Marian Iordache, Po Wu, Feng Zhu, Panos Antsaklis, "Efficient Design of Petri Net Supervisors with Disjunctive Constraints," 9th IEEE International Conference on Automation Science and Engineering (CASE 2013), Madison, WI, August 17-21, 2013.

Efficient Design of Petri-Net Supervisors with Disjunctive Specifications

Marian V. Iordache, Po Wu, Feng Zhu, and Panos J. Antsaklis

***Abstract*— The supervision based on place invariants is an efficient method for the supervision of Petri nets in which each inequality constraint is implemented by one monitor place. However, this method assumes specifications that describe convex legal sets. Non-convex legal sets can be described by disjunctions of inequality constraints. Specifications consisting of disjunctions of inequality constraints are called here disjunctive specifications. Previous work has shown that under certain boundedness assumptions it is possible to implement supervisors enforcing disjunctive specifications with conventional Petri nets. However, in the worst case, the number of places of the least-restrictive supervisors was exponentially related to the size of the specification. This paper introduces an enhanced approach that generates supervisors in which the number of places is linearly related to the size of the specification. The generated supervisors are least restrictive and are implemented with conventional Petri nets.**

I. Introduction

Modern engineering systems increasingly exhibit a complex interaction of multiple subsystems that should operate together seamlessly in order to achieve the desired functionality. Supervisory control methods provide a way to design the coordination algorithms of concurrent subsystems that are represented in the discrete event paradigm. In the context of concurrency, of special interest is the Petri net (PN) representation of systems and the supervisory control methods that are based on PNs. Indeed, note that PNs were created specifically for concurrent systems and thus they tend to offer considerably smaller representations than automata. Furthermore, by using PN representations it is possible to use both PN methods and automata methods, since PNs can be converted to automata. Note that PN methods are especially interesting when they avoid the "state space explosion" problem encountered when concurrent systems are represented as automata. This paper considers the extension of a class of very efficient PN methods to a larger class of specifications involving disjunction operations.

The supervision based on place invariants provides a very efficient way of enforcing specifications consisting of inequality constraints. This method was proposed first for generalized mutual exclusion constraints [3] of the form

$$L\mu \leq b \tag{1}$$

M. V. Iordache is with the Department of Engineering, LeTourneau University, Longview, TX 75607, USA

P. Wu, F. Zhu, and P. J. Antsaklis are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

The authors gratefully acknowledge the support of the National Science Foundation (NSF CNS-0834057).

where L is an integer matrix and b an integer vector. Specifications (1) restrict the operation of a PN to the markings μ satisfying (1). They are enforced by adding places to the PN, one place for each row of L, and by connecting them to the existing transitions of the PN so as to create certain place invariants. As shown in subsequent work, the supervision based on place invariants can be easily extended to inequalities involving not only the marking, but also the firing vector and the Parikh vector [5]. Inequalities involving the marking μ and the firing vector q have the form

$$L\mu + Hq \leq b \tag{2}$$

where H is an integer matrix. Note that the firing vector q indicates the transition or transitions fired at a firing instance.

The constraints (1) have been applied to AGV coordination problems [7], batch chemical processes [10], the representation of liveness constraints [5], and others. The constraints (2) have been applied to pipe/valve networks in chemical process control [11] and railway networks in [4]. Though a wide variety of problems can be described in terms of specifications (1) or (2), there are also interesting problems involving specifications that cannot be represented by convex legal sets. Generalizing the supervision based on place invariants to non-convex legal sets is somewhat difficult if the supervision should be described in terms of conventional PNs. A solution to this problem is presented in this paper. We consider specifications consisting of logic expressions involving arbitrary conjunctions and disjunctions of predicates of the form $l\mu + hq \leq c$, where l and h are integer vectors and c is an integer. For this type of specifications, the paper presents an algorithm that produces a least restrictive PN supervisor. The algorithm applies under certain boundedness assumptions. In this paper we only deal with the fully controllable and observable PNs. We suggest dealing with partial controllability and partial observability according to the structural framework presented in [5].

Related work includes the following. The reference [8] shows how to obtain the least restrictive supervisor enforcing the union of two legal sets, based on two least restrictive supervisors, each enforcing one of the two legal sets. The reference [9] provides a method to calculate the maximal controlled invariant set for disjunctive constraints under certain assumptions on the PN structure.

This paper enhances the method of [6]. Building on the approach for disjunctive constraints of [5], the reference [6] presents a least restrictive method for the enforcement

of disjunction specifications of the form

$$l_1\mu + h_1 q \leq c_1 \vee l_2\mu + h_2 q \leq c_2 \vee \ldots l_n\mu + h_n q \leq c_n \quad (3)$$

where for all $i = 1, 2, \ldots, n$, l_i and h_i are integer vectors and c_i is an integer. The main limitation of the method of [6] is that it expects specifications that are conjunctions of expressions of the form (3). This is a problem because converting a logic expression to its conjunctive normal form may result in an exponential increase in the number of terms. For instance, denoting by S_i the predicate $l_i\mu + h_i q \leq c_i$, the conjunctive normal form of $(S_1 \wedge S_2) \vee (S_3 \wedge S_4) \vee \ldots \vee (S_{2n-1} \wedge S_{2n})$ is the conjunction of the 2^n terms of the form $(S_{i_1} \vee S_{i_2} \vee \ldots \vee S_{i_n})$, with $i_1 \in \{1, 2\}$, $i_2 \in \{3, 4\}$, and so on.

The main contribution of this paper is that it enhances the method of [6] so as to eliminate the need to convert the specification to the conjunctive normal form. The generated supervisor is a PN in which the number of places depends linearly on the number of terms of the specification. Since the supervisor consists of the parallel composition of a number of supervisor components, the number of transitions of the supervisor could be, in the worst case, exponentially related to the number of supervisor components. However, this is not an issue in implementations in which the supervisor components operate independently and are not composed into a single supervisor.

The paper is organized as follows. Notation and preliminary results are described in section II. The supervisor components are introduced in section III. The synthesis algorithm is given in section IV. An example is given in section V. Performance considerations are included in section VI. The reader is referred to [5] for an introduction to PNs and their supervision.

II. Preliminaries

A PN will be denoted by $\mathcal{N} = (P, T, D^-, D^+)$, where P is the set of places, T the set of transitions, D^- the input matrix, and D^+ the output matrix. Firing events will be represented by firing vectors q. A firing vector q is enabled at the marking μ when $D^- q \leq \mu$. A labeled PN will be denoted by $\mathcal{N} = (P, T, D^-, D^+, \rho)$, where $\rho : T \to \Sigma$ is a labeling function, associating events to transitions, and Σ is the set of events. We assume the reader familiar with the parallel composition of labeled PNs [2], [5].

Let $l_i\mu + h_i q \leq c_i$ with $l_i \in \mathbb{Z}^{1\times|P|}$, $h_i \in \mathbb{Z}^{1\times|T|}$, and $c_i \in \mathbb{Z}$ denote an inequality in terms of the marking μ and the firing vector q. A conjunction of n inequalities $l_i\mu + h_i q \leq c_i$ can be written in the form (2), where $L \in \mathbb{Z}^{n\times|P|}$, $H \in \mathbb{Z}^{n\times|T|}$, $b \in \mathbb{Z}^{n\times 1}$, $L(i,\cdot) = l_i$, $H(i,\cdot) = h_i$, and $b(i) = c_i$. The inequalities (2) are interpreted as follows. A marking μ satisfies (2) if $L\mu \leq b$. Further, a transition t may fire at μ only if its corresponding firing vector q satisfies $L\mu + Hq \leq b$ and the next reached marking μ' (that is, $\mu \xrightarrow{t} \mu'$) satisfies $L\mu' \leq b$. As shown in [5], the least restrictive supervisor enforcing (2) is obtained by connecting a number of additional places to the existing transitions of the PN according to the input and output matrices

$$D_c^+ = \max(0, -LD, H - LD) \quad (4)$$

$$D_c^- = \max(0, LD, H) \quad (5)$$

and the initial marking

$$\mu^s = b - L\mu \quad (6)$$

In the equations above the max operation is element-wise (that is, if X, Y, and Z are matrices and $Z = \max(X, Y)$, then $Z(i,j) = \max(X(i,j), Y(i,j))$ for all indices i and j). Note that a supervisor defined by (4)–(5) will enable a firing vector q when $D_c^- q \leq \mu^s$, that is, when $L\mu + D_c^- q \leq b$. Let $H_d = D_c^-$. The concurrency interpretation of (2) is as follows. A marking μ satisfies (2) if $L\mu \leq b$. Further, q may fire at μ only if it satisfies $L\mu + H_d q \leq b$ (which implies $L\mu' \leq b$ for μ' such that $\mu \xrightarrow{q} \mu'$) [5].

For a disjunction of n inequalities $l_i\mu + h_i q \leq c_i$ two possible interpretations are possible [6]. The *state-based interpretation* can be used when $h_i = 0$ for all i. In the state based interpretation, a marking μ satisfies (3) if there is at least one index i for which $l_i\mu \leq c_i$. Further, a transition t may fire at μ only if the next reached marking μ' satisfies (3). The *dynamic interpretation* of (3) is as follows. A marking μ satisfies (3) if there is at least one index i for which $l_i\mu \leq c_i$. Further, a firing vector q may be fired at μ only if there is at least one index i for which the inequality $l_i\mu + h_{d,i} q \leq c_i$ is satisfied, where $h_{d,i}$ is calculated according to (5):

$$h_{d,i} = \max(0, l_i D, h_i) \quad (7)$$

As shown in [6], the state-based interpretation and the dynamic interpretation of (3) are not equivalent.

III. Supervisory Components

The supervisor enforcing the disjunctive specification is obtained as a composition of supervisory components. This section describes the two types of supervisory components that are used. The manner in which supervisory components are composed is described by means of a labeling function $\rho : T \to \Sigma$ assigning a unique label to each transition of the plant $\mathcal{N}$.

A. Eliminating Firing Vector Terms

This section considers an enhancement of a PN $\mathcal{N}$ to a form $\mathcal{N}^*$ so that inequalities (2) in terms of $\mathcal{N}$ correspond to inequalities (1) in terms of $\mathcal{N}^*$.

Given a set of constraints $l_i\mu + h_i q \leq c_i$, let $T_s = \{t \in T : \exists i,\, l_i D(\cdot, t) \neq h_{d,i}(t)\}$, where $h_{d,i}$ is defined in (7). Let ρ be a labeling function associating a unique label to every transition of $\mathcal{N}$. Let $\mathcal{N}_g = (P_g, T_g, D_g^-, D_g^+, \rho_g)$ be a PN defined as follows.

1) For each transition $t \in T_s$ define two transitions $t', t'' \in T_g$ such that $\rho_g(t') = \rho(t)$ and t'' has a label that was not assigned to any other transition.

2) For each transition $t \in T_s$ define one place g_t such that $D_g^-(g_t, t'') = 1$ and $D_g^+(g_t, t') = 1$
3) The initial marking of each place g_t is zero.

For example, if $\mathcal{N}$ is the PN of Figure 2 and $T_s = \{t_e, t_r, t_w\}$, then Figure 4 shows $\mathcal{N}_g$ and Figure 6(a) the closed-loop PN.

Let $\mathcal{N}^*$ be the parallel composition of $\mathcal{N}$ and $\mathcal{N}_g$. Let μ^* be the marking of $\mathcal{N}^*$. Let $l_i^* \in \mathbb{Z}^{1\times|P^*|}$ be defined by

$$\forall p \in P: \; l_i^*(p) = l_i(p) \tag{8}$$
$$\forall t \in T_s: \; l_i^*(g_t) = h_{d,i}(t) - l_i D(\cdot, t). \tag{9}$$

Proposition 3.1 *[6] Given is $(\mathcal{N}^*, \mu^*)$, the parallel composition of $(\mathcal{N}, \mu)$ and $(\mathcal{N}_g, \mu_g)$. Assume $\mu_g = 0$ and that for some $t \in T$ we have that $\mu \xrightarrow{t} \mu_1$ and $\mu_g \xrightarrow{t'} \mu_{g0} \xrightarrow{t''} \mu_{g1}$. Let μ^*, μ_0^*, and μ_1^* be the closed-loop markings representing the pairs (μ, μ_g), (μ_1, μ_{g0}), and (μ_1, μ_{g1}), respectively. Let q denote the firing vector associated with the firing of t. Then $l_i^*\mu^* = l_i\mu$, $l_i^*\mu_0^* = l_i\mu + h_{d,i}q$, and $l_i^*\mu_1^* = l_i\mu_1$.*

B. The Predicate Net

This section considers expressions $l_i^*\mu^* \leq c_i$ and describes the construction of a supervisor component involving a place of marking δ_i that equals the truth value of the proposition $l_i^*\mu^* \leq c_i$. Note that $l_i^*\mu^* \leq c_i$ is viewed here as a predicate in the variable μ^*. Moreover, we denote by $[l_i^*\mu^* \leq c_i]$ the truth value of the proposition $l_i^*\mu^* \leq c_i$ for a given marking μ^*. The truth value is 0 if the constraint is not satisfied and 1 if the constraint is satisfied.

Assume that μ^* in $l_i^*\mu^* \leq c_i$ is the marking of a PN $\mathcal{N}^* = (P^*, T^*, D^{*-}, D^{*+}, \rho^*)$, where ρ^* associates a unique label to each transition. The **predicate net** of $l_i^*\mu^* \leq c_i$ is a PN $\mathcal{N}_i = (P_i, T_i, D_i^-, D_i^+, \rho_i)$ in which P_i consists of a single place d_i and T_i, D_i^-, D_i^+, and ρ_i are defined as follows.

1) For each transition t_k such that $l_i^*D^*(\cdot, t_k) \neq 0$, define two transitions f_k and x_k in T_i having the labels $\rho_i(f_k) = \rho_i(x_k) = \rho^*(t_k)$.
2) If $l_i^*D^*(\cdot, t_k) > 0$, set $D_i^-(d_i, x_k) = 1$.
3) If $l_i^*D^*(\cdot, t_k) < 0$, set $D_i^+(d_i, x_k) = 1$.
4) Denoting by μ_0^* the initial marking of $\mathcal{N}^*$, the initial marking of d_i is 1 if $l_i^*\mu_0^* \leq c_i$ and 0 otherwise.

Let δ_i denote the marking of the place d_i. Assume that $l_i^*\mu^*$ has for all reachable markings a finite lower bound m_i and a finite upper bound M_i. Then, in order to ensure that $\delta_i = [l_i^*\mu^* \leq c_i]$, the following constraints have to be enforced on the parallel composition of $\mathcal{N}^*$ and the predicate net $\mathcal{N}_i$:

$$l_i^*\mu^* + (M_i - c_i)\delta_i \leq M_i, \tag{10}$$
$$l_i^*\mu^* + (c_i + 1 - m_i)\delta_i \geq c_i + 1. \tag{11}$$

Due to the constraints (10)–(11), $\mathcal{N}_i$ will fire f_k when $\mathcal{N}^*$ fires t_k without changing the truth value of the proposition $l_i^*\mu^* \leq c_i$. Moreover, $\mathcal{N}_i$ will fire x_k when $\mathcal{N}^*$ fires t_k and the truth value of $l_i^*\mu^* \leq c_i$ is changed.

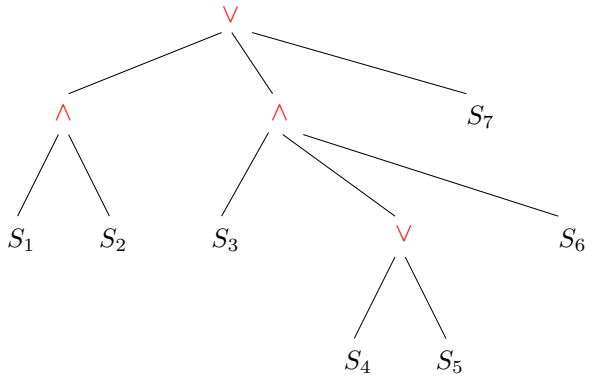

Fig. 1. Tree representing the expression $E = (S_1 \wedge S_2) \vee (S_3 \wedge (S_4 \vee S_5) \wedge S_6) \vee S_7$.

As an example, the PN of transitions f_r^2, x_r^2, $f_r''^2$, and $x_r''^2$ in Figure 6(b) shows the predicate net of $\mu_{g_r}^* \leq 0$ with respect to the PN of Figure 6(a).

IV. Enforcing Disjunctive Specifications

This section presents the algorithm for the enforcement of disjunctive specifications. Let S_i denote a predicate of the form $l_i\mu \leq c_i$ or $l_i\mu + h_iq \leq c_i$. Consider a specification described by a logic expressions E consisting of arbitrary conjunctions and disjunctions of predicates S_i. Thus, E is a compound predicate consisting of predicates S_i connected by $\vee$ operators (logic OR), $\wedge$ operators (logic AND), and parentheses that indicate precedence. For example, a possible expression could be $E = (S_1 \vee S_2) \wedge (S_3 \vee (S_4 \wedge S_5) \vee S_6) \wedge S_7$. Note that any such expression can be represented by a tree (Figure 1) in which each node represents an operation ($\vee$ or $\wedge$) and each leaf a predicate.

The following algorithm shows how to obtain the closed-loop PN based on a plant PN and a disjunctive specification. The goal of the supervision is to ensure that for all reachable states of the system, a truth value of 1 is obtained when substituting the marking and firing vector of the plant in the logic expression describing the specification.

Without loss of generality, it is assumed that in the tree representation of the specification the logic operations alternate on any path from the root to a leaf. That is, the predecessor of a $\vee$ node is a $\wedge$ node and vice-versa.

1) Consider the tree representation of the specification and all leaves connected to the root by a path that includes a $\vee$ node, where the root node itself may be the $\vee$ node. Let $\mathcal{S}$ denote the set of predicates S_i associated with these leaves.
2) In the case of the dynamic interpretation, the set T_s of section III-A will be defined with respect to all constraints of $\mathcal{S}$: $T_s = \{t \in T: \exists S_i \in \mathcal{S}, l_iD(\cdot, t) \neq h_{d,i}(t)\}$.
3) In the case of the dynamic interpretation, let $\mathcal{N}^*$ be the PN obtained as in section III-A. Further, for each predicate $S_i \in \mathcal{S}$, let S_i^* be the predicate $l_i^*\mu^* \leq c_i$ with l^* defined as in (8)–(9). Moreover, for each predicate $S_i \notin \mathcal{S}$, let $S_i^* = S_i$.

4) In the case of the state-based interpretation, let $\mathcal{N}^* = \mathcal{N}$ and $S_i^* = S_i$.
5) Let $\mathcal{S}^*$ be the set of predicates S_i^* such that $S_i \in \mathcal{S}$.
6) Let $\mathcal{L}$ be a set of constraints initialized to the inequalities of the predicates $S_i^* \notin \mathcal{S}^*$.
7) Let $\mathcal{N}^c = \mathcal{N}^*$.
8) Associate recursively a predicate with each node of the tree, starting from the bottom and moving up towards the root. The predicate is obtained as follows.
   a) Let $S_{i1}^*, \ldots, S_{ik}^*$ be the predicates of the successor nodes or leaves.
   b) If the node is the root and the root is a $\wedge$ node, then add to $\mathcal{L}$ the inequalities of $S_{i1}^*, \ldots, S_{ik}^*$ and go to step 9.
   c) Let $\mathcal{N}_{i1}^*, \ldots, \mathcal{N}_{ik}^*$ be the predicate nets (section III-B) of $S_{i1}^*, \ldots, S_{ik}^*$, where the $\mathcal{N}_{i1}^*, \ldots, \mathcal{N}_{ik}^*$ are defined with respect to $\mathcal{N}^c$.
   d) Update $\mathcal{N}^c$ to equal its parallel composition with $\mathcal{N}_{i1}^*, \ldots, \mathcal{N}_{ik}^*$.
   e) Add to $\mathcal{L}$ the inequalities (10)–(11) associated with $S_{i1}^*, \ldots, S_{ik}^*$.
   f) Let $d_{i1}, \ldots, d_{ik}$ be the places of $\mathcal{N}_{i1}^*, \ldots, \mathcal{N}_{ik}^*$.
   g) If the node is a $\wedge$ node, the predicate of the node will be $\sum_{j=1}^{k} \mu^*(d_{ij}) \geq k$. Note that the upper and lower bounds of $\sum_{j=1}^{k} \mu^*(d_{ij})$ are k and 0, respectively.
   h) If the node is a $\vee$ node, the predicate of the node will be $\sum_{j=1}^{k} \mu^*(d_{ij}) \geq 1$. Note that the upper and lower bounds of $\sum_{j=1}^{k} \mu^*(d_{ij})$ are k and 0, respectively.
   i) If the node is the root and the root is a $\vee$ node, then add to $\mathcal{L}$ the inequality $\sum_{j=1}^{k} \mu^*(d_{ij}) \geq 1$ and go to step 9.
9) Let $L\mu^c + Hq \leq b$ denote the constraints of $\mathcal{L}$.
10) Let $\mathcal{N}^t$ denote the closed-loop of $\mathcal{N}^c$ and the supervisor (4)–(5) enforcing $L\mu^c + Hq \leq b$ on $\mathcal{N}^c$.
11) The initial marking of $\mathcal{N}^t$ is determined in terms of μ_0, the initial marking of $\mathcal{N}$, as follows.
   a) $\mu_0^t(p) = \mu_0(p)\ \forall p \in P$.
   b) $\mu_0^t(g_t) = 0\ \forall t \in T_s$.
   c) For every predicate net $\mathcal{N}_i^*$, the initial marking of the place d_i equals the truth value of S_i^* at the initial marking.
   d) The initial marking of the places added at step 10 is calculated according to (6).

In the algorithm above note that $(\mathcal{N}^t, \mu_0^t)$ represents the closed-loop PN of the plant $(\mathcal{N}, \mu_0)$ and the supervisor enforcing the disjunctive specification.

V. Example

Consider the reader/writer example of [6]. In this computer science example a number of reader processes (RPs) and writer processes (WPs) may access a shared region of memory. A process is said to be in the critical section (CS) when accessing the shared region of memory. Any number of RPs may be in the CS at the same time. However, when a WP is in the CS, no other process may be in the CS. Moreover, the WPs have higher precedence than the RPs. This means that no RP may enter the CS when a WP waits to enter the CS.

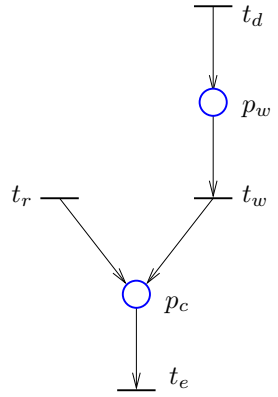

Fig. 2. Model of the reader/writer system.

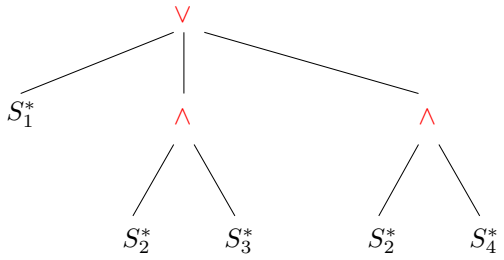

Fig. 3. Tree representing the expression $S_1^* \vee (S_2^* \wedge S_3^*) \vee (S_2^* \wedge S_4^*)$.

A PN model is shown in Figure 2. The marking of p_c represents the number of processes in the CS. The transition t_e is fired when a process exits the CS, t_r when a RP enters the CS, and t_w when a WP enters the CS. The marking of p_w represents the number of WPs waiting to enter the CS. A WP enters the p_w state by firing t_d.

The specification is expressed by the predicate $S_1 \vee (S_2 \wedge S_3) \vee (S_2 \wedge S_4)$, where $S_1, \ldots, S_4$ denote $\mu_w \leq 0$, $q_r \leq 0$, $q_w \leq 0$, and $\mu_c \leq 0$, respectively. (For a place p_i and a transition t_j, μ_i stands for $\mu(p_i)$ and q_j for $q(t_j)$.)

In this example, $S = \{S_1, \ldots, S_4\}$ and $T_s = \{t_e, t_r, t_w\}$. The net $\mathcal{N}^*$ is shown in Figure 6(a). It is the composition of $\mathcal{N}$ (Figure 2) and $\mathcal{N}_g$ (Figure 4). $S_1^*, \ldots, S_4^*$ denote $\mu_w^* + \mu_{g_w}^* \leq 0$, $\mu_{g_r}^* \leq 0$, $\mu_{g_w}^* \leq 0$, and $\mu_c^* + \mu_{g_e}^* \leq 0$, respectively. The tree representing the specification is shown in Figure 3.

In the first iteration of step 8, the predicate nets $\mathcal{N}_2^*$, $\mathcal{N}_3^*$, and $\mathcal{N}_4^*$ are built (Figure 6(b)). These correspond to the predicates S_2^*, S_3^*, and S_4^*. The PN $\mathcal{N}^c$ is updated to equal the parallel composition $\mathcal{N}^c \| \mathcal{N}_2^* \| \mathcal{N}_3^* \| \mathcal{N}_4^*$. Further, the expressions $S_2^* \wedge S_3^*$ and $S_2^* \wedge S_4^*$ are replaced by the predicates S_5^* and S_6^*, where S_5^* and S_6^* denote $-\mu^*(d_2) - \mu^*(d_3) \leq -2$ and $-\mu^*(d_2) - \mu^*(d_4) \leq -2$, respectively. In view of step 8g, the upper and lower bounds

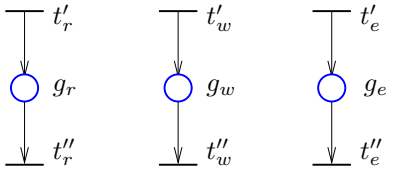

Fig. 4. The $\mathcal{N}_g$ supervisory component.

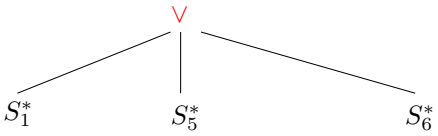

Fig. 5. The expression shown in Figure 3 is reduced to $S_1^* \vee S_5^* \vee S_6^*$ after the first iteration.

of $-\mu^*(d_2) - \mu^*(d_3)$ and $-\mu^*(d_2) - \mu^*(d_4)$ are $M_5 = M_6 = 0$ and $m_5 = m_6 = -2$. At the end of the first iteration, $\mathcal{L}$ contains the inequalities (10)–(11) associated with S_2^*, S_3^*, and S_4^*. The bounds in (10)–(11) are as follows. Assuming that multiple transitions are not fired at the same time and that the supervisor fires t''_r, t''_e, and t''_w as soon as enabled, the maximum value of either of $\mu^*_{g_r}$, $\mu^*_{g_e}$, and $\mu^*_{g_w}$ is 1. Then, $M_2 \geq 1$ and $M_3 \geq 1$. If no more than ten RPs can ever be in the CS at the same time, $M_4 \geq 10$. Using $m_2 = m_3 = m_4 = 0$, $M_2 = M_3 = 1$, and $M_4 = 10$, the inequalities (10)–(11) are:

$$\mu^*_{g_r} + \mu^*(d_2) \leq 1 \quad (12)$$
$$\mu^*_{g_r} + \mu^*(d_2) \geq 1 \quad (13)$$
$$\mu^*_{g_w} + \mu^*(d_3) \leq 1 \quad (14)$$
$$\mu^*_{g_w} + \mu^*(d_3) \geq 1 \quad (15)$$
$$\mu^*_c + \mu^*_{g_e} + 10\mu^*(d_4) \leq 10 \quad (16)$$
$$\mu^*_c + \mu^*_{g_e} + \mu^*(d_4) \geq 1 \quad (17)$$

At the end of the first iteration of step 8 the specification tree of Figure 3 is reduced to the tree shown in Figure 5.

In the second iteration the predicate nets $\mathcal{N}_1^*$, $\mathcal{N}_5^*$, and $\mathcal{N}_6^*$ are built (Figure 6(c)). These correspond to the predicates S_1^*, S_5^*, and S_6^*. The PN $\mathcal{N}^c$ is updated to equal the parallel composition $\mathcal{N}^c \| \mathcal{N}_1^* \| \mathcal{N}_5^* \| \mathcal{N}_6^*$. The predicate of the root node is

$$\mu^*(d_1) + \mu^*(d_5) + \mu^*(d_6) \geq 1 \quad (18)$$

The inequality (18) is added to $\mathcal{L}$ in the step 8i. Additionally, the inequalities (10)–(11) associated with S_1^*, S_5^*, and S_6^* are also added to $\mathcal{L}$. Assuming no more than 3 WPs, we can take $m_1 = 0$ and $M_1 = 3$. Since $m_5 = -2$, $M_5 = 0$, $m_6 = -2$, and $M_6 = 0$, the inequalities (10)–(11) associated with S_1^*, S_5^*, and S_6^* are as follows:

$$\mu^*_w + \mu^*_{g_w} + 3\mu^*(d_1) \leq 3 \quad (19)$$
$$\mu^*_w + \mu^*_{g_w} + \mu^*(d_1) \geq 1 \quad (20)$$
$$-\mu^*(d_2) - \mu^*(d_3) + 2\mu^*(d_5) \leq 0 \quad (21)$$
$$-\mu^*(d_2) - \mu^*(d_3) + \mu^*(d_5) \geq -1 \quad (22)$$
$$-\mu^*(d_2) - \mu^*(d_4) + 2\mu^*(d_6) \leq 0 \quad (23)$$
$$-\mu^*(d_2) - \mu^*(d_4) + \mu^*(d_6) \geq -1 \quad (24)$$

The second iteration is the final iteration of this example. At the end of this iteration $\mathcal{L}$ contains the inequalities (12)–(24). After the last step of the algorithm the supervisor will involve the places $g_r, g_e, g_w, d_1, \ldots, d_6$, and the 13 monitor places enforcing the 13 constraints (12)–(24).

VI. Performance

Section IV has presented an algorithm for enforcing constraints described by arbitrary combinations of disjunctions and conjunctions of linear inequalities. The algorithm assumes that for all predicates $l_i\mu \leq c_i$, the term $l_i\mu$ has finite upper and lower bounds and that such bounds are known. This assumptions was made in the context of inequalities (10)–(11). By construction, the algorithm guarantees that the specification is enforced. However, two assumptions are needed in order to guarantee least restrictive supervision:

- no concurrency (two or more plant transitions may not fire at the same time);
- immediate firing of supervisor transitions (the supervisor transitions that are not synchronized with the plant are fired as soon as they are enabled).

The supervisor transitions that are not synchronized with the plant belong to the supervisor component described in section III-A. Specifically, using the notation of section III-A, the output transitions t'' of the places g_t are not synchronized with any plant transitions. Since equations (8)–(9) ensure that by firing t'' the term $l^*\mu^*$ is not increased, the supervisor will never be prevented from emptying the places g_t when they get marked.

Theorem 6.1 *Consider the closed-loop PN $(\mathcal{N}^t, \mu_0^t)$ constructed with the algorithm of section IV. If the specification is satisfied at the initial marking, it is satisfied also for all reachable markings. Moreover, assuming that all supervisor transitions are fired as soon as enabled and that the plant does not fire multiple transitions at the same time, the supervision is least restrictive.*

The proof of the theorem is similar to the proof of Theorem 4.1 of [6].

The size of the supervisor is as follows. Let N be the total number of nodes and leaves of the tree representing the specification. Note that the total number of inequality constraints enforced by the algorithm is less than $2N$. Indeed, for each node or leaf at most two inequalities are enforced (the inequalities (10)–(11)) and the root node has at most one inequality. Each of the $2N$ inequalities contributes one monitor place. Additionally, each predicate net contributes one place and there is one predicate net for each pair (10)–(11). Moreover, since the number of places g_t generated by the algorithm of section III-A is upper bounded by the number of transitions of the plant, we can conclude that the number of places of the supervisor will not exceed $3N + |T|$. The upper bound is only $3N$ in the

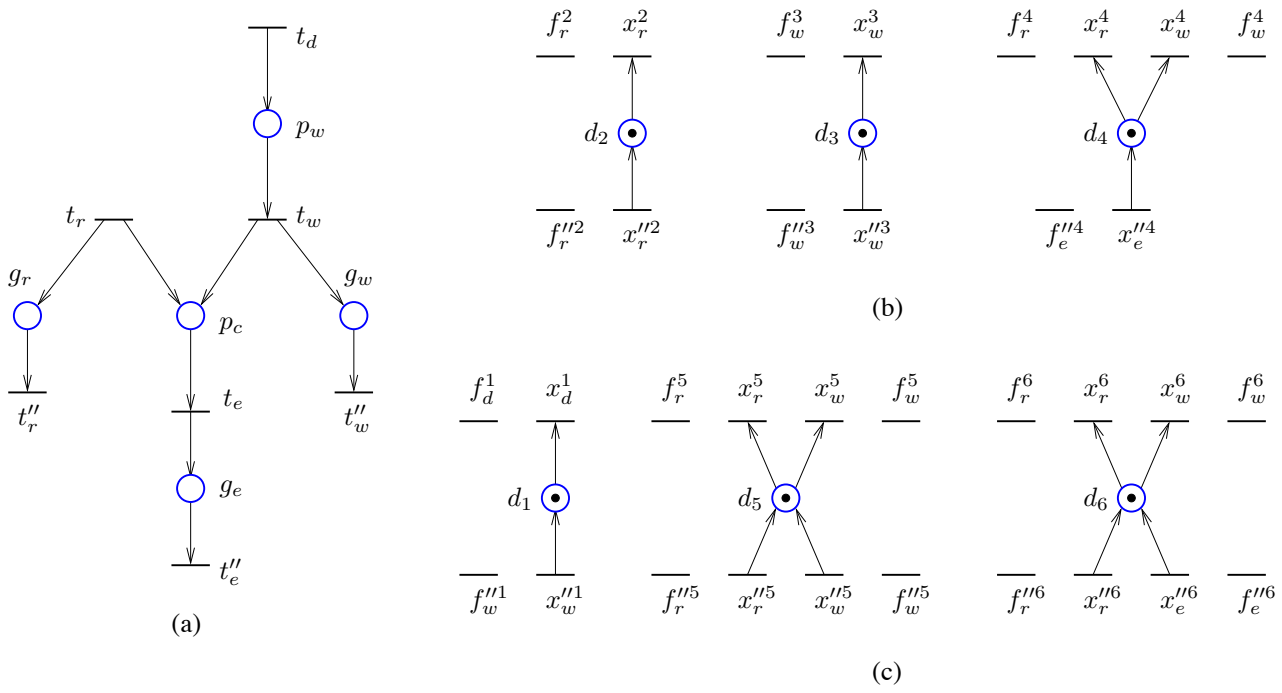

Fig. 6. (a) $\mathcal{N}^c$ before the first iteration; (b) from left to right, $\mathcal{N}^*_2$, $\mathcal{N}^*_3$, and $\mathcal{N}^*_4$; (c) from left to right, $\mathcal{N}^*_1$, $\mathcal{N}^*_5$, and $\mathcal{N}^*_6$.

case of the state based interpretation of the specification, since the algorithm of section III-A is not used in this case.

Now, the parallel composition of PNs can increase dramatically the number of transitions. Assume the notation of section IV. By firing a transition t of $\mathcal{N}^*$, the terms $l^*\mu^*$ of the predicates $l^*\mu^* \leq c$ may be changed. If for n predicate nets the terms $l^*\mu^*$ are affected by t, then t will appear in the form of 2^n transitions in the closed-loop PN. This is due to the fact that t may affect each of the n predicate nets in two ways: it will either change the truth value of the proposition $l^*\mu^* \leq c$, or it will leave it unchanged. Since there are two possible outcomes for each predicate net, there will be 2^n possibilities for the n predicate nets and thus 2^n transitions, one for each possibility. This problem is due to the parallel composition of the supervisor components. A practical implementation of the supervisor would not require a parallel composition of the components. Rather, the components could run independently and synchronize their transitions online. However, the parallel composition of the components could be required if further synthesis or verification methods are to be applied to the closed-loop PN.

Finally, note that a software implementation of the approach of this paper may be downloaded from the supervisory control folder of the ACTS software [1].

VII. Conclusion

The paper presents an efficient method for enforcing specifications described by non-convex legal sets. The closed-loop is represented by conventional PNs. The structure of the closed-loop PN is independent of the initial marking of the plant. Naturally, the initial marking of the supervisor is calculated based on the initial marking of the plant. The supervision method is least restrictive if the plant does not attempt to fire multiple transitions at the same time. The method assumes that finite upper and lower bounds are known for all linear marking expressions appearing in the inequalities of the specification.

References

[1] A Concurrency Tool Suite. http://www.letu.edu/people/marianiordache/acts.

[2] A. Giua and F. DiCesare. Supervisory design using Petri nets. In *Proc. 30th IEEE Conf. Decision Contr.*, pp. 92–97, 1991.

[3] A. Giua, F. DiCesare, and M. Silva. Generalized mutual exclusion constraints on nets with uncontrollable transitions. In *Proc. IEEE Internat. Conf. Syst., Man, Cybern.*, pp. 974–979, 1992.

[4] A. Giua and C. Seatzu. Supervisory control of railway networks with Petri nets. In *Proc. 40th IEEE Conf. Decision Contr.*, pp. 5004–5009, 2001.

[5] M. V. Iordache and P. J. Antsaklis. *Supervisory Control of Concurrent Systems: A Petri Net Structural Approach*. Birkhäuser, 2006.

[6] M. V. Iordache and P. J. Antsaklis. Petri net supervisors for disjunctive constraints. In *Proc. Amer. Contr. Conf.*, pp. 4951–4956, 2007.

[7] B.H. Krogh and L.E. Holloway. Synthesis of feedback control logic for manufacturing systems. *Automatica*, 27(4):641–651, 1991.

[8] G. Stremersch and R. K. Boel. Decomposition of the supervisory control problem for Petri nets under preservation of maximal permissiveness. *IEEE Trans. Automat. Contr.*, 46(9):1490–1496, 2001.

[9] G. Stremersch and R. K. Boel. Structuring acyclic Petri nets for reachability analysis and control. *Discrete Event Dynamic Systems*, 12(1):7–41, 2002.

[10] M. Tittus and B. Egardt. Hierarchical supervisory control for batch processes. *IEEE Trans. Contr. Syst. Technol.*, 7(5):542–554, 1999.

[11] E. Yamalidou and J. Kantor. Modeling and optimal control of discrete-event chemical processes using Petri nets. *Computers and Chemical Engineering*, 15(7):503–519, 1991.