On the Notion of Uncontrollable Marking in Supervisory Control of Petri Nets

Bruno Lacerda and Pedro U. Lima

***Abstract*—We show that the notion of uncontrollable marking commonly used in the literature on supervisory control theory of Petri nets is not sound, by means of a counter-example. We also show how the definition can be corrected and provide an adaptation of a decidability proof for the problem of checking controllability for specifications expressed as deterministic Petri net languages.**

***Index Terms*—Petri Nets, Discrete Event Systems, Supervisory Control, Controllability**

I. Introduction

SUPERVISORY control theory, as introduced in [1], is an important field of study in automatic control and discrete event systems (DES). The basic idea of this approach for control of DES is to restrict the behaviour of a system to an acceptable behaviour, through the use of a supervisor in a feedback loop. An important notion in this scope is the controllability of the supervisor. A supervisor is considered controllable if it does not disable uncontrollable events that would be active in the open-loop behaviour of the system. For the case of finite state automata supervisors (i.e., supervisors corresponding to regular language specifications), it has been proved that checking for controllability is decidable [1].

In this work, we deal with the decidability problem of checking controllability for Petri net (PN) supervisors. This problem has been tackled as early as in [2], where the notion of uncontrollable marking was first defined. With this notion, it was proven that checking for controllability is decidable for deterministic PNs, by reducing the problem to checking the existence of reachable uncontrollable markings – if uncontrollable markings are reachable, then the supervisor is uncontrollable. However, this definition for uncontrollable marking, which has been used in the literature since then, is not correct. In fact, there exist deterministic PNs with reachable uncontrollable markings, according to the original definition, that do not represent an uncontrollable supervisor. In this paper, we show an example of such a PN, provide an adjustment to the definition of uncontrollable marking that makes it correct, and prove the decidability of checking for controllability of a deterministic PN supervisor using the adjusted definition.

II. Petri Nets and Supervisory Control

In this section, we present a brief overview on Petri nets and supervisory control with language specifications. We refer the reader to [3] for a detailed overview on this topic.

A. Petri Nets

Definition 1 (Petri Net)**.** *A Petri net (PN) is a tuple* $G = \langle P, T, W^-, W^+, M_0, E, \ell \rangle$ *where:*

- *P is a finite, not empty, set of places;*
- *T is a finite, not empty, set of transitions;*
- *$W^- \in \mathbb{N}^{|P| \times |T|}$ is the pre-incidence matrix;*
- *$W^+ \in \mathbb{N}^{|P| \times |T|}$ is the post-incidence matrix;*
- *$M_0 \in \mathbb{N}^{|P|}$ is the initial marking.*
- *E is the finite set of events;*
- *$\ell : T \to E$ is the labelling function, that assigns to each transition an event from E.*

The pre-incidence matrix represents arc weights between places and transitions while the post-incidence matrix represents arc weights between transitions and places. The initial marking M_0 is a vector of size $|P|$ that represents the initial state of the system, with $M_0(p) = q$ meaning that there are q tokens in place p in the initial state.

We will use the places and transitions themselves as the indices of the matrices and vectors, e.g., given $p \in P$ and $t \in T$, we use $W^-(p,t)$ to represent the entry W^-_{ij} that corresponds to the arc weight from p to t. The dynamics of a PN are defined by the firing rule, which determines the flow of tokens between places, thus specifying how the initial marking can evolve.

Definition 2 (Firing Rule)**.** *Let G be a PN, $t \in T$ and $M \in \mathbb{N}^{|P|}$. Transition t is said to be active in M if for all $p \in P$, $W^-(p,t) \leq M(p)$. A transition t active in a marking M can fire, resulting in the marking $M' \in \mathbb{N}^{|P|}$, where, for each $p \in P$, $M'(p) = M(p) - W^-(p,t) + W^+(p,t)$. This is denoted $M \xrightarrow{t} M'$.*

Using the firing rule, one can define firing sequences and the set of reachable markings of a given PN.

Definition 3 (Firing Sequence)**.** *Let G be a PN. A firing sequence from a given marking M is a sequence of transitions $\tau = t_1 t_2 ... t_n \in T^*$ such that there exists markings $M_1, ..., M_n$ such that:*

$$M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_2} ... \xrightarrow{t_n} M_n \qquad (1)$$

We also write $M \xrightarrow{\tau} M_n$ to represent that the firing of the sequence τ leads the PN marking from M to M_n.

Definition 4 (Reachable Markings)**.** *The set of all reachable markings in a PN G is denoted as:*

$$\mathscr{R}(G) = \{ M \in \mathbb{N}^{|P|} \mid \text{ exists } \tau \in T^* \text{ such that } M_0 \xrightarrow{\tau} M \} \qquad (2)$$

In addition to reachable markings, we will also be interested in the sequences of events that can be generated by its different runs. This set of sequences is called the language generated by the PN.

Bruno Lacerda is with the School of Computer Science, University of Birmingham, Birmingham, UK. e-mail: b.lacerda@cs.bham.ac.uk.

Pedro Lima is with the Institute for Systems and Robotics, Instituto Superior Técnico, Lisboa, Portugal. e-mail: pal@isr.ist.utl.pt.

Definition 5 (Language Generated by a PN)**.** *Let G be a PN. The language generated by G is defined as:*

$$\begin{aligned}\mathscr{L}(G) = \{\ell(t_1)\ell(t_2)...\ell(t_n) \in E^* \mid \\ t_1t_2...t_n \in T^* \textit{ is a firing sequence from } M_0\}\end{aligned} \tag{3}$$

We require the PNs is this work to be deterministic, in the sense that a sequence of labels uniquely defines the sequence of visited markings.

Definition 6 (Deterministic PN)**.** *A PN G is deterministic if for all $t,t' \in T$ and $M \in \mathscr{R}(G)$:*

$$\textit{If } M \xrightarrow{t} M',\ M \xrightarrow{t'} M'' \textit{ and } M' \neq M'' \textit{ then } \ell(t) \neq \ell(t') \tag{4}$$

As will be seen later, a supervisor is formally a function over sequences of events. It is assumed that the supervisor cannot directly observe neither the markings visited, nor the fired transitions. Thus, we require determinism of the PN model of the system, so that the supervisor can keep track of the current marking of the PN model of the system during execution while only observing the sequence of events generated by the system. We note that, in the cases where the supervisor can directly observe either the markings visited by the PN model of the system, or the fired transitions, the assumption of determinism of the PN model of the system can be removed.

B. Supervisory Control

The purpose of supervisory control (SC), as introduced in [1], is, given a system model – in our case a deterministic PN G – of the open-loop uncontrolled behaviour of a system, to restrict its behaviour to an acceptable language $L_a \subseteq \mathscr{L}(G)$ – in our case given by a deterministic PN H. We start by partitioning the event set E in two disjoint subsets $E = E_c \cup E_{uc}$. E_c is the set of controllable events, i.e., the events that can be prevented from happening by the supervisor and E_{uc} is the set of uncontrollable events, i.e., the events that cannot be prevented from happening. This partition is due to the fact that, in general, there are events that make a system change its state that are not of the "responsibility" of the system itself (e.g., failures in the execution of the system). The set of uncontrollable events induces a set of uncontrollable transitions for G.

Definition 7 (Uncontrollable Transitions)**.** *Let G be a PN with event set $E = E_c \cup E_{uc}$. The set of uncontrollable transitions of G is defined as:*

$$T_{uc} = \{t \in T \mid \ell(t) \in E_{uc}\} \tag{5}$$

We now formally define the notion of supervisor and of language generated by a supervised PN.

Definition 8 (Supervisor)**.** *Let G be a PN. A supervisor for G is a function $S : \mathscr{L}(G) \to 2^E$ that, given $s \in \mathscr{L}(G)$, outputs the set of enabled events for G, i.e., the set of events that G can execute next.*

Definition 9 (Language Generated by a Supervised PN)**.** *Let G be a PN and S be a supervisor. The language generated by G when controlled by S is given by:*

$$\mathscr{L}(S/G) = \{e_1e_2...e_n \in \mathscr{L}(G) \mid e_{i+1} \in S(e_1...e_i) \textit{ for all } i \in \mathbb{N}\} \tag{6}$$

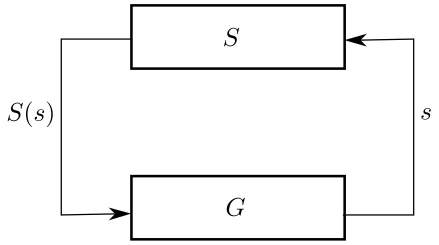

Fig. 1. The feedback loop of SC, where $s \in E^*$.

In words, the supervisor controls the system by, after the firing of an event by G, "reading" the string s executed by G so far, and outputting a set of enabled events $S(s)$. When executing the next event, G can only execute an event which is active in its current state and which is enabled by S, i.e., which is in $S(s)$. This feedback loop is depicted in Figure 1.

To formally define controllability of a supervisor, we need to introduce some notation.

Definition 10 (Resulting Marking)**.** *Let G be a deterministic PN and $s = e_1...e_n \in \mathscr{L}(G)$. We define M_s as the marking reached after executing a firing sequence $\tau = t_1...t_n \in T^*$ such that $\ell(t_i) = e_i$ for all $i \in \{1,...,n\}$ from M_0.*

Note that M_s is well-defined because (i) given that $s \in \mathscr{L}(G)$, there is always at least one firing sequence τ satisfying $\ell(t_i) = e_i$ for all $i \in \{1,...,n\}$, and (ii) given that G is deterministic, the resulting marking after executing any firing sequence satisfying $\ell(t_i) = e_i$ for all $i \in \{1,...,n\}$ is unique. Finally, we also define the active event function.

Definition 11 (Active Event Function)**.** *Let $G = \langle P,T,W^-,W^+,M_0,E,\ell\rangle$. We define the function $\Gamma_G : \mathscr{R}(G) \to 2^E$ as:*

$$\begin{aligned}\Gamma_G(M) = \{e \in E \mid \textit{exists } t \in T \textit{ such that } \ell(t) = e \\ \textit{and } t \textit{ is active in } M\}\end{aligned} \tag{7}$$

Definition 12 (Controllable Supervisor)**.** *Let G be a PN, with uncontrollable events $E_{uc} \subseteq E$, and $S : \mathscr{L}(G) \to 2^E$. S is a controllable supervisor for G if, for all $s \in \mathscr{L}(G)$:*

$$E_{uc} \cap \Gamma_G(M_s) \subseteq S(s) \tag{8}$$

Intuitively, a supervisor is controllable if it never disables uncontrollable events that would be active in the open-loop behaviour of the system.

For analysis and implementation purposes, it is important to represent the supervisor in a convenient way. This representation is referred to as a *realization* of the supervisor. The typical approach is to also represent the supervisor as a PN. Given the PN model G of the open-loop behaviour of the system, this is done by performing the following steps:

1) Build a deterministic PN H, that represents the language specification to be enforced;
2) Build the candidate supervisor realization R, given by the parallel composition $R = G \parallel H$. By definition of parallel composition, the resulting PN represents the running in parallel of G and H, where common events must

occur in a synchronized fashion. Thus, the behaviour of G is restricted to the one specified by H;

3) Check if R realizes a controllable supervisor. If so, it can be used to enforce the specification.

In addition to the determinism of G, we also require the PN representation of the specification language H to be deterministic, so that the result of the supervisor function is unequivocally defined for a given input.

Remark 1. *In general one also wants to guarantee that the supervisor is non-blocking but, for the sake of brevity, in this paper we focus on controllability checking. Also, we will not deal with the problem of trimming an uncontrollable supervisor so that it becomes controllable which, in the case of PNs, is a very challenging – and sometimes impossible – problem. We refer the interested reader to [3].*

.

Before we introduce the notion of parallel composition for PNs, we need to define the set of shared transitions. In the following, let $G_1 = \langle P_1, T_1, W_1^-, W_1^+, M_{0,1}, E_1, \ell_1 \rangle$ and $G_2 = \langle P_2, T_2, W_2^-, W_2^+, M_{0,2}, E_2, \ell_2 \rangle$ be two PNs.

Definition 13 (Shared Transitions)**.** *The set of shared transitions of G_1 and G_2 is given by:*

$$T_{E_1 \cap E_2} = \{(t_1, t_2) \in T_1 \times T_2 \mid \ell_{G_1}(t_1) = \ell_{G_2}(t_2)\} \tag{9}$$

The set of shared transitions is simply the set of pairs of transitions in G_1 and G_2 that share the same event label. These transitions need to be synchronized in the composition.

Definition 14 (Parallel Composition)**.** *The parallel composition of G_1 and G_2 is the PN $G = G_1 \parallel G_2 = \langle P, T, W^-, W^+, M_0, E_G \cup E_H, \ell \rangle$, where:*

- $P = P_1 \cup P_2$
- *T is the union of the non-shared transitions of G_1 and G_2 with the pairs representing the shared transitions, i.e., $T = T_{E_1 \setminus E_2} \cup T_{E_2 \setminus E_1} \cup T_{E_1 \cap E_2}$, where:*

$$T_{E_1 \setminus E_2} = \{t \in T_1 \mid \ell_1(t) \in E_1 \setminus E_2\} \tag{10}$$

$$T_{E_2 \setminus E_1} = \{t \in T_2 \mid \ell_2(t) \in E_2 \setminus E_1\} \tag{11}$$

- *$W^- \in \mathbb{N}^{|P| \times |T|}$ maintains the same arc weights as W_1^- and W_2^-, taking into account that shared transitions of G_1 and G_2 are now merged into a single transition:*

$$W^-(p,t) = \begin{cases} W_1^-(p,t) & \text{if } t \in T_{E_1 \setminus E_2} \text{ and } p \in P_1 \\ W_2^-(p,t) & \text{if } t \in T_{E_2 \setminus E_1} \text{ and } p \in P_2 \\ W_1^-(p,t_1) & \text{if } t = (t_1,t_2) \in T_{E_1 \cap E_2} \\ & \text{and } p \in P_1 \\ W_2^-(p,t_2) & \text{if } t = (t_1,t_2) \in T_{E_1 \cap E_2} \\ & \text{and } p \in P_2 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

- *$W^+ \in \mathbb{N}^{|P| \times |T|}$ is defined analogously to W^-:*

$$W^+(p,t) = \begin{cases} W_1^+(p,t) & \text{if } t \in T_{E_1 \setminus E_2} \text{ and } p \in P_1 \\ W_2^+(p,t) & \text{if } t \in T_{E_2 \setminus E_1} \text{ and } p \in P_2 \\ W_1^+(p,t_1) & \text{if } t = (t_1,t_2) \in T_{E_1 \cap E_2} \\ & \text{and } p \in P_1 \\ W_2^+(p,t_2) & \text{if } t = (t_1,t_2) \in T_{E_1 \cap E_2} \\ & \text{and } p \in P_2 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

- *$M_0 \in \mathbb{N}^{|P|}$ maintains the respective initial markings of G_1 and G_2:*

$$M_0(p) = \begin{cases} M_{0,1}(p) & \text{if } p \in P_1 \\ M_{0,2}(p) & \text{if } p \in P_2 \end{cases} \tag{14}$$

- *$\ell : T \to E_1 \cup E_2$ maintains the respective transition labels of G_1 and G_2, taking into account the merging of shared transitions:*

$$\ell(t) = \begin{cases} \ell_1(t) & \text{if } t \in T_{E_1 \setminus E_2} \\ \ell_2(t) & \text{if } t \in T_{E_2 \setminus E_1} \\ \ell_1(t_1) = \ell_2(t_2) & \text{if } t = (t_1,t_2) \in T_{E_1 \cap E_2} \end{cases} \tag{15}$$

Note that in our case, given that H represents a language specification for G to fulfil, the set of events of H is a subset of the set of events of G, thus the sets of events of G and $R = G \parallel H$ are the same set E. Furthermore, by definition of parallel composition, $\mathscr{L}(R)$ contains exactly the sequences in $\mathscr{L}(G)$ that, when projected on the set of events of H, are in $\mathscr{L}(H)$. Using G and R, the feedback loop depicted in Figure 1 is implemented as follows: at each step, G executes an event e, according to the active events in its current state and the current enabled events by R, evolving to a new marking. This event is then sent to the supervisor realization R, which passively executes e, also evolving to a new marking. The set of enabled events after the execution of e is the set of active events of R in the new marking. Thus, after the execution of a string $s \in E^*$, the set of enabled events for G is given by $\Gamma_R(M_s)$. In addition to simplifying the implementation, the fact that we realize supervisors using the same formalism that we use to model the system models also gives us analysis benefits. These benefits stem from the fact that, for this case, $\mathscr{L}(S/G) = \mathscr{L}(R)$, i.e., R also models the closed-loop behaviour of the system. Thus, we can use all the analysis techniques available for PNs to analyse the controlled system.

To finalize this section, we concretely state the problem of supervisor controllability.

Problem 1. *Let $G = \langle P_G, T_G, W_G^-, W_G^+, M_{G,0}, E, \ell_G \rangle$ and $H = \langle P_H, T_H, W_H^-, W_H^+, M_{H,0}, E_H, \ell_H \rangle$ be deterministic PNs (G represents the system model and H the language specification), where $E = E_c \cup E_{uc}$ and $E_H \subseteq E$. Let $R = \langle P_R, T_R, W_R^-, W_R^+, M_{0,R}, E, \ell_R \rangle$ be the parallel composition of G and H. Define a procedure to decide if R is a realization of a controllable supervisor for G.*

III. Uncontrollable Markings Revisited

To solve Problem 1, the notion of uncontrollable marking has been introduced as early as in [2], as the set of markings M for which an uncontrollable transition $t = (t_G, t_H)$ in R is not

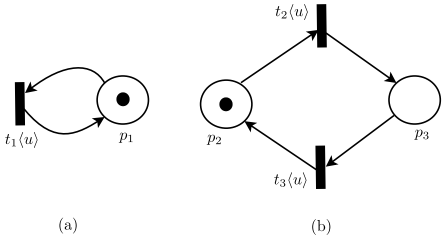

Fig. 2. (a) A PN G, representing the system. (b) A PN H, representing the language specification.

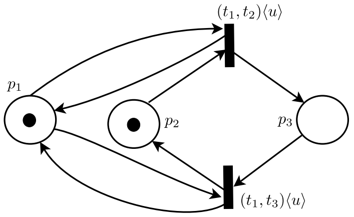

Fig. 3. The parallel composition of G and H of Figure 2.

active but for which t_G in G would be active for the projection of M to the places of G:

$$\begin{aligned}\mathscr{M}_b = \{M \in \mathbb{N}^{|P_G|+|P_H|} \mid & \\ & \text{exists } t=(t_G,t_H)\in T_R \text{ such that } t_G \in T_{uc} \text{ and} \\ & \text{for all } p\in P_G,\ M(p)\geq W_R^-(p,t) \text{ and} \\ & \text{exists } p \in P_H \text{ such that } M(p) < W_R^-(p,t)\}\end{aligned} \tag{16}$$

Then, it is proven in Theorem 1 of [4] that R realizes a controllable supervisor for G and E_{uc} if and only if none of the markings in $\mathscr{M}_b$ is reachable in R. However, this proof does not take into account the fact that, when building the parallel composition $R = G \parallel H$, one might add more than one transition for each transition of G. Recall that if T_G^e is the set of transitions in G labelled with event e and T_H^e is the set of transitions in H labelled with the same event e, then the set of transitions in R labelled with e will be $T_G^e \times T_H^e$. Thus, for each transition $t_G \in T_G^e$, $|T_H^e|$ transitions will be created in R. We illustrate this in the following example.

Example 1. *Consider the deterministic PNs G and H represented in Figure 2 (a) and (b) respectively, where event $E_{uc}=\{u\}$, i.e., t_1 is an uncontrollable transition. It is clear that both of them are deterministic. Their parallel composition $R = G \parallel H$ is depicted in Figure 3. It is also clear that R realizes a controllable supervisor for G and E_{uc} – in fact it does not even restrict the language generated by G. However, marking $M = \begin{bmatrix}1 & 1 & 0\end{bmatrix}^T$ is reachable in R and is clearly in $\mathscr{M}_b$: Transition $t=(t_1,t_3)$ is such that $M(p_1) = W_R^-(p_1,t)$, where $p_1 \in P_G$ and $M(p_3) < W_R^-(p_3,t)$, where $p_3 \in P_H$. Thus, $\mathscr{M}_b$, as defined in* (16)*, is not sound, because we need to take into account all transitions in R obtained from t_1. This fact is related to the creation of more than one transition in R corresponding to t_1.*

Remark 2. *If H is* free-labelled *(i.e., ℓ_H is an injective function), then the previous definition of uncontrollable marking is correct. Thus, results proven using the previous definition still hold when H is free-labelled.*

The above reasoning allows us to define the correct notion of uncontrollable marking in R.

Definition 15 (Uncontrollable Markings)**.** *Let G and H be deterministic PNs and $R = G \parallel H$. We define the set of uncontrollable markings as:*

$$\begin{aligned}\mathscr{M}_{uc} = \{M \in \mathbb{N}^{|P_G|+|P_H|} \mid & \\ & \text{exists } t_G \in T_{uc} \text{ such that} \\ & \text{for all } p \in P_G,\ M(p) \geq W_G^-(p,t_G) \text{ and} \\ & \text{for all } t=(t_G,t_H)\in T_R \text{ exists } p \in P_H \\ & \qquad \text{such that } M(p) < W_R^-(p,t)\}\end{aligned} \tag{17}$$

A marking M is uncontrollable if exists an uncontrollable transition t_G that is active in G by the projection of M to P_G, but all the transitions created from t_G in R, i.e. pairs in $T_G \times T_H$ with the first component equal to t_G, are not active in H by the projection of M to P_H. Thus, the following proposition can be proven by a straightforward adaptation for the correct notion of uncontrollable markings of the proof of Theorem 1 in [4][1].

Proposition 1. *Let G and H be two PNs. $R = G \parallel H$ is a realization of a controllable supervisor for G if and only if none of the markings in $\mathscr{M}_{uc}$ is reachable in R.*

Thus, to solve Problem 1, we need to define a procedure to check if an element of $\mathscr{M}_{uc}$ is reachable in R. To do that, we adapt the procedure given in [3], and represent $\mathscr{M}_{uc}$ in terms of partially covering markings for a given marking M.

Definition 16 (Partially Covering Markings)**.** *Let G be a PN, $M \in \mathbb{N}^{|P|}$, and $P^= \subseteq P$. We define the set of partially covering markings as:*

$$\begin{aligned}\mathscr{S}(M,P^=) = \{M' \in \mathbb{N}^{|P|} \mid M'(p) = M(p) \text{ for all } p \in P^= \\ \text{and } M'(p) \geq M(p) \text{ for all } p \in P\}\end{aligned} \tag{18}$$

The set $\mathscr{S}(M,P^=)$ is the (infinite) set of markings which are equal to M for places in $P^=$ and greater or equal than M for all other places. This set is a generalization of the set defined in [3], where $P^=$ is a singleton. However, in spite of being a generalization, checking the reachability of an element in this set (of possibly infinite cardinality) is still decidable. To prove this, we provide a reduction to the reachability problem of a single marking, which is known to be decidable for general PNs [5].

Proposition 2. *Let G be a PN, $M \in \mathbb{N}^{|P|}$, and $P^= \subseteq P$. Checking if a marking in $\mathscr{S}(M,P^=)$ is reachable in G, i.e., $\mathscr{R}(G) \cap \mathscr{S}(M,P^=) \neq \emptyset$ is decidable.*

Proof. The proof of this proposition relies on a simple adaptation of the construction given in [3]. We will reduce the problem of determining if there exists a reachable marking in $\mathscr{S}(M,P^=)$ to the problem of determining if a single marking is reachable in a modified net $G' = \langle P',T',W^{-'},W^{+'},M_0'\rangle$, where:

- $P' = P \cup \{p_s, p_f\}$

[1] Instead of considering a transition $(t_G,t_H) \in T_R$, one needs to consider, for a given $t_G \in T_G$, the set $\{(t_G,t_H) \in T_R \mid t_H \in T_H\}$.

- $T' = T \cup \{t_f\} \cup T_{P\setminus P^=}$, where:

$$T_{P\setminus P^=} = \{t_p \mid p \in P \setminus P^=\}$$

- $W^{-'}$ is such that:

$$W^{-'}(p,t) = \begin{cases} W^-(p,t) & \text{if } p \in P \text{ and } t \in T \\ 1 & \text{if } p = p_s \text{ and } t \in T \\ 1 & \text{if } p = p_s \text{ and } t = t_f \\ 1 & \text{if } p = p_f \text{ and } t \in T_{P\setminus P^=} \\ M(p) & \text{if } p \in P^= \text{ and } t = t_f \\ 1 & \text{if } p \in P \setminus P^= \text{ and } t = t_p \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

- $W^{+'}$ is such that:

$$W^{+'}(p,t) = \begin{cases} W^+(p,t) & \text{if } p \in P \text{ and } t \in T \\ 1 & \text{if } p = p_s \text{ and } t \in T \\ 1 & \text{if } p = p_f \text{ and } t = t_f \\ 1 & \text{if } p = p_f \text{ and } t \in T_{P\setminus P^=} \\ 0 & \text{if } p \in P^= \text{ and } t = t_f \\ 0 & \text{if } p \in P \setminus P^= \text{ and } t = t_p \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

- M'_0 is such that:

$$M'_0(p) = \begin{cases} M_0(p) & \text{if } p \in P \\ 1 & \text{if } p = p_s \\ 0 & \text{if } p = p_f \end{cases} \tag{21}$$

We were redundant in the definition of $W^{-'}$ and $W^{+'}$ to facilitate understanding the construction, which is illustrated in Figure 4. We explain the structure added to G in words:

- Place p_s is self-looped with all transitions in T.
- Transition t_f has p_s as an input place and p_f as an output place.
- Place p_f is self-looped with all transitions in $\{t_p \mid p \in P \setminus P^=\}$.
- Each place p in $P \setminus P^=$ is an input place of t_f, with weight $M(p)$.
- Each place p in $P \setminus P^=$ is an input place of the corresponding transition t_p, with weight 1.

The PN G' initially exactly mimics the behaviour of G. When G' gets to a marking M' that covers M for places in $P \setminus P^=$, t_f becomes active. After t_f fires, the behaviour of G' changes to emptying the places in $P \setminus P^=$ by firing transitions in $T_{P\setminus P^=}$. After all places in $P \setminus P^=$ are emptied, a deadlocked marking M_d is reached. If $M_d(p) = M(p)$ for all $p \in P^=$, then not only $M'(p) \geq M(p)$ for all $p \in P \setminus P^=$, but also $M(p) = M'(p)$ for all $p \in P^=$, i.e., $M' \in \mathscr{S}(M, P^=)$. Thus, the reachability problem of a marking in $\mathscr{S}(M, P^=)$ in G is equivalent to the reachability problem of M' in G', where:

$$M'(p) = \begin{cases} M(p) & \text{if } p \in P^= \\ 1 & \text{if } p = p_f \\ 0 & \text{if } p \in P \setminus P^= \text{ or } p = p_s \end{cases} \tag{22}$$

□

Before we present the final decidability result, we introduce some helpful notation.

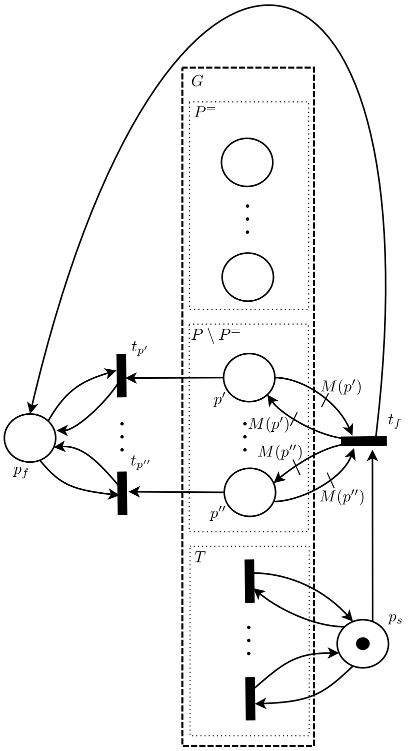

Fig. 4. Illustration of the construction for Proposition 2.

Definition 17. *Let $t_G \in T_G$. We define the set of transitions in H that will be synchronized with t_G as:*

$$T_H(t_G) = \{t' \in T_H \mid \ell(t_G) = \ell(t')\} \tag{23}$$

Definition 18. *Let $t_G \in T_G$ and $T_H(t_G) = \{t_1, ..., t_r\}$. We define the set of combinations of input places for the transitions of H that will be synchronized with t_G as:*

$$P_H(t_G) = \{p \in P_H \mid W_H^-(p,t_1) > 0\} \times ... \times \{p \in P_H \mid W_H^-(p,t_r) > 0\} \tag{24}$$

Note that, in the cases where a place $p \in P_H$ is an input place of more than one transition in $T_H(t_G)$, p can appear more than one time in a member $(p_1, ..., p_r)$ of $P_H(t_G)$.

Definition 19. *Let $t_G \in T_G$, $T_H(t_G) = \{t_1, ..., t_r\}$ and $(p_1, ..., p_r) \in P_H(t_G)$. We define the set of bad k's for t_G as:*

$$K(t_G, (p_1, ..., p_r)) = \{(k_1, ..., k_r) \in \mathbb{N}^r \mid \textit{for all } i,j = 1, ..., r, \; k_i < W_H^-(p_i, t_i) \textit{ and if } p_i = p_j \textit{ then } k_i = k_j\} \tag{25}$$

The elements of $K(t_G, (p_1, ..., p_r))$ will be used to define partially covering markings where $P^= = \{p_1\} \cup ... \cup \{p_r\}$, thus it is necessary to enforce that if $p_i = p_j$ then $k_i = k_j$, so that the partially covering marking is well-defined.

Example 2. *Figure 5 depicts a system model G (a), and a language specification H (b).*

Following the definitions above, we have the following:

- $T_H(t_G^1) = \{t_H^1, t_H^3\}$.
- $P_H(t_G^1) = \{(p_H^1, p_H^3), (p_H^4, p_H^3)\}$.

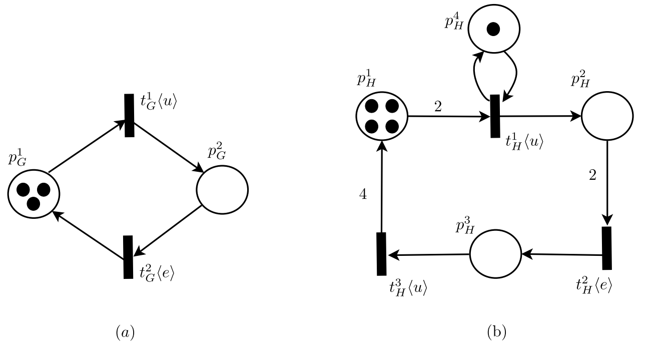

Fig. 5. (a) A PN G, representing the system. (b) A PN H, representing the language specification.

- $K(t_G^1,(p_H^1,p_H^3)) = \{(0,0),(1,0)\}$ *and* $K(t_G^1,(p_H^4,p_H^3)) = \{(0,0)\}$.

The set $K(t_G,(p_1,...,p_r))$ represents all the possible combinations of tokens in the places $p_1,...,p_r$ that will not make any of the transitions of $T_H(t_G)$ active. Thus, if t_G is uncontrollable, a marking $M \in \mathscr{R}(R)$ where (i) $M(p) \geq W_G^-(p,t_G)$ for all $p \in P_G$ and (ii) there exists $(p_1,...,p_r) \in P_H(t_G)$ and $(k_1,...,k_r) \in K(t_G,(p_1,...,p_r))$ such that $M(p_i) = k_i$ for all $i = 1,...,r$, is an uncontrollable marking. From this reasoning, we have the following result, adapted from [3]:

Proposition 3. *Problem 1 can be reduced to checking reachability of a finite union of partially covering markings.*

Proof. To check controllability, we need to check if there exists a marking in $\mathscr{M}_{uc}$ which is reachable in $R = G \parallel H$. To do that, we note that $\mathscr{M}_{uc}$ can be represented by a finite union of sets of partially covering markings. Let $t_G \in T_{uc}$, $(p_1,...,p_r) \in P_H(t_G)$ and $(k_1,...,k_r) \in K(t_G,(p_1,...,p_r))$, and consider the marking $M_{t_G,(p_1,...,p_r),(k_1,...,k_r)}$, defined over P_R as:

$$M_{t_G,(p_1,...,p_r),(k_1,...,k_r)}(p) = \begin{cases} k_i & \text{if } p = p_i \text{ for some} \\ & \text{element } p_i \text{ of} \\ & (p_1,...,p_r) \\ W_G^-(p,t_G) & \text{if } p \in P_G \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Note that $M_{t_G,(p_1,...,p_r),(k_1,...,k_r)}$ is well-defined because $(k_1,...,k_r) \in K(t_G,(p_1,...,p_r))$, and we assume that members of $K(t_G,(p_1,...,p_r))$ are such that if $p_i = p_j$, then $k_i = k_j$, for all $i,j \in \{1,...,r\}$. Furthermore, as we previously discussed, the partially covering markings $\mathscr{S}\left(M_{t_G,(p_1,...,p_r),(k_1,...,k_r)}, \{p_1\} \cup ... \cup \{p_r\}\right)$ represent an infinite set of markings that are uncontrollable. Also, all the uncontrollable markings can be represented by partially covering markings, by going through all possible uncontrollable transitions $t_G \in T_{uc}$, all elements $(p_1,...,p_r)$ of $P_H(t_G)$ and all elements $(k_1,...,k_r)$ of $K(t_G,(p_1,...,p_r))$. Thus, checking supervisor controllability is equivalent to checking that the following finite union of sets of partially covering markings is not reachable in R:

$$\bigcup_{t_G \in T_{uc}} \; \bigcup_{(p_1,...,p_r) \in P_H(t_G)} \; \bigcup_{(k_1,...,k_r) \in K(t_G,(p_1,...,p_r))} \mathscr{S}\left(M_{t_G,(p_1,...,p_r),(k_1,...,k_r)}, \{p_1\} \cup ... \cup \{p_r\}\right) \tag{27}$$

□

This proposition gives us a procedure to check if a supervisor realized as a deterministic PN $R = G \parallel H$ is controllable: check reachability in R for all the sets of partially covering markings in the finite union in equation (27). If one of them is reachable, then the supervisor is not controllable.

IV. Conclusion

In this work, we have shown that the notion of uncontrollable marking defined in [2] and used so far in SC theory of PNs is not sound, by means of a counter-example. We also presented a corrected definition and provided an adaptation for the corrected definition of the procedure given in [3] to check for controllability of deterministic PN supervisors.

Acknowledgments

The discussion in this paper mainly stems from the work of Professor Alessandro Giua, whom the authors would like to thank for his availability to confirm the counter-example for the previous notion of uncontrollable marking. The work of the first author was partially funded by the Fundação para a Ciência e Tecnologia (FCT) (ISR/IST pluriannual funding) through the PIDDAC Program funds and FCT grant FRH/BD/45046/2008, the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No 600623, STRANDS, and the EPSRC grant EP/K014293/1.

References

[1] P. Ramadge and W. Wonham, “Supervisory control of a class of discrete event processes,” *SIAM journal on control and optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[2] A. Giua, “Petri nets as discrete event models for supervisory control,” Ph.D. dissertation, Rensselaer Polytechnic Institute, 1992.

[3] ——, “Supervisory control of Petri nets with language specifications,” *Control of Discrete-Event Systems – Lecture Notes in Control and Information Sciences*, vol. 433, pp. 235–255, 2013.

[4] R. Kumar and L. Holloway, “Supervisory control of deterministic Petri nets with regular specification languages,” *IEEE Transactions on Automatic Control*, vol. 41, no. 2, pp. 245–249, 1996.

[5] E. W. Mayr, “An algorithm for the general Petri net reachability problem,” *SIAM Journal on computing*, vol. 13, no. 3, pp. 441–460, 1984.