12th IFAC/IEEE Workshop on Discrete Event Systems
Cachan, France. May 14-16, 2014

Modeling of timed Petri nets using deterministic (max,+) automata

Sébastien Lahaye * **Jan Komenda** ** **Jean-Louis Boimond** *

* *LUNAM Université, LARIS, Angers, France.*
sebastien.lahaye@univ-angers.fr, jean-louis.boimond@univ-angers.fr.
** *Institute of Mathematics - Brno Branch, Czech Academy of Sciences, Czech Republic.* komenda@ipm.cz

Abstract: Automata with weights (multiplicities) in (max,+) algebra form a class of timed automata. Determinism is a crucial property for numerous results on (max,+) automata and, in particular, for applications to performance evaluation and control of a large class of timed discrete event systems. In this paper, we show how to build a deterministic (max,+) automaton equivalent to a live and safe timed Petri net in which, between any two transitions, there exists an oriented path which contains at most one "conflict place".

Keywords: Petri nets, (max,+) automata, modeling, determinization

1. INTRODUCTION AND MOTIVATIONS

An important class of timed discrete-event systems (TDES) can be captured by means of timed Petri nets (TPNs) with deterministic timing of places and/or transitions [16]. A corresponding class of timed automata with deterministic timing of transitions is known as (max,+) automata [5] that are automata with weights (multiplicities) in (max,+) algebra [1].

(Max,+) automata have been applied to performance evaluation [5,14,17,3], scheduling [8] and control [11,2] problems for a large class of TDES. Beyond the scope of TDES, there are important applications for image and speech processing [15], and more generally, weighted automata constitute a theoretical object extensively studied (see [4] for an overview).

The modeling power of (max,+) automata is studied in [7] where it is shown that the behavior of any safe [2] TPN can be expressed by a *heap model*. It has been shown that the height of heaps of pieces is recognized by a particular (max,+) automaton (especially useful for algebraic computations). The (max,+) automaton derived from the example of heap model is depicted [3] on the right-hand side of Figure 1.

Another approach in [12] proposes a direct transformation of any safe TPN into a (max,+) automaton. For example and as reminded in section 3, the (max,+) automaton of Figure 2 is obtained to represent the TPN of Figure 3.

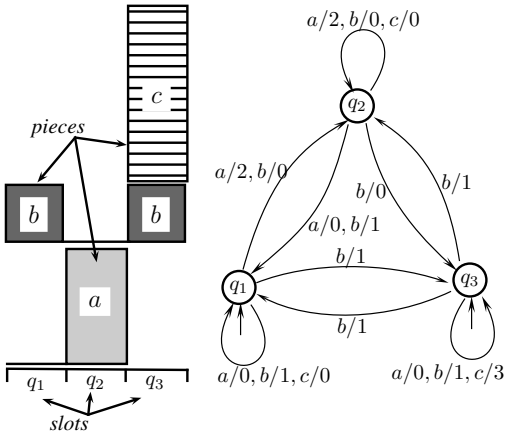

Fig. 1. Heap of pieces and associated (max,+) automaton to represent TPN $\mathcal{G}$ of Figure 3.

Observe that these (max,+) automata have larger languages [4] than the language of the TPN. For example, they recognize sequence $aaa\ldots$ whereas consecutive firings of transition a (without firing b) are impossible. As it is done classically in automata theory, it is then possible to adequately restrict the (max,+) automata languages by using the tensor product with the (Boolean) marking automaton recognizing the TPN language.

In addition, these (max,+) automata are both nondeterministic [5]. Except for trivial examples, this situation always occurs with the two approaches and this is a serious drawback. Determinism is indeed important for

[1] Set $\mathbb{Z} \cup \{-\infty\}$ endowed with the maximum playing the role of addition $\oplus$ and the conventional addition playing the role of multiplication $\otimes$ is an idempotent semiring, usually called (max,+) algebra, and denoted $\mathbb{Z}_{\max}$. We denote $e = 0$ and $\varepsilon = -\infty$ the neutral elements of $\oplus$ and $\otimes$.

[2] At most one token can be in a place at any time.

[3] The graphical representation of a (max,+) automaton is such that: nodes correspond to states, an arrow from a state to another with label a/n denotes a state transition requiring n units of time when event a occurs, an input arrow symbolizes an initial state.

[4] If A is a finite set (*alphabet*), the free monoid on A is defined as the set A^* of finite words with letters in A. A *word* $w \in A^*$ can be written as a sequence $w = a_1 a_2 \ldots a_n$ with a_1, a_2, ..., $a_n \in A$ and n a natural number (the zero-length word is denoted e in the following). *Formal languages* are subsets of the free monoid A^*.

[5] Several initial states and/or several transitions share the same event from certain states.

many results and, for example, it is a required property for (max,+) automata

- to do efficient computations in speech processing [15];
- to compute the optimal, as well as the average behavior, for TDES [5];
- to realize supervisors for TDES [11,2].

Unlike (Boolean) finite automata, it has been shown that nondeterministic (max,+) automata cannot always be determinized, that is transformed into equivalent deterministic (max,+) automata (see for example [13]).

In the continuity of [7], authors proposed in [6] a determinization procedure based on the *completion* of heap models [6]. Returning to the TPN from which the heap model is built, this means that a deterministic (max,+) automaton representation can be obtained if any two transitions share at least one input or output place (see Th.4 and the discussion in [6]). This condition is not satisfied for transitions a and c of the TPN in Fig. 3, and the procedure then fails for this example.

Another approach is to use a determinization procedure applying directly to (max,+) automata. This extension of the classical determinization algorithm of Boolean automata has been extensively studied (see e.g. [5,15,10,13,9]). In few words, a *normalization* operator is defined to quantify the gaps between the dates at which various states are reached at the end of a transition sequence. If a finite number of vectors of gaps is obtained for all the possible sequences, then it is possible to build a normalized automaton which is deterministic and which has the same behavior as the initial nondeterministic automaton. In the case of (max,+) automata derived to represent the TPN in Fig. 3, the determinization procedure based on normalization fails. For (max,+) automata obtained to represent TPNs according to the approach recalled in Section 3, such an unfavourable circumstance arises, in particular, as soon as two transitions have neither an input place nor an output place in common.

In this paper, we propose another way to build deterministic (max,+) automata describing safe TPNs. This new procedure jointly uses the reachability graph of the TPN to cover its language, and the (nondeterministic) (max,+) automaton derived from the TPN to compute the weights to be associated with transitions so that the timed behavior is properly described. As already notified this procedure cannot always succeed (remember that all (max,+) automata cannot be determinized) but, interestingly, it terminates successfully for a wider class than the existing approaches mentioned above. Then a condition on the structure of the TPN is shown to be sufficient for termination of the procedure. More precisely, the main result can be expressed as: if there exists an oriented path between any two transitions which contains at most one "conflict place" (with more than one output transition), then the procedure builds a deterministic (max,+) automaton equivalent to the safe TPN. For example, this condition is satisfied by the TPN in Fig. 3, and the procedure builds the deterministic (max,+) automaton in Fig. 5.

[6] In [14], the case of heap models with two pieces is fully treated.

The paper is organized as follows. In the following section basic concepts and notations are introduced. In section 3, the approach proposed in [12] to transform any safe TPN into a nondeterministic (max,+) automaton is briefly presented. This result is used in the procedure proposed in section 4. This procedure, if it terminates, constructs a deterministic (max,+) automaton equivalent to the safe TPN. Then, a structural condition on TPNs is shown to be sufficient for the termination of the procedure. In section 5, concluding remarks with hints on future extensions are proposed.

2. PRELIMINARIES

Several necessary concepts, results and notations about (max,+) algebra, (max,+) automata and Petri nets are introduced in this section (see [1], [5] and [16] for exhaustive presentations).

2.1 (Max,+) algebra and automata

The set of $n \times n$ matrices with coefficients in (max,+) algebra $\mathbb{Z}_{\max}$, endowed with the matrix addition and multiplication conventionally defined from $\oplus$ and $\otimes$, is also an idempotent semiring, denoted $\mathbb{Z}_{\max}^{n\times n}$. The zero element for the addition is the matrix denoted ε_n and exclusively composed of ε ($= -\infty$). We denote I_n the identity element of the multiplication, which is the matrix with e ($= 0$) on the diagonal and ε elsewhere. As usual in conventional algebra, the multiplication symbol $\otimes$ will be often omitted in the following.

(Max,+) automata can be defined as follows.

Definition 2.1. A *(max,+) automaton* G is a quadruple (Q, A, α, μ) where:

- Q and A are finite sets of states and of events;
- $\alpha \in \mathbb{Z}_{\max}^{1\times|Q|}$ is such that $\alpha_q = e$ or $\alpha_q = \varepsilon$, and q is an initial state if $\alpha_q = e$;
- $\mu : A^* \to \mathbb{Z}_{\max}^{|Q|\times|Q|}$ is a morphism specified by the family of matrices $\mu(a) \in \mathbb{Z}_{\max}^{|Q|\times|Q|}$, $a \in A$, and for a string $a_1 a_2 \dots a_n \in A^*$, we have
$$\mu(a_1 a_2 \dots a_n) = \mu(a_1) \otimes \mu(a_2) \otimes \dots \otimes \mu(a_n).$$
A coefficient $[\mu(a)]_{qq'} \neq \varepsilon$ means that, the occurrence of event a causes a state transition from state q to state q'.

Equivalently G can be defined by the quadruple (Q, A, Q_i, t), in which Q_i denotes the set of *initial* states
$$Q_i = \{q \in Q : \alpha_q = e\},$$
and $t : Q \times A \times Q \to \mathbb{Z}_{\max}$ is the *transition function*
$$t(q, a, q') \triangleq [\mu(a)]_{qq'}.$$

With this def., all states can be thought of as final states (as for automata derived from heap models [7]).

A coefficient $[\mu(a)]_{qq'} \neq \varepsilon$ (equiv., $t(q, a, q') \neq \varepsilon$) means that, if the current state is q, then the occurrence of event a causes a state transition to state q', and value $[\mu(a)]_{qq'}$ is interpreted as the duration associated to event a (namely, the activation time of event a before it can occur). If $[\mu(a)]_{qq'} \neq \varepsilon$, then we denote (q, a, q') the *transition* in G.