Synchronizing sequences on a class of unbounded systems using synchronized Petri nets

Marco Pocci, Isabel Demongodin, Norbert Giambiasi, Alessandro Giua

Abstract

Determining the state of a system when one does not know its current initial state is a very important problem in many practical applications as checking communication protocols, part orienteers, digital circuit reset, etc. Synchronizing sequences have been proposed in the 60's to solve the problem on systems modeled by finite state machines.

This paper presents a first investigation of the synchronizing problem on unbounded systems, using synchronized Petri nets, i.e., nets whose evolution is driven by external input events. The proposed approach suffers from the fact that no finite space representation can exhaustively answer to the reachability problem but we show that synchronizing sequences may be computed for a particular class of unbounded synchronized Petri nets.

Marco Pocci, Isabel Demongodin, Norbert Giambiasi, Alessandro Giua are with Aix Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296, France, email: {marco.pocci,isabel.demongodin,norbert.giambiasi, alessandro.giua}@lsis.org. Marco Pocci and Alessandro Giua are also with the University of Cagliari, DIEE, Italy, email: {marco.pocci, giua}@diee.unica.it

I. INTRODUCTION

Testing problems have assumed an important role in the area of discrete event systems due to the increasing need for performance monitoring and verification of complex man-made systems. Several testing problems have been defined: see [14] for a comprehensive survey. In this work we focus in particular on the *synchronization problem*. It consists in finding an input sequence that drives a system to a known state having no (or at best partial) information on its current state and without observing the system's output. Such an input is called a *synchronizing sequence* (SS). Interesting and practical applications in this setting can be found in robotics [1], [18], bio-computing [3], [2], network theory [12], theory of codes [11] and testing synchronous circuits with no reset [5].

Typical models used for testing are *input/output automata* such as Mealy machines [16]. Recently, however, in a series of papers [21], [20] we have investigated the problem of determining a SS in the setting of Petri nets (PNs). In particular, we have shown how several approaches developed for automata can be easily applied to the class of bounded deterministic synchronized Petri nets using the *reachability graph* (RG) of a net. Such a graph is an automaton whose behavior is equivalent to that of the net and whose states are vectors in $\mathbb{N}^m$ representing reachable markings. Furthermore we have shown that for special classes of nets a SS can be computed without exploring the complete reachability set but simply analyzing the net structure.

This paper is an extended version of our previous work in [22] on synchronizing sequences' construction where we further extend our investigation to the case of *unbounded* Petri nets, i.e., nets whose reachability set is infinite. Note that, to the best of our knowledge, the synchronizing problem for unbounded models (automata or Petri nets) has never been investigated before.

The behavior of an unbounded Petri net can be approximated by a finite *coverability graph* (CG) [13]. Such a graph is an automaton where each state is a vector in $(\mathbb{N} \cup \{\omega\})^m$ representing a set of markings. An ω component denotes a place whose token content may be arbitrarily large.

The coverability graph is not unique and usually not minimal. A minimal CG has been proposed by Finkel [9], using reduction rules based on comparison between computed markings. The approach has been demonstrated to be incorrect and more efficient techniques have been proposed to correctly determine minimal coverability sets by constructing handle sets by Geeraerts *et al.* [10] or by pruning technics by Reynier and Servais [23]. However, the CG entails loss of information in terms of reachable markings and firing sequences, that can somehow prevent one to use it for systematically investigating the net properties. Recently, some efficient algorithms based on a modified CG have been proposed by Zhou and coauthors [8], [24] for the particular purpose of testing if an unbounded PN net has deadlocks or not.

Unfortunately, as pointed out by [7], synchronized PNs have a non-necessarily monotonic evolution. This is why, as explained by [6], for such nets no algorithmic CG construction has been given yet.

This paper contains two distinct contributions.

We first propose a procedure to construct a finite graph, called *modified coverability graph* (MCG), to describe the behavior of unbounded synchronized PNs. This procedure is adapted by Karp and Miller algorithm but requires a new definition of *increasing sequence* to capture those evolutions that lead to arbitrarily large markings. Unfortunately, we show that a MCG may fail to capture some transitions steps that are firable in the net and also some reachable markings: we call them *vanishing steps* and *vanishing markings*, respectively. This motivates us to look for a sufficient condition (called Assumption III.10 in the paper) that rules out these undesirable features. For unbounded synchronized nets satisfying this condition the MCG provides a faithful representation of the net behavior — analogously to the coverability graph of an unbounded Petri net — and as such is a suitable tool to determine synchronizing sequences. For this reason, in the rest of the paper we focus on the class of nets that satisfy this assumption.

Second, we extend to the unbounded case the technique we have presented in [21] for computing SS in the case of bounded synchronized nets. However, the ω symbol that is necessarily introduced in the MCG to keep the graph finite, entails (as is the case for the coverability graph of Petri nets) some loss of information. This is why, in the case of unbounded nets, the graph can only give precise information about the marking of bounded places and we consider a weaker notion of SS as a sequence that yields a marking where only the token content of bounded places must be exactly known. This concept of partial synchronization of the state has been suggested in the context of sequential machines by [4], where if a SS is prohibitively long or non existent, a selected subset of all flip-flops could be electrically reset. The analysis of the MCG gives a set of sequences — called *potentially synchronizing sequences* — and we provide a finite procedure to verify it they are also synchronizing sequences.

The paper is organized as follows. Section II provides background on Petri nets formalisms.

In section III an algorithmic construction of the coverability set of unbounded synchronized Petri nets is provided. Section IV presents our approach for SS construction. Finally, in section V conclusions are drawn and future work presented.

II. Petri net formalisms

In this section, the Petri net formalism used in this paper is recalled. First the basic notions of Petri nets are presented. Then the class of synchronized Petri nets, a non-autonomous Discrete Event System model is presented. For more details on Petri nets the reader is referred to [17], [6].

A. Petri nets

A *Petri net*[1] (PN) is a structure $N = (P, T, Pre, Post)$, where P is a set of m *places*, T is a set of n *transitions*, $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are, respectively, the *pre-incidence* and *post-incidence* matrixes that specify the weights of directed arcs from places to transitions and from transitions to places. $C = Post - Pre$ is the *incidence matrix*.

A *marking* is an application $M : P \to \mathbb{N}$ that assigns to each place of a net a nonnegative integer. A marking will be represented by a vector

$$M = [\ M(p_1)\ M(p_2)\ \ldots M(p_m)\ ]^T$$

where $M(p)$ denotes the number of tokens contained in place p. A marked PN $\langle N, M_0\rangle$ is a net N with an initial marking M_0.

A transition t is enabled at M iff $M \geq Pre(\cdot\,, t)$. An enabled transition t at M may be fired yielding the marking $M' = M + C(\cdot\,, t)$. The set of enabled transitions at M is denoted $\mathcal{E}(M)$. We write $M[\sigma\rangle$ to denote that the sequence of transitions $\sigma = t_1 \ldots t_k$ is enabled at M. Moreover $M[\sigma\rangle M'$ denotes the fact that the firing of σ from M leads to M'.

A marking M is *reachable* in $\langle N, M_0\rangle$ iff there exists a firing sequence σ such that $M_0\ [\sigma\rangle\ M$. The set of all markings reachable from M_0 defines the *reachability set* of $\langle N, M_0\rangle$ and is denoted $R(N, M_0)$.

A place is bounded if there exists $k > 0$ s.t. $\forall M \in R(N, M_0)$, $M(p) \leq k$. A marked PN $\langle N, M_0\rangle$ is said to be *k-bounded* if there exists a positive constant k such that for all

[1] Properly speaking, the model we describe here is called a *place/transition net*.

$M \in R(N, M_0)$, $M(p) \leq k$, $\forall p \in P$. A place is called k-bounded if it does not contain more than k tokens in all reachable markings, including the initial marking.

We conclude this sub-section by introducing some notations and concepts that are used in this paper.

The $preset$ and $postset$ of a place p, denoted ${}^\bullet p$ and $p^\bullet$ are, respectively: ${}^\bullet p = \{t \in T \mid Post(p,t) > 0\}$ and $p^\bullet = \{t \in T \mid Pre(p,t) > 0\}$. The set of input transitions and the set of output transitions for a set of place $\hat{P}$ are defined as: ${}^\bullet\hat{P} = \{t \in T \mid (\exists p \in \hat{P})\ t \in\ {}^\bullet p\}$ and $\hat{P}^\bullet = \{t \in T \mid (\exists p \in \hat{P})\ t \in\ p^\bullet\}$. Analogously, the $preset$ and $postset$ of a transition t are respectively ${}^\bullet t = \{p \in P \mid Pre(p,t) > 0\}$ and $t^\bullet = \{p \in P \mid Post(p,t) > 0\}$. Also, the set of input places and the set of output places for a set of transitions $\hat{T}$ are defined as: ${}^\bullet\hat{T} = \{p \in P \mid (\exists t \in \hat{T})\ p \in\ {}^\bullet t\}$ and $\hat{T}^\bullet = \{p \in P \mid (\exists t \in \hat{T})\ p \in\ t^\bullet\}$.

Let P_u and I_u (resp. P_b and I_b) denote the set of unbounded (resp. bounded) places and the corresponding indexes s.t. $I_u = \{i \mid\ p_i \in P_u\}$ (resp. $I_b = \{i \mid\ p_i \in P_b\}$). Let $m_u = |P_u|$ and $m_b = |P_b|$. We denote $M \uparrow_b$ (resp. $M \uparrow_u$) the *projection* of the marking M onto the set of bounded (resp. unbounded) places P_b (resp. P_u).

The characteristic vector or Parikh vector of a transition sequence, $\sigma = t_1 \ldots t_k$, is a vector $\pi(\sigma) \in \mathbb{N}^n$, where its component π_j represents the number of firing of transition t_j in the given sequence σ.

A graph $\mathcal{G}$ is decomposable in its maximal strongly connected components [19], classified as: i) *ergodic*, if its set of output transitions is included in its set of input transitions ; ii) *transient*, otherwise.

B. Coverability graph

A bounded PN has a finite reachability set. In this case, its behavior can be represented by a *reachability graph* (RG), i.e., a directed graph whose vertices correspond to reachable markings and whose edges correspond to enabled transitions.

The set of reachable markings of an unbounded PN, on the contrary, is not finite. Karp and Miller [13] have proposed a procedure to compute a finite representation of the state-space of unbounded PNs. The procedure requires to identify *increasing sequences* at a marking M, i.e., firing sequences σ that produces an infinitely long evolution $M = M_1[\sigma\rangle M_2[\sigma\rangle M_3 \cdots$ where[2] $M_i \lneq M_{i+1}$ for $i = 1, 2, \ldots$. These sequences strictly increase the number of tokens in certain places and their final behavior is approximated using an acceleration technique, that works because PNs are monotonic, i.e. a sequence of transitions which is firable from a marking M is also firable from all markings M' such that $M' \geq M$.

In a coverability graph, each node is labeled with an m dimensional row vector whose entries may either be an integer number or may be equal to the special symbol ω, while arcs are elements in T. In particular, its nodes are labeled with an ω-marking, defined as follows.

Definition II.1. ***(ω-marking)*** *Let $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$. An ω-marking of a PN N with m places is a column vector $M_\omega \in \mathbb{N}_\omega^m$, whose components may either be an integer number or be equal to ω.* ■

Symbol ω denotes that the marking of the corresponding place may grow indefinitely. Note that for all $n \in \mathbb{N}$ it holds $\omega > n$ and $\omega \pm n = \omega$.

The following example illustrates the coverability graph of an unbounded PN, obtained by the classic Karp-Miller algorithm [13].

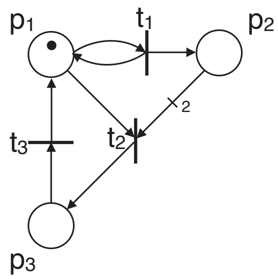
Example II.2. *Consider the PN in Figure 1a where $P = \{p_1, p_2, p_3\}$, $T = \{t_1, t_2, t_3\}$ and $M_0 = [1\,0\,0]^T$. Its reachability graph, depicted in Figure 1b, shows that the firing sequence $\sigma = t_1$ increases the marking of place p_2. The repeated firing of σ makes unbounded the same place. For this reason, σ is an increasing sequence. The corresponding coverability graph is shown in Figure 1c.* ■

We now introduce the notion of covering set that provides a larger approximation of the reachability set.

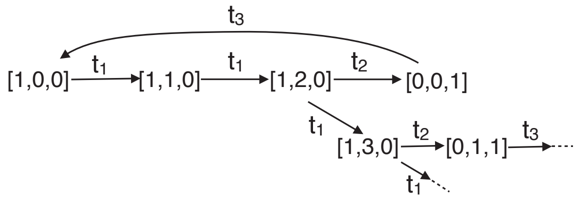
Definition II.3. ***(Covering set)*** *Given a marked PN $\langle N, M_0\rangle$, let V be the set of nodes of its CG. The covering set of $\langle N, M_0\rangle$ is*

$$CS(N, M_0) = \bigcup_{M_\omega \in V} cov(M_\omega),$$

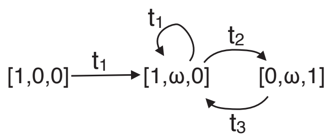
[2]Given two vectors $x, y \in \mathbb{R}^n$ we write $x \lneq y$ to denote that $x \leq y$, i.e., each component of x is smaller than or equal to the corresponding component of y, and that $x \neq y$, i.e., the two vectors are not identical.



(a)



(b)



(c)

Fig. 1: An unbounded PN (a), its partial RG (b) and its CG (c).

where $cov(M_\omega) = \{M \in \mathbb{N}^m \mid \; M(p) = M_\omega(p)$ *if* $M_\omega(p) \neq \omega\}$.

A marking M_ω *is called a* covering marking *for* M *if* $M \in cov(M_\omega)$. *In this case, we write* $M_\omega \geq_\omega M$. ■

We finally recall a classical result, showing that the coverability graph captures all evolutions of a net (but may also contain some evolutions that the net cannot generate).

Proposition II.4 ([17]). *Consider a marked PN* $\langle N, M_0 \rangle$ *and its CG.*

1) *Marking* M *is reachable in the net* $\Longrightarrow$ *in the CG there exists a node* M_w *such that* $M \in cov(M_\omega)$, *i.e.,* $R(N, M_0) \subseteq CS(N, M_0)$.
2) *Sequence* $\sigma = t_{j_1} t_{j_2} \cdots t_{j_k}$ *is firable from* $M \in R(N, M_0)$ *with the evolution* $M[t_{j_1}\rangle M_1[t_{j_2}\rangle M_2 \cdots [t_{j_k}\rangle M$ $\Longrightarrow$ *in the CG from all nodes* M_w *such that* $M \in cov(M_\omega)$ *there exists a directed path* $M_\omega t_{j_1} M_{\omega,1} t_{j_2} M_{\omega,2} \cdots t_{j_k} M_{\omega,k}$ *such that* $M_i \in cov(M_{\omega,i})$ *for* $i = 1, 2, \ldots, k$.

C. Synchronized Petri nets

A *synchronized Petri net* [6] is a structure $\langle N, E, f \rangle$ such that: i) N is a PN; ii) E is an alphabet of input events; iii) $f : T \to E$ is a labeling function that associates with each transition t an input event $f(t)$. A marked synchronized PN $\langle N, M, E, f \rangle$ is a synchronized PN with a marking M.

We denote the set of transitions associated with the input event e by: $T_e = \{t \in T \mid f(t) = e\}$ and the set of enabled transitions associated with event e as: $\mathcal{E}_e(M) = T_e \cap \mathcal{E}(M)$.

The evolution of a synchronized net is driven by the occurrence of an input event sequence that produces a sequence of transition firings. At marking M, transition $t \in T$ is fired only if:

1) transition t is enabled, i.e., $t \in \mathcal{E}(M)$;
2) the event $e = f(t)$ occurs.

Note that the occurrence of an input event $e \in E$ at marking M forces the simultaneous firing of all transitions in $\mathcal{E}_e(M)$ provided there are no conflicts among them. On the contrary, the occurrence of an event e does not produce the firing of a non enabled transition $t \in T_e$.

We say that there exists an *effective conflict* at marking M between two enabled transitions sharing the same label $t, t' \in \mathcal{E}_e(M)$ if the following condition holds: there exist a place p such that $t, t' \in p^\bullet$ and $M(p) < Pre(p, t) + Pre(p, t')$. Moreover, a synchronized PN is said to be deterministic if for all reachable markings there is no effective conflict between enabled transitions sharing the same event.

Definition II.5. ***(Deterministic synchronized PN)*** *A marked synchronized PN $\langle N, M_0, E, f \rangle$ is said to be* deterministic *if the following condition holds:*

$$(\forall M \in R(N, M_0))\ (\forall e \in E) \quad M \geq \sum_{t \in \mathcal{E}_e(M)} Pre(\cdot, t). \qquad \blacksquare$$

Remark II.6. *A sufficient structural condition for a synchronized PN $\langle N, E, f \rangle$ to be deterministic is that there does not exist a place p such that $t, t' \in p^\bullet$ and $f(t) = f(t')$.* $\blacksquare$

Example II.7. *Based on the Petri net in Fig. 1a, consider the synchronized PN in Fig. 2, with $E = \{e_1, e_2\}$, $f(t_1) = f(t_3) = e_1$ and $f(t_2) = e_2$. This net is deterministic as t_1 and t_2 (the only transitions in conflict) do not share the same input event.* $\blacksquare$

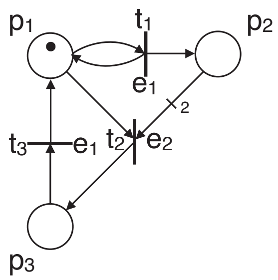


Fig. 2: An unbounded deterministic synchronized Petri net.

In the rest of the paper we will focus on deterministic synchronized PNs: all presented results apply to this class of nets. Furthermore in all examples we will consider nets that satisfy the structural condition mentioned in Remark II.6 so that one may verify by inspection that they are deterministic.

We conclude this section presenting the formal semantics (i.e., the evolution rule) of deterministic synchronized PNs.

Definition II.8 (Evolution of a deterministic synchronized PN)**.** *In a deterministic synchronized PN, when an input event e occurs at a marking M, all enabled transitions receptive to this event $\mathcal{E}_e(M) = T_e \cap \mathcal{E}(M)$ fire simultaneously in a single* step

$$M[e|\tau\rangle M', \qquad \textit{with } \tau = \mathcal{E}_e(M) \textit{ and } M' = M + \sum_{t \in \tau} C(\cdot, t). \tag{1}$$

Here $M[e|\tau\rangle M'$ denotes that the occurrence of the input event e at M yields marking M' by the firing of step τ. When there is no need to specify the firing step corresponding to e we simply write $M[e|\cdot\rangle M'$. ■

According to this definition, an input event sequence[3] $w = e_1 e_2 \cdots e_k \in E^*$ drives the net along the evolution

$$M_0[e_1|\tau_1\rangle M_1[e_2|\tau_2\rangle M_2[e_3|\tau_3\rangle \cdots [e_k|\tau_k\rangle M_k \tag{2}$$

where the steps are $\tau_i = \mathcal{E}_{e_i}(M_{i-1})$ for $i = 1, \ldots, k$, M_0 is the initial marking and

$$M_i = M_{i-1} + \sum_{t \in \tau_i} C(\cdot, t).$$

[3]Here, * denotes the Kleene star operator and E^* represents the set of all sequences on alphabet E.

Note that firing step τ_i is empty if no transition receptive to event e_i is enabled at M_{i-1} and in this case $M_i = M_{i-1}$.

Finally we denote evolution (2) by $M_0[w|\sigma\rangle M_k$, where $\sigma = \tau_1 \cdots \tau_k$ is a sequence of steps. When there is no need to specify the firing step sequence corresponding to w we simply write $M[w|\cdot\rangle M'$.

D. Synchronization problem

Consider a system with an *initial state uncertainty*, i.e., the initial state is not perfectly known but is only known to belong to a set of initial states $\mathcal{M}_0$. In the worst case, the set of initial states may coincide with the entire state space. In a synchronization problem, the goal is to find an input event sequence that, regardless of the initial state, drives the system to a known target state $\bar{M}$. In the case of a synchronized PN $\langle N, E, f\rangle$, we assume that a *starting marking* M_0, i.e., a marking where the net has certainly been in the past, is known. Thus the initial state uncertainty is usually given by $\mathcal{M}_0 = R(N, M_0)$.

In a previous work [21], we have studied the existence of a synchronizing sequence (SS) on bounded synchronized PNs. It has been proved that a bounded deterministic synchronized PN $\langle N, E, f\rangle$, can be synchronized to a marking $\bar{M}$ only if its reachability graph $\mathcal{G}$ has a single ergodic component and $\bar{M}$ belongs to it.

The objective of this paper is to study under which conditions similar approaches can be derived for the class of unbounded synchronized PNs.

III. A MODIFIED COVERABILITY GRAPH FOR UNBOUNDED SYNCHRONIZED PNS

In this section we show how the behavior of unbounded synchronized PNs can be described by a finite graph that we call *modified coverability graph* (MCG) because it derives from the classical construction of Karp and Miller [13]. There are however several issues to be considered in this setting. The first challenge is due to the fact that synchronized PNs do not satisfy the monotonicity property of PN mentioned in Section II-B hence a new definition of increasing sequence is required. This new definition is used to modify the procedure of Karp and Miller so that a MCG can be defined for synchronized net. We also show, however, that this representation may fail to capture all evolutions of a synchronized net. This will motivate the definition of a special class of nets for which the modified coverability graph is ensured to describe all evolutions: in the subsequent sections we will derive a procedure to compute synchronizing sequences for this class of net.

A. Monotonicity property of synchronized nets

In a PN from the definition of transition enabling, it follows that if $M' \geq M$ then $\mathcal{E}(M') \supseteq \mathcal{E}(M)$, i.e., a transition enabled by a marking M is also enabled by any other marking M' that covers it. From this, a well know property follows.

Proposition III.1 ([17])**.** ***(Monotonicity of Petri nets)*** *In a PN, if a transition t may fire from a marking M it can also fire any other marking M' that covers it, i.e.,*

$$M[t\rangle \quad \textit{and} \quad M' \geq M \qquad \Longrightarrow \qquad M'[t\rangle.$$

□

However, for synchronized PNs a weaker property holds.

Proposition III.2. ***(Weak monotonicity of synchronized Petri nets)*** *In a synchronized PN, let $e \in E$ be an input event and T_e be the set of transitions receptive to this event. If input event e produces from marking M a firing step $\tau \subseteq T_e$ then the occurrence of e from a marking M' that covers M produces a step τ' that is a superset of τ, i.e.,*

$$M[e|\tau\rangle \quad \textit{and} \quad M' \geq M \qquad \Longrightarrow \qquad M'[e|\tau'\rangle \quad \textit{with} \quad \tau' \supseteq \tau.$$

Proof. Follows from (1), because $\tau' = \mathcal{E}_e(M') = T_e \cap \mathcal{E}(M') \supseteq T_e \cap \mathcal{E}(M) = \mathcal{E}_e(M) = \tau$.
□ □

This property implies that in a synchronized PN an evolution step $[e|\tau\rangle$ that occurs from a given marking, may not be possible from a larger marking. The main consequence of this is that the notion of increasing sequences for PNs — that follows from the monotonicity property — must be suitably redefined for synchronized PN, as discussed in the following example.

Example III.3. *Consider the net Fig. 3a: this is a synchronized PN where input event e is associated with both transitions t_1 and t_2. We look first at the underlying PN, i.e., we ignore the input event and consider the autonomous behavior where an enabled transition may fire. We observe that from the initial marking $M_0 = [0]$, source transition t_1 is enabled and can fire*

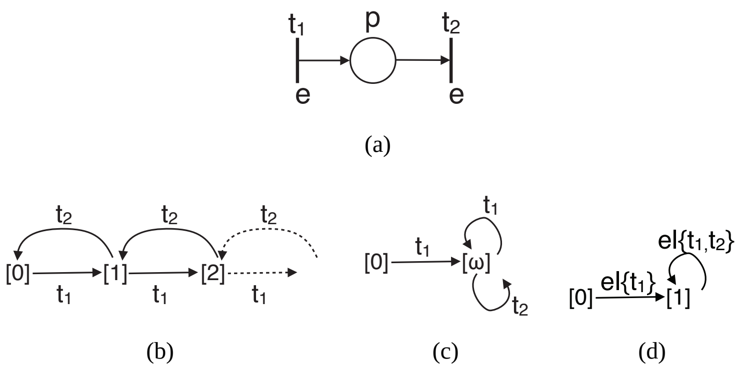


Fig. 3: An unbounded synchronized PN (a), the RG of the underlying PN (b), the CG of the underlying PN (c) and the RG of the synchronized PN (d).

reaching marking $M_1 = [1]$. We have thus identified an increasing transition sequence $\sigma = t_1$ that can fire indefinitely — since by monotonicity it is also enabled by any marking greater than M_0 — increasing the token content of place p, as shown by the reachability graph of the PN in Fig. 3b. The procedure of Karp and Miller recognizes this increasing sequence and produces the finite coverability graph of the PN shown in Fig. 3c, where one can observe that the unbounded place p is marked with ω.

Consider, however, the evolution of the synchronized PN. When input event e is applied at the initial marking $M_0 = [0]$, only transition t_1 fires, because transition t_2, although receptive to this event, is not enabled; thus one obtains the evolution $M_0[e|\{t_1\}\rangle M_1$. However, although $M_1 > M_0$ step $\{t_1\}$ cannot fire from M_1 (hence the non monotonicity) according to the evolution rule presented in Definition II.8. In fact, from this marking the application of the input event produces the evolution $M_1[e|\{t_1, t_2\}\rangle M_1$, i.e., both enabled transitions fire simultaneously and the marking does not change, as shown by the reachability graph in Fig. 3d. We point out that place p is bounded in the synchronized net although it is unbounded in the underlying PN. ■

B. An algorithmic construction of the MCG

In the construction of the coverability graph for PNs, a sequence is identified as increasing if it yields a larger marking. This condition, however, fails to identify increasing sequences in the case of synchronized PNs, as discussed in the previous subsection. In this subsection we propose a new notion of increasing sequence for synchronized nets that will allow us to derive a procedure for constructing a modified coverability graph.

We first introduce the notion of increasing input event sequences.

Definition III.4. ***(Increasing input sequence)*** *Consider a marked synchronized PN* $\langle N, M_0, E, f\rangle$*. An input sequence* $w \in E^*$ *is called* increasing *at marking* $M_1 \in R(N, M_0)$ *if:*

$$\begin{cases} M_1[w|\sigma\rangle M_2[w|\sigma\rangle M_3[w|\sigma\rangle \cdots \\ M_i \lneq M_{i+1} \ \forall i = 1, 2, \ldots \end{cases} \quad \blacksquare$$

In other words, an increasing input sequence applied repetitively starting from M_1, always produces the same firing step sequence σ leading to a greater marking.

The following proposition provides a sufficient condition for a sequence to be increasing.

Proposition III.5. *Consider a synchronized PN* $\langle N, E, f\rangle$ *and a marking* M*. Let* M' *and* M'' *be respectively the marking reached after a first and a second application of input sequence* w*, i.e.,* $M[w|\sigma\rangle M'[w|\sigma'\rangle M''$*. Sequence* w *is an increasing input sequence at* M *if the following three conditions hold:*

C1) $\sigma = \sigma'$;

C2) $M \lneq M' \lneq M''$;

C3) $(\forall p$ *such that* $M'(p) > M(p))$ $(\forall t \in p^\bullet)$ $M'(p) \geq Pre(p, t)$.

Proof. Conditions C1) and C2) are necessary to ensure that input sequence w is increasing. Condition C3) guarantees that all subsequent markings reached by repeated applications of the input sequence will not modify the corresponding step firing sequence. □ □

We now present an algorithm to construct a finite graph that will be used to describe the behavior of unbounded synchronized PNs. Such a graph is called *modified coverability tree* (MCT) because it derives from the classical construction of Karp and Miller [13].

Algorithm 1. ***MCT construction for deterministic synchronized PNs***

Input: a deterministic marked synchronized PN $\langle N, M_0, E, f\rangle$.

Output: a MCT $\mathcal{T}$.

1. Label the root node q_0 with the initial marking M_0 and tag it ”new”.
2. **While** a node tagged ”new” exists, **do**

2.1. Select a node q tagged ”new”.

2.2. Let M be the label of q.

2.3. **For** all $e \in E$ such that $\mathcal{E}_e(M) \neq \emptyset$:

2.3.1. Let $M' = M + \sum_{t \in \mathcal{E}_e(M)} C(\cdot, t)$ be the marking reached firing all enabled transitions $t \in T_e$.

2.3.2. Let $\hat{Q}$ be the set of nodes met on a backward path from q to q_0 whose label is $\hat{M} \lneq M'$.

2.3.3. **For** all nodes $\hat{q} \in \hat{Q}$ labeled $\hat{M}$,

2.3.3.1. let w and σ be the input sequence and the corresponding firing step sequence s.t. $\hat{M}[w|\sigma\rangle M'$.

2.3.3.2. Let apply again w from M', obtaining $M'[w|\sigma'\rangle M''$.

2.3.3.3. **If** the three following conditions hold:

C1) $\sigma = \sigma'$;

C2) $\hat{M} \lneq M' \lneq M''$;

C3) $(\forall p$ such that $M'(p) > M(p))$ $(\forall t \in p^\bullet)$ $M'(p) \geq Pre(p, t)$;

then let $M'(p) = \omega$.

2.3.4. Add a new node q' and label it M'.

2.3.5. Add an arc labeled $e|\mathcal{E}_e(M)$ from q to q'.

2.3.6. **If** there exists already in the tree a node with label M', **then** tag node q' ”duplicate”, **else** tag it ”new”.

2.4. Untag node q. ■

The MCG is obtained from the MCT by fusing duplicate nodes with the untagged node with the same label.

We point out that the complexity of this algorithm is comparable with that of computing the coverability graph of a PN using the algorithm of Karp and Miller [13], which is known to be non-primitive recursive [15]. Algorithm 1 is a reformulation of the algorithm in [13] in terms of synchronized nets, but contains an essential difference: step 2.3.3.3 that embeds the test for repetitive input sequences. This difference does not affect the order of computational complexity.

Example III.6. *By applying Algorithm 1 to the unbounded synchronized PN in Fig. 2, the*

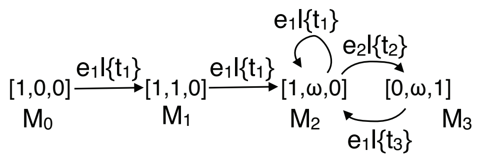


Fig. 4: The MCG of the synchronized PN of Fig. 2.

MCG depicted in Fig. 4 can be constructed. As an example of this construction, note that input sequence e_1 from M_0 satisfies conditions C1) and C2) but does not satisfy condition C3). In fact, it increases the marking of place p_2 but the increased marking $M_1(p_2)$ is not sufficiently large to enable transition t_2 that outputs place p_2. However, the occurrence of input sequence e_1 from M_1 satisfies also condition C3) and this justifies why in the MGC in Fig. 4 the marking of p_2 is set to ω in M_2. ■

Note that in a MCG there is a one-to-one mapping between a node and its label, hence in the following we will not distinguish between a node of the graph and the ω-marking that labels it.

The boundedness of the MCG is proven by the following proposition.

Proposition III.7. *Consider an unbounded marked synchronized PN $\langle N, M_0, E, f\rangle$. Its MCG constructed via Algorithm 1 is a finite graph.*

Proof. We just provide a sketch of the proof that is based on the results presented by Karp and Miller in [13] in the framework of PNs. These authors have shown that by recognising increasing sequences and using the symbol ω to denote the increasing components of the markings the procedure to construct the tree halts in a finite number of steps — i.e., the constructed tree has a finite number of vertices — even if the net has an infinite reachability set. A similar reasoning applies to Algorithm 1 whose main difference from [13] consists in the computation of increasing sequences. Note that in synchronized nets reaching an increased marking after the occurrence of an input sequence — condition C2) — is only a necessary condition for the sequence to be increasing: if the repeated application of such an input sequence produces an ever increasing markings, then in a finite number of steps also conditions C1) and C3) will be satisfied, so that the identification of an increasing input sequence is only postponed for a finite number of steps.

□ □

C. Vanishing steps and vanishing markings

We now address the problem of determining whether the modified coverability graph provides a faithful representation of the behavior of a synchronized net. Unfortunately the relatively strong results — summarized by Proposition II.4 — that hold for the coverability graph of PNs do not hold for the MCG of synchronized nets. Next example shows that there may exist steps and reachable markings that are not represented in the MCG.

Example III.8. *Consider the synchronized PN in Fig. 5a. Its partial RG and its MCG are depicted, respectively, in Fig. 5b and Fig. 5c. Consider the application of an input sequence* $w = e_1 e_3 e_3$ *at initial marking* $M_0 = [0\,0\,1]^T$ *that produces the step sequence* $\{t_1\}\{t_4\}\{t_4\}$*. The net reaches marking* $M = [1\,1\,1]^T$ *from which the following increasing sequence may occur:*

$$[1\,1\,1]^T \; [e_2|\{t_3\}\rangle \; [1\,1\,2]^T \; [e_2|\{t_3\}\rangle \; [1\,1\,3]^T \; [e_2|\{t_3\}\rangle \; [1\,1\,4]^T \; [e_2|\{t_3\}\rangle \; \cdots$$

However in the MCG in Fig. 5c the step $e_2|\{t_3\}$ *is not firable from the* ω*-marking* $M_\omega = [\omega\,\omega\,1]^T$ *that covers* M*, because the application of event* e_2 *from* M_ω *produces the “larger” step* $e_2|\{t_2, t_3\}$*. This in turn implies that in the MCG there exists no marking covering the reachable markings* $\{[\omega\,\omega\,k]^T \mid k = 2, 3, \ldots\}$*. Furthermore, since the missing step* $e_2|\{t_3\}$ *in this particular case is associated with an increasing sequence, the MCG fails to recognize that place* p_3 *is unbounded.* ■

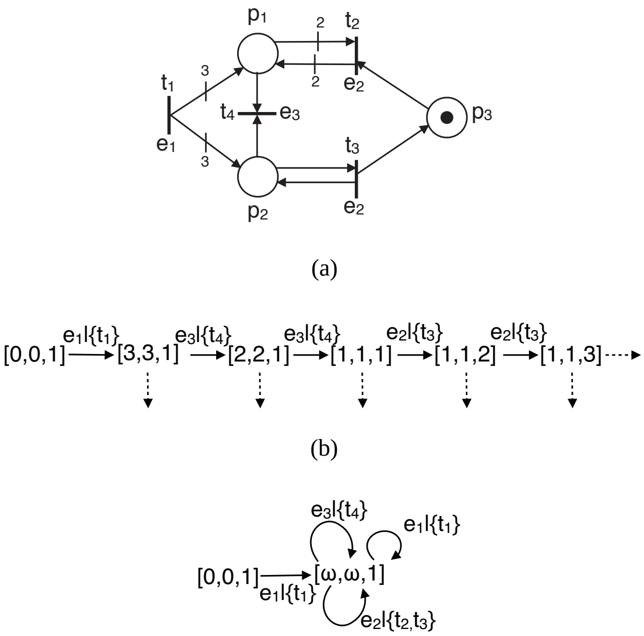
We formalize the steps and marking that are not represented in the MCG with the following definition.

Definition III.9. ***(Vanishing steps and vanishing markings)*** Consider an unbounded marked synchronized PN $\langle N, M_0, E, f\rangle$ and let $\mathcal{G}$ be its MCG, constructed by means of Algorithm 1. Suppose that there exists in $\mathcal{G}$ an ω-marking M_ω and there exists in the net a reachable marking $M \in cov(M_\omega)$ such that for some input event $e \in E$:

i) $M[e|\tau\rangle$, i.e., step $e|\tau$ is firable from M, with $\tau \neq \emptyset$;

ii) $M_\omega[e|\tau'\rangle$ with $\tau' \supsetneq \tau$, i.e., a “larger” step $e|\tau'$ is firable from M_ω.

Then step $e|\tau$ is called a *vanishing step*. A marking that can only be reached in the net by firing a sequence containing a vanishing step is called a *vanishing marking*. ■



(a)

(b)

(c)

Fig. 5: An unbounded synchronized PN with vanishing markings (a), and its RG (b) and its MCG (c).

The proposed MCG lacks in representing vanishing steps and markings and as such does not always provide a faithful representation of the net behavior (this was the case of the net in Example III.8). For this reason we now propose a restricted class of synchronized nets for which we can ensure that the MCG does not contain vanishing steps and markings.

assum III.10. *Given a marked synchronized PN $\langle N, M_0, E, f\rangle$, let $P_u \subseteq P$ be the set of markings that are unbounded according to the MCG $\mathcal{G}$ constructed by means of Algorithm 1, i.e.,*

$$P_u = \{p \in P \ \mid \ (\textit{there exists } M_\omega \textit{ in } \mathcal{G})\ M_\omega(p) = \omega\}. \tag{3}$$

We assume that for every transition $t \in P_u^\bullet$ does not exists $t' \in T$ such that $t \neq t' \wedge f(t) = f(t')$. ■

The previous assumption ensures that any transition outputting a place detected as unbounded by the MCG is associated with an input event which is not shared with any other transition. Obviously the net studied in Example III.8 does not satisfy this assumption because, say, transition t_2, outputting place p_1 — that is detected as unbounded by inspection of the MCG — shares label e_2 with transition t_3.

Proposition III.11. *There exists no vanishing step or vanishing marking in the MCG of an unbounded synchronized PNs $\langle N, M_0, E, f\rangle$ satisfying Assumption III.10, hence $R(N, M_0) \subseteq CS(N, M_0)$.*

Proof. Consider the MCG $\mathcal{G}$ of an unbounded synchronized PN, constructed by Algorithm 1, and let M_ω be an ω-marking in $\mathcal{G}$. Assume that from a reachable marking $M \in cov(M_\omega)$ step $e|\tau$ may occur (with $\tau \neq \emptyset$) and let P_u be the set defined in (3). Then two cases are possible.

Case 1) $\tau \cap P_u^\bullet = \emptyset$. Then step $e|\tau$ is also firable from M_ω because $\mathcal{E}_e(M) = \mathcal{E}_e(M_\omega)$, since $M_\omega(p) = M(p)$ for all places $p \notin P_u$.

Case 2) $\tau \cap P_u^\bullet \neq \emptyset$. In this case, by Assumption III.10, set $\tau = T_e = \{t\}$ is a singleton set. This implies that $\mathcal{E}_e(M_\omega) = T_e \cap \mathcal{E}(M_\omega) = \{t\}$ hence step $e|t$ is also firable from M_ω.

This shows that there exists no vanishing step, and as a consequence, no vanishing marking. □ □

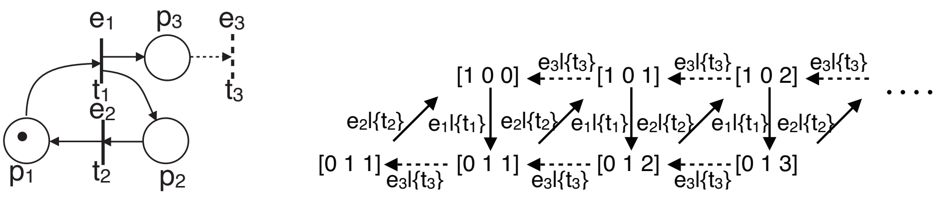
Thus for unbounded synchronized PNs satisfying Assumption III.10 the MCG provides a faithful representation of the net behavior — analogously to the coverability graph of an unbounded PN — and as such can be used to determine synchronizing sequences by the procedure that will be presented in the following section.

We point out that Assumption III.10 is only a sufficient (but not necessary) condition to rule out the existence of vanishing steps and markings. For this reason the procedure presented in the next section can be used with a larger class of synchronized nets, although currently we lack a general characterization of this class.

IV. Synchronizing sequences of unbounded deterministic synchronized PNs

The objective of this section is first to define synchronizing sequences (SSs) for unbounded PNs. Next we discuss the computation of potentially synchronizing sequences from the MCG of a given net. We finally use these sequences to compute SSs for the net itself.

First of all, note that the ω symbol, used to obtain a finite coverability graph, entails loss of information in terms of reachable markings and of firing sequences. Next example shows what kind of problems we may encounter in determining SSs.



(a) (b)

Fig. 6: A unbounded synchronized PN (a) and its RG (b).

Example IV.1. *Consider the synchronized Petri net in Fig. 6a (without the dashed transition t_3) with an initial marking $[1\,0\,0]^T$ and its infinite reachability graph in Fig. 6b (without the dashed arc corresponding to step $e_3|\{t_3\}$). Here the set of bounded places and the set of unbounded places are respectively $P_b = \{p_1, p_2\}$ and $P_u = \{p_3\}$. This graph does not have an ergodic component, because all the nodes are transient components (such a case cannot occur in bounded nets) hence no SS exists.*

Consider now the unbounded PN in Fig. 6a (including dashed transition t_3) with an initial marking $[1\,0\,0]^T$ and its infinite reachability graph in Fig. 6b (including the dashed arcs corresponding to step $e_3|\{t_3\}$). Also in this case $P_b = \{p_1, p_2\}$ and $P_u = \{p_3\}$. Suppose we want to reach target marking $\bar{M} = [1\,0\,0]^T$.

For such a marked net, it holds that $R(N, M_0) = \{M \in \mathbb{N}^3 \mid M(p_1) + M(p_2) = 1,\ M(p_3) = k \in \mathbb{N}\}$.

Obviously, the input sequence $w = e_2\{e_3\}^k$ drives the net to $\bar{M}$ from any marking $M = [1\,0\,u]^T$ and $M = [0\,1\,u]^T$ with $u \leq k$. However, since u can be arbitrarily large, properly speaking no SS to $\bar{M}$ exists for this net.

Finally, note that in both cases we have discussed, it is always possible to reach a marking where the token content of places p_1 and p_2 is known. In fact, from any reachable marking the input sequence $w = e_2$ drives the net to $M(p_1) = 1$ and $M(p_2) = 0$. ■

The previous example shows that in an unbounded net one cannot find a SS that leads from any reachable marking to a marking where the token content of an unbounded place is known. This motivates the following extended definition of SS that only takes into account the set of bounded places P_b.

Definition IV.2. ***(Synchronizing Sequence on unbounded synchronized PNs)*** Consider a marked unbounded synchronized PN $\langle N, M_0, E, f\rangle$ with set of bounded places P_b. An input sequence $w \in E^*$ is called a Synchronizing Sequence (SS) for a target marking $\bar{M} \in R(N, M_0)$ if for all $M \in R(N, M_0)$ it holds $M[w|\cdot\rangle \bar{M}'$ with $\bar{M}' \uparrow_b = \bar{M} \uparrow_b$. ■

The set of all synchronizing sequences for a given marking $\bar{M}$ is denoted $\mathcal{SS}(N, M_0, \bar{M})$.

According to the previous definition a SS for a target marking $\bar{M}$ drives the net from any (unknown) reachable marking to a marking identical to $\bar{M}$ in terms of bounded places. Clearly here we are assuming that a target marking implicitly defines a target set of markings.

Definition IV.3. ***(Synchronization target marking set)*** Consider a marked unbounded synchronized PN $\langle N, M_0, E, f\rangle$ with set of bounded places P_b. The *target marking set* for a given marking $\bar{M} \in R(N, M_0)$ is[4]

$$\mathcal{TM}(\bar{M}) = \{M \in \mathbb{N}_\omega^m \mid M \uparrow_b = \bar{M} \uparrow_b\}.$$

■

The approach we propose to search for SS for unbounded nets is inspired by approach developed in [21] for bounded nets. It requires three main steps.

i) computation of the MCG $\mathcal{G}$;

ii) computation of a potentially SS from the analysis of the MCG $\mathcal{G}$;

iii) validation of a potentially SS.

Step i), i.e., the computation of the MCG, is done by Algorithm 1, so in the rest of this section we focus on the last two steps of the procedure.

A. Potentially synchronizing sequences

The MCG generated by Algorithm 1 is an automaton where each transition is labeled by a pair $e|\tau$ with $e \in E$ and $\tau \subseteq T$; hence it can be seen as an automaton with input alphabet E and output alphabet 2^T.

[4]We are assuming that target marking set can also include ω-markings.

Given a target marking of the net, we want to determine an input sequence that synchronizes the MCG to a node corresponding to the target marking, and call this a *potentially synchronizing sequence* (PSS).

Definition IV.4. ***(Set of potentially synchronizing sequence)*** Given the MCG $\mathcal{G}$ of a marked synchronized PN with input alphabet E and set of bounded placed P_b, the *set of all potentially synchronizing sequences* for a target marking $\bar{M}$ is defined by:

$$\mathcal{PSS}(\mathcal{G}, \bar{M}) = \{w \in E^* \mid (\forall M \in \mathcal{G})\ M \xrightarrow{w} M' \wedge M' \uparrow_b = \bar{M} \uparrow_b\}. \qquad \blacksquare$$

Note that in the previous definition we are denoting the reachability relation on the MCG by $M \xrightarrow{w} M'$, as opposed to the reachability relation on the net denoted $M[w|\cdot\rangle M'$.

To compute a PSS, i.e., a SS for the MCG, we propose to extend the approach presented in [21] for bounded nets that we briefly summarize in the following. Details can be found in [21].

We start by completing the MCG $\mathcal{G}$ (see [21]) to make sure that from any state all input event occurrences are considered. Hence for any reachable marking M and for every input e such that $\mathcal{E}_e(M) = \emptyset$, we add to $\mathcal{G}$ a self loop labelled $e|\emptyset$. The completed MCG is denoted $\tilde{\mathcal{G}}$.

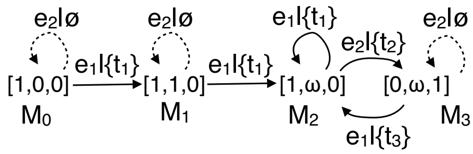
Secondly, we construct the *auxiliary graph* (AG) $\mathcal{A}(\tilde{\mathcal{G}})$ from the completely specified MCG $\tilde{\mathcal{G}}$. This auxiliary graph is a new graph whose nodes are the unordered pairs (M_i, M_j) of markings of $\mathcal{G}$, including pairs (M_i, M_i) of identical markings and such that there is an edge from (M_i, M_j) to (M_p, M_q) labeled with an input event $e \in E$ if in $\mathcal{G}$ there exists an arc from M_i to M_p and an arc from M_j to M_q, both associated to input event e.

The two preliminary steps mentioned above are reviewed by means of the following example.

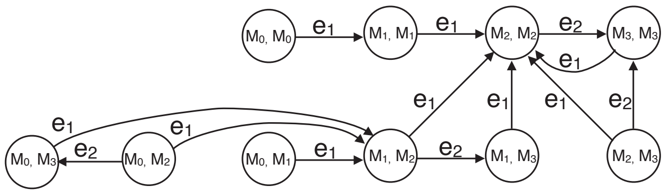
Example IV.5. *Consider the synchronized PN in Fig. 2. Its completely specified MCG and its AG are respectively depicted in Fig. 7a and in Fig. 7b. For the latter, self-loop are omitted since useless for the synchronization scope.* $\blacksquare$

Every path from (M_i, M_j) to $(\bar{M}, \bar{M})$ determines an input sequence that certainly drives the MCG to marking $\bar{M}$ if the previous marking was either M_i or M_j. Hence the PSS is constructed concatenating the input sequences determined by synchronizing two markings at time.

The following algorithm allows one to construct a PSS w, which is not necessarily the shortest one but leads the MCG to a target marking.



(a)



(b)

Fig. 7: Completely specified MCG $\tilde{\mathcal{G}}$ (a) and AG $\mathcal{A}(\tilde{\mathcal{G}})$ (b) of the unbounded synchronized PN in Fig. 2.

Algorithm 2. ***(Computing a PSS for a marking $\bar{M}$)***

Input: *A marked unbounded synchronized PN $\langle N, M_0, E, f\rangle$ satisfying Assumption III.10 and a bounded target marking $\bar{M} \in R(N, M_0)$.*

Ouput: *A PSS w.*

1. *Let $\mathcal{G}$ and $\mathcal{A}(\tilde{\mathcal{G}})$ be respectively the MCG and the AG of the completely specified MCG.*

2. *Let $w = \varepsilon$, the empty initial input sequence.*

3. *Let $\phi(w) = \{M \mid M \in V\}$, the initial current marking uncertainty, where V is the set of nodes of $\mathcal{G}$.*

4. ***While*** *$\phi(w) \not\subseteq \mathcal{TM}(\bar{M})$,* ***do***

4.1. *pick two markings $M_i, M_j \in \phi(w)$ such that the two following conditions hold: i) $M_i \neq M_j$, ii) $M_i \notin \mathcal{TM}(\bar{M})$ or $M_j \notin \mathcal{TM}(\bar{M})$;*

4.2. *find a shortest path in $\mathcal{A}(\tilde{\mathcal{G}})$ from (M_i, M_j) to $(\bar{M}', \bar{M}'')$, where $\bar{M}', \bar{M}'' \in \mathcal{TM}(\bar{M})$.*

4.3. ***If*** *no such a path exists, stop the computation, there exists no PSS for $\bar{M}$.*

Else, *let w' be the input sequence along this path,* ***do***

4.3.1. $\phi(ww') = \{M' | \forall M \in \phi(w), M \xrightarrow{w} M'\}$;

4.3.2. $w = ww'$.

end if

end while ■

We point out that the above procedure is based on the standard algorithm for automata based on the construction of the auxiliary graph as presented in [14] and redefined for PNs in [21]. At each step the cardinality of the current state uncertainty decreases by at least one unit, thus it is ensured to halt in a finite number of steps. The complexity of this procedure is $O(n^3 + n^2 \times |E|)$, where n is the number of nodes of the MCG and $|E|$ denotes the cardinality of the input event set.

There are, however, two main differences with respect to [21], that do not modify the order of complexity of the algorithm. First, the current state uncertainty — used as halting criterion at step 4. — is not required to be singleton, but just to be included in the synchronization target marking set, according to Definition IV.2. Second, the algorithm searches the shortest path in the AG from node (M_i, M_j) to a node $(\bar{M}', \bar{M}'')$, where $\bar{M}'$ and $\bar{M}''$ may be different provided they belong to the set of target markings. M_i and M_j are selected from the current state uncertainty to be synchronized into the target but should not belong both to the target set, accordingly to conditions i) and ii) of step 4.1., otherwise the current state uncertainty would not change.

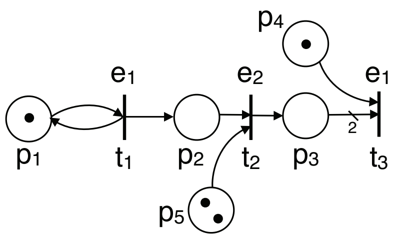
Example IV.6. *Consider again the synchronized PN in Fig. 2, its completely specified MCG (see Fig. 7a) and its AG (cf. Fig. 7b). Here $P_u = \{p_2\}$ and $P_u^\bullet = \{t_2\}$, where $f(t_2) = e_2$ (this net clearly satisfies Assumption III.10, since no other transition is associated with e_2). Given $\bar{M} = [0\,3\,1]^T$, since $P_b = \{p_1, p_3\}$ only node M_3 belongs to $\mathcal{TM}(\bar{M})$. A possible execution of Algorithm 2 is described by the following steps. Let the initial marking uncertainty be $\phi(\varepsilon) = \{M_0,\, M_1,\, M_2,\, M_3\}$. If at step 4.1 markings M_0 and M_1 are selected, path $(M_0, M_1) \xrightarrow{e_1 e_1 e_2} (M_3, M_3)$ will be obtained. The corresponding current state uncertainty is updated to $\phi(e_1 e_1 e_2) = \{M_3\}$, so that the computation ends returning $w = e_1 e_1 e_2$ as the searched PSS.* ■

B. Validation of a PSS

In this section we discuss the relation between SSs and PSSs.

We first show by means of an example that a PSS synchronizing the MCG to a target set $\mathcal{TM}(\bar{M})$ may fail to be a SS for the synchronized PN.

Example IV.7. *Consider the synchronized PN shown in Fig. 8 with its MCG, and let $\bar{M}$ =*



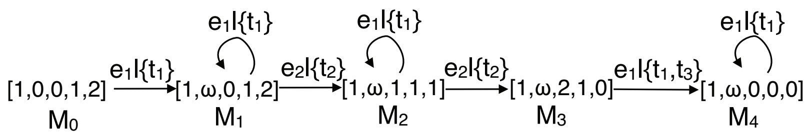


Fig. 8: Unbounded synchronized PN (a) and its MCG (b) in Example IV.7.

$[1\,0\,0\,0\,0]^T$ *be a target marking. The net satisfies Assumption III.10 and has set of bounded places* $P_b = \{p_1, p_3, p_4, p_5\}$, *hence the set of target marking is* $\mathcal{TM}(\bar{M}) = \{[1\,x\,0\,0\,0]^T \mid x \in \mathbb{N}_\omega\}$. *One can verify that in the MCG the input sequence* $w = e_1e_2e_2e_1$ *synchronizes to the marking* $M_4 = [1\,\omega\,0\,0\,0]^T \in \mathcal{TM}(\bar{M})$. *However, in the net from the initial marking this input sequence produces the evolution*

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 2 \end{bmatrix} [e_1|\{t_1\} > \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix} [e_2|\{t_2\} > \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} [e_2|\emptyset > \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} [e_1|\{t_1\} > \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

which does not yield a marking in the target set. ■

To characterize those PSSs that are also SSs, we need to introduce some additional definitions.

Definition IV.8. Given the completed MCG $\hat{\mathcal{G}}$ of a marked synchronized PN with input alphabet E, let $M_j \in \mathbb{N}_\omega^m$ be one of its nodes and let $w \in E^*$ be an input sequence. We define:

- $\sigma_{j,w} \in (2^T)^*$ the step sequence produced in $\hat{\mathcal{G}}$ by the input sequence w starting from M_j, i.e., the sequence of steps such that:

$$M_j \xrightarrow{w|\sigma_{j,w}} .$$

  Note that such a sequence exists and is unique.

- $M_j^{\min} \in \mathbb{N}^m$ the minimal marking in $cov(M_j)$. This marking is

$$M_j^{\min}(p) = \begin{cases} M_j(p) & \text{if } M_j(p) \neq \omega \\ 0 & \text{if } M_j(p) = \omega \end{cases}$$

i.e., it is obtained from M_j setting to 0 all its ω components.

■

Example IV.9. *Consider again the synchronized PN shown in Fig. 8 with its MCG*[5]*. Consider the marking* $M_1 = [1\,\omega\,0\,1\,2]^T$ *in the graph and let* $w = e_1e_2e_2e_1$*. Then* $\sigma_{1,w} = \{t_1\}\{t_2\}\{t_2\}\{t_1, t_3\}$ *and* $M_1^{\min} = [1\,0\,0\,1\,2]^T$*. If* $w' = e_2e_1e_2e_2$ *then* $\sigma_{1,w'} = \{t_2\}\{t_1\}\{t_2\}\emptyset$*.* ■

Based on these new concepts, we can derive a sufficient condition for a PSS to be a SS. Let us first present a lemma that will be used in the following derivation.

Lemma IV.10. *Consider an unbounded marked synchronized PN* $\langle N, M_0, E, f\rangle$*. Let* $\mathcal{G}$ *be its MCG and* $w \in E^*$ *be an input sequence. Given a node* $M_j \in \mathbb{N}_\omega^m$ *in the graph, let* $\sigma_{j,w}$ *be the sequence described in Definition IV.8. Consider a marking* $M \in cov(M_j)$*. Then*[6]

$$M[w|\sigma_{j,w}\rangle \qquad \Longrightarrow \qquad (\forall M' \in cov(M_j),\ M' \geq M)\ M'[w|\sigma_{j,w}\rangle.$$

Proof. We prove this by contradiction. Let $\sigma_{j,w} = \tau_1\tau_2\cdots\tau_k$. Assume that input sequence w produces from M' the evolution $M'[w|\sigma'\rangle$ with $\sigma' = \tau_1'\tau_2'\cdots\tau_k'$. If $\sigma' \neq \sigma_{j,w}$ then let r be the smallest index in $\{1, 2, \ldots, k\}$ such that $\tau_r' \neq \tau_r$. Since $M' \geq M$ then $\tau_r' \supsetneq \tau_r$. However, since $M' \in cov(M_j)$ and $M_j[w|\sigma_{j,w}\rangle$ it also follows that $\tau_r' \subset \tau_r$, clearly a contradiction. □ □

This lemma states a simple monotonicity property. In plain words, if starting from a reachable marking $M \in cov(M_j)$ an input sequence w produces a firing step sequence $\sigma_{j,w}$ (the same that is produced in the MCG from M_j) then starting from any other marking greater than M and still in $cov(M_j)$ input sequence w will also produce firing step sequence $\sigma_{j,w}$.

[5]The completed graph is not shown in figure for sake of simplicity, but can be easily obtained adding selfloop labeled $e|\emptyset$ as discussed in the previous subsection.

[6]Here $M[w|\sigma_{j,w}\rangle$ denotes that in the synchronized PN starting from marking M the input sequence w determines the firing of the step sequence $\sigma_{j,w}$.

Proposition IV.11. *Consider an unbounded marked synchronized PN $\langle N, M_0, E, f\rangle$ satisfying Assumption III.10. Let $\mathcal{G}$ be its MCG with set of nodes V. Given an input sequence $w \in E^*$, for each node $M_j \in V$ let $\sigma_{j,w}$ and $M_j^{\min}$ be the step sequence and marking described in Definition IV.8. It holds:*

$$w \in \mathcal{PSS}(\mathcal{G}, \bar{M}) \wedge (\forall M_j \in V)\ M_j^{\min}[w|\sigma_{j,w}\rangle \quad \Longrightarrow \quad w \in \mathcal{SS}(N, M_0, \bar{M}). \tag{4}$$

Proof. First we recall an elementary property of the coverability graph, that holds for both PNs and synchronized PNs. If in the coverability graph of a PN (resp., MCG of a synchronized PN) the firing of a sequence of transitions (resp. of transition steps) σ yields from a marking M_ω a marking $\bar{M}_\omega$, then the firing of the same sequence — assuming it is possible — in the net starting from any marking $M \in cov(M_\omega)$ yields a marking that coincides with $\bar{M}_\omega$ for all components associated to bounded places. Thus if the condition in eq. (4) holds, by Lemma IV.10 it follows that from any marking in $CS(N, M_0)$ the input sequence w yields a marking in the target set $\mathcal{TM}(\bar{M})$. Finally, since the net satisfies Assumption III.10, we know by Proposition III.11 that $R(N, M_0) \subseteq CS(N, M_0)$ which concludes the proof. □ □

Example IV.12. *Consider again the synchronized PN shown in Fig. 8 with its MCG. Consider the input sequence $w = e_1e_1e_2e_2e_1$. One can verify that this input sequence is a PSS for $\bar{M} = [1\,0\,0\,0\,0]^T$. We want to check if the sequence satisfies the sufficient condition in Proposition IV.11.*

We observe that

$$\begin{array}{lll}
M_0 = [1\,0\,0\,1\,2]^T, & \sigma_{0,w} = \{t_1\}\{t_1\}\{t_2\}\{t_2\}\{t_1,t_3\}, & M_0^{\min} = [1\,0\,0\,1\,2]^T \\
M_1 = [1\,\omega\,0\,1\,2]^T, & \sigma_{1,w} = \{t_1\}\{t_1\}\{t_2\}\{t_2\}\{t_1,t_3\}, & M_1^{\min} = [1\,0\,0\,1\,2]^T \\
M_2 = [1\,\omega\,1\,1\,1]^T, & \sigma_{2,w} = \{t_1\}\{t_1\}\{t_2\}\emptyset\{t_1,t_3\}, & M_2^{\min} = [1\,0\,1\,1\,1]^T \\
M_3 = [1\,\omega\,2\,1\,0]^T, & \sigma_{3,w} = \{t_1,t_3\}\{t_1\}\emptyset\emptyset\{t_1\}, & M_3^{\min} = [1\,0\,2\,1\,0]^T \\
M_4 = [1\,\omega\,0\,0\,0]^T, & \sigma_{4,w} = \{t_1\}\{t_1\}\emptyset\emptyset\{t_1\}, & M_4^{\min} = [1\,0\,0\,0\,0]^T
\end{array}$$

Now one can readily verify that in the synchronized Petri net, for $j = 0, 1, 2, 3, 4$, it holds $M_j^{\min}[w|\sigma_{j,w}^{\min}\rangle$ where $\sigma_{j,w}^{\min} = \sigma_{j,w}$ hence the PSS w is a SS. ■

If a PSS does not satisfy Proposition IV.11, one should look for different PSSs possibly of increasing length. We note, however, that the testing of the condition in eq. (4) may provide some intuition on why the PSS fails to be a SS and possibly how it should be modified to obtain a SS. We do not provide a formal procedure to do this, but discuss such a case in the next example.

Example IV.13. *Consider again the synchronized PN shown in Fig. 8 with its MCG. Consider the PSS* $w = e_1e_2e_2e_1$ *for* $\bar{M} = [1\,0\,0\,0\,0]^T$ *that, as discussed in Example 8, is not a SS. We observe that it holds:*

$$
\begin{array}{lll}
M_0 = [1\,0\,0\,1\,2]^T, & \sigma_{0,w} = \{t_1\}\{t_2\}\{t_2\}\{t_1,t_3\}, & M_0^{\min} = [1\,0\,0\,1\,2]^T \\
M_1 = [1\,\omega\,0\,1\,2]^T, & \sigma_{1,w} = \{t_1\}\{t_2\}\{t_2\}\{t_1,t_3\}, & M_1^{\min} = [1\,0\,0\,1\,2]^T \\
M_2 = [1\,\omega\,1\,1\,1]^T, & \sigma_{2,w} = \{t_1\}\{t_2\}\emptyset\{t_1,t_3\}, & M_2^{\min} = [1\,0\,1\,1\,1]^T \\
M_3 = [1\,\omega\,2\,1\,0]^T, & \sigma_{3,w} = \{t_1,t_3\}\emptyset\emptyset\{t_1\}, & M_3^{\min} = [1\,0\,2\,1\,0]^T \\
M_4 = [1\,\omega\,0\,0\,0]^T, & \sigma_{4,w} = \{t_1\}\emptyset\emptyset\{t_1\}, & M_4^{\min} = [1\,0\,0\,0\,0]^T
\end{array}
$$

Now one can readily verify that:

$$
\begin{array}{lll}
M_0^{\min}[w|\sigma_{0,w}^{\min}\rangle & \textit{with} & \sigma_{0,w}^{\min} = \{t_1\}\{t_2\}\emptyset\{t_1\} \neq \sigma_{0,w}, \\
M_1^{\min}[w|\sigma_{1,w}^{\min}\rangle & \textit{with} & \sigma_{1,w}^{\min} = \{t_1\}\{t_2\}\emptyset\{t_1\} \neq \sigma_{1,w}, \\
M_2^{\min}[w|\sigma_{2,w}^{\min}\rangle & \textit{with} & \sigma_{2,w}^{\min} = \{t_1\}\{t_2\}\emptyset\{t_1,t_3\} = \sigma_{2,w}, \\
M_3^{\min}[w|\sigma_{3,w}^{\min}\rangle & \textit{with} & \sigma_{3,w}^{\min} = \{t_1\}\emptyset\emptyset\{t_1,t_3\} \neq \sigma_{3,w}, \\
M_4^{\min}[w|\sigma_{4,w}^{\min}\rangle & \textit{with} & \sigma_{4,w}^{\min} = \{t_1\}\emptyset\emptyset\{t_1\} = \sigma_{4,w},
\end{array}
$$

This difference between $\sigma_{0,w}^{\min}$ *and* $\sigma_{0,w}$ *is due to a lacking token in* p_2 *after the evolution* $e_1e_2|\{t_1\}\{t_2\}$*. This lacking token can be produced by previously firing the repetitive sequence* e_1 *twice (as opposed to once). This leads to the longer input sequences* $w' = e_1e_2e_1e_2e_1$ *or equivalently* $w'' = e_1e_1e_2e_2e_1$ *which can both be shown to be SSs (the latter was studied in the previous example).* ■

V. Conclusion

The problem of determining synchronizing sequences for unbounded systems is here investigated. We consider deterministic synchronized PNs, for which we first propose an algorithmic construction of a finite coverability graph that describes its behavior. Unfortunately we show that this graph does not always cover all evolutions of the net, due to the presence of vanishing steps and vanishing markings. A condition to rule out this undesirable situation, called Assumption III.10, is introduced.

Second, we propose a simple approach, that builds on the results we have previously derived for bounded nets, to compute synchronizing sequences for unbounded nets. The idea is to use the coverability graph to compute potentially synchronizing sequences and then to test if a sufficient condition for such a sequence for being a synchronizing one is verified.

We point out that Assumption III.10 is only a sufficient (but not necessary) condition to rule out the existence of vanishing steps and markings. For this reasons, the procedure presented in this paper can be used with a larger class of synchronized nets, although currently we lack a general characterization of this class and will address this problem in future works.

REFERENCES

[1] Ananichev, D., Volkov, M.V.: Synchronizing monotonic automata. Lecture Notes in Computer Science pp. 111–121 (2003)

[2] Benenson, Y., Adar, R., Paz-Elizur, T., Livneh, Z., Shapiro, E.: Dna molecule provides a computing machine with both data and fuel. Proc. National Acad. Sci. USA **100**(5), 2191–2196 (2003). DOI doi:10.1073/pnas.0535624100

[3] Benenson, Y., Paz-Elizur, T., Adar, R., Keinan, E., Livneh, Z., Shapiro, E.: Programmable and autonomous computing machine made of biomolecules. Nature **414**, 430 – 434 (2001). DOI doi:10.1038/35106533

[4] Cheng, K.T., Agrawal, V.: Initializability consideration in sequential machine synthesis. IEEE Transactions on Computers **41**, 374–379 (1992). DOI http://doi.ieeecomputersociety.org/10.1109/12.127453

[5] Cho, H., Jeong, S.W., Somenzi, F., Pixley, C.: Multiple observation time single reference test generation using synchronizing sequences. In: Design Automation, 1993, with the European Event in ASIC Design. Proceedings. 4th European Conference on, pp. 494 –498 (1993). DOI 10.1109/EDAC.1993.386426

[6] David, R., Alla, H.: Discrete, Continuous and Hybrid Petri Nets. Springer-Verlag (2004)

[7] Devillers, R., Van Begin, L.: Boundedness undecidability for synchronized nets. Information Processing Letters **99**(5), 208 – 214 (2006). DOI 10.1016/j.ipl.2006.04.009

[8] Ding, Z., Jiang, C., Zhou, M.: Deadlock checking for one-place unbounded Petri nets based on modified reachability trees. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **38**(3), 881 –883 (2008). DOI 10.1109/TSMCB.2008.917177

[9] Finkel, A.: The minimal coverability graph for Petri nets. In: Papers from the 12th International Conference on Applications and Theory of Petri Nets: Advances in Petri Nets 1993, pp. 210–243. Springer-Verlag, London, UK, UK (1993)

[10] Geeraerts, G., Raskin, J.F., Van Begin, L.: On the efficient computation of the minimal coverability set for Petri nets. International Journal of Foundations of Computer Science **21**(02), 135–165 (2010). DOI 10.1142/S0129054110007180

[11] Jürgensen, H.: Synchronization. Information and Computation **206**, 1033–1044 (2008). DOI 10.1016/j.ic.2008.03.005. URL http://dl.acm.org/citation.cfm?id=1410483.1410
835

[12] Kari, J.: Synchronization and stability of finite automata. Journal of Universal Computer Science **8**(2), 270–277 (2002)

[13] Karp, R.M., Miller, R.E.: Parallel program schemata. Journal of Computer and System Sciences **3**(2), 147 – 195 (1969). DOI 10.1016/S0022-0000(69)80011-5

[14] Lee, D., Yannakakis, M.: Principles and methods of testing finite state machine – a survey. Proceedings of the IEEE **84**(8), 1090–1123 (1996)

[15] Mayr, E.W., Meyer, A.R.: The complexity of the finite containment problem for Petri nets. Journal of the ACM **28**(3), 561–576 (1981). DOI 10.1145/322261.322271

[16] Moore, E.F.: Gedanken-experiments on sequential machines. Automata Studies, Annals of Mathematical Studies **34**, 129 – 153 (1956)

[17] Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE **77**(4), 541 –580 (1989). DOI 10.1109/5.24143

[18] Natarajan, B.K.: An algorithmic approach to the automated design of parts orienters. In: SFCS '86: Proc. of the 27th Annual Symposium on Foundations of Computer Science, pp. 132–142. IEEE Computer Society, Washington, DC, USA (1986). DOI http://dx.doi.org/10.1109/SFCS.1986.5

[19] Pocci, M.: Test and diagnosis of discrete event systems using Petri nets. Ph.D. thesis, Aix-Marseille University & University of Cagliari (2013). URL http://www.lsis.org/poccim/phd_thesis_marco_pocci.pdf

[20] Pocci, M., Demongodin, I., Giambiasi, N., Giua, A.: A new algorithm to compute synchronizing sequences for synchronized Petri nets. In: IEEE Tencon Conference for Region 10 (TENCON). Xi'an, Shaanxi, China (2013)

[21] Pocci, M., Demongodin, I., Giambiasi, N., Giua, A.: Testing experiments on synchronized Petri nets. Automation Science and Engineering, IEEE Transactions on **11**(1), 125–138 (2014). DOI 10.1109/TASE.2013.2290774

[22] Pocci, M., Demongodin, I., Giambiasi, N., Giua, A.: Testing experiments on unbounded systems: synchronizing sequences using Petri nets. In: WODES14: 12th Int. Workshop on Discrete Event Systems. Cachan, France (2014)

[23] Reynier, P.A., Servais, F.: Minimal coverability set for Petri nets: Karp and miller algorithm with pruning. In: Proceedings of the 32nd international conference on Applications and theory of Petri Nets, PETRI NETS'11, pp. 69–88. Springer-Verlag, Berlin, Heidelberg (2011)

[24] Wang, S., Zhou, M., Li, Z., Wang, C.: A new modified reachability tree approach and its applications to unbounded Petri nets. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **43**(4), 932–940 (2013). DOI 10.1109/TSMCA.2012.2226878