Parameter Synthesis for Bounded-cost Reachability in Time Petri Nets

Didier Lime[1] Olivier H. Roux[1] Charlotte Seidner[2]

[1]École Centrale de Nantes, LS2N, France

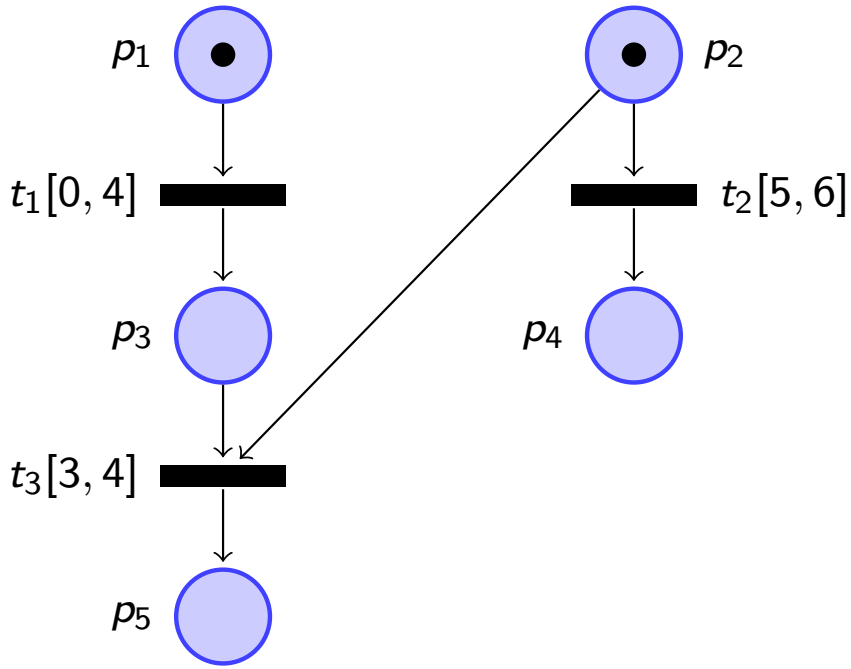
[2]University of Nantes, LS2N, France

Petri Nets 2019
26th of June 2019, Aachen, Germany

Parametric Cost Models

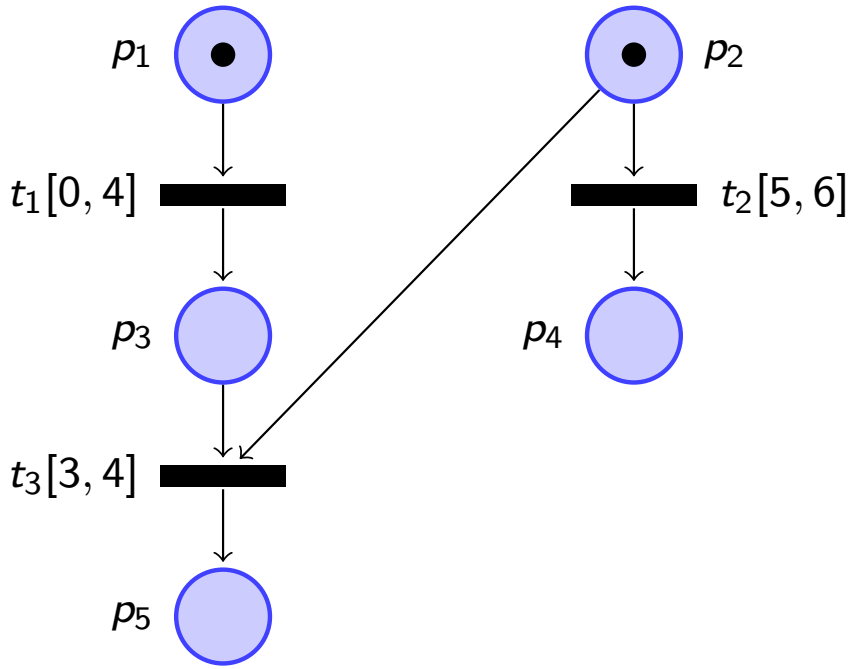
- We want to model systems with:
  - complex interactions,
  - quantitative but **underspecified timing** constraints.
- We want to find paths to states of interest, and compare them according to a **cost** evolving:
  - upon **discrete** events;
  - when letting **time** elapse.
- We define **Parametric Cost Time Petri Nets**

Time Petri Nets



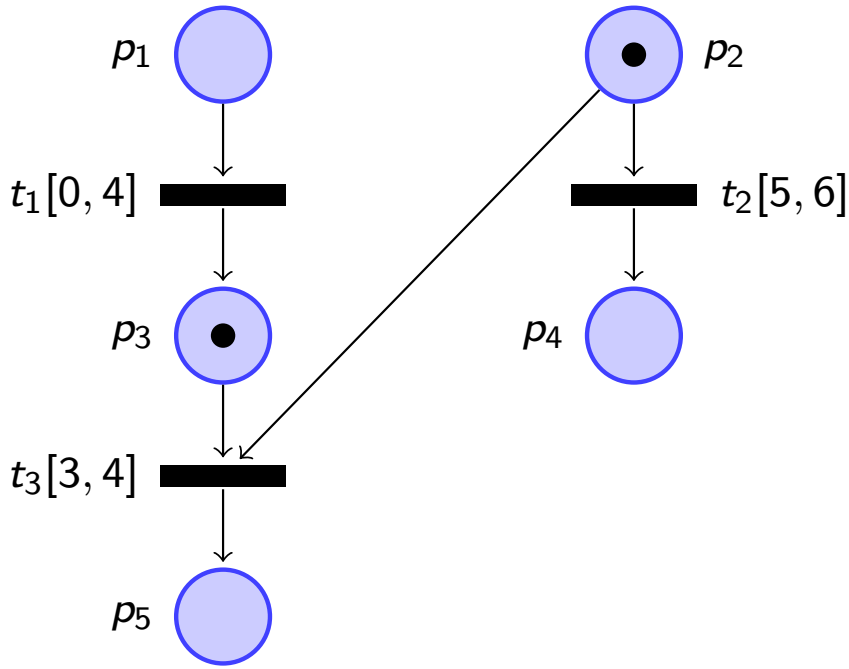
$t_1 \in [0,4]$
$t_2 \in [5,6]$

Time Petri Nets



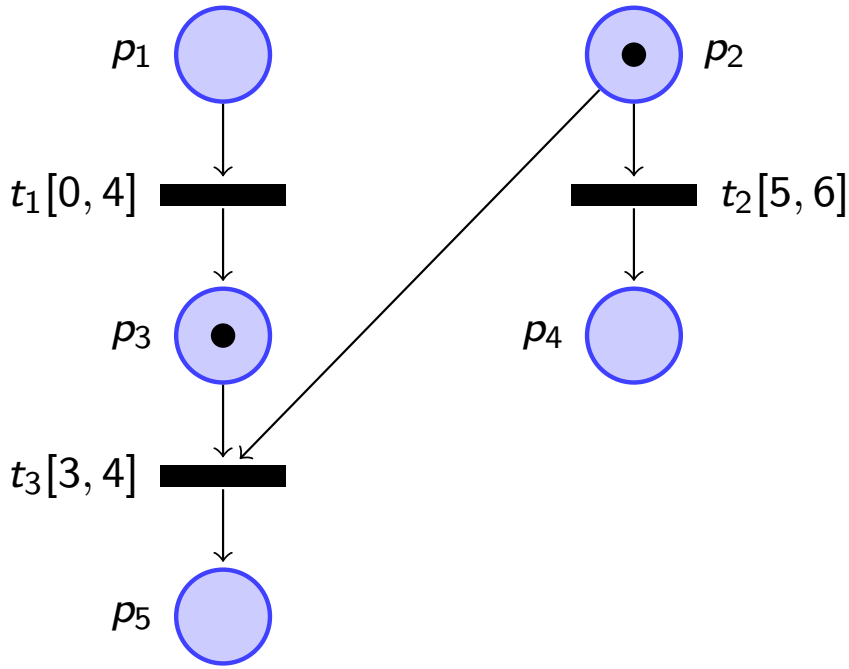
$$\begin{array}{l} t_1 \in [0,4] \\ t_2 \in [5,6] \end{array} \xrightarrow{1.4} \begin{array}{l} t_1 \in [0,2.6] \\ t_2 \in [3.6,4.6] \end{array}$$

Time Petri Nets



$$\begin{matrix} t_1 \in [0,4] \\ t_2 \in [5,6] \end{matrix} \xrightarrow{1.4} \begin{matrix} t_1 \in [0,2.6] \\ t_2 \in [3.6,4.6] \end{matrix} \xrightarrow{t_1} \begin{matrix} t_2 \in [3.6,4.6] \\ t_3 \in [3,4] \end{matrix}$$

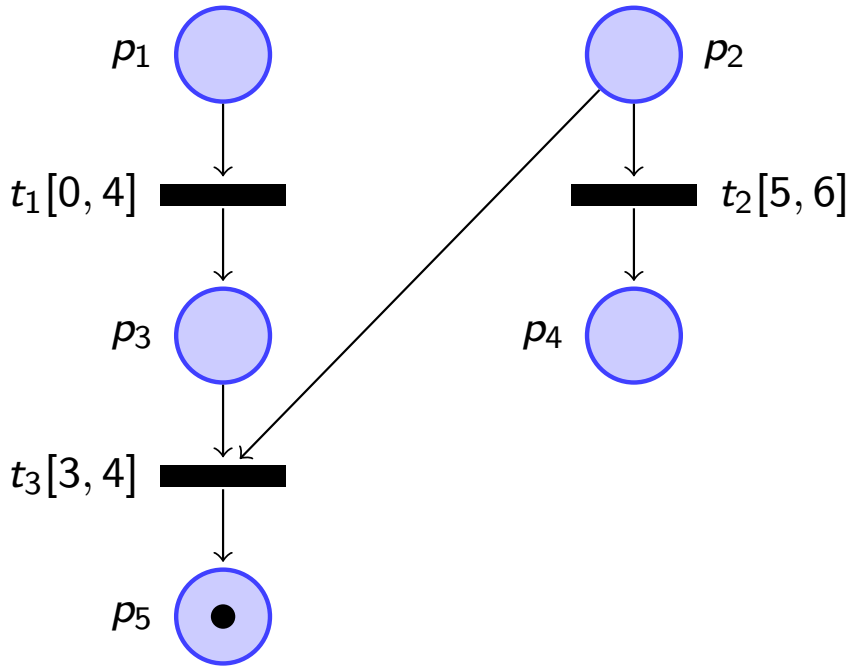
Time Petri Nets



$$\begin{matrix} t_1 \in [0,4] \\ t_2 \in [5,6] \end{matrix} \xrightarrow{1.4} \begin{matrix} t_1 \in [0,2.6] \\ t_2 \in [3.6,4.6] \end{matrix} \xrightarrow{t_1} \begin{matrix} t_2 \in [3.6,4.6] \\ t_3 \in [3,4] \end{matrix}$$

$$\xrightarrow{3.6} \begin{matrix} t_2 \in [0,1] \\ t_3 \in [0,0.4] \end{matrix}$$

Time Petri Nets



$$\begin{matrix} t_1 \in [0,4] \\ t_2 \in [5,6] \end{matrix} \xrightarrow{1.4} \begin{matrix} t_1 \in [0,2.6] \\ t_2 \in [3.6,4.6] \end{matrix} \xrightarrow{t_1} \begin{matrix} t_2 \in [3.6,4.6] \\ t_3 \in [3,4] \end{matrix}$$

$$\xrightarrow{3.6} \begin{matrix} t_2 \in [0,1] \\ t_3 \in [0,0.4] \end{matrix} \xrightarrow{t_3} \bot$$

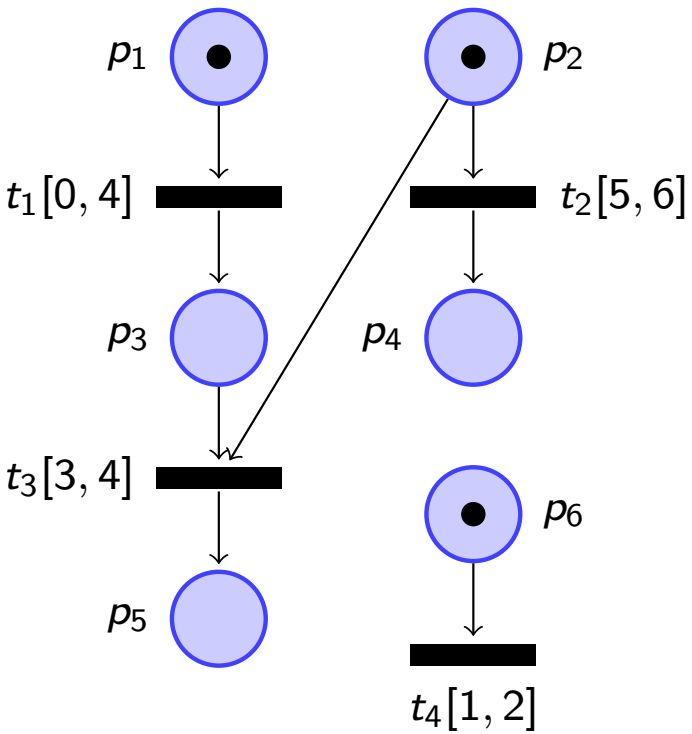
State Classes [BD91]

- A **state class** C_σ is the (collapsed) set of states obtained by the transition sequence σ;
- Those states all share the same marking;
- The union of all points in the intervals in the valuations on the transitions can be represented by a **convex polyhedron**;
- A state class is thus a pair $C = (m, D)$, where m is a marking, and D a polyhedron.

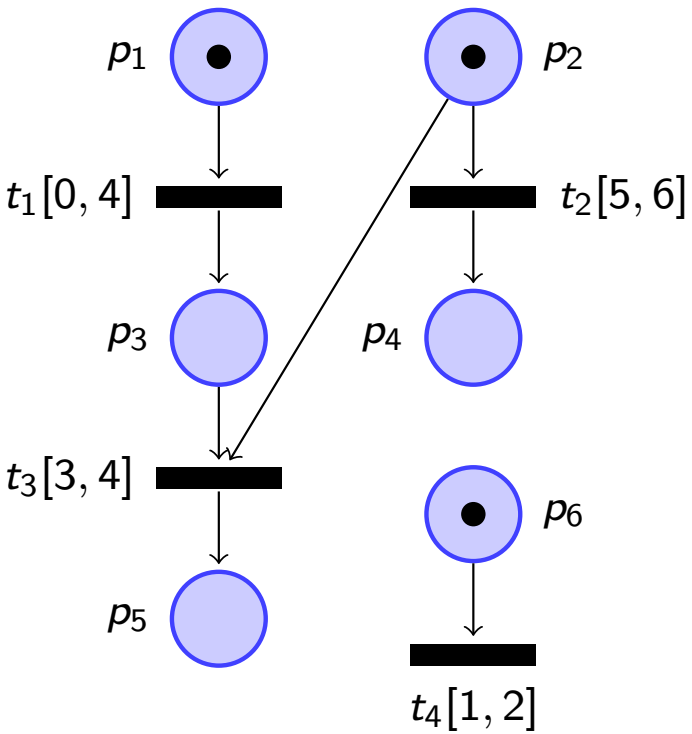
State Class Computation

Initially:

$$\theta_1 \in [0, 4]$$
$$\theta_2 \in [5, 6]$$
$$\theta_4 \in [1, 2]$$



State Class Computation



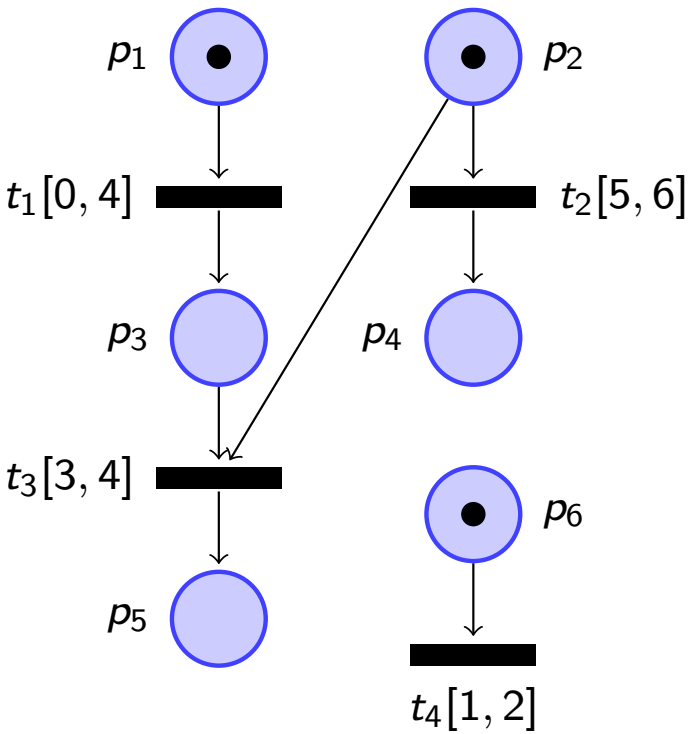
Initially:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [1,2]$

Fire t_1:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [1,2]$
$\theta_1 \leq \theta_2$
$\theta_1 \leq \theta_4$

State Class Computation



Initially:

$$\theta_1 \in [0,4]$$
$$\theta_2 \in [5,6]$$
$$\theta_4 \in [1,2]$$

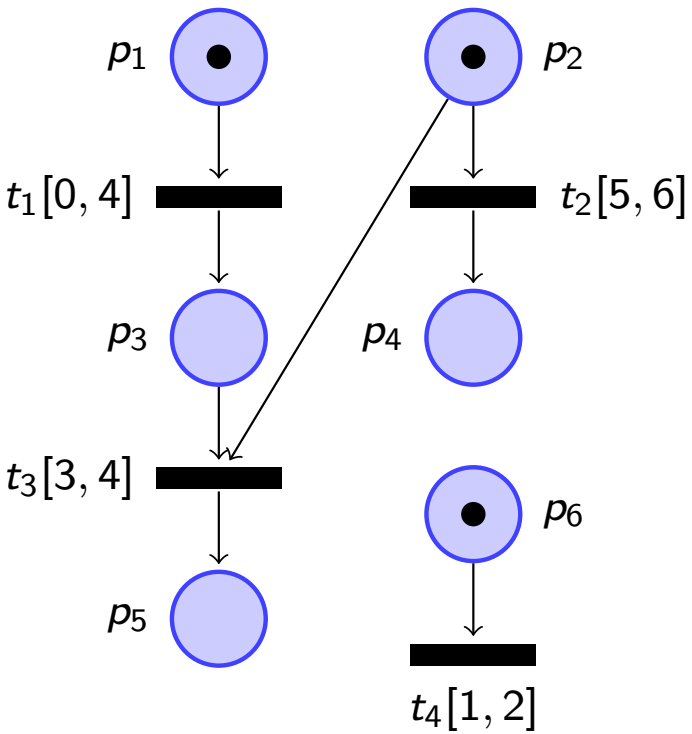
Fire t_1:

$$\theta_1 \in [0,4]$$
$$\theta_2 \in [5,6]$$
$$\theta_4 \in [1,2]$$
$$\theta_1 \leq \theta_2$$
$$\theta_1 \leq \theta_4$$

Change origin:

$$\theta_1 \in [0,4]$$
$$\theta'_2 + \theta_1 \in [5,6]$$
$$\theta'_4 + \theta_1 \in [1,2]$$
$$\theta_1 \leq \theta'_2 + \theta_1$$
$$\theta_1 \leq \theta'_4 + \theta_1$$

State Class Computation



Initially:

$$\theta_1 \in [0,4]$$
$$\theta_2 \in [5,6]$$
$$\theta_4 \in [1,2]$$

Fire t_1:

$$\theta_1 \in [0,4]$$
$$\theta_2 \in [5,6]$$
$$\theta_4 \in [1,2]$$
$$\theta_1 \leq \theta_2$$
$$\theta_1 \leq \theta_4$$

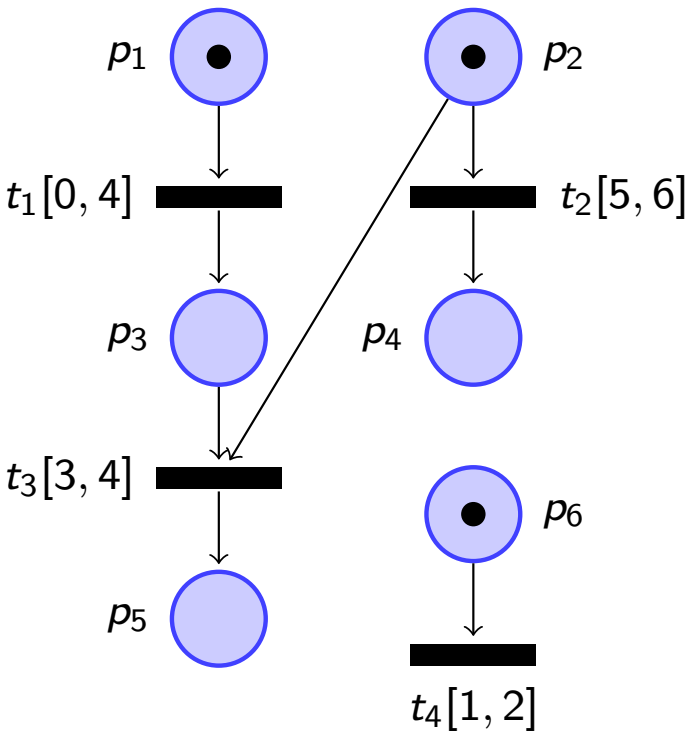
Eliminate disabled:

$$\theta'_2 \in [3,6]$$
$$\theta'_4 \in [0,2]$$
$$\theta'_2 - \theta'_4 \in [3,5]$$

Change origin:

$$\theta_1 \in [0,4]$$
$$\theta'_2 + \theta_1 \in [5,6]$$
$$\theta'_4 + \theta_1 \in [1,2]$$
$$\theta_1 \leq \theta'_2 + \theta_1$$
$$\theta_1 \leq \theta'_4 + \theta_1$$

State Class Computation



Initially:

$$\theta_1 \in [0,4]$$
$$\theta_2 \in [5,6]$$
$$\theta_4 \in [1,2]$$

Fire t_1:

$$\theta_1 \in [0,4]$$
$$\theta_2 \in [5,6]$$
$$\theta_4 \in [1,2]$$
$$\theta_1 \leq \theta_2$$
$$\theta_1 \leq \theta_4$$

Change origin:

$$\theta_1 \in [0,4]$$
$$\theta'_2 + \theta_1 \in [5,6]$$
$$\theta'_4 + \theta_1 \in [1,2]$$
$$\theta_1 \leq \theta'_2 + \theta_1$$
$$\theta_1 \leq \theta'_4 + \theta_1$$

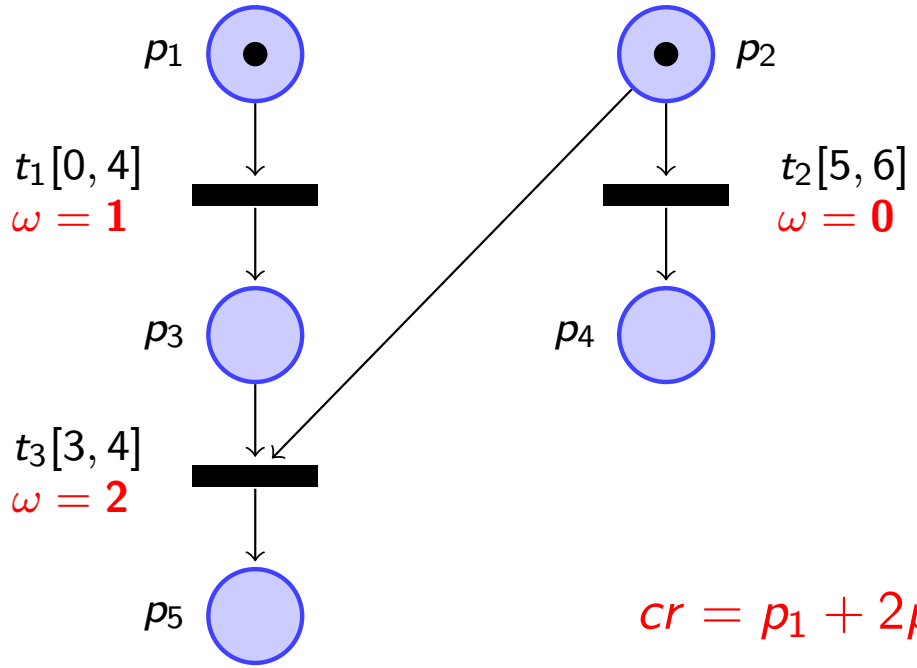
Eliminate disabled:

$$\theta'_2 \in [3,6]$$
$$\theta'_4 \in [0,2]$$
$$\theta'_2 - \theta'_4 \in [3,5]$$

Add newly enabled:

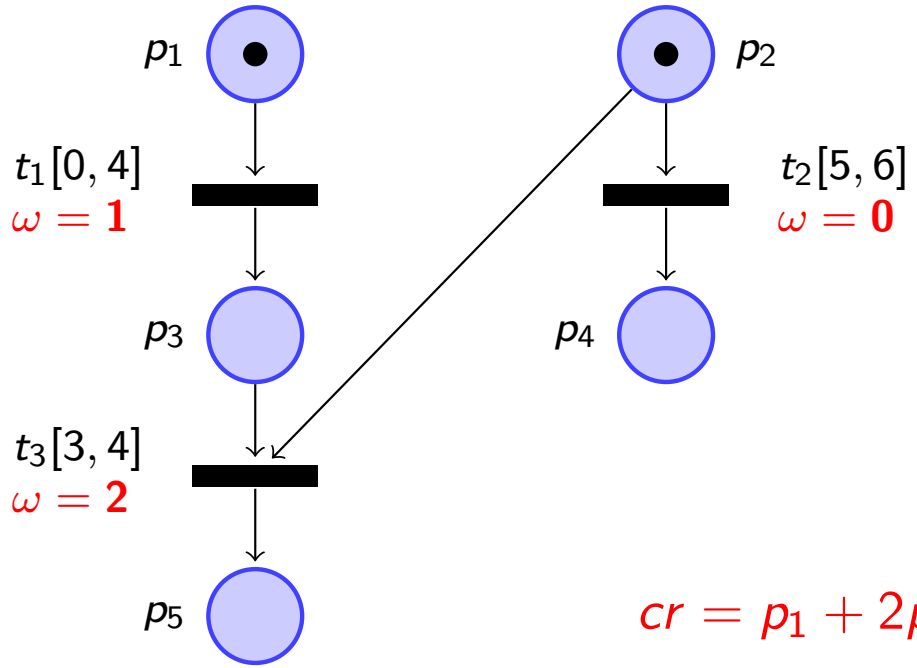
$$\theta'_2 \in [3,6]$$
$$\theta_3 \in [3,4]$$
$$\theta'_4 \in [0,2]$$
$$\theta'_2 - \theta'_4 \in [3,5]$$

Cost Time Petri Nets



$t_1 \in [0,4]$
$t_2 \in [5,6]$
cost $= \mathbf{0}$

Cost Time Petri Nets

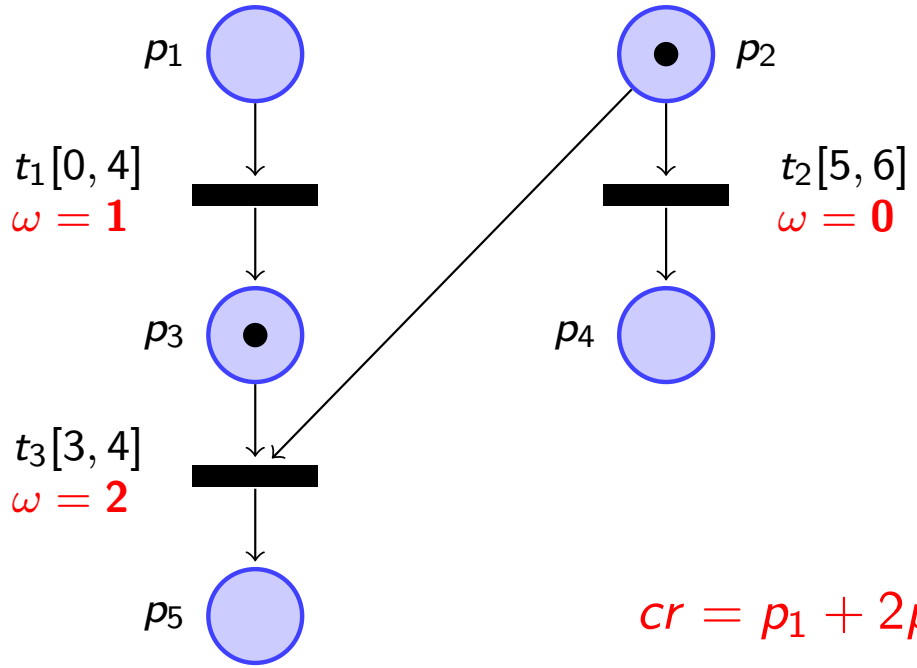


$t_1 \in [0,4]$
$t_2 \in [5,6]$
$\text{cost} = \mathbf{0}$

$\xrightarrow{1.4}$

$t_1 \in [0,2.6]$
$t_2 \in [3.6,4.6]$
$\text{cost} = (\mathbf{1} + \mathbf{2}) * \mathbf{1.4} = \mathbf{4.2}$

Cost Time Petri Nets



$cr = p_1 + 2p_2$

$t_1 \in [0,4]$
$t_2 \in [5,6]$
$\text{cost} = \mathbf{0}$

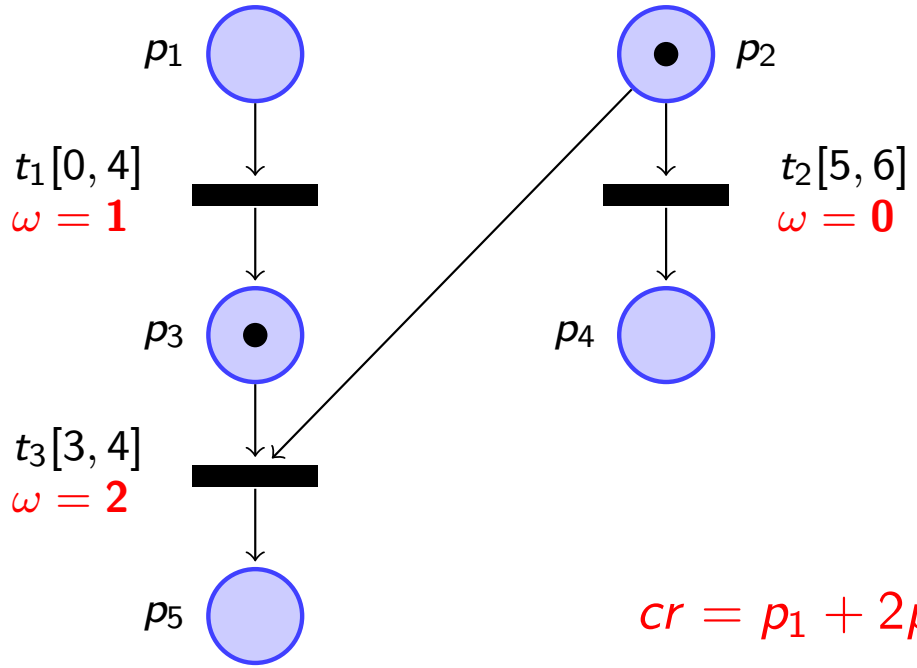
$\xrightarrow{1.4}$

$t_1 \in [0,2.6]$
$t_2 \in [3.6,4.6]$
$\text{cost} = (\mathbf{1}+\mathbf{2}) * \mathbf{1.4} = \mathbf{4.2}$

$\xrightarrow{t_1}$

$t_2 \in [3.6,4.6]$
$t_3 \in [3,4]$
$\text{cost} = \mathbf{4.2} + \mathbf{1} = \mathbf{5.2}$

Cost Time Petri Nets



$t_1 \in [0,4]$
$t_2 \in [5,6]$
$\text{cost} = \mathbf{0}$

$\xrightarrow{1.4}$

$t_1 \in [0,2.6]$
$t_2 \in [3.6,4.6]$
$\text{cost} = (\mathbf{1}+\mathbf{2}) * \mathbf{1.4} = \mathbf{4.2}$

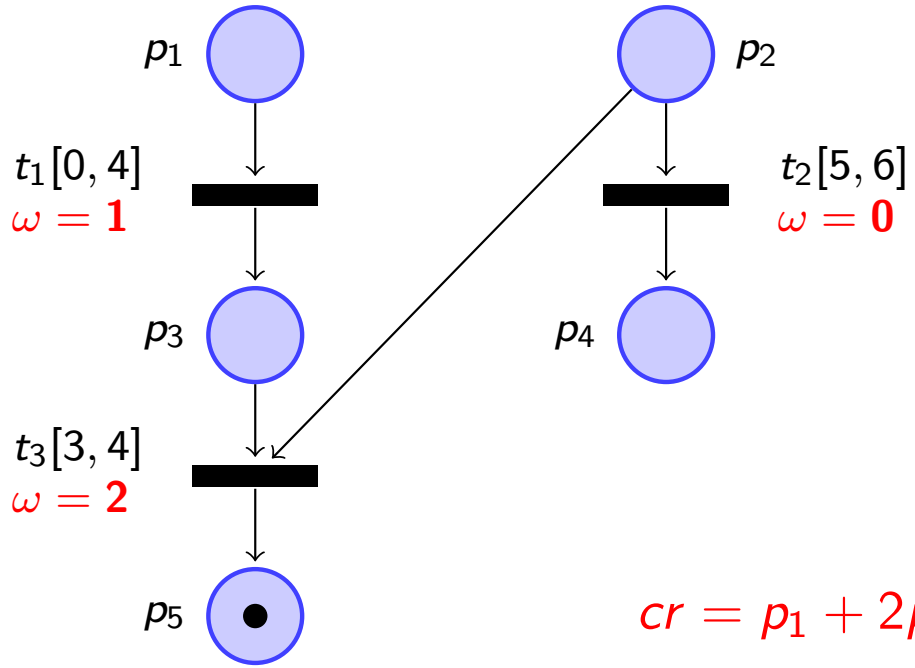
$\xrightarrow{t_1}$

$t_2 \in [3.6,4.6]$
$t_3 \in [3,4]$
$\text{cost} = \mathbf{4.2} + \mathbf{1} = \mathbf{5.2}$

$\xrightarrow{3.6}$

$t_2 \in [0,1]$
$t_3 \in [0,0.4]$
$\text{cost} = \mathbf{5.2} + \mathbf{2} * \mathbf{3.6} = \mathbf{12.4}$

Cost Time Petri Nets



$t_1 \in [0,4]$, $t_2 \in [5,6]$, $\text{cost} = \mathbf{0}$ $\xrightarrow{1.4}$ $t_1 \in [0,2.6]$, $t_2 \in [3.6,4.6]$, $\text{cost} = (\mathbf{1}+\mathbf{2}) * \mathbf{1.4} = \mathbf{4.2}$ $\xrightarrow{t_1}$ $t_2 \in [3.6,4.6]$, $t_3 \in [3,4]$, $\text{cost} = \mathbf{4.2} + \mathbf{1} = \mathbf{5.2}$

$\xrightarrow{3.6}$ $t_2 \in [0,1]$, $t_3 \in [0,0.4]$, $\text{cost} = \mathbf{5.2} + \mathbf{2} * \mathbf{3.6} = \mathbf{12.4}$ $\xrightarrow{t_3}$ $\bot$, $\text{cost} = \mathbf{12.4} + \mathbf{2} = \mathbf{14.4}$

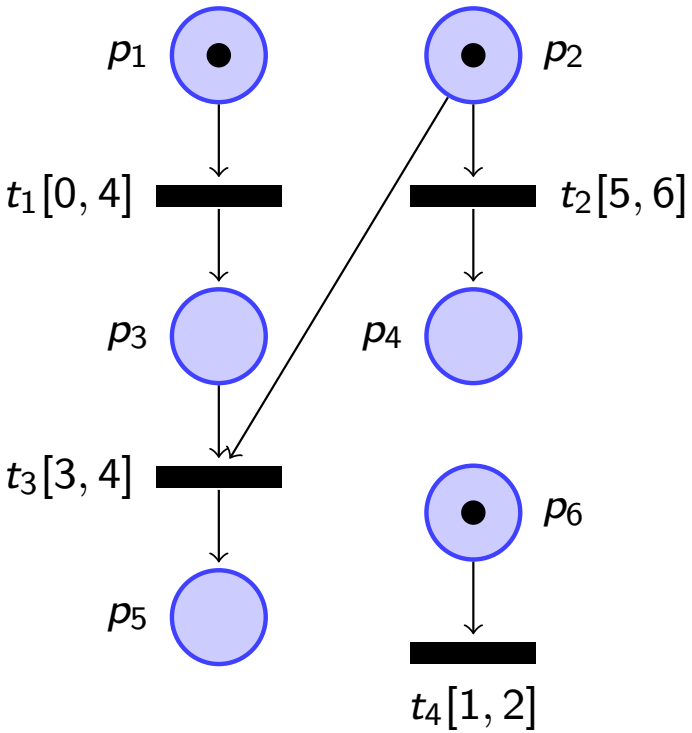
Cost State Classes

- From a transition sequence σ, we want to compute cost(σ) the inf-cost of all runs built on σ
- We extend state class firing domains with a new variable c:
  $\Rightarrow$ Cost state classes: $C_\sigma = (m, D)$
- When firing t_i, c changes by $\omega(t_i) + \theta_i * cr(m)$
- An extended firing domain D is still a **convex polyhedron**;
- cost(σ) = cost(C_σ) = $\inf_{(\vec{\theta},c)\in D} c$ computable using **linear programming**.

Cost State Classes: Example

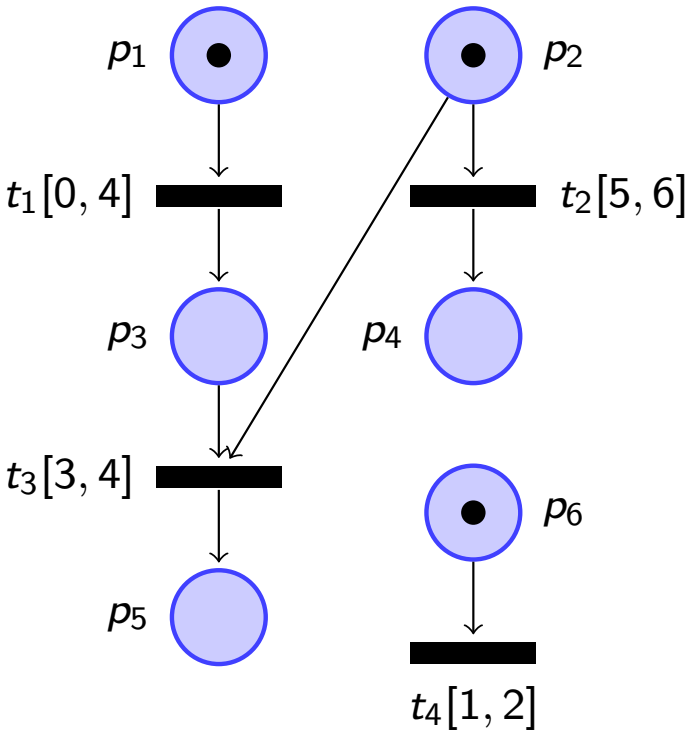
Initially:

$\theta_1 \in [0, 4]$
$\theta_2 \in [5, 6]$
$\theta_4 \in [1, 2]$
$\mathbf{c \geq 0}$



$cr = p_1 + 2p_2$
$\omega(t_1) = 1$

Cost State Classes: Example



$cr = p_1 + 2p_2$
$\omega(t_1) = 1$

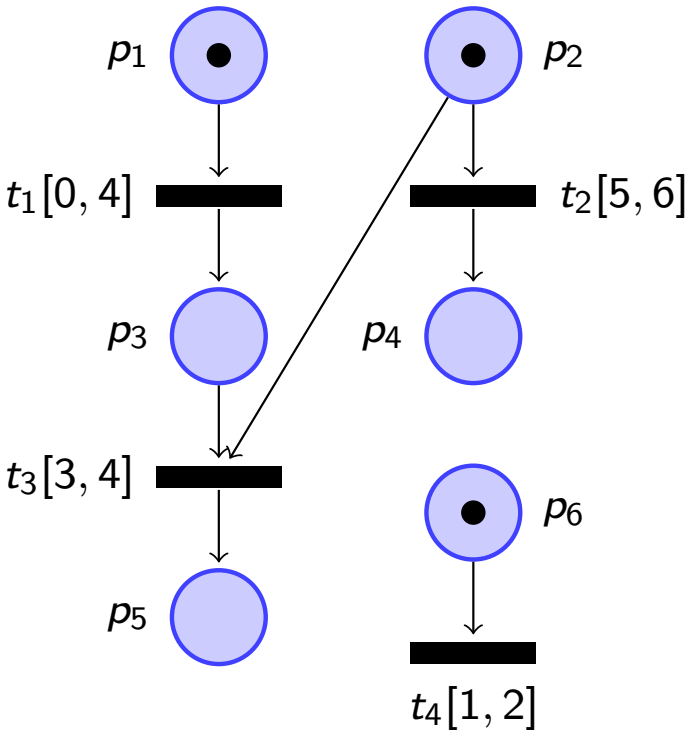
Initially:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [1,2]$
$\mathbf{c \geq 0}$

Fire t_1:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [1,2]$
$c \geq 0$
$\theta_1 \leq \theta_2$
$\theta_1 \leq \theta_4$

Cost State Classes: Example



$cr = p_1 + 2p_2$

$\omega(t_1) = 1$

Initially:

$\theta_1 \in [0,4]$

$\theta_2 \in [5,6]$

$\theta_4 \in [1,2]$

$\mathbf{c \geq 0}$

Fire t_1:

$\theta_1 \in [0,4]$

$\theta_2 \in [5,6]$

$\theta_4 \in [1,2]$

$c \geq 0$

$\theta_1 \leq \theta_2$

$\theta_1 \leq \theta_4$

Change origin:

$\theta_1 \in [0,4]$

$\theta'_2 + \theta_1 \in [5,6]$

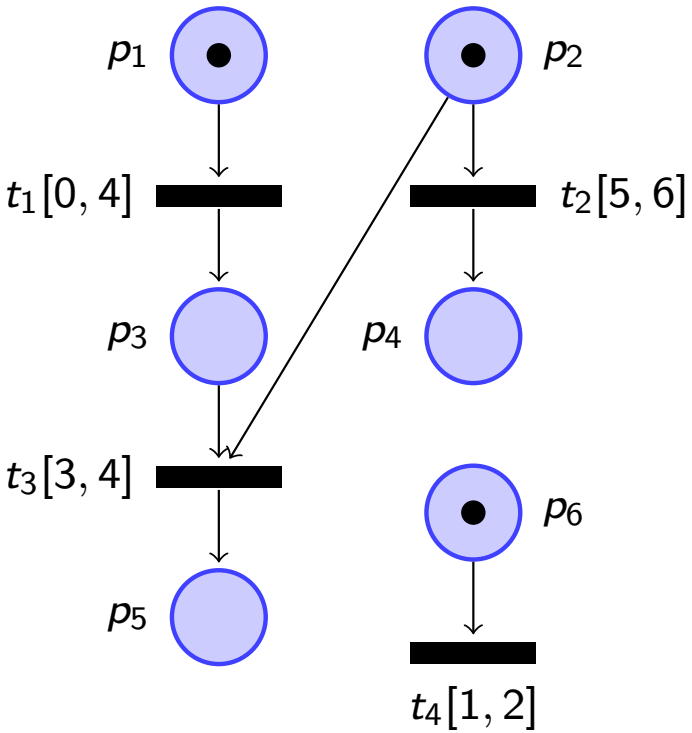
$\theta'_4 + \theta_1 \in [1,2]$

$\mathbf{c' - \omega(t_1) - cr(m_0) * \theta_1 \geq 0}$

$\theta_1 \leq \theta'_2 + \theta_1$

$\theta_1 \leq \theta'_4 + \theta_1$

Cost State Classes: Example



$cr = p_1 + 2p_2$

$\omega(t_1) = 1$

Initially:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [1,2]$
$\mathbf{c \geq 0}$

Fire t_1:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [1,2]$
$c \geq 0$
$\theta_1 \leq \theta_2$
$\theta_1 \leq \theta_4$

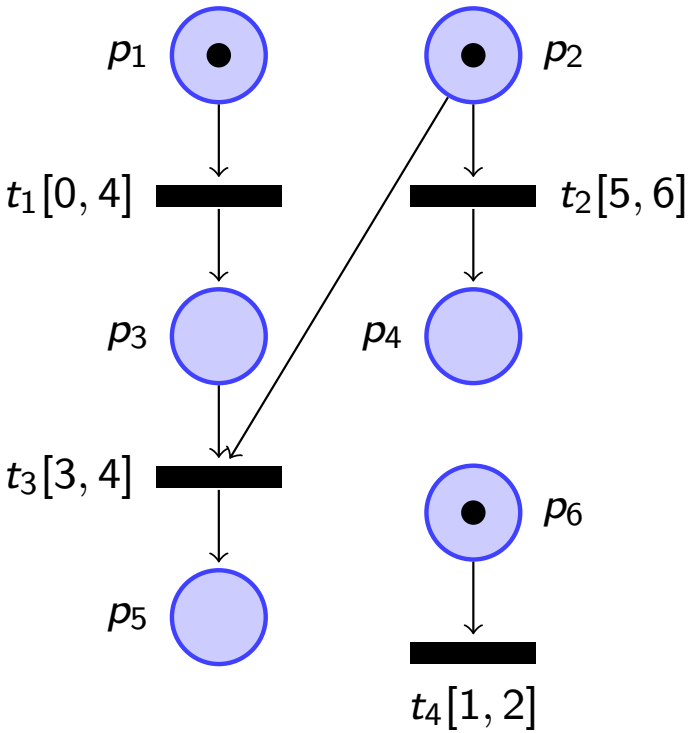
Change origin:

$\theta_1 \in [0,4]$
$\theta'_2 + \theta_1 \in [5,6]$
$\theta'_4 + \theta_1 \in [1,2]$
$\mathbf{c' - \omega(t_1) - cr(m_0) * \theta_1 \geq 0}$
$\theta_1 \leq \theta'_2 + \theta_1$
$\theta_1 \leq \theta'_4 + \theta_1$

Eliminate disabled:

$\theta'_2 \in [3,6]$
$\theta'_4 \in [0,2]$
$\theta'_2 - \theta'_4 \in [3,5]$
$\mathbf{c \geq 16 - 3t'_2}$
$\mathbf{c \geq 4 - 3t'_4}$
$\mathbf{c \geq 1}$

Cost State Classes: Example



$cr = p_1 + 2p_2$

$\omega(t_1) = 1$

Initially:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [1,2]$
$\mathbf{c \geq 0}$

Fire t_1:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [1,2]$
$c \geq 0$
$\theta_1 \leq \theta_2$
$\theta_1 \leq \theta_4$

Change origin:

$\theta_1 \in [0,4]$
$\theta'_2 + \theta_1 \in [5,6]$
$\theta'_4 + \theta_1 \in [1,2]$
$\mathbf{c' - \omega(t_1) - cr(m_0) * \theta_1 \geq 0}$
$\theta_1 \leq \theta'_2 + \theta_1$
$\theta_1 \leq \theta'_4 + \theta_1$

Eliminate disabled:

$\theta'_2 \in [3,6]$
$\theta'_4 \in [0,2]$
$\theta'_2 - \theta'_4 \in [3,5]$
$\mathbf{c \geq 16 - 3t'_2}$
$\mathbf{c \geq 4 - 3t'_4}$
$\mathbf{c \geq 1}$

Add newly enabled:

$\theta'_2 \in [3,6]$
$\theta_3 \in [3,4]$
$\theta'_4 \in [0,2]$
$\theta'_2 - \theta'_4 \in [3,5]$
$c \geq 16 - 3t'_2$
$c \geq 4 - 3t'_4$
$c \geq 1$

Bounded-Cost Reachability

Bounded-Cost Reachability

Can we reach a marking from a given set Goal with a cost less or equal to a given bound c_{max}?

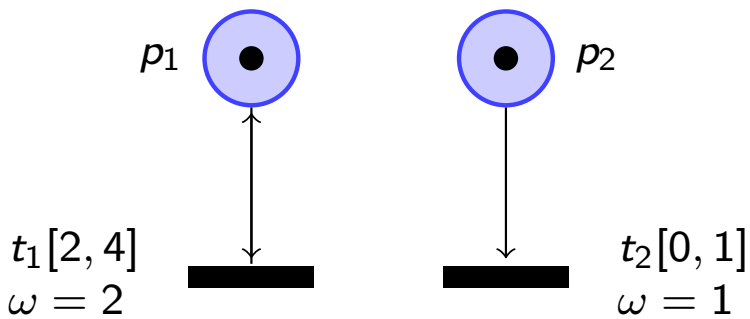
Bounded-Cost Reachability

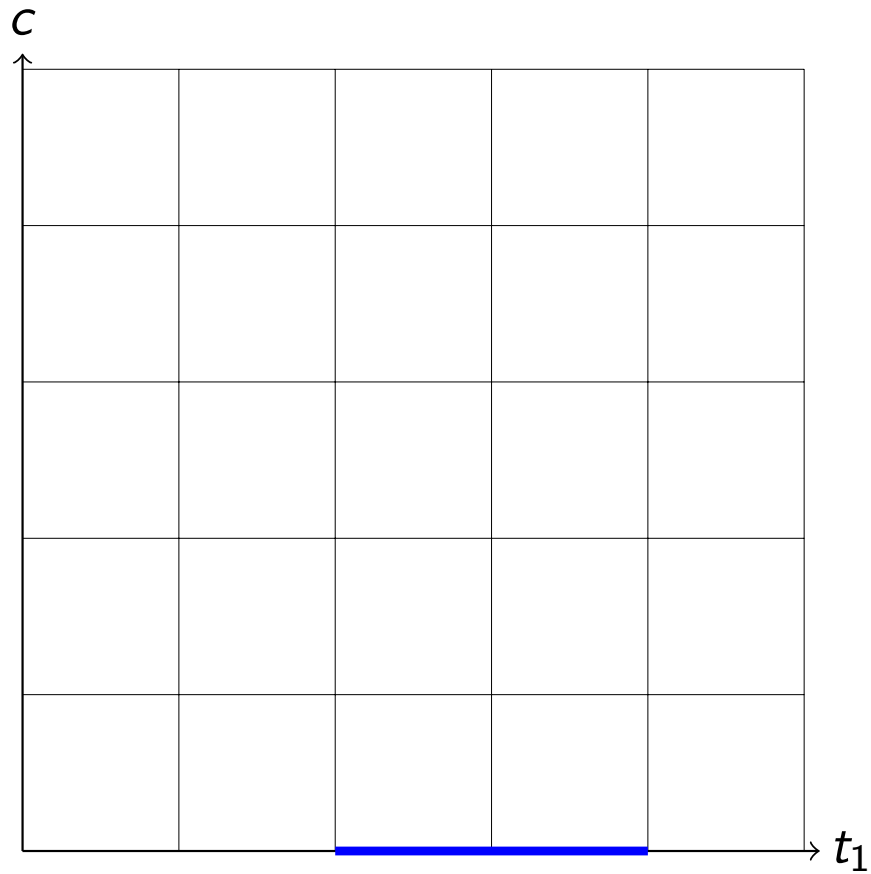
Bounded-Cost Reachability

Can we reach a marking from a given set Goal with a cost less or equal to a given bound $c_{\max}$?

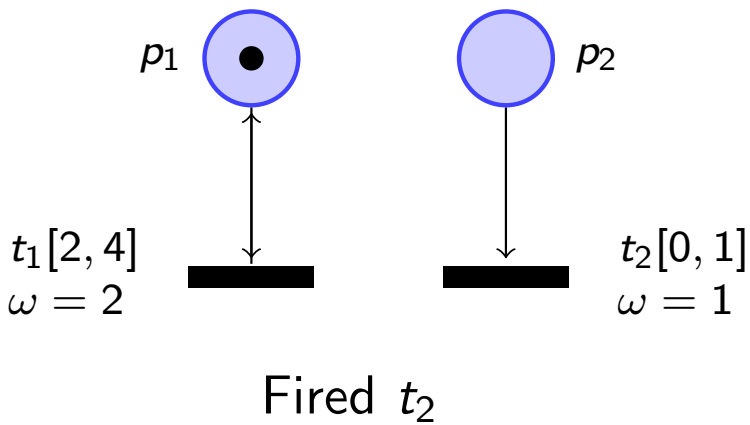
1: $r \leftarrow$ False
2: $\textsc{Passed} \leftarrow \emptyset$
3: $\textsc{Waiting} \leftarrow \{(m_0, D_0)\}$
4: **while** $\textsc{Waiting} \neq \emptyset$ **and not** r **do**
5: select $C_\sigma = (m, D)$ from $\textsc{Waiting}$
6: **if** $m \in$ Goal **and** $\text{cost}(C_\sigma) \leq c_{\max}$ **then**
7: $r \leftarrow$ True
8: **end if**
9: **if** for all $C' \in \textsc{Passed}$, $C_\sigma \not\preccurlyeq C'$ **then**
10: add C_σ to $\textsc{Passed}$
11: for all $t \in \text{firable}(C_\sigma)$, add $C_{\sigma.t}$ to $\textsc{Waiting}$
12: **end if**
13: **end while**
14: **return** r

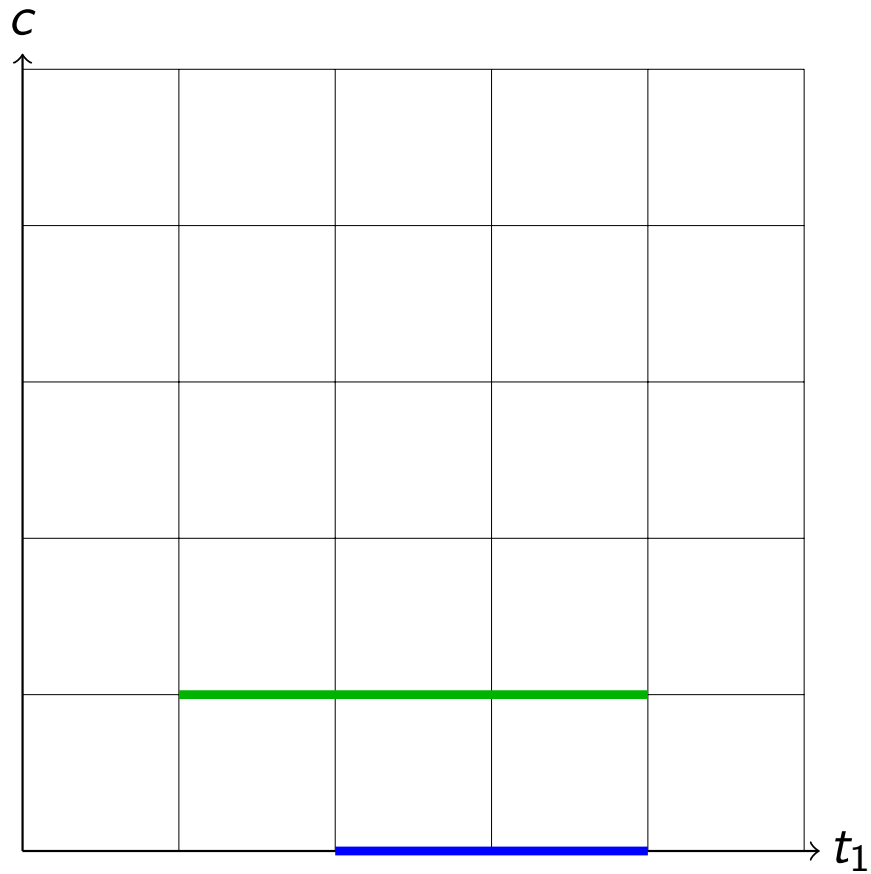
Cost State Class Subsumption: $\preccurlyeq$ [BLP+17]



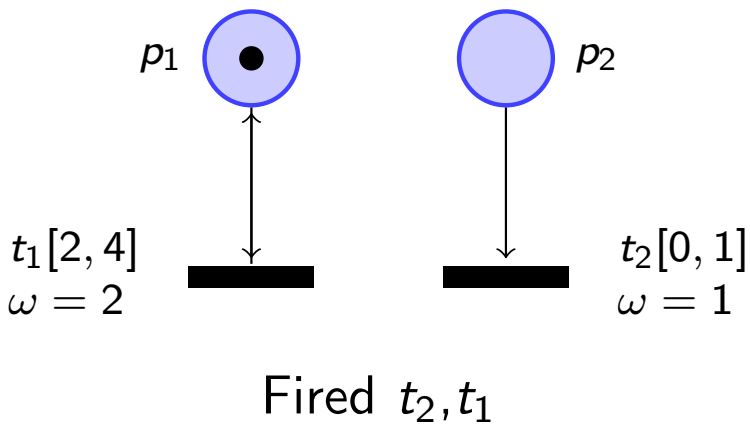


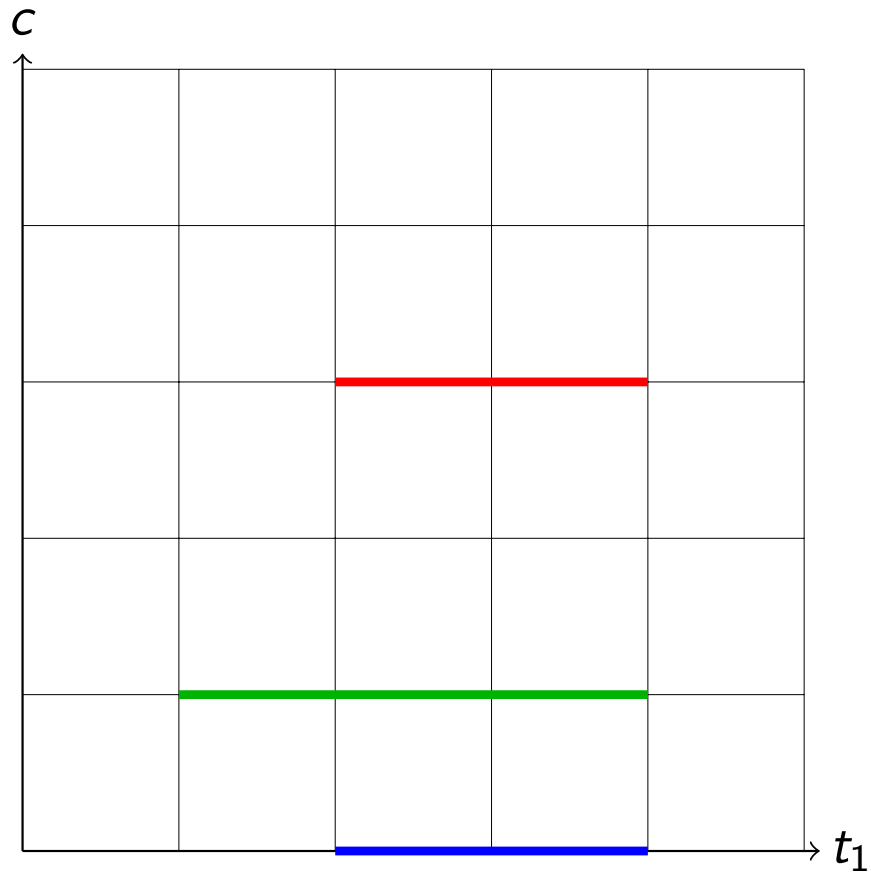
Cost State Class Subsumption: $\preccurlyeq$ [BLP+17]





Cost State Class Subsumption: $\preccurlyeq$ [BLP+17]





Cost State Class Subsumption: $\preccurlyeq$ [BLP+17]

Definition (Cost of a point in a firing domain)

For any cost state class $C = (m, D)$ and any point $\vec{\theta} \in D_{|\theta}$, the optimal cost of $\vec{\theta}$ in D is defined by $\text{cost}_D(\vec{\theta}) = \inf_{(\vec{\theta},c)\in D} c$.

Cost State Class Subsumption: $\preccurlyeq$ [BLP+17]

Definition (Cost of a point in a firing domain)

For any cost state class $C = (m, D)$ and any point $\vec{\theta} \in D_{|\theta}$, the optimal cost of $\vec{\theta}$ in D is defined by $\text{cost}_D(\vec{\theta}) = \inf_{(\vec{\theta},c)\in D} c$.

Definition (Cost State Class Subsumption)

Let $C = (m, D)$ and $C' = (m', D')$ two cost state classes. We say that C is subsumed by C', which we denote by $C \preccurlyeq C'$ iff $m = m'$ and for all $D_{|\theta} \subseteq D'_{|\theta}$, and for all $\vec{\theta} \in D_{|\theta}, \text{cost}_{D'}(\vec{\theta}) \leq \text{cost}_D(\vec{\theta})$.

Cost State Class Subsumption: $\preccurlyeq$ [BLP+17]

Definition (Cost of a point in a firing domain)

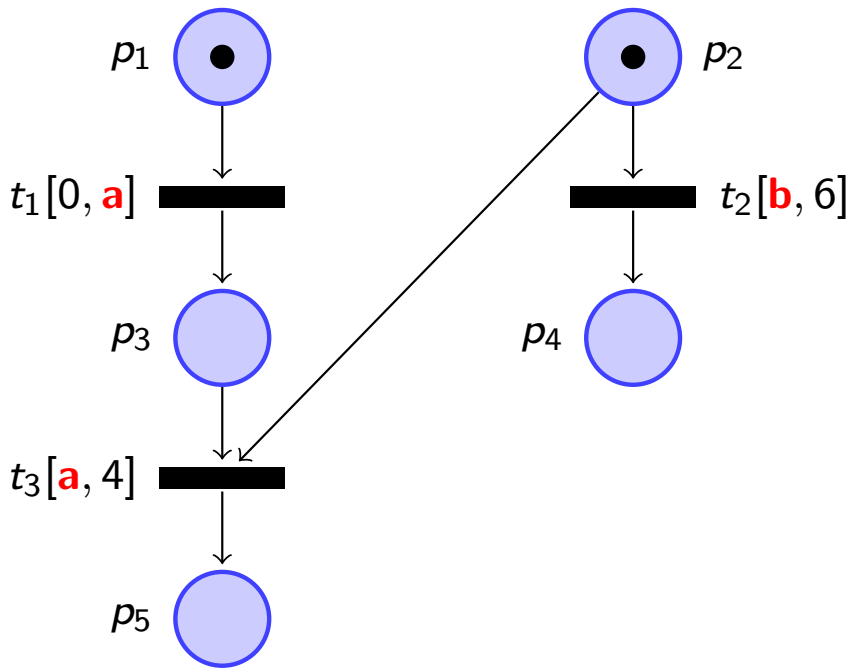
For any cost state class $C = (m, D)$ and any point $\vec{\theta} \in D_{|\theta}$, the optimal cost of $\vec{\theta}$ in D is defined by $\text{cost}_D(\vec{\theta}) = \inf_{(\vec{\theta},c)\in D} c$.

Definition (Cost State Class Subsumption)

Let $C = (m, D)$ and $C' = (m', D')$ two cost state classes. We say that C is subsumed by C', which we denote by $C \preccurlyeq C'$ iff $m = m'$ and for all $D_{|\theta} \subseteq D'_{|\theta}$, and for all $\vec{\theta} \in D_{|\theta}, \text{cost}_{D'}(\vec{\theta}) \leq \text{cost}_D(\vec{\theta})$.

Can be decided using linear programming.

Parametric Time Petri Nets



Parametric Bounded-Cost Problems

- **Existential problem**: given $c_{\max}$ and a marking m, does there exist a value of the parameters such that m is reachable with cost less or equal to $c_{\max}$.
- **Synthesis problem**: find all such parameter values.

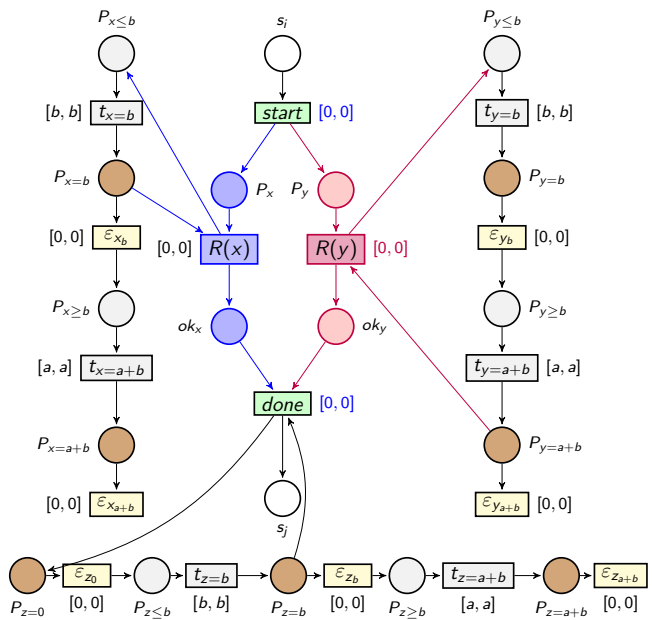
Undecidability of the existential problem

Theorem

The existential problem is undecidable for bounded parametric time Petri nets.

Encode the halting problem of two-counter machines in the existential problem for **time bounded** reachability.

- Start from the encoding of [ALR16];
- Adapt to time Petri nets;
- Make all instructions execute in b time units instead of 1.



Parametric Cost State Classes

- We can compute state classes as before;
- The polyhedra obtained are **parametric DBMs** plus **cost inequalities**;
- Instantiating parameters with integer (or rational) values gives again a **cost state class**.
- Class subsumption is extended naturally:

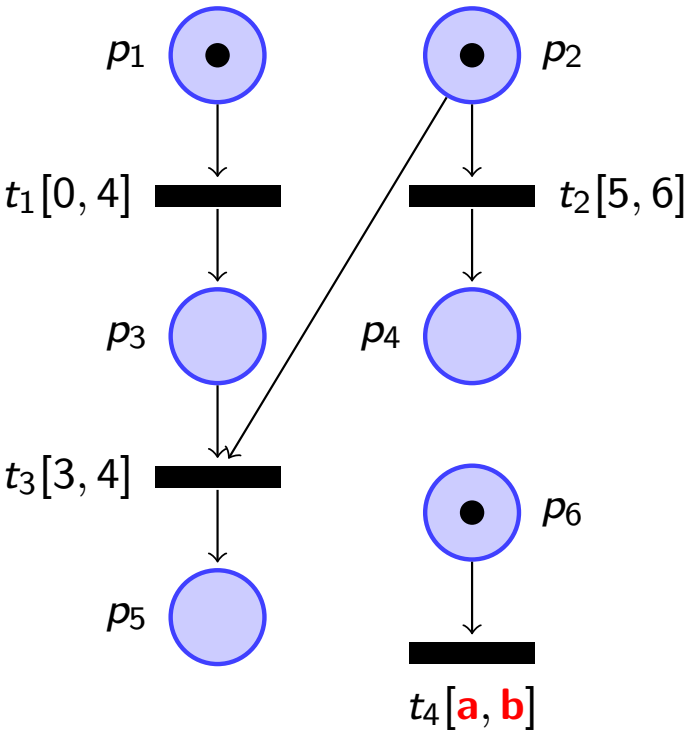
Definition (Parametric Cost State Class Subsumption)

- $\text{cost}_D(\vec{\theta}, \mathbf{v}) = \inf_{(\vec{\theta}, v, c)} c$;
- $C \preccurlyeq C'$ iff $m = m'$ and for all $D_{|\theta \cup \mathbb{P}} \subseteq D'_{|\theta \cup \mathbb{P}}$, and for all $(\vec{\theta}, \mathbf{v}) \in D_{|\theta \cup \mathbb{P}}, \text{cost}_{D'}(\vec{\theta}, \mathbf{v}) \leq \text{cost}_D(\vec{\theta}, \mathbf{v})$.

Parametric Cost State Classes: Example

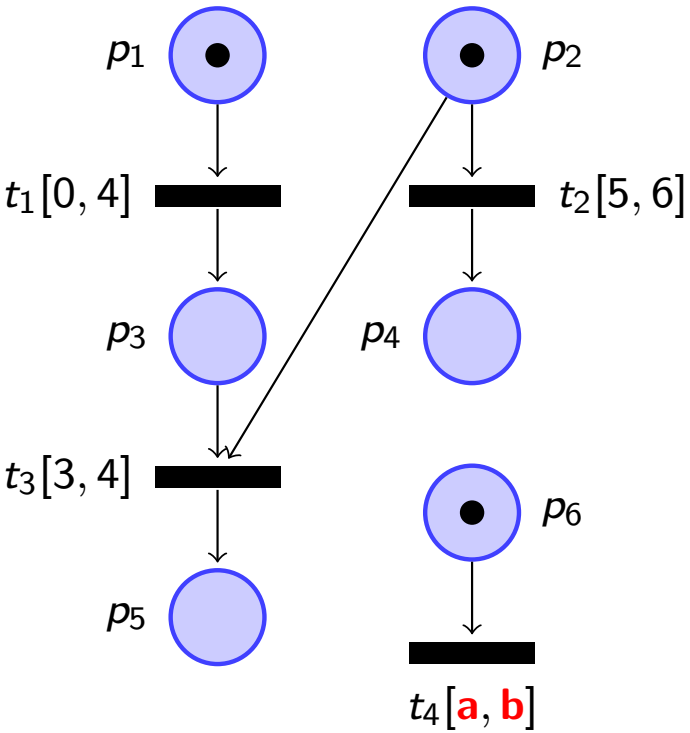
Initially:

$\theta_1 \in [0, 4]$
$\theta_2 \in [5, 6]$
$\theta_4 \in [\mathbf{a}, \mathbf{b}]$
$c \geq 0$



$cr = p_1 + 2p_2$
$\omega(t_1) = 1$

Parametric Cost State Classes: Example



$cr = p_1 + 2p_2$
$\omega(t_1) = 1$

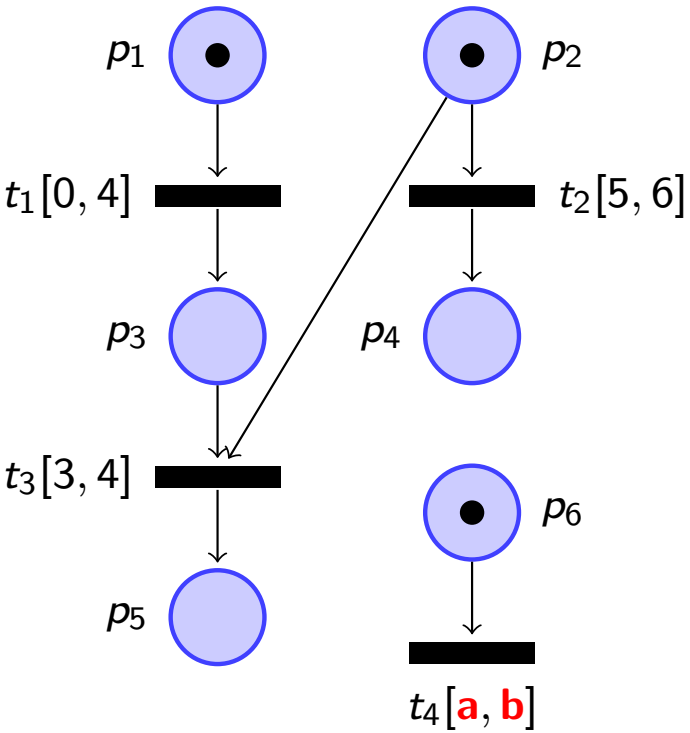
Initially:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [\mathbf{a},\mathbf{b}]$
$c \geq 0$

Fire t_1:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [a,b]$
$c \geq 0$
$\theta_1 \leq \theta_2$
$\theta_1 \leq \theta_4$

Parametric Cost State Classes: Example



$cr = p_1 + 2p_2$
$\omega(t_1) = 1$

Initially:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [\mathbf{a},\mathbf{b}]$
$c \geq 0$

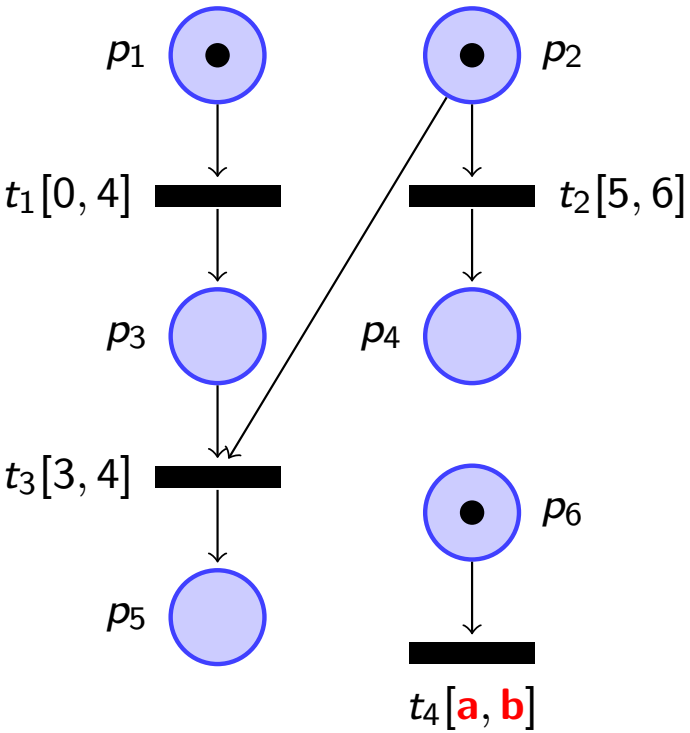
Fire t_1:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [a,b]$
$c \geq 0$
$\theta_1 \leq \theta_2$
$\theta_1 \leq \theta_4$

Change origin:

$\theta_1 \in [0,4]$
$\theta'_2 + \theta_1 \in [5,6]$
$\theta'_4 + \theta_1 \in [a,b]$
$c' - \omega(t_1) - cr(m_0) * \theta_1 \geq 0$
$\theta_1 \leq \theta'_2 + \theta_1$
$\theta_1 \leq \theta'_4 + \theta_1$

Parametric Cost State Classes: Example



$cr = p_1 + 2p_2$
$\omega(t_1) = 1$

Initially:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [\mathbf{a},\mathbf{b}]$
$c \geq 0$

Fire t_1:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [a,b]$
$c \geq 0$
$\theta_1 \leq \theta_2$
$\theta_1 \leq \theta_4$

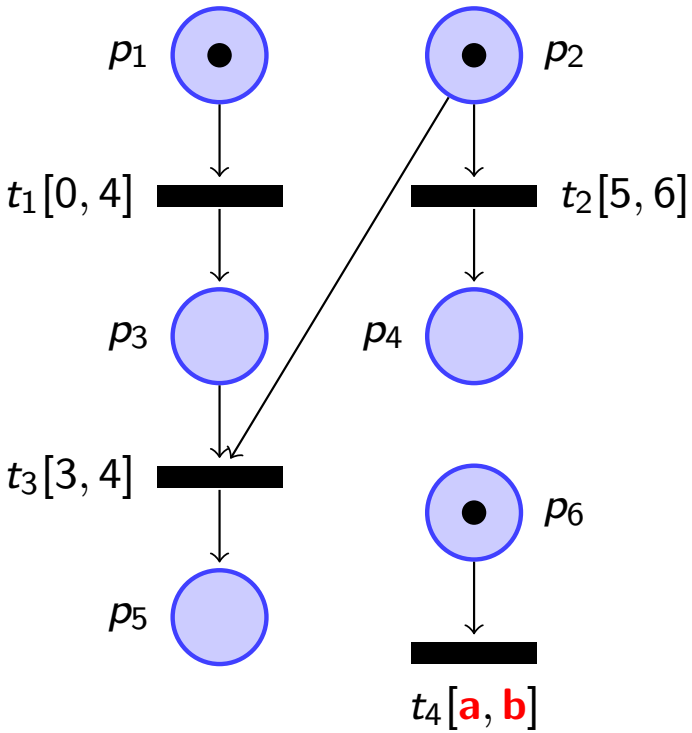
Change origin:

$\theta_1 \in [0,4]$
$\theta'_2 + \theta_1 \in [5,6]$
$\theta'_4 + \theta_1 \in [a,b]$
$c' - \omega(t_1) - cr(m_0) * \theta_1 \geq 0$
$\theta_1 \leq \theta'_2 + \theta_1$
$\theta_1 \leq \theta'_4 + \theta_1$

Eliminate disabled:

$\theta'_2 \in [1,6]$
$\theta'_4 \in [\mathbf{a}-\mathbf{4},\mathbf{b}]$
$\theta'_4 \geq 0$
$\theta'_2 - \theta'_4 \in [\mathbf{5}-\mathbf{b},\mathbf{6}-\mathbf{a}]$
$\mathbf{0} \leq \mathbf{a} \leq \mathbf{b}$
$c \geq 16 - 3t'_2$
$c \geq \mathbf{3}(\mathbf{a}-\mathbf{t'_4})+\mathbf{1}$
$c \geq 1$

Parametric Cost State Classes: Example



$cr = p_1 + 2p_2$
$\omega(t_1) = 1$

Initially:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [\mathbf{a},\mathbf{b}]$
$c \geq 0$

Fire t_1:

$\theta_1 \in [0,4]$
$\theta_2 \in [5,6]$
$\theta_4 \in [a,b]$
$c \geq 0$
$\theta_1 \leq \theta_2$
$\theta_1 \leq \theta_4$

Change origin:

$\theta_1 \in [0,4]$
$\theta'_2 + \theta_1 \in [5,6]$
$\theta'_4 + \theta_1 \in [a,b]$
$c' - \omega(t_1) - cr(m_0) * \theta_1 \geq 0$
$\theta_1 \leq \theta'_2 + \theta_1$
$\theta_1 \leq \theta'_4 + \theta_1$

Eliminate disabled:

$\theta'_2 \in [1,6]$
$\theta'_4 \in [\mathbf{a}-\mathbf{4},\mathbf{b}]$
$\theta'_4 \geq 0$
$\theta'_2 - \theta'_4 \in [\mathbf{5}-\mathbf{b},\mathbf{6}-\mathbf{a}]$
$\mathbf{0} \leq \mathbf{a} \leq \mathbf{b}$
$c \geq 16 - 3t'_2$
$c \geq \mathbf{3}(\mathbf{a}-\mathbf{t'_4})+\mathbf{1}$
$c \geq 1$

Add newly enabled:

$\ldots$
$\theta_3 \in [3,4]$
$\ldots$

Symbolic Semi-algorithm for Bounded-Cost Reachability

1: PolyRes $\leftarrow \emptyset$
2: $\textsc{Passed} \leftarrow \emptyset$
3: $\textsc{Waiting} \leftarrow \{(m_0, D_0)\}$
4: **while** $\textsc{Waiting} \neq \emptyset$ **do**
5: select $C_\sigma = (m, D)$ from $\textsc{Waiting}$
6: **if** $m \in$ Goal **then**
7: PolyRes $\leftarrow$ PolyRes $\cup \left(\mathbf{D} \cap (\mathbf{c} \leq \mathbf{c}_{\mathsf{max}})\right)_{\downarrow \mathbb{P}}$
8: **end if**
9: **if** for all $C' \in \textsc{Passed}$, $C_\sigma \not\preccurlyeq C'$ **then**
10: add C_σ to $\textsc{Passed}$
11: for all $t \in$ firable(C_σ), add $C_{\sigma.t}$ to $\textsc{Waiting}$
12: **end if**
13: **end while**
14: **return** PolyRes

Symbolic Parameter Synthesis for Bounded-Cost Reachability

- When it terminates, the previous algorithm is **sound** and **complete**:

Lemma

For all classes $C_\sigma = (m, D)$, $(\vec{\theta}, c, v) \in D$ *if and only if there exists a run* ρ *in* $v(\mathcal{N})$, *and* $I : \mathsf{en}(m) \to \mathcal{I}(\mathbb{Q}_{\geq 0})$, *such that* $\mathsf{sequence}(\rho) = \sigma$, $(m, I, c) = \mathsf{last}(\rho)$, *and* $\vec{\theta} \in I$.

Lemma

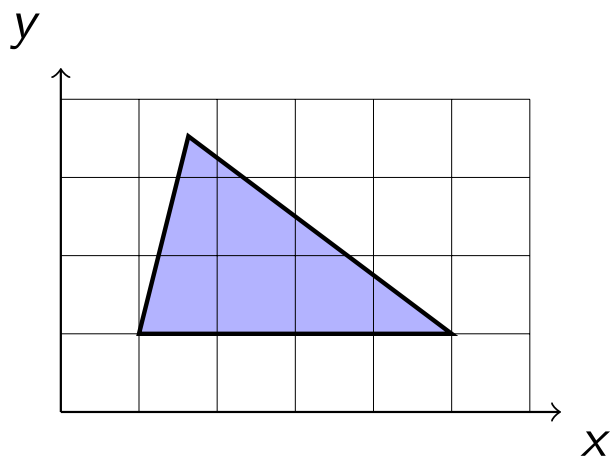
Let C_{σ_1} *and* C_{σ_2} *be two state classes such that* $C_{\sigma_1} \preccurlyeq C_{\sigma_2}$.
If a transition sequence σ *is firable from* C_{σ_1}, *it is also firable from* C_{σ_2} *and* $\mathsf{cost}(C_{\sigma_1.\sigma}) \geq \mathsf{cost}(C_{\sigma_2.\sigma})$.

- Termination is **not** guaranteed;

Integer hull

We use the **integer hull** trick to:

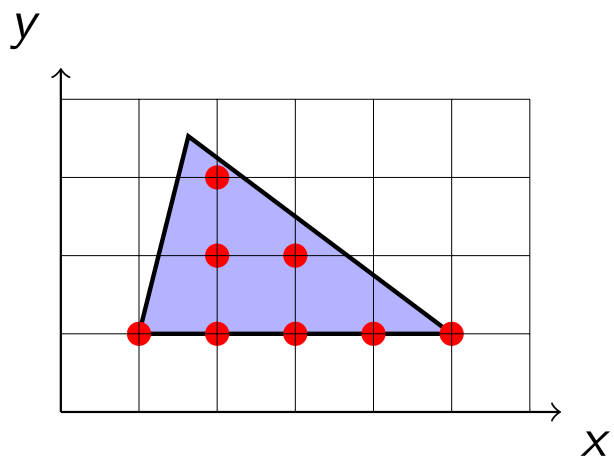
1. make it compute **integer** parameter valuations;
2. ensure termination when parameters are **bounded**.



Integer hull

We use the **integer hull** trick to:

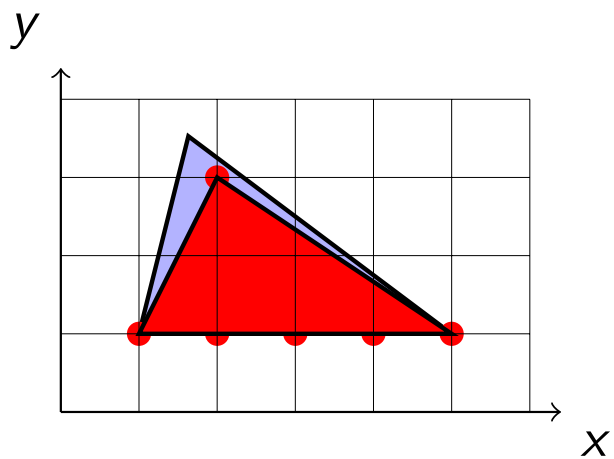
1. make it compute **integer** parameter valuations;
2. ensure termination when parameters are **bounded**.



Integer hull

We use the **integer hull** trick to:

1. make it compute **integer** parameter valuations;
2. ensure termination when parameters are **bounded**.



Integer Parameter Synthesis for Bounded-Cost Reachability

1: PolyRes $\leftarrow \emptyset$
2: $\textsc{Passed} \leftarrow \emptyset$
3: $\textsc{Waiting} \leftarrow \{(m_0, D_0)\}$
4: **while** $\textsc{Waiting} \neq \emptyset$ **do**
5: select $C_\sigma = (m, D)$ from $\textsc{Waiting}$
6: **if** $m \in$ Goal **then**
7: PolyRes $\leftarrow$ PolyRes $\cup \left(\mathsf{IH}(D) \cap (c \leq c_{\max})\right)_{|\mathbb{P}}$
8: **end if**
9: **if** for all $C' \in \textsc{Passed}$, $\mathsf{IH}(C_\sigma) \not\preccurlyeq \mathsf{IH}(C')$ **then**
10: add C_σ to $\textsc{Passed}$
11: for all $t \in$ firable$(\mathsf{IH}(C_\sigma))$, add $C_{\sigma.t}$ to $\textsc{Waiting}$
12: **end if**
13: **end while**
14: **return** PolyRes

Integer Parameter Synthesis for Bounded-Cost Reachability

- When it terminates, the previous algorithm is **sound** and **complete** for **integer** parameter valuations;

Lemma

If v is an **integer** *parameter valuation, then for all classes $C_\sigma = (m, D)$, $(\vec{\theta}, c, v) \in \mathsf{IH}(D)$ if and only if there exists a run ρ in $v(\mathcal{N})$, and $I : \mathsf{en}(m) \to \mathcal{I}(\mathbb{Q}_{\geq 0})$, such that $\mathsf{sequence}(\rho) = \sigma$, $(m, I, c) = \mathsf{last}(\rho)$, and $\vec{\theta} \in I$.*

Lemma

Let C_{σ_1} and C_{σ_2} be two state classes such that $\mathsf{IH}(C_{\sigma_1}) \preccurlyeq \mathsf{IH}(C_{\sigma_2})$. If a transition sequence σ is $\mathbb{N}^{\mathbb{P}}$-firable from C_{σ_1} it is also $\mathbb{N}^{\mathbb{P}}$-firable from C_{σ_2} and $\mathsf{cost}_{\mathbb{N}}(C_{\sigma_1.\sigma}) \geq \mathsf{cost}_{\mathbb{N}}(C_{\sigma_2.\sigma})$.

- Termination is **still not** guaranteed, except when parameters are **bounded**;
- When parameters are bounded, $\succcurlyeq$ is again a well-quasiorder;
- Integer hull can also be computed as part of the successor class computation.

Implementation

- The parameter synthesis symbolic algorithm is implemented in Roméo:
  - for rational parameters;
  - for integer parameters.
- **Linear expressions** on parameters can be used as part of the timing intervals;
- **Prior constraints** can be given on the parameters.

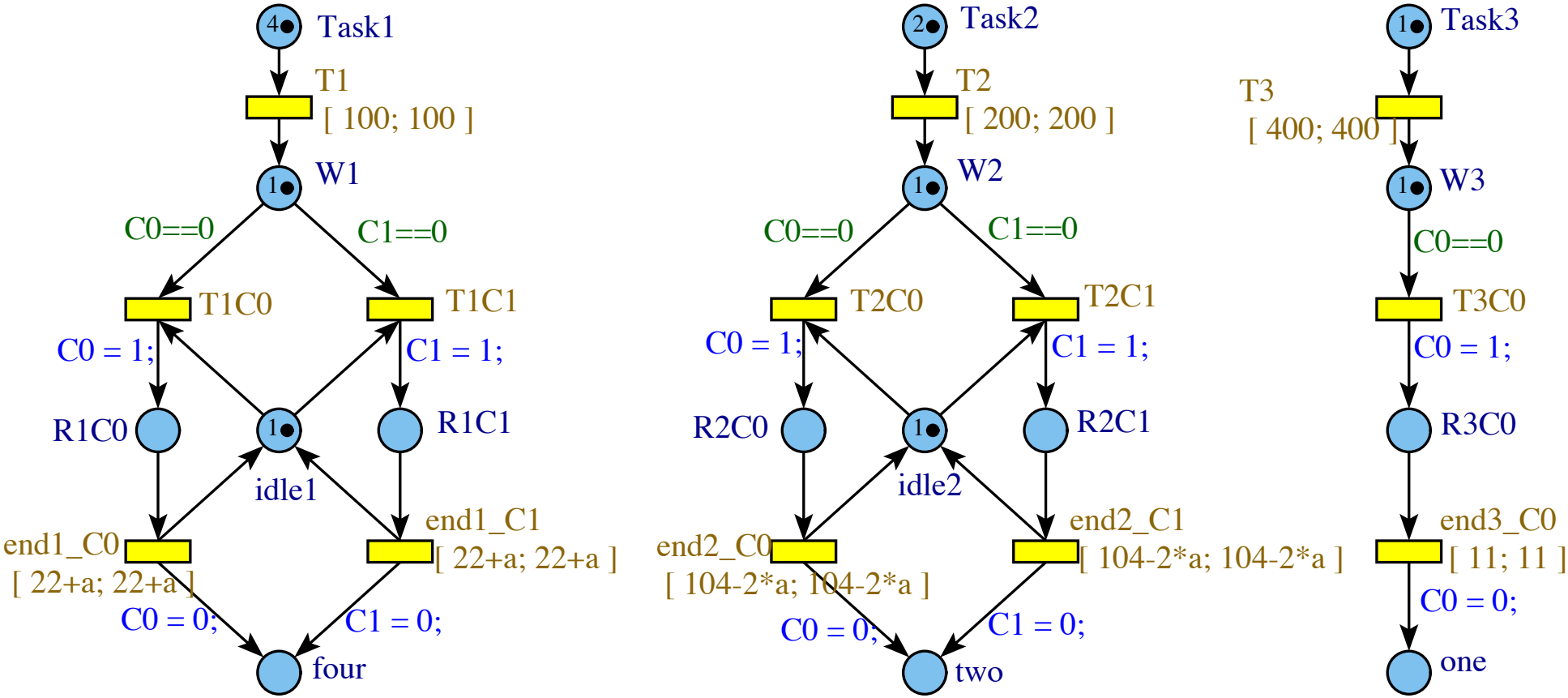
An AUTOSAR Runnable Mapping Problem

- Runnables are the high-level functions of an AUTOSAR application;
- They are mapped on **tasks** for execution;
- Tasks may contain code from unrelated runnables.

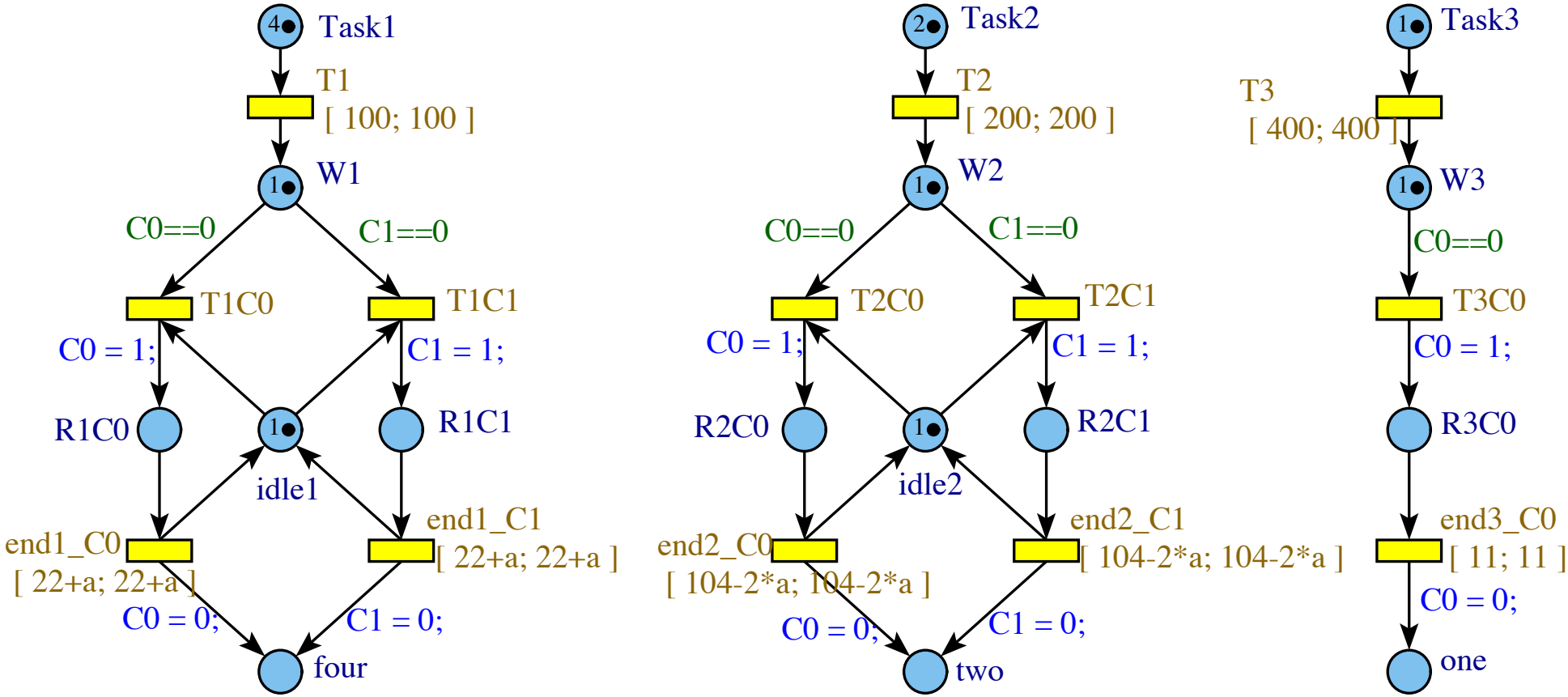
An AUTOSAR Runnable Mapping Problem

- We have some prior mapping on 3 tasks and a new runnable to map;
- We use a dual-core ECU and try to minimize its **energy** consumption.
- It is more efficient when both cores are busy (or both are idle!);
- We must also maintain **schedulability**.

An AUTOSAR Runnable Mapping Problem

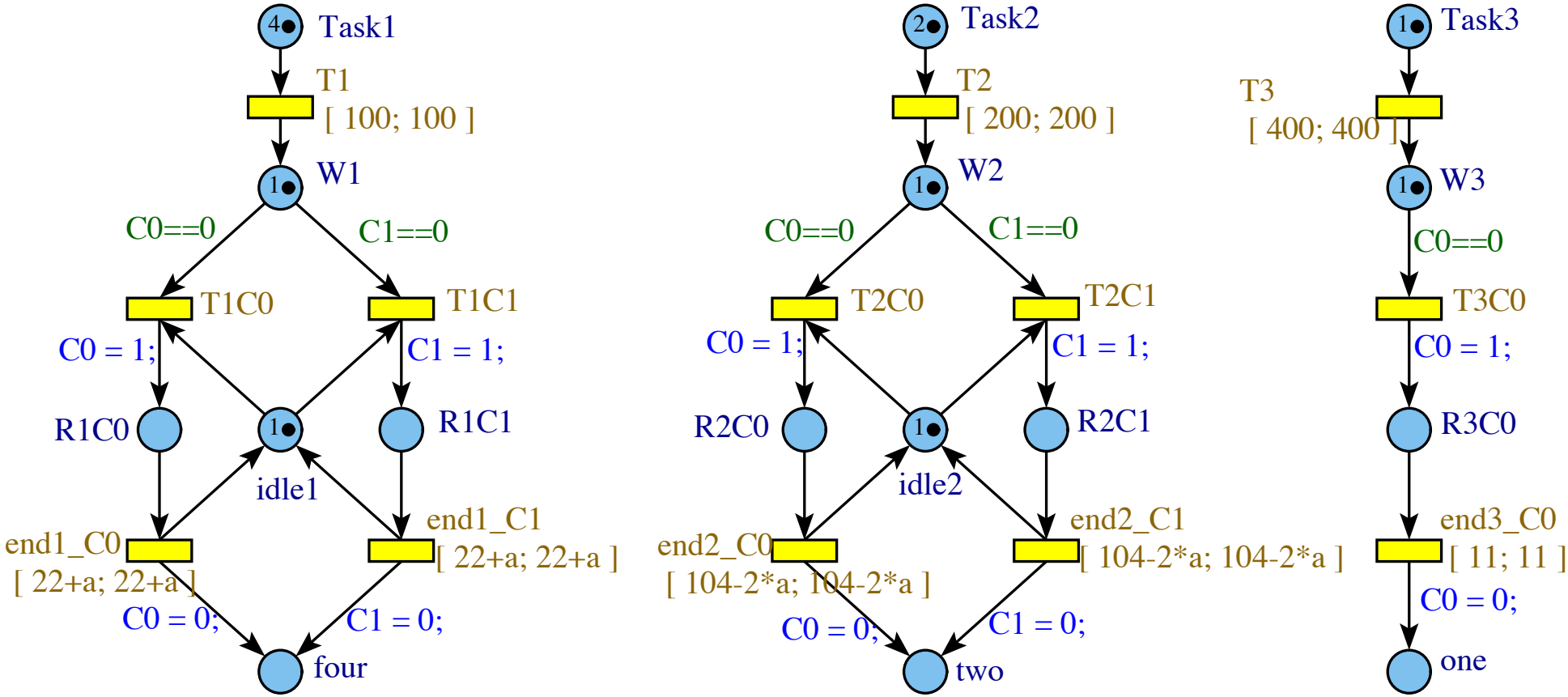


An AUTOSAR Runnable Mapping Problem



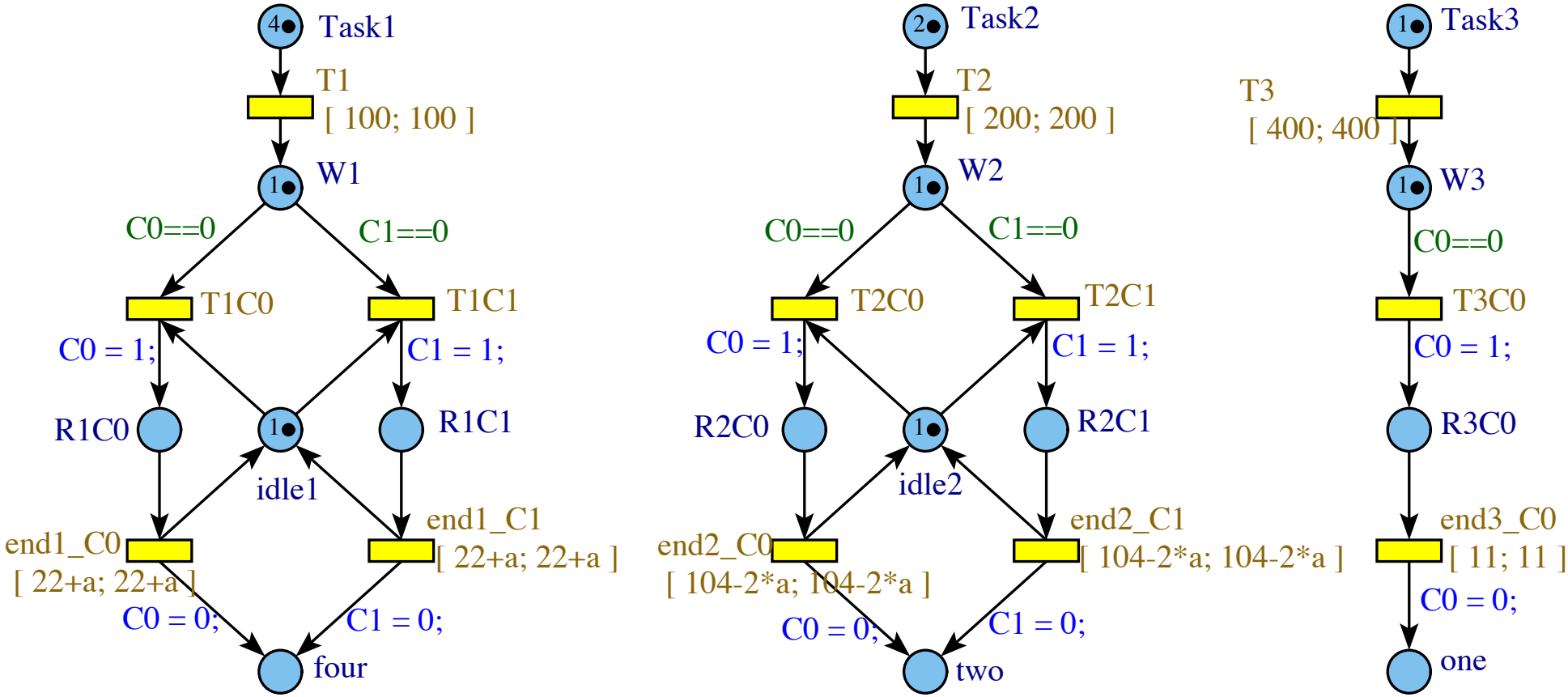
- cost rate: $(R1C0 + R1C1) * W1 * 100 + (R2C0 + R2C1) * W2 * 100 + R3C0 * W3 * 100 + 2 * oneCore(C0, C1) + C0 * C1 * 3$

An AUTOSAR Runnable Mapping Problem



- cost rate: $(R1C0 + R1C1) * W1 * 100 + (R2C0 + R2C1) * W2 * 100 + R3C0 * W3 * 100 + 2 * oneCore(C0, C1) + C0 * C1 * 3$
- we know a priori that we can achieve a cost of less or equal to 466.
- property: $EF(one == 1 \text{ and } two == 2 \text{ and } four == 4 \text{ and } cost \leq 466)$;

An AUTOSAR Runnable Mapping Problem



- cost rate: $(R1C0 + R1C1) * W1 * 100 + (R2C0 + R2C1) * W2 * 100 + R3C0 * W3 * 100 + 2 * oneCore(C0, C1) + C0 * C1 * 3$
- we know a priori that we can achieve a cost of less or equal to 466.
- property: $EF(one == 1 \text{ and } two == 2 \text{ and } four == 4 \text{ and } cost \leq 466)$;
- answer: $a \in [\frac{49}{4}, \frac{71}{4}]$ (real parameters), i.e., $a \in [13, 17]$ (integer parameters).

An AUTOSAR Runnable Mapping Problem

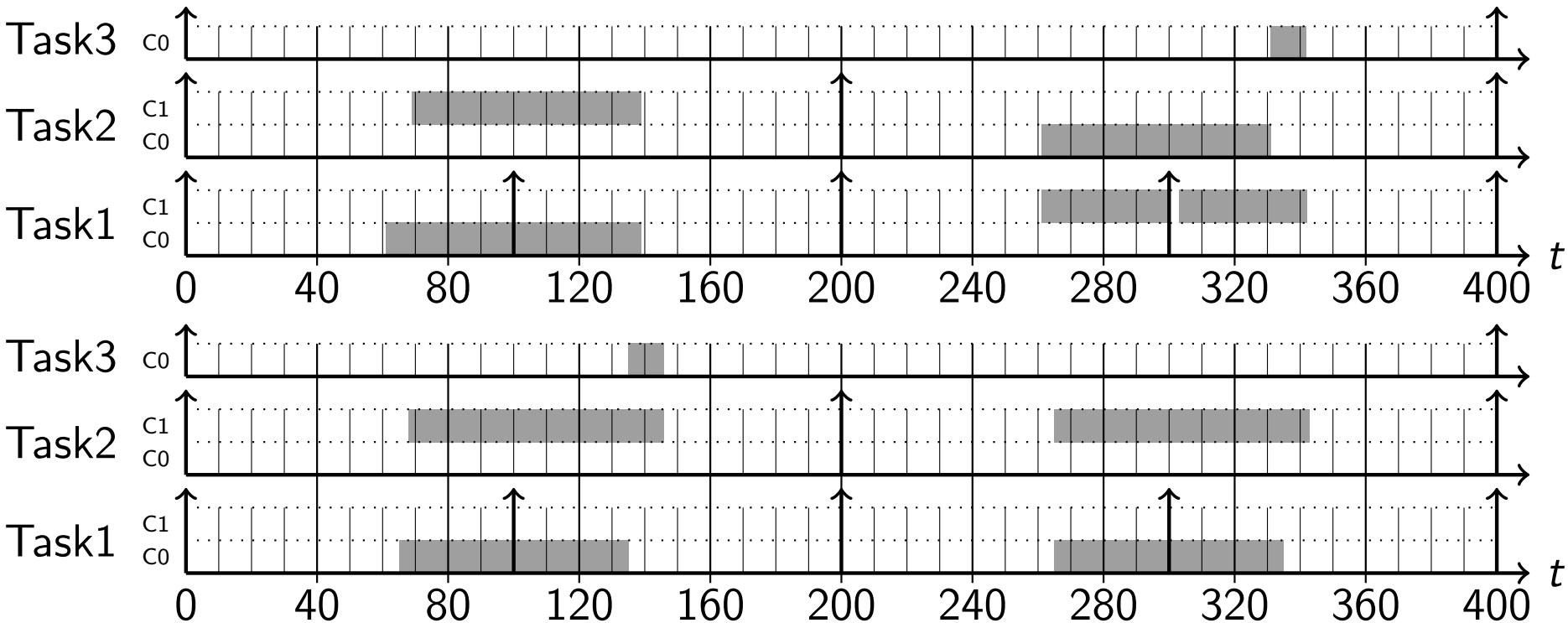


Figure: Gantt charts for $a = 17$ (above) and $a = 13$ (below)

Conclusion and Perspectives

- Summary:
  - We can extend state classes for **parameter synthesis** for bounded-cost reachability;
  - The **integer hull** trick allows for a terminating **symbolic** algorithm for bounded integer parameters;
  - The techniques are **implemented** in the freely available Roméo tool: `http://romeo.rts-software.org/`

Conclusion and Perspectives

- Summary:
  - We can extend state classes for **parameter synthesis** for bounded-cost reachability;
  - The **integer hull** trick allows for a terminating **symbolic** algorithm for bounded integer parameters;
  - The techniques are **implemented** in the freely available Roméo tool: `http://romeo.rts-software.org/`
- Future work:
  - Optimal cost as a function of parameters;
  - Parameter synthesis in **parametric cost** timed models;
  - Integer hull for undecidable non-parametric cost problems (control, upper bound "hard" constraints).

References I


Étienne André, Didier Lime, and Olivier H. Roux.
Decision problems for parametric timed automata.
In Kazuhiro Ogata, Mark Lawford, and Shaoying Liu, editors, *18th International Conference on Formal Engineering Methods (ICFEM 2016)*, volume 10009 of *Lecture Notes in Computer Science*, pages 400–416, Tokyo, Japan, November 2016. Springer.


B. Berthomieu and M. Diaz.
Modeling and verification of time dependent systems using time Petri nets.
IEEE trans. on soft. eng., 17(3):259–273, 1991.


Hanifa Boucheneb, Didier Lime, Baptiste Parquier, Olivier H. Roux, and Charlotte Seidner.
Optimal reachability in cost time Petri nets.
In *FORMATS'17, Berlin, Germany*, LNCS, 2017.