Noname manuscript No.
(will be inserted by the editor)

Delay-dependent partial order reduction technique for real time systems

Hanifa Boucheneb · Kamel Barkaoui

Received: date / Accepted: date

Abstract Partial order reduction techniques aim at coping with the state explosion problem by reducing, while preserving the properties of interest, the number of transitions to be fired from each state of the model. For (time) Petri nets, the selection of these transitions is, generally, based on the structure of the (underlying) Petri net and its current marking. This paper proposes a partial order reduction technique for time Petri nets (TPN in short), where the selection procedure takes into account the structure, including the firing intervals, and the current state (i.e., the current marking and the firing delays of the enabled transitions). We show that our technique preserves non-equivalent firing sequences of the TPN. Therefore, its extension to deal with LTL_{-X} properties is straightforward, using the well established methods based on the stuttering equivalent sequences.

Keywords Time Petri nets · Partial order techniques · State space abstractions · Contracted state class graph method,

1 Introduction

A time Petri net (TPN in short) is a Petri net, where each transition is labelled with an interval specifying, relatively to its enabling date, its minimal and maximal firing

H. Boucheneb
Laboratoire VeriForm, Department of Computer Engineering and Software Engineering,
École Polytechnique de Montréal,
P.O. Box 6079, Station Centre-ville, Montréal, Québec, Canada, H3C 3A7.
Tel.: +1-3404711 ext. 4101
E-mail: hanifa.boucheneb@polymtl.ca

K. Barkaoui
Laboratoire CEDRIC, Conservatoire National des Arts et Métiers,
192 rue Saint Martin, Paris Cedex 03, France
Tel.: +331-40272852 E-mail: kamel.barkaoui@cnam.fr

delays. Time Petri nets are definitely established as a powerful formalism for formal verification of real time systems. The verification techniques, such as reachability analysis, are based on the so-called state space abstraction, where states reachable by the same firing sequence, but at different dates, are grouped in the same set and considered modulo some relation of equivalence (abstract states, state classes or state zones) [5,9,10,12,26]. For bounded time Petri nets, state space abstractions, such as the State Class Graph (SCG) [5] and the Zone Based Graph (ZBG) [9], yield finite representations that preserve marking and firing sequences. However, for highly concurrent systems, these verification techniques face a severe problem of state space explosion.

To alleviate this problem, partial order techniques are proposed in the literature for time Petri nets such as: partial order unfolding [14,15,24] and partial order reduction [6–8,16,22,21,25,27]. The idea of the unfolding techniques is to translate a TPN model into an acyclic Petri net with firing time constraints, respecting the partial order of the originate model. The available unfolding techniques are however limited to 1-safe TPNs[1]. The common characteristics of the partial order reduction methods is that they explore a subset of firing sequences (representative firing sequences) from each (abstract) state. These subsets are sufficient to verify the properties of interest.

Among the TPN state space abstractions in the literature, we consider the Contracted State Class Graph (CSCG in short) [12] and investigate partial order reduction techniques, which preserve non-equivalent firing sequences of the TPN (i.e., there is no maximal firing sequence[2] in the TPN with no equivalent sequence[3] in the reduced space and vice-versa). Since the CSCG preserves markings and firing sequences of the TPN, the purpose is to select a subset of firable transitions to be explored from each state class, so as to cover all and only all non-equivalent firing sequences of the CSCG.

In almost all partial order reduction techniques, the selection procedure of representative transitions is based on an independence relation over transitions. Intuitively, two transitions are independent, if they can neither disable nor enable each other and their firings in both orders lead to the same state. If a transition is selected to be fired from a state, then all its dependent and firable transitions are selected too. Various sufficient conditions, guaranteeing an effective selection of an over-approximation of dependent transitions, are proposed in the literature such as persistent sets [16], ample sets [21,22] and stubborn sets [25].

However, in the context of the TPN state space abstractions such as the CSCG, different interleavings of the same set of transitions lead, in general, to different abstract states and then the relation of independency is difficult to meet. For instance,

[1] A 1-safe time Petri net is a 1-bounded time Petri net (i.e., each place can contain at most one token).

[2] A maximal firing sequence is either infinite or finite ending up in a deadlock state (i.e., a state with no enabled transitions).

[3] Two sequences ω and ω' are equivalent (denoted by $\omega \equiv \omega'$) iff ω' can be obtained from ω by successive permutations of its transitions. By convention, it holds that $\omega \equiv \omega'$.

for the TPN at Fig.1.a, taken from [6], the firing of the non-conflicting transitions t_1 and t_2 in both orders leads to two state classes with different behaviours (see Fig.2.a and Table 1). The transition t_3 is not firable from the state class reached by t_2t_1 but is firable from the successor of the initial state class by t_1t_2. To overcome this limitation, two main techniques are used in the literature: the local time semantics [3,17,20] and Partially Ordered Sets (POSETs) of transitions or events [1,18,19,27].

The local time semantics approaches suppose that the model consists of a set of components, each one is represented by a timed model (timed automaton, TPN, etc.) and has, in addition to its clocks, a reference clock. The reference clocks evolve asynchronously and are synchronized when needed (i.e., when an action of synchronization is executed). Such approaches need additional clocks and the differences between reference clocks may diverge leading to an infinite state space [19].

The partial order reduction approaches based on POSETs aim to force the independency relation by fixing partially the firing order of transitions or events [1,18,19, 27]. The idea is to compute, by exploring one sequence of transitions, the convex hull of the abstract states reachable by some of its equivalent sequences. However, unlike timed automata [23], for TPNs, including 1-safe TPNs, this convex hull is not necessarily the union of the abstract states reached by equivalent sequences of transitions [6]. As an example, for the TPN at Fig.1.b taken from [6], the union of state classes reached by different interleavings of transitions t_1 and t_2 from the initial state class is not equal to their convex hull [6]. From its initial state class α_0, firing sequences t_1t_2 and t_2t_1 lead respectively to state classes α_3 and α_5.Their convex hull is the state class $\alpha_{35} = (p_3 + p_4 + 2p_5, -4 \leq t_3 - t_4 \leq 3 \wedge -2 \leq t_3 - t_5 \leq 5 \wedge 0 \leq t_4 - t_5 \leq 4)$ (see Fig.2.a and Table 1). The firing schedule $(t_3 = 2, t_4 = 2, t_5 = 2)$ of α_{35} belongs neither to α_3 nor to α_5. The union of α_3 and α_5 is then not equal to their convex hull. Moreover, if we replace state classes α_3 and α_5 by their convex hull α_{35}, we preserve neither boundedness nor reachability properties of the model. Fig.2.b shows a firing sequence that is neither feasible from α_3 nor from α_5 but feasible from α_{35}. Indeed, the infinite sequence $t_4t_3t_6t_7t_7....$ is neither firable from α_3 nor from α_5. It is however firable from α_{35} and produces an infinite number of markings. This issue is caused by the fact that the parent of t_5 depends of the firing order of transitions t_1 and t_2. The transition t_5 is enabled by t_2, in case t_1 is fired before t_2. It is enabled by t_1, otherwise. Since, the firing delay of a transition is relative to the firing date of its parent, the firing intervals of t_5 in α_3 and α_5 have different references.

In [27], to deal with this issue, the authors keep, in each abstract state, in addition to the time constraints of the enabled transitions, those of their parents. All the different possible parents of the enabled transitions are considered when computing successors of the abstract states. Moreover, the selection procedure of independent transitions takes into account neither the static nor the dynamic timing information of the model. In [18], the authors have defined a state space abstraction where the firing order constraints between non-related transitions[4] are totally ignored when comput-

[4] Transitions are non-related if no one is enabled by the others (i.e., no one is the parent of the others).

ing successors. The subset of transitions explored from each abstract state is a persistent set [18]. However, the state space abstraction proposed in [18] preserve neither markings nor the firing sequences of the TPN. The counterexample is given by the TPN at Fig.3 [7].

In [7], the authors have revisited, using POSETS, the stubborn method in the context of time Petri nets. This method yields reduced graphs that preserve the non-equivalent firing sequences of time Petri nets. However, its selection procedure of representative transitions is only based on the structure of the untimed underlying Petri nets and markings. In [8], the authors have investigated and proposed a selection procedure that takes into account the static and dynamic firing intervals of transitions. These time constraints allow to relax the selection conditions of representative transitions. For instance, the persistency of an enabled transition t is guarantied, if there is no conflicting transition that can fire before t (i.e., t is eventually fired before all conflicting transitions). So, firing delays between transitions allow to weaken the sufficient condition of persistent transitions. The purpose of the present paper is to improve the approach developed in [8], so as to achieve further reduction. The idea is to weaken the selection conditions by taking into better account the static and dynamic firing time constraints of time Petri net. We show that the resulting reduced graph preserves non-equivalent sequences of the TPN. So, the extension of the verification approach proposed here to LTL_{-X}[5] properties over markings could be achieved as shown in [25].

The rest of the paper is organized as follows. Section 2 is devoted to the TPN, its semantics and its CSCG. Section 3 defines the notions of partial order successor and reduced state class graph. Section 4 is devoted to our reduced state class graph and the proof that it preserves the non-equivalent firing sequences of the TPN. Section 5 reports some experimental results. Finally, the conclusions are presented in Section 6.

α_0 $p_1+p_2+p_8$ $-1 \le t_1-t_2 \le 5$	α_1 $p_2+p_3+p_5+p_8$ $-2 \le t_2-t_3 \le 5$	α_2 $p_1+p_4+p_5+p_8$ $2 \le t_1-t_4 \le 4$	α_3 $p_3+p_4+2p_5+p_8$ $-4 \le t_3-t_4 \le 2$ $-2 \le t_3-t_5 \le 5$ $1 \le t_4-t_5 \le 4$
α_4 $p_2+p_5+p_6+p_8$ $true$	α_5 $p_3+p_4+2p_5+p_8$ $-1 \le t_3-t_4 \le 3$ $1 \le t_3-t_5 \le 5$ $0 \le t_4-t_5 \le 4$	α_6 p_3+p_4 $-4 \le t_3-t_4 \le 2$	α_7 $p_4+2p_5+p_6+p_8$ $1 \le t_4-t_5 \le 4$
α_8 p_3+p_4 $-1 \le t_3-t_4 \le 3$	α_9 $p_3+2p_5+p_7+p_8$ $1 \le t_3-t_5 \le 3$	α_{10} p_4+p_6 $true$	α_{11} p_3+p_7 $true$
α_{12} p_6+p_7 $true$			

Table 1 State classes of the CSCG at Fig.2.a

[5] LTL_{-X} properties are LTL properties where the next operator X is forbidden.

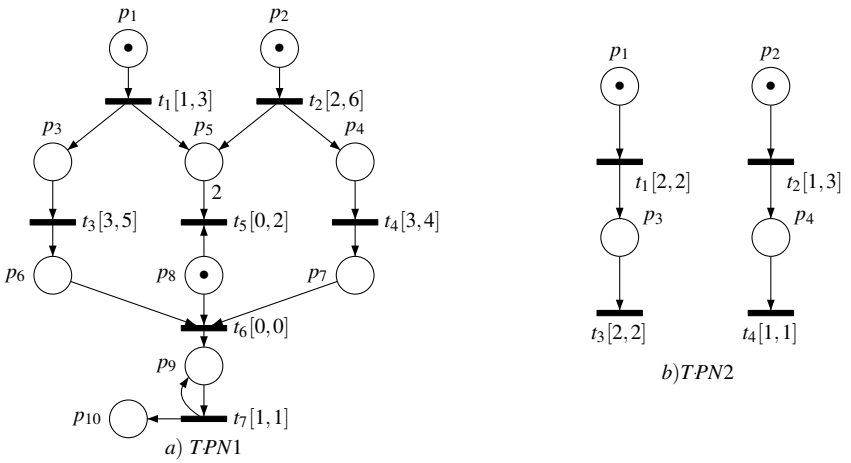

Fig. 1 Time Petri nets used to illustrate features of the interleaving

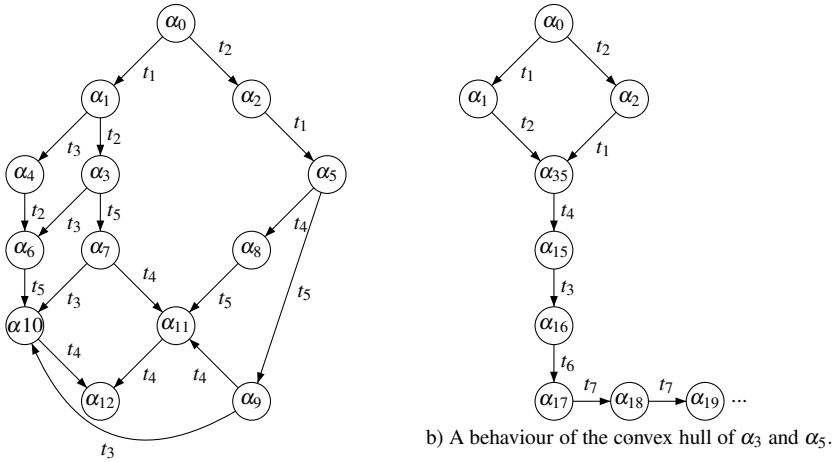

Fig. 2 Convex hull issue of state classes reached by different interleavings of the same set of transitions

2 Time Petri Nets

2.1 Definition and semantics

Let P be a nonempty set. A multi-set over P is a function $M : P \longrightarrow \mathbb{N}$, $\mathbb{N}$ being the set of natural numbers, defined also by the formal sum: $\sum_{p \in P} M(p) \times p$ [6].

We denote by P_{MS} and 0 the set of all multi-sets over P and the empty multi-set, respectively. Let $M_1 \in P_{MS}$, $M_2 \in P_{MS}$ and $\prec \in \{\leq, =, <, >, \geq\}$. Operations on multi-sets are defined as usual:

[6] The symbol $\times$ is an optional separator between elements of M and their occurrence numbers.

1) $\forall p \in P,\ p \in M_1$ iff $M_1(p) > 0$;
2) $M_1 + M_2 = \sum_{p \in P} (M_1(p) + M_2(p)) \times p$;
3) $M_1 \prec M_2$ iff $\forall p \in P, M_1(p) \prec M_2(p)$;
4) $M_1 \not\prec M_2$ iff not $(M_1 \prec M_2)$;
5) $M_1 \times M_2 = \sum_{p \in P} Min(M_1(p), M_2(p)) \times p$;
6) If the multi-sets M_1 and M_2 are s.t. $M_1 \leq M_2$, then $M_2 - M_1$ is the multi-set defined by: $\sum_{p \in P} (M_2(p) - M_1(p)) \times p$.

Let $\mathbb{Q}^+$ and $\mathbb{R}^+$ be the sets of non-negative rational and real numbers, respectively, and $INT_{\mathbb{X}} = \{[a,b] | (a,b) \in \mathbb{X} \times (\mathbb{X} \cup \{\infty\})\}$, for $\mathbb{X} \in \{\mathbb{Q}^+, \mathbb{R}^+\}$, the set of intervals whose lower and upper bounds are in $\mathbb{X}$ and $\mathbb{X} \cup \{\infty\}$, respectively.

Definition 1 A time Petri net is a tuple $\mathcal{N} = (P, T, pre, post, M_0, Is)$ where

- P and T are finite and nonempty sets of places and transitions s.t. $P \cap T = \emptyset$;
- pre and $post$ are the backward and forward incidence functions $(pre, post : T \longrightarrow P_{MS})$;
- $M_0 \in P_{MS}$ is the initial marking; Is is the static firing function $(Is : T \to INT_{\mathcal{Q}^+})$. $\downarrow Is(t)$ and $\uparrow Is(t)$ denote the lower and upper bounds of the static firing interval of transition t.

For $t \in T$, ${}^\circ t = \{p \in P | pre(t)(p) > 0\}$ and $t^\circ = \{p \in P | post(t)(p) > 0\}$ denote the sets of input and output places of t, respectively. Similarly, for $p \in P$, the sets of input and output transitions of p are denoted by ${}^\circ p = \{t \in T | post(t)(p) > 0\}$ and $p^\circ = \{t \in T | pre(t)(p) > 0\}$, respectively.
The transition t is structurally in conflict with a transition t' of T iff they share at least an input place, i.e., ${}^\circ t \cap {}^\circ t' \neq \emptyset$.
We denote by $CFS(t) = \bigcup_{p \in {}^\circ t} p^\circ$ the set of transitions structurally in conflict with t. Note that $t \in CFS(t)$.
We denote by $NwS(t) = \bigcup_{p \in t^\circ} p^\circ$ the set of output transitions of t (the transitions that may be enabled by firing t).

Several semantics are proposed in the literature for the TPN model [4, 11, 13]. An overview and a classification of the TPN semantics can be found in [11]. They differ mainly in the interpretation of the notion of newly enabled transition, the characterization of states and the server policy. The notion of newly enabled transitions may refer to the intermediate markings (markings resulting from the consumption of tokens) or the markings before and after firings (intermediate or atomic firing semantics) [4]. The timing information is either associated with transitions represented by clocks or delays (threshold semantics) or tokens represented by clocks giving their ages (age semantics) [13]. The service policy specifies whether several enabling instances of the same transition may be handled simultaneously (multiple-server semantics) or not (single-server semantics). For the single-server semantics, the multi-enabledness

is not ambiguous, since only one enabling instance of each transition is considered at each state (i.e., sequential management). However, different interpretations can be defined for multiple-server semantics [11]. We consider here the classical and widely used semantics (i.e., the threshold, intermediate and single-server semantics).

Each marking of $\mathscr{N}$ is a multi-set over P. Let M be a marking of $\mathscr{N}$ and $t \in T$ a transition. The transition t is enabled at marking M, denoted by $M[t\rhd$ iff all required tokens for firing t are present in M, i.e., $M \geq pre(t)$. In case t is enabled at M, its firing leads to the marking $M' = M - pre(t) + post(t)$. The notation $M[t\rhd M'$ means that t is enabled at M and M' is the marking reached from M by t. We denote by $En(M)$ the set of transitions enabled at M, i.e., $En(M) = \{t \in T \mid M \geq pre(t)\}$.

For $t \in En(M)$, we denote by $CF(M,t)$ the set of transitions enabled at M but in conflict with t, i.e., $CF(M,t) = \{t' \in En(M) \mid t' = t \vee M \not\geq pre(t) + pre(t')\}$. Note that $CF(M,t) \subseteq CFS(t)$.

For any sequence $t_1t_2...t_n \in T^+$, the usual notation $M[t_1t_2...t_n\rangle$ means that there are markings $M_1,...,M_n$ so that $M_1 = M$ and $M_i[t_i\rhd M_{i+1}$, for $i \in [1,n-1]$ and $M_n[t_n\rangle$. The notation $M[t_1t_2...t_n\rhd M'$ gives, in addition, the marking reached by the sequence.

Let M' be the successor marking of M by t. We denote by $Nw(M,t)$ the set of transitions newly enabled at the marking M' reached from M by firing t. Formally, $Nw(M,t)$ contains t, if t is enabled at M', and also all transitions enabled at the marking M' but not enabled at the intermediate marking $M - pre(t)$, i.e.,
$Nw(M,t) = \{t' \in En(M') \mid t' = t \ \vee \ M - pre(t) \not\geq pre(t')\}$. Note that $Nw(M,t) \subseteq NwS(t)$.

Starting from the initial marking M_0, the marking of $\mathscr{N}$ evolves by firing transitions at irregular intervals of time. When a transition t is newly enabled, its firing interval is set to its static firing interval. Bounds of its interval decrease synchronously with time until it is fired or disabled by a conflicting firing. Transition t is firable, if the lower bound of its firing interval reaches 0. It must fire immediately, without any additional delay, when the upper bound of its firing interval reaches 0, unless it is disabled by another firing. The firing of a transition takes no time but leads to a new marking.

Syntactically, in the context of $\mathscr{N}$, a state is defined as a pair $s = (M,I)$, where M is a marking and I is a firing interval function (I: $En(M) \rightarrow INT_{R^+}$). The initial state of $\mathscr{N}$ is $s_0 = (M_0,I_0)$, where $I_0(t) = Is(t)$, for all $t \in En(M_0)$.

Let $\mathscr{S} = \{(M,I) \mid M \in P_{MS} \wedge I\text{: } En(M) \rightarrow INT_{R^+}\}$ be the set of all syntactically correct states, $s = (M,I)$ and $s' = (M',I')$ two states of $\mathscr{S}$, $dh \in \mathbb{R}^+$ a nonnegative real number, $t \in T$ a transition and $\rightarrow$ the transition relation defined by:

- $s \stackrel{dh}{\rightarrow} s'$ (s' is also denoted by $s+dh$) iff the state s' is reachable from state s by dh time units, i.e., $\forall t \in En(M), dh \ \leq \uparrow I(t), M' = M$ and

$\forall t' \in En(M'), I'(t') = [Max(0, \downarrow I(t') - dh), \uparrow I(t') - dh]$.

- $s \xrightarrow{t} s'$ iff t is immediately firable from s and its firing leads to s', i.e.,
$t \in En(M),\ \downarrow I(t) = 0,\ M' = M - pre(t) + post(t)$, and
$$\forall t' \in En(M'), I'(t') = \begin{cases} Is(t') & \text{if } t' \in Nw(M,t) \\ I(t') & \text{otherwise.} \end{cases}$$

The semantics of $\mathscr{N}$ is defined by the transition system $(S, \rightarrow, s_0)$, where $S \subseteq \mathscr{S}$ is the set of all states reachable from the initial state s_0 by $\xrightarrow{*}$ (the reflexive and transitive closure of $\rightarrow$).

A *run* in $(S, \rightarrow, s_0)$, starting from a state s_1 of S, is a maximal sequence $\rho = s_1 \xrightarrow{dh_1} s_1 + dh_1 \xrightarrow{t_1} s_2 \xrightarrow{dh_2} s_2 + dh_2 \xrightarrow{t_2} s_3 ....$ By convention, for any state s_i, relation $s_i \xrightarrow{0} s_i$ holds. Sequences $dh_1 t_1 dh_2 t_2 ...$ and $t_1 t_2 ...$ are called the timed trace and firing sequence (untimed trace) of ρ, respectively. The total elapsed time during the run ρ, denoted by $time(\rho)$, is $\sum_{i=1,|\rho|} dh_i$, where $|\rho|$ is the length of the firing sequence of ρ.
An infinite run ρ is diverging if $time(\rho) = \infty$, otherwise it is said to be zeno. Runs of $\mathscr{N}$ are all runs of the initial state s_0. A TPN model is said to be non-zeno if all its runs are non-zeno. We consider here only non-zeno TPNs. This restriction ensures that each enabled transition will eventually become firable in the future, unless it is disabled by a conflicting transition. The *timed language* of $\mathscr{N}$ is the set of its timed traces. A marking M is reachable in $\mathscr{N}$ iff $\exists s \in S$ s.t. the marking of s is M.

2.2 Contracted state class graph

Let $\mathscr{N} = (P, T, pre, post, M_0, Is)$ be a TPN. Several state space abstractions have been proposed in the literature for $\mathscr{N}$: the *State Class Graph (SCG)* [5], the *Contracted State Class Graph (CSCG)* [12], the *Geometric Region Graph (GRG)* [26], the *Strong State Class Graph (SSCG)* [5], the *Zone Based Graph (ZBG)* [9] and the *Atomic State Class Graphs (ASCGs)* [5, 10, 26]. In such abstractions, all states grouped in the same node share the same marking and the union of their time domains is represented by a consistent conjunction of atomic constraints [7].

From a practical point of view, every conjunction of atomic constraints is represented by means of a *Difference Bound Matrix* (DBM) [2]. Although the same nonempty domain may be encoded by different conjunction of atomic constraints, their DBMs have a canonical form. The canonical form of a DBM is the representation with tightest bounds on all differences between variables, computed by propagating the effect of each entry through the DBM. Two conjunctions of atomic constraints are equivalent (i.e., represent the same domain) iff their DBMs have the same canonical form. Canonical forms make operations over formulas much simpler [2].

[7] An atomic constraint is a constraint of the form $x \prec c, -x \prec c$ or $x - y \prec c$, where x, y are real-valued variables, $\prec \in \{<, =, \leq, \geq, >\}$ and $c \in \mathbb{Q} \cup \{\infty, -\infty\}$ is a rational number

Among these abstractions, we consider the CSCG. The CSCG is the quotient graph of the SCG [5] w.r.t. some relation of equivalence over state classes of the SCG [12]. Intuitively, this relation groups together all state classes, which have the same marking and triangular constraints[8], but not necessarily the same simple atomic constraints[9]. The CSCG and SCG have the same reachable markings and firing sequences [12]. In other words, the CSCG preserves markings and firing sequences of the SCG, which, in turn, preserves markings and firing sequences of $\mathscr{N}$ [5]. The CSCG of $\mathscr{N}$ is finite iff $\mathscr{N}$ is bounded (i.e. has a finite number of reachable markings).

Syntactically, a CSCG state class is defined as a pair $\alpha = (M,F)$, where M is a marking and F is a consistent conjunction of triangular atomic constraints over firing delays of transitions enabled at M. The formula F characterizes the union of firing time domains of all states within α. By convention, $F = true$ if the number of enabled transitions at M is less than 2 (i.e., there is no triangular atomic constraint in F). A state $s' = (M',I')$ belongs to α iff $M = M'$ and its firing time domain (i.e., $\bigwedge_{t \in En(M')} \downarrow I'(t) \leq t \leq \uparrow I'(t)$) is included in the firing time domain of α (i.e., F).

The CSCG initial state class is $\alpha_0 = (M_0, F_0)$, where

$$F_0 = \bigwedge_{t,t' \in En(M_0) \text{ s.t. } t \neq t'} t - t' \leq \uparrow Is(t) - \downarrow Is(t'),$$

t and t' being real-valued variables representing firing delays of transitions t and t', respectively. It keeps only the triangular atomic constraints of the SCG initial state class.

Let $\mathscr{C}_S$ be the set of all syntactically correct CSCG state classes and *succ* a successor function from $\mathscr{C}_S \times T$ to $\mathscr{C}_S \cup \{\emptyset\}$ defined by: $\forall \alpha \in \mathscr{C}_S, \forall t_f \in T$,

- $succ(\alpha,t_f) \neq \emptyset$ (i.e., t_f is firable from α) iff $t_f \in En(M)$ and the following formula is consistent (its domain is not empty): $F \wedge (\bigwedge_{t \in En(M)} t_f - t \leq 0)$.
  Intuitively, this formula, called the firing condition of t_f from α, means that t_f is firable from α before all other transitions enabled at M. In other words, there is at least a valuation of firing delays in F s.t. t_f has the smallest firing delay.
- If $succ(\alpha,t_f) \neq \emptyset$ then $succ(\alpha,t_f) = (M',F')$, where:
  $M' = M - pre(t_f) + post(t_f)$ and F' is computed in three steps:
  1) Set F' to $F \wedge \bigwedge_{t \in En(M)} t_f - t \leq 0 \wedge \bigwedge_{t' \in Nw(M,t_f)} \downarrow Is(t') \leq t'^f - t_f \leq \uparrow Is(t')$
  (Variables t'^f for $t' \in Nw(M,t_f)$ are new variables introduced for representing the firing delays of the newly enabled transitions. The notation t'^f allows to deal with the situation where t' is enabled before firing t_f and newly enabled by t_f (i.e. $t' \in CF(M,t_f) \cap Nw(M,t_f)$). The new instance of t' is temporally represented by t'^f, in this step);

[8] A triangular atomic constraint is an atomic constraint of the form $x - y \prec c$.

[9] A simple atomic constraint is an atomic constraint of the form $x \prec c$ or $-x \prec c$.

2) Put F' in canonical form[10] and eliminate all transitions of $CF(M,t_f)$;
3) Rename each t'^f into t'.

Let $\alpha = (M,F) \in \mathscr{C}_S$. We denote by $Fr(\alpha) = \{t \in T \mid succ(\alpha,t) \neq \emptyset\}$ the set of transitions firable from α. The function *succ* is extended to sequences of transitions as follows: $\forall \omega \in T^*$, $succ(\alpha,\omega) = succ(succ(\alpha,\omega_1),\omega_2)$, where $\omega = \omega_1\omega_2$ and, by convention, $succ(\alpha,\varepsilon) = \alpha$, ε being the empty sequence. We denote by $||\omega|| \subseteq T$ the set of transitions appearing in ω.

The CSCG of $\mathscr{N}$ is the structure $\mathbb{C} = (\mathscr{C}, succ, \alpha_0)$, where α_0 is the initial CSCG state class of $\mathscr{N}$ and $\mathscr{C}$ is the set of state classes accessible from α_0 by applying repeatedly the successor function *succ*, i.e., $\mathscr{C} = \{\alpha \in \mathscr{C}_S | \exists \omega \in T^*, \alpha = succ(\alpha_0,\omega) \neq \emptyset\}$. A sequence $\omega \in T^+$ is a firing sequence of $\mathbb{C}$ iff $succ(\alpha_0,\omega) \neq \emptyset$.

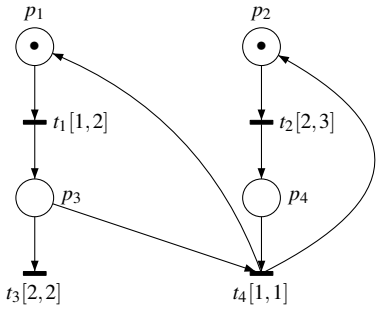

Fig. 3 TPN3

Example 1 Consider the model $TPN2$ at Fig.1.b. Its CSCG initial state class is: $\alpha_0 = (p_1 + p_2, -1 \leq t_1 - t_2 \leq 1)$. There are two enabled transitions t_1 and t_2, which are also firable from α_0, since their firing conditions $-1 \leq t_1 - t_2 \leq 1 \wedge t_1 \leq t_2$ and $-1 \leq t_1 - t_2 \leq 1 \wedge t_2 \leq t_1$ are consistent. For instance, let us compute the successor of α_0 by t_1. The firing of t_1 leads to the state class $\alpha_1 = (p_2 + p_3, -2 \leq t_2 - t_3 \leq -1)$. Its marking is computed as usual. Its formula is computed in three steps:
1) Set the formula to the firing condition of t_1 from α_0 augmented with time constraints of transition t_3 newly enabled by t_1: $-1 \leq t_1 - t_2 \leq 1 \wedge t_1 \leq t_2 \wedge t_3^1 - t_1 = 2$;
2) Put the formula in canonical form and eliminate t_1: $-2 \leq t_2 - t_3^1 \leq -1$;
3) Rename t_3^1 in t_3: $-2 \leq t_2 - t_3 \leq -1$.
Following the same procedure, we get $succ(\alpha,t_1t_2) = (p_3 + p_4, 0 \leq t_3 - t_4 \leq 1)$ and $succ(\alpha,t_2t_1) = (p_3 + p_4, 1 \leq t_3 - t_4 \leq 2)$.

[10] The canonical form of F' is the formula corresponding to the canonical form of its DBM.

3 Partial order reduction based on POSETs

3.1 Partial order successors and reduced state class graphs

The idea of partial order successors is to relax the firing condition of a transition by eliminating some firing order constraints when computing successors of state classes. The aim is to handle concisely the equivalent sequences of transitions, obtained by permuting some independent transitions (i.e., partially ordered sets of transitions). As a result, the union of state classes reached by all these sequences is computed by exploring only one of them.

Definition 2 Let $\alpha = (M,F)$ be a state class of $\mathscr{C}_S$, $t_f \in T$ a transition and $X \subseteq T$ a subset of transitions. The partial order successor of α by t_f w.r.t. X, denoted by $succ_X(\alpha,t_f)$, is either equal $\emptyset$ or a state class of $\mathscr{C}_S$ defined by:

$$succ_X(\alpha,t_f) \neq \emptyset \text{ iff } X \cap En(M) \neq \emptyset \wedge succ(\alpha,t_f) \neq \emptyset.$$

If $succ_X(\alpha,t_f) \neq \emptyset$ then the state class $\alpha' = succ_X(\alpha,t_f)$ is computed as $succ(\alpha,t_f)$, except that the firing condition, used in step 1, is replaced with: $F \wedge \bigwedge_{t \in X \cap En(M)} t_f \leq t$.

Formally, if $succ_X(\alpha,t_f) \neq \emptyset$ then $succ_X(\alpha,t_f) = (M',F')$, where
$M' = M - pre(t_f) + post(t_f)$ and F' is computed in three steps:
1) Set F' to $F \wedge \bigwedge_{t \in X \cap En(M)} t_f \leq t \ \wedge \bigwedge_{t' \in Nw(M,t_f)} \downarrow Is(t') \leq t'^f - t_f \leq \uparrow Is(t')$;
2) Put F' in canonical form and eliminate all transitions of $CF(M,t_f)$;
3) Rename each t'^f in t'.
The formula used in step 1, called the processing formula of $succ_X(\alpha,t_f)$, does not impose any firing order between t_f and transitions of $En(M) - X$. Therefore, it holds that $\forall t_f \in T, succ(\alpha,t_f) \subseteq succ_X(\alpha,t_f)$ and $succ_{En(M)}(\alpha,t_f) = succ(\alpha,t_f)$.

Example 2 Consider the model $TPN2$ at Fig.1.b and its initial state class
$\alpha_0 = (p_1 + p_2, -1 \leq t_1 - t_2 \leq 1)$. Transitions t_1 and t_2 are both enabled and firable from α_0. Therefore, $succ_{\{t_1\}}(\alpha_0,t_1) \neq \emptyset$ and $succ_{\{t_2\}}(\alpha_0,t_2) \neq \emptyset$.
Let $\alpha'_1 = succ_{\{t_1\}}(\alpha_0,t_1)$. Let us show how to compute the firing domain formulas of α'_1 and $\alpha'_2 = succ_{\{t_2\}}(\alpha'_1,t_2)$.
For the state class $\alpha'_1 = (p_2 + p_3, F'_1)$, its firing domain formula F'_1 is computed in three steps as follows:
1) Set F'_1 to $-1 \leq t_1 - t_2 \leq 1 \wedge t^n_3 - t_1 = 2$;
2) Put the formula in canonical form and eliminate t_1: $-3 \leq t_2 - t^n_3 \leq -1$;
3) Rename t^n_3 in t_3: $-3 \leq t_2 - t_3 \leq -1$.
For the state class $\alpha'_2 = (p_3 + p_4, 0 \leq t_3 - t_4 \leq 2)$, its firing domain formula F'_2 is computed in three steps as follows:
1) Set F'_2 to $-3 \leq t_2 - t_3 \leq -1 \wedge t^n_4 - t_2 = 1$;
2) Put the formula in canonical form and eliminate t_1: $0 \leq t_3 - t^n_4 \leq 2$;
3) Rename t^n_4 in t_4: $0 \leq t_3 - t_4 \leq 2$.
Note that $succ_{\{t_2\}}(succ_{\{t_1\}}(\alpha_0,t_1),t_2) = succ(\alpha_0,t_1t_2) \cup succ(\alpha_0,t_2t_1)$.

Therefore, $succ(succ_{\{t_2\}}(succ_{\{t_1\}}(\alpha_0,t_1),t_2),t_3)$ gives the union of state classes reached by sequences $t_1t_2t_3$ and $t_2t_1t_3$. The union of these sequences can be represented by the partially ordered set $(\{t_1,t_2,t_3\}, t_1 \leq t_3 \wedge t_2 \leq t_3)$.

We provide, in the following, some relationships between successors and partial order successors of state classes, which will be helpful to establish a partial order reduction technique and prove that it preserves the non-equivalent firing sequences of the TPN. Let us first define the notion of *effect-independent* transitions used in our partial order reduction technique (instead of the notion of truth parent [27]).

Definition 3 Let $t_i,t_j \in T$ be two transitions. Transitions t_i and t_j are structurally effect-independent, denoted by $t_i||t_j$ iff their effects are independent of their firing order from any marking, i.e.,

$$(CFS(t_i) \cup NwS(t_i)) \cap (CFS(t_j) \cup NwS(t_j)) = \emptyset$$

Let $\alpha = (M,F) \in \mathscr{C}_S$ be a state class, $t_i \in Fr(\alpha)$ and $t_j \in Fr(\alpha)$ two transitions firable from α. Let M_i and M_j be the successor markings of M by t_i and t_j, respectively (i.e., $M[t_i> M_i$ and $M[t_j> M_j$). Transitions t_i and t_j are effect-independent from α, denoted by $t_i||_\alpha t_j$ iff their effects from α are independent of their firing order, i.e.,

$$CF(M,t_i) = CF(M_j,t_i) \wedge CF(M,t_j) = CF(M_i,t_j) \wedge$$

$$Nw(M,t_i) = Nw(M_j,t_i) \wedge Nw(M,t_j) = Nw(M_i,t_j).$$

In other words, from M, each of transitions t_i and t_j will disable (enable) the same set of transitions no matter of which transition is fired first.

Note that relations $||_\alpha$ and $||$ are symmetric (i.e., $t_i||_\alpha t_j$ iff $t_j||_\alpha t_i$ and $t_i||t_j$ iff $t_j||t_i$).

Lemma 1 *Let $\alpha = (M,F) \in \mathscr{C}_S$, $t_i \in Fr(\alpha)$, M_i the successor marking of M by t_i, and X a subset of transitions s.t. $CF(M,t_i) \subseteq X$.*
(i) $\forall t_j \in Fr(\alpha)$, $t_i||t_j \Rightarrow t_i||_\alpha t_j$.
(ii) $\forall t_j \in En(M_i) - Fr(\alpha), succ(succ_X(\alpha,t_i),t_j) = succ(\alpha,t_it_j)$.
(iii) $\forall t_j \in X \cap En(M_i), succ(succ_X(\alpha,t_i),t_j) = succ(\alpha,t_it_j)$.
(iv) $\forall t_j \in Fr(\alpha) - X$, s.t. $X \subseteq En(M) \wedge X \cap CF(M,t_j) = \emptyset \wedge t_i||_\alpha t_j$,
$succ(succ_X(\alpha,t_i),t_j) = succ(\alpha,t_it_j) \cup succ_X(succ(\alpha,t_j),t_i)$.

Proof By assumption, the transition $t_i \in Fr(\alpha)$ and $CF(M,t_i) \subseteq X$. Therefore, $X \cap En(M) \neq \emptyset$, $succ(\alpha,t_i) \neq \emptyset$ and $succ_X(\alpha,t_i) \neq \emptyset$.
Proof of (i): Suppose that $t_i||_\alpha t_j$ does not hold and let us show that $t_i||t_j$ does not hold too. By definition, $\neg t_i||_\alpha t_j$ implies that at least one of the following statements holds $CF(M,t_i) \neq CF(M_j,t_i)$, $CF(M,t_j) \neq CF(M_i,t_j)$, $Nw(M,t_i) \neq Nw(M_j,t_i)$ or

$Nw(M,t_j) \neq Nw(M_i,t_j)$. Therefore, one of the transitions t_i and t_j may disable / enable a transition in conflict with the other or an output transition of the other. It means that $(CFS(t_i) \cup NwS(t_i)) \cap (CFS(t_j) \cap NwS(t_j)) \neq \emptyset$. Therefore, $\neg(t_i||t_j)$.

Proof of (ii) and (iii): The processing formula of $succ(succ_X(\alpha,t_i),t_j)$, denoted by ϕ, is:

$$(F \wedge \bigwedge_{t \in X \cap En(M)} t_i \leq t \wedge \bigwedge_{t' \in Nw(M,t_i)} \downarrow Is(t') \leq t^i - t_i \leq \uparrow Is(t')) \wedge$$

$$(\bigwedge_{t \in En(M_i) - Nw(M,t_i)} t_j \leq t \wedge \bigwedge_{t' \in Nw(M,t_i)} t_j \leq t'^i \wedge \bigwedge_{t' \in Nw(M_i,t_j)} \downarrow Is(t') \leq t'^j - t_j \leq \uparrow Is(t')).$$

By assumption, $t_j \in En(M_i) - Fr(\alpha)$ or $t_j \in X \cap En(M_i)$. We consider three cases: $t_j \in Nw(M,t_i)$ (i.e., t_j is newly enabled in M_i), $t_j \in (En(M) - CF(M,t_i)) \cap X$ (i.e., t_j is not newly enabled in M_i and belongs to X) or $t_j \in En(M) - Fr(\alpha)$ (i.e., t_j is neither newly enabled in M_i nor firable in α). In all cases, it holds that $(\phi \wedge t_i \leq t_j) \equiv \phi$. By definition, $En(M_i) = (En(M) - CF(M,t_i)) + Nw(M,t_i)$. Therefore, the following constraints of ϕ: $t_i \leq t_j \wedge t_j \leq t$, for $t \in En(M) - CF(M,t_i)$ imply $t_i \leq t$ for $t \in En(M) - CF(M,t_i)$. Adding these redundant constraints to ϕ does not affect its domain. Since $CF(M,t_i) \subseteq X$, it follows that $En(M) = (En(M) \cap X) \cup (En(M) - CF(M,t_i))$ and then ϕ is equivalent to:

$$(F \wedge \bigwedge_{t \in En(M)} t_i \leq t \wedge \bigwedge_{t' \in Nw(M,t_i)} \downarrow Is(t') \leq t'^i - t_i \leq \uparrow Is(t')) \wedge$$

$$(\bigwedge_{t \in En(M_i) - Nw(M,t_i)} t_j \leq t \wedge \bigwedge_{t' \in Nw(M,t_i)} t_j \leq t'^i \wedge \bigwedge_{t' \in Nw(M_i,t_j)} \downarrow Is(t') \leq t'^j - t_j \leq \uparrow Is(t')).$$

Therefore, $succ(succ_X(\alpha,t_i),t_j) = succ(\alpha,t_i t_j)$.

Proof of (iv): By assumption, $t_i \in Fr(\alpha), t_j \in Fr(\alpha) - X \subseteq En(M)$ and $CF(M,t_i) \subseteq X$. Then, $X = En(M) \cap X$, $succ(\alpha,t_i t_j) \neq \emptyset$ and $succ(\alpha,t_i t_j) \subseteq succ(succ_X(\alpha,t_i),t_j) \neq \emptyset$. Consider now the processing formula above ϕ of $succ(succ_X(\alpha,t_i),t_j)$. It holds that $\phi \equiv ((\phi \wedge t_i \leq t_j) \vee (\phi \wedge t_j \leq t_i))$. Following the same steps as in the proof of (ii) and (iii), we show that $(\phi \wedge t_i \leq t_j)$ is equivalent to the firing condition of $succ(\alpha,t_i t_j)$. For $(\phi \wedge t_j \leq t_i)$, by definition, $En(M_i) = (En(M) - CF(M,t_i)) + Nw(M,t_i)$ and, by assumption, $X \cap CF(M,t_j) = \emptyset$. Therefore, the following constraints of ϕ: $t_j \leq t_i \wedge t_i \leq t$, for $t \in X \cup Nw(M,t_i)$ imply $t_j \leq t$ for $t \in X \cup Nw(M,t_i)$. Adding the redundant constraints $t_j \leq t$ for $t \in X$ to $\phi \wedge t_j \leq t_i$ does not affect its domain. Moreover, the constraint $t_j \leq t$ for $t \in Nw(M,t_i)$ are redundant in $\phi \wedge t_j \leq t_i$. So, they can be eliminated from $\phi \wedge t_j \leq t_i$ without affecting its domain. Since $CF(M,t_i) \subseteq X$, it follows that $En(M) = (En(M) - CF(M,t_i)) \cup X$. Therefore, we can state that $\phi \wedge t_j \leq t_i$ is equivalent to:

$$(F \wedge \bigwedge_{t \in X} t_i \leq t \wedge \bigwedge_{t' \in Nw(M,t_i)} \downarrow Is(t') \leq t'^i - t_i \leq \uparrow Is(t')) \wedge$$

$$(\bigwedge_{t \in En(M)} t_j \leq t \wedge \bigwedge_{t' \in Nw(M_i,t_j)} \downarrow Is(t') \leq t'^j - t_j \leq \uparrow Is(t')).$$

Let M_j be the successor marking of M by t_j. By definition, $En(M_j) = En(M) - CF(M,t_j) + Nw(M,t_j)$. Since by assumption $X \cap CF(M,t_j) = \emptyset$, it follows that $X \subseteq En(M_j)$ and then $X = En(M_j) \cap X$.
By assumption, $t_i \,||_\alpha\, t_j$, which imply $Nw(M,t_i) = Nw(M_j,t_i)$ and $Nw(M,t_j) = Nw(M_i,t_j)$. Then, $\phi \wedge t_j \leq t_i$ is equivalent to the processing formula of $succ_X(succ(\alpha,t_j),t_i)$. Consequently, $succ(succ_X(\alpha,t_i),t_j) = succ(\alpha,t_it_j) \cup succ_X(succ(\alpha,t_j),t_i)$. □

Intuitively, given a selection procedure (over state classes) of the representative transitions, a reduced state class graph based on POSETs is generated by first computing the partial order successors of the initial state class, by its selected transitions, and then repeating the procedure for each computed but not processed state class.

Definition 4 Let $\mathbb{C} = (\mathscr{C}, succ, \alpha_0)$ be the CSCG of a TPN $\mathscr{N}$ and G a function from $\mathscr{C}_S$ to 2^T called a partial order generator. The reduced state class graph (**RSCG for short**) generated by G is the tuple $\mathbb{R} = (G, \mathscr{C}_G, succ_G, \alpha_0)$, where $\mathscr{C}_G = \{\alpha | \alpha_0 \xrightarrow{*}_G \alpha\}$ is the set of reachable state classes in $\mathbb{R}$ and $\xrightarrow{*}_G$ is the reflexive and transitive closure of the transition relation $\longrightarrow_G$ defined by: $\forall \alpha, \alpha' \in \mathscr{C}_S, \forall t_f \in T$,

$$\alpha \xrightarrow{t_f}_G \alpha' \text{ iff } t_f \in G(\alpha) \wedge succ(\alpha,t_f) \neq \emptyset \wedge \alpha' = succ_{G(\alpha)}(\alpha,t_f).$$

Let $\alpha \in \mathscr{C}_G$ and $\omega = t_1t_2...t_n$ be a sequence of transitions. We write $\alpha \xrightarrow{\omega}_G \alpha_n$ iff $\exists \alpha_1, \alpha_2, ..., \alpha_n \in \mathscr{C}_G$ s.t. $\alpha \xrightarrow{t_1}_G \alpha_1 \xrightarrow{t_2}_G \alpha_2 ... \xrightarrow{t_n}_G \alpha_n$, with $\alpha_n = succ_G(\alpha, \omega)$.
The RSCG $\mathbb{R}$ preserves the non-equivalent sequences of the CSCG $\mathbb{C}$ iff for each maximal sequence of $\mathbb{R}$, there is an equivalent sequence in $\mathbb{C}$ and vice-versa.

4 RSCG preserving non-equivalent sequences of $\mathscr{N}$

We propose, in the following, a partial order generator G and show that it results in a RSCG preserving non-equivalent sequences of the CSCG. The proposed generator takes into account the structure of the TPN, including the static firing intervals of transitions, the marking and the firing domain of the current state class. The timing information derived from the structure of the TPN is captured in a matrix called the delay lower bound matrix.

4.1 Static delay lower bound matrix of $\mathscr{N}$

According to the TPN semantics, when a transition t_j is fired, the conflicting transitions are disabled and new transitions may be enabled. The firing delay interval of each newly enabled transition t_i refers to its enabling date (i.e., the firing date of t_j). The lower bound of the firing delay of transition t_i relatively to the firing date of t_j is $\downarrow Is(t_i)$. We define the delay lower bound matrix L as a square matrix over the set of

transitions T, where: $\forall t_i, t_j \in T$,

$$l_{ij} = \begin{cases} 0 & \text{if } t_i = t_j \\ \downarrow Is(t_i) & \text{if } t_i \neq t_j \wedge t_i \in NwS(t_j) \\ \infty & \text{otherwise.} \end{cases}$$

We denote by $\bar{L}$ the canonical form of L obtained by applying the Floyd-Warshall's shortest path algorithm. This algorithm converges, as the lower bounds of the static firing intervals are non-negative finite rational numbers. Intuitively, $\bar{l}_{ij}$ is a lower bound of the firing delay of t_i, relatively to the firing date of t_j, for the case where t_i is not enabled when t_j is fired. Note that $\bar{l}_{ij} = \infty$ means that there is no path connecting t_j to t_i and then t_i cannot be enabled directly or indirectly by t_j.

Table 2 Firing delay lower bound matrix of TPN3 at Fig.3 and its canonical form

L	t_1	t_2	t_3	t_4
t_1	0	∞	∞	1
t_2	∞	0	∞	2
t_3	1	∞	0	∞
t_4	2	2	∞	0

$\bar{L}$	t_1	t_2	t_3	t_4
t_1	0	2	∞	1
t_2	2	0	∞	1
t_3	2	4	0	3
t_4	1	1	∞	0

Example 3 Table 2 reports the matrices L and $\bar{L}$ of the model TPN3 at Fig.3. For instance, the value 2 of $\bar{l}_{21}$ is a lower bound of the firing delay of t_2, relatively to the firing date of t_1, in case t_2 is not enabled when t_1 is fired. It corresponds to the potential situation where t_1 enables t_4, which, in turn, enables t_2 (i.e., $\bar{l}_{21} = l_{24} + l_{41} = \downarrow Is(t_2) + \downarrow Is(t_4)$). Note that a lower bound of the enabling delay of t_2 relatively to the firing date of t_1, in case t_2 is not enabled when t_1 is fired, is $\bar{l}_{21} - \downarrow Is(t_2) = 1$.

4.2 Computing a partial order generator G

Several algorithms have been proposed in the literature to compute partial order generator G for the RSCG preserving different kinds of properties such as deadlocks and LTL_{-X} properties. In general, these algorithms infer G from the static structure of the model, without taking into account the timing information.

This section proposes an algorithm for computing G inspired from the stubborn sets method [25,27], but does not use the notion of truth parent [27]. As in [8], it uses the notion of effect-independent transitions and the (static and dynamic) timing information of the model. Its purpose is to weaken, by considering firing order time constraints, the selection conditions provided in [8], so as to achieve further reductions.

Formally, let $\alpha = (M,F)$ be a state class, D the canonical form of F (i.e., $d_{ij} = Max(t_i - t_j|F)$, for $t_i,t_j \in En(M)$). The set $G(\alpha)$ is the smallest set of transitions of $En(M)$ that satisfies all the following conditions:
C0: $Fr(\alpha) \neq \emptyset \Leftrightarrow\ G(\alpha) \cap Fr(\alpha) \neq \emptyset$.
C1: $\forall t_i \in G(\alpha) \cap Fr(\alpha), \forall t_j \in En(M)$,
$((t_j \in Fr(\alpha) \wedge \neg t_i\ ||\ t_j) \vee (t_j \notin Fr(\alpha) \wedge d_{ij} \geq 0 \wedge t_j \in CFS(t_i)))\ \Rightarrow t_j \in G(\alpha)$.
C2: $\forall t_i \in G(\alpha) \cap Fr(\alpha), \forall t_j \in Fr(\alpha)$,
$\forall t_k \in CFS(t_i) - En(M), \bar{l}_{kj} \leq d_{ij} \Rightarrow t_j \in G(\alpha)$,
C3: $\exists t_i \in G(\alpha) \cap Fr(\alpha), \forall t_j \in G(\alpha) - Fr(\alpha), t_j \notin CFS(t_i) \vee d_{ij} < 0$.
We denote by **SC** the conjunction $C0 \wedge C1 \wedge C2 \wedge C3$.

Intuitively, C0 ensures that $G(\alpha)$ is empty only for deadlock state classes. This condition is necessary to preserve the deadlock property.

Note that, for $t_i,t_j \in En(M), d_{ij} \geq 0$ means that for some delay valuation of F, the firing delay of t_i is larger or equal to the firing delay of t_j. So, t_j can be fired before t_i from α or from a successor (direct / indirect) of α. In case $d_{ij} < 0$, it means that t_j cannot be fired before t_i from α or from a successor (direct / indirect) of α, unless t_i is disabled by a conflicting transition. Condition C1 means that all firable transitions of $G(\alpha)$ are effect-independent of transitions $Fr(\alpha) - G(\alpha)$ and not structurally in conflict with the transitions of $En(M) - Fr(\alpha) - G(\alpha)$. Therefore, the firing of any transition of $G(\alpha)$ will not disable the transitions of $En(M) - G(\alpha)$ and vice-versa.

Condition C2 ensures that during the enabledness of any firable transition t_i of $G(\alpha)$, no transition t_j outside $G(\alpha)$ may enable directly/indirectly a transition t_k that is structurally in conflict with t_i and firable before t_i (see Fig. 4). The precondition $\bar{l}_{kj} \leq d_{ij}$ means that the firing delay of transition t_k relatively to the firing date of t_j can be smaller ou equal to the maximal delay between the firing dates of transitions t_j and t_i. In other words, after firing t_j, t_k can occur before t_i. In case this precondition is not satisfied, it means that, after firing t_j, t_k cannot occur as long as t_i is enabled. Conditions C1 and C2 imply that the enabledness of t_i will not be affected by firing the transitions outside $G(\alpha)$.

Condition C3 prevents to loose sequences with no equivalent sequence starting with a transition of $G(\alpha) \cap Fr(\alpha)$. For instance, suppose that there is a maximal sequence ωt_j of α s.t. all transitions of ω do not belong to $G(\alpha)$, t_j belongs to $G(\alpha) - Fr(\alpha)$ and is in conflict with all transitions of $G(\alpha)$. The firing of t_j after ω disables all transitions of $G(\alpha)$. In case $G(\alpha)$ does not satisfy Condition C3, no sequence equivalent to ωt_j is represented in the reduced graph. Otherwise, if t_j is fired after ω, at least a transition t_i of $G(\alpha) \cap Fr(\alpha)$ is still firable and not in conflict with t_j. Therefore, sequences $\omega t_j t_i$ and $\omega t_i t_j$ are both firable from α. Conditions C1, C2 and C3 ensure that $t_i \omega t_j$ is also firable from α. As we will show, they also ensure that $succ_G$ handles all equivalent sequences resulting from permuting transitions of $G(\alpha)$ with the other firable transitions.

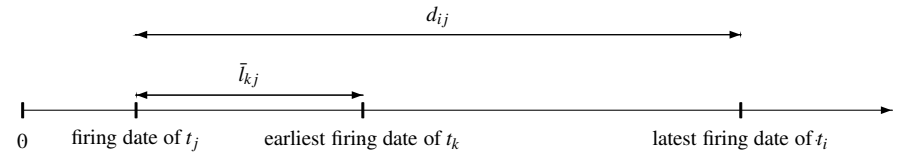

Fig. 4 Condition $C2$ of SC: $\bar{l}_{kj} \leq d_{ij}$ for $t_i \in G(\alpha), t_j \in En(M), t_k \in CFS(t_i)$

4.3 Does G preserve the non-equivalent firing sequences of $\mathscr{N}$?

The proof that G preserves the non-equivalent firing sequences of $\mathscr{N}$ is stated in Theorem 1. It is based on some useful properties established in Lemma 2. The notation $G \models SC$ is an abbreviation of $\forall \alpha \in \mathscr{C}_S, G(\alpha) \models SC$.

Lemma 2
(i) $G \models SC \;\Rightarrow\; \forall \alpha \in \mathscr{C}_S, \forall \omega \in (T - (G(\alpha) \cap Fr(\alpha)))^+$,
$succ(\alpha, \omega) \neq \emptyset \Rightarrow \exists t_i \in G(\alpha) \cap Fr(\alpha), (i)\ succ(\alpha, \omega t_i) \neq \emptyset \wedge\ (ii)\ succ(\alpha, t_i \omega) \neq \emptyset$.
(ii) $G \models C1 \;\Rightarrow\; \forall \alpha \in \mathscr{C}_S, \forall \omega \in T^+, succ_G(\alpha, \omega) \neq \emptyset \Rightarrow \exists \omega' \equiv \omega, succ(\alpha, \omega') \neq \emptyset$.

Proof **(i):** By C3, there is at least a transition t_i of $G(\alpha) \cap Fr(\alpha)$ s.t. $t_j \notin CFS(t_i) \vee d_{ij} < 0$, for each $t_j \in G(\alpha) - Fr(\alpha)$. Let us show that for such a transition t_i, it holds that: $\forall \omega \in (T - (G(\alpha) \cap Fr(\alpha)))^+$,

$$succ(\alpha, \omega) \neq \emptyset \Rightarrow \text{(i)}\ succ(\alpha, \omega t_i) \neq \emptyset \wedge\ (ii)\ succ(\alpha, t_i \omega) \neq \emptyset.$$

Each transition t_j of ω belongs $T - (G(\alpha) \cap Fr(\alpha))$. Let us consider four cases: $t_j \in Fr(\alpha) - G(\alpha)$, $t_j \in G(\alpha) - Fr(\alpha)$, $t_j \in (En(M) - Fr(\alpha)) - G(\alpha)$ and $t_j \in T - En(M)$.
a) if $t_j \in Fr(\alpha) - G(\alpha)$ then, by C1, $t_i || t_j$, which implies that $t_j \notin CFS(t_i)$.
b) If $t_j \in G(\alpha) - Fr(\alpha)$ then $d_{ij} \geq 0$, as t_j can be fired before t_i. By assumption, $t_j \notin CFS(t_i)$.
c) If $t_j \in (En(M) - Fr(\alpha)) - G(\alpha)$ then $d_{ij} \geq 0$. By C1, $t_j \notin CFS(t_i)$.
d) If $t_j \in T - En(M)$ then t_j is enabled directly or indirectly by some transition of $Fr(\alpha) - G(\alpha)$ before firing t_i. By C2, $t_j \notin CFS(t_i)$.
All transitions of ω are not structurally in conflict with t_i. Therefore, $succ(\alpha, \omega) \neq \emptyset$ implies that $succ(\alpha, \omega t_i) \neq \emptyset$ and $succ(\alpha, t_i \omega) \neq \emptyset$.
(ii) (by induction on the length of ω)**:**
a) For $\omega = t_1$, by definition, $succ_{G(\alpha)}(\alpha, t_1) \neq \emptyset$ iff $succ(\alpha, t_1) \neq \emptyset$.
b) For $\omega = t_1 t_2$, $succ_G(\alpha, t_1 t_2) \neq \emptyset$ iff $succ(succ_{G(\alpha)}(\alpha, t_1), t_2) \neq \emptyset$. If $t_2 \in G(\alpha)$ or $t_2 \notin Fr(\alpha)$, by Lemma 1, $succ(succ_{G(\alpha)}(\alpha, t_1), t_2) = succ(\alpha, t_1 t_2) \neq \emptyset$. Otherwise, from C1 of SC and Lemma 1, it follows that $t_1 || t_2$, sequences $t_1 t_2$ and $t_2 t_1$ are firable from α, and $succ(succ_{G(\alpha)}(\alpha, t_1), t_2) = succ(\alpha, t_1 t_2) \cup succ_{G(\alpha)}(succ(\alpha, t_2), t_1)$.
c) For $\omega = t_1 ... t_n$ of length $n > 2$, $succ_G(\alpha, \omega) \neq \emptyset$ iff $succ(succ_G(\alpha, t_1 ... t_{n-1}), t_n) \neq \emptyset$. Let $\alpha_{n-1} = succ_G(\alpha, t_1 ... t_{n-2})$.
If $t_n \in G(\alpha_{n-1})$ or $t_n \notin Fr(\alpha_{n-1})$, according to Lemma 1, $succ(succ_G(\alpha, t_1 ... t_{n-1}), t_n) = succ(succ_G(\alpha, t_1 ... t_{n-2}), t_{n-1} t_n)$.

Otherwise, using C1 of SC and Lemma 1, we can state that $t_{n-1}||t_n$, sequences $t_{n-1}t_n$ and t_nt_{n-1} are firable from $succ_G(\alpha,t_1...t_{n-2})$, and $succ(succ_G(\alpha,t_1...t_{n-1}),t_n) = succ(succ_G(\alpha,t_1...t_{n-2}),t_{n-1}t_n) \cup succ_{G(\alpha_{n-1})}(succ(succ_G(\alpha,t_1...t_{n-2}),t_n),t_{n-1})$.
Now, it suffices to repeat the same process for $succ(succ_G(\alpha,t_1...t_{n-2}),t_{n-1})$ and all derived terms, until reaching terms where *succ* is directly applied on α. As $G \models C1$, each time two adjacent transitions are permuted, they are firable in both order and effect-independent. Otherwise, they are at least firable in one order. Therefore, $succ_G(\alpha,\omega) \neq \emptyset \Rightarrow \exists \omega' \equiv \omega, succ(\alpha,\omega') \neq \emptyset$. □

Theorem 1 *Let $\mathcal{N}$ be a TPN with no unbounded static firing intervals. Then:* $\mathbb{G} \models SC \Rightarrow$ *the RSCG preserves non-equivalent sequences of the CSCG.*

Proof Let M be a marking and ω a firing sequence of M (i.e., $M[\omega>$). The sequence ω of M is maximal iff it is infinite or leads to a deadlock marking. Let $\Omega(M)$ be a set of maximal firing sequences of M. The RSCG preserves the non-equivalent firing sequences of the CSCG if: $\forall \alpha = (M,F) \in \mathscr{C}_S$,
(i) $\forall \omega \in \Omega(M), succ(\alpha,\omega) \neq \emptyset \Rightarrow \exists \omega' \in T^+, \omega \equiv \omega' \wedge succ_G(\alpha,\omega') \neq \emptyset$ and
(ii) $\forall \omega \in \Omega(M), succ_G(\alpha,\omega) \neq \emptyset \Rightarrow \exists \omega' \in T^+, \omega \equiv \omega' \wedge succ(\alpha,\omega') \neq \emptyset$.
(i): By assumption, ω is a maximal sequence of M and $succ(\alpha,\omega) \neq \emptyset$.
Then, $Fr(\alpha) \neq \emptyset$. By C0 of SC, $G(\alpha) \cap Fr(\alpha) \neq \emptyset$. From the fact that the TPN has no unbounded intervals, the non-zenoness, assumed here, guarantees that each enabled transition will eventually fire in the future, unless it is disabled by another firing. Conditions C1 and C2 ensure that the transitions outside $G(\alpha)$ cannot disable any transition of $G(\alpha)$. By C3 and Lemma 2, $\exists t_f \in G(\alpha) \cap Fr(\alpha), \exists \omega_1 \in (T - (G(\alpha) \cap Fr(\alpha)))^*, \exists \omega_2 \in T^*,\ \omega = \omega_1 t_f \omega_2 \wedge succ(\alpha,\omega_1) \neq \emptyset \wedge succ(\alpha,\omega_1 t_f) \neq \emptyset \wedge succ(\alpha,t_f\omega_1) \neq \emptyset$. Since $succ(\alpha,t_f\omega_1) \subseteq succ(succ_{G(\alpha)}(\alpha,t_f),\omega_1)$, it follows that $succ(succ_{G(\alpha)}(\alpha,t_f),\omega_1\omega_2) \neq \emptyset$.
Let $\alpha_1 = succ_{G(\alpha)}(\alpha,t_f) = (M_1,F_1)$. The sequence $\omega_1\omega_2$ is a maximal sequence of M_1. We repeat the same process for α_1 and $\omega_1\omega_2$ until reaching a deadlock or a state class already processed. Therefore, $\exists \omega' \in T^+, \omega' \equiv \omega \wedge succ_G(\alpha,\omega') \neq \emptyset$.
(ii): is immediate from Lemma 2. □

For a TPN with unbounded firing intervals, the non-zenoness, assumed here, guarantees that each enabled transition will become firable in the future, unless it is disabled by another firing. However, the firing of a transition, with an unbounded static firing interval, may be delayed indefinitely to lead in the reduced graph to some cycle such that the transition is firable from all state classes of the cycle but does not belong to their G (unfair sequence). The fairness criterion (we must not indefinitely neglect some transition) is not guaranteed by SC. To deal with the fairness criterion, G has to satisfy, in addition to SC, the Cycle closing condition, i.e., for every cycle in the reduced state class graph, there is at least one state class s.t. its G is equal to its set of firable transitions (fully expanded node) considered in [22] to address the same problem. With this additional condition, Theorem 1 is also valid for TPNs with unbounded static firing intervals.

Table 3 Some experimental results

TPN	RSCG	RSCG'	CSCG	TPN	RSCG	RSCG'	CSCG
KB(1)				*KB*(2)			
NSC	32	40	61	NSC	102135	? > 236977	? > 207685
NCSC	38	54	107	NCSC	133764	> 457101	> 551187
CPU (s)	0	0	0	CPU (s)	269	> 3600	> 3600
HC(1)				*HC*(2)			
NSC	19	31	70	NSC	133	289	1743
NCSC	18	33	110	NCSC	165	455	4603
CPU (s)	0	0	0	CPU (s)	0	0	0
HC(3)				*HC*(4)			
NSC	497	1714	23299	NSC	2895	11524	? > 138335
NCSC	682	2633	84184	NCSC	4362	17279	> 590080
CPU (s)	0	0	45	CPU (s)	0	14	> 3600
HC(5)				*HC*(6)			
NSC	10239	75251	?	NSC	16846	? > 221731	?
NCSC	16831	112606		NCSC	27210	> 338461	
CPU (s)	3	381		CPU (s)	9	> 3600	
FMS(2)				*FMS*(3)			
NSC	928	12668	82665	NSC	84176	? > 284319	? > 227052
NCSC	1201	19337	233208	NCSC	109930	> 430854	> 618528
CPU (s)	0	8	413	CPU (s)	170	> 3600	> 3600

Table 4 Static firing intervals of HC, FMS and KB

Is of *HC*	*Is* of *FMS*	*Is* of *KB*
$t_1[1,2]$	$tp_1[1,2]$	$tsynch4-23[1,3]$
$t_2[2,3]$	$tp_2[1,2]$	$tsynch1-23[3,5]$
$t_3[3,3]$	$tp_3[1,2]$	$tredo1[2,2]$
$t_4[1,1]$	$tm_1[1,1]$	$tok1[3,4]$
$t_5[1,2]$	$tm_2[3,4]$	$tback1[1,3]$
$t_6[1,2]$	$tp_3m_2[4,4]$	$tout1[3,5]$
$t_7[3,3]$	$tp_3s[1,2]$	$tredo2[2,2]$
$t_8[2,2]$	$tp_1m1[1,1]$	$tok2[3,4]$
$t_9[1,1]$	$tp_2m_2[3,3]$	$tback2[1,3]$
$t_{10}[1,1]$	$tp_1e[5,5]$	$tredo3[3,5]$
$t_{11}[1,2]$	$tp_1j[3,4]$	$tok3[2,2]$
$t_{12}[2,3]$	$tp_2j[1,1]$	$tback3[3,4]$
$t_{13}[1,1]$	$tp_2e[1,1]$	$tin4[1,3]$
$t_{14}[1,1]$	$tp_1s[3,3]$	$tredo4[3,5]$
$t_{15}[1,1]$	$tp_{12}[2,2]$	$tok4[2,2]$
$t_{16}[1,2]$	$tp_2s[4,4]$	$tback4[3,4]$
$t_{17}[1,1]$	$tm_3[1,2]$	
$t_{18}[1,4]$	$tp_{12}m_3[1,1]$	
	$tp_{12}s[2,2]$	
	$tx[5,5]$	

5 Experimental results

We have tested the partial order technique, proposed here, on several small TPNs, the extension with static firing intervals of three models taken from the MCC (Model Checking Contest) held within Petri Nets 2013[11]: HouseConstruction (*HC* in short), *FMS* and Kanban (*KB* in short) (see Table 3 for their static firing intervals). Table 4 reports the number of state classes (NSC), the number of computed state classes

[11] http://mcc.lip6.fr

(NCSC) and the CPU time in seconds of the RSCG, RSCG' and CSCG for *HC* and *FMS* and *KB*. The column RSCG' is the reduced graph of the approach proposed in [8]. Note that for a state class α, $G(\alpha)$ is computed by choosing randomly a firable transition t_f from $Fr(\alpha)$, then applying recursively, C1, C2 and C3 until a fix point is reached. Its size is dependent on the first selected transition.

The models $HC(n)$ and $KB(n)$ are free-choice and connected TPNs[12]. For $HC(n)$, n is the initial marking of the source place p_1. For $KB(n)$, n is the initial marking of places p_1, p_2, p_3 and p_4. The model $FMS(n)$ is a strongly-connected TPN[13], n being the initial marking of places p_1, p_2 and p_3.

For all tested models, the RSCG shows a significant reduction in time and number of computed state classes, compared to the CSCG and the RSCG'. The gain (in time and space) of the RSCG over the CSCG and RSCG' is much more significant for the connected TPNs ($HC(n)$) and ($KB(n)$) than the strongly-connected TPN ($FMS(n)$). The reason is that in the strongly-connected TPN, several transitions are in conflict or not effect-dependent. Furthermore, we obtain further reduction, when we increase the marking, as it results in increasing the number of concurrent enabled transitions.

6 Conclusion

In this paper, we have considered the TPN model and proposed, using its CSCG, a partial order reduction technique, which preserves non-equivalent firing sequences of the TPN.

Our technique is inspired from the stubborn sets [25,27] but is based on the notion of effect-independent transitions, instead of the notion of truth parent used in [27]. The notion of truth parent involves to keep, in each abstract state, in addition to time constraints of the enabled transitions, those of their parents. All possible parents of the enabled transitions are considered separately when computing successors of abstract states. Moreover, as in [8], our technique takes into account the (static and dynamic) timing information of the model.

For the tested models, our technique allows a significant gain in time and space, in comparison with the RSCG of [8] and the CSCG.

References

1. Belluomini, W., Myers, C.J.: Timed state space exploration using POSETs. IEEE Transactions on Computer-Aided Design of Integrated Circuits **19(5)**, 501 – 520 (2000)

[12] A free-choiceTPN is a TPN, where for every transition t, $pre(t) \leq P$ and $post(t) \leq P$ and the sets of input places of any pair of transitions are either equal or disjoint. In a strongly-connected TPN, there is a directed path between every two nodes (places or transitions).

[13] In a connected TPN, there is an undirected path between every two nodes.

2. Bengtsson, J.: Clocks, DBMs and States in Timed Systems. PhD thesis, Dept. of Information Technology, Uppsala University (2002)
3. Bengtsson, J., Jonsson, B., Lilius, J., Yi, W.: Partial order reductions for timed systems. In: 9th international conference on Concurrency Theory (CONCUR), *LNCS*, vol. 1466, pp. 485 – 500 (1998)
4. Bérard, B., Cassez, F., Haddad, S., Lime, D., Roux, O.H.: The expressive power of time Petri nets. Theoretical Computer Science **474**, 1–20 (2013)
5. Berthomieu, B., Vernadat, F.: State class constructions for branching analysis of time Petri nets. In: 9th International Conference of Tools and Algorithms for the Construction and Analysis of Systems, *LNCS*, vol. 2619, pp. 442–457 (2003)
6. Boucheneb, H., Barkaoui, K.: Reducing interleaving semantics redundancy in reachability analysis of time Petri nets. ACM Transactions on Embedded Computing Systems (TECS) **12(1)**, 259 – 273 (2013)
7. Boucheneb, H., Barkaoui, K.: Stubborn sets for time Petri nets. ACM Transactions in Embedded Computing Systems (TECS) **14(1)**, 11:1 – 11:25 (2015)
8. Boucheneb, H., Barkaoui, K., Weslati, K.: Delay-dependent partial order reduction technique for time Petri nets. In: 12th International Conference on Formal Modeling and Analysis of Timed Systems, *LNCS*, vol. 8711, pp. 53 – 68 (2014)
9. Boucheneb, H., Gardey, G., Roux, O.H.: TCTL model checking of time Petri nets. Logic and Computation **19(6)**, 1509–1540 (2009)
10. Boucheneb, H., Hadjidj, R.: CTL* model checking for time Petri nets. Theoretical Computer Science TCS **353/1-3**, 208227 (2006)
11. Boucheneb, H., Lime, D., Roux, O.H.: On multi-enabledness in time Petri nets. In: 34th International Conference on Application and Theory of Petri Nets and other models of concurrency (ICATPN), *LNCS*, vol. 7927, pp. 130 – 149 (2013)
12. Boucheneb, H., Rakkay, H.: A more efficient time Petri net state space abstraction useful to model checking timed linear properties. Fundamenta Informaticae **88(4)**, 469–495 (2008)
13. Boyer, M., Diaz, M.: Multiple-enabledness of transitions in time Petri nets. In: 9th IEEE International Workshop on Petri Nets and Performance Models, pp. 219 – 228 (2001)
14. Chatain, T., Jard, C.: Complete finite prefixes of symbolic unfoldings of safe time Petri nets. In: 27th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency ICATPN, *LNCS*, vol. 4024, pp. 125 – 145 (2006)
15. Delfieu, D., Sogbohossou, M., Traonouez, L.M., Revol, S.: Parameterized study of a time Petri net. In: Cybernetics and Information Technologies, Systems and Applications: CITSA, pp. 89 – 90 (2007)
16. Godefroid, P.: An approach to the state-explosion problem. In: Partial-Order Methods for the Verification of Concurrent Systems, *LNCS*, vol. 1032, pp. 1–142 (1996)
17. Hakansson, J., Pettersson, P.: Partial order reduction for verification of real-time components. In: 5th international conference on Formal Modeling and Analysis of Timed Systems (FORMATS), LNCS, pp. 211 – 226 (2007)
18. Lilius, J.: Efficient state space search for time Petri nets. In: MFCS Workshop on Concurrency Algorithms and Tools, *ENTCS*, vol. 8, pp. 113–133 (1998)
19. Lugiez, D., Niebert, P., Zennou, S.: A partial order semantics approach to the clock explosion problem of timed automata. Theoretical Computer Science TCS **345(1)**, 2759 (2005)
20. Minea, M.: Partial order reduction for model checking of timed automata. In: 10th international conference on Concurrency Theory (CONCUR), *LNCS*, vol. 1664, pp. 431 – 446 (1999)
21. Peled, D.: All from one, one for all: on model checking using representatives. In: Computer Aided Verification, *LNCS*, vol. 697, pp. 409–423 (1993)
22. Peled, D., Wilke, T.: Stutter invariant temporal properties are expressible without the next-time operator. Information Processing Letters **63 issue 5**, 243–246 (1997)
23. Salah, R.B., Bozga, M., Maler, O.: On interleaving in timed automata. In: 17th international conference on Concurrency Theory (CONCUR), *LNCS*, vol. 4137, pp. 465 – 476 (2006)
24. Semenov, A., Yakovlev, A.: Verification of asynchronous circuits using time Petri net unfolding. In: 33*rd* annual conference on Design automation (DAC), pp. 59 – 62 (1996)
25. Valmari, A., Hansen, H.: Can stubborn set be optimal. In: 3st International conference on Application and Theory of Petri Nets and Concurrency (Petri Nets), *LNCS*, vol. 6128, pp. 43 – 62 (2010)
26. Yoneda, T., Ryuba, H.: CTL model checking of time Petri nets using geometric regions. EICE Trans. Inf. & Syst. **E99-D(3)**, 297–306 (1998)
27. Yoneda, T., Schlingloff, B.H.: Efficient verification of parallel real-time systems. Formal Methods in System Design **11**(2), 187–215 (1997)