Diagnosability Analysis of Unbounded Petri Nets

Maria Paola Cabasino, Alessandro Giua, Stéphane Lafortune, Carla Seatzu

Abstract

In this paper we consider the property of diagnosability for labeled unbounded Petri nets, namely Petri nets where the number of tokens in one or more places can grow indefinitely. We give necessary and sufficient conditions for diagnosability and we present a test to study diagnosability based on the analysis of the coverability graph of a particular net, called *verifier net*, that is built starting from the initial system. To the best of our knowledge, this is the first available test for diagnosability analysis of labeled unbounded Petri nets. We also discuss existing methods to perform diagnosis of unbounded Petri nets.

Published as:

M.P. Cabasino, A. Giua, S. Lafortune, C. Seatzu, "Diagnosability analysis of unbounded Petri nets," CDC09: 48th IEEE Conf. on Decision and Control (Shanghai, China), Dec. 2009.

This work has been partially supported by the European Community's Seventh Framework Programme under project DISC (Grant Agreement n. INFSO-ICT-224498) and by the US National Science Foundation (Grant ECCS-0624281).

M.P. Cabasino and A. Giua and C. Seatzu are with the Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy {cabasino,giua,seatzu}@diee.unica.it

S. Lafortune is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Mi 48109-2122, USA stephane@eecs.umich.edu

I. INTRODUCTION

Failure detection and diagnosis in industrial systems is a subject that has received a lot of attention in the past few decades. Within this body of research two different problems are addressed: diagnosis and diagnosability. Solving a problem of diagnosis means that we associate to each observed string of events a diagnosis state, such as "normal" or "faulty" or "uncertain". Solving a problem of diagnosability is equivalent to determining if the system is diagnosable, i.e., if once a fault has occurred the system can detect its occurrence in a finite number of steps.

In the framework of finite state systems, such as automata and bounded Petri nets, several approaches have been proposed for diagnosis and a restricted number for diagnosability. In particular, in [1], [2] Sampath *et al.* have presented an approach to perform diagnosis and give necessary and sufficient conditions for diagnosability based on the analysis of indeterminate cycles in an automaton called *diagnoser*. Some of us have presented an approach to perform diagnosis of Petri nets [3] and a procedure to test diagnosability of *bounded* Petri nets [4], both based on the notion of basis markings.

In the case of infinite state systems, very few results have been presented, all in the framework of Petri nets. In [5] Ushio *et al.* give sufficient conditions for diagnosability of unbounded Petri nets extending the approach in [1], [2] for automata. In contrast to [5], Chung in [6] assumes that part of the transitions of the Petri net are observable and shows that the additional information from observed transitions in general enhances the diagnosability of the analyzed system. In [7] Wen and Jeng propose an approach to test diagnosability by checking the structural property of the T-invariants of the nets. In [8] Wen *et al.* present an algorithm, based on a linear programming problem and of polynomial complexity in the number of nodes, for computing a sufficient condition of diagnosability of discrete event systems modeled by Petri nets.

In this paper, we consider the diagnosability problem of unbounded Petri nets. The model we consider is a labeled Petri net where some transitions are indistinguishable. Faults are modeled by unobservable transitions, but not all unobservable transitions represent faults. We present necessary and sufficient conditions for diagnosability and we give a test based on the analysis of the coverability graph of a new type of net, called *verifier net*, built from the Petri net model of the system to be diagnosed. To the best of our knowledge, this is the first test for diagnosability analysis of labeled Petri nets with unbounded state space. This contribution is relevant because

it provides a diagnosability test for discrete event systems that are capable of generating some non-regular languages. In fact, as is well known, finite state automata, as well as bounded Petri nets, only generate regular languages while unbounded Petri nets can generate both regular and non-regular languages.

Note that our problem statement is related to prior works on diagnosability analysis of regular languages represented by finite-state automata and is different from prior work on diagnosability analysis of Petri nets. Specifically, we consider that only a subset of transitions are unobservable, while in [5], [7], [8] the authors consider that some places are unobservable as well. Moreover, we consider labeled Petri nets where two or more transitions can share the same label, rather than free-labeled Petri nets.

Let us finally observe that in [4] we present an original approach for diagnosability of *bounded* Petri nets. This approach is based on the notion of basis markings and allows to reduce the state space *enumeration*. Unfortunately, in the case of unbounded Petri nets, the basis marking approach cannot be used because in such a case we need an exhaustive enumeration of the nodes of the coverability graph. On the other hand, when applicable, the approach in [4] may be preferable to the approach in this paper because it also allows to solve the diagnosis problem, using the same framework we use for the diagnosability analysis.

II. Background on Petri nets

In this section we recall the formalism used in the paper. For more details on Petri nets we refer the reader to [9].

A *Place/Transition net* (P/T net) is a structure $N = (P, T, Pre, Post)$, where P is a set of m places; T is a set of n transitions; $Pre: P \times T \rightarrow \mathbb{N}$ and $Post: P \times T \rightarrow \mathbb{N}$ are the pre– and $post$– incidence functions that specify the arcs; $C = Post - Pre$ is the incidence matrix.

A *marking* is a vector $M: P \rightarrow \mathbb{N}$ that assigns to each place of a P/T net a non–negative integer number of tokens, represented by black dots. We denote $M(p)$ the marking of place p. A P/T *system* or *net system* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0.

A transition t is enabled at M iff $M \geq Pre(\cdot, t)$ and may fire yielding the marking $M' = M + C(\cdot, t)$. We write $M\ [\sigma\rangle$ to denote that the sequence of transitions $\sigma = t_{j_1} \cdots t_{j_k}$ is enabled at M, and we write $M\ [\sigma\rangle\ M'$ to denote that the firing of σ yields M'.

The set of all sequences that are enabled at the initial marking M_0 is denoted $L(N, M_0)$, i.e., $L(N, M_0) = \{\sigma \in T^* \mid M[\sigma\rangle\}$. We use λ to denote an empty sequence of transitions, i.e., $\sigma\lambda = \sigma \ \ \forall \ \ \sigma \in T^*$.

Given a sequence $\sigma \in T^*$, we call $\pi : T^* \to \mathbb{N}^n$ the function that associates to σ a vector $y \in \mathbb{N}^n$, named the *firing vector (or Parick vector)* of σ. Specifically, $y = \pi(\sigma)$ is such that $y(t) = k$ if the transition t is contained k times in σ.

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 $R(N, M_0)$. Finally, we denote $PR(N, M_0)$ the *potentially reachable set*, i.e., the set of all markings $M \in \mathbb{N}^m$ for which there exists a vector $y \in \mathbb{N}^n$ that satisfies the *state equation* $M = M_0 + C \cdot y$, i.e., $PR(N, M_0) = \{M \in \mathbb{N}^m \ \mid \ \exists\ y \in \mathbb{N}^n \ : \ M = M_0 + C \cdot y\}$. We have that $R(N, M_0) \subseteq PR(N, M_0)$.

A Petri net having no directed circuits is called *acyclic*. For this subclass, it can be shown [10] that the state equation gives necessary and sufficient conditions.

A net system $\langle N, M_0\rangle$ is *bounded* if there exists a positive constant k such that, for $M \in R(N, M_0)$, $M(p) \leq k$. If such is not the case, namely if the number of tokens in one or more places can grow indefinitely, then the Petri net system is *unbounded*.

***Definition** 2.1:* Given a Petri net system $\langle N, M_0\rangle$, a transition t is:

- *dead* if there does not exist a reachable marking $M \in R(N, M_0)$ that enables t;
- *semi-live* if there exists at least one marking $M \in R(N, M_0)$ that enables t;
- *live* if for each reachable marking $M \in R(N, M_0)$, t is semi-life in $\langle N, M\rangle$. ■

A net system $\langle N, M_0\rangle$ is *live* if all transitions $t \in T$ are live. A *deadlock* occurs at marking M if no transition is enabled at M.

***Definition** 2.2:* A sequence $\sigma \in T^*$ is called *repetitive* if there exists a marking $M_1 \in R(N, M_0)$ such that

$$M_1[\sigma\rangle M_2[\sigma\rangle M_3[\sigma\rangle \cdots \tag{1}$$

i.e., if it can fire infinitely often starting from M_1. It is possible to distinguish two different types of repetitive sequences:

- *stationary* sequence: if in (1) $M_i = M_{i+1}$ for all $i = 1, 2, \ldots$.
- *increasing* sequence: if in (1) $M_i \lneq M_{i+1}$ for all $i = 1, 2, \ldots$. ■

There exists a simple structural condition to characterize repetitive sequences.

***Fact** 2.3:* [9] If sequence σ is enabled at M_1, a necessary and sufficient condition for it being repetitive is that in (1) it holds $M_i \leq M_{i+1}$ for all $i = 1, 2, \ldots$, or equivalently $C \cdot y \geq \vec{0}$, where $y = \pi(\sigma)$.

Furthermore if $C \cdot y = \vec{0}$ the sequence is stationary, else if $C \cdot y \gneq \vec{0}$ it is increasing. ■

A nonnegative integer vector $y \in \mathbb{N}^n$ satisfying $C \cdot y = 0$ is called ***T-invariant***.

A *labeling function* $\mathcal{L} : T \to L \cup \{\varepsilon\}$ assigns to each transition $t \in T$ either a symbol from a given alphabet L or the empty string ε.

We denote as T_u the set of transitions whose label is ε, i.e., $T_u = \{t \in T \mid \mathcal{L}(t) = \varepsilon\}$. Transitions in T_u are called *unobservable* or *silent*.

We denote as T_o the set of transitions labeled with a symbol in L. Transitions in T_o are called *observable* because when they fire their label can be observed. Note that in this paper we assume that the same label $l \in L$ can be associated with more than one transition. In particular, two transitions $t_1, t_2 \in T_o$ are called *undistinguishable* if they share the same label, i.e., $\mathcal{L}(t_1) = \mathcal{L}(t_2) = l \in L$.

We extend the labeling function to define the *projection operator* $\mathcal{L} : T^* \to L^*$ recursively as follows:

(i) $\mathcal{L}(t_j) = l_j \quad \forall t_j \in T_o : \mathcal{L}(t_j) = l_j$;

(ii) $\mathcal{L}(t_i) = \varepsilon \quad \forall t_i \in T_u$;

(iii) $\mathcal{L}(\sigma t_j) = \mathcal{L}(\sigma)\mathcal{L}(t_j) \quad \forall \sigma \in T^*, \ \forall t_j \in T$.

Moreover, $\mathcal{L}(\lambda) = \varepsilon$. The *inverse projection operator* $\mathcal{L}^{-1}$ is defined as $\mathcal{L}^{-1}(d) = \{s \in L(N, M_0) : \mathcal{L}(s) = d\}$.

We denote as w the word of events associated with the sequence σ, i.e., $w = \mathcal{L}(\sigma)$. Note that the length of a sequence σ (denoted $|\sigma|$) is always greater than or equal to the length of the corresponding word w (denoted $|w|$). In fact, if σ contains k' transitions labeled ε then $|\sigma| = k' + |w|$.

Given a language $K \in T^*$, we denote by K/s the post-language of K after s, i.e., $K/s = \{g \in T^* \mid sg \in K\}$.

We conclude with the following definition:

***Definition** 2.4:* Given a net $N = (P, T, Pre, Post)$, and a subset $T' \subseteq T$ of its transitions, we define the *$T'-$induced subnet of* N as the new net $N' = (P, T', Pre', Post')$ where $Pre', Post'$

are the restrictions of $Pre, Post$ to T'. In this case, we write $N' \prec_{T'} N$. ■

The net N' can be thought of as obtained from N by removing all transitions in $T \setminus T'$ and all dangling arcs.

III. COVERABILITY GRAPH

One technique used for the analysis of unbounded Petri nets is based on the construction of the coverability tree/graph (see also [9]) that provides a description in finite terms of the infinite reachability set. In particular, each node of the graph 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. The symbol ω denotes that the marking of the corresponding place may grow indefinitely. Note that for all $n \in \mathbb{N}$ we have that $\omega > n$ and $\omega \pm n = \omega$.

***Algorithm** 3.1:* **Construction of the coverability tree for $\langle N, M_0 \rangle$.**

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**
   a) Select a node q tagged "new" and let M be its label.
   b) **For** all t enabled at M, i.e., such that $M \geq Pre(\cdot, t)$:
      i) Let $M' = M + C(\cdot, t)$ be the marking reached from M by firing t.
      ii) Let $\bar{q}$ be the first node met on the backward path from q to q_0 whose label is $\bar{M} \lneq M'$. **If** such a node exists **then** for all $p \in P$ such that $M'(p) > \bar{M}(p)$ let $M'(p) = \omega$.
      iii) Add a new node q' and label it M'.
      iv) Add an arc labeled t from q to q'.
      v) **If** there exists already in the tree a node with label M', **then** tag node q' "duplicate", **else** tag it "new".
   c) Untag node q.

■

From the coverability tree (CT) one can obtain the coverability graph (CG) by fusing duplicate nodes with the untagged node with the same label: one can always convert a CT in a graph and viz.

In the construction of the CT the existence of a sequence σ that leads from a marking $\bar{M}$ to a greater marking M' is identified at step $2.(b).ii$. The places that by the repeated firing of such

a sequence σ grow unbounded are denoted with a special symbol ω. Note that if $\bar{M}$ contains no ω places then σ is an increasing sequence. However, if $\bar{M}$ contains ω places we can only say that σ is increasing for all places p such that $\bar{M}(p) < \omega$: nothing can be said for the remaining places.

The coverability graph gives us only sufficient conditions for reachability. Let us consider a property of the CG that will be useful in the following.

***Proposition** 3.2:* [9] Let us consider a Petri net $\langle N, M_0 \rangle$ and its CG. If a transition t is in the CG then t is *semi-live*.

This means that if a transition appears in the CG then there will exist for sure a firing sequence that enables it.

IV. Diagnosability of Petri net systems

Assume that the set of transitions is partitioned as $T = T_o \cup T_u$, where T_o is the set of observable transitions, and T_u is the set of unobservable transitions. When an observable transition fires we observe its label, thus our observations consist in sequences of symbols in the alphabet L. The set of unobservable transitions is partitioned into two subsets, namely $T_u = T_f \cup T_{reg}$ where T_f includes all fault transitions (modeling anomalous or fault behavior), while T_{reg} includes all transitions relative to unobservable but regular events. The set T_f is further partitioned into r different subsets T_f^i, where $i = 1, \ldots, r$, that model the different fault classes. Let us introduce a definition for diagnosability of Petri nets inspired by the definition of diagnosability for languages introduced in [11].

***Definition** 4.1:* A Petri net system $\langle N, M_0 \rangle$ having no deadlock after the occurrence of transition $t_f \in T_f^i$, for $i = 1, \ldots, r$, is *diagnosable wrt the fault class* T_f^i if there do not exist two firing sequences σ_1 and $\sigma_2 \in T^*$ satisfying the following conditions:

- $\mathcal{L}(\sigma_1) = \mathcal{L}(\sigma_2)$,
- $\forall t_f \in T_f^i$, $\sigma_1 \in (T \setminus T_f^i)^*$,
- $\exists$ at least one $t_f \in T_f^i$ such that $t_f \in \sigma_2$,
- σ_2 is of "arbitrary length" (see [11]) after fault $t_f \in T_f^i$.

A Petri net system $\langle N, M_0 \rangle$ is said *diagnosable* if it is diagnosable wrt all fault classes. ■

The previous definition is the immediate Petri net counterpart of the definition of diagnosability commonly used in the framework of diagnosis with automata. It is important to point out,

however, that a different notion may be introduced. This is done in the following definition.

Let T' be a subset of T. We define $\Psi(T') = \{st' \in L(N, M_0) : t' \in T'\}$, i.e, the set of all firing sequences in $L(N, M_0)$ that end in a transition $t' \in T'$.

***Definition** 4.2:* [1] A Petri net system $\langle N, M_0\rangle$ is *diagnosable in K steps wrt the fault class T_f^i* if $\exists K \in \mathbb{N}$ such that

$$\begin{aligned} &\forall s \in \Psi(T_f^i), \quad \forall g \in L(N, M_0)/s \text{ with } |g| \geq K \\ &\Rightarrow \quad \forall w \in \mathcal{L}^{-1}(\mathcal{L}(sg)),\ \exists t_f \in T_f^i \ :\ t_f \in w. \end{aligned} \tag{2}$$

A Petri net system $\langle N, M_0\rangle$ is said *diagnosable in K steps* if it is diagnosable in K steps wrt all fault classes. ■

The above definition means that a Petri net system is diagnosable in K steps wrt the $i-$th fault class if for any sequence s that terminates in a transition in T_f^i and for any continuation g of s of length greater than or equal to K, all sequences w having the same observable projection of sg contain some fault transitions in T_f^i. In other words, diagnosability in K steps wrt a given fault class implies that the occurrence of a fault in that class can be detected after a finite number K of transition firings.

***Proposition** 4.3:* A Petri net system $\langle N, M_0\rangle$ that is *diagnosable in K steps wrt T_f^i* is also *diagnosable wrt T_f^i*.

If the language $L(N, M_0)$ is regular the converse also holds.

Proof: One can verify that Definition 4.1 is equivalent to

$$\begin{aligned} &\forall s \in \Psi(T_f^i), \quad \exists K_s \in \mathbb{N}, \quad \forall g \in L(N, M_0)/s \text{ with } |g| \geq K_s \\ &\Rightarrow \quad \forall w \in \mathcal{L}^{-1}(\mathcal{L}(sg)),\ \exists t_f \in T_f^i \ :\ t_f \in w, \end{aligned} \tag{3}$$

where the bound K_s depends on the particular string s. Definition 4.2 is obviously stronger because it requires that the bound K should be the same for all strings $s \in \Psi(T_f^i)$; hence the first part of the statement holds.

Assume now that $L(N, M_0)$ is a regular language, hence can be generated by a finite-state automaton with transition function $\delta : X \times L \to X$ and initial state x_0. It is not difficult to understand that if $\langle N, M_0\rangle$ is diagnosable wrt T_f^i, for any two strings $s, s' \in \Psi(T_f^i)$ with $\delta(x_0, s) = \delta(x_0, s')$ one may choose the same integer K_s in (3), i.e., K_s actually depends on the

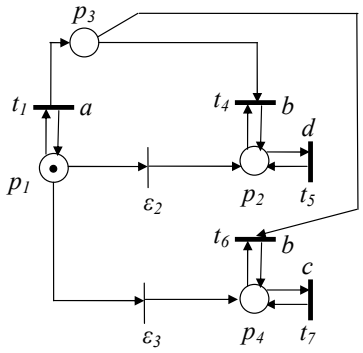

Fig. 1. The Petri net system of Example 4.4.

state $\delta(x_0, s)$. Since there are a finite number of states, by taking the largest K over all states reached by strings in $\Psi(T_f^i)$ we conclude that $\langle N, M_0\rangle$ is diagnosable in K steps wrt T_f^i. □

The second part of the previous statement also shows that in the case of regular languages (as it is the case with the diagnostic approach based on automata) it is not necessary to distinguish between the two notions of diagnosability. This result was also observed in [12].

The next example shows an unbounded net that is diagnosable but not diagnosable in K steps.

***Example** 4.4:* Let us consider the Petri net system in Fig. 1, where $T_o = \{t_1, t_4, t_5, t_6, t_7\}$, $T_u = \{\varepsilon_2, \varepsilon_3\}$ and $T_f = \{\varepsilon_2\}$. Let $\mathcal{L}(t_1) = a$, $\mathcal{L}(t_4) = \mathcal{L}(t_6) = b$, $\mathcal{L}(t_5) = d$ and $\mathcal{L}(t_7) = c$.

For all strings $s(k) \in \Psi(T_f)$, where $s(k) = t_1^k\varepsilon_2$, in (3) one may choose $K_{s(k)} = k + 2$ or greater to prove that the system is diagnosable. Since this value of $K_{s(k)}$, however, grows arbitrarily large with k, the system is not diagnosable in K steps for any finite K.

Note that if we modify the label of transition t_5 as $\mathcal{L}(t_5) = c$ then this system is not diagnosable with respect to either definition of diagnosability. ■

In the following section we investigate the problem of providing necessary and sufficient conditions for diagnosability (in the sense of Definition 4.1) of unbounded Petri nets under the following assumptions.

(A1) The structure of the net and its initial marking M_0 is known.

(A2) Two or more observable transitions can share the same label.

(A3) The T_u-induced subnet is acyclic.

(A4) The system does not enter a deadlock after the firing of any fault transition.

Since our diagnosis procedure is based on observing transition labels, assumption A3 is necessary to avoid cycles of unobservable transitions On the other hand, assumption A4, has been made for the sake of simplicity and could be removed with slight modifications to the procedure.

The problem of deriving analysis criteria for diagnosability in K steps is not addressed in this paper.

V. Analysis of Diagnosability

In this section we show how the diagnosability of an unbounded Petri net system can be checked by looking at the CG of a special Petri net called *Verifier Net* (VN).

A. Verifier Net

For the sake of simplicity, we consider in the following the case of a single fault class; hence, the superscript i is omitted in T_f^i hereafter.

Let us consider the labeled Petri net system $\langle N, M_0\rangle$, where $N = (P, T, Pre, Post)$, $T = T_o \cup T_u$, and $T_u = T_{reg} \cup T_f$. Let $\mathcal{L} : T \to L \cup \{\varepsilon\}$ be its labeling function.

Let $N' = (P', T', Pre', Post')$ be its T'-induced subnet, where $T' = T \setminus T_f = T_o \cup T_{reg}$.

Since we need to distinguish among places of N and N' we denote them as P and P', respectively, and assume that they are disjoint even if they represent the same places. Analogously, since we need to distinguish among regular transitions of N and N' we denote them as T_{reg} and T'_{reg}, respectively, and assume that they are disjoint even if they represent the same transitions.

We assume that the Petri net system associated with N' is $\langle N', M'_0\rangle$ where $M'_0 = M_0$. Finally, we assume that the labeling function of N' is equal to $\mathcal{L}$ restricted to T'.

The *Verifier Net* (VN) system is the Petri net system obtained by a composition (related to parallel composition) of $\langle N, M_0\rangle$ and $\langle N', M'_0\rangle$ assuming that the synchronization is performed on the observable transitions labels. We denote it as $\langle \tilde{N}, \tilde{M}_0\rangle$, where $\tilde{N} = (\tilde{P}, \tilde{T}, \tilde{Pre}, \tilde{Post})$, $\tilde{P} = P' \cup P$ and $\tilde{T} = (T'_o \times T_o) \cup (T'_{reg} \times \{\lambda\}) \cup (\{\lambda\} \times T_{reg}) \cup (\{\lambda\} \times T_f)$.

The algorithm below shows how to construct the two matrices $\tilde{Pre}$ and $\tilde{Post}$.

***Algorithm** 5.1:* **Construction of the Verifier Net.**

Input: a labeled Petri net system $\langle N, M_0\rangle$ where $N = (P, T, Pre, Post)$, $T = T_o \cup T_{reg} \cup T_f$ and $\mathcal{L} : T \to L \cup \{\varepsilon\}$.

Output: the VN system $\langle \tilde{N}, \tilde{M}_0 \rangle$, where $\tilde{N} = (\tilde{P}, \tilde{T}, \tilde{Pre}, \tilde{Post})$.

1) Let $\langle N', M'_0 \rangle$ be a labeled Petri net system defined as discussed above.
2) Let $\tilde{P} = P' \cup P$.
3) Let $\tilde{M}_0 = \begin{bmatrix} M'_0 \\ M_0 \end{bmatrix}$.
4) For all transitions $t_f \in T_f$,
   - add a transition $t \in \tilde{T}$ denoted as (λ, t_f);
   - for all $p \in P'$, let $\tilde{Pre}(p,t) = \tilde{Post}(p,t) = 0$;
   - for all $p \in P$, let $\tilde{Pre}(p,t) = Pre(p,t_f)$ and $\tilde{Post}(p,t) = Post(p,t_f)$.
5) For all transitions $t_{reg} \in T_{reg}$,
   - add a transition $t \in \tilde{T}$ denoted as (λ, t_{reg});
   - for all $p \in P'$, let $\tilde{Pre}(p,t) = \tilde{Post}(p,t) = 0$;
   - for all $p \in P$, let $\tilde{Pre}(p,t) = Pre(p,t_{reg})$ and $\tilde{Post}(p,t) = Post(p,t_{reg})$.
6) For all transitions $t'_{reg} \in T'_{reg}$,
   - add a transition $t \in \tilde{T}$ denoted as (t'_{reg}, λ);
   - for all $p \in P'$, let $\tilde{Pre}(p,t) = Pre'(p,t'_{reg})$ and $\tilde{Post}(p,t) = Post'(p,t'_{reg})$;
   - for all $p \in P$, let $\tilde{Pre}(p,t) = \tilde{Post}(p,t) = 0$.
7) For all labels $l \in L$,
   - for any couple t'_o, t_o with $t'_o \in T_o$, $t_o \in T_o$, $\mathcal{L}(t'_o) = \mathcal{L}(t_o) = l$,
     - add a transition $t \in \tilde{T}$ denoted as (t'_o, t_o);
     - for all $p \in P'$, let $\tilde{Pre}(p,t) = Pre'(p,t'_o)$ and $\tilde{Post}(p,t) = Post'(p,t'_o)$;
     - for all $p \in P$, let $\tilde{Pre}(p,t) = Pre(p,t_o)$ and $\tilde{Post}(p,t) = Post(p,t_o)$;
     - label transition t with (l,l).

■

***Example** 5.2:* Fig. 2 shows the VN of the Petri net system in Fig. 1, already introduced in Example 4.4.

The set of places of the VN is composed by the union of the set of places P of the Petri net system $\langle N, M_0 \rangle$ in Fig. 1 and the set of places P' of the T'-induced subnet. The T'-induced subnet is obtained from $\langle N, M_0 \rangle$ by removing fault transition ε_2; it is not drawn here.

Observable transitions, denoted in Fig. 2 as black bars, are indicated by two pairs (l,l) and

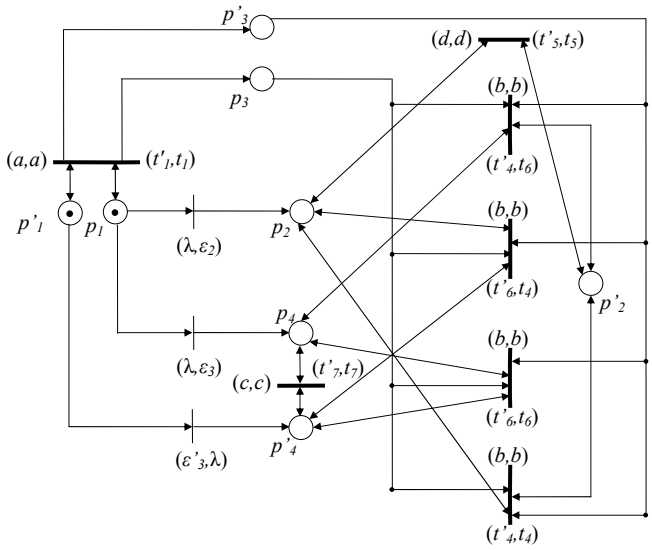

Fig. 2. Verifier net $\langle \tilde{N}, \tilde{M}_0 \rangle$ where $\langle N, M_0 \rangle$ is the Petri net in Fig. 1.

(t'_o, t_o) (e.g. $(a, a,)$ (t'_1, t_1)), while unobservable transitions are indicated by only one pair (e.g. (λ, ε_3)), since no label is associated with them.

Let us observe that since label b is associated with two transitions (t_4 and t_6) the VN contains four transitions labeled (b, b).

Note that, to improve readability, if a place p has both a pre and a post arc with a transition t we use a double arrow (e.g. line between p_1 and (t'_1, t_1)). ■

***Proposition** 5.3:* Given a Petri net system $\langle N, M_0 \rangle$ and its VN, if a sequence $\tilde{\sigma} = (t'_{i_1}, t_{i_1})(t'_{i_2}, t_{i_2}) \dots (t'_{i_k}, t_{i_k})$ is repetitive in the VN, then there exists a repetitive sequence $\sigma = t_{i_1} t_{i_2} \dots t_{i_k}$ in $\langle N, M_0 \rangle$ and a repetitive sequence $\sigma' = t'_{i_1} t'_{i_2} \dots t'_{i_k}$ in $\langle N', M'_0 \rangle$.

Proof: It follows from the definition of VN. In fact, the existence of a sequence $\tilde{\sigma} \in L(\tilde{N}, \tilde{M}_0)$ implies that $\mathcal{L}(\tilde{\sigma}) \in L(N, M_0)$ and $\mathcal{L}'(\tilde{\sigma}) \in L(N', M'_0)$, where $\mathcal{L}((t'_{i_1}, t_{i_1})(t'_{i_2}, t_{i_2}) \dots (t'_{i_k}, t_{i_k})) = t_{i_1} t_{i_2} \dots t_{i_k} = \sigma$ and $\mathcal{L}'((t'_{i_1}, t_{i_1})(t'_{i_2}, t_{i_2}) \dots (t'_{i_k}, t_{i_k})) = t'_{i_1} t'_{i_2} \dots t'_{i_k} = \sigma'$. The firing sequences σ and σ' are repetitive respectively in $\langle N, M_0 \rangle$ and in $\langle N', M'_0 \rangle$, given that $\tilde{\sigma}$ is repetitive in the VN. □

B. Necessary and sufficient conditions for diagnosability

The following theorem shows how to determine the diagnosability of a Petri net system, starting from the CG of its VN.

***Theorem** 5.4:* A Petri net system $\langle N, M_0 \rangle$ satisfying assumptions A1 to A4 is *diagnosable* iff starting from any node of the CG of its VN reached by firing the fault there does not exist any cycle associated with a repetitive sequence in the VN.

Proof: We prove the *if* and *only if* statements separately.

(Necessity) By contradiction, assume that in the CG of the VN there exists a cycle associated with a repetitive sequence for the VN and enabled after the fault occurrence. From Propositions 3.2 and 5.3, this means that in the Petri net system $\langle N, M_0 \rangle$ there exist two firing sequences $s = \sigma_p(r)^q$ and $s' = \sigma'_p(r')^q$ with $q \in \mathbb{N}$, such that: σ_p contains the fault and σ'_p does not, $\mathcal{L}(\sigma_p) = \mathcal{L}(\sigma'_p)$, r and r' are two repetitive sequences and $\mathcal{L}(r) = \mathcal{L}(r')$. Thus there exist in $L(N, M_0)$ two sequences s and s', one containing the fault and the other one not, having the same observable projection, that can be made arbitrarily long using Definition 2.2. This violates the definition of diagnosability of $L(N, M_0)$ given in Definition 4.1, hence the Petri net is not diagnosable.

(Sufficiency) We show that if the CG of the VN does not contain a cycle associated with a repetitive sequence in the VN after the fault event has occurred, then the system is diagnosable. Let us consider what happens after an occurrence of the fault event in the system. Since assumptions A3 and A4 hold, this occurrence will be captured in the CG and consequently in the VN. In this case, if we consider two strings of events $s = \sigma_p$ and $s' = \sigma'_p$ such that s contains the fault and s' does not, $\mathcal{L}(s) = \mathcal{L}(s')$, and attempt to extend these two strings in a manner that keeps their projections identical, the absence of a repetitive sequence in the VN after the said occurrence of the fault event will prevent this extension from growing arbitrarily long. Namely, we are unable to construct σ_1 and σ_2, as characterized in Definition 4.1. This means that there is no violation of diagnosability. □

Note that our procedure, not only gives necessary and sufficient conditions for the diagnosability of unbounded Petri nets, but also offers a test for diagnosability. After the verification of the existence of a cycle that is enabled after a fault in the CG, it is necessary to verify if that cycle is associated with a repetitive sequence in the VN. By Fact 2.3 it is possible to verify if

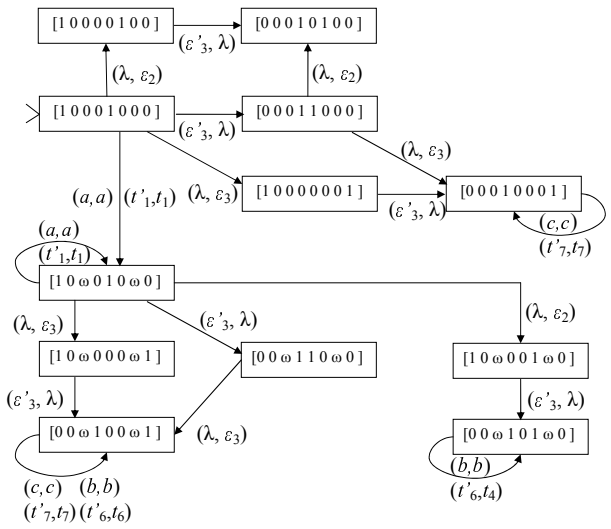

Fig. 3. Coverability graph of the Verifier Net in Fig. 2.

these sequences are repetitive by solving a simple integer programming problem.

***Example** 5.5:* In Fig. 3 is shown the CG of the VN in Fig. 2.

The Petri net system $\langle N, M_0 \rangle$ in Fig. 1 is diagnosable. In fact, looking at the CG of the VN there is only one cycle ($[0\ 0\ \omega\ 1\ 0\ 1\ \omega\ 0]^T \xrightarrow{(b,b),(t'_6,t_4)} [0\ 0\ \omega\ 1\ 0\ 1\ \omega\ 0]^T$) enabled by a path starting from a node reached firing the fault transition ε_2. However, this cycle is not associated with any repetitive sequence, since (t'_6, t_4) is not a repetitive sequence for the VN. ■

Note that the VN technique can only handle one fault class at a time. In the case of more than one fault classes, we need to built a VN for each fault class i for which all faults belonging to a fault class $j \neq i$ will be considered as regular transitions.

VI. DIAGNOSIS

In the previous section we considered the problem of diagnosability of unbounded Petri nets. Unfortunately, this approach does not allow us to solve the problem of diagnosis because the CG gives only sufficient conditions for reachability. We already proposed solutions to this problem in [13], [14] that are briefly recalled in the following. Note that we presented these methods for bounded Petri nets but they still hold for unbounded Petri nets.

The approach in [13] requires an exhaustive enumeration of the reachable markings each time an event is observed. To each reachable marking M is associated a vector with cardinality r,

where r is the number of fault classes. The i-th entry of this vector is equal to 1 if reaching M one or more fault transitions belonging to the i-th fault class have occurred, 0 otherwise.

In [14] we give a method to perform diagnosis using the notion of basis marking and justification. Given an observed word w, a basis marking M_b is a marking that is reached firing w and all those unobservable transitions strictly necessary to enable w. A justification is the minimal firing sequence of unobservable transitions that interleaved with w enables its firing. The notion of basis marking allows us to reduce the reachability space; in fact each time an observable transition fires we do not have to enumerate all the markings consistent with the observation but only a subset of them. Each time an observable transition fires, a diagnosis state is computed based on the set of pairs reached basis marking, corresponding justification.

VII. Conclusion

The main result of this paper is to present a necessary and sufficient condition for diagnosability of languages modeled by unbounded Petri nets. While the state space of such Petri nets is infinite, the property of diagnosability (as defined in Definition 4.1) can be completely characterized on a finite structure, called the *Verifier Net*. The notion of repetitive sequences is key in that regard.

References

[1] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. Automatic Control*, vol. 40 (9), pp. 1555–1575, 1995.

[2] ——, "Failure diagnosis using discrete-event models," *IEEE Trans. Control Systems Technology*, vol. 4, no. 2, pp. 105–124, 1996.

[3] A. Giua and C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," in *Proc. 44th IEEE Conf. on Decision and Control*, Dec. 2005, pp. 6323–6328.

[4] M. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of bounded Petri nets," in *Proc. 48th IEEE Conf. on Decision and Control*, Dec. 2009.

[5] T. Ushio, L. Onishi, and K. Okuda, "Fault detection based on Petri net models with faulty behaviors," in *Proc. SMC'98: IEEE Int. Conf. on Systems, Man, and Cybernetics (San Diego, CA, USA)*, Oct. 1998, pp. 113–118.

[6] S. Chung, "Diagnosing pn-based models with partial observable transitions," *International Journal of Computer Integrated Manufacturing*, vol. 12 (2), pp. 158–169, 2005.

[7] Y. Wen and M. Jeng, "Diagnosability analysis based on t-invariants of Petri nets," in *Networking, Sensing and Control, 2005. Proceedings*, Mar. 2005, pp. 371– 376.

[8] Y. Wen, C. Li, and M. Jeng, "A polynomial algorithm for checking diagnosability of Petri nets," in *Proc. SMC'05: IEEE Int. Conf. on Systems, Man, and Cybernetics*, Oct. 2005, pp. 2542– 2547.

[9] T. Murata, "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[10] D. Corona, A. Giua, and C. Seatzu, "Marking estimation of Petri nets with silent transitions," *IEEE Trans. Automatic Control*, vol. 52, no. 9, pp. 1695–1699, Sep. 2007.

[11] C. Cassandras and S. Lafortune, *Introduction to discrete event systems, Second Edition*. Springer, 2007.

[12] T. Yoo and H. Garcia, "Event diagnosis of discrete-event systems with uniformly and nonuniformly bounded diagnosis delays," *Discrete Events Dynamical Systems*, 2008.

[13] S. Genc and S. Lafortune, "Distributed diagnosis of discrete event systems using Petri nets," in *Proc. of the 24th ATPN*, Jun. 2003, pp. 316–336.

[14] M. Cabasino, A. Giua, and C. Seatzu, "Diagnosis of discrete event systems using labeled Petri nets," in *Proc. 2nd IFAC Workshop on Dependable Control of Discrete Systems (Bari, Italy)*, Jun. 2009.