TREX and IF

Petr Matoušek

matousp@fit.vutbr.cz

Brno University of Technology, Czech republic

1. Introduction

❖ **Talk outline**

1. **TREX - A Tool for REachability Analysis of CompleX Systems**
2. **Architecture**
3. **Input language**
4. **Extended Timed Automata**
5. **Issues on Parametric Analysis**
6. **Running TREX**
7. **Examples**

TREX overview

- **developed by A. Annichini-Collomb, A. Bouajjani and M. Sighireanu in Verimag, Grenoble and LIAFA, Paris**

❖ **Model checker that implements:**

- **real-time constraints**
- **counters**
- **communication through unbounded channels**
- **parametric reasoning**

TREX overview

❖ **Features:**

- **description based on parametrized (continuous-time) timed automata**
- **variables of infinite-domain data structures (PDMBs) with parameters**
- **analysis based on symbolic reachability analysis**
- **termination is not guaranteed but extrapolation can help it**
- **checking on-the-fly safety properties**
- **generates a set of reachable configurations and a finite symbolic graph**

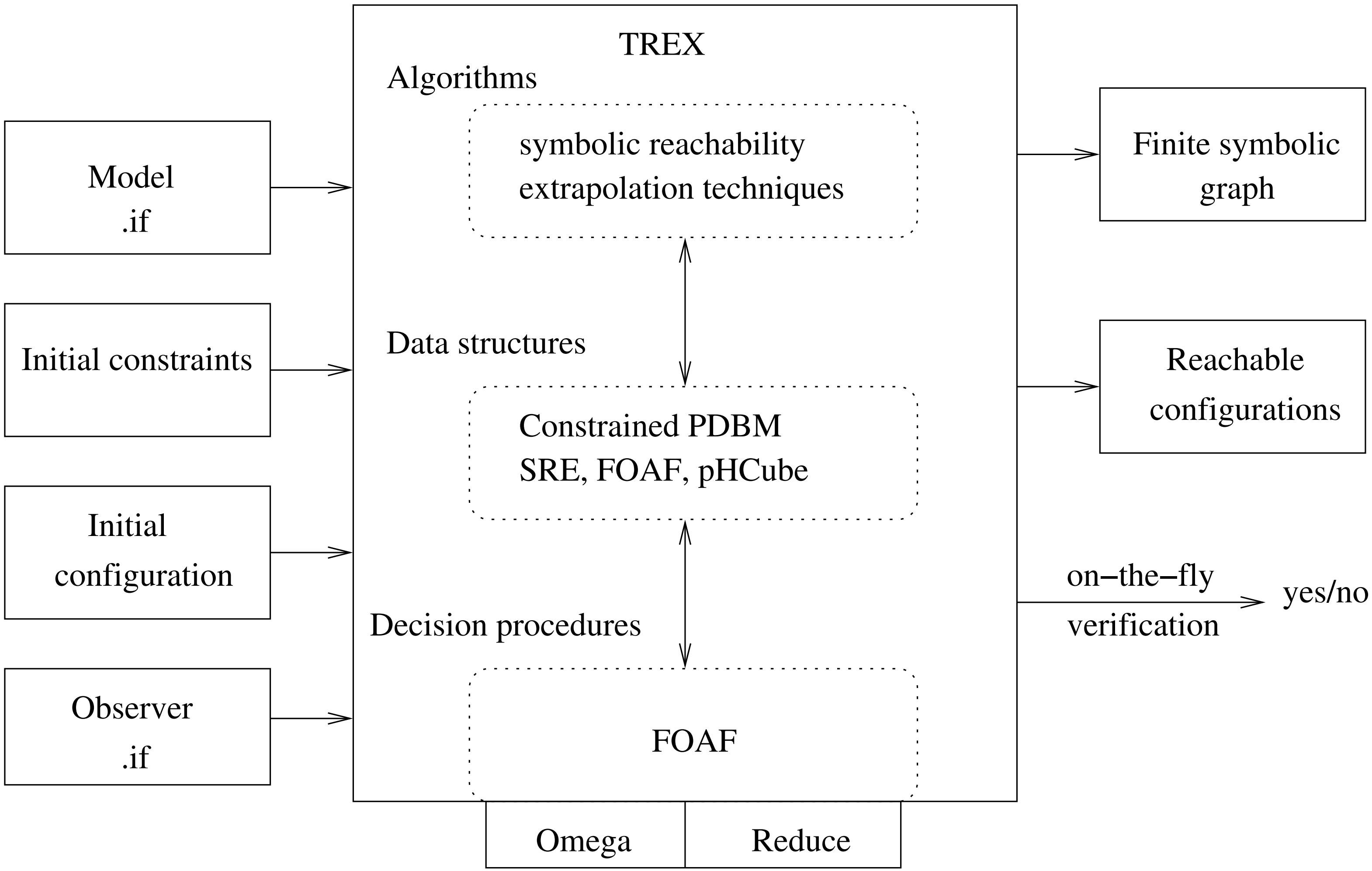
TREX overview

❖ Input models

- parametric (continuous-time) timed automata,
- extended with integer counters
- and finite-domain variables,
- and communicating through unbounded lossy FIFO channels
- and shared variables.

Architecture

- **Generic kernel algorithm for any kind of data structures**



Architecture

❖ Three modules

1. **algorithms — computes generic reachability states for every data structure; post(), pre() and extrapolation techniques**
2. **data structure — several symbolical data structure; PDBMs, SRE, FOAF**
3. **decision procedures — satisfiability of formulas is checked using external decision procedures.**
   - **for formulas over integers** TREX **uses** OMEGA
   - **for formulas over reals** TREX **uses** REDUCE

Architecture

❖ **Data Representation**

- **simple regular expressions (SRE) - manipulating SREs during symbolic analysis; it is used for lossy FIFO channels**
- **Constrained PDBMs - counter and clock automata and their manipulation**
- **First-Order Arithmetical Formulas (FOAF) - terms and formulas used in Constrained PDBMs; used for linear and non-linear constraints on parameters**

Input language

❖ IF version 1.0

- **declaration of a system**

```
system lift;
```

- **declaration of signals**

```
signal overflow;
       empty;
```

- **declaration of variables - predefined types: bool, int, real, clock**

```
var
c(1) : int;      // a counter
g(1) : int;      // a counter
N : int;         // a parameter
```

Input language

- **declaration of buffers**

```
buffer env :queue: toenv of overflow, empty;
```

- **synchronization between process**

```
gate
  channel1;
  channel2;

sync
 ((process1 | [channel1] | process2) | [channel2] | process3)
end;
```

Input language

- **declaration of processes**

```
process motor
state
   m_1 : init;
transition
   from m_1
     if a = 1
     do a:=0, c := c+1
   to m_1;

  from m_1
    sync channel1
    if a = 2
    do a := 0, c := c-1
  to -;
```

Input language

- **declaration of processes**

```
state
   receiving :init tpc(t<=0); end;
   sending tpc(t<=320); end;
transition
  from receiving
    do reset(t),
       no : = no + 1,
       output overflow to env
    to sending;
```

Input language

TREX **supports the following sub-set of IFv1.0:**

- **pre-defined types: pid, sid, int, real (float), clock**
- **user defined types: enum, range**
- **buffers: fifo lossy, finite fifo non lossy**
- **gates and n-ary rendez-vous**
- **invariants on states -** `tpc` **expressions**
- **transition: normal and eager**

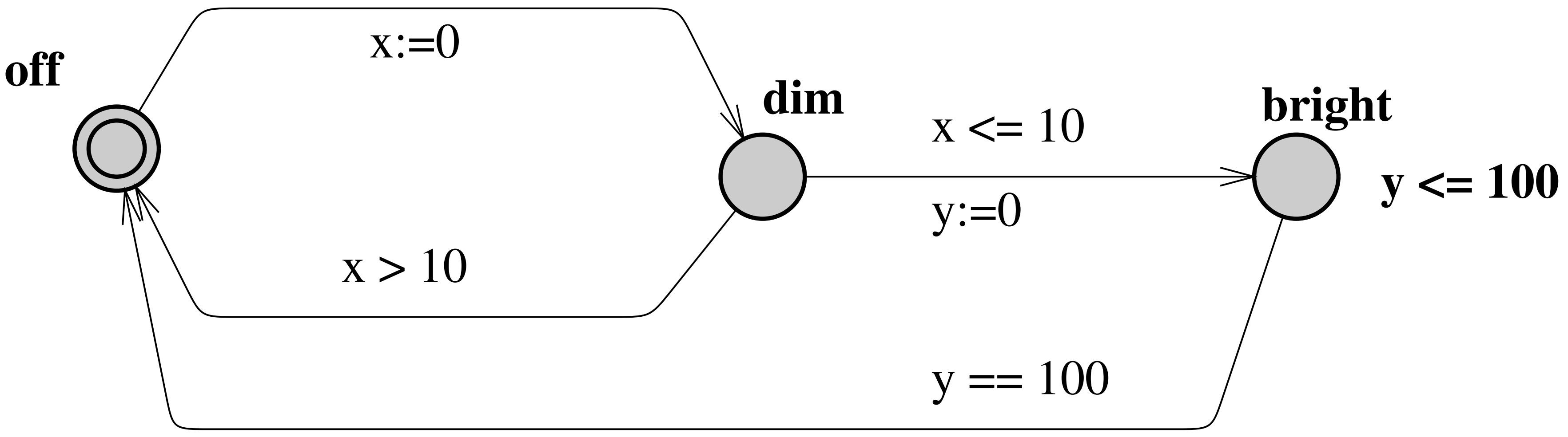
Input language

❖ Symbolic representations

type	*guards*	*assignment*	*package*	*analysis with* TREX
finite	usual	usual	explicit, BDD	back/forward
counter (int,nat)	$x - y \sharp t$	$x := y + t$	PDBMs	forward + interpolation
counter (int,nat)	$x \sharp t$	linear	intervals	-
counter (int,nat)	$+/- x + / - y \sharp t$	$x := +/- y + t$	octagons	-
clock	$x - y \sharp t$	$x := y + t$	PDBMs	forward + interpolation
clock	$x - y \sharp t$	$x := y + t$	RVA	-
clock	linear	linear	polyhedra, RVA	-
lossy fifos	input	output	SRE	forward + widening
lossy fifos	input	output	UPC	backward
parameters	convex	none	STREE	
parameters	linear over integers	none	NDD	
parameters	linear over reals	none	RVA	

Timed Automata - Introduction

- An example: light-switch



- States: off, dim, bright
- Clocks: x,y

Timed Automata - Syntax

- **Timed automaton is a tuple $A = \langle L, L^0, \Sigma, X, I, E \rangle$, where**
  - L **is a finite set of locations,**
  - L^0 **is a finite set of initial locations,**
  - Σ **is a finite set of labels,**
  - X **is a finite set of clocks,**
  - I **is a mapping that labels each locatins s with some clock constraint $\Phi(X)$, and**
  - $E \subseteq L \times \Sigma \times 2^X \times \Phi(X) \times L$ **is a set of switches.**

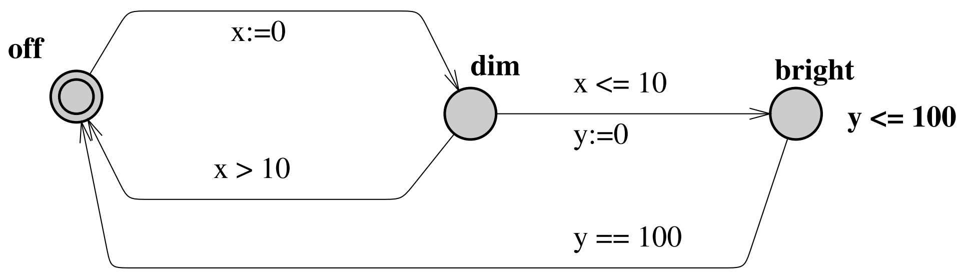
- **Clock constraint**

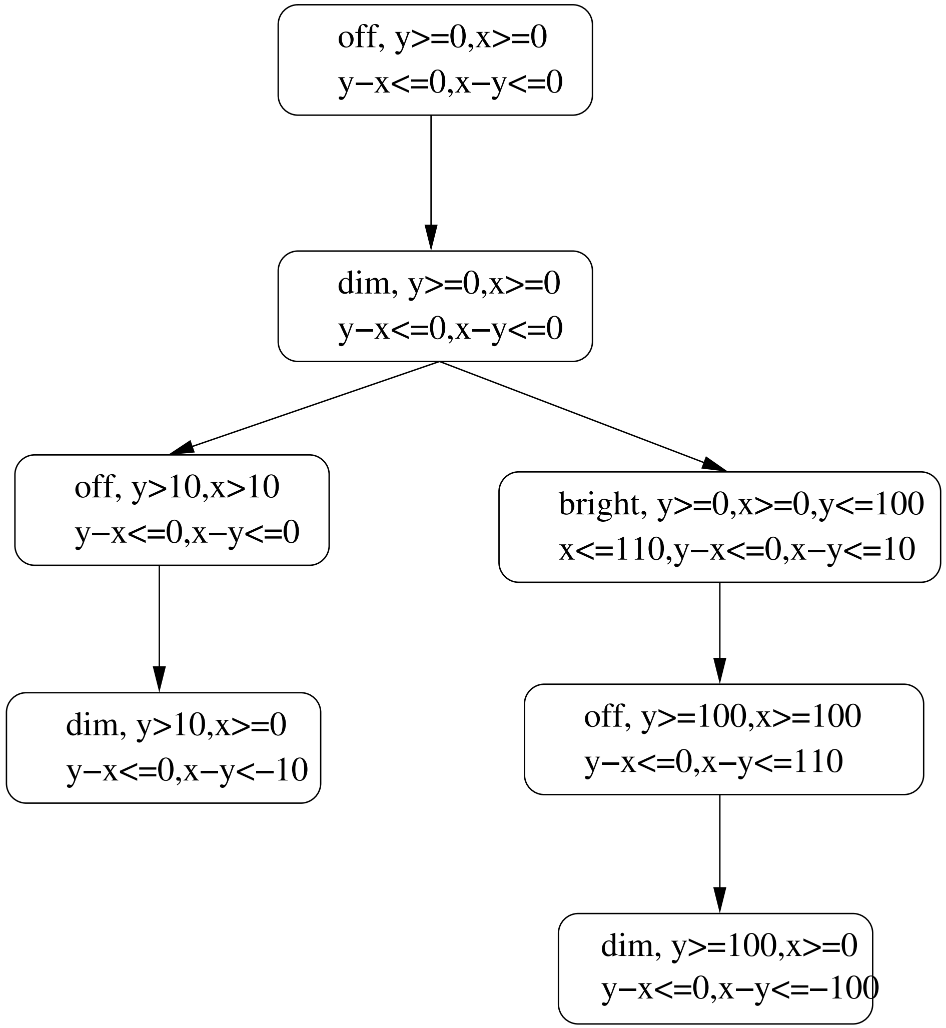
$$\varphi(X) := x \leq c \mid c \leq x \mid x < c \mid c < x \mid \varphi_1 \wedge \varphi_2,$$

where x is a clock in X and c is a constant in $\mathbb{Q}$.

Timed Automata - Semantics, exam.

- **Configuration graph**





Time Automata - Semantics

- **Semantics of a TA is a transition system $S_A = (\Sigma, Q, Q_0, R)$, where**
  - Σ **is a finite set of labels**
  - **Q is a set of states; each state is a pair (s, ν)**
  - Q_0 **is a set of initial states**
  - **R is a transition of type:**

    - **delay transition** $\boxed{(s,\nu) \xrightarrow{\delta} (s, \nu + \delta), \delta \geq 0}$

    for all $0 \leq \delta' \leq \delta, \nu + \delta'$ satisfies the invariant $I(s)$.

    - **action transition** $\boxed{(s,\nu) \xrightarrow{a} (s', \nu[\lambda := 0])}$

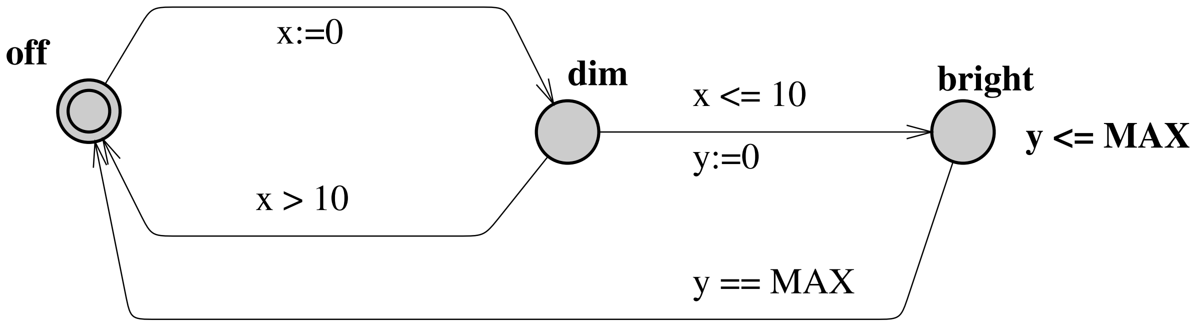
    where $a \in \Sigma$ and transition $\langle s, a, \varphi, \lambda, s' \rangle$ satisfies φ.

Timed Automata - Verification

- **The semantics is the basis for verification of TA.**
- **Reachability problem of S_A**
  - **We will write $(s, \nu) \rightarrow (s', \nu')$ if there is $(s, \nu) \xrightarrow{\sigma} (s', \nu')$ for $\sigma \in \Sigma \cup \mathbb{R}_+$**
  - **State (s, ν) is reachable iff. $(s_0, \nu_0) \rightarrow^* (s, \nu)$, where (s_0, ν_0) is an initial state.**
- **Reachability problem is nontrivial - transitional system S_A has an infinite number of states.**
- **Alur, Dill: Reachability of TA is decidable. - finite partitioning of infinite state space**

Introduction to parametric analysis

❖ **Example - TA with parameter** `MAX`



❖ **A parameter - a variable that is not modified by the system.**

❖ **Parametric verification - to verify a system for all possible values of the parameters**

❖ **Parametric synthesis - to find constraints on the parameters defining values that satisfies a property.**

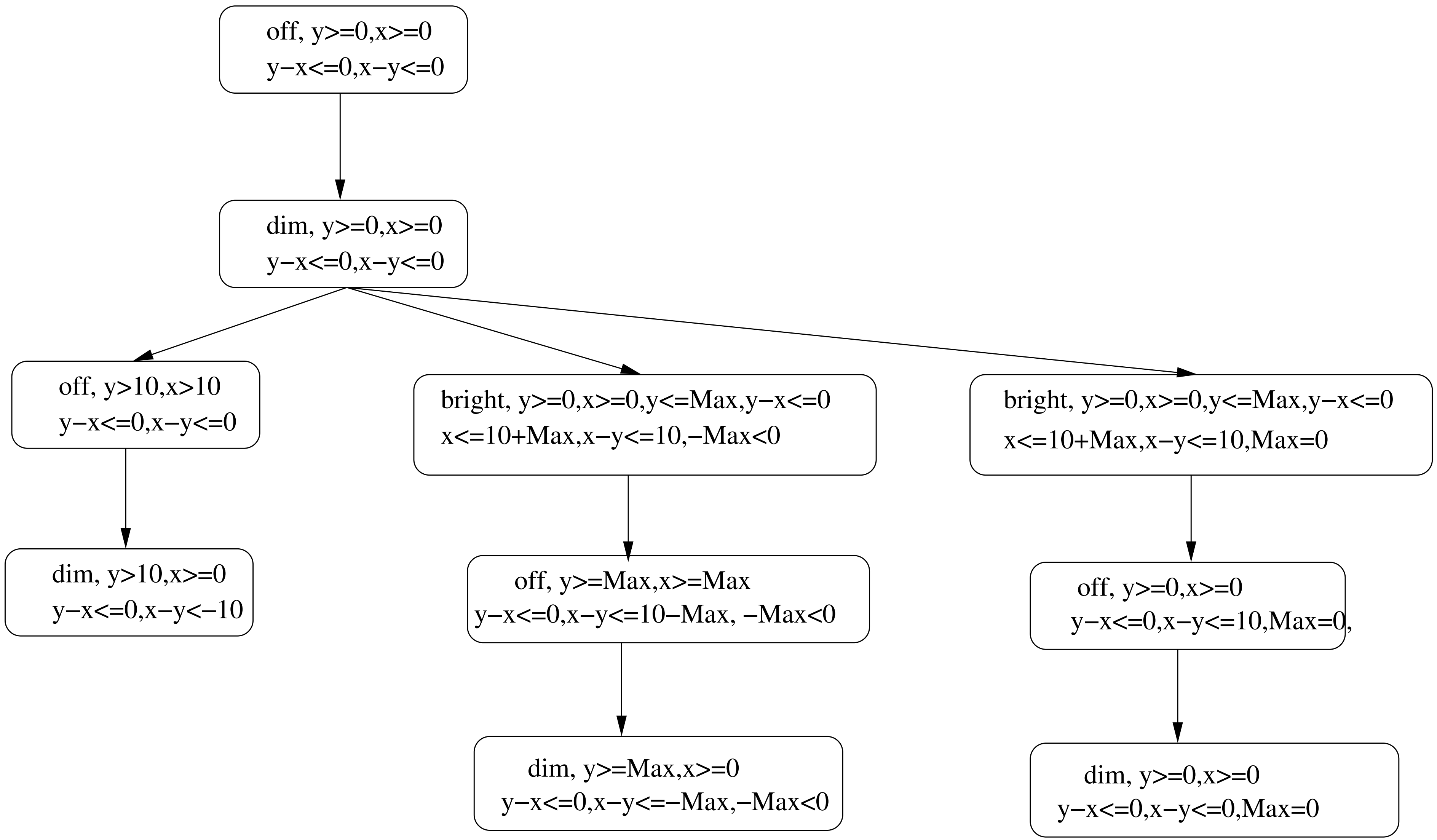
❖ **Can be solved as reachability problems in parametric models.**

Parametric analysis - Syntax

- **A Parametric Timed System (PTS) is tuple $\mathcal{T} = (Q, Q_0, X, P, I, E)$ where:**
  - Q **is a finite set of *control states*,** Q_0 **are *initial states***
  - X **is a finite set of *clocks*,**
  - P **is a finite set of *parameters*,**
  - $I : Q \rightarrow SC(X, P)$ **is a mapping *invariants* with control states,**
  - E **is a finite set of *transitions* of the form** (q_1, g, sop, q_2) **where** $q_1, q_2 \in Q$**,** $g \in SC(X, P) \in SC(X, P)$ **is a *guard* and *sop* is a simple operation over X.**
- **A simple parametric constraints SC(X,P) - conjuction of formulas**

  $\boxed{x \prec t, x - y \prec t}$ **where** $x, y \in X, \prec \in \{<, \leq\}, t \in AT(\mathcal{P})$**.**

Parametric analysis - Semantics, ex.



Issues on Parametric Analysis

❖ **Symbolic representations**

Simple Regular Expressions (SRE)

- **represent configurations of lossy fifo channels of infinite length which may lose messages at any moment**
- **SRE is a sum of products of regular expressions of the form** $(m + e)$ **and** $m_1 + \ldots + m_n)^*$**.**
- **simple operations on fifos — put, pop, top, empty**
- **widening operation — computation of postcondition for iteration of simple loops**

Issues on Parametric Analysis

❖ **Symbolic representations**

Parametric DBMs (PDBMs)

- **represent configurations of clocks and counters with parameters.**
- **a matrix on dimension** $n+1$**, each element** (i,j) **is a bound** $(\prec, t)$
- **it encodes a constraint** $x_i - x_j \prec t$ **where** $\prec \in \{<, \leq\}$**, and** t **is a term built from arithmetic operators, constants and parameters by grammar**

$$t ::= c|p|t-t|t+t|c*t$$

- **Constrained PDBM — a PDBM with constraints on the parameters** $\tilde{\mathcal{M}} = (\mathcal{M}, \Phi)$

Issues on Parametric Analysis

❖ Symbolic representations

TREX provides an implementation of Constrained PDMBs

- a data structure for PDBM and CPDBM
- simple operations - empty and universal set, inclusion, variable assignment and intersection with a constraint
- normal form computation
- widening operation corresponding to the computation of postcondition for iteration of simple loops

Issues on Parametric Analysis

❖ Symbolic representations

Syntactical Trees (STREE)

- **represent constraints over a set of parameters**
- **constraints may be non-linear**
- **represent syntactically arithmetical terms used for symbolic bounds**
- **a projection operator on real variables,**
- **a parameterized version of Fourier-Motzkin procedure**
- **heuristics to obtain an order for projection**

Issues on Parametric Analysis

❖ **Symbolic representations**

Upward Closed Sets (UC)

- **for model-checking using backward reachability of lossy fifo channels**
- **data structure to represent UC sets**
- **simple operations for empty, universal, inclusion, input and output of messages following post() or pre() semantics.**

Issues on Parametric Analysis

❖ **Manipulation with parametric structures**

Let $\tilde{b_1} = ((t_1, \prec_1), \varphi_1),\ \tilde{b_2} = ((t_2, \prec_2), \varphi_2) \in \tilde{\mathcal{PB}}$ **are two constraint parameterized bounds.**

- **Operator** $\oplus : \tilde{\mathcal{PB}} \times \tilde{\mathcal{PB}} \rightarrow \tilde{\mathcal{PB}}$ **such that**

$$\tilde{b_1} \oplus \tilde{b_2} = (t_1 + t_2, (min(\prec_1, \prec_2)), \varphi_1 \ \wedge \ \varphi_2)$$

where for all $t \in AT(\mathcal{P})$ **we define:**

$$\begin{aligned} t + \infty &= \infty \\ t + (-\infty) &= -\infty \\ \infty + \infty &= \infty \\ \infty + (-\infty) &= \infty \\ (-\infty) + (-\infty) &= -\infty \end{aligned}$$

This operator is needed for canonization operation over PDBMs.

Issues on Parametric Analysis

❖ **Manipulation with parametric structures**

- **Operator $\otimes$ - minimum between two terms**

$$\begin{aligned}
\Phi_< &\equiv \exists p \in \mathcal{P}.\varphi_1 \wedge \varphi_2 \wedge t_1 < t_2 \\
\Phi_= &\equiv \exists p \in \mathcal{P}.\varphi_1 \wedge \varphi_2 \wedge t_1 = t_2 \\
\Phi_> &\equiv \exists p \in \mathcal{P}.\varphi_1 \wedge \varphi_2 \wedge t_1 > t_2
\end{aligned}$$

Operator $\otimes : \tilde{\mathcal{PB}} \times \tilde{\mathcal{PB}} \longrightarrow 2^{\tilde{\mathcal{PB}}}$ is such that $\tilde{b_1} \otimes \tilde{b_2} = min(\tilde{b_1}, \tilde{b_2})$.

$$\begin{aligned}
min(\tilde{b_1}, \tilde{b_2}) &= min_{\leq}(\tilde{b_1}, \tilde{b_2}, \Phi_<) \\
&\cup \; min_{=}(\tilde{b_1}, \tilde{b_2}, \Phi_=) \\
&\cup \; min_{>}(\tilde{b_1}, \tilde{b_2}, \Phi_>)
\end{aligned}$$

Issues on Parametric Analysis

- **Manipulation with parametric structures**

where

$$\textit{min}_{<}(\tilde{b}_1, \tilde{b}_2, \Phi_<) = \begin{cases} \{((t_1, \prec_1), \varphi_1 \wedge \varphi_2 \wedge (t_1 < t_2))\} & \textbf{if } \Phi_< \\ \emptyset & \textbf{otherwise} \end{cases}$$

$$\textit{min}_{=}(\tilde{b}_1, \tilde{b}_2, \Phi_=) = \begin{cases} \{(t_1, \prec_1), \varphi_1 \wedge \varphi_2 \wedge (t_1 = t_2))\} & \textbf{if } \Phi_= \wedge \prec_1 \leq \prec_2 \\ \{(t_2, \prec_2), \varphi_1 \wedge \varphi_2 \wedge (t_1 = t_2))\} & \textbf{if } \Phi_= \wedge \prec_2 < \prec_1 \\ \emptyset & \textbf{otherwise} \end{cases}$$

$$\textit{min}_{>}(\tilde{b}_1, \tilde{b}_2, \Phi_>) = \begin{cases} \{((t_2, \prec_2), \varphi_1 \wedge \varphi_2 \wedge (t_1 > t_2))\} & \textbf{if } \Phi_> \\ \emptyset & \textbf{otherwise} \end{cases}$$

The result of min operation — a set of one, two or three constrained parameterized bounds.

Issues on Parametric Analysis

❖ Acceleration

Let (q, ph) be a symbolic configuration and let θ be a control loop.

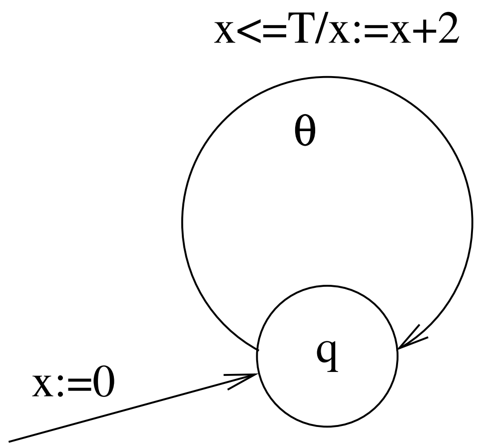
- **difference between (q, ph) and $post_\theta(q, ph)$ is Δ**
- **difference between $post_\theta^2(q, ph)$ and $post_\theta(q, ph)$ is Δ too**
- **we suspect the effect of iterating loop θ —adding an increment Δ to the original set**

❖ Condition C2 must hold

$C2 : \forall n \geq 0 \ . \ post_\theta^2((q, ph + n * \Delta)) = post_\theta((q, ph + (n+1) * \Delta))$.

Issues on Parametric Analysis

❖ **Acceleration - an example**



- **an initial configuration is** $c_1 = (q_0, 0, T > 0)$
- **after applying control loop** θ **a new configuration is** c_2**:**

$$\begin{aligned} c_2 &= post_\theta(c_1) = (q_0, (0 \leq x \leq 0) \cap (x \leq T), 0 < T)|_{x:=x+2} \\ &= (q_0, 0 \leq x \leq 0, 0 < T)|_{x:=x+2} = (q_0, 2 \leq x \leq 2, 0 < T) \end{aligned}$$

Issues on Parametric Analysis

❖ Acceleration - an example

- **difference of c_2 and c_1 is an interval $\Delta_1 = \langle (2, \leq), (2, \leq) \rangle$**
- **another iteration of $post()$ is**

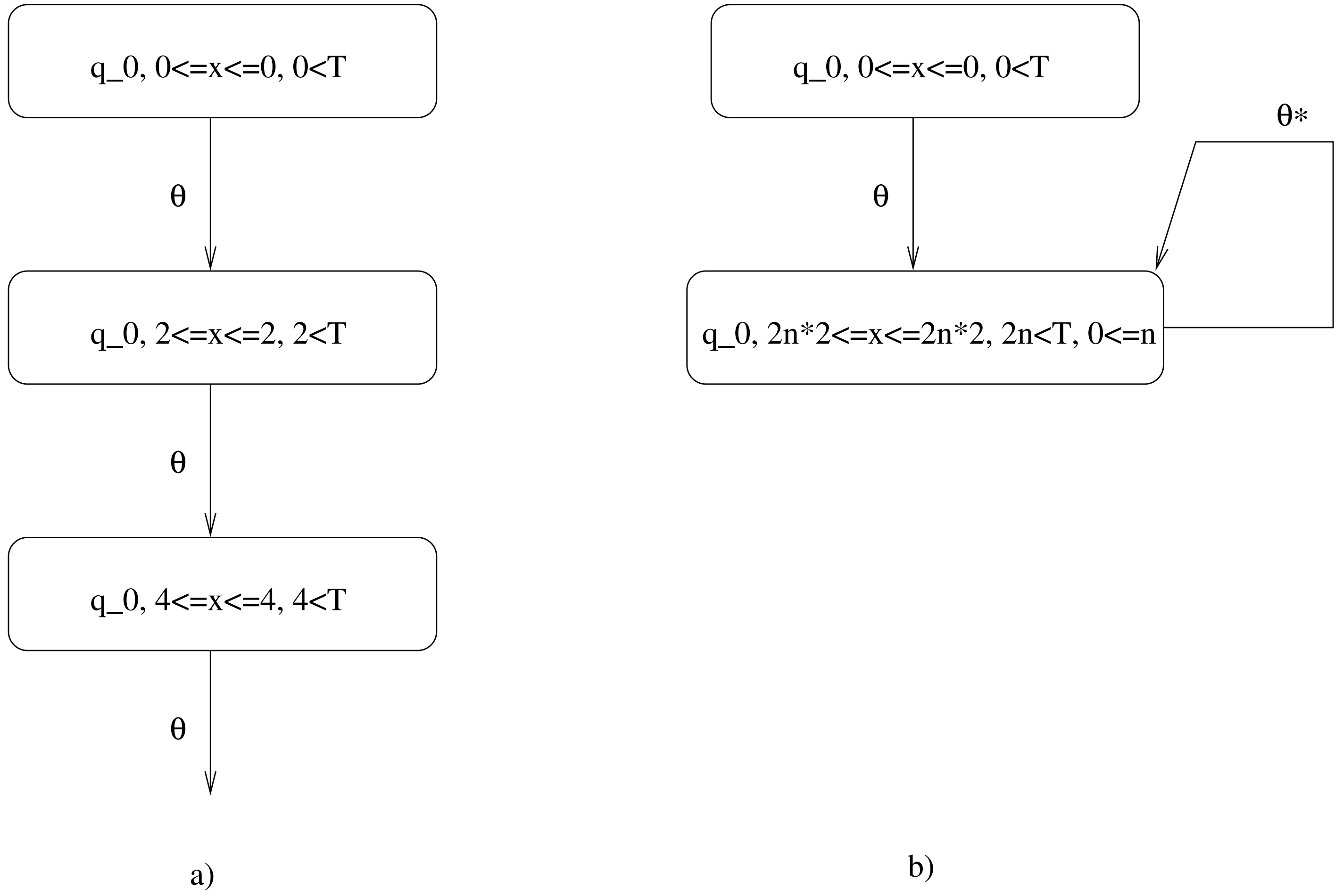
$$
\begin{aligned}
c_3 &= post_\theta(c_2) \\
&= (q_0, (2 \leq x \leq 2) \cap (x \leq T), 0 < T)|_{x:=x+2} \\
&= (q_0, 2 \leq x \leq 2, 2 < T)|_{x:=x+2} \\
&= (q_0, 4 \leq x \leq 4, 2 < T)
\end{aligned}
$$

- **difference between c_3 and c_2 is $\Delta_2 = \langle (2, \leq), (2, \leq) \rangle$**
- $\Delta_1 = \Delta_2$

Issues on Parametric Analysis

❖ Acceleration - an example

- **Effect of acceleration—before acceleration (a) and after (b)**



Running TREX

❖ Starting TREX

```
trex -c <init_constraints.cnd> -init <model.if> -sg <symbolic_grap.aut> -tr
<traces.tr> -res <reachable_set.scf> -cp <num> -e
```

Options:

-a	**filename.acc - acceleration applied on given states**
-c	**filename.cnd - initial constraints on parameters**
-cp	**a number for the deep of searching iterating states in order to apply accelerations**
-e	**use the FOAF package instead of REDUCE for elimination of quantifiers over reals**
-init	**filename.if - specification of the input model**
-pre	**backward analysis**
-prop	**filename.if - use the model described in the file like an observer specifying a safety property to be verified**
-res	**filename.scf - writes the reachable configurations in filename.scf**
-sg	**filename.aut - produces the finite graph of symbolic reachable configurations using the Aldebaran format**
-tr	**filename.tr - prints the analysis trace in file filename.tr**

Running TREX

❖ Results of the analysis

Reachable configurations.

- **TREX generates a set of reachable configurations**
- **a finite symbolic graph**

A symbolic graph generated by TREX is given by several files:

- **a file of transitions between reachable symbolic configurations (in ALDEBARAN format, ie. one of the graph formats of CADP) - can be used by CADP for finite model-checking and minimization**
- **a file listing the reachable symbolic configurations - can be used to extract new initial constraints to do abstraction with INVEST.**

Running TREX

❖ Results of the analysis

On-the-fly check of safety properties.

- **the property is given as an observer**
- **if the property is not satisfied,** TREX **generates a sequence of transitions from the initial state of the model to the state with bad behavior**
- **a symbolic configuration of the bad state can be used to synthesize constraints under which the safety property is satisfied.**

Check of some kind of liveness property.
TREX **can synthetise fairness constraints stating about the bounded iterability of some kind of loops.**

References

[TREX] TREX at http://www.liafa.jussieu.fr/~sighirea/trex/index.html

[AAB99] P.A. Abdulla, A. Annichini, and A. Bouajjani. Symbolic verification of lossy channel systems: Application to the bounded retransmission protocol. In *Proceedings of 5th TACAS*, volume 1579 of *LNCS*. Springer, 1999.

[AAB00] A. Annichini, E. Asarin, and A. Bouajjani. Symbolic techniques for parametric reasoning about counter and clock systems. In E.A. Emerson and A.P. Sistla, editors, *Proceedings of the 12th CAV*, volume 1855 of *LNCS*, pages 419–434. Springer Verlag, July 2000.

[ABJ98] P.A. Abdulla, A. Bouajjani, and B. Jonsson. On-the-fly analysis of systems with unbounded, lossy, FIFO channels. In *Proceedings of the 10th CAV*, volume 1427, pages 305–317, 1998.

[BCAS01] A. Bouajjani, A. Collomb-Annichini, and M. Sighireanu. Trex: A tool for reachability analysis of complex systems. In *Proceedings of CAV*, volume 2102 of *LNCS*, pages 368–372. Springer Verlag, June 2001.

[Col01] A. Annichini Collomb. *Vérification d'automates étendus: algorithmes d'analyse symbolique et mise en oeuvre.* PhD thesis, Joseph Fourier University, December 2001.

[Hea99] A.C. Hearn. *REDUCE — User's and Contributed Packages Manual*. Codemist Ltd., February 1999. version 3.7.

[Ome96] Omega Team. *The Omega Library*, November 1996. version 1.1.0.