Model Checking Process Algebra of Communicating Resources for Real-time Systems

A.Jalil Boudjadar, Jin Hyun Kim,
Kim. G. Larsen, Ulrik Niman
CISS, Alborg University
{jalil, jin, klg, ulrik}@cs.aau.dk

Contents

- Introduction
- PACoR (Process Algebra of Communicating Resources)
  - Syntax
  - Timed Operational Semantics
- Analysis
  - gPACoR
  - Parameterized Stopwatch Automata
- Examples
- Conclusions

Context

Timing requirements

"The brake should react within less than 500ms"

Resource constraints

"The braking process runs on one of CPU1 or CPU2 at priority 2"

How can we make the system satisfy not only **timing requirements** but also **resource constraints**?

Context (cont'd)

- **Process Algebra**, e.g. CCS, CSP,
  - Rigorous analysis method for concurrent behaviors and communicating systems
- For real-time systems (RTS),
  - ACSR, mCRL2, Timed CSP, tock CSP…

Context (cont'd)

- Algebra of Communicating Shared Resources (ACSR)
  - Expressive to capture **resource-constrained aspect** of RTS as well as timing requirements,

- However, ACSR
  - Not supported by advanced analysis methods.

Contributions

- A new process algebra, called PACoR, for RTS, extending ACSR,
  - Oriented to resource-constrained aspects of RTS,
- Translation rules from PACoR to Uppaal models

Contributions (cont'd)

- PACoR
  - Enables to use **both** **symbolic** and **statistical** model checker using **the same models.**
  - to answer **qualitative** and **quantitative** questions, such as **schedulability** and **worst-case response time**, and so on,

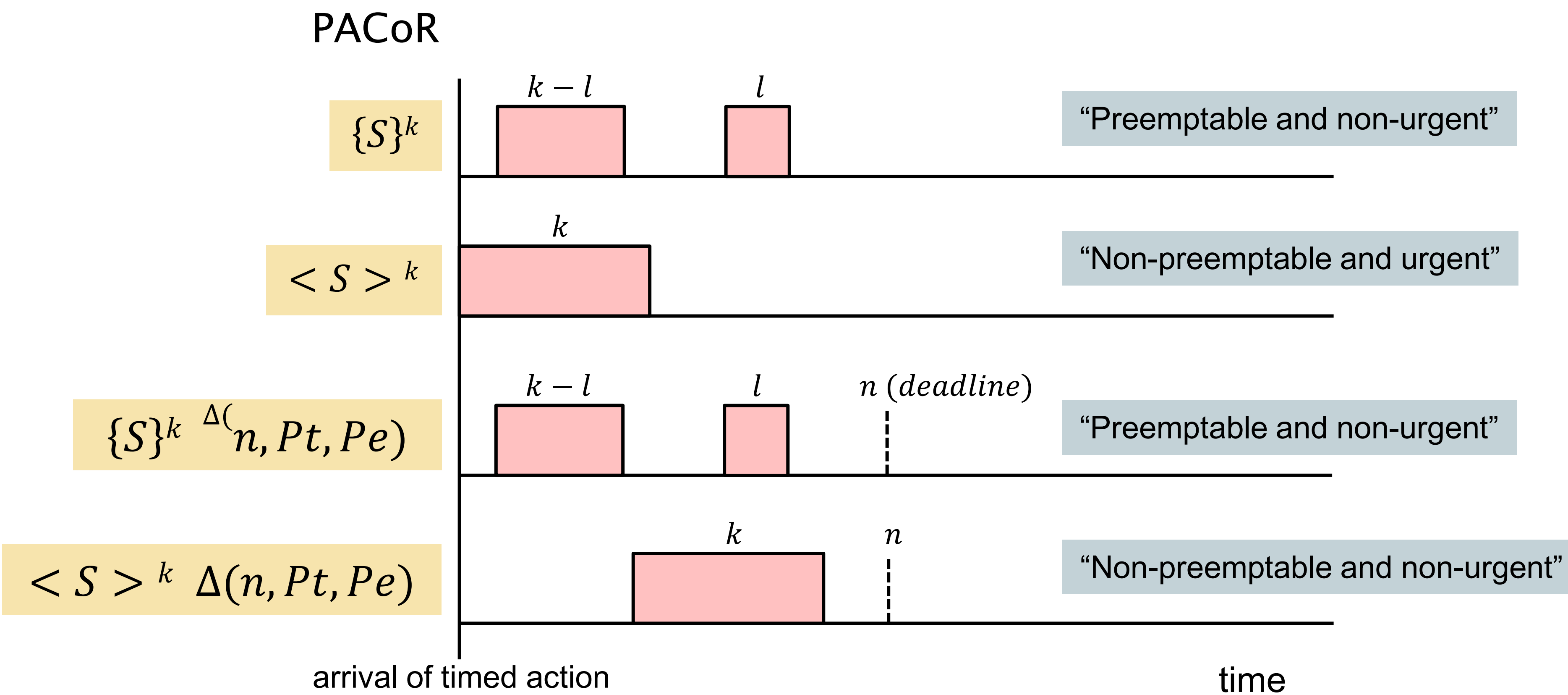
PACoR

PACoR:
Process Algebra of Communicating Resources

:scheduled by priorities for shared resources

:Instantaneous

Timed Actions



PACoR vs ACSR

	Attribute	ACSR	PACoR
Timed Action	Non-preemptable and Urgent	Yes	Yes
	Preemptable and Non-urgent	No	**Yes**
	Non-Preemptable and Non-urgent	No	**Yes**
Process Creation & Termination		Static	Static & **Dynamic**
Verification		No	**Uppaal & Uppaal SMC**

ACSR: ACSR Dense-Time

PACoR : Syntax

Deadlock | No action | Timed action | Event action | Choice

Parallel | Close | Restrict | Recursive

$$P ::= NIL \mid DONE \mid A : P \mid E.P \mid P_1 + P_2$$

$$\mid \quad P_1 \parallel P_2 \mid [P] \mid P \backslash F \mid rec X.P$$

Scoped timed action | Timing requirement

$$A ::= A^\delta \triangle (n, P_t, P_c) \mid \mathcal{A}^\delta$$

Scoped event action

$$E ::= \mathcal{E} \nabla (n, P_t, P_c) \mid \mathcal{E}$$

Preemptive timed action | Non-preemptive timed action

$$\mathcal{A} ::= \{S\} \mid \langle S \rangle$$

Event | tau

$$\mathcal{E} ::= e \mid \bar{e} \mid \tau$$

Resource | Priority

$$S ::= \emptyset \mid (r, pri).S$$

Lower-bound | Upper-bound

$$\delta ::= [l; u]$$

Example

- Two tasks, T_1 and T_2, execute jobs using shared resource *cpu* with the following attributes:
  - T_1 (5, 2, 5)
  - T_2 (10, 3, 7)

* T(period, wcet, deadline)

Example

Process composition

Restricted events

$$System \stackrel{def}{=} [(\ D_1 \ || \ D_2 \ || \ T_1 \ || \ T_2 \ )\backslash\{s_1,\ s_2\}]$$

5 time units delay

$$D_1 \stackrel{def}{=} \emptyset^5 : \bar{s_1}.D_1$$

$$D_2 \stackrel{def}{=} \emptyset^{10} : \bar{s_2}.D_2$$

C_1 is delayed until s1 arrives

$$T_1 \stackrel{def}{=} s_1\nabla(\infty, NIL, NIL).C_1$$

$$T_2 \stackrel{def}{=}$$

CPU is used for 2 time units at priority 3 until the deadline of 5 time units elapses

$$C_1 \stackrel{def}{=} \{(cpu, 3)\}^{[2,2]} \triangle(5, NI$$

C1 has priority over C2 when they want cpu at the same time for the higher priority

$$C_2 \stackrel{def}{=} \{(cpu, 2)\}^{[3,3]} \triangle(7, NI$$

CPU is used for 3 time units at priority 2 until the deadline of 5 time units elapses

PACoR: Semantics

- Timed Transition System (TTS):

A timed transition system over an alphabet Σ is a tuple $<\boldsymbol{S}, \boldsymbol{S^0}, \rightarrow>$ where S is a set of states, $S^0 \subseteq S$ is the set of initial states and $\rightarrow \subseteq \boldsymbol{S} \times \boldsymbol{\Sigma} \times \{\boldsymbol{\tau}\} \cup \mathbb{R}_{\geq \mathbf{0}} \times \boldsymbol{S}$ is the transition relation.

PACoR : Semantics

- Event Actions

$$P \stackrel{def}{=} E.P$$

$$\frac{Ident(P) := Ident(E.P)}{\langle E.P, x, \mathtt{ID} \rangle \xrightarrow{\mathcal{E}} \langle P, x, \mathtt{ID} \rangle}$$

PACoR : Timed Operational Semantics

- Timed Actions

$$P \stackrel{def}{=} \mathcal{A}^{[l,u]} : P$$

$$\frac{m \in \mathbb{R}_{\geq 0},\ l \leq m \leq u,\ Ident(P) := Ident(\mathcal{A}^{[l,u]} : P)}{\langle \mathcal{A}^{[l,u]} : P, x, \mathtt{ID} \rangle \xrightarrow{\mathcal{A}} \langle P, x + m, \mathtt{ID} \rangle}$$

$Note$: x is the global clock

PACoR : Timed Operational Semantics

- E-choiceL

$$P \stackrel{def}{=} E.P_1 + P_2$$

$$\frac{Ident(P_1) := Ident(E.P_1 + P_2)}{\langle E.P_1 + P_2, x, \mathtt{ID}\rangle \xrightarrow{E} \langle P_1, x, \mathtt{ID}\rangle}$$

PACoR : Timed Operational Semantics

- A–choiceL

$$P \stackrel{def}{=} \mathcal{A} : P_1 + P_2$$

$$\frac{m \in \mathbb{R}_{\geq 0},\ \ Ident(P_1) := Ident(\mathcal{A}^{[l,u]} : P_1 + P_2)}{\langle \mathcal{A}^{[l,u]} : P_1 + P_2, x, \mathtt{ID} \rangle \xrightarrow{\mathcal{A}} \langle P_1, x + m, \mathtt{ID} \rangle}$$

PACoR : Timed Operational Semantics

- E-sync

$$P \overset{def}{=} E.P_1 \| \overline{E}.P_2$$

$$\frac{-}{\langle E.P_1 \| \overline{E}.P_2, x, \mathtt{ID} \rangle \xrightarrow{\tau} \langle P_1 \| P_2, x, \mathtt{ID} \rangle}$$

PACoR : Timed Operational Semantics

- E-async

$$P \stackrel{def}{=} E.P_1 \| P_2$$

$$\frac{-}{\langle E.P_1 \| P_2, x, \mathtt{ID} \rangle \xrightarrow{E} \langle P_1 \| P_2, x, \mathtt{ID} \rangle}$$

PACoR : Timed Operational Semantics

- A-sync

$$P \overset{def}{=} \mathcal{A}_1 : P_1 \| \mathcal{A}_2 : P_1, \quad \rho(\mathcal{A}_1) \cap \rho(\mathcal{A}_1) = \emptyset$$

$$\frac{\rho(\mathcal{A}_1) \cap \rho(\mathcal{A}_2) = \emptyset, m = max_t(\mathcal{A}_1, \mathcal{A}_2)}{\langle \mathcal{A}_1 : P_1 \| \mathcal{A}_2 : P_2, x, \mathtt{ID} \rangle \xrightarrow{\mathcal{A}_1 \cup \mathcal{A}_2} \langle P_1 \| P_2, x + m, \mathtt{ID} \rangle}$$

$Note$: $max_t()$ is the maximum execution time of an action (set of actions) that may also include delays caused by preemption, depending on the action deadline

PACoR : Timed Operational Semantics

- A-async

$$P \stackrel{def}{=} \mathcal{A}_1 : P_1 \| \mathcal{A}_2 : P_2, \quad \rho(\mathcal{A}_1) \cap \rho(\mathcal{A}_1) \neq \emptyset$$

$$\{(r,1)\} : P_1 \parallel \{(r,2)\} : \ P_2$$

PACoR : Semantics

- Priority relation:

Given two actions α and β we say that β has priority over α denoted by $\alpha \prec \beta$, if one of the following cases holds:

1) $\alpha \in DR$ and $\beta \in DE$

2) Both α and β are actions in D_R, where

$$\forall r \in \rho(\beta) \cap \rho(\alpha),\ (r,p) \in \alpha \wedge (r,p') \in \beta \Rightarrow p < p'$$

PACoR : Semantics

- Priority relation

$$\{(\mathrm{r}_1, 2)\} \prec \{(r_1, 7)\}$$

$$\{\underline{(\mathrm{r}_1, 2)}, (r_2, 0)\} \prec \{\underline{(r_1, 7)}\}$$

$$\{(\mathrm{r}_1, 2), \underline{(r_2, 5)}\} \prec \{\underline{(r_2, 7)}, (r_3, 5)\}$$

$$\{\underline{(\mathrm{r}_1, 2)}, \underline{(r_2, 5)}\} \not\prec \{\underline{(r_1, 7)}, \underline{(r_2, 3)}\}$$

$$\{\underline{(\mathrm{r}_1, 3)}, \underline{(r_2, 3)}, \underline{(r_3, 1)}\} \not\prec \{\underline{(r_1, 1)}, \underline{(r_2, 1)}, \underline{(r_3, 1)}\}$$

PACoR : Timed Operational Semantics

- A-async

$$P \overset{def}{=} \mathcal{A}_1 : P_1 \| \mathcal{A}_2 : P_2, \quad \rho(\mathcal{A}_1) \cap \rho(\mathcal{A}_1) \neq \emptyset$$

$$\frac{\rho(\mathcal{A}_1) \cap \rho(\mathcal{A}_2) \neq \emptyset, \neg(\mathcal{A}_1 \prec \mathcal{A}_2)}{\langle \mathcal{A}_1 : P_1 \| \mathcal{A}_2 : P_2, x, \mathtt{ID} \rangle \xrightarrow{\mathcal{A}_1} \langle P_1 \| \mathcal{A}_2 : P_2, x + m, \mathtt{ID} \rangle}$$

Example

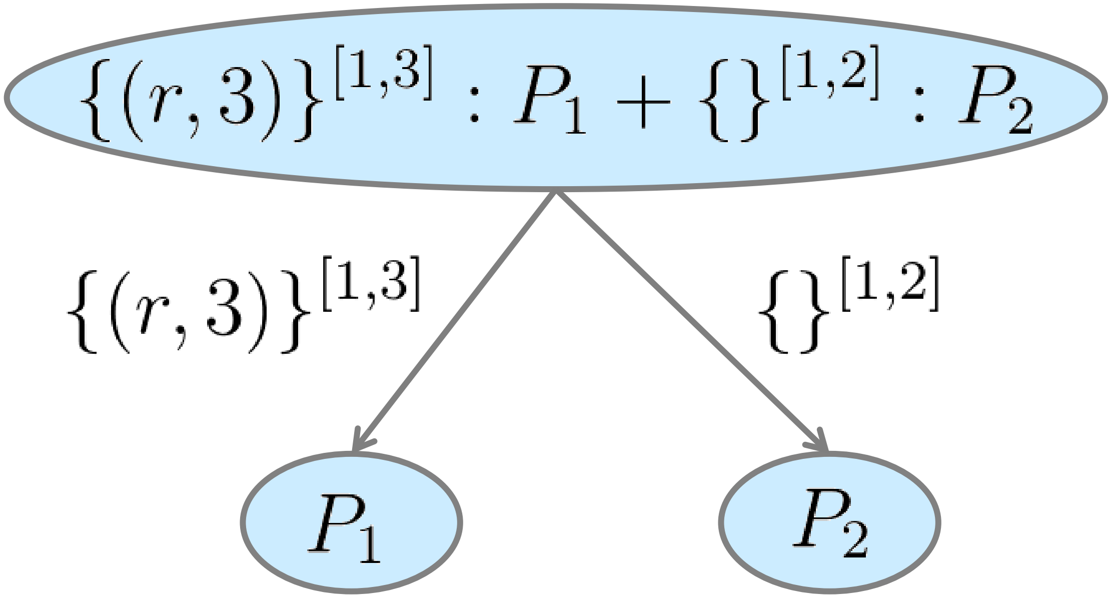
- A-async

$$P \stackrel{def}{=} \mathcal{A}_1 : P_1 \| \mathcal{A}_2 : P_2, \quad \rho(\mathcal{A}_1) \cap \rho(\mathcal{A}_1) \neq \emptyset$$

$$\begin{array}{c} \{(r,1)\} : P_1 \parallel \underline{\{(r,2)\}} : P_2 \\ \Big\downarrow \{(r,2)\} \\ \{(r,1)\} : P_1 \parallel P_2 \end{array}$$

PACoR : Semantics

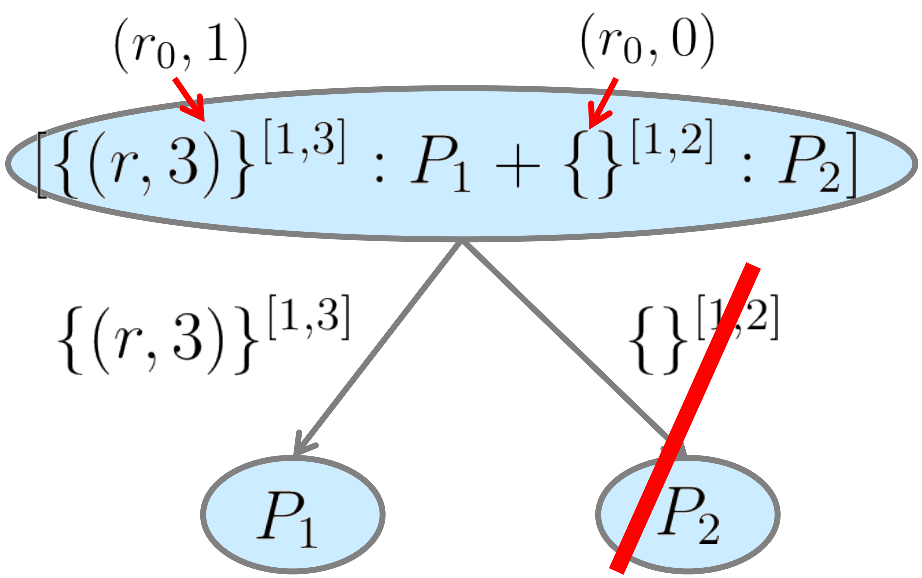
- Close



PACoR : Semantics

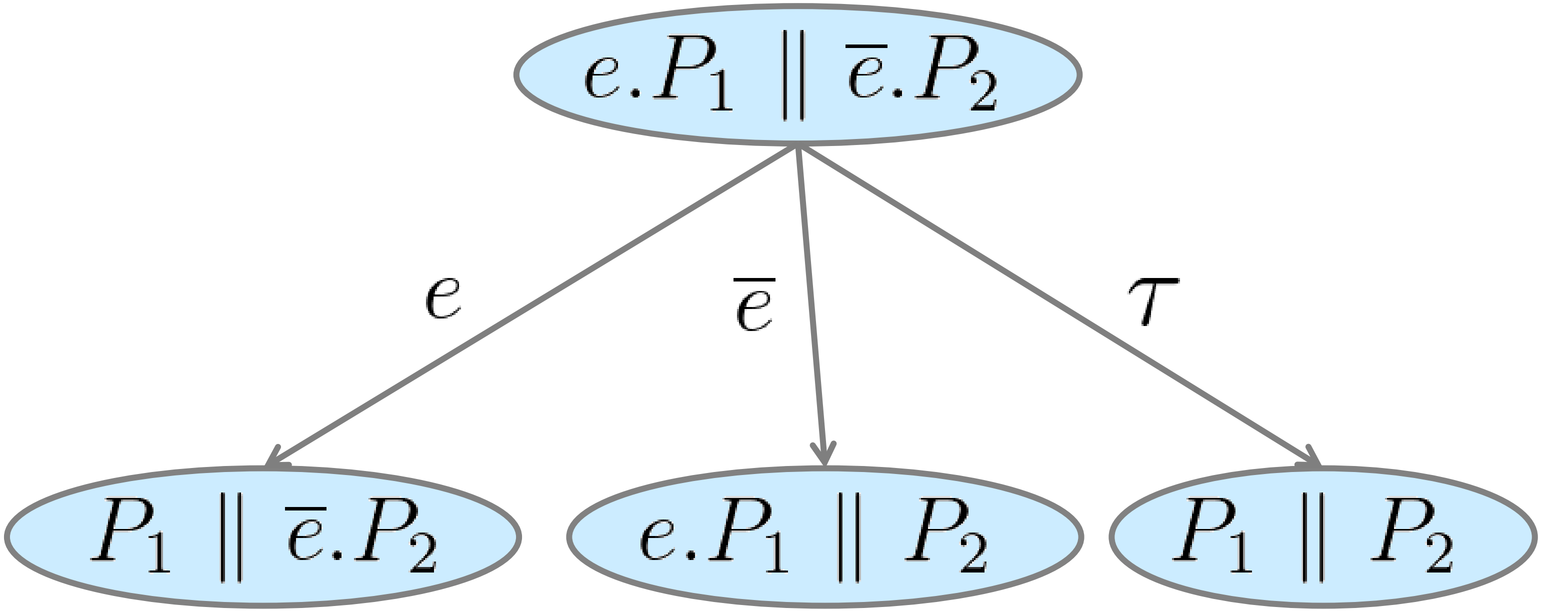
- Close

$$\frac{m \in \mathbb{R}_{\geq 0},\ Ident(P) := Ident([\mathcal{A}^{[l,u]} : P]_I)}{\langle [\mathcal{A}^{[l,u]} : P]_I, x, \mathtt{ID}\rangle \xrightarrow{\mathcal{A} \cup \{(r_0, \underline{lp(\mathcal{A})})\}} \langle P, x+m, \mathtt{ID}\rangle}$$



PACoR : Semantics

- Restrict



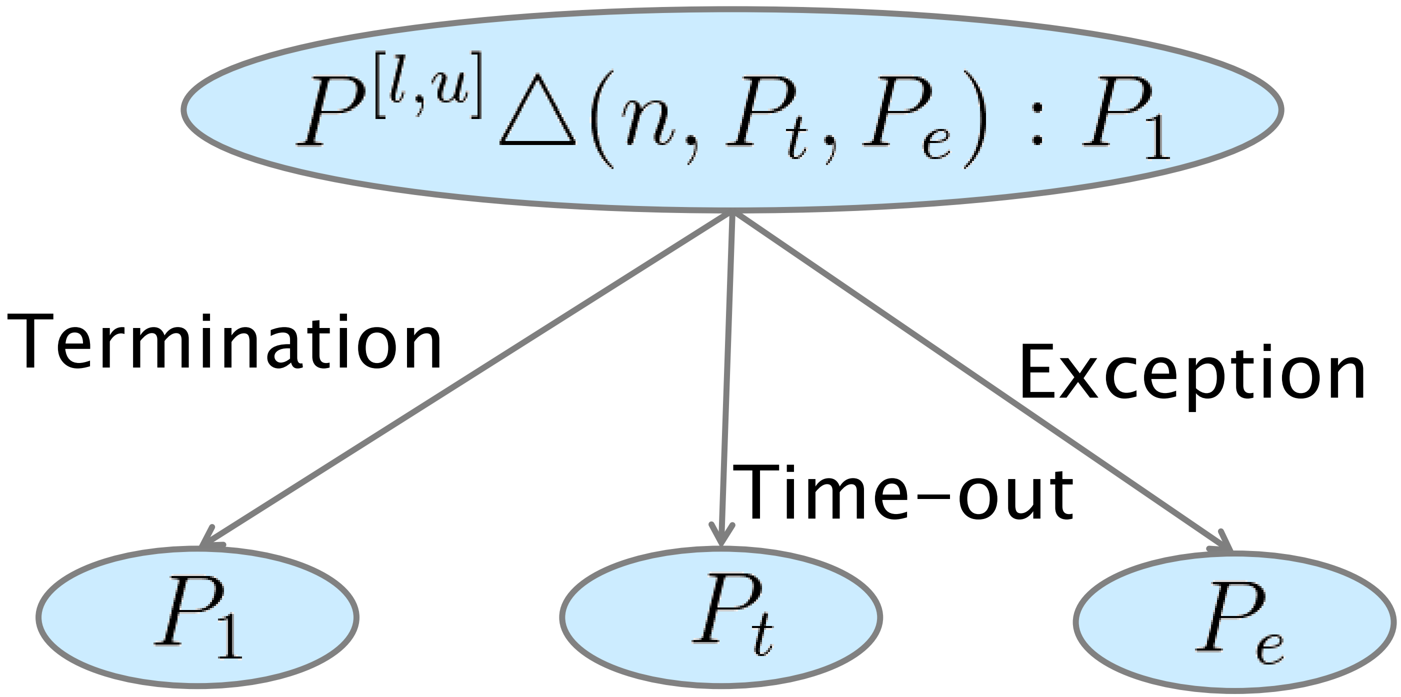
PACoR : Semantics

- TTS Restriction (Def 3.4)

$$\begin{tikzcd}[column sep=small] & (e.P_1 \parallel \overline{e}.P_2)\backslash\underline{\{e\}} \arrow[dl, "\cancel{e}"'] \arrow[d, "\cancel{\overline{e}}"] \arrow[dr, "\tau"] & \\ \cancel{P_1 \parallel \overline{e}.P_2} & \cancel{e.P_1 \parallel P_2} & P_1 \parallel P_2 \end{tikzcd}$$

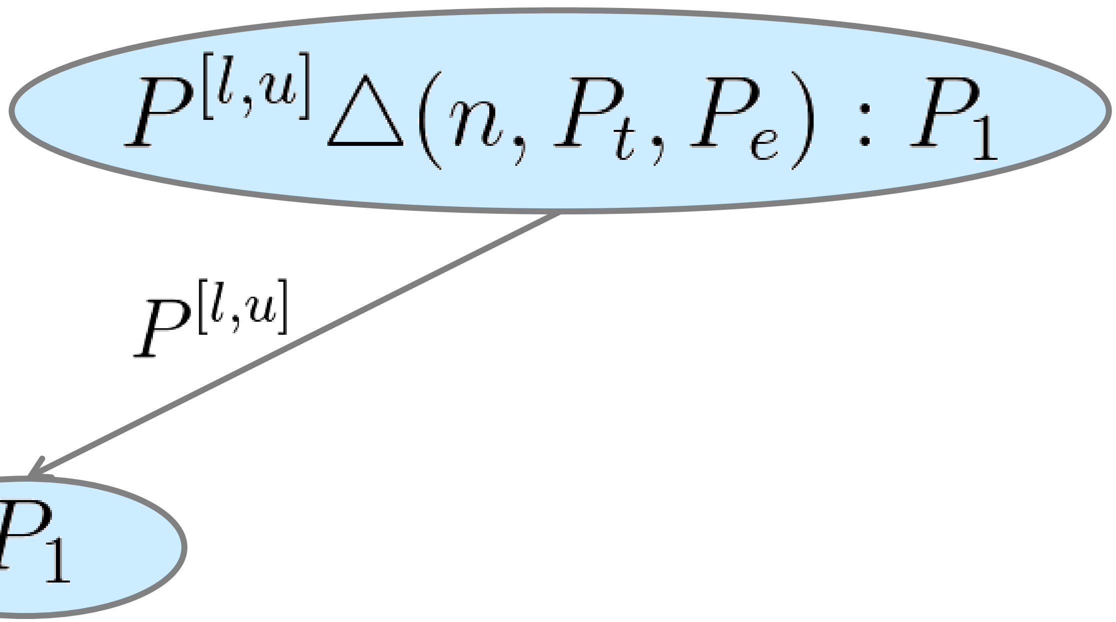
PACoR : Semantics

- Scope Operator for Timed Action



PACoR : Semantics

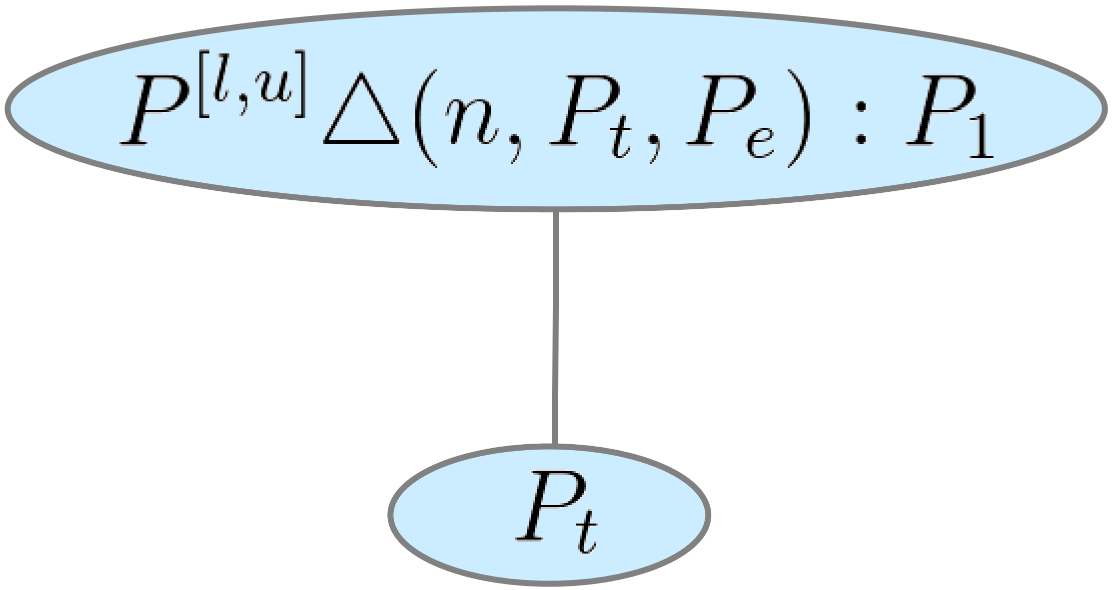
- Scope Operator for Timed Action



$$\frac{m \in \mathbb{R}_{\geq 0},\ l \leq m \leq u,\ m \leq n,\ Ident(P) := Ident(\mathcal{A}^{[l,u]}\triangle(..) : P)}{\langle \mathcal{A}^{[l,u]}\triangle(n, P_t, P_e) : P, x, \mathtt{ID}\rangle \xrightarrow{\mathcal{A}} \langle P, x + m, \mathtt{ID}\rangle}$$

PACoR : Semantics

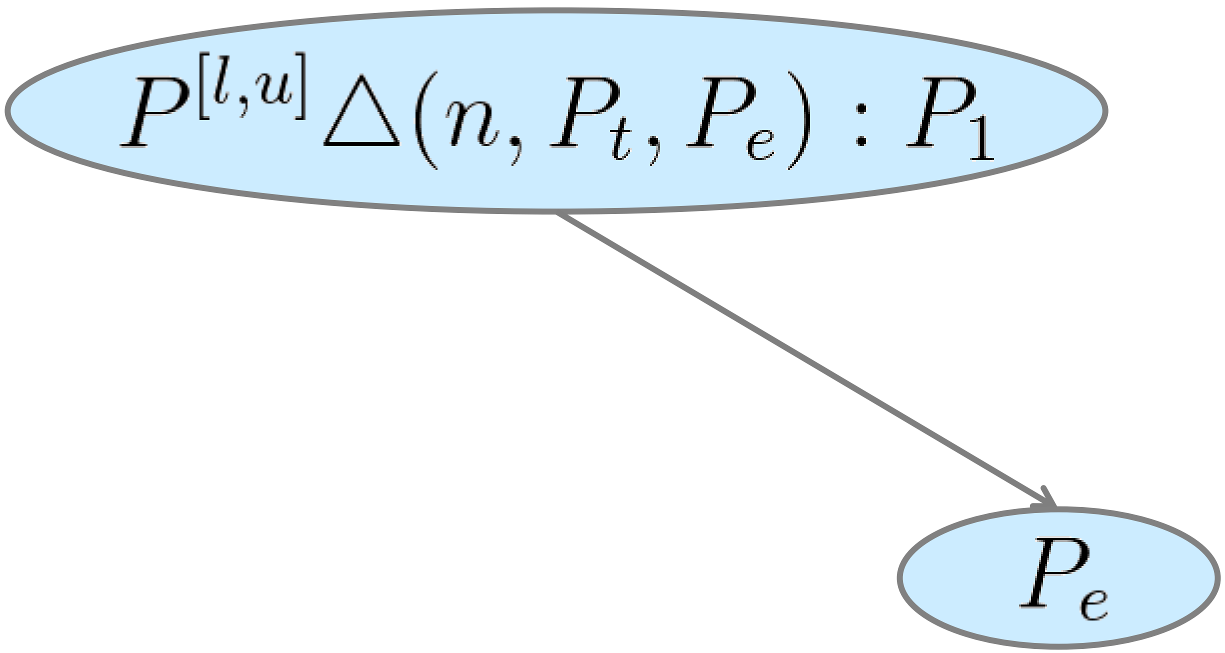
- Scope Operator for Timed Action



$$\frac{n = 0}{\langle \mathcal{A}^{[l,u]}\triangle(n, P_t, P_e), x, \mathtt{ID}\rangle \xrightarrow{\tau} \langle P_t, x, \mathtt{ID}\rangle}$$

PACoR : Semantics

- Scope Operator for Timed Action

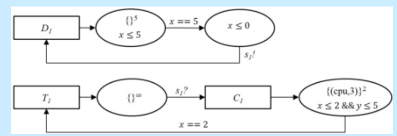


Analysis

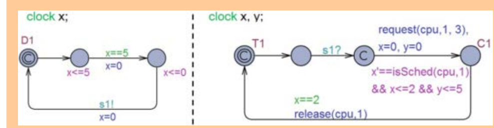
PACoR

$$D_1 \overset{def}{=} \emptyset^5 : \bar{s}_1.D_1$$
$$T_1 \overset{def}{=} s_1\nabla(\infty, NIL, NIL).C_1$$
$$C_1 \overset{def}{=} \{(cpu,3)\}^{[2,2]}\triangle(5, NIL, NIL) : T_1$$

gPACoR



Parameterized Stopwatch Automata



Uppaal
MC & SMC

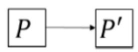
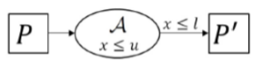
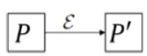
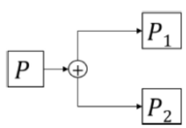
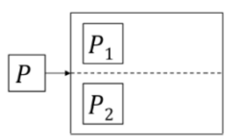
Property Formula

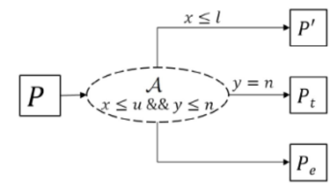
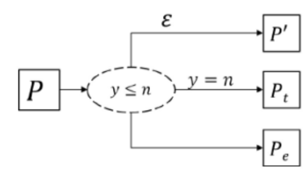
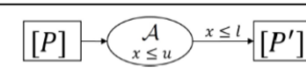
Yes/No

Prob. Dist

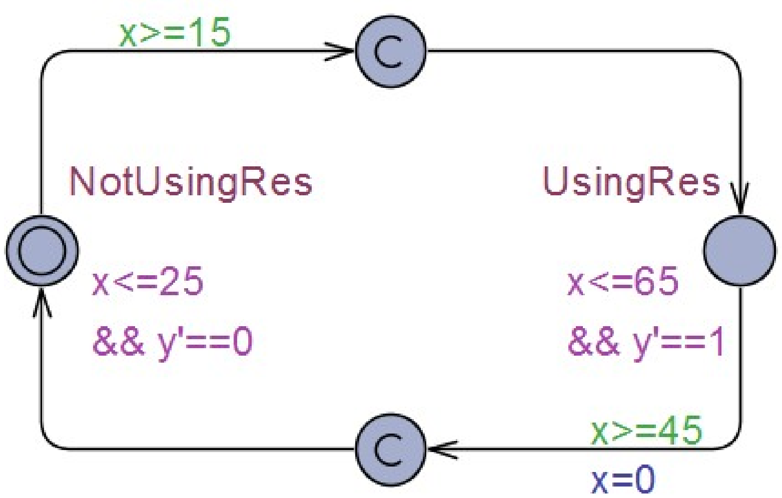
gPACoR

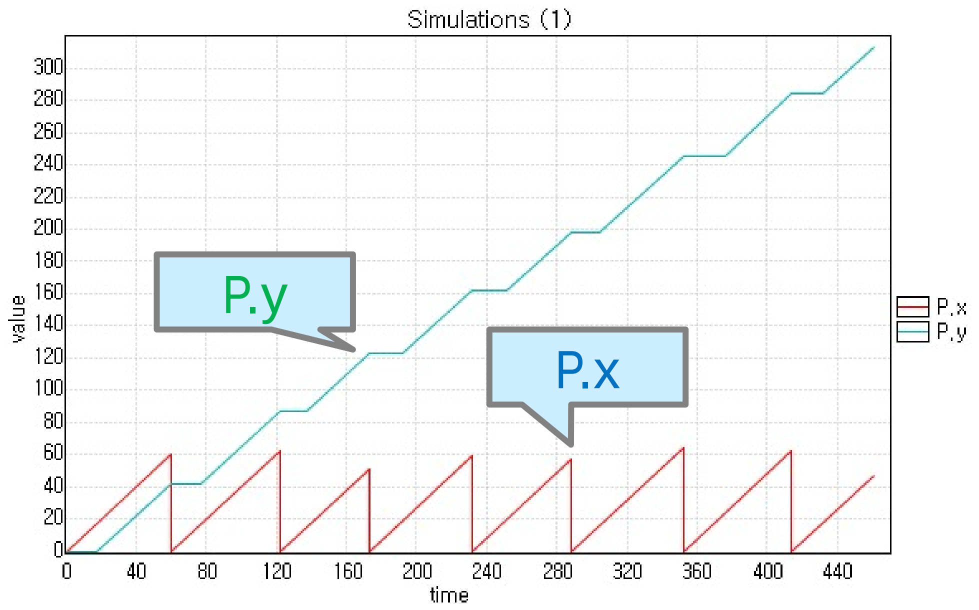
- The Graphical PACoR

Rule	PACoR	gPACoR
1	$P \stackrel{def}{=} P'$	
2	$P \stackrel{def}{=} \mathcal{A}^{[l,u]} : P'$	
3	$P \stackrel{def}{=} \mathcal{E}.P'$	
4	$P \stackrel{def}{=} P_1 + P_2$	
5	$P \stackrel{def}{=} P_1 \parallel P_2$	

Rule	PACoR	gPACoR
6	$P \stackrel{def}{=} \mathcal{A}^{[l,u]} \triangle(n, P_t, P_e) : P'$	
7	$P \stackrel{def}{=} \mathcal{E}\nabla(n, P_t, P_e).P'$	
8	$[P] \stackrel{def}{=} \mathcal{A}^{[l,u]} : P'$	

Parameterized Stopwatch Automata





clock x, y

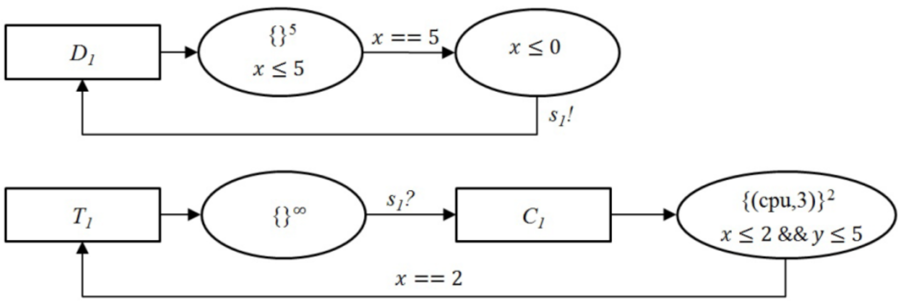
Translation

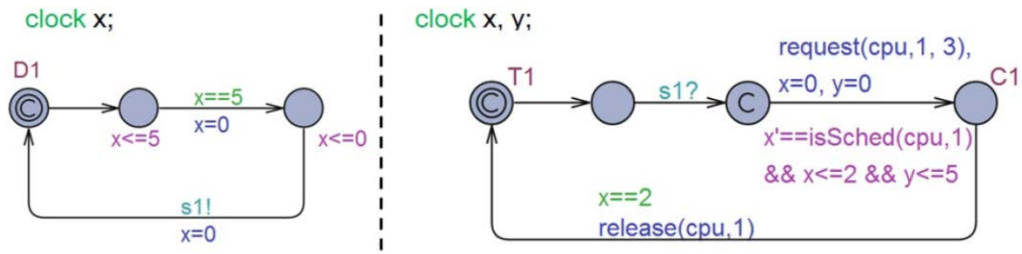
PACoR

gPACoR

Parameterized Stopwatch Automata

$$D_1 \stackrel{def}{=} \emptyset^5 : \bar{s_1}.D_1$$
$$T_1 \stackrel{def}{=} s_1 \nabla(\infty, NIL, NIL).C_1$$
$$C_1 \stackrel{def}{=} \{(cpu, 3)\}^{[2,2]} \triangle (5, NIL, NIL) : T_1$$

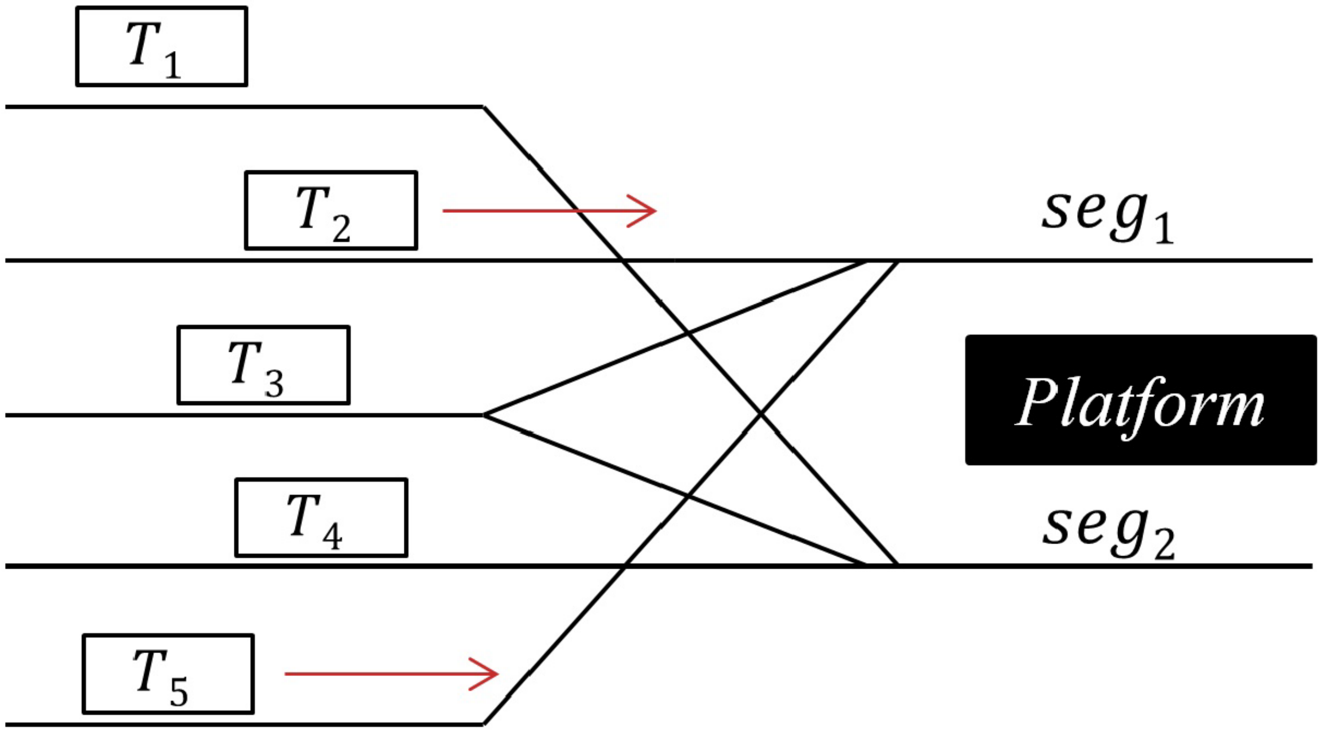




Translation

$$
\begin{tikzcd}
 & \mathrm{PACoR} \arrow[ddl, "\text{Semantics}"'] \arrow[dr, "\text{Translation}"] & \\
 & & \mathrm{PSA} \arrow[d, "\text{Semantics}"'] \\
TTS_{\mathrm{PACoR}} \arrow[rr, phantom, "\overset{\text{bi-similar}}{\approx}"] & & TTS_{\mathrm{PSA}} \backslash \Omega
\end{tikzcd}
$$

Example: Train Platform System (TPS)

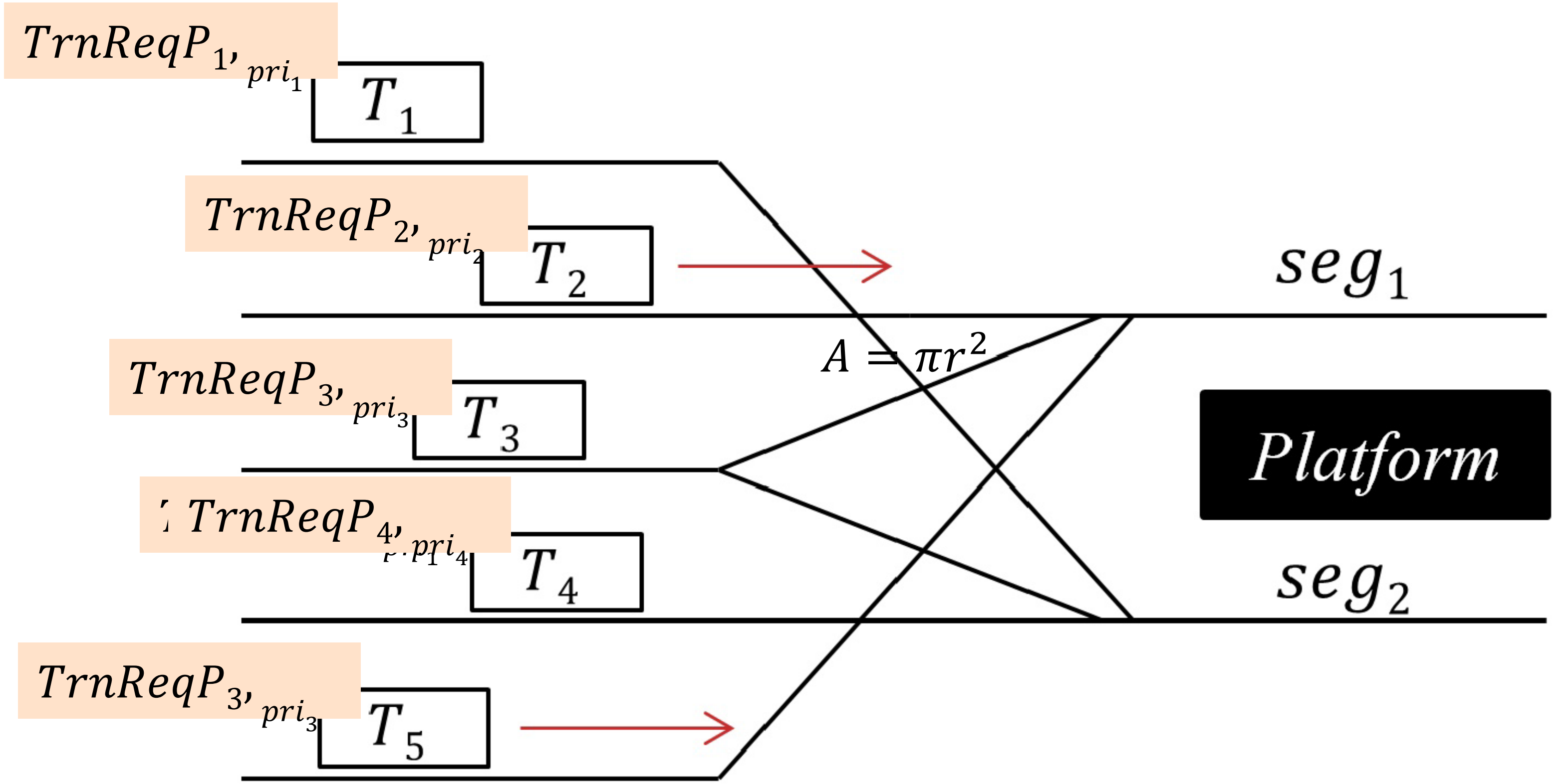


TPS: PACoR Specification

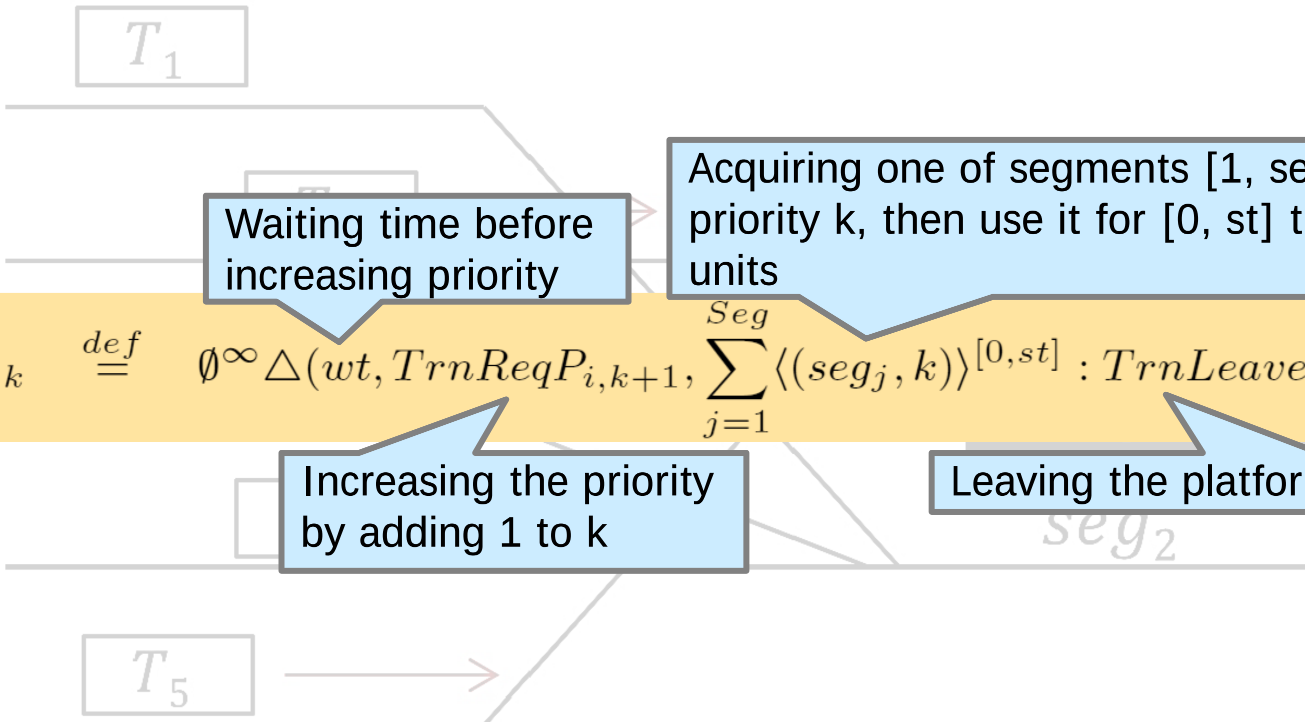
$$
\begin{aligned}
System &\stackrel{def}{=} [\ TrnReqP_{1,pri_1} \parallel TrnReqP_{2,pri_2} \parallel \ldots \parallel TrnReqP_{T,pri_T}\ ]_{\{seg_j\}} \qquad 1 \leq j \leq Seg \\
TrnReqP_{i,k} &\stackrel{def}{=} \emptyset^{\infty} \triangle(wt, TrnReqP_{i,k+1}, \sum_{j=1}^{Seg} \langle (seg_j, k) \rangle^{[0,st]} : TrnLeaveP_{j,i}) : NIL \\
TrnLeaveP_{j,i} &\stackrel{def}{=} \emptyset^{mt} : TrnArvl_i \\
TrnArvl_i &\stackrel{def}{=} \emptyset^{rt} : TrnArvl_i + TrnReqP_{i,k}
\end{aligned}
$$

TPS: PACoR Specification (cont'd)

$$System \overset{def}{=} [\ TrnReqP_{1,pri_1} \parallel TrnReqP_{2,pri_2} \parallel \ldots \parallel TrnReqP_{T,pri_T}\ ]_{\{seg_j\}} \qquad 1 \le j \le Seg$$



TPS: PACoR Specification (cont'd)



Waiting time before increasing priority

Acquiring one of segments [1, seg] at priority k, then use it for [0, st] time units

$$TrnReqP_{i,k} \stackrel{def}{=} \emptyset^{\infty} \triangle (wt, TrnReqP_{i,k+1}, \sum_{j=1}^{Seg} \langle (seg_j, k) \rangle^{[0,st]} : TrnLeaveP_{j,i}) : NIL$$

Increasing the priority by adding 1 to k

Leaving the platform

TPS: Analysis

- Deadlock freedom (using Uppaal MC)

A[ ] not deadlock

TPS: Analysis

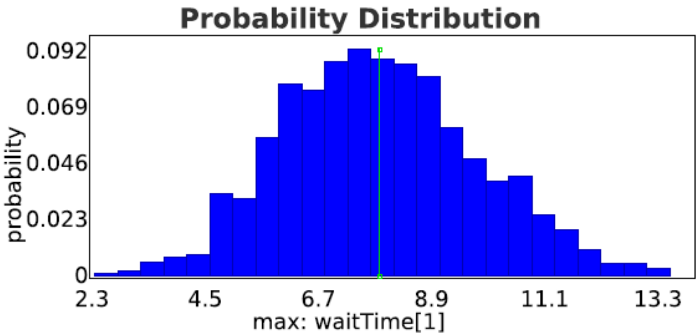
- Maximum waiting time (using Uppal SMC)

  E[<=simTime;simCount](max:waitTime[i])

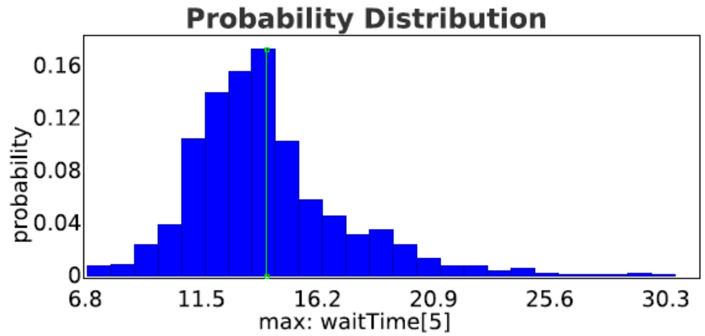
TPS: Analysis

The maximum waiting time with *st* = 10 5 and *wt* = 10

	Case 1 ($Seg = 2$)		Case 2 ($Seg = 2$)		Case 3 ($Seg = 3$)	
i	pri_i	Wait Time	pri_i	Wait Time	pri_i	Wait Time
1	1	14.14±0.22	5	8.04±0.12	5	4.13±0.13
2	1	14.22±0.22	4	9.42±0.14	4	4.40±0.15
3	1	14.16±0.22	3	11.76±0.16	3	4.61±0.15
4	1	14.34±0.23	2	14.41±0.19	2	7.67±0.10
5	1	14.20±0.22	1	20.94±0.31	1	9.15±0.08



(a) Track 1



(b) Track 5

Conclusions

- **RTS** is a time and resource constrained system,
- For the rigorous analysis, this paper presents
  - A new formalism, **PACoR**,
  - Translation rules for the analysis using **Uppaal** and **Uppaal SMC**

Conclusions

- PACoR
  - Rigorous and complete specification of various scheduling systems
  - Supported by Uppaal tools to answer to qualitative and quantitative qualities of RTS, such as schedulability, and worst-case response time (WCRT), and so on, using **the same model.**


Thank you