# SMT-based Bounded Model Checking for Parametric Reaction Systems


**Wojciech Penczek**

joint work with

Artur Męski

Institute of Computer Science, Polish Academy of Sciences
University of Natural Sciences and Humanities



SynCoP, April 7, 2019, Prague

# Related Work

1. *Model checking temporal properties of reaction systems*
   Information Sciences 313, 2015; A. Męski, W. Penczek, G. Rozenberg
2. *Complexity of model checking for reaction systems*, TCS 623, 2016
   S. Azimi, C. Gratie, S. Ivanov, L. Manzoni, I. Petre, A. E. Porreca
3. *Verification of Linear-Time Temporal Properties for Reaction Systems with Discrete Concentrations*
   Fundamenta Informaticae, 2017; A. Męski, M. Koutny, W. Penczek
4. *Reaction Mining for Reaction Systems*
   UCNC, 2018; A. Męski, M. Koutny, W. Penczek

# Outline

# Reaction systems

A **reaction system** is a pair $rs = (S, A)$, where:

- $S$ – finite background set *entities/molecules*
- $A$ – set of reactions over $S$

Each reaction in $A$ is a triple $b = (R, I, P)$ such that $R$, $I$, $P$ are nonempty subsets of $S$ with $R \cap I = \varnothing$.

- $R$ – reactants, $R_b$
- $I$ – inhibitors, $I_b$
- $P$ – products, $P_b$

# Example

$$(S, A) = (\{1, 2, 3, 4\}, \{a, b, c, d\})$$

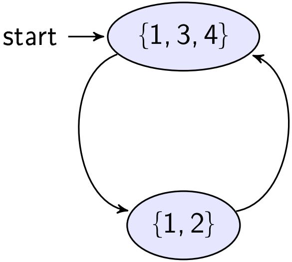
$$\begin{array}{lllllll} a & = & (\{1,4\},\{2\},\{1,2\}) & & b & = & (\{2\},\{4\},\{1,3,4\}) \\ c & = & (\{1,3\},\{2\},\{1,2\}) & & d & = & (\{3\},\{2\},\{1\}) \end{array}$$

In state $\{1, 3, 4\}$:

- $a$, $c$, $d$ – enabled reactions

Individual results for the reactions:

- $a \longrightarrow \{1, 2\}$
- $b \longrightarrow \varnothing$
- $c \longrightarrow \{1, 2\}$
- $d \longrightarrow \{1\}$




Result state: $\{1, 2\}$

# Environment

- Execution of reaction systems depends on their environment
- Environment is defined in reaction systems as **context**
- **Context** – **sequence** of sets of entities
- Supplied at **each step** of execution
- Affects reactions **enablement**:

  states are extended with a corresponding context

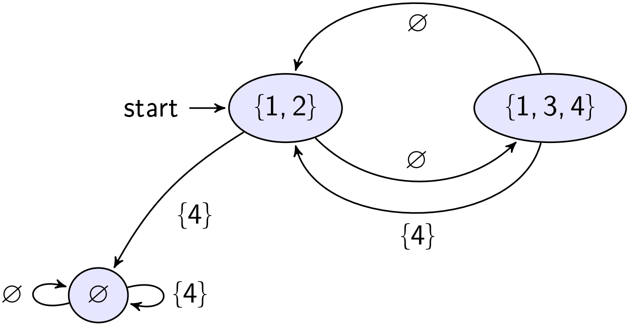
# Example

$$(S, A) = (\{1,2,3,4\}, \{a, b, c, d\})$$

**initial state**: $\{1,2\}$ **environment** (context): $2^{\{4\}} = \{\emptyset, \{4\}\}$

$$\begin{array}{lllllll} a & = & (\{1,4\}, \{2\}, \{1,2\}) & \quad & b & = & (\{2\}, \{4\}, \{1,3,4\}) \\ c & = & (\{1,3\}, \{2\}, \{1,2\}) & & d & = & (\{3\}, \{2\}, \{1\}) \end{array}$$

---



# Model checking for rsCTL [MPR15]

Input:

- Initialised context restricted reaction system: $\mathcal{icrrs}$
- rsCTL formula $\phi$
  (rsCTL - CTL with path selection by referring to contexts)

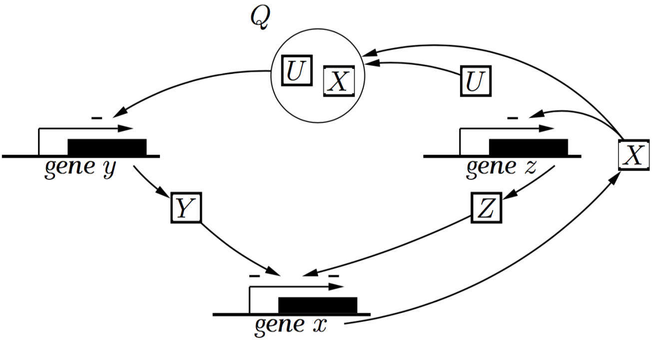
Decision problem:

$$\underbrace{\mathcal{M}}_{\text{model for icrrs}} \overset{?}{\models} \underbrace{\phi}_{\text{formula}}$$

**Theorem.** The model checking problem for rsCTL is PSPACE-complete.

**Model checking algorithm** is based on BDDs.

# Example. Gene regulatory network




Three (abstract) genes $x$, $y$, $z$ expressing proteins X, Y, Z, respectively, protein U, and protein complex Q formed by X and U.

The expression of X by $x$ is inhibited by Y and Z, the expression of Z by $z$ is inhibited by X, and expression of Y by $y$ is inhibited by the protein complex Q.

# Example. Gene regulatory network: properties

1. It is possible that the protein $Q$ will never be produced:

$$\mathbf{EG}(\neg Q).$$

2. If we do not supply $U$ in the context, then $Q$ will never be produced:

$$\mathbf{A}_{\Psi}\mathbf{G}(\neg Q), \text{ where } \Psi = \{\alpha \subseteq \mathcal{E} \mid U \notin \alpha\} = \{\varnothing\}.$$

# Linear-Time Temporal Properties of RS

*Verification of Linear-Time Temporal Properties*
*for Reaction Systems with Discrete Concentrations*
Fundamenta Informaticae, 2017; A. Męski, M. Koutny, W. Penczek

## Multisets over $S$ : $\mathcal{B}(S)$

- $s \mapsto i$ – multiplicity of $s$ e.g. $\{s \mapsto 2, x \mapsto 3, y\}$

**Multiset expressions**: $\mathfrak{a} \in BE(S)$

$$\mathfrak{a} ::= \mathtt{true} \mid e \sim c \mid e \sim e \mid \neg \mathfrak{a} \mid \mathfrak{a} \vee \mathfrak{a}$$

where:

- $\sim \in \{<, \leqslant, =, \geqslant, >\}$
- $e \in S$
- $c \in \mathbb{N}$

Then $\mathbf{b} \models_b \mathfrak{a}$ means that $\mathfrak{a}$ holds for $\mathbf{b} \in \mathcal{B}(S)$:

$$\begin{array}{lll}
\mathbf{b} \models_b \mathtt{true} & \text{for every } \mathbf{b} \in \mathcal{B}(S) & \\
\mathbf{b} \models_b e \sim c & \text{iff} & \mathbf{b}(e) \sim c \\
\mathbf{b} \models_b e \sim e' & \text{iff} & \mathbf{b}(e) \sim \mathbf{b}(e') \\
\mathbf{b} \models_b \neg \mathfrak{a} & \text{iff} & \mathbf{b} \not\models_b \mathfrak{a} \\
\mathbf{b} \models_b \mathfrak{a} \vee \mathfrak{a}' & \text{iff} & \mathbf{b} \models_b \mathfrak{a} \text{ or } \mathbf{b} \models_b \mathfrak{a}'
\end{array}$$

# Reaction systems with concentrations: definition

$\mathrm{rsc} = (S, A)$ – *reaction system with (discrete) concentrations*:

- $S$ – finite **background set**
- $A$ – nonempty finite set of **c-reactions** over $S$

$\mathcal{B}(S)$ – set of all bags over $S$;
$a = (\mathbf{r}, \mathbf{i}, \mathbf{p}) \in A$ – **c-reaction**

- $\mathbf{r}, \mathbf{i}, \mathbf{p} \in \mathcal{B}(S)$ with $\mathbf{r}(e) < \mathbf{i}(e)$, for every $e \in \mathrm{carr}(\mathbf{i})$
  $(\mathrm{carr}(\mathbf{b}) = \{s \in S \mid \mathbf{b}(s) > 0\})$

- $\mathbf{r}$, $\mathbf{i}$, $\mathbf{p}$ – reactant, inhibitor, and product concentration levels
- denoted: $\mathbf{r}_a$, $\mathbf{i}_a$, and $\mathbf{p}_a$

# Reaction systems with concentrations: enablement

A **c-reaction** $a \in A$ is enabled by $\mathbf{t} \in \mathcal{B}(S)$, denoted $en_a(\mathbf{t})$, if $\mathbf{r}_a \leqslant \mathbf{t}$ and $\mathbf{t}(e) < \mathbf{i}_a(e)$, for every $e \in carr(\mathbf{i}_a)$

$res_a(\mathbf{t})$ – the result of $a$ on $\mathbf{t}$:

- $res_a(\mathbf{t}) = \mathbf{p}_a$ if $en_a(\mathbf{t})$
- $res_a(\mathbf{t}) = \varnothing_S$ otherwise

$res_A(\mathbf{t}) = \bigwedge\!\!\!\bigwedge\{res_a(\mathbf{t}) \mid a \in A\} = \bigwedge\!\!\!\bigwedge\{\mathbf{p}_a \mid a \in A \text{ and } en_a(\mathbf{t})\}$.

$\bigwedge\!\!\!\bigwedge(\mathbf{B})(x) = \max(\{\mathbf{b}(x) \mid \mathbf{b} \in \mathbf{B}\})$ for non-empty $\mathbf{B} \subseteq \mathcal{B}(S)$,
$\mathbf{b} \leqslant \mathbf{b}'$ if $\mathbf{b}(x) \leqslant \mathbf{b}'(x)$ for every $x \in X$

# Context-restricted rsc

**Context automaton** over the set $\mathcal{B}(S)$:
$ca = (Q, q_0, R)$, where:

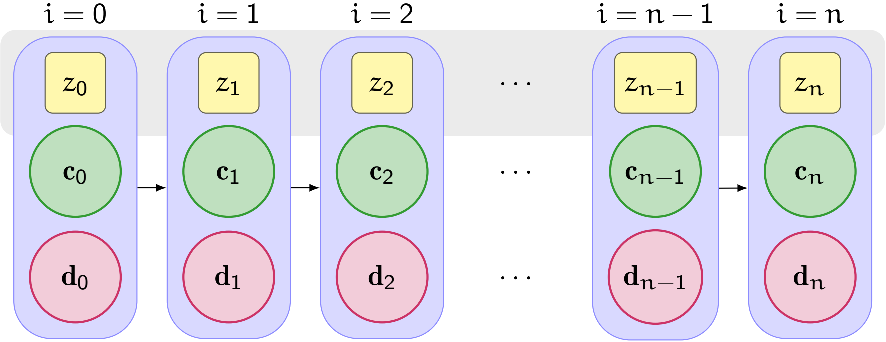
- $Q$ – finite set of states
- $q_0 \in Q$ – the initial state
- $R \subseteq Q \times \mathcal{B}(S) \times Q$ – transition relation

$crrsc = (rsc, ca)$ — **context-restricted rsc**:

- $rsc = (S, A)$ – reaction system with discrete concentrations
- $ca = (Q, q_0, R)$ – context automaton over $\mathcal{B}(S)$

# Interactive processes of crrsc

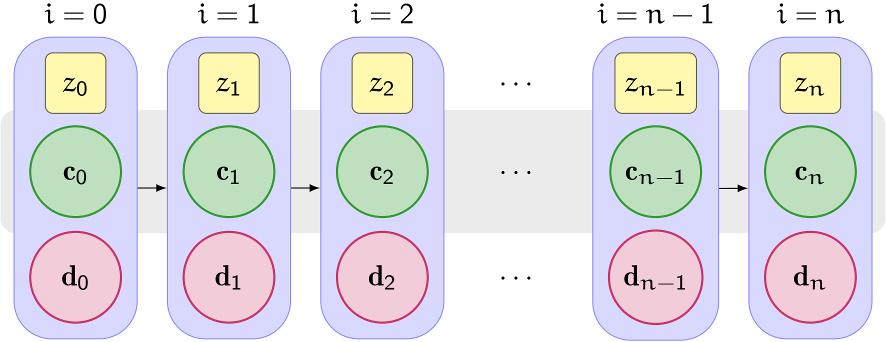
$\pi = (\zeta, \gamma, \delta)$ – ($n$-step) interactive process in crrsc




$\zeta = (z_0, z_1, \ldots, z_n)$ $\qquad$ $z_0, z_1, \ldots, z_n \in Q$ with $z_0 = q_0$

# Interactive processes of crrsc

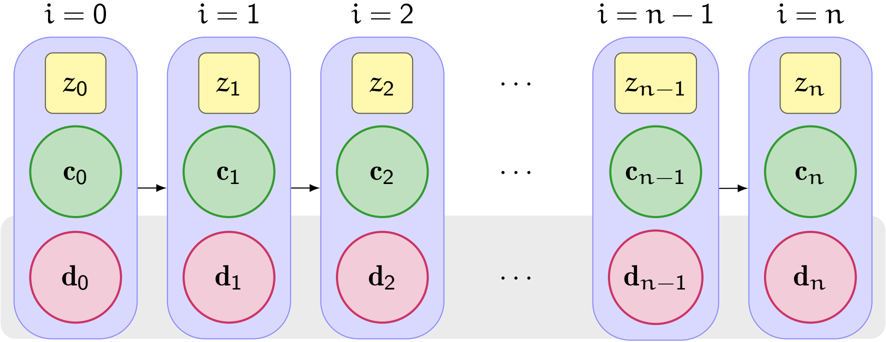
$\pi = (\zeta, \gamma, \delta)$ – ($n$-step) interactive process in crrsc




$\gamma = (\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_n)$ $\qquad$ $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_n \in \mathcal{B}(S)$

# Interactive processes of crrsc

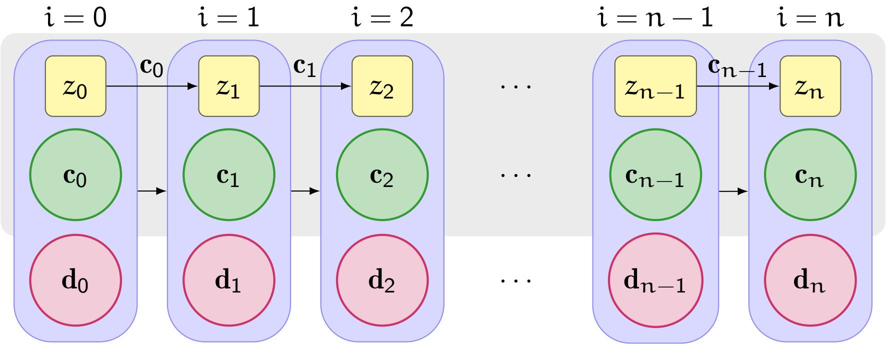
$\pi = (\zeta, \gamma, \delta)$ – ($n$-step) interactive process in crrsc




$\delta = (\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_n) \qquad \mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_n \in \mathcal{B}(\mathcal{S})$

# Interactive processes of crrsc

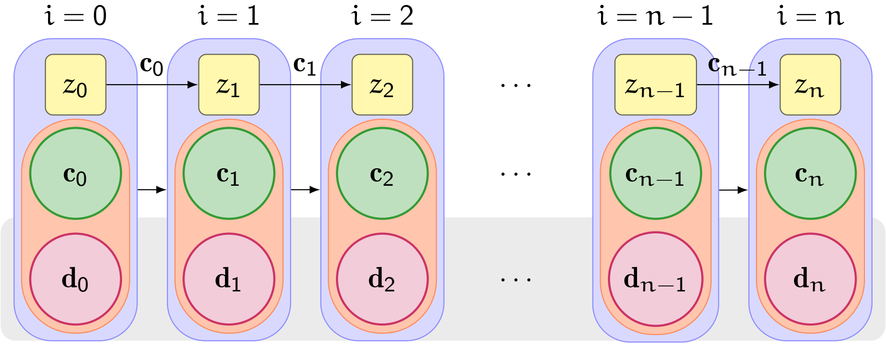
$\pi = (\zeta, \gamma, \delta)$ – ($n$-step) interactive process in crrsc




$(z_i, \mathbf{c}_i, z_{i+1}) \in R$, for every $i \in \{0, \ldots, n-1\}$

# Interactive processes of crrsc

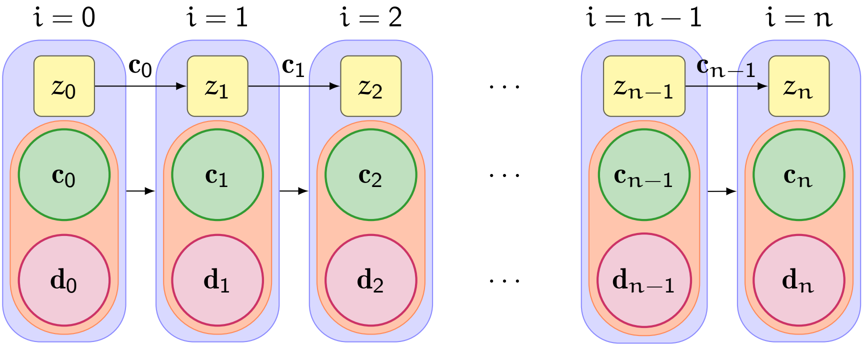
$\pi = (\zeta, \gamma, \delta)$ – ($n$-step) interactive process in crrsc




$\mathbf{d}_0 = \emptyset_{\mathcal{B}(S)}$, $\mathbf{d}_i = \mathrm{res}_A(\bigwedge\!\!\!\bigwedge\{\mathbf{d}_{i-1}, \mathbf{c}_{i-1}\})$, for every $i \in \{1, \ldots, n\}$

# Interactive processes of crrsc

$\pi = (\zeta, \gamma, \delta)$ – ($n$-step) interactive process in crrsc




*state sequence* of $\pi$: $(\mathbf{w}_0, \ldots, \mathbf{w}_n) = (\bigwedge\{\mathbf{c}_0, \mathbf{d}_0\}, \ldots, \bigwedge\{\mathbf{c}_n, \mathbf{d}_n\})$

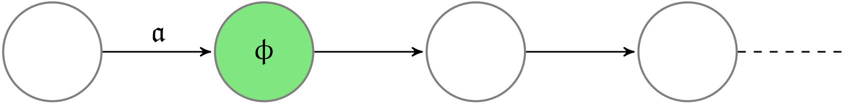
# LTL for RS – rsLTL

The syntax of rsLTL is given by the following grammar:

$$\phi ::= \mathfrak{a} \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathbf{X}_{\mathfrak{a}}\phi \mid \phi\mathbf{U}_{\mathfrak{a}}\phi \mid \phi\mathbf{R}_{\mathfrak{a}}\phi$$

where $\mathfrak{a} \in BE(S)$

$$\sigma \models \mathbf{X}_{\mathfrak{a}}\phi$$



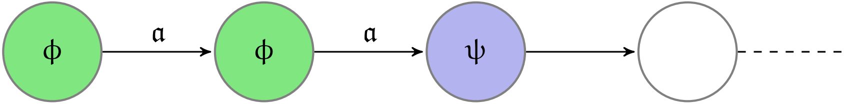
# LTL for RS – rsLTL

The syntax of rsLTL is given by the following grammar:

$$\phi ::= \mathfrak{a} \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathbf{X}_{\mathfrak{a}}\phi \mid \phi \mathbf{U}_{\mathfrak{a}}\phi \mid \phi \mathbf{R}_{\mathfrak{a}}\phi$$

where $\mathfrak{a} \in BE(S)$

$$\sigma \models \phi \mathbf{U}_{\mathfrak{a}} \psi$$



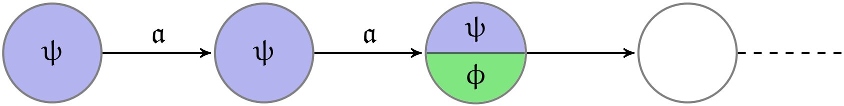
# LTL for RS – rsLTL

The syntax of rsLTL is given by the following grammar:

$$\phi ::= \mathfrak{a} \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathbf{X}_{\mathfrak{a}}\phi \mid \phi \mathbf{U}_{\mathfrak{a}} \phi \mid \phi \mathbf{R}_{\mathfrak{a}} \phi$$

where $\mathfrak{a} \in BE(S)$

$$\sigma \models \phi \mathbf{R}_{\mathfrak{a}} \psi$$



# Reaction Mining

- Reactions may be defined partially
  e.g., missing information about inhibitors, reactants, etc.

- From experiments we obtain observations which help fill in the missing information about reactions

- Assumption: experiments result in *existential observations*

# Reaction Mining

- Reactions may be defined partially
  e.g., missing information about inhibitors, reactants, etc.
  - Partially defined reaction systems:
    **parametric reaction systems**
- From experiments we obtain observations which help fill in the missing information about reactions
  - rsLTL is used to express these observations
- Assumption: experiments result in *existential observations*
  - rsLTL interpreted existentially

# Parametric reaction systems (with discrete concentrations)

**Parametric reaction system**: $prs = (S, P, A)$, where:

- $S$ – *background* set
- $P$ – set of *parameters*
- $A$ – set of *parametric reactions*, $A \neq \varnothing$

$S$, $P$, $A$ are finite

Let $a = (\mathfrak{r}, \mathfrak{i}, \mathfrak{p}) \in A$: $\quad \mathfrak{r}, \mathfrak{i}, \mathfrak{p} \in \mathcal{B}(S) \cup P$

- $\mathfrak{r}$, $\mathfrak{i}$, $\mathfrak{p}$ — denoted by $\mathfrak{r}_a$, $\mathfrak{i}_a$, and $\mathfrak{p}_a$
- *reactants*, *inhibitors*, and *products* of parametric reaction $a$

Example: Let $\lambda_1, \lambda_2 \in P$

- Parametric reactions: $(\{x, y\}, \lambda_1, \{z\})$, $(\lambda_1, \{x\}, \lambda_2)$

# Parameter valuations

**Parameter valuation** of $prs$:

$$\mathtt{v} : P \to \mathcal{B}(S)$$

- we write $\mathfrak{b}^{\leftarrow \mathtt{v}}$ for $\mathtt{v}(\mathfrak{b})$
- $\mathtt{PV}_{prs}$ – all the parameter valuations for $prs$

Parameter substitutions

- Parameters are substituted according to $\mathtt{v} \in \mathtt{PV}_{prs}$
- $X^{\leftarrow \mathtt{v}} \stackrel{\mathrm{def}}{=} \{(a_{\mathfrak{r}}^{\leftarrow \mathtt{v}}, a_{\mathfrak{i}}^{\leftarrow \mathtt{v}}, a_{\mathfrak{p}}^{\leftarrow \mathtt{v}}) \mid a \in X\}$ for $X \subseteq A$
- $\mathrm{prs}^{\leftarrow \mathtt{v}} \stackrel{\mathrm{def}}{=} (S, A^{\leftarrow \mathtt{v}})$

$\mathtt{v} \in \mathtt{PV}_{prs}$ is a **valid parameter valuation** if $\mathrm{prs}^{\leftarrow \mathtt{v}}$ yields an rsc

# Context-restricted PRS

**Context-restricted parametric reaction system** (crprs):

$$crprs = (prs, ca)$$

where:

- $prs = (S, P, A)$
- $ca = (Q, q_0, R)$ – context automaton over $\mathcal{B}(S)$

For $\mathtt{v} \in \mathtt{PV}_{prs}$ we define:

$$crprs^{\leftarrow \mathtt{v}} = (prs^{\leftarrow \mathtt{v}}, ca)$$

# Parameter constraints

$\mathfrak{c} \in PC(prs)$:

$$\mathfrak{c} ::= \mathtt{true} \mid \lambda[\mathrm{e}] \sim \mathrm{c} \mid \lambda[\mathrm{e}] \sim \lambda[\mathrm{e}] \mid \neg\mathfrak{c} \mid \mathfrak{c} \vee \mathfrak{c},$$

where:

$$\lambda \in \mathsf{P} \qquad \mathrm{e} \in \mathsf{S} \qquad \mathrm{c} \in \mathbb{N} \qquad \sim \in \{<, \leqslant, =, \geqslant, >\}$$

Let $\mathrm{v} \in \mathsf{PV}_{prs}$

- $\mathfrak{c}$ holds in $\mathrm{v}$ is denoted $\mathrm{v} \models_{\mathrm{p}} \mathfrak{c}$:

$$\begin{array}{lll}
\mathrm{v} \models_{\mathrm{p}} \mathtt{true} & \text{for every } \mathrm{v} & \\
\mathrm{v} \models_{\mathrm{p}} \lambda[\mathrm{e}] \sim \mathrm{c} & \text{if} & \lambda^{\leftarrow \mathrm{v}}(\mathrm{e}) \sim \mathrm{c} \\
\mathrm{v} \models_{\mathrm{p}} \lambda_1[\mathrm{e}_1] \sim \lambda_2[\mathrm{e}_2] & \text{if} & \lambda_1^{\leftarrow \mathrm{v}}(\mathrm{e}_1) \sim \lambda_2^{\leftarrow \mathrm{v}}(\mathrm{e}_2) \\
\mathrm{v} \models_{\mathrm{p}} \neg\mathfrak{c} & \text{if} & \mathrm{v} \not\models_{\mathrm{p}} \mathfrak{c} \\
\mathrm{v} \models_{\mathrm{p}} \mathfrak{c}_1 \vee \mathfrak{c}_2 & \text{if} & \mathrm{v} \models_{\mathrm{p}} \mathfrak{c}_1 \text{ or } \mathrm{v} \models_{\mathrm{p}} \mathfrak{c}_2
\end{array}$$

# Constrained PRS

**Constrained parametric reaction system**: $cprs = (\mathsf{S}, \mathsf{P}, \mathcal{A}, \mathfrak{c})$ where:

- $prs = (\mathsf{S}, \mathsf{P}, \mathcal{A})$
- $\mathfrak{c} \in PC(prs)$

**Context-restricted cprs**: $cr\text{-}cprs = (cprs, ca)$ where:

- $cprs = (\mathsf{S}, \mathsf{P}, \mathcal{A}, \mathfrak{c})$
- $ca = (\mathsf{Q}, \mathsf{q}_0, \mathsf{R})$ – context automaton over $\mathcal{B}(\mathsf{S})$

$cr\text{-}cprs^{\leftarrow \mathtt{v}} = (cprs^{\leftarrow \mathtt{v}}, ca)$

# Parameter synthesis

- $\mathit{cr\text{-}cprs} = (\mathit{cprs}, \mathit{ca})$
- $F = \{\phi_1, \ldots, \phi_n\}$ – rsLTL formulae
- $\mathfrak{c}$ – parameter constraint

Calculate a valid parameter valuation $v$ of $\mathit{cr\text{-}cprs}$ such that:

$$(\mathcal{M}(\mathit{cr\text{-}cprs}^{\leftarrow v}) \models_\exists \phi_1) \wedge \cdots \wedge (\mathcal{M}(\mathit{cr\text{-}cprs}^{\leftarrow v}) \models_\exists \phi_n)$$

**Theorem.** The problem whether there is a valid parameter valuation is PSPACE-complete.

<u>Incremental approach</u>:

Keep increasing $k \geqslant 0$ until a valid parameter valuation is found:

$$(\mathcal{M}(\mathit{cr\text{-}cprs}^{\leftarrow v}) \models_\exists^k \phi_1) \wedge \cdots \wedge (\mathcal{M}(\mathit{cr\text{-}cprs}^{\leftarrow v}) \models_\exists^k \phi_n)$$

# Encoding of parameter synthesis into SMT

$$f_{\mathbf{ps}} = \left( \bigwedge_{\phi_f \in F} \mathsf{Paths}_f^k \wedge \mathsf{Loops}_f^k \wedge [\![\phi_f]\!]_0^k \right) \wedge \mathsf{PC}(\overline{\mathbf{p}}^{par})$$

1. Test satisfiability of $f_{\mathbf{ps}}$
2. When $f_{\mathbf{ps}}$ is *SAT* $\rightarrow$ extract valuation of parameters
3. When $f_{\mathbf{ps}}$ is *UNSAT* $\rightarrow$ no valid valuation exists

# Experimental evaluation

- Incremental approach: unrolling of interactive processes
- Two implementations:
  - **Parametric**:
    - with SMT encoding allowing for parameter synthesis
  - **Non-parametric** – using different SMT encoding (optimised for non-parametric verification)
- Using Python and Z3 SMT-solver (4.5.0)

# Mutex

- $n \geqslant 2$ processes
- competing for exclusive access to critical section
- Background set: $S = \bigcup_{i=1}^{n} S_i$:
  - $i$-th process: $S_i = \{out_i, req_i, in_i, act_i, lock, done, s\}$
  - $lock$, $done$, $s$ – shared amongst all the processes
- Reactions: $A = \bigcup_{i=1}^{n} A_i \cup \big\{(\{lock\}, \{done\}, \{lock\})\big\}$
  - $A_i$ is the set of reactions associated with the $i$-th process
- Context automaton provides:
  - the initial context set
  - context sets – at most two active processes allowed

# Mutex

- Assumption: open system
- $n$-th process: additional (malicious) reaction with parameters: $P = \{\lambda_r, \lambda_i, \lambda_p\}$

$$cr\text{-}cprs_M = ((S, P, A \cup \{(\lambda_r, \lambda_i, \lambda_p)\}, \mathfrak{c}), ca)$$

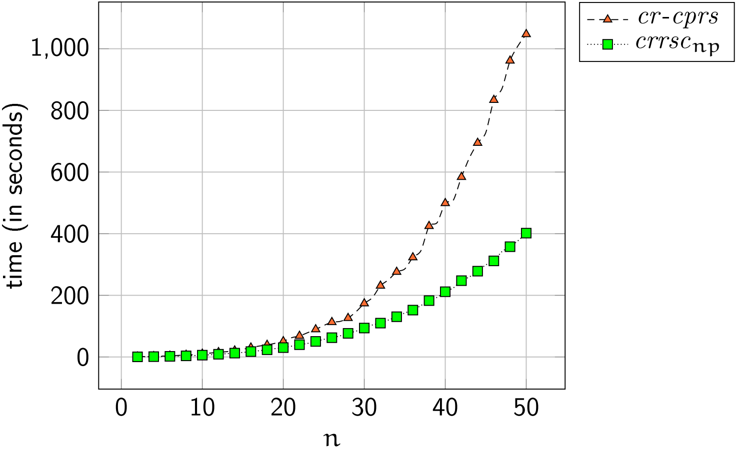
- $\mathfrak{c} = (\lambda_p[in_n] = 0) \wedge \bigwedge_{\lambda \in P, e \in S \setminus S_n} (\lambda[e] = 0)$ – additional reaction:
  - produces only entities related to the $n$-th process
  - cannot produce $in_n$ (to avoid trivial solutions)

**Synthesis**: parameter valuation $v$ of $cr\text{-}cprs_M$:

- $\phi = \mathbf{F}(in_1 \wedge in_n)$ – violation of mutual exclusion

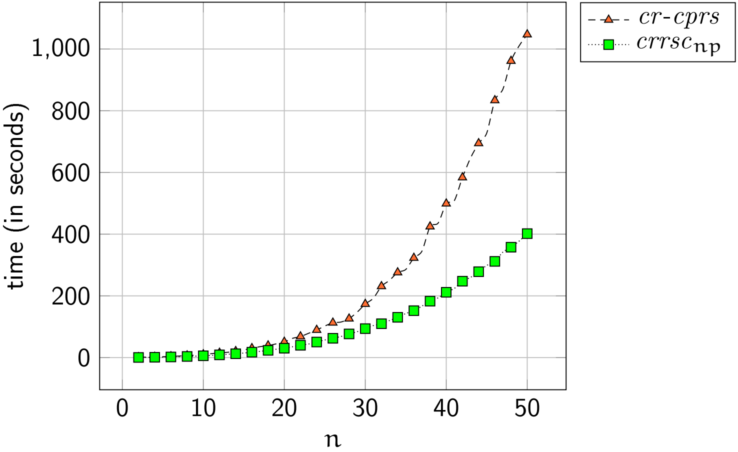
$$\mathcal{M}(cr\text{-}cprs_M^{\leftarrow v}) \models_\exists \phi$$

# Results: time



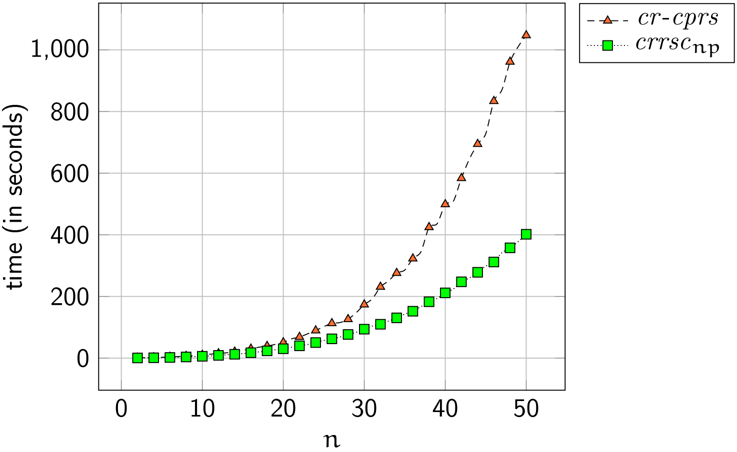

$\lambda_r^{\leftarrow v} = \{out_n\}$, $\lambda_i^{\leftarrow v} = \{s\}$, and $\lambda_p^{\leftarrow v} = \{req_n, done\}$

# Results: time



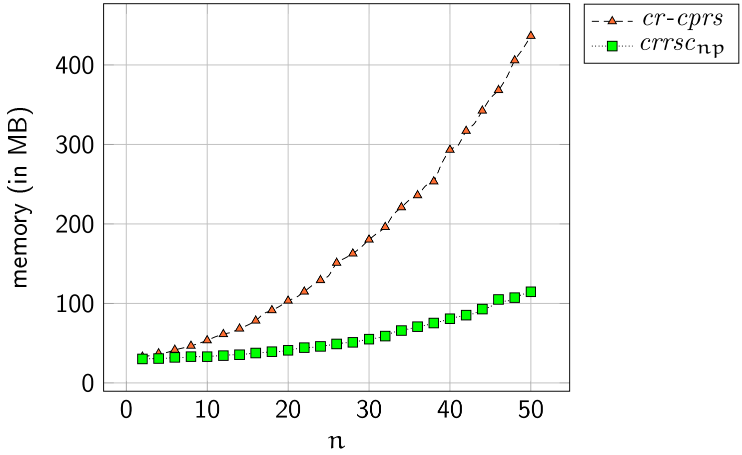

$cr$-$cprs$ – parametric implementation

# Results: time




$crrsc_{\mathrm{np}}$ – non-parametric (hard-coded valuation)

# Results: memory



# Reaction Systems Model Checking Toolkit

✓ rsCTL:

- Binary Decision Diagrams used for storing and performing operations on Boolean functions
- Uses BDD-based bounded model checking for efficient verification of existential formulae

✓ rsLTL:

- Based on translation to the SAT problem (SMT)
- Existential verification

✓ Reaction mining for rsLTL:

- Observations expressed in rsLTL
- Uses SMT for BMC-based parameter synthesis

# Reaction Systems Model Checking Toolkit

| Formalism | rsCTL | rsLTL |
|---|---|---|
| rs | umc/bmc | bmc |
| rsc | ✗ | bmc |
| prs | ✗ | bmc |

# Conclusions

- Synthesis method for partially defined reaction systems (RS)
- Properties specified using linear-temporal logic for RS
- Demonstrated application in attack synthesis

Further work:

- Tackle universal observations
- Optimisation of SMT-encoding

# Conclusions

- Synthesis method for partially defined reaction systems (RS)
- Properties specified using linear-temporal logic for RS
- Demonstrated application in attack synthesis

Further work:

- Tackle universal observations
- Optimisation of SMT-encoding

`http://reactionsystems.org`

# Conclusions

- Synthesis method for partially defined reaction systems (RS)
- Properties specified using linear-temporal logic for RS
- Demonstrated application in attack synthesis

Further work:

- Tackle universal observations
- Optimisation of SMT-encoding

`http://reactionsystems.org`

Thank you!