M.V. Iordache, P.J. Antsaklis, “T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

CDC01-REG1398

T-Liveness Enforcement in Petri Nets Based on Structural Net Properties

Marian V. Iordache and Panos J. Antsaklis*

Abstract

We introduce a semidecidable procedure which, given a Petri net structure and a set T of transitions, synthesizes a supervisor enforcing the transitions in T to be live. We call this liveness property T-liveness. When T equals the total set of Petri net transitions, T-liveness corresponds to liveness. Enforcing only a subset of transitions to be live is useful when some Petri net transitions model undesired events such as failures, and/or when the Petri net structure does not allow enforcing all transitions to be live. Our procedure is based on structural net properties, and so the synthesized supervisors are independent of the initial marking. The supervisors are least restrictive for a wide class of Petri nets. No assumptions are made on the Petri net structure: the Petri nets may be unbounded and have integer weights. In this paper we restrict our attention to fully controllable and observable Petri nets. However we note that the procedure is rather easily extendable to Petri nets having uncontrollable and unobservable transitions.

1 Introduction

Petri nets are a compact representation of concurrent discrete event systems. One of the difficult problems in the study of Petri nets is liveness. Liveness means that regardless of the current state of the Petri net, any transition can be eventually fired. In other words no deadlocks occur. In this paper we introduce a procedure which synthesizes supervisors for T-liveness. Such a supervisor restricts the operation of the Petri net such that the supervised Petri net is T-live. While other liveness related problems have been extensively studied [1], there are not many literature results on the synthesis of liveness supervisors. Moreover, to the authors' knowledge, there are no results on T-liveness other than our results in [6]. The usual way to enforce liveness is based on the reachability graph of the Petri net [14]. Due to the state explosion problem, this approach is not considered to be satisfactory. A new approach attempting to reduce the computational load is based on *unfolding* [5]. Both afore mentioned approaches are limited to bounded Petri nets and

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

M.V. Iordache, P.J. Antsaklis, "T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

require a given initial marking. While our approach is a semidecidable procedure and so does not address satisfactorily the computational problem, it is the only procedure known to the authors which is applicable to general Petri nets, including unbounded Petri nets. Furthermore, it does not depend on the initial marking. This together with the facts that: (a) the procedure generates least restrictive liveness supervisors and usually least restrictive T-liveness supervisors, and (b) the domain of the supervisor is the feasible set of a set of linear inequalities, favor the usage of our procedure for solving resource optimization problems in manufacturing systems. In the literature there is an approach which allows the synthesis of liveness enforcing supervisors and does not depend on the initial marking [4]. However it only applies to a class of ordinary and conservative Petri nets. The approach of [4] has been extended to a larger class of conservative Petri nets in [12]. The algorithm of [12] has polynomial complexity, however the supervisors are not least restrictive. Another liveness enforcing approach for a restricted class of ordinary Petri nets is given in [13]

In this paper we generalize our liveness enforcement procedure in [8] to T-liveness enforcement. The only input required by our procedure is the Petri net structure and the set T. Since the supervisors are designed independently of the initial marking, when we say they are least restrictive we mean two things. First, for any initial marking for which the supervisor is defined, there is no other supervisor enforcing T-liveness which is less restrictive. Second, T-liveness cannot be enforced for all initial markings for which the supervisor is not defined. In this paper we consider Petri nets with controllable and observable transitions. However, the procedure is easily extendable to Petri nets with uncontrollable and unobservable transitions, and we include this extension in the technical report [7]. The procedure has been computer implemented and the program is available from the authors.

We begin with preliminary notations and definitions in section 2. The remaining part of the paper presents our contribution. Thus section 3 gives two theoretical results on which the T-liveness procedure relies; we considered them in detail in [6]. Section 4 describes the T-liveness procedure, beginning with the description of the operations it involves; the procedure is stated in section 4.5. Section 5 illustrates the procedure on simple examples. In section 6 we formally prove that the procedure synthesizes a least restrictive T-liveness supervisor.

2 Preliminaries

We assume the reader familiar with Petri nets. If this is not the case, we recommend the Petri net survey in [11]. We will denote a Petri net structure 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 $(\mathcal{N}, \mu_0)$ 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

M.V. Iordache, P.J. Antsaklis, "T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

D. We use the notation $\mu \xrightarrow{\sigma} \mu'$ to express that the marking μ enables the firing sequence σ and μ' is reached by firing σ.

We will refer to Petri nets in which the arcs from places to transitions have weights equal to one. We call such Petri nets **PT-ordinary**, because the only arcs $a \in F$ satisfying the requirement of an ordinary Petri net that $W(a) = 1$ are the arcs $a = (p, t)$ from a place p to a transition t. An **asymmetric choice** Petri net is defined by the property that for any two places p_i and p_j such that $p_i \bullet \cap p_j \bullet \neq \emptyset$, either of $p_i \bullet \subseteq p_j \bullet$ or $p_j \bullet \subseteq p_i \bullet$ is satisfied.

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.

A transition t of $(\mathcal{N}, \mu_0)$ is **live** if for all reachable markings there is an enabled firing sequence containing t. Given $T' \subseteq T$, $(\mathcal{N}, \mu_0)$ is T'**-live** if all transitions $t \in T'$ are live. In particular, when $T' = T$ and $(\mathcal{N}, \mu_0)$ is T'-live, $(\mathcal{N}, \mu_0)$ is **live**.

Given $\mathcal{N}$, let $\mathcal{M}$ the set of all markings of $\mathcal{N}$ and $U \subseteq \mathcal{M}$. We define a **supervisor** as a function $\Xi : U \to 2^T$ 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 $(\mathcal{N}, \mu_0)$ is supervised with Ξ. We say that **liveness can be enforced** in $\mathcal{N}$ if there is an initial marking μ_0 and a supervisor Ξ such that $(\mathcal{N}, \mu_0)$ supervised by Ξ is live. The Petri net structures in which liveness can be enforced (for some initial markings) are the *repetitive* Petri nets [11], and the Petri net structures in which T-liveness can be enforced for some T are the *partially repetitive* Petri nets.

The supervisory technique used in this paper is supervision based on place invariants [10, 15], in which the supervisor is defined by a set of linear marking inequalities. The supervision is accomplished by extending the Petri net with additional places, called **control places**. The construction is summarized in the following theorem.

Theorem 2.1 [10, 15] *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 supervisor with incidence matrix $D_c = -LD_p$ and initial marking $\mu_{c0} = b - L\mu_{p0}$ enforces the constraint $L\mu_p \leq b$ in the supervised system $D = [D_p^T, D_c^T]^T$. Furthermore, the supervision is maximally permissive.*

3 Theoretical Background

In this section we introduce a number of new results necessary for our T-liveness enforcement method. We consider these results in detail in [6, 7]. We begin with a technical result.

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 $Dx \geq 0$ and*

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

Next we define several key concepts for our method: *active subnets* and *active siphons*. An active subnet is a part of a Petri net which can be made live by supervision for appropriate markings. A siphon is active with respect to an active subnet if it contains places from that subnet.

Definition 3.1 *Let* $\mathcal{N} = (P, T, F, W)$ *and* D *the incidence matrix of* $\mathcal{N}_0$. *Then* $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$ *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* $Dx \geq 0$. *We say that* $\mathcal{N}^A$ *is* **T′-minimal** *if* $T' \subseteq T^A$ *and* $T^A \not\subseteq T^A_x$ *for any other active subnet* $\mathcal{N}^A_x = (P^A_x, T^A_x, F^A_x, W^A_x)$ *such that* $T' \subseteq T^A_x$.

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.)*

The next result is fundamental for our T-liveness procedure.

Theorem 3.1 *Given a PT-ordinary asymmetric choice Petri net* $(\mathcal{N}, \mu_0)$, *let* T *be a set of transitions and* $\mathcal{N}^A$ *a* T*-minimal active subnet. The Petri net is* T*-live (and also* T^A*-live) if all minimal active siphons with respect to* $\mathcal{N}^A$ *are controlled.*

4 The T-Liveness Enforcing Procedure

4.1 Introduction to the Procedure for T-Liveness Enforcement

Given a target Petri net $\mathcal{N}_0$, the liveness enforcing procedure generates a sequence of asymmetric choice PT-ordinary Petri nets, $\mathcal{N}_1$, $\mathcal{N}_2$, ... $\mathcal{N}_k$, 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 3.1, $\mathcal{N}_k$ 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

M.V. Iordache, P.J. Antsaklis, “T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

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 [9], 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: $t_{j,0}$, $t_{j,1}$, $t_{j,2}$, $\ldots$ $t_{j,m-1}$, where $n(t_j) = \max\{W(p, t_j) : (p, t_j) \in F\}$. Also, $m - 1$ new places are added: $p_{j,1}$, $p_{j,2}$, $\ldots$ $p_{j,m-1}$. 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 $i = 1 \ldots m-1$
>
> (ii) $\bullet t_{j,i} = \{p \in \bullet t_j : W(p, t_j) > i\}$, for $i = 0 \ldots m-1$
>
> (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. 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).

Algorithm of the AC-Transformation

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

Output: $\mathcal{N}'$

M.V. Iordache, P.J. Antsaklis, "T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

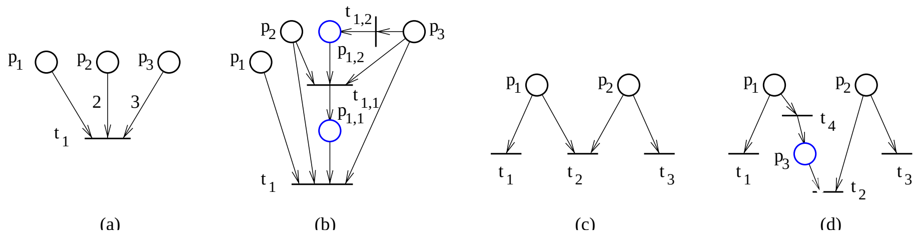

Figure 1: Illustration of the transition split: (a) initial configuration; (b) the effect of the PT-transformation; (c) initial configuration; (d) the effect of the AC-transformation.

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 $W'(p', t) = W(p, t)$.

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. We call the transformation to asymmetric choice Petri nets **AC-transformation**.

The second argument of the transformation, 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. 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

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

where μ is the marking. The siphon S can be invariant controlled in order to always satisfy (1). The invariant is created by adding an additional place, called **control place**, which we denote by C. See Theorem 2.1 or [4, 2, 3]. Thus the equation of the marking of C is

$$\mu(C) = \sum_{p \in S} \mu(p) - 1 \tag{2}$$

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 (2) is modified. It can be proved that the new form is

$$\mu(C) + \sum_{z=1}^{r} \mu(p_z) + \sum_{i=1}^{k} \sum_{j=1}^{m_i - 1} j\mu(p_{i,m_i-j}) = \sum_{p \in S} \mu(p) - 1 \tag{3}$$

where 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 used to describe the PT-transformation. The places p_z are the places resulting from the AC-transformation which satisfy $\bullet\bullet p_z = C$. Note that the siphon S remains controlled, that is (1) is still true. The procedure insures that (3) 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 (1) 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 (1) can be eliminated by using (3). By eliminating all control place markings, (1) can be written as:

$$l^T \mu \geq c \tag{4}$$

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 (4) 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 (1). If $C\bullet \subseteq \bullet S$, C is not needed and so it is deleted. In this case (1) 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 (1) stored in $L_0\mu \geq b_0$ are in the form (4). 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 (1) 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 $L_0\mu \geq b_0$.

4.4 The Computation of a T-minimal Active Subnet

The following algorithm computes a T-minimal active subnet or signalizes if none exists. A T-minimal active subnet does not exist iff some of the transitions of T cannot be made live under any circumstances.

Input: The Petri net $\mathcal{N}_0 = (P_0, T_0, F_0, W_0)$, its incidence matrix D, and a nonempty set of transitions $T \subseteq T_0$.

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

1. Check the feasibility of $Dx \geq 0$ s.t. $x \geq 0$, $x(i) \geq 1$ $\forall t_i \in T$.

   If feasible **then**

   i. Let $M = \|x_0\|$ and $x_s = x_0$, where x_0 is a solution of the feasibility problem.

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

      A. 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 $x(j) \geq 1$ $\forall t_j \in T$.

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

   Else no T-minimal subnet exists and so **return**.

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

Because of the iterative nature of the liveness procedure, the active subnet needs to be reevaluated at every iteration. However the algorithm above needs to be used only once, to compute $\mathcal{N}_0^A$. The

M.V. Iordache, P.J. Antsaklis, "T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

active subnets $\mathcal{N}_1^A$, $\mathcal{N}_2^A$, $\mathcal{N}_3^A$, ... can be computed by simply repeating the changes done to $\mathcal{N}_{i-1}$ in $\mathcal{N}_{i-1}^A$, $i = 1, 2, \ldots$. Such an update of the active subnets is done in 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.

4.5 The T-Liveness Enforcing Procedure

Input: The target Petri net $\mathcal{N}_0$ and a nonempty set of transitions T.

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. $\mathcal{N}_0$ is PT-transformed and then AC-transformed (section 4.2). The transformed net is $\mathcal{N}_1$.

B. T-minimal active subnets of $\mathcal{N}_0$ and $\mathcal{N}_1$ are computed (section 4.4). If none is found, T-liveness is impossible, and so the procedure terminates.

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 is *uncontrolled* if not implicitly controlled.)
   2. For every new uncontrolled minimal active siphon S:

      Let C be the control place which would result by controlling the siphon.

      (a) If $C\bullet \subseteq \bullet S$, then S does not need control and C is not added to $\mathcal{N}_i$. The constraint (4) is added to (L_0, b_0).

      (b) If $C\bullet \not\subseteq \bullet S$ then C is added to $\mathcal{N}_i$ to enforce (1), and (4) is added to (L, b).
   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.

M.V. Iordache, P.J. Antsaklis, "T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

6. The active subnet is updated according to the changes made in the steps 2, 3 and 4.
7. 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 $\mathcal{N}_0$, 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).

5 Examples

Example 5.1 Consider the Petri net of figure 2(a), which is not PT-ordinary and not with asymmetric choice. Three transitions cannot be made live, regardless of the initial marking: t_1, t_2, t_3. Therefore liveness cannot be enforced, however we may enforce T-liveness for $T = \{t_4, t_5\}$.

The first iteration begins with the PT and AC-transformed net $\mathcal{N}_1$. The active subnets are shown in Figure 2(c). 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 2(d)). In this case the inequality (1) is $\mu(p_1)+\mu(p_2)+\mu(p_3) \geq 1$, and 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 (3): $\mu(C_1) = \mu(p_1) + \mu(p_2) + \mu(p_3) - \mu(p_{1,2}) - \mu(p_{2,2}) - \mu(p_{3,2})$.

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 2(f). By Theorem 6.1 it is T-live for all initial markings μ_0 such that $L\mu_0 \geq b$, and by Theorem 6.2, the supervision is least restrictive.

Example 5.2 Consider the Petri net structure of Figure 3(a). The only transitions which can be made live are $\{t_1, t_2, t_3, t_4, t_5, t_6\}$. Assume that we only desire the transitions $t \in T$ to be live, where $T = \{t_1, t_2, t_3\}$. For this T-liveness problem the procedure generates the intermediary Petri nets $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ of Figure 3(c-e), 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 active siphon, $\{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$, and the control place p_{11} is added. In the second iteration there is again a single new minimal active siphon: $\{p_1, p_2, p_3, p_7, p_8, p_{10}, p_{11}, p_{14}\}$, and a control place p_{15}

M.V. Iordache, P.J. Antsaklis, "T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

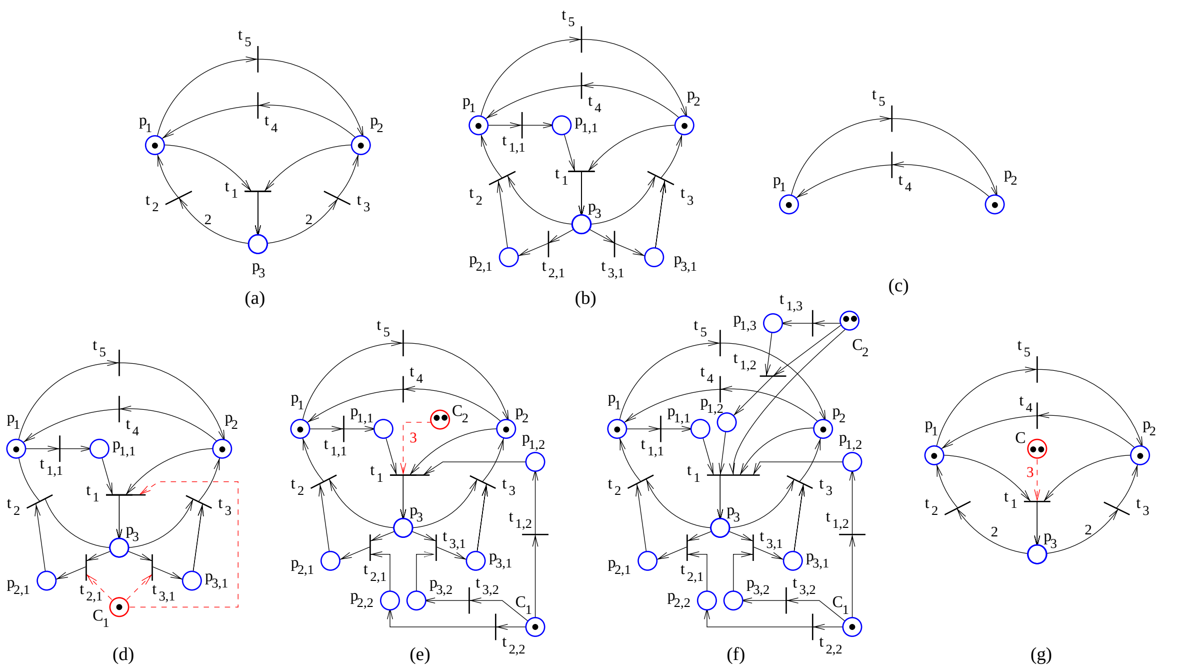

Figure 2: Example 5.1: (a) $\mathcal{N}_0$; (b) $\mathcal{N}_1$; (c) $\mathcal{N}_1^A$, the same as $\mathcal{N}_2^A$ and $\mathcal{N}_3^A$; (d) $\mathcal{N}_1$; (e) $\mathcal{N}_2$; (f) $\mathcal{N}_3$; (g) $\mathcal{N}_0$ supervised for T-liveness

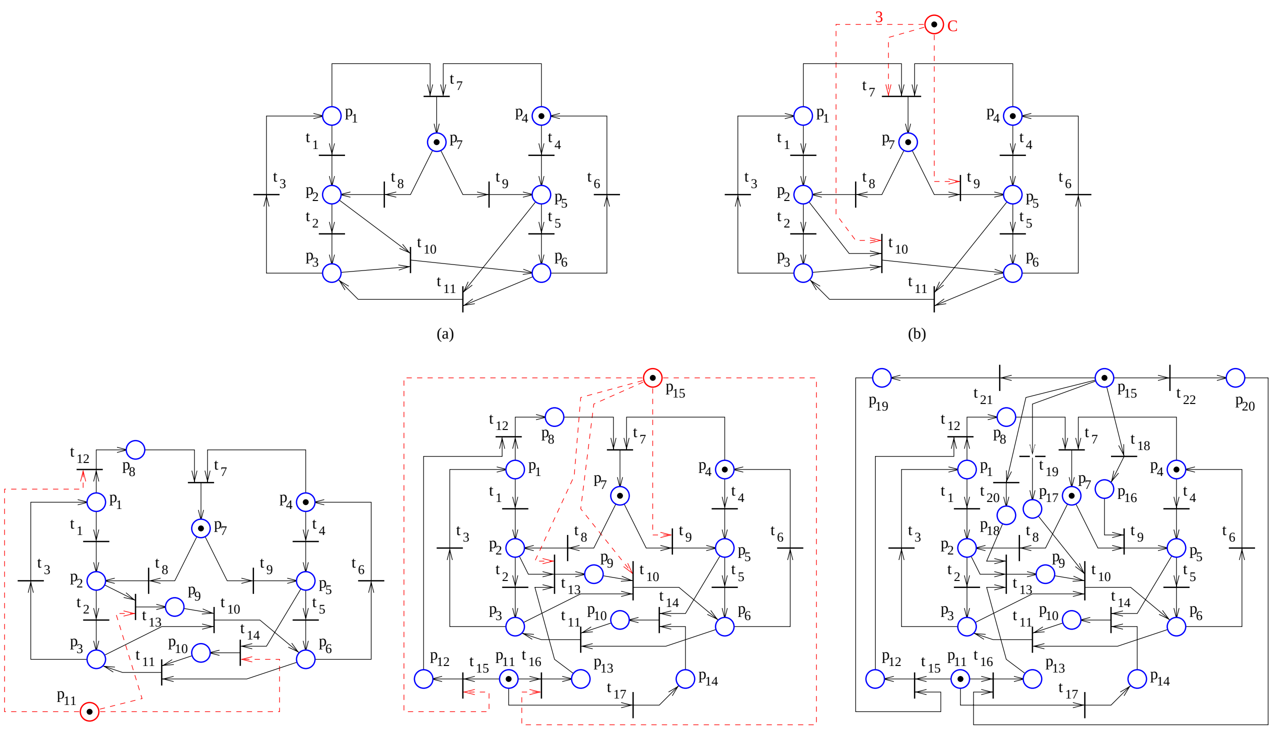

Figure 3: Example 5.2: (a) $\mathcal{N}_0$; (b) the supervised Petri net. (c) $\mathcal{N}_1$; (d) $\mathcal{N}_2$; (e) $\mathcal{N}_3$.

M.V. Iordache, P.J. Antsaklis, "T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

is thus added. The procedure terminates at the third iteration with the inequalities (3) below (for simplicity, we let $\mu_i = \mu(p_i)$):

$$\begin{aligned} \mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_7 - \mu_{12} - \mu_{13} - \mu_{14} &\geq 1 \\ 2\mu_1 + 2\mu_2 + 2\mu_3 + \mu_4 + \mu_5 + \mu_6 + 2\mu_7 + \mu_8 + \mu_{10} - \mu_{12} - \mu_{13} - \mu_{16} - \mu_{17} - \mu_{18} - \mu_{19} - \mu_{20} &\geq 2 \end{aligned}$$

After removing a redundant constraint, the supervisor of $\mathcal{N}_0$ is defined by $L = [2, 2, 2, 1, 1, 1, 2]$ and $b = 2$, and is the least restrictive T-liveness enforcing supervisor (Theorems 6.1 and 6.2).

6 Proof of the Liveness Enforcing Procedure

The proofs of the following results use the notations of the liveness procedure (section 4.5). 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 $1 \ldots i-1$ the invariant equations of the form (3) 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 $\mathcal{N}_0$. 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 $\sigma = t_1 t_2 t_3 \ldots$, we let $\sigma_{i,j}(\sigma) = \sigma_{i,j}(t_1)\sigma_{i,j}(t_2)\sigma_{i,j}(t_3)\ldots$. For instance, in Example 5.1 $\sigma_{0,2}(t_2) = t_{2,1}t_2$, in Example 5.2 $\sigma_{0,1}(t_4)$ is any of $t_8 t_9 t_4$ and $t_9 t_8 t_4$ and $\sigma_{2,3}(t_{10}) = t_{14}t_{10}$. Also, we let $\mathcal{N}_i = (P_i, T_i, F_i, W_i)$.

The T-liveness enforcement procedure may terminate at either of step B or step F. The procedure terminates at step B when no solution exists, as stated in the next result.

Proposition 6.1 *The procedure terminates at step B iff there is no initial marking for which a T-liveness enforcing supervisor exists.*

The proof of Proposition 6.1 results easily from Lemma 3.1. When the procedure terminates at step F, we say that it ***generates a T-liveness enforcing supervisor***. The supervisor is obtained by enforcing $L\mu \geq b$ on the Petri net and is defined for all initial markings μ_0 satisfying $L_0\mu_0 \geq b_0$ and $L\mu_0 \geq b$. The next result states that the supervised Petri net is indeed T-live for all such initial markings μ_0.

Theorem 6.1 *The supervisors generated by the T-liveness procedure enforce T-liveness.*

M.V. Iordache, P.J. Antsaklis, “T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

Proof: Let $\mathcal{N}_k$ be the Petri net at the last iteration. By construction, every marking μ of $\mathcal{N}_0$ which satisfies the constraints $L_0\mu \geq b_0$ and $L\mu \geq b$ 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 Theorem 3.1, where T_k^A is the set of transition of the active subnet $\mathcal{N}_k^A$. Assume that from an initial marking μ_0 of $\mathcal{N}_0$ satisfying $L_0\mu_0 \geq b_0$ and $L\mu_0 \geq b$ the supervised 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. (Note that $T \subseteq T_0 \cap T_k^A$.) We show that t dead 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, 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 places added to the net as control places. It can be shown by induction on k that firing any $t \in T_R$ always reduces the marking of some places in $P_0 \cup \mathcal{C}$. However, 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. Then, it can be shown that σ contains a subsequence $\sigma_{0,k}(t_1)$ such that the transitions of $\sigma_{0,k}(t_1)$ other than t_1 appear before t_1 in σ. Since all transitions 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. Let t_2 be the next transition of σ which is 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 $t_1t_2\ldots t_j$, and since $t_j = t$, t is not dead in $\mathcal{N}_S$, which is a contradiction. □

Theorem 6.2 *Given T and a target Petri net, if the T-liveness procedure generates a supervisor and the target net has a single T-minimal active subnet, the supervisor is least restrictive.*

Proof: Note that when μ_0 and $\mu_{0,1}$ are equivalent, $(\mathcal{N}_0, \mu_0)$ cannot be made T-live if $(\mathcal{N}_1, \mu_{0,1})$ cannot be made T-live. 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 $\sigma = \sigma_1 t_1 \sigma_2$, $\mu_{0,i} \xrightarrow{\sigma_1} \mu_1$, $\mu_{0,i+1} \xrightarrow{\sigma_{i,i+1}(\sigma_1)} \mu_1'$, 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 $(\mathcal{N}_i, \mu_1)$.

An empty active siphon implies a set T_x of dead transitions from the active subnet. Therefore the

M.V. Iordache, P.J. Antsaklis, “T-Liveness Enforcement in Petri Nets Based on Structural Net Properties," Proceedings of the 40th IEEE Conference on Decision and Control, pp. 4984-4989, Orlando, FL, December 4-7, 2001.

transitions in T_x do not appear infinitely often in σ. Let $T_{x1} = \{t \in T_1^A : \exists t_u \in \sigma_{1,i}(t) \text{ and } t_u \in T_x\}$. 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 $(\mathcal{N}_0, \mu_0)$ 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 $L_0\mu \not\geq b_0$. □

7 Final Remarks

The procedure is able to find out immediately whether a T-liveness supervisor exists (Proposition 6.1). However the construction of the supervisor may not always terminate. Let T^A be the transition set of the active subnet $\mathcal{N}_0^A$ of the target net $\mathcal{N}_0$. The proof of Theorem 6.1 guarantees not only T-liveness but also T^A-liveness ($T \subseteq T^A$). Furthermore, there is a single T^A-minimal active subnet and so, by Theorem 6.2, the supervisors are always least restrictive with regard to T^A-liveness. This tells us that the procedure cannot terminate for the problems in which the least restrictive T^A-liveness supervisor is not representable as a set of linear marking inequalities.

The modification of the liveness procedure in [8] for guaranteed termination is also applicable to the T-liveness procedure of this paper. The modification may affect the permissivity of the T-liveness supervisor and applies to bounded Petri nets.

With regard to Theorem 6.2, note that in the case of liveness enforcement ($T = T_0$), the whole net is the only T-minimal active subnet, so the generated supervisors are least restrictive.

Finally, to extend the procedure to Petri nets with uncontrollable and unobservable transitions, we use the concept of *admissible constraints* [10]. Thus, we check for each siphon whether the constraint of (1) is *inadmissible* with respect to $\mathcal{N}_0$. If this is the case, we transform (1) to an admissible constraint of the form

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

where, $\alpha_p \in \mathbb{Z}_+$ and at least two α_p are nonzero; then we enforce (5) instead of (1). Failure to find an admissible constraint means failure of the procedure to synthesize a T-liveness supervisor. While T-liveness enforcement can still be guaranteed, Theorem 6.2 is much harder to extend as its proof relies on $\alpha_p > 0\ \forall p \in S$.

References

[1] The Petri net bibliography. http://www.informatik.uni-hamburg.de/TGI/pnbib/.

[2] 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.

[3] K. Barkaoui and J. F. Pradat-Peyre. On liveness and controlled siphons in Petri nets. In *Proc. 17th ICATPN*, pages 57–72. Springer-Verlag, 1996.

[4] 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.

[5] K.X. He and M.D. Lemmon. On the transformation of maximally permissive marking-based supervisors into monitor supervisors. In *Proc. of the IEEE Conference on Decision and Control*, December 2000.

[6] M. V. Iordache and P. J. Antsaklis. Generalized conditions for liveness enforcement and deadlock prevention in Petri nets. To appear in the *Proceedings of the ICATPN 2001.*

[7] 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.

[8] M. V. Iordache, J. O. Moody, and P. J. Antsaklis. A method for the synthesis of liveness enforcing supervisors in Petri nets. To appear in the *Proceedings of the 2001 American Control Conference.*

[9] 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.

[10] J. Moody and P. Antsaklis. Petri net supervisors for des with uncontrollable and unobservable transitions. *IEEE Transactions on Automatic Control*, 45(3):462–476, March 2000.

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

[12] J. Park and S. Reveliotis. A polynomial complexity deadlock avoidance policy for sequential resource allocation systems with multiple resource aquisitions and flexible routings. In *Proceedings of the 39th Conference on Decision and Control*, pages 2663–2669, 2000.

[13] 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.

[14] 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.

[15] 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.