On the Equivalence of Observation Structures for Petri Net Generators

Yin Tong[1], Zhiwu Li[2] and Alessandro Giua[3]

Abstract

Observation structures considered for Petri net generators usually assume that the firing of transitions may be observed through a static mask and that the marking of some places may be measurable. These observation structures, however, are rather limited, namely they do not cover all cases of practical interest where complex observations are possible. We consider in this paper more general ones, by correspondingly defining two new classes of Petri net generators: labeled Petri nets with outputs (LPNOs) and adaptive labeled Petri nets (ALPNs). To compare the modeling power of different Petri net generators, the notion of observation equivalence is proposed. ALPNs are shown to be the class of bounded generators possessing the highest modeling power. Looking for bridges between the different formalisms, we first present a general procedure to convert a bounded LPNO into an equivalent ALPN or even into an equivalent labeled Petri net (if any exists). Finally, we discuss the possibility of converting an unbounded LPNO into an equivalent ALPN.

Keywords: Discrete event system, Petri net, observation, state estimation.

To appear as:

Y. Tong, Z.W. Li, A. Giua, "On the Equivalence of Observation Structures for Petri Net Generators," IEEE Trans. on Automatic Control, Vol. 61, No. 9, 2016. DOI: 10.1109/TAC.2015.2496500

[1]Yin Tong is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China yintong@stu.xidian.edu.cn

[2]Zhiwu Li is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia, and also with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China zhwli@xidian.edu.cn

[3]Alessandro Giua is with Aix Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296, Marseille 13397, France and also with DIEE, University of Cagliari, Cagliari 09124, Italy alessandro.giua@lsis.org; giua@diee.unica.it

I. Introduction

Discrete event systems (DESs) are processes whose state space is discrete and whose evolution is driven by the occurrence of events. The behavior of a *logical*, i.e., *untimed*, DES can be described in terms of sequences that specify the order of event occurrences. Such behavior can be represented by means of a *formal language* whose alphabet is the event set of a DES and the event sequences are words in that language. The issue of representing languages using appropriate modeling formalisms is a key issue for performing analysis and control of DESs [1].

It is often assumed that the initial state of a system is known but the system dynamics is not perfectly known due to partial observations provided by sensors. The set of events is partitioned into two disjoint sets: observable events whose occurrences can be detected by sensors and unobservable ones whose occurrences cannot be detected. In this paper, Petri net generators are considered, where the state is given by token distribution on places, and events are represented by transitions. A classical Petri net generator to model systems with the aforementioned observation structure is the so called *labeled Petri net* (LPN). LPNs have been adopted by many researchers to analyze and control a DES [2]–[6]. In [7]–[11] a more general model where state information may also be provided by sensors is considered: in particular they assume that some places of a Petri net may be observable, i.e., the number of tokens that they contain can be measured. In this case, there are two types of observations: labels of transitions and components of markings. Such a class of Petri net generators is usually called *partially observed Petri nets* (POPNs) [10], [11]. This class of generators has been extend in [12] considering observations that are linear functions of the marking and thus can model sensors that are not able to provide precise measurements of the state components but only information such as the total amount of available resources regardless of their distribution. However, this type of observation cannot describe affine or general nonlinear functions of the marking.

In this paper we aim to better formalize and generalize current Petri net observation structures. We show how different structures can be used to naturally model different types of sensing devices, we compare them and present algorithms to convert one structure into another one if possible. Two more general classes of Petri net generators are considered: *labeled Petri nets with outputs* (LPNOs) and *adaptive labeled Petri nets* (ALPNs) [13], [14].

- An LPNO can be thought of as a labeled Petri net endowed with additional state sensors: an *output function* provides an observation that is an arbitrary function of the current net marking. Therefore, in an LPNO there are two types of observations: event observations generated by the labeling function and state observations generated by the output function.
- An ALPN can be regarded as a labeled Petri net whose labeling function depends on the current marking, i.e., the observation produced by a transition firing may change as the net evolves.

We believe that each of these two classes of generators represents a useful modeling formalism in the system design stage providing an intuitive way to capture the logical observation structure (in terms of event and state sensors) needed to solve a control or optimization problem. Examples are given in Section III. When a physical system must be equipped with available commercial sensors it may be convenient to substitute a state sensor (possibly too expensive or unreliable or difficult to implement if the state is not accessible) with additional event sensors that provide an equivalent observation, or vice versa. For this reason it may be useful to study procedures for transforming between models.

In the first part of this paper, to compare the modeling power among different classes of Petri net generators, the notion of *observation equivalence* is proposed. Two Petri net generators are said to be observation equivalent if i) they have the same net structure and, ii) for an arbitrary firing sequence that occurs in the two generators, the state and sequence estimations reconstructed the two observations are identical (we refer to Definition 11 for a precise definition). We point out that the notion of observation equivalence proposed in this paper is not related to *what the observer sees* (e.g., the observed language) but rather to *what the observer can infer about the system's dynamical evolution*. Thus the results presented herein are relevant to addressing a wide range of problems that are currently investigated in the DES literature, such as state estimation, failure diagnosis or opacity [4], [5], [15], [16].

Ru and Hadjicostis [10] showed that for any POPN there exists an observation equivalent LPN. In the paper, we generalize this result to the larger class of LPNOs whose output function is an affine function, called *labeled Petri nets with an affine output function* (LPNAFs). We also show that LPNOs and ALPNs have higher modeling power than LPNs and are not comparable between them.

Finally, we restrict our attention to bounded Petri net generators that describe systems with a finite state space. In this case we prove that any bounded LPNO can be converted into an observation equivalent ALPN. This implies that ALPNs are the class of bounded generators with higher modeling power. This motivates us to study the conversion from bounded LPNOs into ALPNs. In particular, an algorithm to convert a bounded LPNO into an observation equivalent ALPN is proposed. The algorithm relies on the *vertex coloring* of a special graph and can be used to determine the ALPN with a minimal alphabet or, if it exists, an LPN observation equivalent to the given bounded LPNO. A sufficient and necessary condition for the existence of an LPN equivalent to a given bounded LPNO is also developed. Finally, we show that in some cases the conversion is applicable to unbounded LPNOs.

We believe the aforementioned conversion procedure to have several merits.

- First, finding a conversion procedure between two different formalisms has a theoretical interest *per se* and in the literature several approaches of this type have been proposed for models of concurrent systems (e.g., communicating sequential processes, place/transition nets, process algebra) or performance models (e.g., stochastic Petri nets, queueing networks).
- Second, if an automatic conversion procedure from LPNOs to ALPNs is available, it is sufficient to derive approaches for analysis and design of the most general class ALPNs rather than for each model. As a particular case, in this conversion an LPN may be obtained and several results concerning this model have already been presented in the literature [2]–[6].

This paper is organized as follows. Section II recalls the notions of Petri nets and existing Petri net generators. Formal definitions of labeled Petri nets with outputs and adaptive labeled Petri nets are presented in Section III. In Section IV we formally state the notion of observation equivalence based on which the modeling power between different classes of Petri net generators is compared. An algorithm that converts a bounded LPNO into an observation equivalent ALPN is developed in Section V where the number of labels of the observation equivalent ALPN is also discussed. Then in Section VI, a sufficient and necessary condition for the existence of the observation equivalent LPN to a bounded LPNO is reported, and the corresponding conversion algorithm is presented. In Section VII the conversion of unbounded LPNOs is studied. Finally, conclusions and future work are presented.

II. Background

In this section the formalism used in the paper is introduced. For more details on Petri nets, we refer readers to [17].

A. Petri Nets

A *Petri net* is a structure $N = (P, T, Pre, Post)$, where P is a set of m *places* graphically represented by circles; T is a set of n *transitions* graphically represented by bars; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the *pre-* and *post-incidence functions* that specify the arcs directed from places to transitions, and vice versa, where $\mathbb{N}$ is the set of non-negative integers. We also denote by $C = Post - Pre$ the incidence matrix of a net.

A *marking* is a vector $M : P \to \mathbb{N}$ that assigns to each place a non-negative integer number of tokens, represented by black dots. We denote by $M(p)$ the marking of place p. For economy of space, a marking can also be denoted as $M = \sum_{p \in P} M(p) \cdot p$. A *Petri net system* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0.

A transition t is *enabled* at marking M if $M \geq Pre(\cdot, t)$ and may fire yielding a new marking $M' = M + C(\cdot, t)$. We use $M[\sigma\rangle$ to denote that the sequence of transitions $\sigma = t_{j1} \cdots t_{jk}$ is enabled at M, and $M[\sigma\rangle M'$ to denote that the firing of σ yields M'. The set of all transition sequences that can fire in a net system $\langle N, M_0 \rangle$ is denoted by $\mathcal{L}(N, M_0) = \{\sigma \in T^* | M_0[\sigma\rangle\}$, where T^* denotes the Kleene closure of T, i.e., the set of all sequences over T including the empty sequence ε.

A marking M is *reachable* in $\langle N, M_0 \rangle$ if there exists a firable sequence $\sigma \in \mathcal{L}(N, M_0)$ 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 by $R(N, M_0)$. A Petri net system is *bounded* if there exists a non-negative integer $K \in \mathbb{N}$ such that for any place $p \in P$ and for any reachable marking $M \in R(N, M_0)$, $M(p) \leq K$ holds.

B. Labeled Petri Nets

We consider the case in which an external agent (e.g. the supervisor in a supervisory control problem, or the intruder in an opacity problem) that knows the initial marking and the structure of the PN but observes the firing of transitions through a mask. A common assumption is that of considering the mask as a projection from the set of transitions T to an alphabet Σ which represents available sensors readings [5], [15], [18]. The mask is possibly evasive, i.e., the output label assigned to a transition may either be a symbol from the alphabet or the empty string ε to denote that the firing of the transition does not produce an observable reading. A transition of the latter type is said to be *unobservable* (or *silent*). Such an observation structure can be formalized as follows.

Definition 1: A *labeled Petri net* (LPN) is a generator $G_L = (N, M_0, \Sigma, \ell)$, where $\langle N, M_0 \rangle$ is a Petri net system, Σ is an *alphabet* (a set of labels), and $\ell : T \to \Sigma \cup \{\varepsilon\}$ is a *labeling function* that assigns to each transition $t \in T$ either a symbol from Σ or the empty word ε. ◇

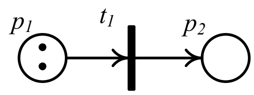


Fig. 1. Petri net system

The labeling function used in this work and defined above is called an *arbitrary labeling function*, i.e., different transitions may share the same label and also a transition may be labeled with the empty word.

Given a firing sequence σ, the corresponding observation generated by the observation function is defined as follows.

Definition 2: Given an LPN $G_L = (N, M_0, \Sigma, \ell)$, the *observation function* of G_L is defined as a mapping $L_L : T^* \to \Sigma^*$ that associates a firing sequence $\sigma = t_1 t_2 \cdots t_k$ with the observation $w = L_L(\sigma) = \ell(t_1)\ell(t_2)\cdots\ell(t_k)$. $\diamond$

Example 1: Consider an LPN $G_L = (N, M_0, \Sigma, \ell)$, where $\langle N, M_0 \rangle$ is shown in Fig. 1, $\Sigma = \{a\}$ and $\ell(t_1) = a$. Let $\sigma = t_1 t_1$. Then the observation produced is $w = L_L(\sigma) = aa$. $\diamond$

C. Partially Observed Petri Nets

In addition to sensors that detect the firing of transitions, it may also be possible to have sensors that provide information on the markings of a net. Several researchers studied Petri nets where some places are observable, i.e., their token content [7]–[10], or even more general, a linear combination of their token contents [12] is known. Such observation structures can be formalized as follows.

Definition 3: A partially observed Petri net (POPN) is a generator $G_P = (N, M_0, \Sigma, \ell, V)$, where (N, M_0, Σ, ℓ) is an LPN, $P_O \subset P$ is a set of observable places and $V \in \mathbb{R}^{|P_O| \times |P|}$ is a *place sensor configuration*, where $\mathbb{R}$ denotes the set of real numbers. $\diamond$

The observations in a POPN are strings of triple (observation of the start state, label of the transition, observation of the reached state).

Definition 4: Let $G_P = (N, M_0, \Sigma, \ell, V)$ be a POPN and $\sigma = t_1 \cdots t_k$ be a firing sequence with $M_0[t_1\rangle M_1 \cdots M_{k-1}[t_k\rangle M_k$. The *observation function* of G_P is defined as a mapping $L_P : T^* \to \mathbb{N}^{|P_o|} \times \Sigma \times \mathbb{N}^{|P_o|}$ that associates sequence σ with the observation

$$s_P = L_P(\sigma) = (M_{V0}, \ell(t_1), M_{V1}) \cdots (M_{Vk-1}, \ell(t_k), M_{Vk}),$$

where $M_{Vi} = V \cdot M_i$ and $i \in \{0, 1, 2, \cdots, k\}$.

As a particular case, $(M_{Vi}, \ell(t), M_{Vj})$ is defined as the null observation, if $\ell(t) = \varepsilon$ and $M_{Vi} = M_{Vj}$. $\diamond$

Example 2: Consider a POPN $G_P = (N, M_0, \Sigma, \ell, V)$, where $\langle N, M_0 \rangle$ is shown in Fig. 1, $\Sigma = \emptyset$, $\ell(t_1) = \varepsilon$, and $V = [0\ 1]$. Let $\sigma = t_1 t_1$. Then we have $M_0[t_1\rangle M_1[t_1\rangle M_2$, where $M_1 = [1\ 1]^T$ and $M_2 = [0\ 2]^T$. Therefore, $M_{V0} = V \cdot M_0 = 0$, $M_{V1} = V \cdot M_1 = 1$ and $M_{V2} = V \cdot M_2 = 2$. The corresponding observation is $s_P = L_P(\sigma) = (0, \varepsilon, 1)(1, \varepsilon, 2)$. $\diamond$

III. General Petri Net Generators

The two classes of generators defined in the previous section, namely labeled Petri nets and partially observed Petri nets, have been studied by several authors and are by now well understood. However, they do not cover all cases of practical interest where more complex observations are possible. This has motivated us to define in [13], [14] two new more general classes of generators that are called *labeled Petri nets with outputs* and *adaptive labeled Petri nets*, respectively. In this section these new classes are proposed and we also define a new type of generators that is an interesting special class of labeled Petri nets with outputs. Furthermore, a discussion about modeling using different generators and the motivation of this work are developed. At the end of the section the structural relationships between all these generators are presented, while in the next section we will formalize and discuss the notion of equivalence between generators in terms of observations.

A. Labeled Petri Nets with Outputs

Partially observed Petri nets consider a very particular class of state observations where the exact marking of some places, or a linear combination of the markings, is measured. However, in many cases a sensor may provide more general information about the state: consider, as an example, the case of a buffer where a sensor only detects if the buffer is empty or not. This motivated us in [13] to define a class of labeled Petri nets endowed with a general observation function associated to state sensors.

Definition 5: A *labeled Petri net with outputs* (LPNO) is a generator $G_O = (N, M_0, \Sigma, \ell, f)$, where (N, M_0, Σ, ℓ) is an LPN and $f: R(N, M_0) \to \mathbb{R}^k$ is an *output function* associated with $k \in \mathbb{N}$ state sensors. ⋄

In an LPNO there are two types of observations: transition labels and marking information.

Definition 6: Given an LPNO $G_O = (N, M_0, \Sigma, \ell, f)$, let $\sigma = t_1 \cdots t_k$ be a firing sequence producing the trajectory $M_0[t_1\rangle M_1 \cdots M_{k-1}[t_k\rangle M_k$. The *observation function* of G_O is defined as a mapping $L_O : T^* \to (\Sigma \times \mathbb{R}^k)^*$ that associates σ with the observation

$$s = L_O(\sigma) = (\ell(t_1), \Delta f(M_0, t_1)) \cdots (\ell(t_k), \Delta f(M_{k-1}, t_k)),$$

where $\Delta f(M_{i-1}, t_i) = f(M_i) - f(M_{i-1}) \in \mathbb{R}^k$, $i = 1, 2, \cdots, k$.

If $\ell(t_i) = \varepsilon$ and $\Delta f(M_{i-1}, t_i) = 0$, the observation $(\varepsilon, 0)$ is the *null observation* as no transition firing is detected. ⋄

Remark: since the initial marking is assumed to be known, the initial observation $f(M_0)$ provides no additional information. In this case the two sequences $f(M_0), f(M_1), \ldots$ and $\Delta f(M_0, t_1), \Delta f(M_1, t_2), \ldots$ contain the same information. This also implies that the observation s_P in a POPN (see Definition 4) contains the same information as the observation $(\ell(t_1), V \cdot M_1 - V \cdot M_0) \cdots (\ell(t_k), V \cdot M_k - V \cdot M_{k-1})$ and we can conclude that POPNs are a special subclass of LPNOs whose output function is $f(M) = V \cdot M$.

Example 3: Consider an LPNO $G_O = (N, M_0, \Sigma, \ell, f)$, where $\langle N, M_0 \rangle$ is shown in Fig. 1, $\Sigma = \emptyset$, $\ell(t_1) = \varepsilon$, and the output function is $f(M) = min\{M(p_2), 1\}$. Let the firing sequence be $\sigma = t_1 t_1$. Based on the result in Example 2, we have $f(M_0) = 0$, $f(M_1) = 1$ and $f(M_2) = 1$. Therefore, $\Delta f(M_0, t_1) = 1$ and $\Delta f(M_1, t_1) = 0$.

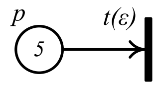


Fig. 2. LPNO model of a manufacturing cell

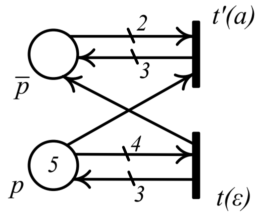


Fig. 3. LPN model of the manufacturing cell

The corresponding observation would be $s = (\varepsilon, 1)$, since the second firing of t_1 produces the null observation $(\varepsilon, 0)$. ⋄

The following example shows that LPNOs provide an intuitive way to model systems with arbitrary state sensors.

Example 4: Consider the net in Fig. 2 describing a manufacturing cell: there is a buffer modeled by places p, and a robot modeled by transition t that moves products. The action of the robot is not detectable, i.e., transitions t is labeled with the empty string. However, on the buffer there is a counter whose measuring range is from 0 to 3: if the content is lower than three, the device counts the products in p; otherwise a saturation will be reached. Therefore, the output function is

$$f(M) = \begin{cases} 3 & if\ M(p) \geq 3; \\ M(p) & otherwise. \end{cases}$$

We note that it may also be possible to use LPNs to describe this system since place p is 5-bounded, i.e., for all reachable markings M it holds $M(p) \leq 5$. The corresponding LPN is the much less intuitive net shown in Fig. 3. Here place $\overline{p}$ is the complementary place of p (i.e., $M(p) + M(\overline{p}) = 5$) and t' is a duplicate of t. If $M(p) \geq 3$, t is activated; otherwise, t' is activated. The LPN has a larger size and, moreover, if the bound of p or the range of the counter changes, the LPN structure has to be changed. In addition, if p is unbounded, no LPN can model the system, since no complementary place can be defined.

We next define a particular subclass of LPNOs called labeled Petri nets with an affine output function.

Definition 7: A *labeled Petri net with an affine output function* (LPNAF) is an LPNO $G_O = (N, M_0, \Sigma, \ell, f)$ whose output function is an affine function $f(M) = A \cdot M + B$ with constant matrices $A \in \mathbb{R}^{k \times m}$ and $B \in \mathbb{R}^k$. ⋄

Note that the POPNs considered in [7]–[10], [12] are all subclasses of LPNAF where matrix $B = 0$.

Example 5: Consider an LPNAF $G_O = (N, M_0, \Sigma, \ell, f)$, where $\langle N, M_0 \rangle$ is shown in Fig. 1, $\Sigma = \emptyset$, $\ell(t_1) = \varepsilon$ and $f(M) = A \cdot M + B$ with $A = [1 \ \ -1.5]$ and $B = 2$. Let $\sigma = t_1 t_1$. The corresponding observation would be $s = (\varepsilon, -2.5)(\varepsilon, -2.5)$. ⋄

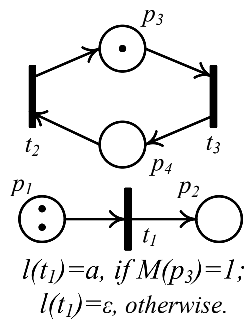


Fig. 4. ALPN model of the system with sensors switched on/off.

B. Adaptive Labeled Petri Nets

In the framework of DESs with partial event observations, it is usually assumed that the observation corresponding to an event is static, i.e., it does not change as the system evolves. However, there are some situations where the observation produced by the occurrence of an event also depends on the states. Let us consider, as an example, the case of a sensor that may be turned off in some states or where communication failures change the observation. Some studies have considered this paradigm in DESs modeled by automata [19]–[22]. To the best of our knowledge, it has never been defined in the framework of Petri nets, which motivated us to define a Petri net generator where the labeling function depends on the state: we call it an adaptive labeled Petri net.

Definition 8: An *adaptive labeled Petri net* (ALPN) is a generator $G_A = (N, M_0, \Sigma_A, \ell_A)$, where $\langle N, M_0 \rangle$ is a Petri net system, Σ_A is an alphabet and $\ell_A : R(N, M_0) \times T \to \Sigma_A \cup \{\varepsilon\}$ is an *adaptive labeling function*. ⋄

According to the definition of ALPNs, the label assigned to a transition need not be fixed but may depend on the states. However, the observation corresponding to a firing sequence is a string of labels the same as the one in an LPN.

Definition 9: Given an ALPN $G_A = (N, M_0, \Sigma_A, \ell_A)$, let $\sigma = t_1 \cdots t_k$ be a firing sequence producing the trajectory $M_0[t_1\rangle M_1 \cdots M_{k-1}[t_k\rangle M_k$. The *observation function* of G_A is defined as a mapping $L_A : T^* \to \Sigma_A^*$ that associates sequence σ with the observation

$$w_A = L_A(\sigma) = \ell_A(M_0, t_1) \cdots \ell_A(M_{k-1}, t_k).$$

⋄

Example 6: Consider an ALPN $G_A = (N, M_0, \Sigma_A, \ell_A)$, where $\langle N, M_0 \rangle$ is shown in Fig. 1, $\Sigma_A = \{a\}$, and the adaptive labeling function is $\ell_A(M_0, t_1) = a$ and $\forall M \in \{[1\ 1]^T, [0\ 2]^T\}$, $\ell_A(M, t_1) = \varepsilon$. Let the firing sequence still be $\sigma = t_1 t_1$. The observation would be $w_A = a$. ⋄

The following example shows that ALPNs provide an intuitive way to model systems with state dependent event labels.

Example 7: Consider the manufacturing cell modeled by the ALPN in Fig. 4, where transition t_1 represents the operation of a robot moving products from an upstream buffer (p_1) to a downstream buffer (p_2). A sensor on the

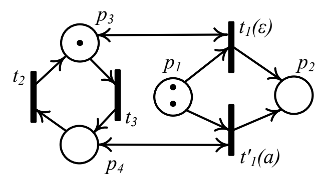


Fig. 5. LPN model of the system with sensors switched on/off.

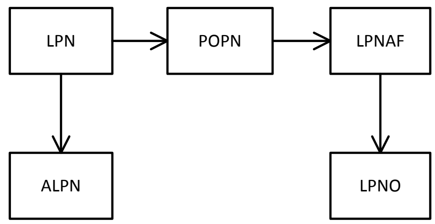


Fig. 6. Structural relationships between generators.

robot may be turned on (place p_3 is marked) and off (place p_3 is empty) by suitable commands (transitions t_2 and t_3). When the sensor is on, the operation of the robot is detected, otherwise it is unobservable. We model this with a state dependent label

$$\ell(t_1) = \begin{cases} a & if \ M(p_3) \geq 0; \\ \varepsilon & otherwise. \end{cases}$$

Note that in this particular case the system can also be modeled by the LPN in Fig. 5, where t_1' is a duplicate of t_1. However, such an LPN model has a larger size and is less intuitive.

From a structural point of view, the relationships between the classes of generators previously defined can be summarized in Fig. 6. For each arc, the class corresponding to the head node is more general than that corresponding to the tail node.

IV. Observation Equivalence

In the previous section we have compared the different generators introduced in this paper in terms of structural relationships. Here we address the problem of comparing them in terms of modeling power by introducing an appropriate notion of observation equivalence.

We point out a fact: if a model is structurally more general than another, it does not necessarily mean that it has greater modeling power. As an example, it is well known that nondeterministic automata are a generalization of deterministic automata but as far as the languages are concerned, the two models have the same power. In fact, there exists a well known procedure [1] to convert a nondeterministic automaton into an equivalent deterministic one that accepts the same language.

We assume that the purpose of observing a system is that of reconstructing both the sequence of events that has occurred and the current state of the system. To this end, we propose a notion of observation equivalence that

TABLE I

FIRING ESTIMATES IN G_O AND G_A

		$\sigma = \varepsilon$	$\sigma = t_1$	$\sigma = t_1 t_1$
G_O	s	ε	$(\varepsilon, 1)$	$(\varepsilon, 1)$
	$\mathcal{S}(s)$	$\{\varepsilon\}$	$\{t_1, t_1 t_1\}$	$\{t_1, t_1 t_1\}$
G_A	w_A	ε	a	a
	$\mathcal{S}(w_A)$	$\{\varepsilon\}$	$\{t_1, t_1 t_1\}$	$\{t_1, t_1 t_1\}$

applies to generators having the same underlying net structure but a different observation structure: two generators are observation equivalent if their observation structures provide the same information on the transition firings and on the markings.

In the following let

$$\mathcal{G} = \{LPN, POPN, LPNAF, LPNO, ALPN\}$$

denote the set of all these classes of generators.

Definition 10: Consider a generator G in class $\mathcal{X} \in \mathcal{G}$, whose underlying net system $\langle N, M_0 \rangle$ is assumed to be known. Let L be its observation function, and x an observation. We define:

- the set of *firing sequences consistent with* x as

$$\mathcal{S}(x) = \{\sigma \in \mathcal{L}(N, M_0) \mid L(\sigma) = x\};$$

- the set of *markings consistent with* x as

$$\mathcal{C}(x) = \{M \in \mathbb{N}^m \mid \exists \sigma \in \mathcal{S}(x) : M_0[\sigma\rangle M\}.$$

$\diamond$

Using these sets we define the notion of observation equivalence between generators.

Definition 11: A generator G in class $\mathcal{X}$ is said to be *observation equivalent* to a generator G' in class $\mathcal{X}'$ if the following two conditions hold:

i) G and G' have the same net system $\langle N, M_0 \rangle$,

ii) for any sequence $\sigma \in \mathcal{L}(N, M_0)$ that produces an observation x in G and an observation x' in G', $\mathcal{S}(x) = \mathcal{S}(x')$ holds.

Note that in Definition 11, $\mathcal{S}(x) = \mathcal{S}(x')$, together with condition i), implies $\mathcal{C}(x) = \mathcal{C}(x')$. In this paper, "equivalence" always refers to "observation equivalence". The notion of observation equivalence between generators induces a meaningful relationship between classes of generators.

Example 8: Consider the LPNO G_O in Example 3 and the ALPN G_A in Example 6. These two generators are observation equivalent, since they have the same net system and according to Table I, for all $\sigma \in T^*$ it holds $\mathcal{S}(L_O(\sigma)) = \mathcal{S}(L_A(\sigma))$. $\diamond$

Definition 12: Given two classes of Petri net generators $\mathcal{X}, \mathcal{X}' \in \mathcal{G}$, class $\mathcal{X}$ is said to be *observation weaker* than $\mathcal{X}'$ if for any generator G in class $\mathcal{X}$ there exists an observation equivalent generator G' in class $\mathcal{X}'$. This relation is denoted by

$$\mathcal{X} \preccurlyeq \mathcal{X}'.$$

We also write:

- $\mathcal{X} \approx \mathcal{X}'$ if $\mathcal{X} \preccurlyeq \mathcal{X}'$ and $\mathcal{X}' \preccurlyeq \mathcal{X}$: in this case we say that the two classes are *observation equivalent*;
- $\mathcal{X} \precnapprox \mathcal{X}'$ if $\mathcal{X} \preccurlyeq \mathcal{X}'$ and $\mathcal{X}' \not\preccurlyeq \mathcal{X}$ hold[1]: in this case we say that class $\mathcal{X}$ is *strictly observation weaker* than $\mathcal{X}'$;
- $\mathcal{X} \not\sim \mathcal{X}'$ if $\mathcal{X} \not\preccurlyeq \mathcal{X}'$ and $\mathcal{X}' \not\preccurlyeq \mathcal{X}$ hold: in this case we say that the two classes are *not observation comparable*.

$\diamond$

Obviously if class $\mathcal{X}'$ is structurally more general than $\mathcal{X}$ (see Fig. 6), then $\mathcal{X} \preccurlyeq \mathcal{X}'$ holds; here we complete the analysis by discussing when two classes are observation equivalent or not comparable.

A. LPNs, POPNs and LPNAFs

In this section we show that LPNs, POPNs and LPNAFs are observation equivalent. This generalizes a result by Ru and Hadjicostis [10] who proved the equivalence between LPNs and POPNs.

Proposition 1: LPNs, POPNs and LPNAFs are observation equivalent, i.e., $LPN \approx POPN \approx LPNAF$.

Proof: The relationship $LPN \preccurlyeq POPN \preccurlyeq LPNAF$ immediately follows from the structural relationship in Fig. 6. We now complete the proof by showing that $LPNAF \preccurlyeq LPN$. To do this we provide a constructive procedure that, given an arbitrary LPNAF $G_O = (N, M_0, \Sigma, \ell, f)$ with $f = A \cdot M + B$, determines an equivalent LPN $G_L = (N, M_0, \Sigma', \ell')$.

Given an LPNAF G_O, let $T_e = \{t \in T | \ell(t) = e\}$ with $e \in \Sigma \cup \{\varepsilon\}$ be the set of transitions that have the same label e and C_e be the incidence matrix restricted to T_e. For any $e \in \Sigma$, set T_e is further divided into $T_e = T_{e1} \cup \cdots \cup T_{el}$ such that $\forall t \in T_{ei}$ $(i \in \{1, 2, \cdots, l\})$ the corresponding columns $C_e^A(\cdot, t)$ of matrix $C_e^A = A \cdot C_e$ are identical. For $e = \varepsilon$, set T_ε is divided into $T_\varepsilon = T_{\varepsilon 0} \cup T_{\varepsilon 1} \cup \cdots \cup T_{\varepsilon l}$ such that $\forall t \in T_{\varepsilon 0}$, the corresponding columns $C_\varepsilon^A(\cdot, t) = \vec{0}$ and $\forall t \in T_{ei}$ $(i \in \{1, 2, \cdots, l\})$, the corresponding columns $C_\varepsilon^A(\cdot, t)$ are identical. Then the equivalent LPN $G_L = (N, M_0, \Sigma', \ell')$ has labeling: $\forall e \in \Sigma \cup \{\varepsilon\}$, $\forall t \in T_{ei}$ with $i \in \{1, \cdots, l\}$, $\ell'(t) = ei$ and $\forall t \in T_{\varepsilon 0}$, $\ell'(t) = \varepsilon$. In the following, we prove that G_L is equivalent to G_O.

[1]Here $\mathcal{X}' \not\preccurlyeq \mathcal{X}$ denotes that the relation $\mathcal{X}' \preccurlyeq \mathcal{X}$ does not hold, i.e., there exists at least one generator in $\mathcal{X}'$ such that there is no generator in $\mathcal{X}$ observation equivalent to it.

Let transition $t' \in T$ fire at marking $M \in R(N, M_0)$ with $M[t'\rangle M'$. The corresponding observation in G_O would be $s = (\ell(t'), \Delta f)$, where $\ell(t') = e$ and

$$\begin{aligned}\Delta f &= f(M') - f(M) \\ &= A \cdot M' + B - (A \cdot M + B) \\ &= A \cdot (M' - M) \\ &= A \cdot C(\cdot, t') \\ &= C_e^A(\cdot, t').\end{aligned}$$

Therefore, for G_O the set of firing transitions consistent with s from marking M is $\mathcal{S}(s) = \{t \in T_e | M[t\rangle \wedge C_e^A(\cdot, t) = \Delta f\}$. Assume that the observation in G_L is $w = ej$. For G_L the set of firing transitions consistent with w from marking M is $\mathcal{S}(w) = \{t \in T_{ej} | M[t\rangle\}$. According to the definition of T_{ej}, we have $\forall t \in \mathcal{S}(w), C_e^A(\cdot, t) = C_e^A(\cdot, t') = \Delta f$. Namely, $\mathcal{S}(w) = \mathcal{S}(s)$. Furthermore, it also indicates that, given a transition t, at every marking where transition t is enabled, the firing of t will cause the same observation $(\ell(t), f(M') - f(M))$. Thus the proof can be easily extended to firing sequences. ■

Example 9: Consider the LPNAF $G_O = (N, M_0, \Sigma, \ell, f)$ in Example 5, whose incidence matrix is $C = [-1\ 1]^T$. Based on the constructive procedure in the proof of Proposition 1, for transition t_1, we have that $\Delta f = A \cdot C(\cdot, t_1) = -2.5$, different from 0. Therefore, the equivalent LPN is $G_L = (N, M_0, \Sigma', \ell')$, where $\ell'(t) = a$ and $\Sigma' = \{a\}$. ⋄

B. LPNs and LPNOs

In this section we discuss the observation relationship between LPNs and LPNOs.

Proposition 2: LPNs are strictly observation weaker than LPNOs, i.e., $LPN \precnapprox LPNO$.

Proof: Fig. 6 shows that LPNOs are structurally more general than LPNs, which implies $LPN \preccurlyeq LPNO$. According to Definition 12, it is sufficient to prove $LPNO \not\preccurlyeq LPN$ by giving an LPNO whose equivalent LPN does not exist.

Consider the LPNO G_O in Example 3. Assume that there is an LPN $G_L = (N, M_0, \Sigma', \ell')$ equivalent to G_O. Since the labeling function in G_L is static, the labeling function only could be $\ell'(t_1) = \varepsilon$ or $\ell'(t_1) = a$, i.e., transition t_1 in G_L is either observable or not.

- Assume that the labeling function in G_L is $\ell'(t_1) = \varepsilon$. At the initial marking, the firing of t_1 will produce the observation $w = \varepsilon$ in G_L. The set of firing sequences consistent with w is $\mathcal{S}(w) = \{\varepsilon, t_1, t_1t_1\}$. On the other hand, in G_O the corresponding observation is $s = (\varepsilon, 1)$, and thus the set of possible firing sequences is $\mathcal{S}(s) = \{t_1, t_1t_1\}$. According to Definition 11, these two generators are not equivalent.
- Assume the labeling function in G_L is $\ell'(t_1) = a$. At the initial marking, the firing of t_1 will produce the observation $w = a$ in G_L and $\mathcal{S}(w) = \{t_1\}$, while in G_O, the observation is $s = (\varepsilon, 1)$ and $\mathcal{S}(s) = \{t_1, t_1t_1\}$. Therefore, G_O and G_L are still not equivalent. In conclusion, there is no LPN equivalent to G_O.

■

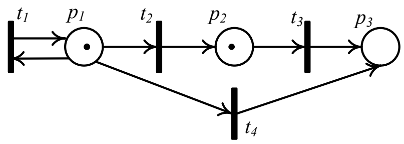


Fig. 7. ALPN that cannot be converted into an LPNO.

From the equivalence between LPNs, POPNs and LPNAFs, a result also follows.

Corollary 1: POPNs and LPNAFs are strictly observation weaker than LPNOs. $\diamond$

C. LPNs and ALPNs

Now we consider the observation relation between LPNs and ALPNs, two classes of generators where only event occurrences are observed.

Proposition 3: LPNs are strictly observation weaker than ALPNs, i.e., $LPN \precneqq ALPN$.

Proof: The relationship $LPN \preccurlyeq ALPN$ trivially follows from the structural relationship in Fig. 6. Now we prove $ALPN \not\preccurlyeq LPN$ by giving an ALPN whose equivalent LPN does not exist.

Consider the ALPN G_A in Example 6. Assume that there is an LPN $G_L = (N, M_0, \Sigma, \ell)$ equivalent to G_A. G_L has the same net system $\langle N, M_0 \rangle$ with G_A. The possible labeling function of G_L is either $\ell(t_1) = \varepsilon$ or $\ell(t_1) = b \in \Sigma$. Namely, in G_L transition t_1 is either unobservable or observable. Let $\ell(t_1) = b$ (the case that transition t_1 is unobservable can be proved in the same way) and a firing sequence be $\sigma = t_1 t_1$. Then, the corresponding observations in G_A and G_L are $w_A = a$ and $w = bb$, respectively. Therefore, in G_A, the set of firing sequences consistent with w_A is $\mathcal{S}(w_A) = \{t_1, t_1 t_1\}$; in G_L, the set of firing sequences consistent with w is $\mathcal{S}(w) = \{t_1 t_1\}$, i.e., $\mathcal{S}(w_A) \neq \mathcal{S}(w)$. We conclude that G_L is not equivalent to G_A. There is no LPN equivalent to G_O. ■

From the equivalence between LPNs, POPNs and LPNAFs, the following result is drived.

Corollary 2: POPNs and LPNAFs are strictly observation weaker than ALPNs. $\diamond$

D. LPNOs and ALPNs

Fig. 6 shows that there is no specific structural relation between LPNOs and ALPNs. In this section, we will show that these classes are not comparable either with respect to the observation equivalence relation.

Proposition 4: ALPNs and LPNOs are not observation comparable, i.e., $ALPN \nsim LPNO$.

Proof: a) First, we prove that $ALPN \not\preccurlyeq LPNO$ is true by means of an example. Let us consider the ALPN in Fig. 7 with initial marking $M_0 = [1\ 1\ 0]^T$ and the adaptive labeling function given by Table II.

For observed words $w_A = aa$ and $w_A = b$, the sets $\mathcal{S}(aa)$ and $\mathcal{S}(b)$ can be iteratively computed as shown in Fig. 8, where $M_1 = [0\ 2\ 0]^T$, $M_2 = [0\ 1\ 1]^T$ and $M_3 = [1\ 0\ 1]^T$.

We claim that there does not exist an LPNO equivalent to this ALPN. We prove this by contradiction. If we assume that such a generator exists, then its output function necessarily satisfies $f(M_0) = f(M_1) = f(M_2) = f(M_3)$ since

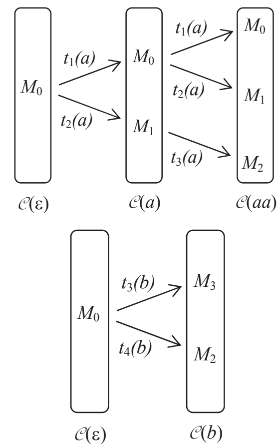


Fig. 8. Computation of sets $\mathcal{C}(aa)$ and $\mathcal{C}(b)$ for the ALPN.

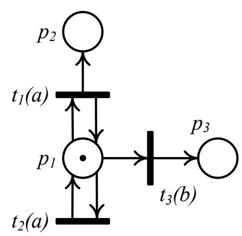


Fig. 9. LPNO whose equivalent ALPN has an infinite number of labels.

otherwise we would be able to distinguish between the three firing sequences $\sigma_1 = t_1t_1, \sigma_2 = t_1t_2$ and $\sigma_3 = t_2t_3$ after the firing of two a's or between the two firing sequences $\sigma_4 = t_3$ and $\sigma_5 = t_4$ after the firing of b. In addition, all transitions necessarily have the same label, say a. However such an LPNO after a would produce a set of consistent firing sequences $\mathcal{S}((a,0)) = \{t_1,t_2,t_3,t_4\} \neq \mathcal{S}(a) = \{t_1,t_2\}$, which contradicts the assumption.

b) Now we show that $LPNO \not\preceq ALPN$ is true by means of another example. Consider the LPNO in Fig. 9, where the output function is

$$f(M) \begin{cases} 0 & if\ M(p_3) = 0; \\ M(p_2) & otherwise. \end{cases}$$

TABLE II

ADAPTIVE LABELING FUNCTION OF THE ALPN

$\ell_A(M,t)$	t_1	t_2	t_3	t_4
M_0	a	a	b	b
$M \neq M_0$	b	b	a	a

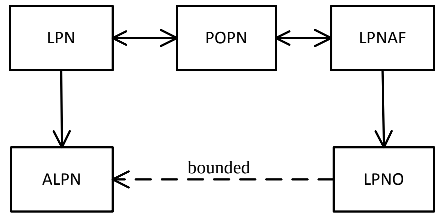


Fig. 10. Observation relationships between generators.

If the observation is $s = \underbrace{(a,0)(a,0)\cdots(a,0)}_{k}(b,x)$, x could be any number from 0 to k. To find the equivalent ALPN we have to assign infinite labels to transition t_3: $\ell_A(M_i, t_3) = [b, i]$, where $M_i = [0\ 1\ i]^T$ for $i \in \mathbb{N}$. In that case the equivalent ALPN needs an infinite alphabet, a condition that is not consistent with Definition 8. ■

Even though LPNOs and ALPNs are not observation comparable, when generators whose underlying net system is bounded are considered, bounded LPNOs are strictly observation weaker than bounded ALPNs.

Proposition 5: Bounded LPNOs ($LPNO_{bound}$) are strictly observation weaker than bounded ALPNs ($ALPN_{bound}$), i.e., $LPNO_{bound} \precneqq ALPN_{bound}$.

Proof: The relation $ALPN_{bound} \not\preccurlyeq LPNO_{bound}$ follows from part a) of the proof of Proposition 4. Thus we are left to prove $LPNO_{bound} \preccurlyeq ALPN_{bound}$. To show this, we present a brute force approach that determines the equivalent ALPN $G_A = (N, M_0, \Sigma_A, \ell_A)$ of a bounded LPNO $G_O = (N, M_0, \Sigma, \ell, f)$. Given a bounded G_O, the adaptive labeling function of its equivalent ALPN $G_A = (N, M_0, \Sigma_A, \ell_A)$ can be determined by the following rule: for any $t \in T$ and $M \in R(N, M_0)$ with $M[t\rangle M'$, $\ell_A(t) = [\ell(t), f(M') - f(M)]$, i.e., the corresponding observation in G_O is assigned as a label to the transition in G_A. The alphabet of G_A is a finite set $\Sigma_A = \{[\ell(t), f(M') - f(M)] | t \in T, M \in R(N, M_0), M[t\rangle M'\}$, since G_O is bounded. Once the transition fires in G_A, the new label exactly describes the observation of G_O and the sets of firing sequences consistent with the observations in G_A and G_O must be identical. Thus the two generators are equivalent. ■

In conclusion, equivalence relations between all classes of Petri net generators discussed in this work are illustrated in Fig. 10. A double arrowed arc $\leftrightarrow$ connects two classes that are observation equivalent while an arrow $\rightarrow$ denotes that the class at the tail is strictly observation weaker than the one at the head. The arrow tagged "bounded" denotes that bounded LPNOs are strictly observation weaker than bounded ALPNs.

V. Conversion of Bounded LPNOs into ALPNs

As mentioned in the introduction, bridges between different formalisms have both theoretical significance and practical relevance. The conversion between LPNs, POPNs, and LPNAFs was discussed in the proof of Proposition 1. According to the structural relations shown in Fig. 6, POPNs and LPNAFs are both subclasses of LPNOs, and LPNs is a subclass of both ALPNs and LPNOs. Therefore, the procedure to convert LPNOs to ALPNs can be also applied to convert generators of all those subclasses to an equivalent ALPNs. Moreover, ALPNs are the class that has the highest modeling power among bounded Petri net generators. For this reason, in the rest of this work we focus on the conversion from LPNOs to ALPNs.

In this section we present an algorithm, that improves the procedure of [14], to convert a bounded LPNO into an equivalent ALPN with a minimal number of labels. The interest for finding a minimal alphabet relies on the following observations: i) applying the brute force approach (in Proposition 5) may introduce unnecessary labels; if we consider the cardinality of the alphabet corresponding to the number of event sensors in the system, reducing the cardinality of the alphabet leads to a cost reduction in the implementation of an observation structure; ii) it may allow us to determine an equivalent net with a finite alphabet even when the brute-force procedure generates an infinite number of labels (we will give such an example in Section VII); and iii) this procedure may allow us to verify that a given LPNO cannot be converted into an LPN, which will be discussed in Section VI.

The proposed conversion algorithm reduces the computation of the adaptive labeling function of the equivalent ALPN to solving the *vertex coloring problem* [23] of a graph called a *conflict graph*. A running example illustrates the algorithm. We assume that LPNOs discussed in this section and the following two are bounded.

A. Problem Reduction

According to Definition 11, two equivalent generators have the same net system. Thus, given an LPNO $G_O = (N, M_0, \Sigma, \ell, f)$, to compute its equivalent ALPN $G_A = (N, M_0, \Sigma_A, \ell_A)$, we only need to determine the adaptive labeling function. We show that this issue can be reduced to solving a *vertex coloring*. The proposed procedure requires three main steps.

Step 1 Since observation equivalence requires the set of consistent markings of the two generators to be identical for all observations, we first determine which pairs of markings are *confusable*, i.e., belong to the same consistent set for some observations.

Step 2 Using this information, we determine which pairs $[M, t] \in R(N, M_0) \times T$ should have the same label in the ALPN constructing the *agreement graph* $\mathcal{A}$. We also determine which pairs $[M, t]$ should have a different label in the ALPN constructing the *conflict graph* $\hat{\mathcal{A}}$.

Step 3 Finally, the problem of finding the label assignment that determines the equivalent ALPN is reduced to solving the vertex coloring of graph $\hat{\mathcal{A}}$.

B. Computation of the Confusion Relation

Given an observation in an LPNO, there may be more than one marking consistent with the observation. First, we define the *confusable relation* between two markings.

Definition 13: Given an LPNO G_O, a marking M is said to be *confusable* with M', denoted by $M \sim M'$, if there exists an observation $s \in \mathcal{L}(N, M_0)$ s.t. $M, M' \in \mathcal{C}(s)$. ⋄

One can readily verify that $M \sim M'$ is a symmetric, reflexive but not transitive relation. An intuitive way to compute the confusion relation among all markings is to construct an *observer* [1]. First, since the net is bounded, its reachability graph (RG) can be constructed. This is a graph where each node is a marking M and each arc

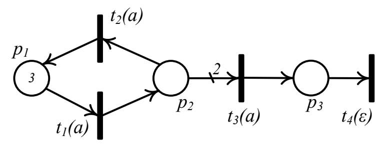
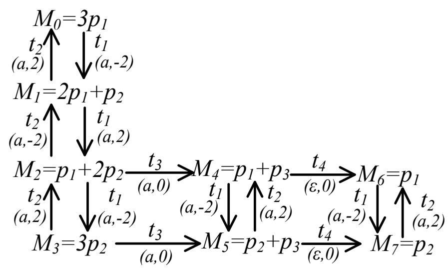
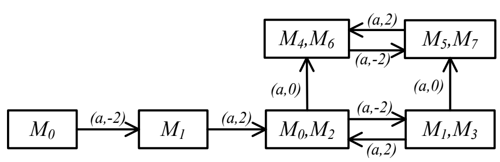


Fig. 11. LPNO with a nonlinear output function.



(a)



(b)

Fig. 12. The RG (a) and the observer of the LPNO (b).

corresponds to a transition t. We tag each arc t exiting node M with the label $(\ell(t), \Delta f(M,t))$, thus constructing a nondeterministic finite automaton (NFA). Then, the corresponding observer, i.e., the equivalent deterministic finite automaton (DFA), can be constructed. Each state of the DFA corresponds to a set $\mathcal{C}(s)$ and all markings in $\mathcal{C}(s)$ are confusable with each other.

Example 10: Let us consider the LPNO G_O in Fig. 11, where $M_0 = [3\ 0]^T$ and the output function is

$$f(M) = \begin{cases} 1 & if\ M(p_2)\ is\ an\ even\ number; \\ -1 & otherwise. \end{cases}$$

Its RG[2] and the observer are shown in Fig. 12. Hence, the confusion relations between reachable markings are: $M_0 \sim M_2$, $M_1 \sim M_3$, $M_4 \sim M_6$ and $M_5 \sim M_7$. ⋄

It is known that the worst-case complexity of computing a DFA equivalent to an NFA is exponential with respect to the number of states of the NFA. Therefore, the complexity to determine the confusion relation is exponential with respect to the number of markings.

Remark: there may exist more efficient ways to determine the confusion relation. Such a case is discussed in Section VII.

[2]For clarity, the corresponding transition is also labeled on the arcs.

C. Construction of the Agreement and Conflict Graphs

If two transitions t and t' of an LPNO may fire at two confusable markings M and M', respectively, and produce the same non-null observation $(e, \Delta f)$, then the two labels $\ell_A(M,t)$ and $\ell_A(M',t')$ must coincide in the equivalent ALPN. Furthermore, any transition t that may fire at a marking M producing the null observation $(\varepsilon, 0)$ should receive a label $\ell_A(M,t) = \varepsilon$ in the equivalent ALPN. These two types of constraints can be captured by a graph whose nodes are marking-transition pairs $[M,t]$ and whose edges connect nodes that should have the same label in the equivalent ALPN.

Definition 14: Given an LPNO G_O, its *agreement graph* is an undirected graph $\mathcal{A} = (V, E)$ whose set of vertexes is $V = \{[M,t] \in R(N, M_0) \times T | M[t\rangle\}$ and whose set of edges is $E = E' \cup E''$ where

$$\begin{aligned} E' =& \{([M,t],[M',t']) \in V \times V | [M,t] \neq [M',t'], \\ & (\ell(t), \Delta f(M,t)) = (\ell(t'), \Delta f(M',t')) = (\varepsilon, 0)\} \end{aligned}$$

and

$$\begin{aligned} E'' =& \{([M,t],[M',t']) \in V \times V | \\ & [M,t] \neq [M',t'], M \sim M', \\ & (\ell(t), \Delta f(M,t)) = (\ell(t'), \Delta f(M',t')) \neq (\varepsilon, 0)\}. \end{aligned}$$

$\diamond$

In an agreement graph there are two types of arcs E' and E''. Arcs in E' connect all pairs $[M,t]$ that produce the null observation; arcs in E'' connect pairs $[M,t]$ where markings are confusable and the firings of transitions produce the same non-null observation. Note that there is no self-loop in an agreement graph. After the confusion relation has been determined, the complexity of constructing the agreement graph is $\mathcal{O}(|V|^2)$, since in the worst case, computing the set of edges requires checking $|V|^2$ pairs of nodes $[M,t]$ and $[M't']$.

Example 11: Consider Example 10 again. In order to clearly illustrate all possible observations, Table III is built. Based on the confusion relations obtained in Example 10 and Table III, the agreement graph in Fig. 13 is constructed. To give an example of its construction consider M_0 and M_2. From Example 10, M_0 and M_2 are confusable. From Table III, $[M_0, t_1], [M_2, t_1]$ and $[M_2, t_2]$ produce the same observation. Therefore, by Definition 14, these three nodes in the agreement graph are connected by arcs in E''. Markings M_4 and M_5 are not confusable, however, nodes $[M_4, t_4]$ and $[M_5, t_4]$ are connected by arcs in E' since they produce the null observation $(\varepsilon, 0)$. $\diamond$

We now consider the connected components of the agreement graph and partition its set of nodes as

$$V = \hat{v}_0 \,\dot\cup\, \hat{v}_1 \,\dot\cup\, \hat{v}_2 \,\dot\cup \cdots \dot\cup\, \hat{v}_l$$

where for $i \in \{0, 1, 2, \cdots, l\}$, the $\hat{v}_i$-induced subgraph is a component of $\mathcal{A}$ and in particular

$$\hat{v}_0 = \{[M,t] \in V | \ell(t) = \varepsilon, \Delta f(M,t) = 0\}$$

TABLE III

ALL POSSIBLE OBSERVATIONS AT EACH MARKING

(e,Δ)	$\{[M,t] \vert \ell(t)=e, \Delta f(M,t)=\Delta\}$
$(a,-2)$	$[M_0,t_1],[M_2,t_1],[M_2,t_2],[M_4,t_1],[M_6,t_1]$
$(a,2)$	$[M_1,t_1],[M_1,t_2],[M_3,t_2],[M_5,t_2],[M_7,t_2]$
$(a,0)$	$[M_2,t_3],[M_3,t_3]$
$(\varepsilon,0)$	$[M_4,t_4],[M_5,t_4]$

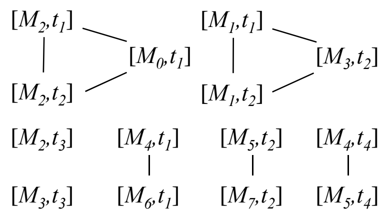


Fig. 13. Agreement graph $\mathcal{A}$.

is the (possibly empty) set of pairs $[M,t]$ that produce the null observation. Correspondingly we define the partition

$$\hat{V} = \{\hat{v}_0, \hat{v}_1, \hat{v}_2, \cdots, \hat{v}_l\}. \tag{1}$$

Example 12: Consider Example 10 again. Based on the agreement graph, we have $\hat{V} = \{\hat{v}_0, \hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}_4, \hat{v}_5, \hat{v}_6\}$, where $\hat{v}_0 = \{[M_4,t_4],[M_5,t_4]\}$, $\hat{v}_1 = \{[M_0,t_1],[M_2,t_1],[M_2,t_2]\}$, $\hat{v}_2 = \{[M_1,t_1],[M_1,t_2],[M_3,t_2]\}$, $\hat{v}_3 = \{[M_2,t_3]\}$, $\hat{v}_4 = \{[M_3,t_3]\}$, $\hat{v}_5 = \{[M_4,t_1],[M_6,t_1]\}$, and $\hat{v}_6 = \{[M_5,t_2],[M_7,t_2]\}$. ◇

By means of the agreement graph, we have determined the classes of pairs $[M,t]$ that produce the same observation. We now determine, by means of the conflict graph, which classes must be assigned a different label in the ALPN.

Definition 15: Given an LPNO G_O, the *conflict graph* $\hat{\mathcal{A}} = (\hat{V}, \hat{E})$ is an undirected graph whose set of vertexes is $\hat{V}$ as defined in Eq. (1) and whose set of edges is $\hat{E} = \hat{E}' \cup \hat{E}''$ where

$$\hat{E}' = \{(\hat{v}_0, \hat{v}_i) \vert \hat{v}_i \in \hat{V}, i \in \{1,2,\cdots,l\}\}$$

and

$$\begin{aligned}\hat{E}'' =& \{(\hat{v}_i, \hat{v}_j) \in \hat{V} \times \hat{V} \vert i,j \in \{1,2,\cdots,l\}, \\ & \exists [M,t] \in \hat{v}_i, \exists [M',t'] \in \hat{v}_j : \\ & M \sim M', (\ell(t), \Delta f(M,t)) \neq (\ell(t'), \Delta f(M',t'))\}\end{aligned}$$

◇

Note that $\hat{v}_0$ may not exist, i.e., $\hat{v}_0 = \emptyset$. In this case, $\hat{E}' = \emptyset$ and $\hat{E} = \hat{E}''$. The nodes of graph $\hat{\mathcal{A}}$ are classes of nodes $[M, t] \in V$ that produce the same observation. There are also two types of arcs in a conflict graph: $\hat{E}'$ and $\hat{E}''$. Since pairs $[M, t] \in \hat{v}_0$ must be assigned the empty word different from any label from the alphabet, arcs from $\hat{E}'$ connect node $\hat{v}_0$ with every other node; if there exist $[M, t] \in \hat{v}_i$ and $[M', t'] \in \hat{v}_j$ such that M and M' are confusable but t and t' will produce different observations $(e, \Delta f(M, t))$, $(e', \Delta f(M', t'))$, then in the ALPN, different labels must be assigned to them, i.e., $\ell_A(M, t) \neq \ell_A(M', t')$. Thus arcs from $\hat{E}''$ connect such two nodes $\hat{v}_i$ and $\hat{v}_j$.

The complexity of computing the connected components of a graph is known to be linear with respect to the number of edges of a graph using either breadth-first search (BFS) or depth-first search (DFS), i.e., the computation of $\hat{V}$ is $\mathcal{O}(|E|)$. In the worst case, computing the set of edges requires checking $|\hat{V}|^2$ pairs of nodes $\hat{v}_i$ and $\hat{v}_j$. Therefore, based on the agreement graph, the complexity of constructing the conflict graph is $\mathcal{O}(|\hat{V}|^2)$.

D. Solving the Vertex Coloring Problem

The conflict graph $\hat{\mathcal{A}}$ of an LPNO exactly describes the relabeling rule following which an equivalent ALPN can be obtained. We will show that given a bounded LPNO G_O, a vertex coloring of its conflict graph determines an equivalent ALPN and vice versa. Let us first formally define the notion of a vertex coloring.

Definition 16: Given a graph $\hat{\mathcal{A}} = (\hat{V}, \hat{E})$, a *vertex coloring* is a pair $(\Sigma_{col}, \ell_{col})$ where Σ_{col} is a finite set of colors and $\ell_{col} : \hat{V} \to \Sigma_{col}$ is a coloring function that assigns to each vertex a color and satisfies the constraint that if $(\hat{v}, \hat{v}') \in \hat{E}$ then $\ell_{col}(\hat{v}) \neq \ell_{col}(\hat{v}')$, i.e., two adjacent vertexes cannot be assigned the same color.

The *vertex coloring problem* is the problem of finding a vertex coloring with a *minimal* number of colors, which is called the *chromatic number* of $\hat{\mathcal{A}}$, denoted by $\chi(\hat{\mathcal{A}})$. A graph is called k*-chromatic*, if its chromatic number is k. ◇

Proposition 6: Let $G_O = (N, M_0, \Sigma, \ell, f)$ be a bounded LPNO with conflict graph $\hat{\mathcal{A}} = (\hat{V}, \hat{E})$. An ALPN $G_A = (N, M_0, \Sigma_A, \ell_A)$ is equivalent to G_O if and only if there exists a vertex coloring $(\Sigma_{col}, \ell_{col})$ of $\hat{\mathcal{A}}$ such that $\Sigma_A = \Sigma_{col} \setminus \{\varepsilon\}$ and $[M, t] \in \hat{v}$ with $\hat{v} \in \hat{V} \Rightarrow \ell_A(M, t) = \ell_{col}(\hat{v})$ holds.

Proof: ($\Rightarrow$) To prove the sufficiency of the statement, we show that an ALPN G_A whose adaptive labeling function is defined by a vertex coloring of $\hat{\mathcal{A}}$ is equivalent to G_O, namely $\forall \sigma \in \mathcal{L}(N, M_0)$, $\mathcal{S}(L_O(\sigma)) = \mathcal{S}(L_A(\sigma))$. This is done by induction on the length of firing sequences.

(Basis step) For any $\sigma \in \mathcal{L}(N, M_0)$ of length 0, observations in G_O and G_A are $s = L_O(\sigma) = (\varepsilon, 0)$ and $w_A = L_A(\sigma) = \varepsilon$, respectively. Let $\sigma' = t_1 t_2 \cdots t_k$ with $M_0[t_1\rangle M_1[t_2\rangle M_2 \cdots M_{k-1}[t_k\rangle M_k$.

- First we prove $\mathcal{S}(s) \subseteq \mathcal{S}(w_A)$. Assume $\sigma' \in \mathcal{S}(s)$. It satisfies $\ell(t_i) = \varepsilon$ and $f(M_i) = f(M_0)$, $i = 1, 2, \cdots, k$. According to the definition of $\hat{v}_0$ and the obtained vertex coloring, we have $[M_{i-1}, t_i] \in \hat{v}_0$ and $\ell_{col}(\hat{v}_0) = \varepsilon$, i.e., $\ell_A(M_{i-1}, t_i) = \varepsilon$. Thus, $L_A(\sigma') = \varepsilon$, i.e., $\sigma' \in \mathcal{S}(\varepsilon)$, and $\mathcal{S}(s) \subseteq \mathcal{S}(w_A)$.
- Next we prove $\mathcal{S}(w_A) \subseteq \mathcal{S}(s)$. Let $\sigma' \in \mathcal{S}(w_A)$. Then we have $\ell_A(M_{i-1}, t_i) = \varepsilon$ and $[M_{i-1}, t_i] \in \hat{v}_0$. Therefore, in G_O, $\ell(t_i) = \varepsilon$ and $\Delta f(M_{i-1}, t_i) = 0$ that implies $L_O(\sigma') = (\varepsilon, 0)$, i.e., $\sigma' \in \mathcal{S}(s)$.

As a result, $\mathcal{S}(s) = \mathcal{S}(w_A)$.

(Inductive step) Assume that for any $\sigma \in \mathcal{L}(N, M_0)$ of length k, $\mathcal{S}(L_O(\sigma)) = \mathcal{S}(L_A(\sigma))$ holds. In the following, we prove that this is also true for firing sequences of length $k+1$.

Let $\sigma = \sigma_0 t$ with $M_0[\sigma_0\rangle M_1[t\rangle M_2$, where $|\sigma_0| = k$, $s = L_O(\sigma) = L_O(\sigma_0 t) = s_0(e_1, \Delta)$ and $w_A = L_A(\sigma) = L_A(\sigma_0 t) = w_0 e_2$. In other words, $L_O(\sigma_0) = s_0$, $\ell(t) = e_1$, $\Delta f(M_1, t) = \Delta$, $L_A(\sigma_0) = w_0$ and $\ell_A(M_1, t) = e_2$. Let $\sigma' = \sigma'_0 \sigma'_1$ with $\sigma'_1 = t'_1 t'_2 \cdots t'_k$ and $M_0[\sigma'_0\rangle M'_0[t'_1\rangle M'_1 \cdots M'_{k-1}[t'_k\rangle M'_k$.

- Assume $\sigma' \in \mathcal{S}(s)$ and $\sigma'_0 \in \mathcal{S}(s_0)$. Then there exists $j \in \{1, 2, \cdots, k\}$ such that $\ell(t'_j) = e_1$ and $\Delta f(M'_{j-1}, t'_j) = \Delta$. However, $\forall i \in \{1, 2, \cdots, k\}$ with $i \neq j$, $\ell(t_i) = \varepsilon$ and $\Delta(M'_{i-1}, t'_i) = 0$. According to the definition of $\hat{v}_0$ and the coloring rule, $[M'_{i-1}, t'_i] \in \hat{v}_0$ and in the obtained ALPN $\ell_A(M'_{i-1}, t'_i) = \varepsilon$. Since $\sigma_0, \sigma'_0 t'_1 t'_2 \cdots t'_{j-1} \in \mathcal{S}(s_0)$, M_1 and M'_{j-1} are confusable, i.e., $M_1 \sim M'_{j-1}$. Meanwhile, $(\ell(t), \Delta f(M_1, t)) = (\ell(t'_j), \Delta f(M'_{j-1}, t'_j)) = (e_1, \Delta)$ and hence $[M_1, t]$ and $[M'_{j-1}, t'_j]$ are in a same node of the conflict graph of G_O that indicates in the obtained ALPN $\ell_A(M'_{j-1}, t'_j) = \ell_A(M_1, t) = e_2$. By induction, $\sigma'_0 \in \mathcal{S}(w_0)$, and therefore, $L_A(\sigma'_0 \sigma'_1) = w_0 e_2$ and $\sigma' \in \mathcal{S}(w_A)$, i.e., $\mathcal{S}(s) \subseteq \mathcal{S}(w_A)$.
- Analogously, it can be proved $\mathcal{S}(w_A) \subseteq \mathcal{S}(s)$. Assume $\sigma' \in \mathcal{S}(w_A)$ and $\sigma'_0 \in \mathcal{S}(w_0)$. Then there exists $j \in \{1, 2, \cdots, k\}$ such that $\ell_A(M'_{j-1}, t'_j) = e_2$ and $\forall i \in \{1, 2, \cdots, k\}$ with $i \neq j$, $\ell_A(M'_{i-1}, t'_i) = \varepsilon$. Based on the vertex coloring, in the LPNO we have $\ell(t_i) = \varepsilon$ and $\Delta f(M'_{i-1}, t'_i) = 0$. Since $\ell_A(M'_{j-1}, t'_j) = e_2$, by induction $\sigma'_0 \in \mathcal{S}(s_0)$ which means that $\sigma_0, \sigma'_0 t'_1 t'_2 \cdots t'_{j-1} \in \mathcal{S}(s_0)$ and $M_1 \sim M'_{j-1}$, $[M_1, t]$ and $[M'_{j-1}, t'_j]$ are in a same node of the conflict graph of G_O. Therefore, $\ell(t'_j) = \ell(t) = e_1$, $\Delta f(M'_{j-1}, t'_j) = \Delta f(M_1, t) = \Delta$ and $L_O(\sigma'_0 \sigma'_1) = L_O(\sigma') = s_0(e_1, \Delta)$, i.e., $\sigma' \in \mathcal{S}(s)$.

The result follows by induction.

($\Leftarrow$) We prove by contradiction the necessity of the statement. Let $G_A = (N, M_0, \Sigma_A, \ell_A)$ be an ALPN equivalent to G_O. Assume that the adaptive labeling function of G_A is not defined by a vertex coloring to $\hat{\mathcal{A}}$, i.e., there exist $[M, t] \in \hat{v}_i$ and $[M', t'] \in \hat{v}_j$ such that $\ell_A(M, t) = \ell_A(M', t')$ and $(\hat{v}_i, \hat{v}_j) \in \hat{E}$. Since $\hat{v}_i$ and $\hat{v}_j$ are adjacent, according to the definition of conflict graphs, there are two possibilities in G_O: i) $M \sim M'$ and $(\ell(t), \Delta f(M, t)) \neq (\ell(t'), \Delta f(M', t'))$; and ii) $(\ell(t), \Delta f(M, t)) = (\varepsilon, 0)$ and $(\ell(t'), \Delta f(M', t')) \neq (\varepsilon, 0)$ (or $(\ell(t'), \Delta f(M', t')) = (\varepsilon, 0)$ and $(\ell(t), \Delta f(M, t)) \neq (\varepsilon, 0)$). For case i), since M and M' are confusable, there exist firing sequences σ and σ' such that $M_0[\sigma\rangle M$, $M_0[\sigma'\rangle M'$ and $L_O(\sigma) = L_O(\sigma') = s$. Therefore, we have $\sigma t \in \mathcal{S}(L_O(\sigma t))$ but $\sigma t \notin \mathcal{S}(L_O(\sigma' t'))$. Assume that the corresponding observation of σ in G_A is w_A. Since G_A is equivalent to G_O, $\sigma t \in \mathcal{S}(L_A(\sigma' t'))$ holds, which implies that, however, $\mathcal{S}(L_A(\sigma' t')) \neq \mathcal{S}(L_O(\sigma' t'))$ and G_A is not equivalent to G_O. Then, we reach a contradiction. Case ii) can be proved analogously.

■

Based on the previous results, the ALPN with a minimal alphabet equivalent to a given LPNO can be obtained by solving a vertex coloring problem, i.e, finding a vertex coloring such that the number of colors is minimal. The general procedure to convert a bounded LPNO into an equivalent ALPN with a minimal alphabet is summarized in Algorithm 1.

Since Steps 2 and 3 have polynomial complexity $\mathcal{O}(|V|^2)$ and $\mathcal{O}(|\hat{V}|^2)$, respectively, as we have discussed in the previous sections, the complexity to convert a bounded LPNO into an equivalent ALPN with a minimal alphabet

Algorithm 1 Conversion of a bounded LPNO into an equivalent ALPN with a minimal alphabet

Input: a bounded LPNO $G_O = (N, M_0, \Sigma, \ell, f)$

Output: an equivalent ALPN $G_A = (N, M_0, \Sigma_A, \ell_A)$

Compute the confusion relation.

Construct $\mathcal{A}$ according to Definition 14.

Construct $\hat{\mathcal{A}}$ according to Definition 15.

Solve the vertex coloring problem of $\hat{\mathcal{A}}$.

$\Sigma_A := \Sigma_{col} \setminus \{\varepsilon\}$, $\ell_A := \ell_{col}$.

Output G_A.

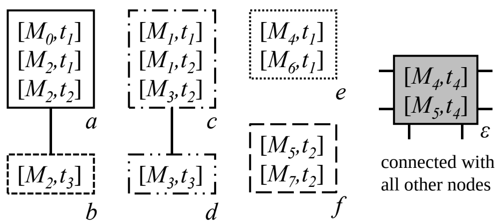


Fig. 14. Colored conflict graph $\hat{\mathcal{A}}$.

mainly depends on the computation of the confusion relation and on solving the vertex coloring problem, which is known to be in general NP-complete. In the worst case, the RG and the corresponding observer have to be constructed. Note that in general there is no obvious relation between the size of a net (i.e., the number of places, transitions and tokens that the initial marking assigned to the places) and that of its RG. Therefore, the size of the RG cannot be a priori determined based on the structure of the net. However, in Section VII we show that in some cases without computing the RG the conflict graph can be constructed by just characterizing the output function. Meanwhile, for some special classes of graphs, for example, perfect graphs, the vertex coloring problem can be solved in polynomial time with respect to the number of nodes of the graph (see more results in [24]). Solving the vertex coloring is needed only if one aims to find an equivalent ALPN with a minimal alphabet. On the contrary, the computation of a vertex coloring — not necessarily minimal — is polynomial: one trivial solution is to color every vertex of the conflict graph in different colors and there exist suboptimal solutions with polynomial complexity [24], the greedy algorithm for instance.

Example 13: The colored conflict graph of the LPNO in Example 10 is shown in Fig. 14 (different colors are denoted by different boxes around the nodes), which is a trivial way to color the graph. The equivalent ALPN is $\ell_A(M_0,t_1) = \ell_A(M_2,t_1) = \ell_A(M_2,t_2) = a$, $\ell_A(M_2,t_3) = b$, $\ell_A(M_1,t_1) = \ell_A(M_1,t_2) = \ell_A(M_3,t_2) = c$, $\ell_A(M_3,t_3) = d$, $\ell_A(M_4,t_1) = \ell_A(M_6,t_1) = e$, $\ell_A(M_5,t_2) = \ell(M_7,t_2) = f$ and $\ell_A(M_4,t_4) = \ell_A(M_5,t_4) = \varepsilon$; the alphabet is $\Sigma_A = \{a,b,c,d,e,f\}$.

Nevertheless, the coloring problem of graph $\hat{\mathcal{A}}$ can be solved by using three colors. Thus, the equivalent ALPN with a minimal alphabet is $\forall M \in R(N, M_0)$, $\ell_A(M, t_1) = \ell_A(M, t_2) = a$, $\ell_A(M, t_3) = b$ and $\ell_A(M, t_4) = \varepsilon$. This ALPN is also an LPN.

If we apply the brute-force approach, according to Table III, the equivalent ALPN is $\ell_A(M_0, t_1) = \ell_A(M_2, t_1) = \ell_A(M_2, t_2) = \ell_A(M_4, t_1) = \ell_A(M_6, t_1) = [a, -2]$, $\ell_A(M_1, t_1) = \ell_A(M_1, t_2) = \ell_A(M_3, t_2) = \ell_A(M_5, t_2) = \ell_A(M_7, t_2) = [a, 2]$, $\ell_A(M_2, t_3) = \ell_A(M_3, t_3) = [a, 0]$, $\ell_A(M_4, t_4) = \ell_A(M_5, t_4) = \varepsilon$ and the alphabet is $\Sigma_A = \{[a, -2], [a, 2], [a, 0]\}$. ⋄

Note that Algorithm 1 is a general procedure that can be applied to any arbitrary bounded LPNO. For some special subclasses, e.g., LPNAFs, the conversion from LPNOs to ALPNs has polynomial complexity (trivially follows from the proof of Proposition 1). However, this method cannot assure a minimal alphabet for the LPN. In some cases, even the brute-force approach may provide a fast way to compute the equivalent ALPN, especially for LPNOs with very simple output functions. However, the alphabet of the obtained ALPN is not necessarily minimal and many redundant labels may be introduced. To eliminate redundant labels, further analysis on the confusion relation is required, i.e., the vertex-coloring-based approach is needed.

VI. Conversion of Bounded LPNOs into LPNs

The results in the previous section show that how any bounded LPNO can be converted into an equivalent ALPN not only with a finite alphabet, but with a minimal alphabet. This however does not ensure the existence of an equivalent LPN. In this section, for bounded LPNOs, a sufficient and necessary condition for the existence of an equivalent LPN is proposed. If the condition is satisfied, the LPNO can be converted into an equivalent LPN by applying the algorithm presented in this section.

Considering that LPNs are a special case of ALPNs, a necessary condition for the existence of an equivalent LPN is obtained.

Proposition 7: Let $G_O = (N, M_0, \Sigma, \ell, f)$ be a bounded LPNO whose conflict graph is k-chromatic. If $|T| < k$, there is no LPN equivalent to G_O.

Proof: Assume that there is an LPN $G_L = (N, M_0, \Sigma, \ell)$ equivalent to G_O. Then the maximal number of labels of G_L is $|T|$, i.e., $|\Sigma| \leq |T|$. Since the conflict graph of G_L is k-chromatic, there is an equivalent ALPN $G_A = (N, M_0, \Sigma_A, \ell_A)$ with $|\Sigma_A| = k$. Considering LPNs are a special class of ALPNs, we have $|\Sigma_A| \leq |\Sigma| \leq |T|$. Then the proposition holds. ■

The following counter example shows that the condition is not sufficient.

Example 14: Consider the LPNO and the corresponding RG in Fig. 15. By applying Algorithm 1 and solving the vertex coloring problem, the colored conflict graph is shown in Fig. 16. The equivalent ALPN with a minimal alphabet is $\ell_A(M_1, t_1) = \ell_A(M_2, t_2) = \varepsilon$, $\ell_A(M_0, t_1) = \ell_A(M_1, t_2) = \ell_A(M_3, t_1) = \ell_A(M_4, t_2) = \alpha$ and $\Sigma_A = \{\alpha\}$. It satisfies $|\Sigma_A| < |T|$ but there is no LPN equivalent to the LPNO, since all vertex colorings of $\hat{\mathcal{A}}$ correspond to ALPNs. ⋄

By characterizing the conflict graph, a sufficient and necessary condition that verifies the existence of an equivalent LPN is proposed. First, we introduce some new notations for the conflict graph $\hat{\mathcal{A}} = (\hat{V}, \hat{E})$ of a given LPNO G_O.

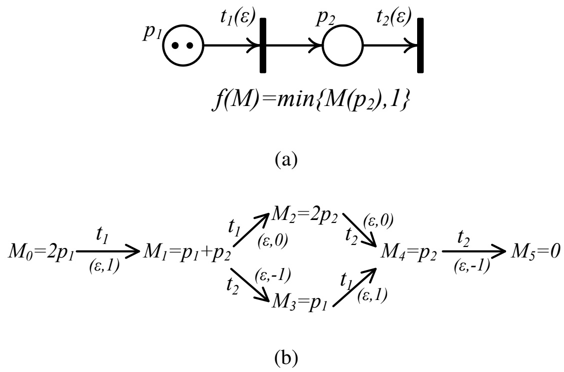


Fig. 15. LPNO without equivalent LPN (a) and its RG (b).

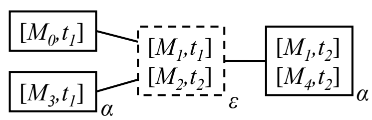


Fig. 16. Colored conflict graph $\hat{\mathcal{A}}$.

For a transition $t \in T$, the notation $[\cdot, t]$ denotes a marking-transition pair $[M, t]$ without specifying marking M. The set $T_c(t)$ of a given transition t is defined as

$$T_c(t) = \{t' \in T | \exists \hat{v} \in \hat{V} : [\cdot, t], [\cdot, t'] \in \hat{v})\};$$

If a transition $t' \in T_c(t)$, there exists a node $\hat{v} \in \hat{V}$ to which both $[\cdot, t]$ and $[\cdot, t']$ belong. The set $T_c(t)$ of t is a nonempty set as $t \in T_c(t)$. According to the analysis in the previous section, in the equivalent ALPN, transitions $t' \in T_c(t)$ will be assigned the same label of transition t at some markings. The set $T_l(t)$ of a given transition t is defined as

$$\begin{aligned} T_l(t) = \{ \quad & t' \in T | \exists \hat{v}_i, \hat{v}_j \in \hat{V} : \\ & [\cdot, t] \in \hat{v}_i, [\cdot, t'] \in \hat{v}_j, (\hat{v}_i, \hat{v}_j) \in \hat{E}\}. \end{aligned}$$

If $t' \in T_l(t)$, in $\hat{\mathcal{A}}$ there are two adjacent nodes $\hat{v}_i$ and $\hat{v}_j$ that contain $[\cdot, t]$ and $[\cdot, t']$, respectively. Therefore, there are markings at which transitions t' and t are assigned different labels in the equivalent ALPN.

Now we discuss the complexity of computing sets $T_c(t)$ and $T_l(t)$ of a given transition t. To compute $T_c(t)$, we first compute the set of nodes $\hat{v}_i \in \hat{V}$ such that $[\cdot, t] \in \hat{v}_i$. The transitions t' of which $[\cdot, t'] \in \hat{v}_i$, belong to $T_c(t)$. Therefore, the complexity of computing $T_c(t)$ is $\mathcal{O}(|\hat{V}|)$. On the other hand, to compute $T_l(t)$, first we select a node $\hat{v}_i \in \hat{V} : \exists [\cdot, t] \in \hat{v}_i$ and then compute a set of nodes $\hat{v}_j \in \hat{V}$ such that there is a edge between $\hat{v}_i$ and $\hat{v}_j$. Finally, the transitions t' of which $[\cdot, t'] \in \hat{v}_j$, belong to $T_l(t)$. Therefore, the complexity of computing $T_l(t)$ is $\mathcal{O}(|\hat{V}|^2)$.

Example 15: Consider the conflict graph in Fig. 14. We have $T_c(t_1) = T_c(t_2) = \{t_1, t_2\}$, $T_c(t_3) = \{t_3\}$, $T_c(t_4) = \{t_4\}$, $T_l(t_1) = T_l(t_2) = \{t_3, t_4\}$, $T_l(t_3) = \{t_1, t_2, t_4\}$ and $T_l(t_4) = \{t_1, t_2, t_3\}$. ◇

Proposition 8: Given an LPNO G_O and its conflict graph $\hat{\mathcal{A}} = (\hat{V}, \hat{E})$, there exists an LPN equivalent to G_O if and only if $T_c(t) \cap T_l(t) = \emptyset$ holds, $\forall t \in T$.

Proof: If the LPNO G_O satisfies $T_c(t) \cap T_l(t) = \emptyset$, then no transition has to be assigned to different labels at different markings. Thus there is a vertex coloring that corresponds to an equivalent LPN. Suppose $T_c(t) \cap T_l(t) \neq \emptyset$. Let $t' \in T_c(t) \cap T_l(t)$. There is a node $\hat{v}_i \in \hat{V}$ that includes $[M_a, t]$ and $[M_b, t']$. Therefore $\ell_A(M_a, t) = \ell_A(M_b, t')$. Since $t' \in T_l(t)$, there are adjacent nodes $\hat{v}_j$ and $\hat{v}_k$ in $\hat{\mathcal{A}}$ where $[M_c, t] \in \hat{v}_j$ and $[M_d, t'] \in \hat{v}_k$. We have $\ell_A(M_c, t) \neq \ell_A(M_d, t')$. Hence there exists no vertex coloring corresponding to an LPN and based on Proposition 6, there is no equivalent LPN. ■

Note that $\forall t' \in T_c(t) \cap T_l(t)$, t' is adaptively labeled in the equivalent ALPN. If an LPNO satisfies Proposition 8, there exists a vertex coloring by which the equivalent LPN can be computed. Given a transition $t \in T$, the nodes in $\hat{\mathcal{A}}$ containing $[\cdot, t]$ can be merged as one, since $[\cdot, t]$ can be in the same label. To obtain a vertex coloring that corresponds to an LPN, the set of vertexes $\hat{V}$ needs to be reconstructed and Algorithm 2 realizes such a reconstruction.

Algorithm 2 Reconstruction of $\hat{V}$

Input: the set $\hat{V}$ of $\hat{\mathcal{A}}$

Output: a new set $\hat{V}_{new}$

$\hat{V}_{new} := \hat{V}$
for all $\hat{v}_i \in \hat{V}_{new}$, **do**
  for all $\hat{v}_j \in \hat{V}_{new} \setminus \{\hat{v}_i\}$, **do**
    if $\exists [\cdot, t] \in \hat{v}_i : [\cdot, t] \in \hat{v}_j$, **then**
      $\hat{v}_i = \hat{v}_i \cup \hat{v}_j$;
      $\hat{V}_{new} = \hat{V}_{new} \setminus \{\hat{v}_j\}$;
    end if
  end for
end for
Output $\hat{V}_{new}$.

To obtain the final set $\hat{V}_{new}$, first we select a node $\hat{v}_i$ in $\hat{V}$ and find another node $\hat{v}_j \in \hat{V}$ such that $\hat{v}_i$ and $\hat{v}_j$ contain the same transition t. Then we merge $\hat{v}_i$ and $\hat{v}_j$ and remove $\hat{v}_j$ from $\hat{V}$. Note that the obtained node $\hat{v}_i$ will not be treated as a new node. Therefore, the complexity of Algorithm 2 is $\mathcal{O}(|\hat{V}|^2)$. As soon as the set $\hat{V}$ is rebuilt as $\hat{V}_{new}$, the conflict graph $\hat{\mathcal{A}}$ should also be reconstructed by Definition 15 (in order to avoid confusion, the reconstructed conflict graph is denoted as $\hat{\mathcal{A}}_{new}$). Then, by computing a vertex coloring of $\hat{\mathcal{A}}_{new}$ the equivalent LPN is obtained. In conclusion, the procedure of finding an equivalent LPN of a bounded LPNO is stated as follows:

Step 1 Construct the conflict graph $\hat{\mathcal{A}}$.

Step 2 Check if Proposition 8 is verified:

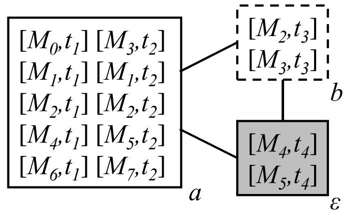


Fig. 17. Colored conflict graph $\hat{\mathcal{A}}_{new}$.

"Yes" — go to Step 3;

"No" — stop, as there is no equivalent LPN.

Step 3 Apply Algorithm 2.

Step 4 Construct the new conflict graph $\hat{\mathcal{A}}_{new}$ by Definition 15.

Step 5 Compute a vertex coloring of $\hat{\mathcal{A}}_{new}$.

Example 16: Example 15 shows that $\forall t \in T, T_c(t) \cap T_l(t) = \emptyset$, i.e., the LPNO in Fig. 11 satisfies Proposition 8 and thus there is an LPN equivalent to the LPNO. The conflict graph is reconstructed by applying Algorithm 2 and the colored one is shown in Fig. 17. Therefore, the equivalent LPN is $\ell(t_1) = \ell(t_2) = a$, $\ell(t_3) = b$, $\ell(t_4) = \varepsilon$ and $\Sigma = \{a, b\}$.

Consider the LPNO in Example 14. According to the conflict graph in Fig. 16, there is no LPN equivalent to it since $\forall t \in T, T_c(t) \cap T_l(t) = T$. Results in Example 14 also verify this. ◇

A. Further Discussion on the Number of Labels

It is known that the number of colors that can be used to color a graph is not unique, as well as the way of coloring it. If the conflict graph $\hat{\mathcal{A}} = (\hat{V}, \hat{E})$ of an LPNO G_O is k-chromatic, and $|\hat{V}| = \lambda$, the bound of labels of the equivalent ALPN is $k \leq |\Sigma_A| \leq \lambda$. Then, it is important to answer the question whether the lower bound of labels necessarily increases/decreases when an equivalent LPN is required.

Proposition 9: Given an LPNO satisfying Proposition 8, the minimal number of labels in equivalent LPNs is k if and only if the conflict graph $\hat{\mathcal{A}} = (\hat{V}, \hat{E})$ of G_O is k-chromatic.

Proof: Since the LPNO satisfies Proposition 8, there is an equivalent LPN and its conflict graph $\hat{\mathcal{A}} = (\hat{V}, \hat{E})$ can be reconstructed into $\hat{\mathcal{A}}_{new}$ by applying Algorithm 2 and Definition 15. The reconstruction of $\hat{V}$ does not change the coloring relation between $[M, t]$ pairs. That is to say, even though some $[M, t]$ pairs that are not necessarily in the same node in $\hat{\mathcal{A}}$ are absorbed into the same node of $\hat{\mathcal{A}}_{new}$, this does not violate the coloring rule since the nodes that belong to $\hat{\mathcal{A}}$ are not connected. Therefore, if $\hat{\mathcal{A}}$ is k-chromatic, so is $\hat{\mathcal{A}}_{new}$, i.e., the minimal number of labels of equivalent LPNs is k.

Proposition 6 shows that the vertex colorings of the conflict graph characterize all equivalent ALPNs. Since LPNs are a special class of ALPNs, if the minimal number of labels of equivalent LPNs is k, then $\hat{\mathcal{A}} = (\hat{V}, \hat{E})$ of G_O

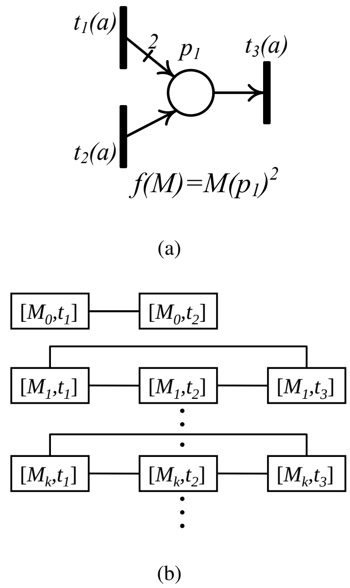


Fig. 18. Unbounded LPNO (a) and its conflict graph $\hat{\mathcal{A}}$ (b).

is k-chromatic. ■

Therefore, the bound of labels of the equivalent LPN is $k \leq |\Sigma| \leq |T|$. The requirement of equivalent LPNs does not change the minimal number of labels. Proposition 9 also implies that if there is no vertex coloring with the chromatic number of labels corresponding to an LPN, then there is no LPN equivalent to the LPNO.

VII. Conversion of Unbounded LPNOs

The conversion algorithms and propositions proposed in the previous sections are applicable to bounded LPNOs. For unbounded LPNOs, the conflict graph may not be feasible to be constructed and analyzed because of the infinite number of markings. Although we lack general results, we give an example to show that in some cases it is possible to convert an unbounded LPNO into an equivalent ALPN by using the same technique.

Example 17: Consider the LPNO in Fig. 18(a). Since at each marking, the firings of t_1, t_2 and t_3 produce different observations, no marking is confusable with others. The conflict graph $\hat{\mathcal{A}}$ is shown in Fig. 18(b), where $M_i = [i], i = 0, 1, 2, \cdots$. For any $t \in T$, $T_c(t) = \{t\}$ and $T_l(t) = T \setminus \{t\}$. Therefore, the LPNO satisfies Proposition 8. By applying Algorithms 1 and 2, the colored conflict graph $\hat{\mathcal{A}}_{new}$ is shown in Fig. 19 and the equivalent LPN is $\ell(t_1) = a_1, \ell(t_2) = a_2, \ell(t_3) = a_3$ and $\Sigma = \{a_1, a_2, a_3\}$. If we apply the brute force approach to obtain the equivalent ALPN, we need an infinite number of labels. ◇

Example 17 shows that even though the conflict graph is infinite, the alphabet of the equivalent ALPN could be finite. This result can be explained by the following theorem concerning the coloring problem in infinite graphs.

Theorem 1: [De Bruijin-Erdős theorem (1951)] If and only if all finite subgraphs of an infinite graph $\hat{\mathcal{A}}$ can be colored by ρ colors, then $\chi(\hat{\mathcal{A}}) \leq \rho$.

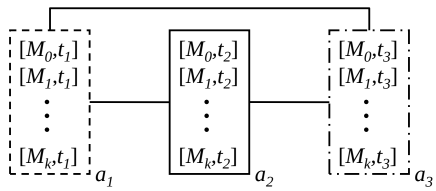


Fig. 19. Colored conflict graph $\hat{\mathcal{A}}_{new}$.

VIII. Conclusion and Future Work

In the paper different observation structures for Petri net generators are developed. In particular two classes of Petri net generators are defined: labeled Petri nets with outputs (LPNOs) and adaptive labeled Petri nets (ALPNs). The two classes are proper generalizations of labeled Petri nets (LPNs) usually considered in the literature. The notion of observation equivalence is formulated and used to compare the modeling power of different classes of Petri net generators. It is shown that LPNOs and ALPNs have the highest modeling power. Algorithms converting bounded LPNOs to equivalent ALPNs and LPNs with a minimal alphabet are proposed, whose complexity mainly depends on the computation of confusion relations and solving the vertex coloring problem of a particular graph that is called a conflict graph. In the case of unbounded LPNOs, the algorithms may also be applicable.

We believe that LPNOs provide an intuitive way to model systems with various kinds of sensors. However, it may be difficult to analyze the system behavior according to the information provided by the labeling function and output functions in a systematic way. This work addressing the conversion from LPNOs to equivalent LPNs provides some useful tools to analyze LPNOs.

The future work will focus on characterizing a class of LPNOs whose LPNs or ALPNs can be obtained with polynomial complexity, using the basis reachability graph introduced in [15] to reduce the conversion complexity, and finding a systematic way to analyze ALPNs.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61374068, 61472295, the Recruitment Program of Global Experts, the Science and Technology Development Fund, MSAR, under Grant No. 066/2013/A2.

References

[1] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Springer, 2008.

[2] A. Giua and C. Seatzu, “Observability of place/transition nets,” *IEEE Trans. Autom. Control*, vol. 47, no. 9, pp. 1424–1437, September 2002.

[3] A. Giua, C. Seatzu, and F. Basile, “Observer-based state-feedback control of timed Petri nets with deadlock recovery,” *IEEE Trans. Autom. Control*, vol. 49, no. 1, pp. 17–29, 2004.

[4] J. W. Bryans, M. Koutny, and P. Y. A. Ryan, “Modelling opacity using Petri nets,” *Electronic Notes in Theoretical Computer Science*, vol. 121, pp. 101–115, 2005.

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

[6] M. P. Cabasino, A. Giua, and C. Seatzu, “Diagnosis using labeled Petri nets with silent or undistinguishable fault events,” *IEEE Trans. Sys. Man Cybern., Syst.*, vol. 43, no. 2, pp. 345–355, 2013.

[7] T. Ushio, I. Onishi, and K. Okuda, “Fault detection based on Petri net models with faulty behaviors,” in *Proc. IEEE Conference on Systems, Man, and Cybernetics*, San Diego, USA, October 1998, pp. 113–118.

[8] A. Bourij and D. Koenig, “An original Petri net state estimation by a reduced luenberger observer,” in *Proceedings of American Control Conference*, vol. 3, San Dieago, USA, June 1999, pp. 1986–1989.

[9] S. L. Chung, “Diagnosing PN-based models with partial observable transitions,” *International Journal of Computer Integrated Manufacturing*, vol. 18, no. 2-3, pp. 158–169, 2005.

[10] Y. Ru and C. Hadjicostis, “Fault diagnosis in discrete event systems modeled by partially observed Petri nets,” *Discrete Event Dynamic Systems*, vol. 19, no. 4, pp. 551–575, December 2009.

[11] D. Lefebvre, “Diagnosis with Petri nets according to partial events and states observation,” in *IFAC Fault Detection, Supervision and Safety of Technical Processes*, vol. 8, no. 1, Mexico City, Mexico, October 2012, pp. 1244–1249.

[12] ——, “On-line fault diagnosis with partially observed petri nets,” *IEEE Trans. on Autom. Control*, vol. 59, no. 7, pp. 1919–1924, 2014.

[13] Y. Tong, Z. W. Li, and A. Giua, “General observation structures for Petri nets,” in *IEEE Conference on Emerging Technologies and Factory Automation*, Cagliari, Italy, September 2013.

[14] ——, “Observation equivalence of Petri net generators,” in *Proceedings of 12th Intern. Workshop on Discrete Event Systems*, vol. 12, Cachan, France, May 2014, pp. 338–343.

[15] M. Cabasino, A. Giua, and C. Seatzu, “Fault detection for discrete event systems using Petri nets with unobservable transitions,” *Automatica*, vol. 46, no. 9, pp. 1531–1539, September 2010.

[16] Y. Tong, Z. W. Li, C. Seatzu, and A. Giua, “Verification of current-state opacity using Petri nets,” in *Proceedings of 2015 American Control Conference (accepted)*, Chicago, USA, July 2015.

[17] T. Murata, “Petri nets: Properties, analysis and applications,” *Procedings of the IEEE*, vol. 77, no. 4, pp. 541–580, April 1989.

[18] M. Fanti, A. Mangini, and W. Ukovich, “Fault detection by labeled Petri nets in centralized and distributed approaches,” *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 2, pp. 392–404, April 2013.

[19] F. Cassez and S. Tripakis, “Fault diagnosis with dynamic observers,” in *Proceedings of 9th International Workshop on Discrete Event Systems*, Göteborg, Sweden, May 2008, pp. 212–217.

[20] T. Ushio and S. Takai, “Supervisory control of discrete event systems modeled by mealy automata with nondeterministic output functions,” in *Proceedings of 2009 American Control Conference*, St. Louis, MO, USA, June 2009, pp. 4260–4265.

[21] W. Wang, A. R. Girard, S. Lafortune, and F. Lin, “On codiagnosability and coobservability with dynamic observations,” *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1551–1566, 2011.

[22] L. K. Carvalho, J. C. Basilio, and M. V. Moreira, “Robust diagnosis of discrete event systems against intermittent loss of observations,” *Automatica*, vol. 48, no. 9, pp. 2068–2078, 2012.

[23] R. Diestel, *Graph theory*. New York: Springer, 2006.

[24] T. R. Jensen and B. Toft, *Graph coloring problems*. John Wiley & Sons, 2011, vol. 39.