preferable to enforcing that the whole system is live. In terms of Petri nets, an event corresponds to firing a transition. When each transition corresponds to a distinct event, enforcing liveness with respect to a desirable set of events corresponds to enforcing liveness with respect to a set of transitions T , i.e. enforcing T -liveness.

Given an arbitrary Petri net, the procedure we present in this paper synthesizes a T -liveness enforcing supervisor. The Petri nets we consider are allowed to be unbounded,

*Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 (e-mail: iordache.1, antsaklis.1@nd.edu). The authors gratefully acknowledge the support of the National Science Foundation (ECS-9912458) and of the Army Research Office (DAAG55-98-1-0199).

generalized (i.e. with integer arc weights) and with uncontrollable and unobservable transitions. In the case of Petri nets with all transitions controllable and observable, the supervisor is least restrictive when the procedure is used to enforce liveness and typically least restrictive when the procedure is used to enforce T -liveness. The supervisor is defined as a conjunction of linear marking inequalities. Thus the procedure is not dependent on the initial marking. Instead, the set of initial markings for which T -liveness is enforced is characterized as the feasible set of a system of linear marking inequalities. Thus the T -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 least restrictive, enforcing T -liveness is impossible for all initial markings for which the supervisor is not defined. Therefore our procedure can also be used for verification.

We note that our procedure may not always terminate and even when it terminates, the computations may be complex. However these computations are performed offline. Once a supervisor has been synthesized, running it in real-time involves few computations. We provide a procedure extension for guaranteed termination, but the trade-off is that the supervisor may be more restrictive and that the extension is only useful for bounded Petri nets.

Compared to the existing methodologies for liveness enforcement, the supervisory problem solved by our procedure cannot be solved with finite automata based approaches. Indeed, since we consider Petri net structures rather than a Petri net with an initial marking, an automaton which would include the behavior of the Petri net for any initial marking would have an infinite number of states. Of course, this is not the case for the approaches which consider a single initial marking and a bounded Petri net. Applications which may benefit from considering the initial marking to be unknown are in the area of Flexible Manufacturing, as the initial marking corresponds to the number of available resources. Symbolic Model Checking (SMC) could be used for T -liveness enforcement in Petri nets. However we note that SMC approaches are not either guaranteed to terminate. Also, the output of an SMC approach would be a set of markings rather than a compact representation as a conjunction of linear marking inequalities, thus increasing the complexity of the supervisor.

To our knowledge, there are no results in the literature on enforcing the transitions in a given set to be live. Thus the T -liveness problem we consider is new. Liveness is a special case of T -liveness, as it means that all transitions in a Petri net are live. There are not many results on liveness enforcement in Petri nets, although numerous results exist on other liveness topics. Previous constructive results consider restricted classes of Petri nets. A necessary and sufficient condition for the existence of liveness supervisors appears in [13]. A method for liveness enforcement in a class of conservative ordinary Petri nets

has been given in [3]; the approach is not least restrictive. The approach of [3] has been recently extended to generalized Petri nets in [11]. Polynomial complexity has been proved, however the considered Petri nets are conservative and the approach is not least restrictive. A liveness enforcing approach for a restricted class of ordinary Petri nets is given in [14]. Another liveness enforcing approach appears in [15]; it is based on the coverability graph, and hence the initial marking is required. In [4] 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 [4] 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 [7], and its improvement in [5]. 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 paper is iterative, at every iteration correcting new deadlock situations. Using iterations to correct deadlock situations has also been used in [8]. In our procedure we employ supervisory control based on place invariants [9, 16], 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 [8]. A transformation to free choice nets, which is a particular class of asymmetric choice nets, has been used in [12]. In [12] 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], which relates liveness in asymmetric choice Petri nets to siphons.

We begin in section 2 by introducing notations, definitions and results important for our procedure. The theoretical background of our procedure is given in section 3. To the authors' knowledge, the material presented beginning with section 3 is new. The T -liveness enforcing procedure is described in section 4 and formally stated in section 4.5; sections 4.1 to 4.5 describe the operations involved in the procedure. Section 5 includes illustrative examples. The procedure is analytically proved in section 6. Thus Theorem 6.1 proves that the synthesized supervisor enforces T -liveness and Theorem 6.2 shows that the supervisor is least restrictive for a large class of Petri nets. We conclude with two procedure extensions in sections 7 and 8. Section 7 shows how to obtain the least restrictive supervisor in one of the cases when the synthesized supervisor is not least restrictive. Section 8 gives a procedure extension for guaranteed termination.

2 Preliminaries

We denote a Petri net by $\mathcal{N} = (P, T, F, W)$, where P is the set of places, T the set of transitions, F the set of transition arcs and W the transition arc weight function. We use the symbol μ to denote a marking and we write (\mathcal{N}, μ_0) when we consider the Petri net \mathcal{N} with the initial marking μ_0 . The incidence matrix of a Petri net is denoted by D , where the rows correspond to places and the columns to transitions. Also, by denoting a place by p_i or a transition by t_j , we assume that p_i corresponds to the i 'th row of D and t_j to the j 'th column of D . We use the notation $\mu[\sigma > \mu'$ to express that the marking μ enables the firing sequence σ and μ' is reached by firing σ .

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 2.1 *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 $W(p, t) = 1$.*

A **siphon** is a set of places $S \subseteq P$, $S \neq \emptyset$, such that $\bullet S \subseteq S \bullet$. A siphon S is **minimal** if there is no siphon $S' \subset S$. A siphon S is **controlled** if for all reachable markings it contains at least one token. Also, S is an **empty siphon** if the current total marking of S is zero. Given a Petri net (\mathcal{N}, μ_0) , a transitions t is **live** if any reachable marking enables some firing sequence which includes t ; the Petri net is live if all transitions are live and deadlock-free if for all reachable markings there is a transition which can be fired.

Definition 2.2 *Let (\mathcal{N}, μ_0) be a Petri net and T a subset of the set of transitions. We say that the Petri net is **T-live** if all transitions $t \in T$ are live.*

Note that T -liveness corresponds to liveness when T equals the total set of transitions.

Definition 2.3 *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 **supervisor** 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.*

We denote by $\mathcal{R}(\mathcal{N}, \mu_0, \Xi)$ the set of reachable markings when (\mathcal{N}, μ_0) is supervised with Ξ . We say that **deadlock can be prevented** in \mathcal{N} if an initial marking μ_0 and a supervisor Ξ exist such that (\mathcal{N}, μ_0) supervised by Ξ is deadlock-free. Similarly, we say that **liveness can be enforced** in \mathcal{N} if an initial marking μ_0 and a supervisor Ξ exist such that (\mathcal{N}, μ_0) supervised by Ξ is live.

A Petri net is said to be **(partially) repetitive** [10] if a marking μ_0 and a firing sequence σ enabled by μ_0 exist such that every (some) transition occurs infinitely often in σ . It is known [10] that a Petri net is (partially) repetitive iff a vector x of positive (nonnegative) integers exists, such that $Dx \geq 0$ and $x \neq 0$, where D is the incidence matrix. Consequently we can use linear programming techniques to check whether a Petri net is (partially) repetitive. Note that liveness can be enforced iff the Petri net structure is repetitive. Also, if T -liveness can be enforced, then the Petri net is partially repetitive. A necessary and sufficient condition for T -liveness to be enforcible results from Lemma 3.1, in section 3.

The supervisory technique used by our procedure for liveness enforcement is supervision based on place invariants [9, 16]. In this approach the supervisor is defined by a set of linear marking inequalities $L\mu \geq b$. The supervision can be accomplished by extending the Petri net with additional places, called **control places**. The construction is summarized in the following theorem.

Theorem 2.1 [9, 16] *Let a plant Petri net with all transitions controllable and observable, incidence matrix D and initial marking μ_0 be given. A set of n_c linear constraints $L\mu \leq b$ are to be imposed. If $b - L\mu_0 \geq 0$ then a Petri net supervisor with incidence matrix $D_c = -LD$ and initial marking $\mu_{c0} = b - L\mu_0$ enforces the constraint $L\mu \leq b$ when included in the closed loop system $D_S = [D^T, D_c^T]^T$. Furthermore, the supervision is least restrictive.*

Theorem 2.1 can still be used for Petri nets with uncontrollable and unobservable transitions if $L\mu \leq b$ is *admissible*. Uncontrollable and unobservable transitions correspond to uncontrollable and unobservable events of the modeled 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. Any constraints $L\mu \leq b$ satisfying this requirement are called *admissible constraints*. Constraint admissibility may depend on the initial marking of the Petri net. However 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 iff the following equations of [9] are true:

$$LD_{uc} \leq 0 \tag{1}$$

$$LD_{uo} = 0 \tag{2}$$

where D_{uc} and D_{uo} denote the columns of the incidence matrix which correspond to uncontrollable and unobservable transitions, respectively. In this paper we consider $L\mu \leq b$ **admissible** if L satisfies (1) and (2).

3 Theoretical Background

In this section we briefly introduce a number of definitions and results necessary for the description and the proof of our T -liveness enforcement method. Please refer to the appendix for the proofs.

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 $Dx \geq 0$, $\forall t_i \in U: x(i) \neq 0$ and $\forall t_i \in T \setminus U: x(i) = 0$.*

Theorem 3.1 *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 supervisor Ξ such that $\forall \mu \in \mathcal{R}(\mathcal{N}, \mu_0, \Xi)$, no transition in $T \setminus T_D$ is dead.*

Next we denote by *active subnets* parts of a Petri net which can be made live by supervision for appropriate initial markings. Then we define a subclass of siphons.

Definition 3.1 *Let $\mathcal{N} = (P, T, F, W)$ be a Petri net and D the incidence matrix. $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$ 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 satisfying $Dx \geq 0$. 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 $T \subseteq T_x^A$.*

Definition 3.2 *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.)*

Even though we consider T -liveness enforcement in arbitrary Petri nets, the following theorem is fundamental for our approach. In our approach we iteratively generate intermediary Petri nets which are PT-ordinary and with asymmetric choice. The proof that the supervisor generated by our procedure enforces T -liveness follows from the fact that the last intermediary Petri net is T -live, which in turn follows from the result below.

Theorem 3.2 *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. 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).*

4 The Liveness Enforcing Procedure

4.1 Introduction to the Procedure for Liveness Enforcement

Given a target Petri net \mathcal{N}_0 , the liveness enforcing procedure generates a sequence of asymmetric choice PT-ordinary Petri nets, $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k$, 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, the Petri net is transformed to be with asymmetric choice and PT-ordinary. Thus the iteration i produces the asymmetric choice PT-ordinary net \mathcal{N}_{i+1} . The active siphons (Definition 3.2) of each \mathcal{N}_i are taken with respect to an active subnet \mathcal{N}_i^A computed for every iteration i ; if T is the set of transitions of \mathcal{N}_0 to be enforced live, \mathcal{N}_i^A is a T -minimal active subnet of \mathcal{N}_i (Definition 3.1). Controlling a siphon involves enforcing a linear marking inequality. Let $L_i\mu \geq b_i$ be the total set of inequalities 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 Theorem 3.2, \mathcal{N}_k is T -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 , which define the supervisor for liveness enforcement in \mathcal{N}_0 .

The liveness enforcement procedure is defined in section 4.5. The sections preceding section 4.5 define in detail operations performed by the procedure. Section 4.2 shows how the Petri nets are transformed to be PT-ordinary and with asymmetric choice. The precise way in which the constraints are generated is considered in section 4.3. Then section 4.4 presents algorithms for the computation of the active subnets.

4.2 Transforming Petri Nets to PT-ordinary asymmetric choice Petri nets

We are interested in using PT-ordinary asymmetric choice Petri nets because our T -liveness test requires such Petri nets. However, as we will show in the next sections, by using the transformations of this section we can synthesize T -liveness supervisors for Petri nets not necessarily PT-ordinary or with asymmetric choice.

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

We use a modified form of the similar transformation from [8], 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:

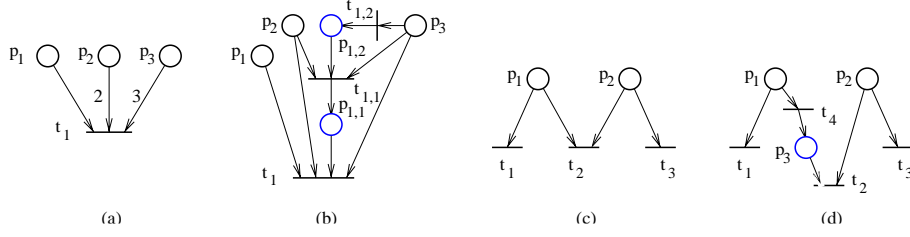


Figure 1: Illustration of the transition split: (a) initial configuration; (b) the effect of the PT-transformation; (c) initial configuration; (d) the effect of the AC-transformation.

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** consists 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. Note that:

$$|p_{j,i} \bullet| = |\bullet p_{j,i}| = 1 \quad i = 1 \dots m - 1 \quad (3)$$

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

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 .

4.2.2 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 1(c-d). Note that the operations correspond to a modified form of transition split operations (section 4.2.1). We call the transformation to asymmetric choice Petri nets **AC-transformation**.

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 $U = \{(p_i, p_j) \in P \times P : p_i \in \bullet t, p_j \in \bullet t, p_i \bullet \not\subseteq p_j \bullet \text{ and } p_j \bullet \not\subseteq p_i \bullet\}$.
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 $W'(p', t) = W(p, t)$.

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

4.3 Generating Marking Constraints

Each marking constraint generated by the procedure corresponds to the requirement that a minimal active siphon is not empty. Thus, if S is such a siphon, the requirement is that

$$\sum_{p \in S} \mu(p) \geq 1 \quad (5)$$

where μ is the marking. The siphon S can be invariant controlled in order to always satisfy (5). The invariant is created by adding an additional place, called **control place**, which we denote by C . See Theorem 2.1 or [3, 1, 2]. Thus the equation of the marking of C is

$$\mu(C) = \sum_{p \in S} \mu(p) - 1 \quad (6)$$

The constraint (5) may not be admissible when the Petri net has unobservable and uncontrollable transitions. If this is the case, the constraint is replaced with a stronger constraint which is admissible and has the form

$$\sum_{p \in S} \alpha_p \mu(p) \geq 1 \quad (7)$$

where α_p are nonnegative integers. Note that (5) is a special case of (7), and so, for the remainder of the paper, we consider the constraints enforced by control places to have the form (7). See section 4.3.2 for the transformation of (5) to an admissible form (7). The liveness procedure considers a *siphon control failure* when no such admissible constraint exists. The control place enforcing (7) is added with the methodology of Theorem 2.1:

$$\mu(C) = \sum_{p \in S} \alpha_p \mu(p) - 1 \quad (8)$$

In an iteration the liveness procedure controls in this way all minimal active siphons. The Petri net in which the control places are added is PT-ordinary and with asymmetric choice, but the Petri net resulting after the control places have been added may no longer be so. By applying the PT and AC transformations to make again the Petri net PT-ordinary and with asymmetric choice, the relation (8) is modified. In general, it can be proved that if $l^T \mu \geq b$ is a marking constraint enforced in a Petri net \mathcal{N} for initial markings in a set \mathcal{M}_I , and the Petri net is PT and AC transformed to \mathcal{N}_t , then the form of $l^T \mu \geq b$ in \mathcal{N}_t is $l_t^T \mu_t \geq b_t$, where this form is obtained from $l^T \mu \geq b$ with the substitution

$$\mu(p) \longrightarrow \mu_t(p) + \sum_{z=1}^r \mu_t(p_z) + \sum_{i=1}^k \sum_{j=1}^{m_i-1} j \mu_t(p_{i,m_i-j}) \quad (9)$$

for each place p of \mathcal{N} , where the notations are as follows. k and m_i are determined in \mathcal{N} : $k = |p \bullet|$, $m_i = W(p, t_i) \forall t_i \in p \bullet$. The places $p_{i,j}$ are the places resulted by splitting the transitions $t_i \in p \bullet$, where the notation of section 4.2.1 is used. The places p_z are the places resulting from the AC-transformation which satisfy $\bullet \bullet p_z = p$. We use (9) in order to derive the new form of (8) after the PT and AC transformations. Thus assume that C is one of the control places added by the liveness enforcing procedure in the iteration i to \mathcal{N}_i . By applying the PT-transformation and then the AC-transformation with the argument M equal to the set of the control places added to \mathcal{N}_i , (8) is transformed to

$$\mu(C) + \sum_{z=1}^r \mu(p_z) + \sum_{i=1}^k \sum_{j=1}^{m_i-1} j \mu(p_{i,m_i-j}) = \sum_{p \in S} \alpha_p \mu(p) - 1 \quad (10)$$

where the notations are similar to (9): $k = |C \bullet|$, $m_i = W(C, t_i) \forall t_i \in C \bullet$, $p_{i,j}$ are the places resulted by splitting the transitions $t_i \in C \bullet$, and p_z are the places resulting from the AC-transformation such that $\bullet \bullet p_z = C$. Note that the siphon S remains controlled, that is (7) (and so (5)) is still true. The procedure insures that (10) is not further modified by the operations performed in subsequent iterations. This is accomplished by selecting in each iteration the parameter M of the AC-transformation to equal the set of the control places added in that iteration.

4.3.1 The sets of inequalities (L, b) and (L_0, b_0)

The siphons in a iteration i may contain control places added in previous iterations. Thus (7) may involve not only places of the target net \mathcal{N}_0 , but also control places. However, the marking of the control places appearing in (7) can be eliminated by using (10). By eliminating all control place markings, (7) can be written as:

$$l^T \mu \geq c \quad (11)$$

where l is a column vector of integers, c a positive integer, and $l(i) = 0$ for all places p_i which are control places. The set of inequalities $L\mu \geq b$ contains the inequalities (11) corresponding to each siphon controlled by the liveness procedure. When the procedure terminates, the supervisor of the target net is defined by $L_R\mu \geq b$, where L_R is L restricted to the columns which correspond to the places of the target net.

The test we use to check whether a siphon S does not need a control place is as follows. First, a control place C is added to enforce (5). If $C \bullet \subseteq \bullet S$, C is not needed and so it is deleted. In this case (5) is true for all markings if true for the initial marking. Such initial marking constraints are not stored in $L\mu \geq b$, but in a separate set of constraints,

$L_0\mu \geq b_0$. As in the case of $L\mu \geq b$, the constraints (5) stored in $L_0\mu \geq b_0$ are in the form (11). When the procedure terminates, the initial marking μ_0 of the target net is required to satisfy $L_{0,R}\mu_0 \geq b_0$, where $L_{0,R}$ is L_0 restricted to the columns which correspond to the places of the target net.

4.3.2 Implicitly controlled siphons

Let S be a siphon considered for control. We say that S is **(implicitly) controlled** if (5) is satisfied for all markings μ which satisfy the current $L\mu \geq b$ and $L_0\mu \geq b_0$. For a controlled siphon a control place is not necessary and no new constraint needs to be added in $L_0\mu \geq b_0$.

4.3.3 Transforming Constraints to Admissible Constraints

This section describes an algorithm to generate the admissible constraints (7) and the additional requirements that the coefficients α_p of (7) are to satisfy. Assume that S in (7) is an active siphon which appears in the iteration i . 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 (7) is admissible in \mathcal{N}_0 .

Let a be the vector whose elements are zero for the places $p \notin S$ and α_p for the places $p \in S$; then (7) can be written as $a^T\mu \geq 1$. We require that at least two coefficients α_p are nonzero, and at least one coefficient α_p such that p is in the active subnet is nonzero. Let D_s be the restriction of the current incidence matrix D to the columns of the transitions resulted by split operations in all previous iterations. An additional constraint is

$$a^T D_s \leq 0 \tag{12}$$

The last requirement is necessary for the proof of Theorem 6.1. It ensures that the control place C which results by enforcing (7) satisfies $C \notin t\bullet$ for all transitions t generated by transition split operations. (This requirement can be proved to be satisfied when (5) is directly enforced.) As shown in section 4.3.1, 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 (7) can be written as

$$a_z^T [U^T, I]^T \mu_p \geq 1 + a_c^T g \tag{13}$$

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:

$$\begin{aligned} a^T N_r D_{uc} &\geq 0 \\ a^T N_r D_{uo} &= 0 \end{aligned} \tag{14}$$

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 (14). Then we can choose a_n such that:

$$a^T D_{sn} = 0 \tag{15}$$

Thus the control place C results with less connections, and this may help reduce the number of siphons in the next iteration. The following algorithm finds the coefficients α_p . The algorithm does not fail if a solution of the form (7) exists.

Input: $\mathcal{N}_i = (P_0, T_0, F_0, W_0)$, 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 (7).

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** (14) is satisfied **then** exit and declare (5) admissible constraint.
4. Let R be the set of transitions which correspond to the constraints of (14) not satisfied by α .
5. **If** initial constraints¹ have been given **then**²
 - (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** $R = R \setminus \{t\}$

¹initial constraints are an optional input of the liveness procedure; see section 4.5

²without initial constraints the step below will not reduce R

- (b) **If** $R = \emptyset$ **then** exit and declare (5) admissible.
6. Keep in (14) only the constraints which correspond to transitions in R . Then write (12), (14) and (15) 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) Check³ 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.⁴
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 .

4.4 The Computation of a T -minimal Active Subnet

The following algorithm computes a T -minimal active subnet or, if none exists, a T_x -minimal active subnet such that $T_x \subset T$. A T -minimal active subnet does not exist iff some of the transitions of T cannot be made live (see Definition 3.1 and Lemma 3.1).

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 $T \subseteq T_0$; optionally a set X of transitions which must not appear in the T -minimal active subnet (by default $X = \emptyset$.)

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 $x(i) = 0 \forall t_i \in X$.

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

else $T^A = \text{maxactn}(T_0, D, T, X)$ (no T -minimal solution exists, and so an approximation is constructed)

³The feasibility check involves solving a linear program

⁴ $|A|$ denotes the number of elements of A

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)

Let⁵ $M = \|x_0\|$ 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, X)

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 $x(i) = 0 \forall t_i \in X$.

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

Else $M = \emptyset$.

$N = \text{minactn}(T_0, x_s, D, T \cap \|x_s\|)$

Return N

Using a nonempty set X adds to the feasibility problems of the algorithm above the additional constraints that $x(j) = 0 \forall j \in X$. The set X 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 procedure, the active subnet needs to be reevaluated at every iteration. If X has not been changed in the iteration $i - 1$, the new active subnet \mathcal{N}_i^A can be updated by simply repeating the changes done to \mathcal{N}_{i-1} in \mathcal{N}_{i-1}^A . We give below the *update 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: $\mathcal{N}_i^A = (P_i^A, T_i^A, F_i^A, W_i^A)$.

⁵ $\|x\|$ denotes $\{t_i : x(i) \neq 0\}$

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 .

4.5 The Liveness Enforcing Procedure

Input: The target Petri net \mathcal{N}_0 , a nonempty set of transitions T and, optionally, a set of initial constraints $L_I \mu \geq b_I$. (See remark 3 of section 4.6 for the usage of initial constraints.)

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, μ_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 PT-transformed and then AC-transformed (section 4.2). The transformed net is \mathcal{N}_1 . The initial constraints (L_0, b_0) , if any, are transformed according to (9). Let $i = 1$. If not previously defined, let $X = \emptyset$.

B. A T -minimal active subnet for each of \mathcal{N}_0 and \mathcal{N}_1 is computed such that the transitions in X are not included (section 4.4). If none exists, the algorithm of section 4.4 computes a T_x -minimal active subnet such that $T_x \subset T$ and the transitions in X are not included. If no such $T_x \neq \emptyset$ exists, the procedure terminates and declares failure.

C. While true 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 is *uncontrolled* if not implicitly controlled (section 4.3.2).)
2. For every new uncontrolled minimal active siphon S :

Let C be the control place which would result by enforcing (5).

(a) If $C \bullet \subseteq \bullet S$, then S does not need control, C is not added to \mathcal{N}_i and (5) is added to (L_0, b_0) .

(b) If $C \bullet \not\subseteq \bullet S$ then

i. if (5) is admissible, then C is added to \mathcal{N}_i to enforce (5), and (5) is added to (L, b) .

ii. if (5) is not admissible but it can be transformed to an admissible constraint (7), then C is added to \mathcal{N}_i to enforce (7), and (7) is added to (L, b) .

iii. if (5) is not admissible and the procedure could not transform it to an admissible constraint (7), let $X = X \cup S \bullet$.

- (c) If a constraint has been added to (L, b) or (L_0, b_0) in the previous step (b) and the procedure started with initial constraints, it is verified whether the system $L\mu \geq b$ and $L_0\mu \geq b_0$ is feasible. If not feasible, the constraint added to (L, b) or (L_0, b_0) at (b) is deleted, the control place (if any) added at (b) is deleted, and $X = X \cup S\bullet$.
3. If the Petri net is no longer PT-ordinary, the Petri net is PT-transformed.
 4. If the Petri net is no longer with asymmetric choice, the Petri net is AC-transformed, 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 2, 3 and 4.
 6. **If** the set X has been changed in the current iteration (X represents the set of transitions which must not appear in the active subnet) the new active subnet is recomputed, **else** it is obtained using the update algorithm (see section 4.4).
 7. Let T^A be the set of transitions of the new 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 and $i = i + 1$.

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 by removing the columns of L and L_0 corresponding to places not in \mathcal{N}_0 .

E. The redundant constraints of (L, b) and (L_0, b_0) are removed.

F. The supervisor of \mathcal{N}_0 is built according to the constraints (L, b) (Theorem 2.1).

4.6 Remarks

1. The purpose of the procedure is to produce two sets of linear constraints on the marking of the target net: $L\mu \geq b$ 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 by Theorem 6.1. Liveness enforcement corresponds to $T = T_0$. When an infeasibility occurs in a step C.2.c or a failure at step C.2.b.iii, the supervisor enforces T' -liveness, for some $T' \subset T$.

2. If no initial constraints are given and all transitions are controllable and observable, the procedure is optimal, in the sense that if it terminates it gives the least restrictive T -liveness supervisor, if a T -liveness supervisor exists. This is proved in Theorem 6.2.
3. Optionally, initial constraints (L_I, b_I) may be used to tell the procedure that all reachable markings μ satisfy $L_I\mu \geq b_I$ whenever $L_I\mu_0 \geq b_I$ for the initial marking μ_0 , and that markings μ such that $L_I\mu \not\geq b_I$ are undesired. A more general usage of the initial constraints appears in section 8. Initial constraints reflect structural properties of the target net \mathcal{N}_0 , such as place invariants. For instance, when \mathcal{N}_0 is the closed loop obtained by supervision based on place invariants ([9, 16], Theorem 2.1), $L_I\mu \geq b_I$ can be used to specify the place invariants.
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 \mathcal{N}_0 in closed loop with the supervisor enforcing (L, b) for all initial markings μ_0 of \mathcal{N}_0 which satisfy $L_0\mu_0 \geq b_0$ and $L\mu_0 \geq b$.
5. The failure at step B may occur only if the structure of the Petri net does not allow any of the transitions in T to be made live, or if initial constraints are given and none of the transitions in T can be made live for any of the markings satisfying the initial constraints, or if uncontrollable and/or unobservable transitions are present and the procedure systematically fails to generate admissible constraints at the steps C.2.b.ii.

5 Examples

Example 5.1 (Liveness enforcement) Consider the repetitive Petri net of Figure 2(a), where t_1 is unobservable. In the first iteration there are two minimal siphons: $\{p_1, p_3\}$ and $\{p_2, p_3\}$. Consider the siphon $\{p_1, p_3\}$. The marking constraint (5) is $\mu(p_1) + \mu(p_3) \geq 1$, and is not admissible. Thus it is transformed to the admissible constraint $2\mu(p_1) + \mu(p_3) \geq 1$, which is in the form (7). Then the control place C_1 is added to enforce this constraint, and the constraint is added to (L, b) . The place invariant (8) is $\mu(C_1) = 2\mu(p_1) + \mu(p_3) - 1$. Similarly, C_2 enforces $2\mu(p_2) + \mu(p_3) \geq 1$ on $\{p_2, p_3\}$, which is added to (L, b) , and (8) is $\mu(C_2) = 2\mu(p_2) + \mu(p_3) - 1$.

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 C.2.a of the procedure. The form (11) of $\mu(C_1) + \mu(C_2) \geq 1$ is $2\mu(p_1) + 2\mu(p_2) + 2\mu(p_3) \geq 3$, which is added to (L_0, b_0) .

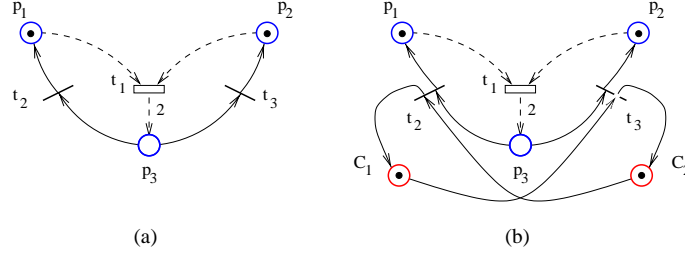


Figure 2: Example 5.1: (a) \mathcal{N}_0 ; (b) the final Petri net supervised for deadlock-freedom

Then the procedure terminates, since there is no new uncontrolled siphon in the third iteration. The final matrices (L, b) and (L_0, b_0) are:

$$L = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad L_0 = \begin{bmatrix} 2 & 2 & 2 \end{bmatrix} \quad b_0 = \begin{bmatrix} 3 \end{bmatrix}$$

The supervised net is shown in Figure 2(b). For all initial markings μ_0 such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$, liveness is enforced in a least restrictive manner (Theorems 6.1 and 6.2).

Example 5.2 (T-liveness enforcement) Consider the Petri net of Figure 3(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 . We want to enforce T -liveness for $T = \{t_4, t_5\}$.

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 3(d)). The active subnets are shown in Figure 3(c). The inequality associated with C_1 is $\mu(p_1) + \mu(p_2) + \mu(p_3) \geq 1$, which is added to (L, b) . Due to the subsequent AC-transformation, the invariant introduced by C_1 has the form (10): $\mu(C_1) = \mu(p_1) + \mu(p_2) + \mu(p_3) - \mu(p_{1,2}) - \mu(p_{2,2}) - \mu(p_{3,2})$.

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:

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

There is one redundant constraint, so the final constraints are $L = [2, 2, 1]$ and $b = 2$. The supervised net is shown in Figure 3(f). For all initial markings μ_0 satisfying $L\mu_0 \geq b$, T -liveness is enforced in a least restrictive manner (Theorems 6.1 and 6.2).

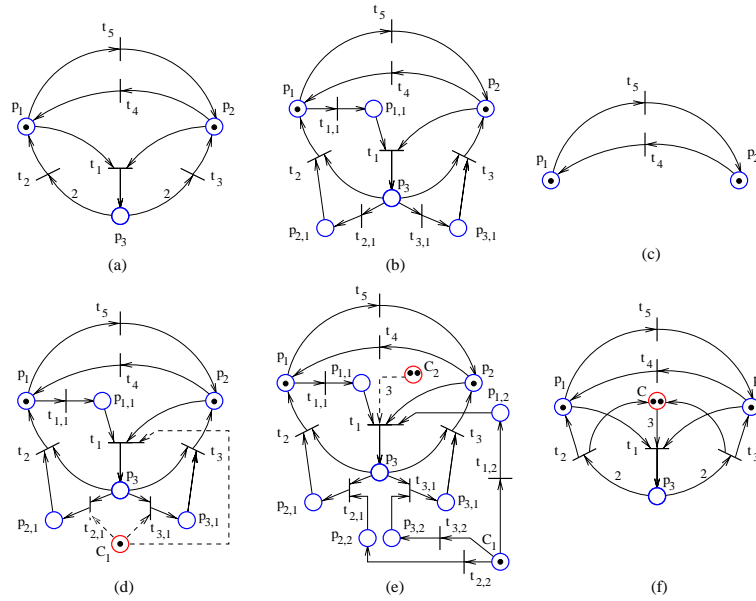


Figure 3: Example 5.2: (a) \mathcal{N}_0 ; (b) \mathcal{N}_1 ; (c) \mathcal{N}_1^A , the same as \mathcal{N}_2^A and \mathcal{N}_3^A ; (d) \mathcal{N}_1 and the added control place; (e) \mathcal{N}_2 and added control place; (f) \mathcal{N}_0 supervised for T -liveness

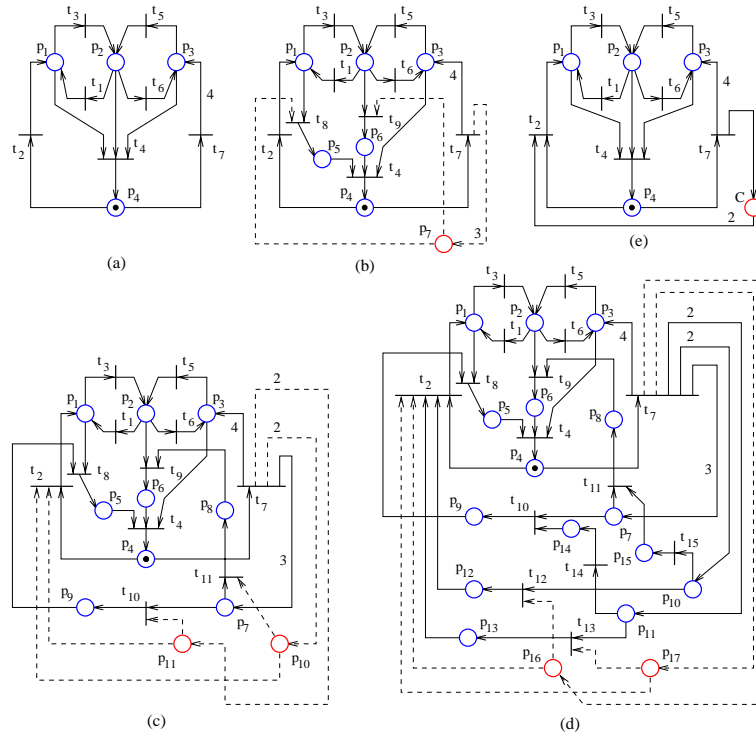


Figure 4: Example 5.3: (a) \mathcal{N}_0 ; (b) \mathcal{N}_1 ; (c) \mathcal{N}_2 ; (d) \mathcal{N}_3 ; (e) the supervised Petri net.

Example 5.3 (Liveness enforcement) Consider the Petri net of Figure 4(a) for liveness enforcement. The intermediary Petri nets \mathcal{N}_1 , \mathcal{N}_2 and \mathcal{N}_3 are represented in Figure 4(b-d), where the control places added to \mathcal{N}_1 , \mathcal{N}_2 and \mathcal{N}_3 are connected with dashed lines to the existing transitions. In the first iteration there is a single minimal siphon, $\{p_1, p_2, p_3, p_4\}$, and the control place p_7 is added. In the second iteration there are two new minimal siphons: $\{p_4, p_5, p_7, p_8\}$ and $\{p_4, p_6, p_7, p_9\}$ and two control places p_{10} and p_{11} , respectively, are thus added. In the third iteration there are two new minimal siphons: $\{p_4, p_6, p_9, p_{10}, p_{15}\}$ and $\{p_4, p_5, p_8, p_{11}, p_{14}\}$, and so the control places p_{16} and p_{17} , respectively, are added. At the fourth iteration no new minimal siphons are found, and so the procedure terminates. The constraints enforced by p_7 , p_{10} , p_{11} , p_{16} and p_{17} are respectively

$$\begin{aligned} \mu(p_1) + \mu(p_2) + \mu(p_3) + \mu(p_4) &\geq 1 \\ \mu(p_1) + \mu(p_2) + \mu(p_3) + 2\mu(p_4) + \mu(p_5) - \mu(p_9) &\geq 2 \\ \mu(p_1) + \mu(p_2) + \mu(p_3) + 2\mu(p_4) + \mu(p_6) - \mu(p_8) &\geq 2 \\ \mu(p_1) + \mu(p_2) + \mu(p_3) + 3\mu(p_4) + \mu(p_5) + \mu(p_6) - \mu(p_{12}) &\geq 3 \\ \mu(p_1) + \mu(p_2) + \mu(p_3) + 3\mu(p_4) + \mu(p_5) + \mu(p_6) - \mu(p_{13}) &\geq 3 \end{aligned}$$

After removing the redundant constraints, the supervisor of \mathcal{N}_0 is defined by $L = [1, 1, 1, 3]$ and $b = 3$, and is the least restrictive liveness enforcing supervisor (Theorems 6.1 and 6.2).

6 Proof of the Liveness Procedure

The proofs of the following results use the notations of the liveness procedure (section 4.5), e.g. the Petri net at the beginning of iteration i is $\mathcal{N}_i = (P_i, T_i, F_i, W_i)$, and the active subnet $\mathcal{N}_i^A = (P_i^A, T_i^A, F_i^A, W_i^A)$. Additionally we introduce the following definitions and notations. A marking μ of \mathcal{N}_i is **valid** if for all control places added in the iterations $1 \dots i - 1$ the invariant equations of the form (10) hold true, and if $\mu(p) = 0$ for all places p other than control places and places of \mathcal{N}_0 . Two *valid* markings μ_i and μ_j of \mathcal{N}_i and \mathcal{N}_j are **equivalent** if $\mu_i(p) = \mu_j(p)$ for all places p of \mathcal{N}_0 . We also say that a **siphon control failure** occurs when no admissible constraint is found at step C.2.b.ii or in case of infeasibility at step C.2.c. Next we introduce firing sequence notations. Both the PT and AC transformations (section 4.2) perform transition splits. A transition t_i may be split in more than just one iteration, the transitions $t_{i,k}$ (where $t_{i,k}$ resulted by splitting t_i) may also be split in subsequent iterations, and so on. We denote by $\sigma_{0,j}(t)$ an arbitrary transition sequence of \mathcal{N}_j such that (a) $\sigma_{0,j}(t)$ enumerates the transitions (including t itself) in which t of \mathcal{N}_0 is successively split until (and including) the iteration $j - 1$, and (b) valid markings μ of \mathcal{N}_j exist such that μ enables

$\sigma_{0,j}(t)$. In this way firing the sequence $\sigma_{0,j}(t)$ in \mathcal{N}_j corresponds to firing t in \mathcal{N}_0 . If t is not split, we let $\sigma_{0,j}(t) = t$. The notation $\sigma_{i,j}(t)$ for $i < j$ and t in \mathcal{N}_i , is similarly defined by taking \mathcal{N}_i instead of \mathcal{N}_0 . If $\sigma = t_1 t_2 t_3 \dots$, we let $\sigma_{i,j}(\sigma) = \sigma_{i,j}(t_1) \sigma_{i,j}(t_2) \sigma_{i,j}(t_3) \dots$. For instance, in Example 5.2 $\sigma_{0,2}(t_2) = t_{2,1} t_2$, in Example 5.3 $\sigma_{0,1}(t_4)$ is any of $t_8 t_9 t_4$ and $t_9 t_8 t_4$ and $\sigma_{2,3}(t_{10}) = t_{14} t_{10}$.

Theorem 6.1 *Assume that the procedure terminates and that it does not fail at a step B. Let $\mathcal{N}_0 = (P_0, T_0, F_0, W_0)$ be the target 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 and $\mathcal{N}_k^A = (P_k^A, T_k^A, F_k^A, W_k^A)$ the active subnet at the last iteration. If \mathcal{N}_k^A is nonempty, then \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.ii or C.2.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 μ of the original Petri net \mathcal{N}_0 which satisfies the constraints $L\mu \geq b$ and $L_0\mu \geq b_0$ has an equivalent marking μ_k in \mathcal{N}_k such that all active siphons of \mathcal{N}_k are not empty. (Indeed, it can be easily noticed that there is no siphon which only contains places resulted from transition splits. Therefore no constraint is “lost” at the step D of the procedure, and so the equivalent marking μ_k exists). Thus (\mathcal{N}_k, μ_k) has all siphons controlled (otherwise \mathcal{N}_k would not be the Petri net of the last iteration) and so is T_k^A -live, by Theorem 3.2. Assume that from an initial marking μ_0 of \mathcal{N}_0 satisfying $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$, 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, (\mathcal{N}_k, μ_k) is T_k^A -live, and so 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 transition split operations in all iterations. Let \mathcal{C} be the set of control places. It can be proved by induction on i , the iteration number, that firing any $t \in T_R$ always reduces the marking of some places in $P_0 \cup \mathcal{C}$, while firing $t_x \in T_0$ (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 $t_x \in T_0$. Let t_1 be the first transition in T_0 that appears in σ . Then we can write σ as $\sigma = \sigma_1 \sigma'_1$, where t_1 appears only once in σ_1 . It can be proved that σ_1 contains a subsequence $\sigma_{0,k}(t_1)$ (we prove this as Proposition 6.13 in [6]). 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 $\sigma_{0,k}(t_1)\sigma_{0,k}(t_2) \dots \sigma_{0,k}(t_j)$. But this implies that μ enables $t_1 t_2 \dots t_j$ in \mathcal{N}_S , and since $t_j = t$, t is not dead in (\mathcal{N}_S, μ) , which is a contradiction.

The only situation in which \mathcal{N}_{i+1}^A does not contain all transitions of \mathcal{N}_i^A is when a siphon control failure has occurred in the iteration i . Therefore, if no siphon control failures occur, $T_0^A \subseteq T_k^A$ and so $T_0^A \subseteq T_x$, which implies that \mathcal{N}_S is T_y -live, for $T_y = T \cap T_0^A$ and μ_0 satisfying $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$. \square

If the problem is well formulated and thus T -liveness is possible, by Theorem 6.1, the procedure will enforce T -liveness, if no siphon control failures occur. This always is the case for Petri nets which have all transitions controllable and observable when no initial constraints are given. When initial constraints are given, assuming that the Petri net can be made T -live for some initial marking satisfying the initial constraints, we guarantee in Proposition 6.1 that T -liveness is enforced, under an additional assumption. When siphon control failures occur, Theorem 6.1 shows that the procedure enforces T_y -liveness for some $T_y \subset T$. The case when T -liveness cannot be enforced due to the structure of the Petri net is when $T \not\subseteq T_0^A$. In such a case the procedure attempts enforcing $T \cap T_0^A$ -liveness.

The siphon control approach used by the procedure enforces inequalities of the form (7) 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 result is 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 all transitions controllable and observable. The meaning of the requirement is that all minimal active siphons S are controlled in a maximally permissive way (that is, only the markings μ which satisfy $\mu(p) = 0 \forall p \in S$ are forbidden.) In the next theorem we prove that the procedure generates least restrictive T -liveness supervisors for a large class of Petri nets. In addition to the assumption on (7) we also require that there is a single T -minimal active subnet. Slightly more general conditions can be used instead, with little changes in the proof, such as that in all iterations the active siphons are the same regardless of which T -minimal active subnet we would take.

Theorem 6.2 *Assume that for all minimal active siphons S the procedure is able to find admissible constraints of the form (7) with all 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 which enforce T -liveness for some initial marking(s) satisfying the initial constraints. 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, μ_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 $(\mathcal{N}_i, \mu_{0,i})$ 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 $\mu_{i,0}$. Assume the contrary. Let σ be an infinite firing sequence enabled by $\mu_{i,0}$ such that all transitions of T occur infinitely often in σ . Since $(\mathcal{N}_{i+1}, \mu_{0,i+1})$ cannot be made T -live, $\sigma' = \sigma_{i,i+1}(\sigma)$ is not enabled in \mathcal{N}_{i+1} . Then $\sigma = \sigma_1 t_1 \sigma_2$, $\mu_{0,i}[\sigma_1 > \mu_1$, $\mu_{0,i+1}[\sigma_{i,i+1}(\sigma_1) > \mu'_1$, 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, μ_1) .

An empty active siphon implies a set T_x of dead transitions from the active subnet. Therefore the transitions in T_x do not appear infinitely often in σ . Let $T_{x1} = \{t \in T_1^A : \exists t_u \in \sigma_{1,i}(t) \text{ and } t_u \in T_x\}$. Since we are in the case of no initial constraints, the theorem assumption implies that no siphon control failures occur. Then, the active subnets \mathcal{N}_i^A for $i > 1$ are computed using the update algorithm of section 4.4, so $T_{x1} \subseteq T_1^A$. Using the same construction as in the proof of Theorem 6.1, 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 . Note that the transitions of T_{x1} do not appear infinitely often in σ_1 . We apply Lemma 3.1 for \mathcal{N}_1 and σ_1 , and using the notation of Lemma 3.1, we let $T_x^A = \|x\|$; T_x^A defines an active subnet and $T \subseteq T_x^A$, as all transitions of T appear infinitely often in σ_1 . However T_1^A is not a subset of T_x^A , for $T_{x1} \subseteq T_1^A \setminus T_x^A$. Therefore \mathcal{N}_1^A is not the single T -minimal subnet, and this is a contradiction.

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 . The markings forbidden at every iteration i are those for which there are empty active siphons. Therefore \mathcal{N}_i has an empty active siphon for $\mu_{0,i}$, where $\mu_{0,i}$ is the equivalent marking of μ_0 in \mathcal{N}_i . By the paragraph above, this implies that $(\mathcal{N}_i, \mu_{0,i})$ cannot be made T -live, and by the first part of the proof this implies that (\mathcal{N}_0, μ_0) cannot be made T -live, which is a contradiction. Therefore all T -liveness enforcing supervisors forbid the markings such that $L\mu \not\geq b$ or $L_0\mu \not\geq b_0$.

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. \square

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 6.1 *Assume that for all minimal active siphons S the procedure is able to find admissible constraints of the form (7) 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 6.2 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 first assumption of the theorem is always true for Petri nets with all transitions controllable and observable. Therefore, whenever the procedure is used to enforce liveness, no uncontrollable and unobservable transitions exist, and the procedure ends successfully, the supervisor generated is 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 include section 8.

When Theorem 6.2 applies, we know that T -liveness is not enforceable in \mathcal{N}_0 for any initial marking μ_0 which does not satisfy $L_0\mu_0 \geq b_0$ and $L\mu_0 \geq b$. The next result is a consequence of Theorem 6.2, and it deals with the question whether there is some supervisor enforcing T -liveness when the supervisor generated by our procedure does not enforce T -liveness.

Proposition 6.1 *T -liveness is not enforceable in \mathcal{N}_0 for any initial marking if $T \not\subseteq T_0^A$. Furthermore, in the conditions of Theorems 6.1 and 6.2, if $T \not\subseteq T_k^A$, then T -liveness is not enforceable in \mathcal{N}_0 for any initial marking satisfying the initial constraints.*

Proof: The first part is a consequence of Theorem 3.1, as the algorithm of section 4.4 does not fail to find a T -minimal active subnet if such a subnet exists. 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 at step C.2.c 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. No transition $t \in S_x^\bullet$ can be live in \mathcal{N}_j for valid initial markings; similarly to the proof of Theorem 6.2, 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 6.2, $T \subseteq T_l$ is not possible, as it would imply that \mathcal{N}_0^A is not the only T -minimal active subnet. 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 $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_j$ and the marking of the places of 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 . \square

7 Extending Permissivity

Theorem 6.2 and Corollary 6.1 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 $\mathcal{N}_0^{A,1}, \mathcal{N}_0^{A,2}, \dots, \mathcal{N}_0^{A,p}$. Theorem 6.2 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 $i = 1 \dots p$). Assume that the procedure terminates for all $i = 1 \dots p$ 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

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

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

Theorem 7.1 *Assume that for each $i = 1 \dots u$, for all minimal active siphons S the procedure is able to find admissible constraints of the form (7) with all α_p positive integers. Assume also that for each $i = u+1 \dots p$ 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 (7) 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 6.2 applies, and so for the given initial constraints $T_0^{A,i}$ -liveness cannot be enforced for all $i = u + 1 \dots p$. 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 $\mathcal{N}_0^{A,i}$, $i = 1 \dots p$, 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 6.2 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. \square

8 An Extension for Guaranteed Termination

The siphon control method is modified as follows. Let S be an uncontrolled siphon. Instead of enforcing (5), the constraint $\sum_{p \in S \cap R} \mu(p) \geq 1$ is used, where R is the set of places which are not obtained from transition splits. Similarly, the admissible constraints (7) are built such that $\alpha_p = 0 \forall p \in R$.

Theorem 8.1 *Let \mathcal{N}_0 be a Petri net and (L_I, b_I) be a set of marking constraints $L_I\mu \geq b_I$, with bounded feasible region. The modified liveness enforcement procedure terminates if started with initial constraints (L_I, b_I) .*

Proof: Let $L'_I\mu \geq b'_I$ be $L_I\mu \geq b_I$ transformed according to (9) in the step A. The usage of initial constraints assumes that there is a set of initial markings \mathcal{M}_I of \mathcal{N}_0 such that

$\forall \mu_0 \in \mathcal{M}_I \forall \mu \in \mathcal{R}(\mathcal{N}_0, \mu_0): L_I \mu \geq b_I$. It can be noticed that if $L_I \mu \geq b_I$ is satisfied $\forall \mu \in \mathcal{R}(\mathcal{N}_0, \mu_0)$ for some μ_0 , $\mu_{0,i}$ is the equivalent marking of μ_0 in \mathcal{N}_i , $i \geq 1$, μ_i is the marking of \mathcal{N}_i , and $\mu_{r,i}$ is the restriction of μ_i to the places of \mathcal{N}_1 , then $\forall \mu_{r,i} \in \mathcal{R}(\mathcal{N}_i, \mu_{0,i}): L'_I \mu_{r,i} \geq b'_I$. Let \mathcal{M}_R be the feasible set of $L'_I \mu \geq b'_I$. Since the feasible set of $L_I \mu \geq b_I$ is bounded (and so finite), so is \mathcal{M}_R . The modification above of (5) and (7) ensures that all constraints added by the procedure are only expressed in terms of the markings of the places of the target net \mathcal{N}_0 ; the marking of the places of the split replacements is never taken in account. So each time a new constraint is added to (L, b) , at least one new marking of \mathcal{M}_R is forbidden. Because \mathcal{M}_R is finite, after a finite number of iterations, an iteration is reached in which all new siphons (if any) considered in the step C.1 of the procedure are implicitly controlled, and so the procedure terminates. \square

Given a Petri net \mathcal{N} , the usage of the modified procedure is summarized below:

- Let \mathcal{M}_I be the set of initial markings of interest. Find a set of marking constraints $L_I \mu \geq b_I$ with bounded feasible set F , such that $\forall \mu_0 \in \mathcal{M}_I: \mathcal{R}(\mathcal{N}, \mu_0) \subseteq F$.
- Use the modified procedure with initial constraints (L_I, b_I) (or with (L_I, b_I) in addition to other initial constraints).
- The supervisor can be used for all initial markings $\mu_0 \in \mathcal{M}_I$ satisfying $L \mu_0 \geq b$ and $L_0 \mu_0 \geq b_0$.

The disadvantage of this procedure extension is that Theorem 6.2 does not always apply. Another problem of this extension is that in principle it is possible to get infeasible constraints $L \mu_0 \geq b$ and $L_0 \mu_0 \geq b_0$, in which case the procedure will fail and exit at step B. In the case of normal termination, Theorem 6.1 still applies, as shown in the next theorem.

Theorem 8.2 *Let $L_I \mu \geq b_I$ be a set of marking constraints with bounded feasible region F and \mathcal{M}_I a set of initial markings of \mathcal{N}_0 such that $\forall \mu_0 \in \mathcal{M}_I: \mathcal{R}(\mathcal{N}_0, \mu_0) \subseteq F$. Consider the modified liveness enforcement procedure with initial constraints (L_I, b_I) . Theorem 6.1 applies with the only additional requirement that the initial markings μ_0 of \mathcal{N}_0 should satisfy $\mu_0 \in \mathcal{M}_I$ in addition to $L \mu_0 \geq b$ and $L_0 \mu_0 \geq b_0$.*

Proof: The proof of Theorem 6.1 applies for the following reasons. First, Theorem 8.1 guarantees termination. Second, either of $\sum_{p \in S \cap R} \mu(p) \geq 1$ and $\sum_{p \in S \cap R} \alpha_p \mu(p) \geq 1$ implies $\sum_{p \in S} \mu(p) \geq 1$ when $S \cap R \neq \emptyset$ (in view of our requirement for admissible constraints (7) that at least two coefficients α_p are positive). But it can be shown that it is impossible to have

siphons only made up of places from transition splits, so $S \cap R \neq \emptyset$. Therefore the siphons are indeed controlled at the step C.2 of the procedure. Third, using the notations of the proof of Theorem 8.1, the siphons considered to be implicitly controlled due to constraints in $L'_I \mu \geq b'_I$ are indeed controlled, since the constraints $L'_I \mu \geq b'_I$ are true in all the intermediary nets \mathcal{N}_i , $i \geq 1$, for all markings reachable from valid markings $\mu_{0,i}$ equivalent to markings $\mu_0 \in \mathcal{M}_I$ of \mathcal{N}_0 . \square

9 Conclusion

In this paper we have introduced a procedure which synthesizes a supervisor enforcing the liveness of the transitions in a given set. The approach is based on structural properties of Petri nets and may be used for arbitrary Petri nets. For a large class of Petri nets the procedure is optimal, in the sense that the synthesized supervisors are least restrictive. Termination is not guaranteed, and so we have included a suboptimal procedure with guaranteed termination.

APPENDIX

A Proof of Lemma 3.1

Proof: Consider firing σ , and let μ_0 be the marking reached after all transitions which appear finitely often in σ have fired. Let $\sigma = \sigma_0 \sigma_1 \sigma_2 \dots \sigma_k \dots$ 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, \dots 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 $V_n = \{y \in \mathbb{N}^n : \bar{\exists} y_i \in V_n, y \neq y_i, y \geq y_i \text{ or } y \leq y_i\}$. Next it is proved by induction that V_n is finite (i.e. it cannot have infinitely many elements). Assume that any V_{n-1} is finite. Then, let $y_{s,n} \in V_n$; $V_n \subseteq \bigcup_{k,u} C_{k,u}$, where $C_{k,u} = \{y \in \mathbb{N}^n : y(j_k) = u, y(i_k) > y_{s,n}(i_k), \bar{\exists} y_i \in V_n, y \neq y_i, y \geq y_i \text{ or } y \leq y_i\}$, is defined for $0 \leq u < y_{s,n}(j_k)$ and $k = 1, 2, \dots, n(n-1)$ corresponds to the possibilities in which $i_k \neq j_k$, $0 \leq i_k, j_k \leq n$ can be chosen. The induction assumption implies that each $C_{k,u}$ is finite, because the component j_k of the vectors is fixed and only the remaining $n-1$ can be varied. So V_n is finite.

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

Next we show 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 $\mathcal{M}^{(2)}, \dots, \mathcal{M}^{(j)}$, 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 $x(i) = 0 \forall t_i \in T \setminus U$. \square

B Proof of Theorem 3.1

Proof: Let $\|x\|$ be the *support* of the vector x , that is $\|x\| = \{i : x(i) \neq 0\}$. There is an integer vector $x \geq 0$ such that $Dx \geq 0$ and $\|w\| \subseteq \|x\|$ for all integer vectors $w \geq 0$ which satisfy $Dw \geq 0$. 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 $y(j) > 0$. By the definition of x , $\|y\| \subseteq \|x\|$ and so $t_j \in \|x\|$. This proves that all transitions which 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 such that every $t_i \in T$ appears $x(i)$ times in σ_x . Then there is a marking μ_0 which enables the infinite firing sequence $\sigma_x \sigma_x \sigma_x \dots \sigma_x \dots$. Also, we may choose Ξ to restrict all possible firings to the former infinite firing sequence. Hence all transitions in $\|x\|$ can be made live. \square

C Proof of Theorem 3.2

We first need to prove the following result.

Lemma C.1 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: Since \mathcal{N} has asymmetric choice, any n places such that $p_i \bullet \cap p_j \bullet \neq 0$, $\forall i, j \in \{1, 2, \dots, n\}$ satisfy $p_{i_1} \bullet \subseteq p_{i_2} \bullet \subseteq \dots \subseteq p_{i_n} \bullet$, where i_1, \dots, i_n are distinct and $i_j \in \{1, 2, \dots, n\}$ for all $j = 1 \dots n$. Let $\bullet t = \{p_1, \dots, p_n\}$, where the notation is chosen such that $p_1 \bullet \subseteq p_2 \bullet \subseteq \dots \subseteq p_n \bullet$. We first prove that $\exists \mu_1 \in \mathcal{R}(\mathcal{N}, \mu)$ and $\exists j \in \{1, \dots, n\}$ 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 $\{1, \dots, n\}$ 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 $\exists \mu_{i,2} \in \mathcal{R}(\mathcal{N}, \mu_{i,1})$: $\mu_{i,2}(p_i) \geq 1$. 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 $\{1, \dots, n\}$ 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 $t_i \in p_j \bullet \forall j = i \dots n$, so $\mu_{i,4}(p_j) \geq 1 \forall j = i \dots n$. By the choice of i , $\mu_{i,4}(p_j) \geq 1 \forall j = 1 \dots i - 1$. Therefore $\mu_{i,4}$ enables t . Contradiction.

Therefore, $\exists \mu_1 \in \mathcal{R}(\mathcal{N}, \mu)$ and $\exists j \in \{1, \dots, n\}$ 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, \dots, S_{m+1} and the markings μ_2, \dots, μ_{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 $S_{i+1} = S_i \cup \bigcup_{t_x \in \bullet(S_i \setminus S_{i-1})} \{p \in \bullet t_x : \forall \mu_x \in \mathcal{R}(\mathcal{N}, \mu_{i+1}) : \mu_x(p) = 0\}$. 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 $\mu' \in \mathcal{R}(\mathcal{N}, \mu)$. \square

We can now prove Theorem 3.2.

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 Lemma C.1 there is a reachable marking such that a siphon S is empty and $t \in S\bullet$. Then $t \in S\bullet$ implies $S \cap P^A \neq \emptyset$. From $\bullet S \subseteq S\bullet$ we infer $\bullet S \cap T^A \subseteq S\bullet \cap T^A$. If $t_x \in T^A$ and for some $p \in P$: $t_x \in p\bullet$, then $p \in P^A$, by Definition 3.1. 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 . Therefore S is an active siphon. Contradiction, for S is empty. \square

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 *LNCS; Proc. 17th ICATPN*, volume 1091, pages 57–72. Springer-Verlag, June 1996.
- [3] J. Ezpeleta, J. M. Colom, and J. Martínez. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Robot. Autom.*, 11(2):173–184, 1995.
- [4] 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.
- [5] 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.
- [6] M. V. Iordache, J. O. Moody, and P. J. Antsaklis. Automated synthesis of liveness enforcement supervisors using Petri nets. Technical report of the isis group, isis-2000-004, University of Notre Dame, September 2000.
- [7] 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 American Control Conference*, pages 3167–3171, June 2000.
- [8] 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.
- [9] J. O. Moody and P. J. Antsaklis. *Supervisory Control of Discrete Event Systems Using Petri Nets*. Kluwer Academic Publishers, 1998.
- [10] T. Murata. Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*, pages 541–580, April 1989.
- [11] 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.
- [12] S. R. Sreenivas. On a free-choice equivalent of a Petri net. In *Proceedings of the 36th IEEE Conference on Decision and Control*, San Diego, California, December 1997.

- [13] S. R. Sreenivas. On the existence of supervisory policies that enforce liveness in discrete event dynamic systems modeled by controlled Petri nets. *IEEE Trans. Autom. Contr.*, 42(7):928–945, July 1997.
- [14] 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.
- [15] S. R. Sreenivas. On supervisory policies that enforce liveness in a class of completely controlled Petri nets obtained via refinement. *IEEE Trans. on Autom. Contr.*, 44(1):173–177, January 1999.
- [16] 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.