Rewriting Modulo SMT Techniques for Parametric Analysis

Carlos Olarte

LIPN-Université Sorbonne Paris Nord.

Work based on

Rewriting Logic Semantics and Symbolic Analysis for Parametric Timed Automata

Authors: Jaime Arias, Kyungmin Bae, Carlos Olarte, Peter Csaba Ölveczky, Laure Petrucci, Fredrik Rømming Authors Info & Claims

FTSCS 2022: Proceedings of the 8th ACM SIGPLAN International Workshop on Formal Techniques for Safety-Critical Systems • November 2022 • Pages 3–15 • https://doi.org/10.1145/3563822.3569923

International Conference on Applications and Theory of Petri Nets and Concurrency

↳ PETRI NETS 2023: **Application and Theory of Petri Nets and Concurrency** pp 369–392

Home › Application and Theory of Petri Nets and Concurrency › Conference paper

Symbolic Analysis and Parameter Synthesis for Time Petri Nets Using Maude and SMT Solving

Jaime Arias, Kyungmin Bae, Carlos Olarte, Peter Csaba Ölveczky, Laure Petrucci & Fredrik Rømming

ELSEVIER

Science of Computer Programming

Volume 233, March 2024, 103074

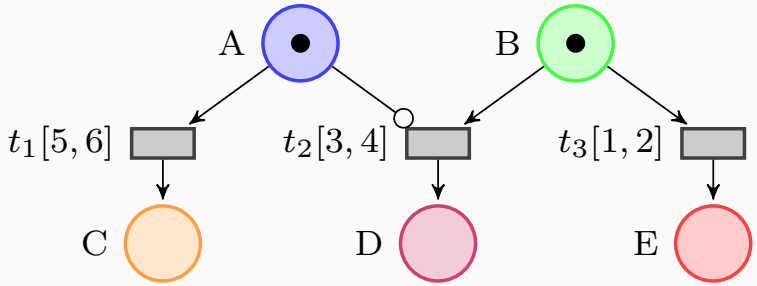
Symbolic analysis and parameter synthesis for networks of parametric timed automata with global variables using Maude and SMT solving

Jaime Arias [a], Kyungmin Bae [b], Carlos Olarte [a], Peter Csaba Ölveczky [c], Laure Petrucci [a], Fredrik Rømming [d]

Joint work with: Jaime Arias, Kyungmin Bae, Peter Csaba Ölveczky, Laure Petrucci and Fredrik Rømming.

Motivation: Verification of real-time systems

- Timed automata
- Time Petri nets



Pros

- Decidable fragments
- Efficient verification procedures
- Extended with parameters.

Cons

1. No support for user-defined data types
2. No support for other forms of communication and dynamic object creation/deletion

Motivation: Verification of real-time systems

Rewriting logic / Maude

```
mod SYSTEM
  eq t = t' .
  rl l => r if C .
  ...
```

Pros

- Very expressive and general
- User-defined data types
- Large applications
- Executable specification
- Maude system: full LTL model checking, reachability, ...

Cons

- Most analysis problems are undecidable
- Explicit-state analysis of real-time theories is unsound for dense time

Contributions

- Interpreter: Executable rewrite semantics for parametric timed Automata (PTA) and parametric time Petri nets with inhibitor arcs (PITPN)
- Sound and complete symbolic analysis (Rewriting + SMT)
- Analysis methods: Reachability, parameter synthesis, model checking
- New analysis:
  - Full LTL model checking
  - Executions with strategies
  - Synthesis also of initial markings
- Novel folding procedure (termination)
- Long term goal: Symbolic analysis of real-time rewrite theories

Outline

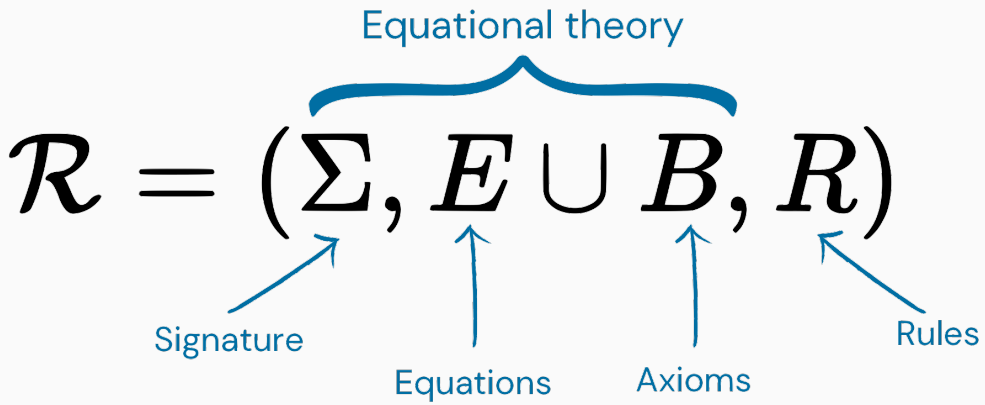
1. Rewriting logic
2. PTAs
3. PITPNs
4. Back to PTAs
5. Concluding Remarks

Plan

1. **Rewriting logic**
2. PTAs
3. PITPNs
4. Back to PTAs
5. Concluding Remarks

Rewriting Logic

- Computational logic: concurrent computation + logical deduction



- States are terms modulo $E \cup B$
- Rules (`crl l => r if C`) in R define system transitions

Rewriting Logic

- Equational theory (algebraic data types): defining states
- Rewriting rules: behavior of the system
- Executable specification

Maude

- A high-performance rewriting logic engine
- Executes admissible theories (confluence and termination of *E*, coherence of *R* w.r.t. *E*, ...)
- Several generic formal analysis tools (rewrite, search, LTL model checker, narrowing, SMT, etc).

Plan

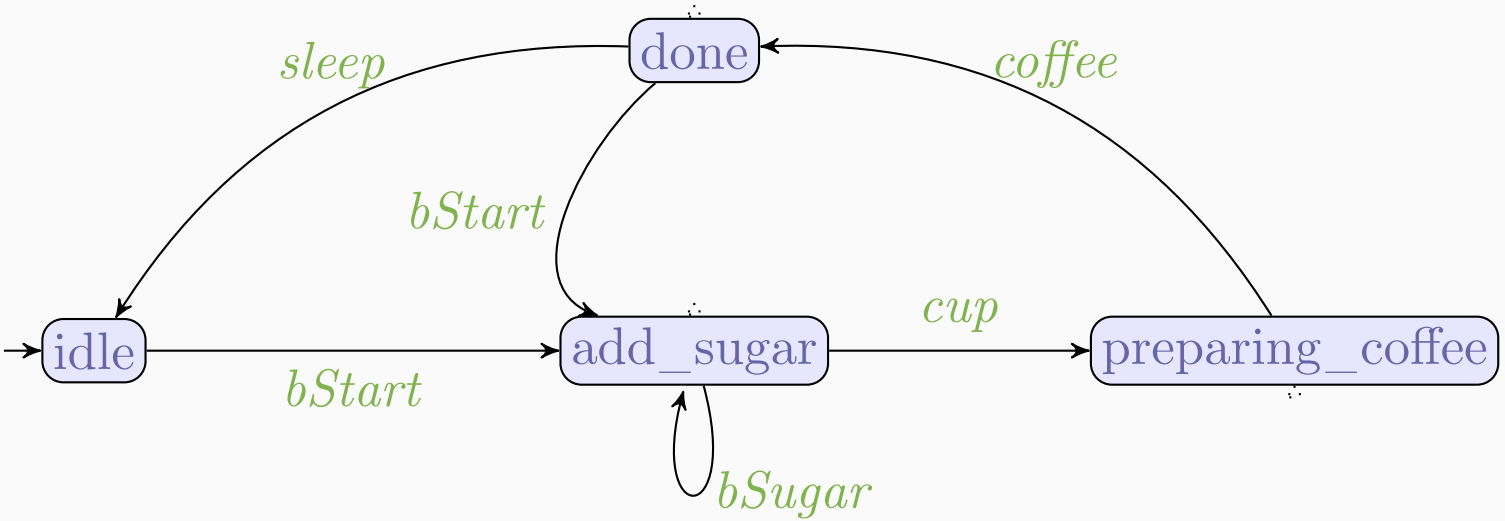
Parametric Verification

Models: not all the choices, components, response times, are known.

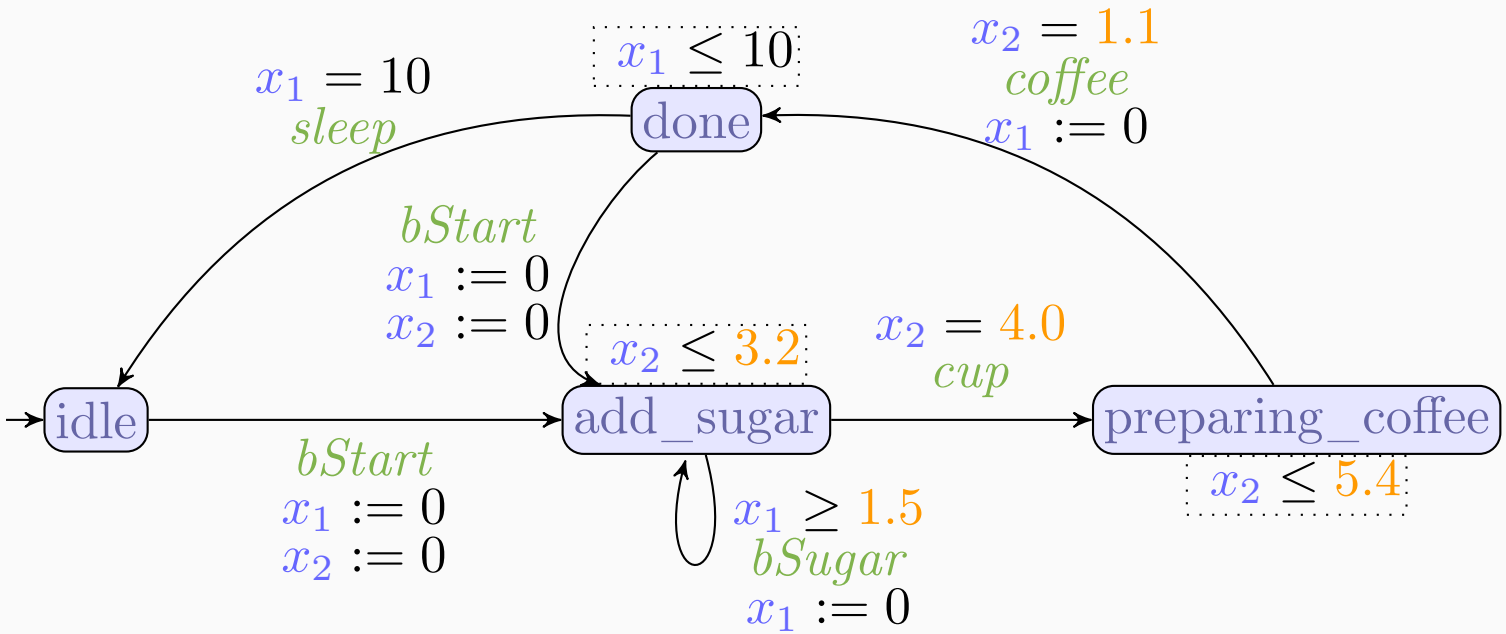
Parameters

- Flexibility
- Avoid verifying the system when the unknown components change
- Central problem: Parameter synthesis

Automata


done
sleep
coffee
bStart
idle
add_sugar
cup
preparing_coffee
bStart
bSugar

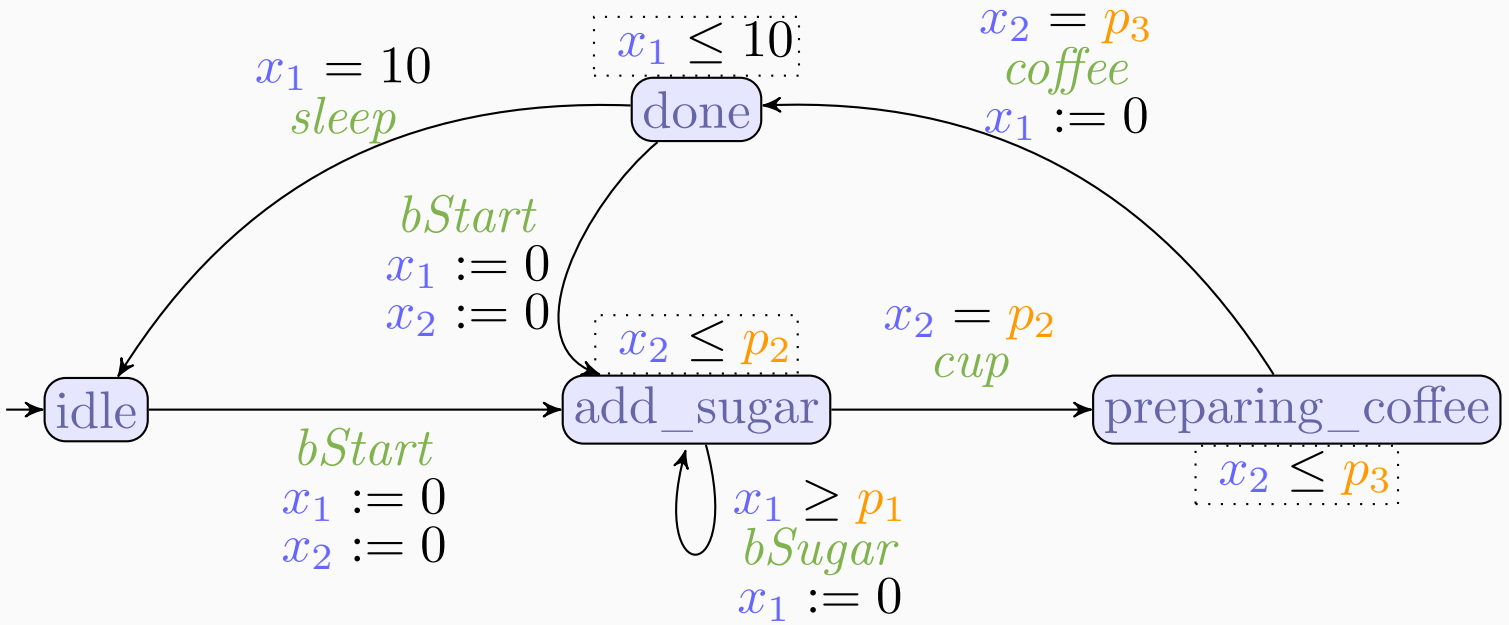
Timed Automata



clocks and constants

Constraints for invariants and guards.

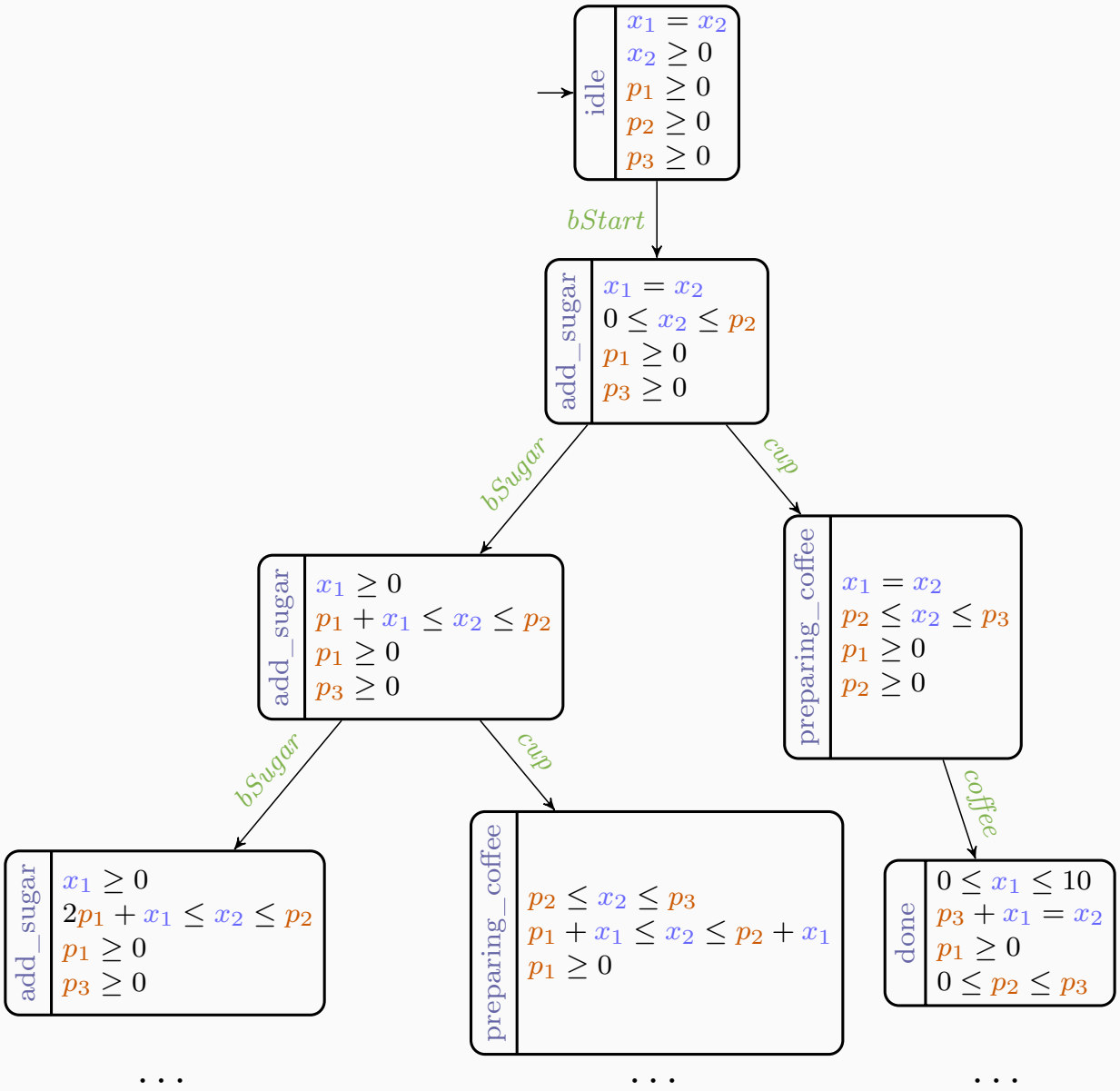
Parametric Timed Automata



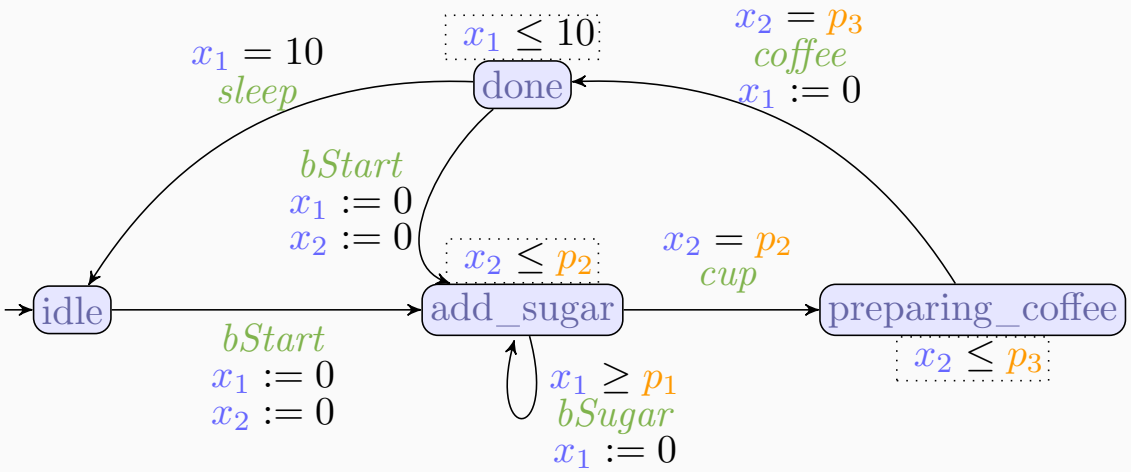
clocks and parameters

Synthesis: If $p_2 \leq p_3$ we can have a coffee!

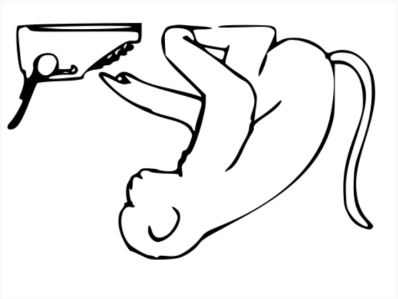
Parametric Zone Graph (PZG)



Parametric Timed Automata



Imitator



- Timed model checking
- Parameter synthesis with dedicated algorithms
- Restricted to PTA-based systems

PTA/Imitator vs Rewriting Logic/Maude

PTA/Imitator

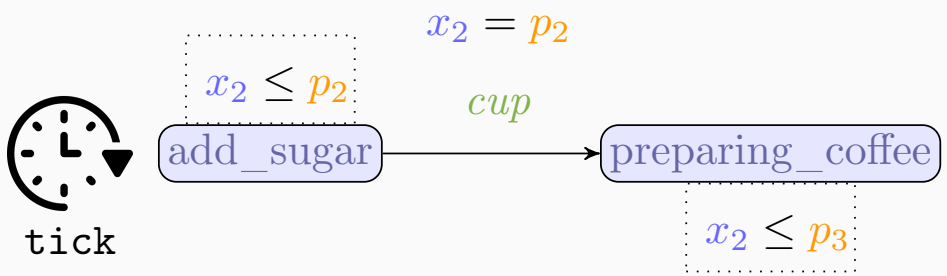
- Heuristic, optimizations and approximation techniques.
- Decidable fragments

RL/Maude

- More expressive (alg. data types)
- Undecidable in general
- No parametric analysis

Learn and take the best from both worlds!

RW semantics for PTA: The tick rule problem

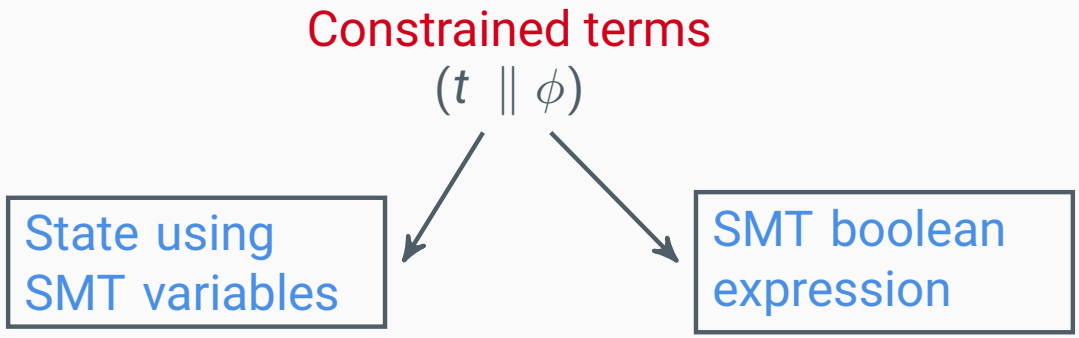



```
vars T P1 P2 P3 : PosRat .
  crl [tick] :
      [ add_sugar : X1 ; X2 ] < P1 ; P2 ; P3 > =>
      < add_sugar : X1 + T ; X2 + T > < P1 ; P2 ; P3 >
      if (X2 + T <= P2 and T >= 0/1) = true [nonexec] .
```

- What is the value of `T`? Not executable rule!
- Parameters P_1, P_2, P_3 are indeed constants (ground rewriting)

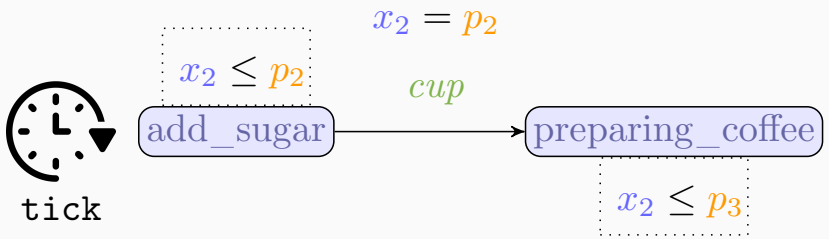
Solution: Symbolic techniques (Rewriting Modulo SMT)

Symbolic semantics: rewriting modulo SMT



- $[\![(t \parallel \phi)]\!]$: possibly infinite set of concrete states (instances of t)
- Symbolic rewrite relation $\rightsquigarrow$: adding constraints + checking satisfiability
- A single symbolic transition captures all possible delays!

RW semantics for PTA



Symbolic states: $t \parallel \phi$ (term + SMT boolean expression)

```
var T P1 P2 P3 : RExpr .  --- SMT Real Variables
crl [add_sugar-tick] :
    [ add_sugar : X1 ; X2 ] < P1 ; P2 ; P3 > =>
    < add_sugar : X1 + T ; X2 + T > < P1 ; P2 ; P3 >
    if smtCheck(X2 + T <= P2 and T >= 0/1) .
```

- Symbolic executions correspond to transitions of the PZG
- Executable in Maude-with-SMT

PTA in Maude: Verification techniques

- Sound and complete reachability analysis:

```
smt-search [1] < idle ; X ; y >   < P1 ; P2 ; P3 >  =>*
               < done ; X' ; X' > < P1 ; P2 ; P3 >
                such that ψ .
```

Parameter synthesis: accumulated constraint $t \parallel \phi$:

- EFϕ synthesis: reachability plus quantifier elimination $\exists X.\phi$
- Only available for the SMT solver Z3.
- AG$\neg\phi$: finding all solution (termination problem!) and negating the result.

Termination Problem

- `smt-search` stop exploring when it sees the same symbolic state.
- A new fresh variable is created when the tick rule is applied.
- No termination for negative queries.

new `reachability` command

Subsumption:

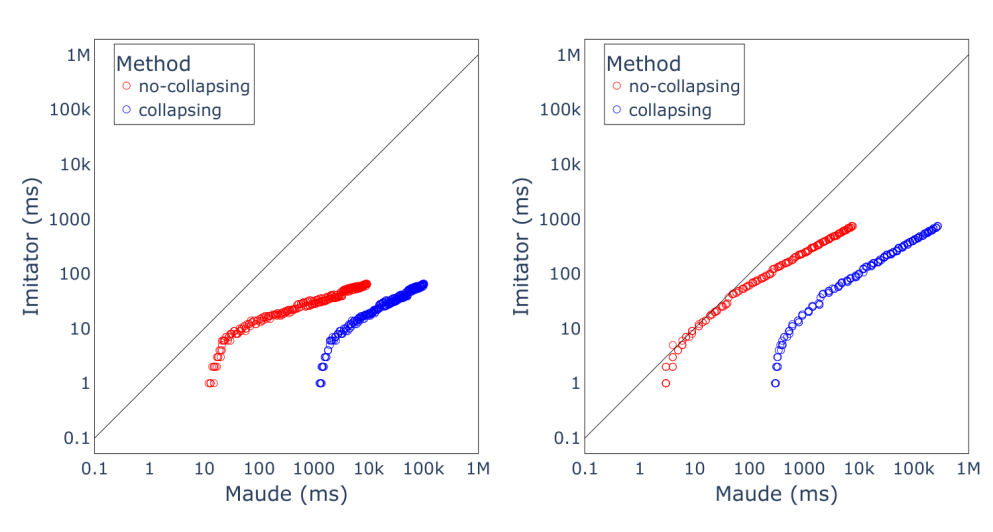
$$\phi_u \parallel t_u \sqsubseteq \phi_v \parallel t_v$$

iff there exists θ s.t. $t_u = t_v\theta$ and the implication $\phi_u \Rightarrow \phi_v\theta$ holds

Theorem

The command `reachability` terminates iff the PZG is finite.

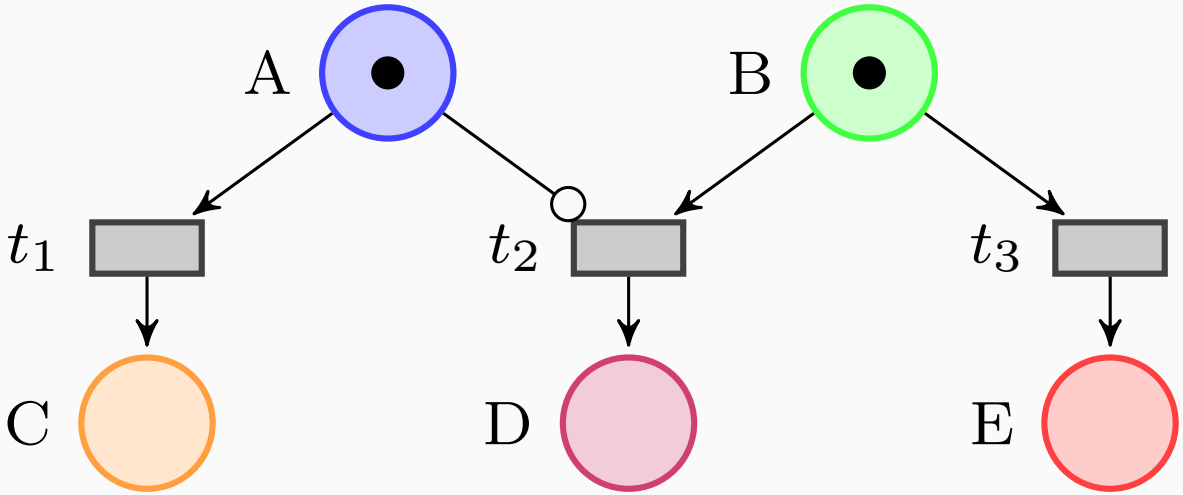
What we have so far...



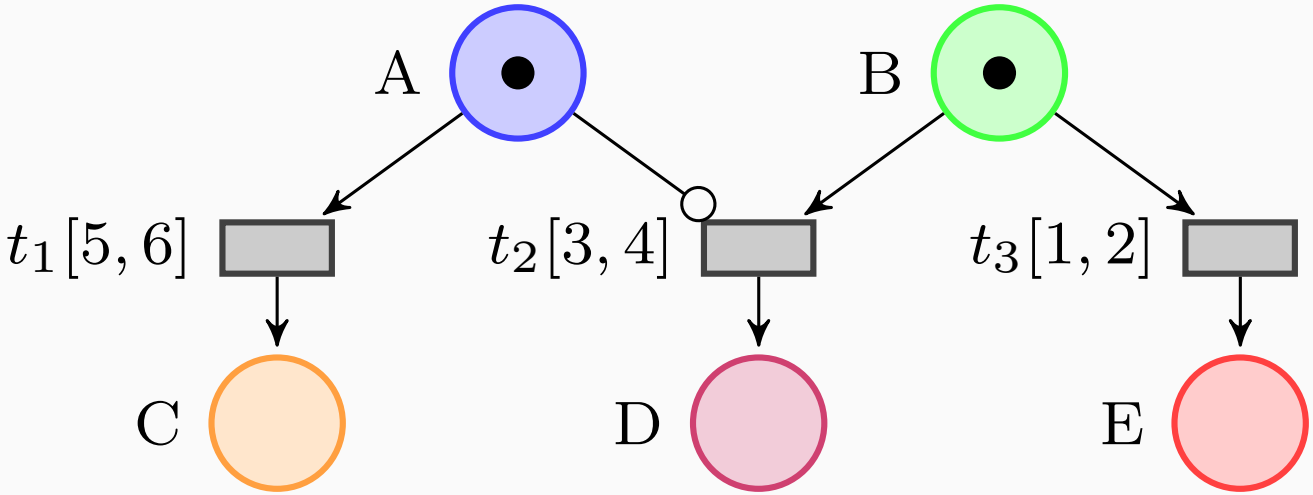
- Relatively simple model (“simple” states)
- Each PTA is compiled into a theory $\mathcal{R}$
- No equations in the theory (a requirement for smt-search)
- Sound and complete reachability analysis
- “Standard” folding was enough for guaranteeing termination
- Parameter synthesis only available with Z3.
- But this is not enough for networks of PTAs with global variables.....

Plan

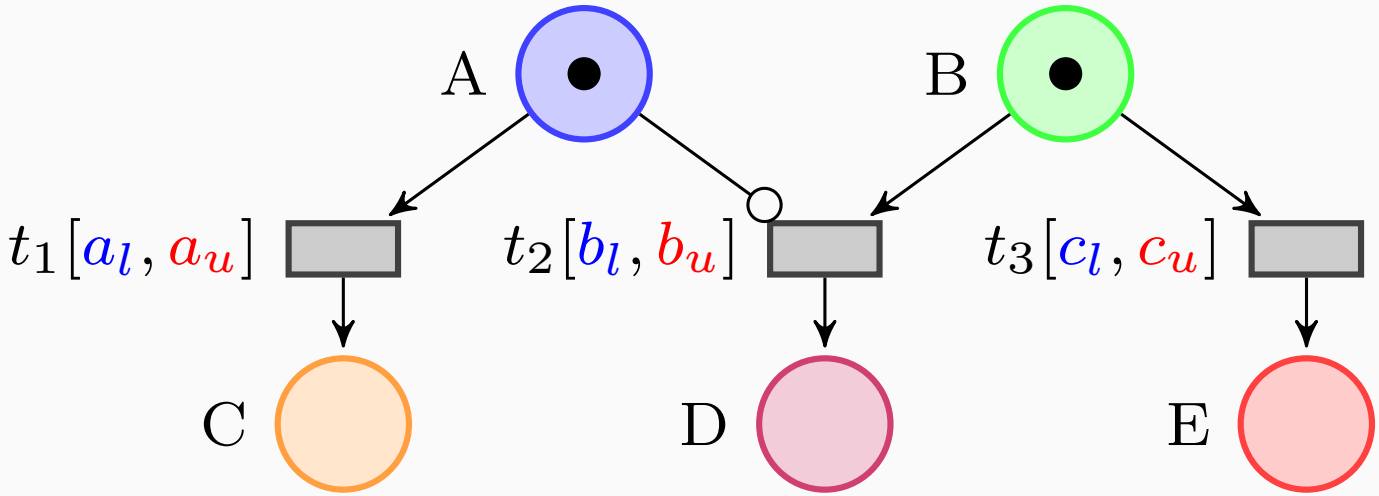
Petri net with inhibitor arcs



Time Petri net with inhinitor arcs



PITPN: Parametric time Petri net with inhibitor arcs



Synthesis problem: finding values for the parameters s.t. certain property holds

Semantics

Concrete semantics: $\xrightarrow{\delta}$; $\xrightarrow{t}$

- $\xrightarrow{\delta}$: time advances $\delta \in \mathcal{R}_+$ and intervals are updated
- δ is constrained so that no enabled transition is missed
- $\xrightarrow{t}$: updates the marking and newly enabled transitions are reset

Symbolic semantics: $\overset{\delta}{\Rightarrow}$; $\overset{t}{\Rightarrow}$

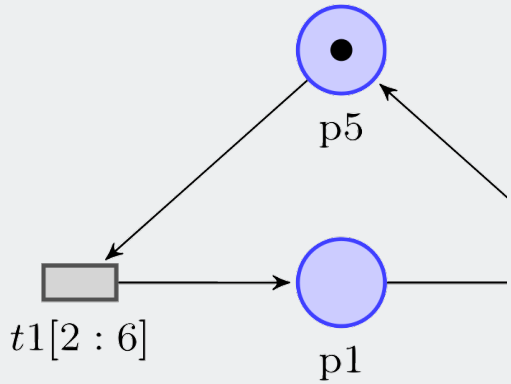
- State classes (M, D)
- D is a constraint (conjunction of inequalities) on parameters Λ

Representing PITPNs in Maude

Syntax and representation: $[\![\mathcal{N}]\!]$

```
--- Sorts and constructors
sorts Label Place Marking ...
op _|->_ : Place Nat -> Marking [ctor] .

--- Example of a concrete PITPN (no parameters)
  "t1" : "p5" |-> 1 -->
         "p1" |-> 1
         inhibit empty in [2 : 6]
```



Dynamics: transition $\xrightarrow{t}$

```
 crl [applyTransition] :
    M  :  (L -> T) ; CLOCKS  :
    (L : PRE --> POST inhibit INHIBIT in INTERVAL) ; NET
=>  (M - PRE) + POST :
    L -> 0 ; updateClocks(CLOCKS, M - PRE, NET) :
    (L : PRE --> POST inhibit INHIBIT in INTERVAL) ; NET'
if active(M, L : PRE --> POST inhibit INHIBIT in INTERVAL)
   and (T in INTERVAL) .
```

Representing PITPNs in Maude

The tick rule: delay transition $\xrightarrow{\delta}$

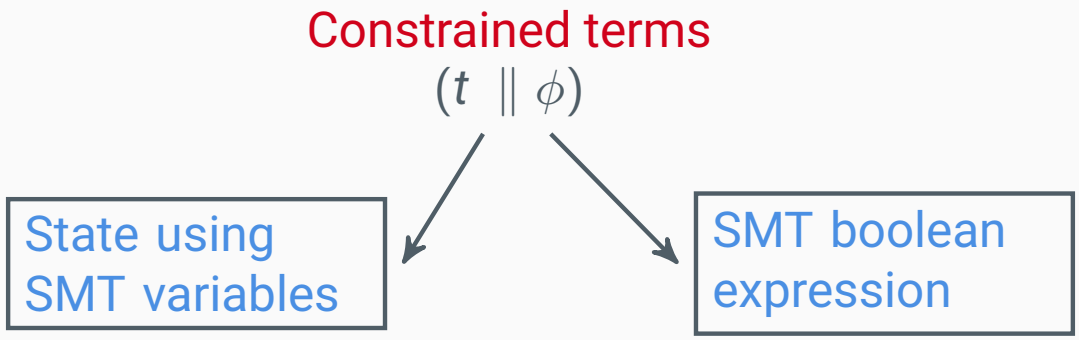
```
crl [tick] : M : CLOCKS : NET  =>  M : increaseClocks(M, CLOCKS, NET, T) : NET
if  T <= mte(M, CLOCKS, NET) [nonexec] .
```

- Enabled transitions are not missed (predicate *mte*)
- Non executable : *T* needs to be sampled

Theorem (Bisimulation)

For any PITPN $\mathcal{N}$, the transition system induced by the rewrite relation in theory $[\![\mathcal{N}]\!]$ is bisimilar to the concrete semantics of $\mathcal{N}$.

Symbolic semantics: rewriting modulo SMT



- $[\![(t \parallel \phi)]\!]$: possibly infinite set of concrete states (instances of t)
- Symbolic rewrite relation $\rightsquigarrow$: adding constraints + checking satisfiability
- A single symbolic transition captures all possible delays!

```
crl [tick] :  tickOk    : M : CLOCKS     : NET => ....
           =>  tickNotOk : M : increaseClocks(M, CLOCKS, NET, T) : NET
if (T >= 0 and mte(M, CLOCKS, NET, T)) .
```

Theorem (Adequacy)

For any PIPTPN $\mathcal{N}$, the symbolic semantics of $\mathcal{N}$ corresponds to $\rightsquigarrow$-transitions.

Termination problem

- Each application of rule `tick` creates a fresh SMT variable
- Maude+SMT analyses do not terminate (even if the PZG is finite)
- Standard subsumption relation is not sufficient

New folding procedure

- New relation $\preceq$: based on matching + existential quantifier elimination
- Let U, V be two symbolic states and $U \Downarrow_{now} = t_u \parallel \phi_u$, $V \Downarrow_{now} = t_v \parallel \phi_v$

$$U \preceq V \text{ iff } t_u = t_v\theta \text{ and } \exists(U \Downarrow_{now}) \Rightarrow \exists(V \Downarrow_{now})\theta$$

- $\exists(U \Downarrow_{now})$ hides the information about ticks.
- Soundness and completeness: $[\![U]\!] \subseteq [\![V]\!]$ iff $U \preceq V$
- Collapsing states: $M \mapsto C_1 \vee \cdots \vee C_n$

Theorem (Termination)

If the symbolic transition system for $\mathcal{N}$ is finite, then so is the resulting symbolic rewrite relation with $\preceq$-folding.

Analyses

EF-synthesis (**EF** ϕ). Standard Maude's search

```
search [1] init(net, m0, φ) =>* S : PHI' || ( TICK : M : CLOCKS : NET )
       such that smtCheck(PHI' and not k-safe(1,M)) .
```

Safety synthesis (**AG**($\neg\phi$)). Search + Folding

```
safety-syn(net, m0, a:Real >= 30/1 and a:Real <= 70/1, k-safe(1,M)) .
```

Strategies. Rewriting + Strategy

New analysis: What happens if t_3 has a higher priority?

```
t3-first := ( applyTransition[L <- "t3"] or-else all )!
```

Analyses

Model Checking. (Search + Maude's LTL model checker)

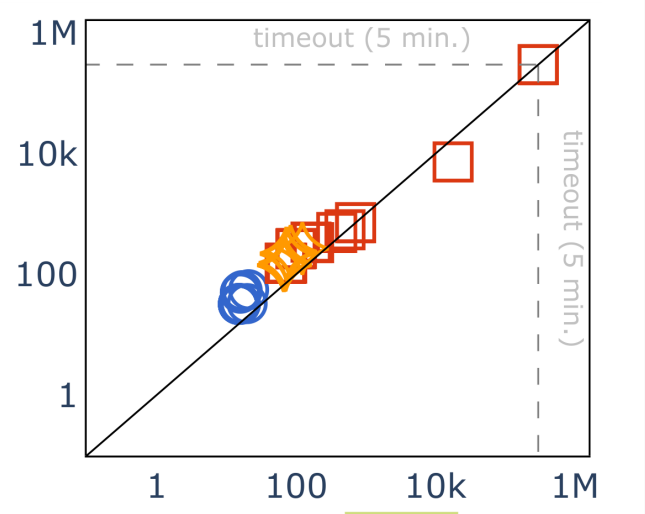
- The TCTL fragment $\exists \mathsf{F}_J\, \phi \mid \forall \mathsf{G}_J\, \phi \mid \phi \rightsquigarrow_{\leq b} \psi$ can be checked with search + folding
- The TCTL fragment $\mathbf{Q}\, \phi\, \mathsf{U}_J\, \psi \mid \forall \mathsf{F}_J\, \phi \mid \exists \mathsf{G}_J\, \phi$ can be checked using Maude's LTL model checker (+ some theory transformations)
- New analysis: full LTL model checking is available to PITPNs

Parametric initial marking

- The number of tokens at a place *p* is an SMT integer variable
- New analysis: Initial marking synthesis

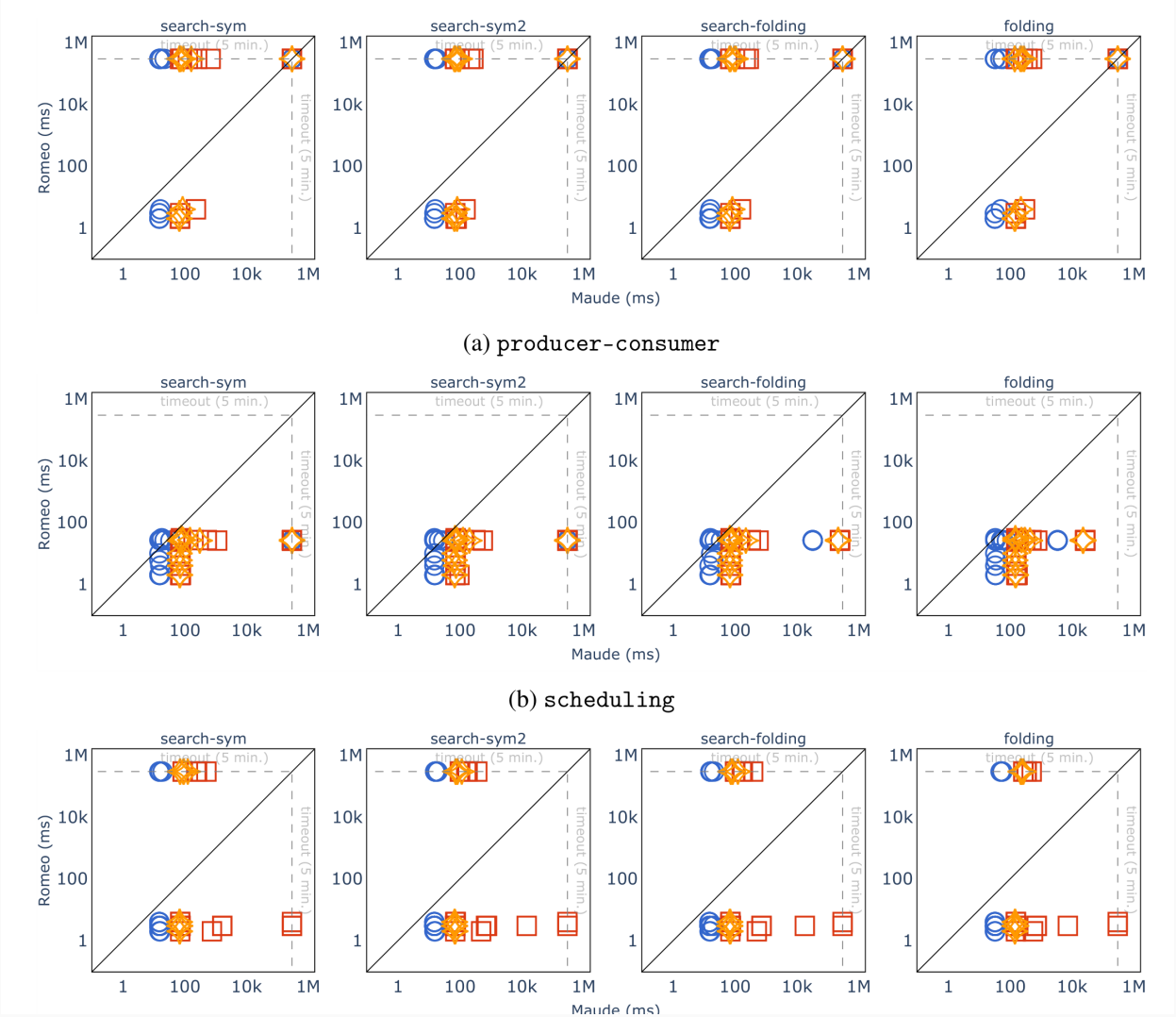
Folding and synthesis

- Quantifier elimination is needed for folding and synthesis.
- Z3 was the only option available for $\exists X.\phi$... but...



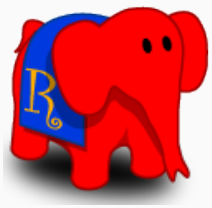
- We have implemented the FME procedure in Maude
- Yices2 (the fastest in our benchmarks) can be used now!

Benchmarks



(a) `producer-consumer`

(b) `scheduling`

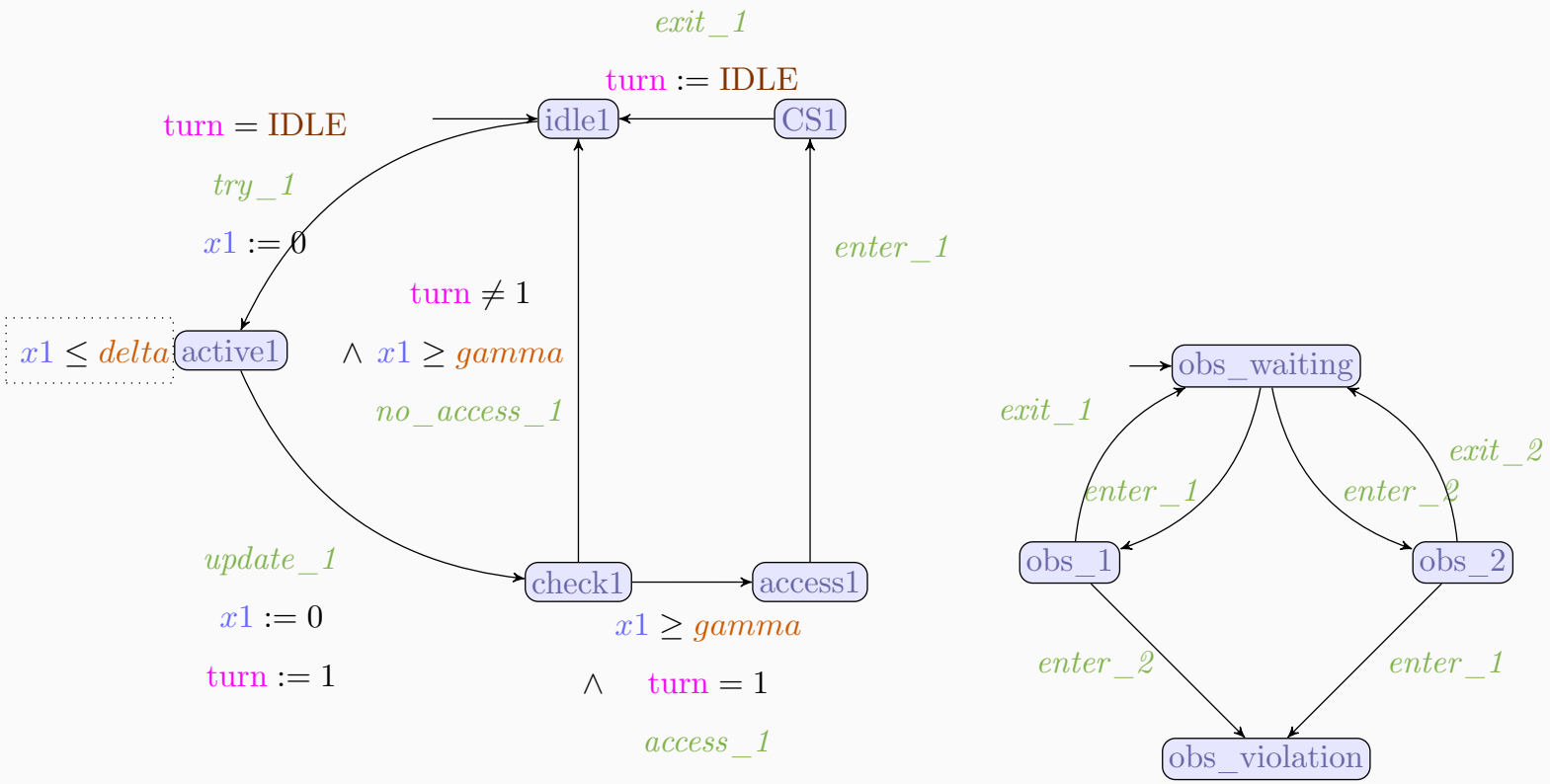


What we have so far...

- Equations are part of the theory and `smt-search` cannot be used directly.
- Standard subsumption does not work.
- A new folding eliminating tick variables is needed.
- Our benchmarks look much better with our own FME procedure (Yices2).

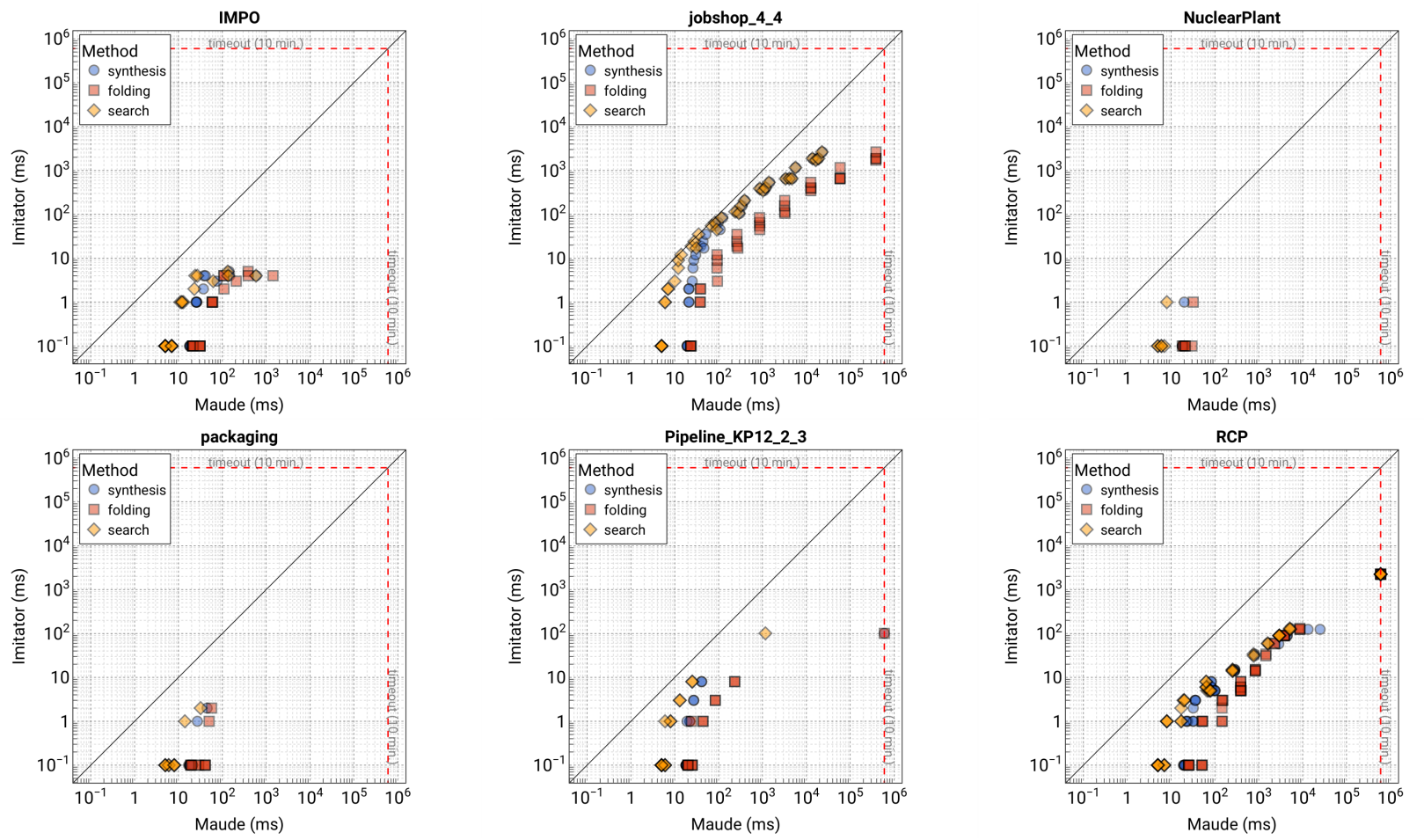
Plan

Networks of PTAs with Variables



- We need an interpreter (product of automata)
- However, the interpreter is not as fast as the “compiled” PTA.

Imitator vs Maude



Plan

Concluding Remarks

- Executable symbolic rewrite semantics for PTAs and PITPNs
- Sound and complete analyses: synthesis, reachability, TCTL model check
- New Analyses:
  - Full LTL model checking
  - Strategies
  - Synthesizing initial markings
- System for quick prototyping of new analysis methods
- Concrete steps for symbolic analysis of real-time rewrite theories
- New strategy language for real-time theories (WRLA'24).
- Full LTL/CTL (symbolic) model checking (in progress).

Thanks!

We acknowledge support from the PHC project Aurora AESIR and the NATO Science for Peace and Security Programme SymSafe.