A state class based controller synthesis approach for Time Petri Nets

Loriane Leclercq, Didier Lime and Olivier H. Roux

June 8, 2023

Introduction

- Controller synthesis for Time Petri Nets

Introduction

- Controller synthesis for Time Petri Nets
- Timed game for reachability

Introduction

- Controller synthesis for Time Petri Nets
- Timed game for reachability
- Explicit firing dates semantics

Introduction

- Controller synthesis for Time Petri Nets
- Timed game for reachability
- Explicit firing dates semantics
- State classes

Example

The Tortoise and the Hare

For the race:

- the tortoise needs 5 to 6 minutes to arrive
- the hare needs only 3 to 4 minutes to arrive but,

Example

The Tortoise and the Hare

For the race:

- the tortoise needs 5 to 6 minutes to arrive
- the hare needs only 3 to 4 minutes to arrive but,
- the hare stays grazing between 0 and 4 minutes before starting the race

Example

The Tortoise and the Hare

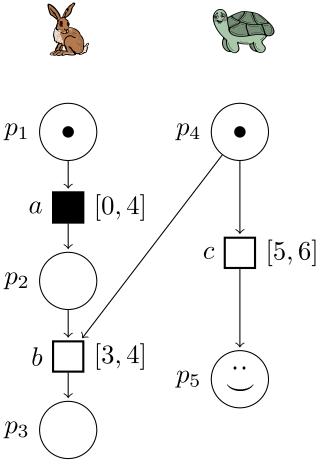
For the race:

- the tortoise needs 5 to 6 minutes to arrive
- the hare needs only 3 to 4 minutes to arrive but,
- the hare stays grazing between 0 and 4 minutes before starting the race

How long does the hare have to wait for the tortoise to win?

Time Petri Net (TPN)

Example



Time Petri Net (TPN)

Definition

A *time Petri net* (TPN) is a tuple $\mathcal{N} = (P, T, F, I_s)$ where:

- P is a finite non-empty set of *places*,
- T is a finite set of *transitions* such that $T \cap P = \emptyset$,
- $F : (P \times T) \cup (T \times P)$ is the *flow function*,
- $I_s : T \to \mathcal{I}(\mathbb{N})$ is the *static firing interval* function,

Semantics

States of the TPN: (m, θ) with

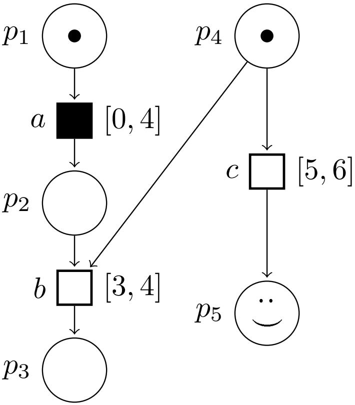
- $m \subseteq P$ a marking and
- θ the firing dates for every enabled transition in m

Firing dates are choosen when the transition become enable instead of at the firing time (moment of firing).

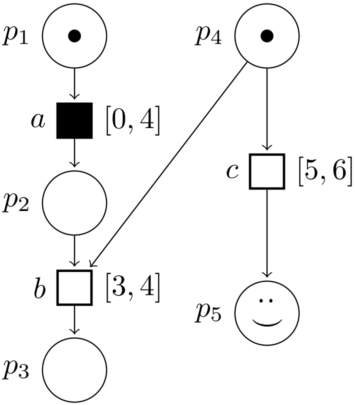
The transition relation $\rightarrow \subseteq S \times \Sigma \times S$:

- either $(m, \theta) \xrightarrow{t_f} (m', \theta')$ for the firing of a transition t_f
- or $(m, \theta) \xrightarrow{d} (m, \theta')$ for the time delay transition

Example

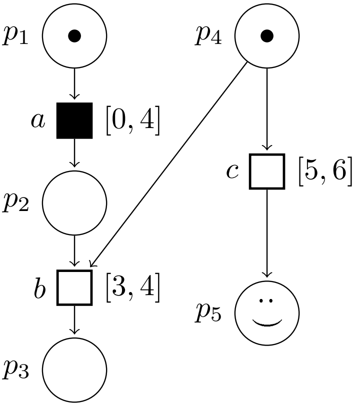


Example



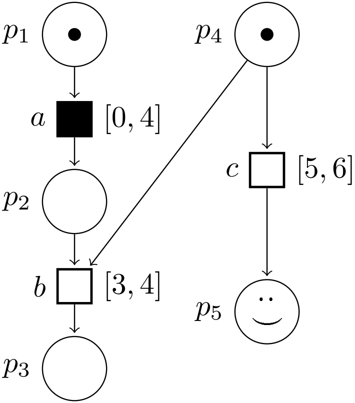
$(1, \perp, 5)$

Example



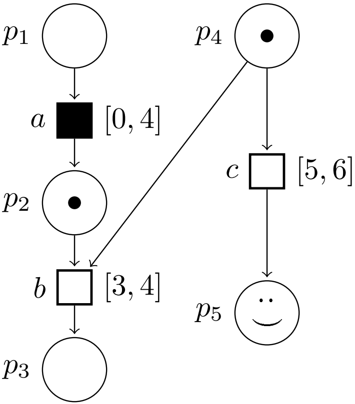
$$\begin{tikzcd}
(1, \perp, 5) \arrow[d, "0.3"'] \\
(0.7, \perp, 4.7)
\end{tikzcd}$$

Example



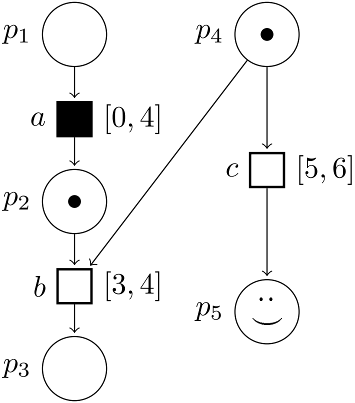
$$
\begin{array}{c}
(1, \perp, 5) \\
\big\downarrow 0.3 \\
(0.7, \perp, 4.7) \\
\big\downarrow 0.7 \\
(0, \perp, 4)
\end{array}
$$

Example



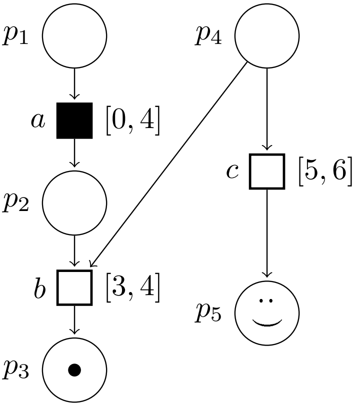
$$
\begin{array}{c}
(1, \bot, 5) \\
\downarrow {\scriptstyle 0.3} \\
(0.7, \bot, 4.7) \\
\downarrow {\scriptstyle 0.7} \\
(0, \bot, 4) \\
\downarrow {\scriptstyle a} \\
(\bot, 3, 4)
\end{array}
$$

Example



$$\begin{tikzcd}
(1,\perp,5) \arrow[d, "0.3"'] \\
(0.7,\perp,4.7) \arrow[d, "0.7"'] \\
(0,\perp,4) \arrow[d, "a"'] \\
(\perp,3,4) \arrow[d, "3"'] \\
(\perp,0,1)
\end{tikzcd}$$

Example



$$
\begin{array}{c}
(1, \perp, 5) \\
\downarrow 0.3 \\
(0.7, \perp, 4.7) \\
\downarrow 0.7 \\
(0, \perp, 4) \\
\downarrow a \\
(\perp, 3, 4) \\
\downarrow 3 \\
(\perp, 0, 1) \\
\downarrow b \\
(\perp, \perp, \perp)
\end{array}
$$

Reachability game

Definition

$\mathcal{R} = (\mathcal{A}, \mathsf{Goal})$ with:

- an arena $\mathcal{A} = (S, \rightarrow, Pl, (Mov_i)_{i \in Pl}, Trans)$
  - Pl_c : controllable transitions
  - Pl_u : uncontrollable transitions
- a set of target states $\mathsf{Goal} \in S$ that Pl_c wants to reach and Pl_u wants to avoid.

Reachability game

Definition

$\mathcal{R} = (\mathcal{A}, \mathsf{Goal})$ with:

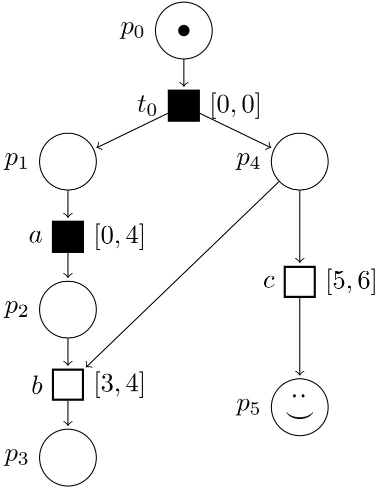
- an arena $\mathcal{A} = (S, \rightarrow, Pl, (Mov_i)_{i \in Pl}, Trans)$
  - Pl_c : controllable transitions
  - Pl_u : uncontrollable transitions
- a set of target states $\mathsf{Goal} \in S$ that Pl_c wants to reach and Pl_u wants to avoid.

Turn :

1. Pl_c chooses $t_c \in T_c$
2. Pl_u chooses $t_u \in T_u \cup \{t_c\}$
3. Both player chooses firing times for their newly enabled transitions, controllable or uncontrollable.

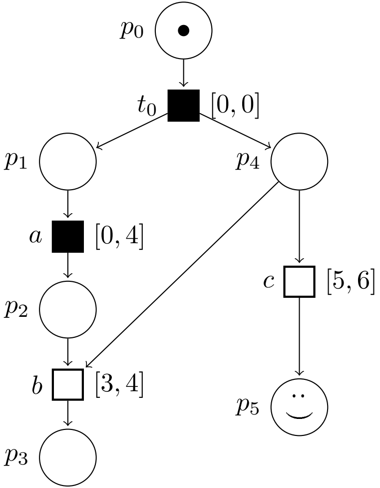
Reachability game

Example



Reachability game

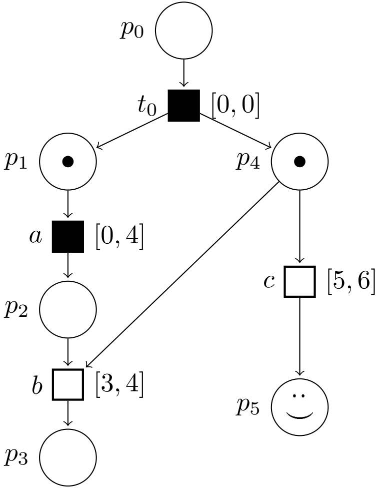
Example



$$s_0 = (\{p_0\}, \theta(t_0) = 0)$$

Reachability game

Example

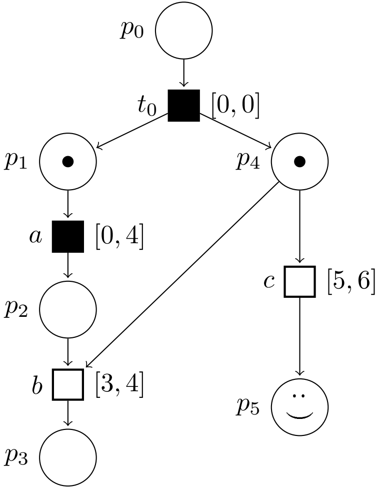


$$s_0 = (\{p_0\}, \theta(t_0) = 0)$$

$$t_c = t_0, t_u = t_c, \theta(a) = 2, \theta(c) = 6$$

Reachability game

Example

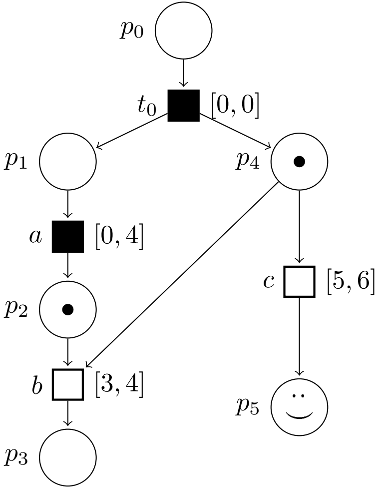


$$s_0 = (\{p_0\}, \theta(t_0) = 0)$$

$$s_1 = (\{p_1, p_4\}, \theta(a) = 2, \theta(c) = 6)$$

Reachability game

Example



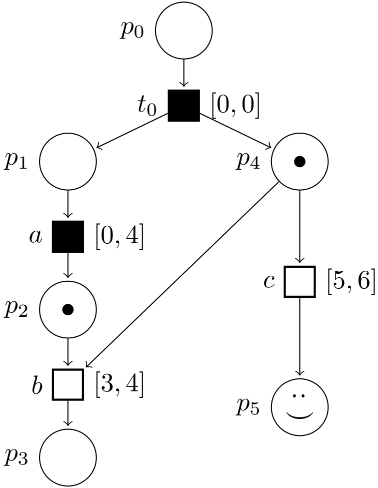
$$s_0 = (\{p_0\}, \theta(t_0) = 0)$$

$$s_1 = (\{p_1, p_4\}, \theta(a) = 2, \theta(c) = 6)$$

$$t_c = a, t_u = t_c, \theta(b) = 4$$

Reachability game

Example



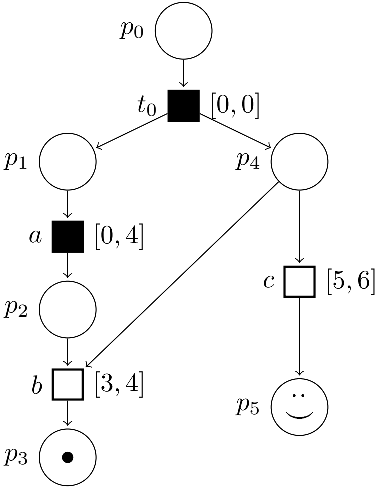
$s_0 = (\{p_0\}, \theta(t_0) = 0)$

$s_1 = (\{p_1, p_4\}, \theta(a) = 2, \theta(c) = 6)$

$s_2 = (\{p_2, p_4\}, \theta(b) = 4, \theta(c) = 4)$

Reachability game

Example



$$s_0 = (\{p_0\}, \theta(t_0) = 0)$$

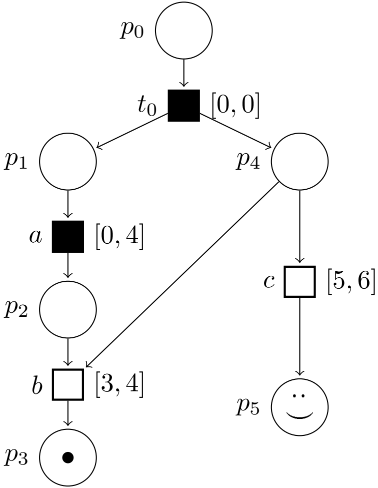
$$s_1 = (\{p_1, p_4\}, \theta(a) = 2, \theta(c) = 6)$$

$$s_2 = (\{p_2, p_4\}, \theta(b) = 4, \theta(c) = 4)$$

$$t_u = b$$

Reachability game

Example



$$s_0 = (\{p_0\}, \theta(t_0) = 0)$$

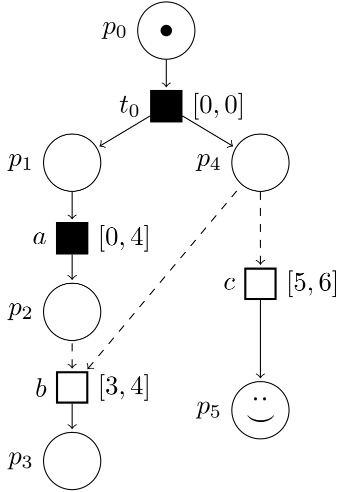
$$s_1 = (\{p_1, p_4\}, \theta(a) = 2, \theta(c) = 6)$$

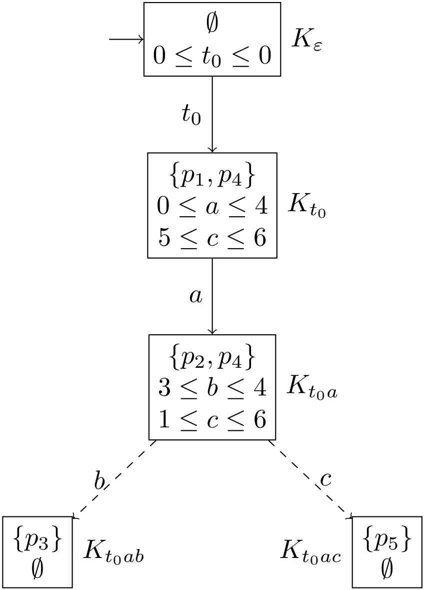
$$s_2 = (\{p_2, p_4\}, \theta(b) = 4, \theta(c) = 4)$$

$$t_u = b$$

$$s_3 = (\{p_3\}, \bot)$$

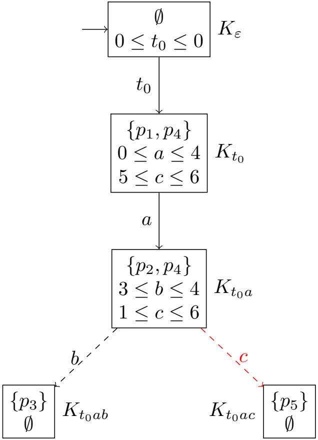
State-Class Graph (SCG)[1]





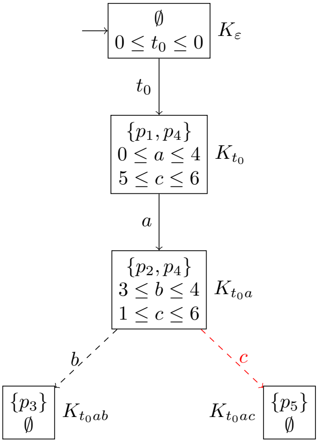
[1] Berthomieu and Menasche, “An Enumerative Approach For Analyzing Time Petri Nets”.

Subset of winning states



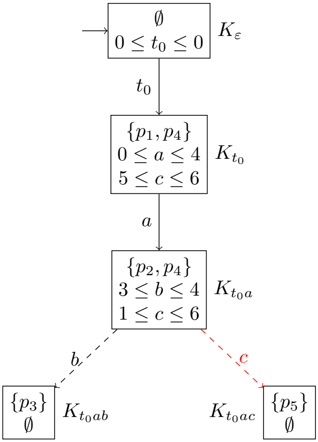
$$C = \begin{pmatrix} \{p_2, p_4\} \\ 3 \leq b \leq 4 \\ 1 \leq c \leq 6 \\ c < b \end{pmatrix}$$

Subset of winning states



$$C = \begin{pmatrix} \{p_2, p_4\} \\ 3 \le b \le 4 \\ 1 \le c \le 6 \\ c < b \end{pmatrix} = \begin{pmatrix} \{p_2, p_4\} \\ 3 \le b \le 4 \\ 1 \le c < 3 \end{pmatrix}$$

Subset of winning states



$$C' = \begin{pmatrix} \{p_1, p_4\} \\ 0 \leq a \leq 4 \\ 5 \leq c \leq 6 \\ \color{red}{1 \leq c - a < 3} \end{pmatrix} = \begin{pmatrix} \{p_1, p_4\} \\ \color{red}{2 <} a \leq 4 \\ 5 \leq c \leq 6 \\ 1 \leq c - a < 3 \end{pmatrix}$$

$$C = \begin{pmatrix} \{p_2, p_4\} \\ 3 \leq b \leq 4 \\ 1 \leq c \leq 6 \\ c < b \end{pmatrix} = \begin{pmatrix} \{p_2, p_4\} \\ 3 \leq b \leq 4 \\ \color{red}{1 \leq c < 3} \end{pmatrix}$$

Future work

- reachability with parametric firing-time constraints: using an efficient linear programming algorithm when the dimension is small and/or fixed
- explicit firing dates semantics expressivity
- applications to concrete problems

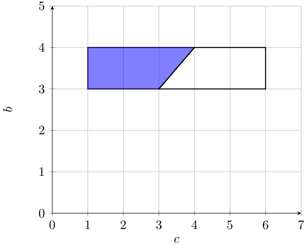
Références

Berthomieu, Bernard and Miguel Menasche. “An Enumerative Approach For Analyzing Time Petri Nets”. In: *Proceedings IFIP*. Elsevier Science Publishers, 1983, pp. 41–46.

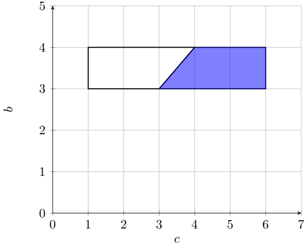
Annexes

Projection of non trivial constraints

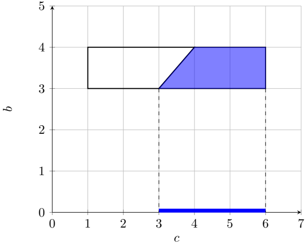
Universal projection : $\pi^{\forall}_{\{c\}}(K_{t_0 a}, C)$



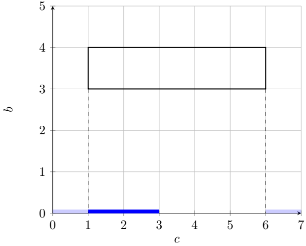
(a) C



(b) $K_{t_0 a} \cap \overline{C}$



(c) $\pi^{\exists}_{\{c\}}(K_{t_0 a} \cap \overline{C})$



(d) $\pi^{\exists}_{\{c\}}(K_{t_0 a}) \cap \overline{\pi^{\exists}_{\{c\}}(K_{t_0 a} \cap \overline{C})}$

Example

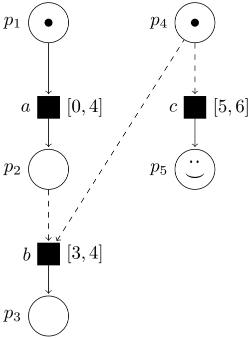


Figure: Example with 2 uncontrollable transitions

Semantics

Transition system $(S, S_0, \Sigma, \rightarrow)$

- $S \subseteq 2^P \times (\mathbb{R}_{\geq 0} \cup \{\perp\})^{|T|}$ with $S_0 \in S$
- $\Sigma = T \times \mathbb{R}_{\geq 0}$ with t transition fired and d time since last firing
- $\rightarrow : S \times \Sigma \rightarrow S$ transition function :

$$
\begin{aligned}
(m, \theta) \xrightarrow{t@d} (m', \theta') \iff \\
& t \in \mathsf{en}(m) \wedge \theta(t) = d \\
& \wedge (\forall k \in T, k \in \mathsf{en}(m) \implies \theta(t) \leq \theta(k)) \\
& \wedge m' = (m \setminus Pre(t)) \cup Post(t) \\
& \wedge \forall k \in T, k \in \mathsf{en}(m') \implies \\
& \qquad \begin{pmatrix} \theta'(k) \in I_s(k) \text{ if } k \in \mathsf{newen}(m, t) \\ \wedge \theta'(k) = \theta(k) - \theta(t) \text{ if } k \in \mathsf{pers}(m, t) \\ \wedge \theta'(k) = \perp \text{ otherwise} \end{pmatrix}
\end{aligned}
$$

Semantics

Loriane: TODO redo just states

Timed Transition System $(S, s_0, \Sigma, \rightarrow)$ with:

- S the set of states,
- initial state $s_0 = (\{p_0\}, \theta_0) \in S$ with $\theta_0(t_{\mathsf{init}}) = 0$
- a labelling alphabet Σ containing letters $t_f \in T$ and $d \in \mathbb{R}_{\geq 0}$,
- the transition relation $\rightarrow \subseteq S \times \Sigma \times S$:
  $s \xrightarrow{a} s' \Leftrightarrow$
  - either $(m, \theta) \xrightarrow{t_f} (m', \theta')$ for $t_f \in T$ when:
    1. $t_f \in \mathsf{en}(m)$ and $\theta_f = 0$
    2. $m' = (m \setminus \mathsf{Pre}(t_f)) \cup \mathsf{Post}(t_f)$
    3. $\forall t_k \in T, \begin{pmatrix} \theta'_k \in I_s(t_k) & \text{if } t_k \in \mathsf{newen}(m, t_f) \\ \wedge \theta'_k = \theta_k & \text{if } t_k \in \mathsf{pers}(m, t_f) \\ \wedge \theta'_k = \perp & \text{otherwise} \end{pmatrix}$
  - or $(m, \theta) \xrightarrow{d} (m, \theta')$ when:
    - $d \in \mathbb{R}_{\geq 0} \setminus \{0\}$,
    - $\forall t_k \notin \mathsf{en}(m), \theta_k = \perp$, and
    - $\forall t_k \in \mathsf{en}(m), \theta_k - d \geq 0$ and $\theta'_k = \theta_k - d$.

State Class Graph

algorithm from Berthomieu et al.[2]

- Initial system $K_\epsilon = \{\theta_k \in I_s(k) \mid t_k \in \mathsf{en}(m_0)\}$
- if σ firable then $\sigma.t_k$ firable if and only if :
  1. $t \in \mathsf{en}(m)$
  2. $K_\sigma \wedge \{\theta_k \leq \theta_i \mid i \neq k \wedge t_i \in \mathsf{en}(m)\}$ consistent
- If $\sigma.t_k$ is firable, then $K_{\sigma.t_k}$ is computed from K_σ :
  - Add $\{\theta_k \leq \theta_i \mid i \neq k \wedge t_i \in \mathsf{en}(m)\}$ to K_σ
  - $\forall t_i \in \mathsf{en}(m')$ we add θ'_i such that:
    $\theta'_i = \theta_i - \theta_k$ if $k \neq i \wedge t_i \notin \mathsf{newen}(m, t_k)$
    $\theta'_i \in I_s(i)$ otherwise
  - Eliminate θ_i variables $\forall i$

[2] Berthomieu and Menasche, "An Enumerative Approach For Analyzing Time Petri Nets".

State Class Graph

algorithm from Berthomieu et al.[3]

Algorithm Successor (m', D') of (m, D) by firing firable transition t_f

---

1: $m' \leftarrow (m \setminus \mathsf{Pre}(t_f)) \cup \mathsf{Post}(t_f)$
2: $D' \leftarrow D \wedge \bigwedge_{i \neq f, i \in \mathsf{en}(m)} \theta_f \leq \theta_i$
3: for all $i \in \mathsf{en}(m \setminus \mathsf{Pre}(t_f)), i \neq f$, add variable θ'_i to D', constrained by $\theta'_i = \theta_i - \theta_f$
4: eliminate (by existential projection) variables θ_i for all i from D'
5: for all $i \in \mathsf{newen}(m, t_f)$, add variable θ''_i to D', constrained by $\theta''_i \in I_s(i)$

---

[3] Berthomieu and Menasche, "An Enumerative Approach For Analyzing Time Petri Nets".

Game definition

A reachability game $\mathcal{R} = (\mathcal{A}, \mathsf{Win})$
an arena $\mathcal{A} = (S, \to, Pl, (Mov_i)_{i \in Pl}, Trans)$
a target set $\mathsf{Win} \in S$

Game definition

Moves

$$\mathsf{Movt}_c(m,\theta) = t_c \in \mathsf{en}(m) \cap T_c \qquad (\cup\{\emptyset\} \text{ if enable empty ?})$$

$$\mathsf{Movt}_c(m,\theta) = t_u \in \mathsf{en}(m)$$

$$\mathsf{Movf}_c(m,\theta,t) = \left\{ \theta'_c \in \mathbb{R}_{\geq 0}^{|T_c|} \;\middle|\; \begin{array}{l} \theta'_c(k) \in I_s(k) \text{ if } k \in \mathsf{newen}(m,t) \\ \theta'_c(k) = \theta(k) - \theta(t) \text{ if } k \in \mathsf{pers}(m,t) \\ \theta'_c(k) = \perp \text{ otherwise} \end{array} \right\}$$

$$\mathsf{Movf}_u(m,\theta,t) = \left\{ \theta'_u \in \mathbb{R}_{\geq 0}^{|T_u|} \;\middle|\; \begin{array}{l} \theta'_u(k) \in I_s(k) \text{ if } k \in \mathsf{newen}(m,t) \\ \theta'_u(k) = \theta(k) - \theta(t) \text{ if } k \in \mathsf{pers}(m,t) \\ \theta'_u(k) = \perp \text{ otherwise} \end{array} \right\}$$

Game definition

Trans

$$Trans : S \times T \times T \times \mathbb{R}_{\geq 0}^{|T_c|}, \mathbb{R}_{\geq 0}^{|T_u|} \rightarrow S$$

$$Trans((m,\theta), t_c, t_u, \theta'_c, \theta'_u) = \text{if } \begin{pmatrix} t_u \in \mathsf{Movt}_u(m,\theta) \\ \wedge t_c \in \mathsf{Movt}_c(m,\theta) \\ \wedge \theta(t_u) = min_k(\theta(t_k)) \\ \wedge (t_u \in T_u \vee t_c = t_u) \\ \wedge \theta'_c \in \mathsf{Movf}_c(m,\theta,t_u) \\ \wedge \theta'_u \in \mathsf{Movf}_u(m,\theta,t_u) \end{pmatrix}$$

$$\text{then } ((m \setminus Pre(t)) \cup Post(t), \theta'_c \sqcup \theta'_u)$$

$$\text{otherwise } \perp$$

Predecessor

$$\mathsf{Pred}_{C \xrightarrow{t} C'}(B) = \{s = (m, \theta) \in C \mid \theta(t) = min_k(\theta(t_k)) \wedge \exists s'.s \xrightarrow{t} s' \in \mathcal{G} \wedge s' \in B\}$$

$$\begin{aligned}\mathsf{cPred}_{C \xrightarrow{t} C'}(B) = \{s \mid s \in C \wedge \\ & \exists s \xrightarrow{t} s' \in \mathcal{G}, s' \in B \wedge \\ & \forall s' \in C'.s \xrightarrow{t} s' \in \mathcal{G} \implies \\ & \quad \big(s' \in B \vee [\exists s'' \in B.s \xrightarrow{t} s'' \in \mathcal{G} \wedge D'|_{\substack{(\mathsf{newen}(C,t) \cap T_u) \\ \cup \mathsf{pers}(C,t)}} =\end{aligned}$$

Proj ∃ ∀

For two state classes C, C' s.t. $C \xrightarrow{t} C'$.

An *existential projection*:

$$\pi^{\exists}_{\{t_1,\ldots,t_n\}}(C) = \{(m, \theta_1 \ldots \theta_n) \mid \exists \theta_{n+1}, \ldots, \theta_{n+k}, (m, \theta_1 \ldots \theta_{n+k}) \in C\}$$

A *universal projection*:

$$\pi^{\forall}_{\{t_1,\ldots,t_n\}}(C, C') = \{(m, \theta_1 \ldots \theta_n) \mid \exists \theta_{n+1}, \ldots, \theta_{n+k}, (m, \theta_1 \ldots \theta_{n+k}) \in C \\ \wedge \forall \theta_{n+1}, \ldots, \theta_{n+k}, (m, \theta_1 \ldots \theta_{n+k}) \in C \implies (r$$

Extension and substitution operations

For a class $C = (m, D)$ with D of dimension n and t a constant $\in \mathbb{R}_{\geq 0}$.
An *extension operation*:

$$\pi^{-1}_{\{t_1,\ldots,t_{n+k}\}}(C) = \{(m, \theta_1 \ldots \theta_{n+k}) \mid (m, \theta_1 \ldots \theta_n) \in C\}$$

A *substitution operation*:

$$C[\theta_i \leftarrow (\theta_i + t)] = \{(m, \theta'_1 \ldots \theta'_n) \mid \exists (m, \theta_1 \ldots \theta_n) \in C s.t. \forall 1 \leq i \leq n, \theta'_i = \theta_i - t\}$$

$$(or \;\; C[\theta_i \leftarrow \theta'_i] = \{(m, \theta'_1 \ldots \theta'_n) \mid \exists (m, \theta_1 \ldots \theta_n) \in C\})$$

The universal projection is expressible with set complements and existential projection only as stated in the following proposition.

Proposition

$\pi_T^{\forall}(C, C') = \pi_T^{\exists}(C) \cap \overline{\pi_T^{\exists}(\overline{C'} \cap C)}$.

Symbolic computing of predecessors

$$\mathsf{Pred}_{C \xrightarrow{t} C'}(B) = C \cap \pi^{-1}_{\mathsf{en}(C)}\left[\left(\pi^{\exists}_{\mathsf{pers}(C,t)}(B)\right)[\theta_i \leftarrow (\theta_i - t)]\right]$$

$$\begin{aligned}\mathsf{cPred}_{C \xrightarrow{t} C'}(B) &= C \cap \pi^{-1}_{\mathsf{en}(C)}\left(\pi^{\exists}_{\mathsf{pers}(C,t)}\left(\pi^{\forall}_{\substack{(\mathsf{newen}(C,t)\cap T_c) \\ \cup \mathsf{pers}(C,t)}}(C', B)\right)[\theta_i \leftarrow (\theta_i - \theta_t)]\right) \\ &= \mathsf{Pred}_{C \xrightarrow{t} C'}\left(\pi^{\forall}_{\substack{(\mathsf{newen}(C,t)\cap T_c) \\ \cup \mathsf{pers}(C,t)}}(C', B)\right)\end{aligned}$$