On interleaving in \{P,A\}-Time Petri nets with strong semantics

Hanifa Boucheneb
Laboratoire VeriForm, École Polytechnique de Montréal, P.O. Box 6079, Station Centre-ville, Montréal, Québec, Canada, H3C 3A7
hanifa.boucheneb@polymtl.ca

Kamel Barkaoui
Laboratoire CEDRIC, Conservatoire National des Arts et Métiers, 292 rue Saint Martin, Paris Cedex 03, France
kamel.barkaoui@cnam.fr

This paper deals with the reachability analysis of \{P,A\}-Time Petri nets (\{P,A\}-TPN in short) in the context of strong semantics. It investigates the convexity of the union of state classes reached by different interleavings of the same set of transitions. In [6], the authors have considered the T-TPN model and its Contracted State Class Graph (CSCG) [7] and shown that this union is not necessarily convex. They have however established some sufficient conditions which ensure convexity. This paper shows that for the CSCG of \{P,A\}-TPN, this union is convex and can be computed without computing intermediate state classes. These results allow to improve the forward reachability analysis by agglomerating, in the same state class, all state classes reached by different interleavings of the same set of transitions (abstraction by convex-union).

1 introduction

Petri nets are established as a suitable formalism for modeling concurrent and dynamic systems. They are used in many fields (computer science, control systems, production systems, etc.). Several extensions to time factor have been defined to take into account different features of the system as well as its time constraints. The time constraints may be expressed in terms of stochastic delays of transitions (stochastic Petri nets), fixed values associated with places or transitions (\{P,T\}-Timed Petri nets), or intervals labeling places, transitions or arcs (\{P,T,A\}-Time Petri Nets) [9, 11, 13]. For \{P,T,A\}-Time Petri Nets, there are two firing semantics: Weak Time Semantics (WTS) and Strong Time Semantics (STS). For both semantics, each enabled transition has an explicit or implicit firing interval derived from time constraints associated with places, transitions or arcs of the net. A transition cannot be fired outside its firing interval, but in WTS, its firing is not forced when the upper bound of its firing interval is reached. Whereas in STS, it must be fired within its firing interval unless it is disabled. The STS is the most widely used semantics. There are also multiple-server and single-server semantics. The multiple-server semantics allows to handle, at the same time, several time intervals per place (P-TPN), per arc (A-TPN) or per transition (T-TPN) whereas it is not allowed in the single-server semantics.

In [8], the authors have compared the expressiveness of \{P,T,A\}-TPN models with strong (X-TPN, X ∈ \{P,T,A\}) and weak semantics (X-TPN, X ∈ \{P,T,A\}) (see Figure 1). They have established that [11].

- For the single-server semantics, bounded \{P,T,A\}-TPN and safe \{P,T,A\}-TPN are equally expressive w.r.t. timed-bisimilarity and then w.r.t. timed language acceptance.

---

1 A Petri net is bounded if the number of tokens in each reachable marking is bounded. It is safe if the number of tokens in each reachable marking cannot exceed one.
On interleaving in \{P,A\}-Time Petri nets

- T-TPN and P-TPN are incomparable models.
- A-TPN includes all the other models.
- The strong semantics includes the weak one for P-TPN and A-TPN, but not for T-TPN.

Figure 1: Comparison of the expressiveness of \{P,T,A\}-TPNs given in [8]

The reachability analysis of \{P,T,A\}-TPN is, in general, based on abstractions preserving properties of interest (markings or linear properties). In general, in the abstractions preserving linear properties, we distinguish three levels of abstraction. In the first level, states reachable by time progression may be either represented or abstracted. In the second level, states reachable by the same sequence of transitions independently of their firing times are agglomerated in the same node. In the third level, the agglomerated states are considered modulo some equivalence relation: the firing domain of the state class graph (SCG) [4], the bisimulation relation over the SCG of the contracted state class graph (CSCG) [7], the approximations of the zone based graph (ZBG) [5]). An abstract state is then an equivalence class of this relation. Usually, all states within an abstract state share the same marking and the union of their time domains is convex and defined as a conjunction of atomic constraints.

From the practical point of view, the Difference Bound Matrices (DBMs) are a useful data structure for representing and handling efficiently sets of atomic constraints [1].

The classical forward reachability analysis consists of computing, on-the-fly, all abstract states that are reachable from the initial abstract state. The reachability problem is known to be decidable for bounded \{P,T,A\}-TPN but the reachability analysis suffers from the state explosion problem. For timed models, this problem is accentuated by the fact that, in the state space abstraction, a node represents, in fact, a finite/infinite set of states (abstract state) and interleavings of concurrent transitions lead, in general, to different abstract states.

To attenuate the state explosion problem, the reachability analysis is usually based on an abstraction by inclusion or by convex-union. During the construction of an abstraction, each newly computed abstract state is compared with the previously computed ones. In the abstractions by inclusion, two abstract states, with the same marking, having domains such that one is included in the other are grouped into one node. In the abstractions by convex-union, two abstract states, with the same marking, having domains such that their union is convex (and then can be represented by a single DBM), are grouped into one node. Convex-union abstractions are more compact than inclusion abstractions [10].

\[^2\] An atomic constraint is of the form \(x - y \leq c, x \leq c \text{ or } -x \leq c\), where \(x, y\) are real valued variables representing clocks or delays, \(c \in \mathbb{Q} \cup \{\infty\}\) and \(\mathbb{Q}\) is the set of rational numbers (for economy of notation, we use operator \(\leq\) even if \(c = \infty\)).
ever, it is known that DBMs are not closed under union and the convex-union test is a very expensive operation relatively to the test of inclusion [10]. The convex-union test of $n$ (with $n > 1$) abstract states $\alpha_1 = (M,D_1), \alpha_2 = (M,D_2), \ldots, \alpha_n = (M,D_n)$ involves computing the smallest enclosing DBM $\alpha = (M,D)$ of their union, the difference between $D$ and $D_1,D_2,\ldots D_{n-1}$, and finally checking that this difference is included in $D_n$.

Another interesting reachability analysis approach, proposed in [2] for a CSS-like parallel composition of timed automata, consists of computing abstract states in breadth-first manner and at each level grouping, in one abstract state, all abstract states reached by different interleavings of the same set of concurrent transitions. The authors have shown that this union is convex, and then does not need any test of convexity. To use this approach in the context of \{P,T,A\}-TPN, we need to show that the union of abstract states reached by different interleavings of the same set of transitions is convex. In [3], the authors have shown that for the T-TPN model, this union is not necessarily convex in the SCG and the CSCG. This paper shows that for the P-TPN, this union is not necessarily convex in the SCG but is convex in the CSCG. Finally, it shows that these results are also valid for the A-TPN model.

The next section is devoted to the P-TPN model, its semantics, its SCG, its CSCG, and the proof that the union of abstract states (i.e., state classes) reached by different interleavings of the same set of transitions is not necessarily convex in the SCG but is convex in the CSCG. Moreover, this union can be computed directly without computing beforehand intermediate state classes. Section 3 extends the results shown in Section 2 to the A-TPN model. Section 4 contains concluding remarks.

## 2 P-Time Petri Nets

In this paper, for reasons of clarity, we consider safe P-Time Petri nets.

### 2.1 Definition and behavior

A P-Time Petri net is a Petri net augmented with time intervals associated with places. Formally, a P-TPN is a tuple $(P,T,Pre,Post,M_0,Isp)$ where:

1. $P = \{p_1,\ldots,p_m\}$ and $T = \{t_1,\ldots,t_n\}$ are nonempty and finite sets of places and transitions such that $P \cap T = \emptyset$,
2. $Pre$ and $Post$ map each transition to its preset and postset $(Pre, Post : T \longrightarrow 2^P, Pre(t_i) = ^\circ t_i \subseteq P, Post(t_i) = t_i^\circ \subseteq P)$,
3. $M_0$ is the initial marking ($M_0 \subseteq P$),
4. $Isp$ is the static residence interval function $(Isp : P \rightarrow \mathbb{Q}^+ \times (\mathbb{Q}^+ \cup \{\infty\}))$, $\mathbb{Q}^+$ is the set of nonnegative rational numbers. $Isp(p_i)$ specifies the lower $\downarrow Isp(p_i)$ and the upper $\uparrow Isp(p_i)$ bounds of the static residence interval in place $p_i$.

Let $M \subseteq P$ be a marking and $t_i$ a transition of $T$. Transition $t_i$ is enabled for $M$ iff all required tokens for firing $t_i$ are present in $M$, i.e., $Pre(t_i) \subseteq M$. The firing of $t_i$ from $M$ leads to the marking $M' = (M - Pre(t_i)) \cup Post(t_i)$. The set of transitions enabled for $M$ is denoted $En(M)$, i.e., $En(M) = \{t_i \in T \mid Pre(t_i) \subseteq M\}$. A transition $t_k \in En(M)$ is in conflict with $t_i$ in $M$ iff $Pre(t_k) \cap Pre(t_i) \neq \emptyset$. The firing of $t_i$ will disable $t_k$.

In this model, a token may die. A token of place $p$ dies when its interval becomes empty. Dead tokens will never be used and are considered as modeling flaws that should be avoided. To detect the dead tokens, we add a special transition named $Err$ whose role is limited to die tokens.
The P-TPN state is defined as a triplet \( s = (M, \text{Dead} p, \text{Ip}) \), where \( M \subseteq P \) is a marking, \( \text{Dead} p \subseteq M \) is the set of dead tokens in \( M \) and \( \text{Ip} \) is the residence interval function \( (\text{Ip} : M \rightarrow \text{Dead} p \rightarrow \mathbb{Q}^+ \times (\mathbb{Q}^+ \cup \{\infty\})) \). The initial state of the P-TPN model is \( s_0 = (M_0, \text{Dead} p_0, \text{Ip}_0) \) where \( \text{Dead} p_0 = \emptyset \), \( \text{Ip}_0(p_i) = I\text{sp}(p_i) \), for all \( p_i \in M_0 \). When a token is created in place \( p_i \), its residence interval is set to its static residence interval \( I\text{sp}(p_i) \). The bounds of this interval decrease synchronously with time, until the token of \( p_i \) is consumed or dies. A transition \( t_i \) can fire if all its input tokens are available, i.e., the lower bounds of their residence intervals have reached 0, but must fire, without any additional delay, if the upper bound of, at least, one of its input tokens reaches 0. The firing of a transition takes no time.

We define the P-TPN semantics as follows: Let \( s = (M, \text{Dead} p, \text{Ip}) \) and \( s' = (M', \text{Dead} p', \text{Ip}') \) be two states of a P-TPN, \( d \in \mathbb{R}^+ \) a nonnegative real number and \( t_f \in T \) a transition of the net.

- We write \( s \xrightarrow{d} s' \), also denoted \( s + d \), iff the state \( s' \) is reachable from state \( s \) by a time progression of \( d \) units, i.e., \( \forall p_i \in M - \text{Dead} p \), \( d \leq \uparrow \text{Ip}(p_i) \), \( M' = M, \text{Dead} p' = \text{Dead} p \), and \( \forall p_j \in M' - \text{Dead} p', \text{Ip}'(p_j) = \left[\text{Max}(0, \downarrow \text{Ip}(p_j) - d), \uparrow \text{Ip}(p_j) - d\right] \). The time progression is allowed while we do not overpass residence intervals of all non dead tokens. No token may die by this time progression.

- We write \( s \xrightarrow{t_f} s' \) iff state \( s' \) is immediately reachable from state \( s \) by firing transition \( t_f \), i.e., \( \text{Pre}(t_f) \subseteq M - \text{Dead} p \), \( \forall p_i \in \text{Pre}(t_f), \downarrow \text{Ip}(p_i) = 0, M' = (M - \text{Pre}(t_f)) \cup \text{Post}(t_f), \text{Dead} p' = \text{Dead} p \), and \( \forall p_i \in M' - \text{Dead} p', \text{Ip}'(p_i) = I\text{sp}(p_i) \), if \( p_i \in \text{Post}(t_f) \) and \( \text{Ip}'(p_i) = \text{Ip}(p_i) \) otherwise.

- We write \( s \xrightarrow{Err} s' \) iff state \( s' \) is immediately reachable from state \( s \) by firing transition \( \text{Err} \). Transition \( \text{Err} \) is immediately firable from \( s \) if there exists no transition firable from \( s \) and there is, at least, a token in \( M - \text{Dead} p \) s.t. the upper bound of its interval has reached 0 (token to die) i.e., \( \exists t_k \in \text{En}(M - \text{Dead} p), \exists p_j \in \text{Pre}(t_k), \downarrow \text{Ip}(p_j) > 0, (\exists p_i \in M - \text{Dead} p, \uparrow \text{Ip}(p_i) = 0), M' = M, \text{Dead} p' = \text{Dead} p \cup \{ p_j \in M - \text{Dead} p | \uparrow \text{Ip}(p_j) = 0 \} \), and \( \forall p_i \in M' - \text{Dead} p', \text{Ip}'(p_i) = \text{Ip}(p_i) \).

According to the above semantics, states from which transition \( \text{Err} \) is firable, are timelock states. Therefore, transition \( \text{Err} \) allows to detect timelock states and dead tokens, and also to unblock the time progression.

The P-TPN state space is the timed transition system \( (\mathcal{S}, \rightarrow, s_0) \), where \( s_0 \) is the initial state of the P-TPN and \( \mathcal{S} = \{ s | s \xrightarrow{} s \} \) is the set of reachable states of the model, \( \xrightarrow{} \) being the reflexive and transitive closure of the relation \( \rightarrow \) defined above.

A run in the P-TPN state space \( (\mathcal{S}, \rightarrow, s_0) \), starting from a state \( s \), is a maximal sequence \( \rho = s_1 \xrightarrow{d_1} s_1 + d_1 \xrightarrow{t_2} s_2 \cdots \) such that \( s_1 = s \). By convention, for any state \( s_i \), relation \( s_i \xrightarrow{} s_i \) holds. The sequence \( d_1 t_1 d_2 t_2 \cdots \) is called the timed trace of \( \rho \). The sequence \( t_1 t_2 \cdots \) is called the untimed trace of \( \rho \). Runs of the P-TPN are all runs starting from the initial state \( s_0 \). Its timed (resp. untimed) traces are timed (resp. untimed) traces of its initial state.

### 2.2 The SCG and CSCG of P-TPN

The SCG of P-TPN is defined in a similar way as the SCG of T-TPN, except that time constraints are associated with places, and tokens may die. A SCG state class is defined as a triplet \( \alpha = (M, \text{Dead} p, \phi_p) \), where \( M \subseteq P \), \( \text{Dead} p \subseteq M \) is the set of dead tokens in \( M \) and \( \phi_p \) is a conjunction of atomic constraints characterizing the union of the residence intervals of its non dead tokens. Each place \( p_i \) of \( M - \text{Dead} p \) has a variable denoted \( p_i \) in \( \phi_p \) representing the residence delay of its token (i.e., the waiting time before its consummation or its death).

---

3 A state \( s \) is a timelock state iff no progression of time is possible and no transition is firable from \( s \).

4 An atomic constraint is of the form \( x - y \leq c, x \leq c, -y \leq c \), where \( x, y \) are real valued variables, \( c \in \mathbb{Q} \cup \{\infty\} \) and \( \mathbb{Q} \) is the set of rational numbers (for economy of notation, we use operator \( \leq \) even if \( c = \infty \)).
From the practical point of view, $\phi_p$ is represented by a Difference Bound Matrix (DBM). The DBM of $\phi_p$ is a square matrix $D$ of order $|M - \text{Dead}| + 1$, indexed by variables of $\phi_p$ and a special variable $p_0$, whose value is fixed at 0. Each entry $d_{ij}$ represents the atomic constraint $p_i - p_j \leq d_{ij}$. Hence, entries $d_{00}$ and $d_{0j}$ represent simple atomic constraints $p_0 \leq d_{00}$ and $-p_j \leq d_{0j}$, respectively. If there is no upper bound on $p_i - p_j$ with $i \neq j$, $d_{ij}$ is set to $\infty$. Entry $d_{ii}$ is set to 0. Though the same nonempty domain may be represented by different DBMs, they have a unique form called 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. It can be computed in $O(n^3)$, $n$ being the number of variables in the DBM, using a shortest path algorithm, like Floyd-Warshall’s all-pairs shortest path algorithm [1]. Canonical forms make operations over DBMs much simpler [3].

The initial state class is $\alpha_0 = (M_0, \text{Dead}p_0, \phi_0)$ where $M_0$ is the initial marking, $\text{Dead}p_0 = \emptyset$ and $\phi_0 = \bigwedge_{p_i \in M_0} \downarrow \text{Isp}(p_i) \leq p_i \leq \uparrow \text{Isp}(p_i)$.

Successor state classes are computed using the following firing rule [4]: Let $\alpha = (M, \text{Dead}p, \phi_p)$ be a state class and $t_f$ a transition of $T$. The state class $\alpha$ has a successor by $t_f$ (i.e., $\text{succ}(\alpha, t_f) \neq \emptyset$) iff $\text{Pre}(t_f) \subseteq M - \text{Dead}p$ and the following formula is consistent [5]

$$\phi_p \land \bigwedge_{p_j \in \text{Pre}(t_f), p_i \in M - \text{Dead}p} (p_f - p_i \leq 0).$$

This firing condition means that $t_f$ is enabled in $M - \text{Dead}p$ and there is a state s.t. the residence delay of each input token of $t_f$ is less or equal to the residence delays of all non dead tokens in $M$.

If $\text{succ}(\alpha, t_f) \neq \emptyset$ then $\text{succ}(\alpha, t_f) = (M', \text{Dead}p', \phi'_p)$ is computed as follows:

1. $M' = (M - \text{Pre}(t_f)) \cup \text{Post}(t_f)$;
2. $\text{Dead}p' = \text{Dead}p$;
3. Set $\phi'_p$ to $\phi_p \land \bigwedge_{p_j \in \text{Pre}(t_f), p_i \in M - \text{Dead}p} (p_f - p_i \leq 0)$;
4. Rename, in $\phi'_p$, $p_f$ in $L_f$, for all $p_f \in \text{Pre}(t_f)$;
5. Add constraints: $\bigwedge_{p_i \in \text{Post}(t_f)} (\downarrow \text{Isp}(p_i) \leq p_i - L_f \leq \uparrow \text{Isp}(p_i))$;
6. Replace each variable $p_i$ by $p_i + L_f$ (this substitution actualizes delays (old $p_i$ = new $p_i + L_f$));
7. Eliminate by substitution $L_f$.

If $t_f$ is firable then its firing consumes its input tokens and creates a token in each of its output places. Step 2) means that no token may die by firing $t_f$. Step 3) isolates states of $\alpha$ from which $t_f$ is firable. Note that this firing condition implies that $\forall p_f, p'_f \in \text{Pre}(t_f), p_f = p'_f$, and then the firing delay $L_f$ of $t_f$ is equal to $p_f$. Step 4) renames variables associated with tokens consumed by $t_f$ in $L_f$. Step 5) adds constraints of the created tokens. The residence interval of a token created by $t_f$ is relative to the firing date of $t_f$. Step 6) updates the delays of tokens not used by $t_f$. Step 7) eliminates variable $L_f$.

For example, consider the P-TPN shown in Figure 2.a. From its initial SCG state class $\alpha_0 = (p_1 + p_3, 0, 1 \leq p_1 \leq 3 \land 2 \leq p_3 \leq 4)$, transition $t_1$ is firable from $\alpha_0$, since $1 \leq p_1 \leq 3 \land 2 \leq p_3 \leq 4 \land p_1 - p_3 \leq 0$ is consistent. The firing of $t_1$ leads to the state class $(p_2 + p_3, 0, 0 \leq p_2 \leq 3 \land p_3 = 1)$. Its formula is derived from the firing condition of $t_1$ from $\alpha_0$ as follows: rename $p_1$ in $L_1$, add the constraint

\footnote{A formula $\phi$ is consistent iff there is, at least, one tuple of values that satisfies, at once, all constraints of $\phi$.}
1 \leq p_3 - L_1 \leq 1$, replace $p_3$ and $p_4$ by $p_3 + \mathcal{L}_1$ and $p_3 + \mathcal{L}_1$, respectively, and finally eliminate by substitution $\mathcal{L}_1$.

The transition $\text{Err}$ is firable from $\alpha = (M, \text{Dead} p, \phi_p)$ iff there is no possibility to reach the intervals of input places of any enabled transition without overpassing the interval of a non dead token, i.e., $\exists p_1 \in M - \text{Dead} p$, s.t. $\forall t_f \in \text{En}(M - \text{Dead} p), \phi_p \land (\bigwedge_{p_f \in \text{Pre}(t_f)} p_f - p_1 \leq 0)$ is not consistent.

If $\text{Err}$ is firable from $\alpha$ (i.e., $\text{suc}(\alpha, \text{Err}) \neq \emptyset$), its firing leads to the state class $\alpha' = \text{suc}(\alpha, \text{Err}) = (M', \text{Dead} p', \phi'_p)$ where: $M' = M - \text{Dead} p \cup \{p_i \in M - \text{Dead} p | \forall t_f \in \text{En}(M - \text{Dead} p), \phi_p \land (\bigwedge_{p_f \in \text{Pre}(t_f)} p_f - p_1 \leq 0)$ is not consistent}, $\phi'_p$ is obtained from $\phi_p$ by eliminating by substitution all variables associated with places of $\text{Dead} p' - \text{Dead} p$ (i.e., by putting $\phi_p$ in canonical form and eliminating all variables associated with places of $\text{Dead} p' - \text{Dead} p$).

Let $\alpha, \alpha'$ be two state classes and $\chi \in T \cup \{\text{Err}\}$ a transition. We write $\alpha \xrightarrow{\chi} \alpha'$ iff $\text{suc}(\alpha, \chi) \neq \emptyset \land \alpha' = \text{suc}(\alpha, \chi)$. The SCG of the P-TPN is the structure $(\chi, \rightarrow, \alpha_0)$ where $\alpha_0$ is the initial state class and $\chi = \{\alpha | \alpha_0 \xrightarrow{} \alpha\}$ is the set of reachable state classes.

Note that dead tokens have no effect on the future behavior. Therefore, we can abstract dead tokens when we compare state classes. Two state classes $\alpha = (M, \text{Dead} p, \phi_p)$ and $\alpha' = (M', \text{Dead} p', \phi'_p)$ are said to be equal iff they have the same set of non dead tokens (i.e., $M - \text{Dead} p = M' - \text{Dead} p'$) and the DBMs of their formulas have the same canonical form (i.e., $\phi_p \equiv \phi'_p$).

In the same way as for the SCG of T-TPN [4], we can prove that the SCG of P-TPN is finite and preserves linear properties.

According to the firing rule given above, simple atomic constraints (i.e., atomic constraints of the form $p_{ij} \leq c$ or $-p_{ij} \leq c$) are not necessary to compute the successor state classes. It follows that all classes with the same triangular atomic constraints (i.e., atomic constraints of the form $p_{ij} - p_{ij} \leq c$) have the same firing sequences. They can be agglomerated into one node while preserving linear properties of the model. This kind of agglomeration has been successfully used in [7] for the SCG of the T-TPN.

Formally, we define a bisimulation relation, denoted $\simeq$, over the SCG of the P-TPN by: $\forall \alpha = (M, \text{Dead} p, \phi_p), \alpha' = (M', \text{Dead} p', \phi'_p) \in \chi$, let $D$ and $D'$ be the DBMs in canonical form of $\phi_p$ and $\phi'_p$, respectively, $(M, \text{Dead} p, \phi_p) \simeq (M', \text{Dead} p', \phi'_p)$ iff $M - \text{Dead} p = M' - \text{Dead} p'$ and $\forall p_i, p_j \in M - \text{Dead} p, d_{ij} = d'_{ij}$.

The CSCG of the P-TPN is the quotient graph of the SCG w.r.t. $\simeq$. A CSCG state class is an equivalence class of $\simeq$. It is defined as a triplet $\beta = (M, \text{Dead} p, \psi_p)$, where $\psi_p$ is a conjunction of triangular atomic constraints. The initial CSCG state class is $\beta_0 = (M_0, \text{Dead} p_0, \psi_{p_0})$ where $M_0$ is the initial marking, $\text{Dead} p_0 = \emptyset$ and $\psi_{p_0} = \bigwedge_{p_i, p_j \in M_0} p_i - p_j \leq \uparrow \text{Isp}(p_i) - \downarrow \text{Isp}(p_j)$.

The CSCG state classes are computed in the same manner as the SCG state classes, except that step
of the firing rule given above, is not needed because the substitution of each $p_i$ by $p_i + L_f$ has no effect on triangular atomic constraints $((p_i + L_f) - (p_j + L_f) = p_i - p_j)$. Steps 6) and 7) are replaced by: Put the resulting formula in canonical form and then eliminate all constraints containing $L_f$.

### 2.3 Interleaving in the P-TPN state class graph

Note that transition $Err$, used to detect timelock states and dead tokens, cannot be concurrent to any transition of $T$. So, there is no interleaving between $Err$ and transitions of $T$.

Let us first show, by means of a counterexample, that the union of the SCG state classes of a P-TPN, reached by different interleavings of the same set of transitions, is not generally convex.

Consider the P-TPN shown in Figure 2.a. From its initial SCG state class $\alpha_0 = (p_1 + p_2, \emptyset, 1 \leq L_1 \leq 3 \land 2 \leq L_2 \leq 4)$, sequences $t_1t_2$ and $t_2t_1$ lead respectively to the SCG state classes:

$\alpha_1 = (p_3 + p_4, \emptyset, 0 \leq L_3 \leq 1 \land L_4 = 2 \land -2 \leq L_3 - L_4 \leq -1)$ and

$\alpha_2 = (p_3 + p_4, \emptyset, L_3 = 1 \land 1 \leq L_4 \leq 2 \land -1 \leq L_3 - L_4 \leq 0)$.

The union of domains of $\alpha_1$ and $\alpha_2$ is obviously not convex.

Consider now the CSCG of the same net. From its initial CSCG state class $\beta_0 = (p_1 + p_2, \emptyset, -3 \leq L_1 \leq p_1 - p_2 \leq 1)$, sequences $t_1t_2$ and $t_2t_1$ lead to the CSCG state classes:

$\beta_1 = (p_3 + p_4, \emptyset, -2 \leq L_3 \leq p_4 \leq -1)$ and $\beta_2 = (p_3 + p_4, \emptyset, -1 \leq L_3 - p_4 \leq 0)$, respectively.

The union of domains of $\beta_1$ and $\beta_2$ is convex ($-2 \leq L_3 - p_4 \leq 0$).

We will show, in the following, that this result is always valid for the union of all the CSCG state classes reached by different interleavings of the same set of transitions. Let us first establish the firing condition of a sequence of concurrent transitions.

**Proposition 1** Let $\beta = (M, \text{Dead } p, \psi_p)$ be a CSCG state class, and $T_m \subseteq T$ a set of transitions enabled and not in conflict in $M - \text{Dead } p$, $\Omega(T_m)$ the set of all interleavings of transitions of $T_m$ and $\omega = t_1t_2...t_m \in \Omega(T_m)$. The successor of $\beta$ by $\omega$ is non empty (i.e., $\text{suc}(\beta, \omega) \neq \emptyset$) iff the following formula, denoted $\varphi_p$, is consistent:

$$\exists f \in [1, m] \left[ \sum_{i \in \text{Pre}(t_f)} p_i = L_f \land \sum_{i \in (M - \text{Dead } p) - \bigcup_{l \in [1, f]} \text{Pre}(t_l)} L_f - p_i \leq 0 \land \sum_{k \in [1, f], p_k \in \text{Post}(t_k)} L_f - p_k \leq 0 \land \sum_{k \in [1, f], p_k \in \text{Post}(t_k)} \downarrow \text{Isp}(p_k) \leq p_i^f - L_f \leq \uparrow \text{Isp}(p_k) \right]$$

**Proof** By assumption, all transitions of $T_m$ are not in conflict (i.e., $\forall t_1, t_2 \in T_m$ s.t. $t_1 \neq t_2$, $\text{Pre}(t_1) \cap \text{Pre}(t_2) = \emptyset$). The firing condition of the sequence $t_1t_2...t_m$ from $\alpha$ adds to $\psi_p$ the firing constraints of transitions of the sequence (for $f \in [1, m]$). We add for each transition $t_f$ of the sequence, a variable, denoted $L_f$, representing its firing delay. The added constraints consist of five blocks. The first block fixes the firing order of transitions of $T_m$. The second block means that the residence delays of tokens used by each transition $t_f$ must be equal to $L_f$. The third and the fourth blocks mean that the firing delay $L_f$ is less or equal to the residence delays of tokens that are present (and not dead) when $t_f$ is fired (i.e., $t_f \in (M - \text{Dead } p) - \bigcup_{l \in [1, f]} \text{Pre}(t_l)$ and $p_n \in \bigcup_{k \in [1, f]} \text{Post}(t_k)$). The fifth block of constraints specifies the residence delays of tokens created by $t_f$ (i.e., $p_n \in \text{Post}(t_f)$). Note that $p_i^f$ denotes the residence delay of the token $p_n$ created by $t_f$.

$suc(\beta, \omega)$ is the set of all states reachable from any state of $\beta$ by a timed run supporting $\omega$. 
As an example, consider the P-TPN shown in Figure 2.b) and its initial CSCG state class $\beta_0 = (p_1 + p_2, 0, -5 \leq p_2 - p_3 \leq 1)$. The firing condition $\varphi_{p_1}$ of the sequence $t_1t_2$ is computed as follows:

1. Set $\varphi_{p_1}$ to $-5 \leq p_2 - p_3 \leq 1$.
2. Add variables $t_1$ and $t_2$ and the constraint $t_1 \leq t_2$.
3. Add constraints specifying the firing delays of $t_1$ and $t_2$: $L_1 = p_1 \wedge L_2 = p_2$.
4. Add constraints of tokens created by $t_1$: $1 \leq p_2 - L_1 \leq 5 \wedge 0 \leq p_3 - L_1 \leq 2$.
5. Add constraints specifying that the firing delay of $t_2$ is less or equal to the residence delays of the tokens created by $t_1$: $L_2 \leq p_3 \wedge t_2 \leq p_3$.

Then: $\varphi_{p_1} = \neg 5 \leq p_2 - p_3 \leq 1 \wedge (L_1 = p_1 \wedge L_2 = p_2) \wedge (t_1 \leq t_2) \wedge (L_2 \leq p_3 \wedge t_2 \leq p_3) \wedge (1 \leq p_2 - L_1 \leq 5 \wedge 0 \leq p_3 - L_1 \leq 2) \wedge (4 \leq p_4 - L_2 \leq 4 \wedge 0 \leq p_6 - L_2 \leq 2)$

In the same manner, we obtain the firing condition $\varphi_{p_2}$ of the sequence $t_2t_1$ from $\beta_0$:

$\varphi_{p_2} = \neg 5 \leq p_2 - p_3 \leq 1 \wedge (L_1 = p_1 \wedge L_2 = p_2) \wedge (t_2 \leq t_1) \wedge (L_1 \leq p_4 \wedge L_1 \leq p_5) \wedge (4 \leq p_4 - L_2 \leq 4 \wedge 0 \leq p_6 - L_2 \leq 2) \wedge (1 \leq p_3 - L_1 \leq 5 \wedge 0 \leq p_5 - L_1 \leq 2)$

Since $\varphi_{p_1} \Rightarrow L_1 \leq p_4 \wedge L_1 \leq p_5$ and $\varphi_{p_2} \Rightarrow L_2 \leq p_3 \wedge L_2 \leq p_5$, it follows that:

$\varphi_{p_1} \lor \varphi_{p_2} = \neg 5 \leq p_2 - p_3 \leq 1 \wedge (L_1 = p_1 \wedge L_2 = p_2) \wedge (t_2 \leq p_3 \wedge t_2 \leq p_6) \wedge (L_1 \leq p_4 \wedge L_1 \leq p_5) \wedge (4 \leq p_4 - L_2 \leq 4 \wedge 0 \leq p_6 - L_2 \leq 2) \wedge (1 \leq p_3 - L_1 \leq 5 \wedge 0 \leq p_5 - L_1 \leq 2)$

Formula $\varphi_{p_1} \lor \varphi_{p_2}$ is the firing condition of $t_1$ and $t_2$ from $\beta_0$, in any order. Its domain is convex (representable by a single DBM). The following theorem (Theorem 1) establishes that this result is valid for any set of transitions of $T$ not in conflict and firable from a CSCG state class. The proof of this theorem follows the same ideas as those used in the previous example to show that $\varphi_{p_1} \lor \varphi_{p_2}$ can be rewritten as a conjunction of atomic constraints.

**Theorem 1** Let $\beta = (M, \text{Dead } p, \psi_p)$ be a CSCG state class and $T_m \subseteq T$ a set of transitions firable from $\beta$ and not in conflict in $\beta$.

Then $\bigcup_{\omega \in \Omega(T_m)} \text{succ}(\beta, \omega) \neq \emptyset$ and $\bigcup_{\omega \in \Omega(T_m)} \text{succ}(\beta, \omega)$ is a state class $\beta' = (M', \text{Dead } p', \psi'_p)$ where $M' = (M - \bigcup_{t_f \in T_m} \text{Pre}(t_f)) + \bigcup_{t_f \in T_m} \text{Post}(t_f)$, $\text{Dead } p' = \text{Dead } p$ and $\psi'_p$ is a conjunction of triangular atomic constraints that can be computed as follows:

- set $\psi'_p$ to

$$\psi_p \wedge \bigwedge_{f \in [1,m]} \bigwedge_{p_i \in \text{Pre}(t_f)} \bigwedge_{p_n \in \text{Post}(t_f)} \left[ L_f \leq \downarrow \text{Is } (p_n) \leq p_f - L_f \leq \uparrow \text{Is } (p_n) \wedge \left( L_f - P_j \leq 0 \wedge \bigwedge_{k \in [1,m], p_n \in \text{Post}(t_f)} L_f - P_i^k \leq 0 \right) \right]$$

- Put $\psi'_p$ in canonical form, then eliminate variables $L_1, L_2, \ldots, L_m$ and variables associated with their input places.

- Rename each variable $p_i^f$, s.t. $p_n \in \text{Post}(t_f)$ and $f \in [1,m]$, in $p_n$. 
Proof 2 If transitions of $T_n$ are all firable from $\beta$ and not in conflict then the firing of one of them cannot disable the others. So, all sequences of $\Omega(T_n)$ are firable from $\beta$. Then: $\bigcup_{\omega \in \Omega(T_n)} \text{succ}(\beta, \omega) \neq \emptyset$. Let us first rewrite the firing condition $\varphi_p$, given in Proposition 7 of the sequence $\omega = t_1 t_2 \ldots t_m$, so as to isolate the part that is independent from the firing order. In other words, let us show that: $\varphi_p \equiv$

$$\Psi_p \land \bigwedge_{f \in [1,m]} [\bigwedge_{p \in \text{Pre}(t_f)} p_f \leq t_f \land \bigwedge_{p \in \text{Post}(t_f)} \downarrow Isp(p_n) \leq p_f - t_f \leq \uparrow Isp(p_n) \land$$

$$\bigwedge_{p_j \in (M-\text{Dead}) - \bigcup_{i \in [1,m]} \text{Pre}(t_i)} \land \bigwedge_{k \in [1,m], p_n \in \text{Post}(t_i)} L_f - p_f \leq 0 \land \bigwedge_{k \in [1,m], p_n \in Post(t_i)} L_f - p_f \leq 0$$

Consider the following sub-formula, denoted $\varphi_1$, of $\varphi_p$:

$$t_1 \leq t_2 \ldots \leq t_m \land \bigwedge_{f \in [1,m]} [\bigwedge_{p \in \text{Pre}(t_f)} p_f = t_f \land \bigwedge_{p \in \text{Post}(t_f)} \downarrow Isp(p_n) \leq p_f - t_f \leq \uparrow Isp(p_n)]$$

This formula implies that: (1) $\forall f \in [1,m], \forall l \in [f,m], L_f \leq L_l$.

(2) $\forall f \in [1,m], \forall l \in [f,m], \forall p_j \in \text{Pre}(t_j), L_f \leq L_l = p_j$.

Then: (2') $\varphi_1 \Rightarrow \bigwedge_{f \in [1,m], p_j \in \bigcup_{i \in [1,m]} \text{Pre}(t_i)} L_f - p_j \leq 0$.

(3) $\forall f \in [1,m], \forall l \in [f,m], \forall p_n \in \text{Post}(t_i), L_f \leq L_l \leq p_f^l$.

Then: (3') $\varphi_1 \Rightarrow \bigwedge_{f \in [1,m], l \in [f,m], p_n \in \text{Post}(t_i)} L_f - p_f^l \leq 0$.

Consider now the following sub-formula, denoted $\varphi_2$, of $\varphi_p$:

$$\bigwedge_{f \in [1,m], p_j \in (M-\text{Dead}) - \bigcup_{i \in [1,m]} \text{Pre}(t_i)} L_f - p_j \leq 0$$

From (2'), it follows that constraints (2) are redundant in the part $\varphi_2$ of $\varphi_p$ and then can be eliminated from the part $\varphi_2$ of $\varphi$, without altering the domain of $\varphi_p$:

$$\bigwedge_{f \in [1,m], p_j \in (M-\text{Dead}) - \bigcup_{i \in [1,m]} \text{Pre}(t_i)} L_f - p_j \leq 0$$

Let $\varphi_3$ be the following part of $\varphi$:

$$\bigwedge_{f \in [1,m], l \in [1,f], p_n \in \text{Post}(t_i)} L_f - p_f^l \leq 0$$

From (3'), it follows that constraints (3) are redundant in the part $\varphi_3$ of $\varphi_p$ and then can be added to the part $\varphi_3$ of $\varphi_p$, without altering the domain of $\varphi_p$:

$$\bigwedge_{f \in [1,m], l \in [1,m], p_n \in \text{Post}(t_i)} L_f - p_f^l \leq 0$$

Therefore, $\varphi_p \equiv$

$$\Psi_p \land L_1 \leq L_2 \ldots \leq L_m$$
Consider, in the following, safe A-TPN.

Theorem 1 is also valid for unsafe P-TPNs in the context of multiple-server semantics. The proof of this claim is similar, except that markings, presets and postsets of transitions are multisets over places. In this case, a variable is associated with each token (instead of each place). Transitions can be multi-enabled. Each enabling instance of a transition is defined as a couple composed by the name of the transition and the multiset of tokens participating in its enabling. Its firing delay depends on time constraints of its tokens. A variable is associated with each enabling instance of the same transition. In the next section, we will extend the result established in Theorem 1 to the A-TPN model.

3 A-Time Petri Nets

The A-TPN model is the most powerful model in the class of \{P,T,A\}-TPN [8]. Like in P-TPN, A-TPN uses the notion of availability intervals of tokens but each token of a place \( p \) has an availability interval per output arc of \( p \), whereas, in P-TPN, each token has only one availability interval. As for P-TPN, we consider, in the following, safe A-TPN.

Formally, A-TPN is a tuple \((P,T,Pre,Post,M_0,Isa)\) where:

1. \( P, T, \text{Pre, Post} \) and \( M_0 \) are defined as for P-TPN,

2. Let \( IE = \{(p_i,t_j) \in P \times T | p_i \in Pre(t_j)\} \) be the set of input arcs of all transitions. \( Isa : IE \to \mathbb{Q}^+ \times (\mathbb{Q}^+ \cup \{\infty\}) \) is the static availability interval function. \( Isa(p_i,t_j) \) specifies the lower \( \downarrow Isa(p_i,t_j) \) and the upper \( \uparrow Isa(p_i,t_j) \) bounds of the static availability interval of tokens of \( p_i \) for \( t_j \).

Since, in A-TPN, intervals are associated with arcs connecting places to transitions, the notion of dead tokens of the P-TPN model is replaced by dead arcs. If a place \( p_i \) is marked and connected to a transition \( t_j \), the arc \((p_i, t_j)\) will die if the residence time of the token of \( p_i \) overpasses the availability interval of the arc \((p_i, t_j)\). To detect dead arcs, we use the special transition \( Err \), as for the P-TPN model.

Let \( EE(M) = \{(p_i,t_j) \in M \times T | p_i \in Pre(t_j)\} \) be the set of enabled arcs in \( M \). The A-TPN state is defined as a triplet \((M, Deada, Ia)\), where \( M \subseteq P \) is a marking, \( Deada \subseteq EE(M) \) is the set of dead arcs in \( EE(M) \) and \( Ia \) is the interval function \((Ia : EE(M) \to \mathbb{Q}^+ \times (\mathbb{Q}^+ \cup \{\infty\}))\) which associates with each enabled and non dead arc an availability interval. The initial state of the A-TPN model is
\(s_0 = (M_0, \text{Deada}_0, \text{Ia}_0)\) where \(\text{Deada}_0 = \emptyset, \text{Ia}_0(p_i, t_j) = \text{Ia}(p_i, t_j)\), for all \((p_i, t_j) \in EE(M_0)\). When a token is created in place \(p_i\), the availability interval of each output arc \((p_i, t_j)\) is set to its static interval \(\text{Ia}(p_i, t_j)\) and then decreases, synchronously with time, until the token within \(p_i\) is consumed or the arc dies. A transition \(t_j\) can fire iff all its input arcs are not dead and have reached their availability intervals, i.e., the lower bounds of the intervals of its input arcs have reached 0. But, it must fire, without any additional delay, if the upper bound of, at least, one of its input arcs has reached 0. The firing of a transition takes no time.

The A-TPN state space is the timed transition system \((S, \rightarrow, s_0)\), where \(s_0\) is the initial state of the A-TPN and \(S = \{s \mid s_0 \rightarrow s\}\) is the set of reachable states of the model, \(\rightarrow\) being the reflexive and transitive closure of the relation \(\rightarrow\) defined as follows.

Let \(s = (M, \text{Deada}, \text{Ia}), s' = (M', \text{Deada}', \text{Ia}')\) be two A-TPN states, \(d \in \mathbb{R}^+, t_f \in T,\)

- \(s \xrightarrow{d} s'\), iff \(\forall (p_i, t_j) \in EE(M) - \text{Deada}, \text{d} \leq \uparrow \text{Ia}(p_i, t_j), M' = M, \text{Deada}' = \text{Deada}\) and \(\forall (p_k, t_i) \in EE(M') - \text{Deada}', \text{Ia}'(p_k, t_i) = \lceil \text{Max}(\downarrow \text{Ia}(p_k, t_i) - d, 0), \uparrow \text{Ia}(p_k, t_i) - d\rceil\). The time progression is allowed while we do not overpass intervals of all non dead arcs of \(EE(M')\).

- \(s \xrightarrow{t_f} s'\) iff state \(s'\) is immediately reachable from state \(s\) by firing transition \(t_f\), i.e., \(\text{Pre}(t_f) \times \{t_f\} \subseteq EE(M) - \text{Deada}, \forall p_i \in \text{Pre}(t_f), \downarrow \text{Ia}(p_i, t_f) = 0, M' = (M - \text{Pre}(t_f)) \cup \text{Post}(t_f), \text{Deada}' = \text{Deada} - (\text{Pre}(t_f) \times T), \text{and} \forall (p_k, t_i) \in EE(M') - \text{Deada}', \text{Ia}'(p_k, t_i) = \text{Ia}(p_k, t_i)\), if \(p_k \in \text{Post}(t_f)\) and \(\text{Ia}'(p_k, t_i) = \text{Ia}(p_k, t_i)\) otherwise. It means that all input arcs of \(t_f\) are enabled, not dead and have reached their availability intervals. The firing of \(t_f\) consumes tokens of its input places and produces tokens in its output places (one token per output place). The consumed tokens and their output arcs are removed. The produced tokens are added to the marking. The availability intervals of their output arcs are set to their static availability intervals.

- \(s \xrightarrow{\text{Err}} s'\) iff state \(s'\) is immediately reachable from state \(s\) by firing transition \(\text{Err}\). Transition \(\text{Err}\) is immediately firable from \(s\) if there is no transition of \(T\) firable from \(s\) and there is at least an arc in \(EE(M) - \text{Deada}\) s.t. \(\text{the upper bound of its interval has reached 0}\) i.e., \(\forall t_k \in T, s.t. \text{Pre}(t_k) \times \{t_k\} \subseteq EE(M) - \text{Deada}, \exists p_j \in \text{Pre}(t_k), \downarrow \text{Ia}(p_j, t_k) > 0, (\exists (p_i, t_j) \in EE(M) - \text{Deada}, \uparrow \text{Ia}(p_i, t_j) = 0), M' = M, \text{Deada}' = \text{Deada} \cup \{(p_j, t_k) \in EE(M) - \text{Deada}| \downarrow \text{Ia}(p_j, t_k)) = 0\}, \text{and} \forall (p_i, t_j) \in EE(M') - \text{Deada}', \text{Ia}'(p_i, t_j) = \text{Ia}(p_i, t_j)\).

### 3.1 The CSCG of the A-TPN

The definition of the CSCG of the P-TPN is extended to the A-TPN by replacing the notion of dead tokens by dead arcs and constraints on availability of tokens by those of arcs. The CSCG state class of A-TPN is defined as a triplet \(\gamma = (M, \text{Deada}, \phi_a)\) where \(M \subseteq P\) is a marking, \(\text{Deada} \subseteq EE(M)\) is the set of dead arcs in \(EE(M)\) and \(\phi_a\) is a conjunction of triangular atomic constraints over variables associated with non dead arcs of \(EE(M)\). Each arc \((p_i, t_j)\) of \((EE(M) - \text{Deada})\) has a variable, denoted \(pt_{ij}\) in \(\phi_a\), representing its availability interval.

The initial CSCG state class is: \(\gamma_0 = (M_0, \text{Deada}_0, \phi_{a0})\) where \(M_0 \subseteq P\) is the initial marking, \(\text{Deada}_0 = \emptyset\) and \(\phi_{a0} = \bigwedge_{(p_i, t_j) \in EE(M_0), (p_k, t_i) \in EE(M_0)} pt_{ij} - L_{ij} \leq \uparrow \text{Ia}(p_i, t_j) - \downarrow \text{Ia}(p_k, t_i)\).

Successor state classes are computed using the following firing rule: Let \(\gamma = (M, \text{Deada}, \phi_a)\) be a state class and \(t_f\) a transition of \(T\). The state class \(\gamma\) has a successor by \(t_f\) (i.e., \(\text{succ}(\gamma, t_f) \neq \emptyset\)) iff \(\text{Pre}(t_f) \times \{t_f\} \subseteq EE(M) - \text{Deada}\) and the following formula is consistent:

\[
\phi_a \land \left( \bigwedge_{p_i \in \text{Pre}(t_f), (p_j, t_k) \in EE(M) - \text{Deada}} pt_{ij} \leq pt_{jk} \right)
\]
This firing condition means that $t_f$ is enabled in $M$, its input arcs are not dead, and there is a state s.t. the input arcs of $t_f$ will reach their intervals before overpassing intervals of all non dead arcs in $EE(M)$. If $\text{succ}(\gamma, t_f) \neq \emptyset$ then $\text{succ}(\gamma, t_f) = (M', \text{Deada}', \psi'_a)$ is computed as follows:

1. $M' = (M - \text{Pre}(t_f)) \cup \text{Post}(t_f)$;
2. $\text{Deada}' = \text{Deada} - (\text{Pre}(t_f) \times T)$
3. Set $\psi_a$ to $\psi_a \land (\bigwedge_{p_j \in \text{Pre}(t_f)} \downarrow \text{Isa}(p_n, t_t) \leq \frac{\text{pt}_f}{\text{pt}_p} \leq \downarrow \text{Isa}(p_n, t_t) \land \bigwedge_{p_j \in \text{Post}(t_f), t_i \in p_n^a} \downarrow \text{Isa}(p_n, t_t) \leq \frac{\text{pt}_f}{\text{pt}_p} \leq \downarrow \text{Isa}(p_n, t_t) \land \bigwedge_{(p_j, t_i) \in (EE(M) - \text{Deada}) - (\bigcup_{t_f \in T_m} \text{Pre}(t_f) \times T)} \frac{\text{tf} - \text{pt}_{ik}}{\text{pt}_p} \leq 0 \land \bigwedge_{k \in [1, m], p_n \in \text{Post}(t_k), t_i \in p_n^a} \frac{\text{tf} - \text{pt}_{ik}}{\text{pt}_p} \leq 0)$
4. Replace variables $\text{pt}_f$ associated with input arcs of $t_f$ by $\psi_f$.
5. Add constraints $\bigwedge_{p_n \in \text{Post}(t_f), t_i \in p_n^a} \downarrow \text{Isa}(p_n, t_t) \leq \frac{\text{pt}_f}{\text{pt}_p} \leq \downarrow \text{Isa}(p_n, t_t)$.
6. Put $\psi'_a$ in canonical form before eliminating variable $\psi_f$.

If $t_f$ is firable then its firing consumes its input tokens and creates tokens in its output places (one token per output place). The consumed tokens and their output arcs are eliminated. Step 3) isolates states of $\gamma$ from which $t_f$ is firable (i.e., states where input arcs of $t_f$ reach their availability interval before overpassing the availability intervals of all non dead enabled arcs). This step implies that for all $p_i, p_j \in \text{Pre}(t_f), \text{pt}_f = \text{pt}_j$. Step 4) replaces all these equal variables by $\psi_f$. Steps 5) adds the time constraints of the created tokens. Step 6) puts $\psi'_a$ in canonical form before eliminating variable $\psi_f$.

### 3.2 Interleaving in the CSCG of A-TPN

The following theorem extends to A-TPN, the result established in Theorem 1.

**Theorem 2** Let $\gamma = (M, \text{Deada}, \psi_a)$ be a CSCG state class and $T_m \subseteq T$ a set of transitions firable from $\gamma$ and not in conflict in $\gamma$. Then $\bigcup_{\omega \in \Omega(T_m)} \text{succ}(\gamma, \omega) \neq \emptyset$ and $\bigcup_{\omega \in \Omega(T_m)} \text{succ}(\gamma, \omega)$ is a state class $\gamma' = (M', \text{Deada}', \psi'_a)$ where $M' = (M - \bigcup_{t_f \in T_m} \text{Pre}(t_f)) \cup \bigcup_{t_f \in T_m} \text{Post}(t_f)$, $\text{Deada}' = \text{Deada} - (\bigcup_{t_f \in T_m} \text{Pre}(t_f) \times T)$ and $\psi'_a$ is a conjunction of triangular atomic constraints that can be computed as follows:

- Set $\psi_a$ to

$$\psi_a \land \bigwedge_{f \in [1, m]} \bigwedge_{p_j \in \text{Pre}(t_f)} \downarrow \text{Isa}(p_n, t_t) \leq \frac{\text{pt}_f}{\text{pt}_p} \leq \downarrow \text{Isa}(p_n, t_t) \land \bigwedge_{(p_j, t_i) \in (EE(M) - \text{Deada}) - \bigcup_{t_f \in T_m} \text{Pre}(t_f) \times T} \frac{\text{tf} - \text{pt}_{ik}}{\text{pt}_p} \leq 0 \land \bigwedge_{k \in [1, m], p_n \in \text{Post}(t_k), t_i \in p_n^a} \frac{\text{tf} - \text{pt}_{ik}}{\text{pt}_p} \leq 0$$

- Put $\psi'_a$ in canonical form, then eliminate variables $L_1, L_2, \ldots, L_m$ and variables associated with their input places.
- Rename each variable $\text{pt}_f$, s.t. $p_n \in \text{Post}(t_f), t_i \in p_n^a$ and $f \in [1, m]$, in $\text{pt}_f$.

**Proof 3** We first extend the firing condition of a sequence $\omega = t_1t_2\ldots t_n$ of $\Omega(T_m)$ given in Proposition 7 to the case of A-TPN. $\omega$ is firable from $\gamma$ (i.e., $\text{succ}(\beta, \omega)$) iff the following formula, denoted $\varphi_a$ is consistent:

$$\psi_a \land L_1 \leq L_2 \leq \ldots \leq L_n$$
The firing condition of the sequence $t_1t_2...t_m$ from $\gamma$ adds to $\psi_a$ for each transition $t_f$ of the sequence, a variable, denoted $\mathcal{L}_f$, representing its firing delay and five blocks of constraints. The first block fixes the firing order of transitions of $T_m$. The second block means that the residence delays of arcs used by each transition $t_f$ must be equal to $\mathcal{L}_f$. The third and the fourth blocks mean that the firing delay $\mathcal{L}_f$ is less or equal to the residence delays of all enabled and non dead arcs present when $t_f$ is fired (i.e., $(p_j,t_k) \in (EE(M)−Deada)−( \bigcup_{l \in [1..f]} Pre(t_l) \times T)$ and $(p_n,t_l)$ s.t. $p_n \in \bigcup_{l \in [1..f]} Post(t_l)$ and $t_l \in p_n^n$). The fifth block of constraints specifies the residence delays of arcs enabled by $t_f$ (i.e., $(p_n,t_l)$ s.t. $p_n \in Post(t_f)$ and $t_l \in p_n^n$).

The rest of the proof follows the same steps as the proof of Theorem 1. In other words, let us show that $\Phi_a \equiv \psi_a \land \mathcal{L}_1 \leq \mathcal{L}_2 \leq ... \leq \mathcal{L}_m \land$

\[
\bigwedge_{f \in [1..m]} \bigwedge_{p_f \in Pre(t_f)} \mathcal{P}_{f} = \mathcal{L}_f \land \bigwedge_{p_n \in Post(t_f)} \mathcal{L}_f - \mathcal{P}_{nk} \leq 0 \land \bigwedge_{(p_j,t_k) \in (EE(M)−Deada)−\bigcup_{l \in [1..f]} Pre(t_l) \times T} \mathcal{L}_f - \mathcal{L}_{jk} \leq 0 \land \bigwedge_{k \in [1..f]} \bigwedge_{p_n \in Post(t_k)} \mathcal{L}_f - \mathcal{L}_{nk} \leq 0 \]

Consider the following sub-formula, denoted $\Phi_1$, of $\Phi_a$:

\[
\mathcal{L}_1 \leq \mathcal{L}_2 \leq ... \leq \mathcal{L}_m \land \bigwedge_{f \in [1..m]} \bigwedge_{p_f \in Pre(t_f)} \mathcal{P}_{f} = \mathcal{L}_f \land \bigwedge_{p_n \in Post(t_f)} \mathcal{L}_f - \mathcal{P}_{nk} \leq 0 \land \bigwedge_{(p_j,t_k) \in (EE(M)−Deada)−\bigcup_{l \in [1..f]} Pre(t_l) \times T} \mathcal{L}_f - \mathcal{L}_{jk} \leq 0 \land \bigwedge_{k \in [1..f]} \bigwedge_{p_n \in Post(t_k)} \mathcal{L}_f - \mathcal{L}_{nk} \leq 0 \]

This formula implies that:

(1) $\forall f \in [1..m], \forall k \in [f..m], \mathcal{L}_f \leq \mathcal{L}_k$.

(2) $\forall f \in [1..m], \forall k \in [f..m], \forall p_j \in Pre(t_k), \mathcal{L}_f \leq \mathcal{L}_k = \mathcal{P}_{jk}$.

Then: (2') $\Phi_1 \Rightarrow \bigwedge_{f \in [1..m], k \in [f..m], p_f \in Pre(t_k)} \mathcal{L}_f - \mathcal{L}_{jk} \leq 0$.

(3) $\forall f \in [1..m], \forall k \in [f..m], \forall p_n \in Post(t_k), \forall t_l \in p_n^n, \mathcal{L}_f \leq \mathcal{L}_k = \mathcal{P}_{nk}$.

Then: (3') $\Phi_1 \Rightarrow \bigwedge_{f \in [1..m], k \in [f..m], p_n \in Post(t_k), t_l \in p_n^n} \mathcal{L}_f - \mathcal{L}_{nl} \leq 0$.

Consider the following sub-formula, denoted $\Phi_2$, of $\Phi_a$:

\[
\bigwedge_{f \in [1..m], (p_j,t_k) \in (EE(M)−Deada)−\bigcup_{l \in [1..f]} Pre(t_l) \times T} \mathcal{L}_f - \mathcal{P}_{jk} \leq 0
\]

From (2'), it follows that constraints (2) are redundant in the partition $\Phi_2$ of $\Phi_a$ and then can be eliminated from the partition $\Phi_2$ of $\Phi_a$, without altering the domain of $\Phi_a$:

\[
\bigwedge_{f \in [1..m], (p_j,t_k) \in (EE(M)−Deada)−\bigcup_{l \in [1..f]} Pre(t_l) \times T} \mathcal{L}_f - \mathcal{P}_{jk} \leq 0
\]

Let $\Phi_3$ be the following part of $\Phi_a$:

\[
\bigwedge_{f \in [1..m], k \in [1..f], p_n \in Post(t_k), t_l \in p_n^n} \mathcal{L}_f - \mathcal{L}_{nl} \leq 0
\]
From (3'), it follows that constraints (3) are redundant in the part $\varphi_1$ of $\varphi_a$ and then can be added to the part $\varphi_3$ of $\varphi_a$, without altering the domain of $\varphi_a$:

$$\bigwedge_{f \in [1,m], k \in [1,m], p_n \in \text{Post}(t_k), t_i \in P_n^k} L_f - p_{i+n}^k \leq 0$$

Therefore, $\varphi_a \equiv \psi_a \land L_1 \leq L_2 \leq \ldots \leq L_n \land$

$$\bigwedge_{f \in [1,m]} \left[ \bigwedge_{p_i \in \text{Pre}(t_f)} p_f = L_f \land \bigwedge_{p_n \in \text{Post}(t_k), t_k \in P_n^k} \downarrow \text{Isa}(p_n, t_k) \leq p_{f+}^k - L_f \leq \uparrow \text{Isa}(p_n, t_k) \land \bigwedge_{(p_j, a_i) \in (EE(M)-Deada)- \bigcup_{i \in [1,m]} \text{Pre}(t_i) \times T} L_f - p_{j+k} \leq 0 \land \bigwedge_{k \in [1,m], p_n \in \text{Post}(t_k), a_0 \in P_n^k} L_f - p_{i+n}^k \leq 0 \right]$$

The firing condition of transitions of $T_m$ in any order, denoted $\psi'_a$, is obtained by eliminating the part fixing the firing order. To obtain the formula of $\psi'_a$, it suffices to put $\psi_a$ in canonical form and then eliminate variables associated with transitions of $T_m$ and their input places.

The extension of this result to unsafe A-TPN is straightforward by considering multisets of tokens, multisets of enabled arcs, and associating a variable with each instance of multiple enabled arcs. Each enabled transition is defined by the name of the transition and a set of enabled arcs.

Using the translation into A-TPN of the P-TPN shown in Figure 2.a), we prove that the union of the SCG state classes of the A-TPN reached by different interleavings of the same set of transitions is not necessarily convex. Indeed, its initial SCG state class $(p_1 + p_2, \emptyset, 1 \leq pt_{1} \leq 3 \land 2 \leq pt_{2} \leq 4)$, sequences $t_1t_2$ and $t_2t_1$ lead respectively to the SCG state classes: $(p_3 + p_4, \emptyset, 0 \leq pt_{1} \leq 1 \land pt_{3} \leq 2 \land pt_{4} \leq 4) = 2 \land -2 \leq pt_{33} - pt_{44} \leq -1)$ and $(p_3 + p_4, \emptyset, pt_{33} = 1 \land 1 \leq pt_{44} \leq 2 \land -1 \leq pt_{33} - pt_{44} \leq 0).$ The union of their domains is not convex.

4 Conclusion

In this paper, we have considered the P-TPN and A-TPN models, their SCG and CSCG. We have investigated the convexity of the union of state classes reached by different interleavings of the same set of transitions. We have shown that this union is not convex in the SCG but is convex in the CSCG. This result allows to use the reachability analysis approach proposed in [2], which reduces the redundancy caused by the interleaving semantics.

This result is however not valid for the T-TPN [3], in spite of the fact that A-TPN is the most powerful model. This could be explained by the fact that the firing interval of a transition refers to the instant when it becomes enabled in T-TPN, whereas, in $\{P_A\}$-TPN, it is equal to the intersection of intervals of all its input tokens/arc. In T-TPN, the firing interval can be related to the last transition of a sequence and then dependent of the firing order. For example, consider the net shown in Figure 2.b) and suppose that the net becomes enabled in T-TPN, whereas, in the model. This could be explained by the fact that the firing interval of a transition refers to the instant when it is related to $t_1$ in T2t1. The union of the CSCG state classes reached by $t_1t_2$ and $t_2t_1$ from the initial state class is: $(p_3 + p_4 + p_5 + p_6, (-8 \leq l_1 - l_4 \leq 1 \land -6 \leq l_3 - l_5 \leq 1 \land 2 \leq l_4 - l_5 \leq 4) \lor (-3 \leq l_3 - l_4 \leq 2 \land -1 \leq l_3 - l_5 \leq 5 \land 1 \leq l_4 - l_5 \leq 4))$. Its domain is not convex.

The P-TPN is translated into A-TPN by replacing the static residence interval function $Isa$ by $Isa$ defined by: $\forall p_i \in P, t_j \in P^j_i, Isa(p_i, t_j) = Isa(p_i)$.
Therefore, A-TPN is more powerful than T-TPN and also more suitable for abstractions by convex-
union. However, the translation of T-TPN into A-TPN is not easy and needs to add several places
and transitions [8], which may offset the benefits of abstractions by convex-union. The choice of the
appropriate \{P,T,A\}-TPN model for a given problem should be a good compromise between the easiest
of modeling the problem and the verification complexity.

As immediate perspective, we will use the results established here and in [6] to investigate the exten-
sion, to \{P,T,A\}-TPN, of the reachability approach proposed in [12] for a variant of safe P-TPN. In this
variant, there are two kinds of places (behaviour and constraint places) and each transition can have at
most one behaviour place in its preset. A transition is fireable, if the age of its behaviour place reaches its
static residence interval. It must be fired before overpassing this interval, unless it is disabled.

References

[1] G. Behrmann, P. Bouyer, K. G. Larsen, and R. Pelánek Lower and upper bounds in zone-based abstractions
of timed automata, International Journal on Software Tools for Technology Transfer Volume 8(3), 2006.
[2] R. Ben Salah, M. Bozga and O. Maler, On Interleaving in Timed Automata, CONCUR’06, 465-476, volume
4137 of LNCS, 2006.
[3] Bengtsson, J.: Clocks, DBMs and States in Timed Systems, PhD thesis, Dept. of Information Technology,
Uppsala University, 2002.
[4] B. Berthomieu and F. Vernadat, State class constructions for branching analysis of Time Petri nets, volume
2619 of LNCS, 2003.
[5] H. Boucheneb, G. Gardey, and O. (H.) Roux. TCTL model checking of time Petri nets, Journal of Logic and
Computation, 19(6):1509-1540, December 2009.
[6] H. Boucheneb and K. Barkaoui. Covering steps graphs of time Petri nets In Proc. of the 10th International
Workshop on Verification of Infinite-State Systems (INFINITY), 2008.
[7] H. Boucheneb and H. Rakkay, A more efficient time Petri net state space abstraction useful to model checking
timed linear properties In journal of Fundamenta Informaticae, volume 88, number 4, pp 469-495, 2008.
[8] M. Boyer and O. H. Roux,On the compared expressiveness of arc, place and transition time Petri Nets,
journal Fundamenta Informaticae, vol. 88, n° 3, pages 225-249, 2008.
[9] W. Khansa, J.-P Denat and S. Collart-Dutilleul, P-Time Petri Nets for manufacturing systems, International
Workshop on Discrete Event Systems, WODES’96, pp 94-102, 1996.
[10] R. Hadjidj and H. Boucheneb, Improving state class constructions for CTL* model checking of Time Petri
Nets, International Journal on Software Tools Technology Transfer (STTT), volume 10, number 2, pp 167-
184, 2008.
[11] P. M. Merlin, A study of the recoverability of computing systems., Department of Information and Computer
Science, University of California, Irvine CA, 1974.
[12] C. J. Myers, T. G. Rokicki, T. H.-Y. Meng, POSET timing and its application to the timed circuicts, in IEEE
Transactions on CAD, 18(6), 1999.
[13] B. Walter, Timed net for modeling and analysing protocols with time, IFIP Conference on Protocol Specifica-
tion Testing and Verification, 1983.