On Interleaving in Timed Automata

Oded Maler, Marius Bozga, Ramzi Ben Salah

VERIMAG

24th August 2006

Introduction

- Exploring the state space of Timed Automata is important (circuits timing analysis, scheduling, etc). However, it's a very difficult problem limited by the *state-explosion problem*.
- Part of the explosion is coming from the effect of interleaving on splitting of zones. We show how to get rid of this explosion.
- We prove a simple convexity result and use it to modify slightly the "classical" reachability algorithm for TA and avoid this explosion.

Plan

Quick Review On Timed Automata

State Explosion Due to Interleaving Semantics

Convexity Result

Application to reachability computation

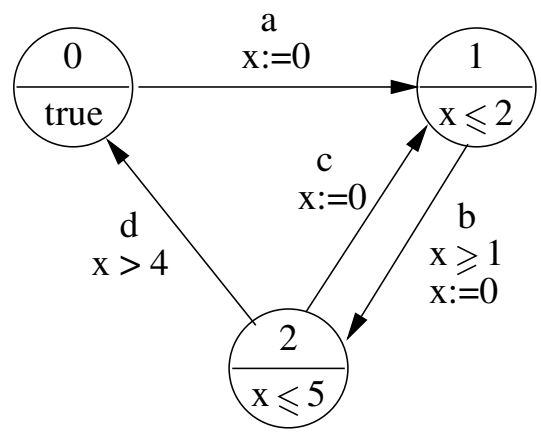
Conclusion

Timed Automata

A *Timed Automaton* is $\mathcal{A} = (\Sigma, Q, C, I, \Delta)$ where:

- Σ is a finite set of transition labels.
- Q is a finite set of states.
- C is a finite set of clocks.
- I is the invariant (staying condition), assigning to every $q \in Q$ a conjunction of time constraints I_q.
- Δ is the transition relations of the form (q, g, a, r, q') where:
  - $q, q' \in Q$ are the source and target states of the transition.
  - $a \in \Sigma$ is the transition label.
  - g is the transition guard (a conjunction of time constraints).
  - $r \subseteq C$ is a set of clocks to be reset by the transition.

Example:



Runs of Timed Automata

A configuration is a pair (q, v) consisting of a *discrete state* q and a *clocks valuation* $v{:}C \to \mathbb{R}_+ \cup \{0\}$.

A step of the automaton is one of the following:

- A time step: $(q, v) \xrightarrow{d} (q, v + d)$, $d \in \mathbb{R}_{\geq 0}$ such that $v + d$ satisfies I_q.
- A discrete step: $(q, v) \xrightarrow{a} (q', v')$ for some transition $(q, g, a, r, q') \in \Delta$ such that v satisfies g and $v' = r(v)$.

A compound step is a time step followed by a discrete step:

$$(q, v) \xrightarrow{d,a} (q', v') \equiv (q, v) \xrightarrow{d} (q, v + d) \xrightarrow{a} (q', v')$$

A run of the automaton starting from the configuration (q_0, v_0) is a finite sequence of compound steps ending in a time step:

$$\xi : (q_0, v_0) \xrightarrow{d_1,a_1} (q_1, v_1) \xrightarrow{d_2,a_2} \ldots \xrightarrow{d_k,a_k} (q_k, v_k) \xrightarrow{d_*} (q_k, v_k + d_*)$$

Composition of Timed Automata

A *composition of timed automata* is $\mathcal{A} = \mathcal{A}^1 \parallel \mathcal{A}^2 \parallel ... \parallel \mathcal{A}^n$ where each automaton is of the form $\mathcal{A}^i = (\Sigma^i, Q^i, C^i, I^i, \Delta^i)$. The action alphabets can overlap, but the set of clocks of the automata are *mutually disjoint*.

The Global Automaton obtained from the composition is $\mathcal{A} = (\Sigma, Q, C, I, \Delta)$ where $\Sigma = \bigcup_{i=1}^{n} \Sigma^i$, $Q = \prod_{i=1}^{n} Q^i$ and $C = \bigcup_{i=1}^{n} C^i$. We note a global state as $q = (q^1, q^2, ..., q^n)$ and a global clock valuation over C as $v = (v^1, v^2, ..., v^n)$.

The semantics of the composition is given in term of global steps as follows:

- Time step: $(q, v) \xrightarrow{d} (q, v + d)$, $d \in \mathbb{R}_{\geq 0}$ such that $v + d$ satisfies $\bigwedge_{i=1}^{n} I_{q^i}$.

- Discrete step: $(q, v) \xrightarrow{a} (q', v') \begin{cases} \text{if } a \in \Sigma^i,\ (q^i, v^i) \xrightarrow{a} (q'^i, v'^i)\ (\text{local step of } \mathcal{A}^i) \\ \text{if } a \notin \Sigma^i,\ (q^i, v^i) = (q'^i, v'^i) \end{cases}$

- Global compound steps and global runs are defined similarly to their local counterparts.

The Symbolic Representation

- The semantics of a timed automaton yields an infinite transition system which is not an appropriate basis for verification algorithms
  $\Rightarrow$ Symbolic representation.

- The standard reachability algorithm (Kronos and Uppaal,...) computes a reachability graph $S = (N, \rightarrow)$, the nodes of which are symbolic states.

- A symbolic state is of the form (q, Z), where q is a discrete state and Z is a zone, a convex set of clocks valuations satisfying clock constraints.

- **NB**: There is a path of S from (q, Z) to (q', Z') iff for every $v' \in Z'$ there exists $v \in Z$ and a run of $\mathcal{A}$ from (q, v) to (q', v').

The Standard Reachability Computation

Standard algorithm: Starting by the initial symbolic state $(q_0, true)$ $Succ^{\delta}$ is applied until termination

$$Succ^{\delta}(q,Z) = Post^{t}\left(Post^{\delta}(q,Z)\right)$$

- The δ-transition successor of (q,Z) is the set of configurations reachable from (q,Z) by taking the transition $\delta = (q,g,a,r,q') \in \Delta$:

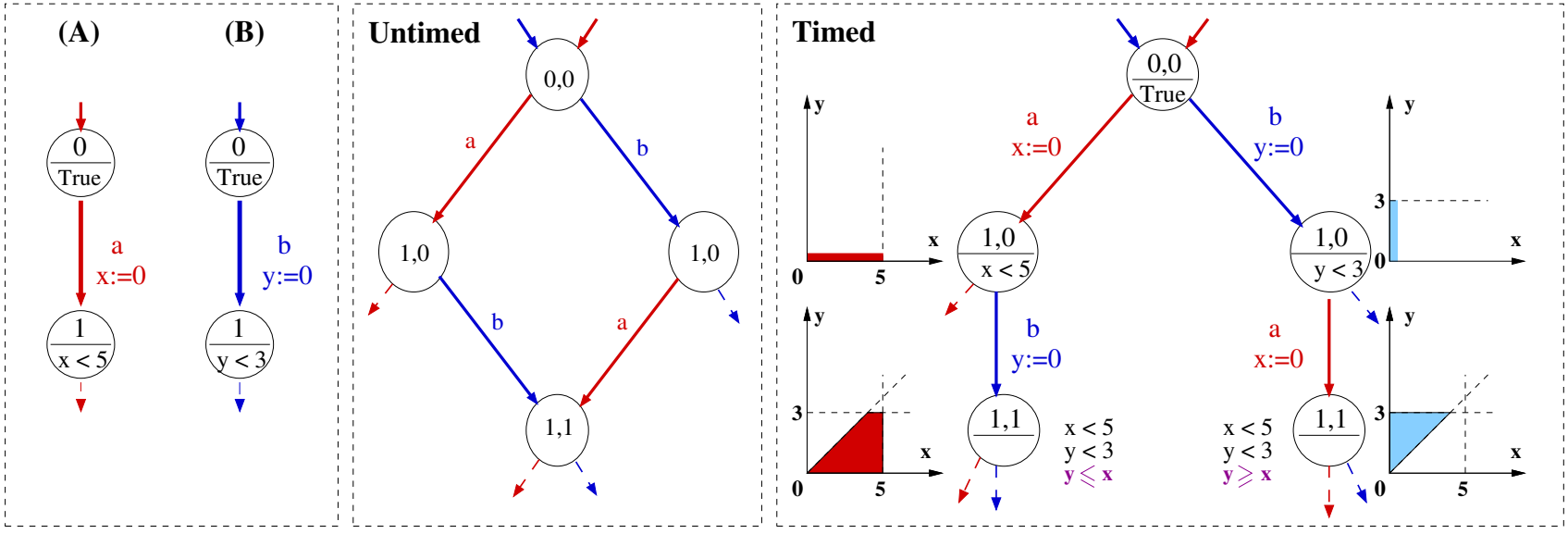
$$Post^{\delta}(q,Z) = \left\{(q', r(z)) : z \in Z \cap g\right\}$$

- The time successor of (q,Z) is the set of configurations reachable from (q,Z) by letting the time progress without violating the staying condition:

$$Post^{t}(q,Z) = \{(q, z+d) : z \in Z, d \geq 0, \text{and } z+d \in I_q\}$$

Plan

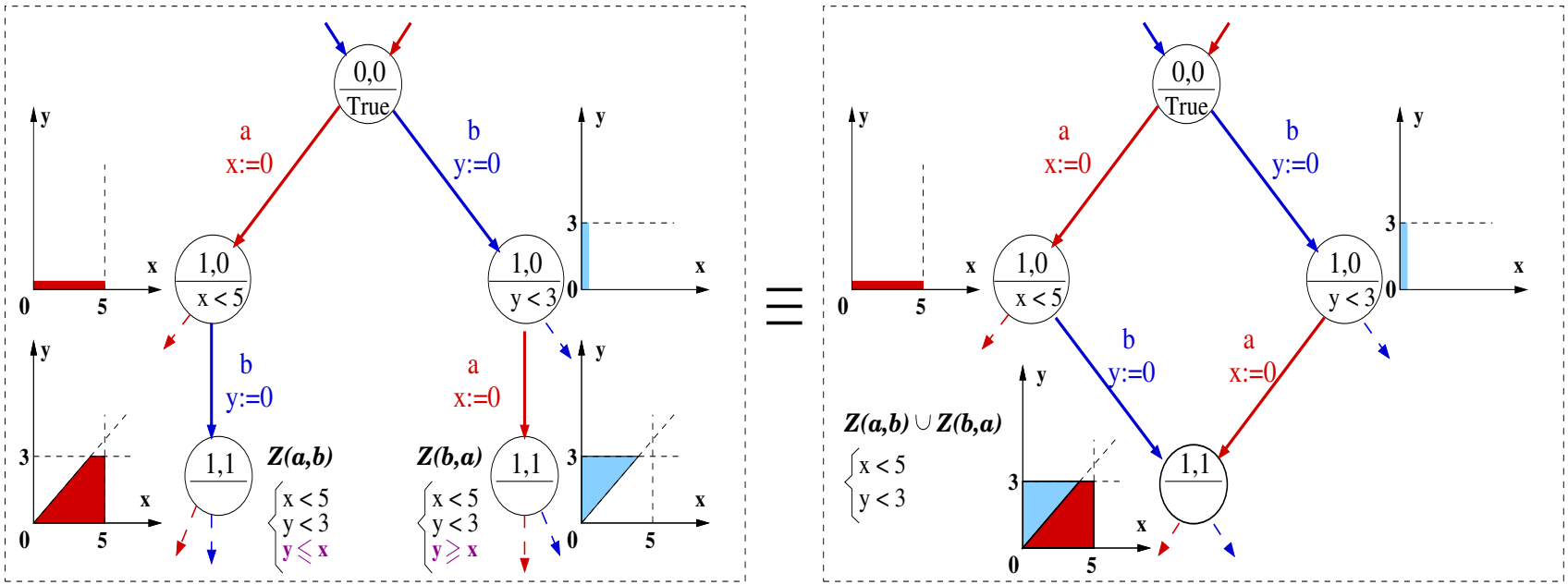
Example: Interleaving in TA Splits Zones



Untimed reachability will converge to single state, where Timed reachability using the standard algorithm will generate several symbolic states - two in the example:

- One with the zone $Z(a,b)$ in which $y \leq x$ because in all runs along the first path x is reset before y.
- One with the zone $Z(b,a)$ in which $y \geq x$ because in all runs along the second path x is reset after y.

Example: Interleaving in TA Splits Zones



- Notice that $Z(a,b) \cup Z(b,a)$ is a convex set.
- Convexity $\Rightarrow$ Exact reduction through states merging.
- General criterion for convexity : The union of all zones reached by different locally-equivalent runs is convex.

Local Runs of the Global Automaton

- A local run ξ^i is the projection of a global run ξ of the global automaton $\mathcal{A} = \mathcal{A}^1 \parallel \mathcal{A}^2 \parallel \ldots \parallel \mathcal{A}^n$ on the automaton $\mathcal{A}^i$.
- The projection ξ^i of ξ is obtained by "hiding" the transitions in which $\mathcal{A}^i$ does not participate, projecting the run on the states and clocks of $\mathcal{A}^i$, and collapse the time passage.

Example

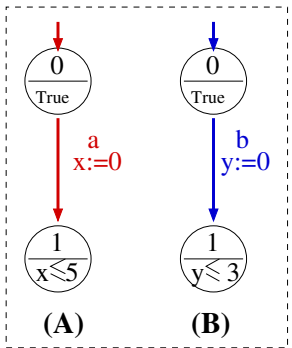
A possible global run ξ: $(\,(q,v)=((q_A,q_B),(x,y))\,)$

$$((0,0),(0,0)) \xrightarrow{6} (0,0),(6,6)) \xrightarrow{a} ((1,0),(0,6)) \xrightarrow{3} ((1,0),(3,9)) \xrightarrow{b} ((1,1),(3,0)) \xrightarrow{1.3} ((1,1),(4.3,1.3))$$

The projection of ξ on B: $(\,(q,v)=(q_B,y)\,)$

$$[(0,0) \xrightarrow{6} (0,6) \xrightarrow{\varepsilon} (0,6) \xrightarrow{3} (0,9)] \xrightarrow{b} (1,0) \xrightarrow{1.3} (1,1.3)$$ After projection

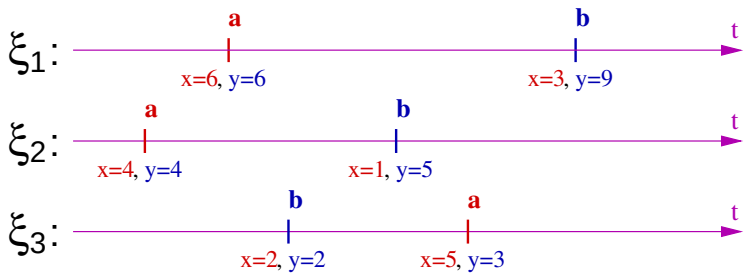
$$[(0,0) \xrightarrow{9} (0,9)] \xrightarrow{b} (1,0) \xrightarrow{1.3} (1,1.3)$$ After the time merging



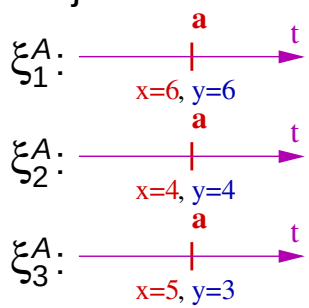
Qualitative & local equivalence between runs

- Two runs ξ and ξ' are qualitatively equivalent ($\xi \approx \xi'$) if they go through the same sequence of discrete transitions and differ only in timing. The class of runs qualitatively equivalent to ξ is denoted $[\xi]$.
- Two runs ξ and ξ' are locally equivalent ($\xi \sim \xi'$) if all their local projections are qualitatively equivalent: $\bigwedge_{1 \leq i \leq n} (\xi^i \approx \xi'^i)$. The class of runs locally equivalent to ξ will be denoted $\langle \xi \rangle$.

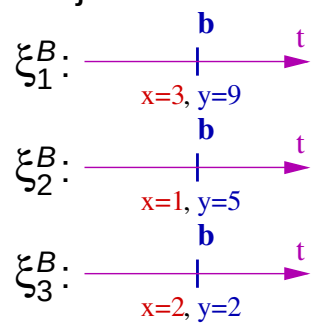
Example of valid global runs :

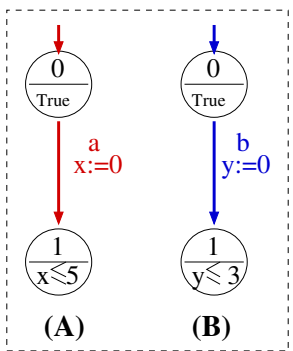


Projection on A



Projection on B





$$\left(\xi_1^A \approx \xi_2^A \approx \xi_3^A\right) \wedge \left(\xi_1^B \approx \xi_2^B \approx \xi_3^B\right) \Rightarrow \xi_1 \sim \xi_2 \sim \xi_3$$

Notice: $\xi_1 \approx \xi_2 \Rightarrow \xi_1 \sim \xi_2$

Plan

Convexity Result

Theorem *Let Z be a convex timed polyhedron and let q and q' be two global states of $\mathcal{A}$. Let ξ be a run starting at q and ending at q'. Then the set*

$$R_{Z,\langle\xi\rangle} \equiv \bigcup_{\xi' \in \langle\xi\rangle} \left\{ v' : \exists v \in Z, (q,v) \xrightarrow{\xi'} (q',v') \right\} \text{ is convex}$$

Proof

We proved that the condition for a valid global run starting at Z_0 and locally equivalent to a given run ξ is expressed as a conjunctive formula:

$$\Phi(t,v) = \begin{pmatrix} t_0^1 = t_0^2 = \ldots = t_0^n & \wedge \\ v_0 \in Z_0 & \wedge \\ \bigwedge_{i=1}^{n} \Phi^i(v^i, t^i) & \wedge \\ \bigwedge_{a \in \Sigma} \Psi_a(t) & \wedge \\ t_{k+1}^1 = t_{k+1}^2 = \ldots = t_{k+1}^n & \end{pmatrix}$$

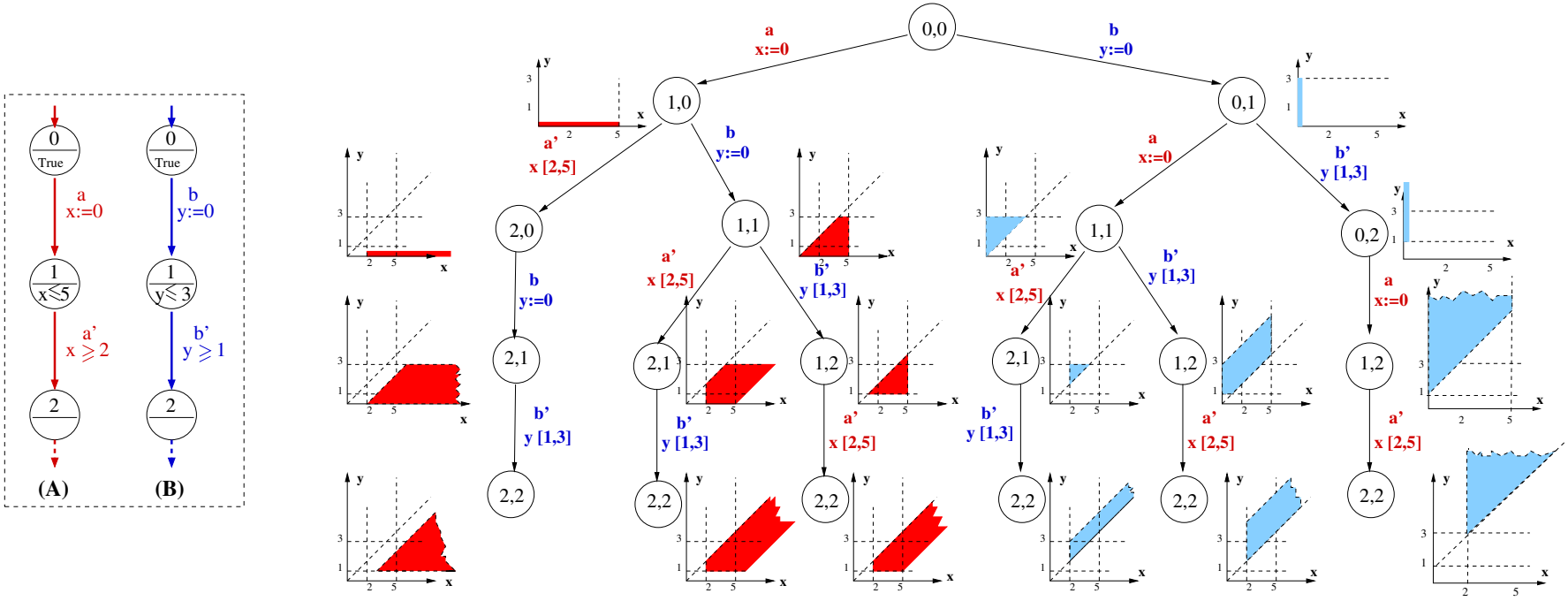
where:

$$\begin{cases} \Phi^i(t^i, v^i) = \bigwedge_{j=1}^{k} \begin{pmatrix} \exists d, d = t_j^i - t_{j-1}^i & \wedge \\ l_{j-1}^i(v_{j-1}^i + d) & \wedge \\ g_j^i(v_{j-1}^i + d) & \wedge \\ v_j^i = r_j^i(v_{j-1}^i + d) & \end{pmatrix} & \textit{and} \\ \\ \Psi_a(t) = \bigwedge_{(i,j),(i',j') \in \left\{(i,j):a_j^i = a\right\}} t_j^i = t_{j'}^{i'} & \end{cases}$$

This set is a convex subset in the space consisting of all valuations and time stamps.

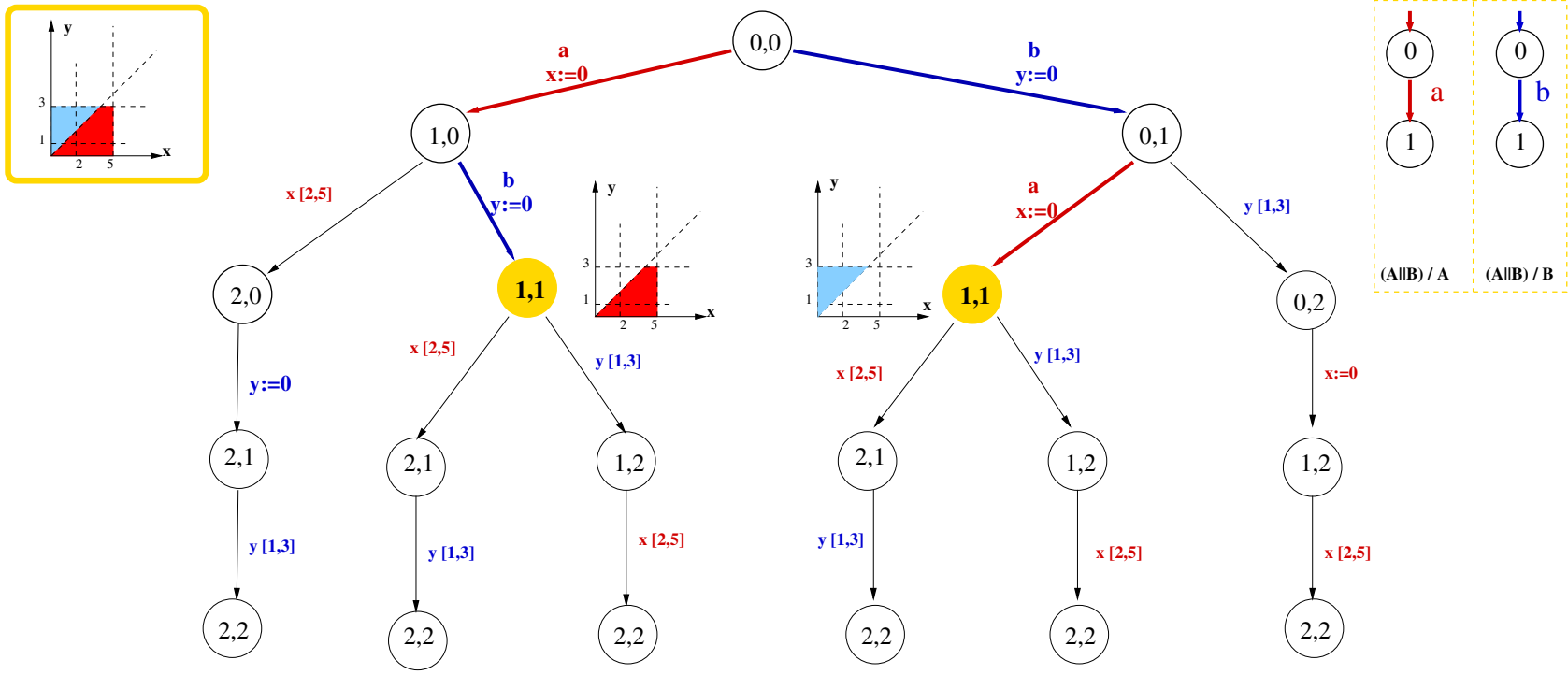
$R_{Z,\langle\xi\rangle}$ could be defined as the projection of this convex set $\Rightarrow$ $R_{Z,\langle\xi\rangle}$ is convex.

Example



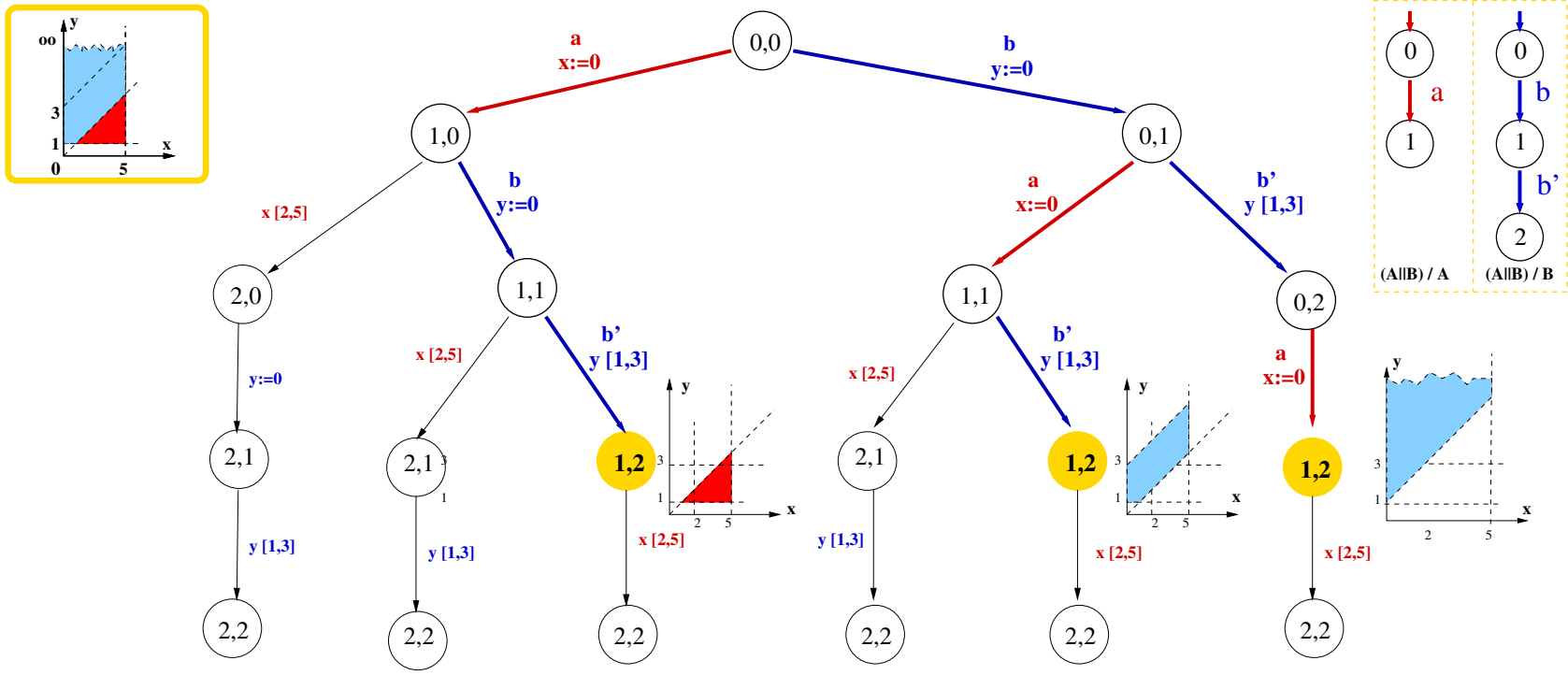
The graph generated by the standard reachability algorithm: 19 symbolic states.

Example



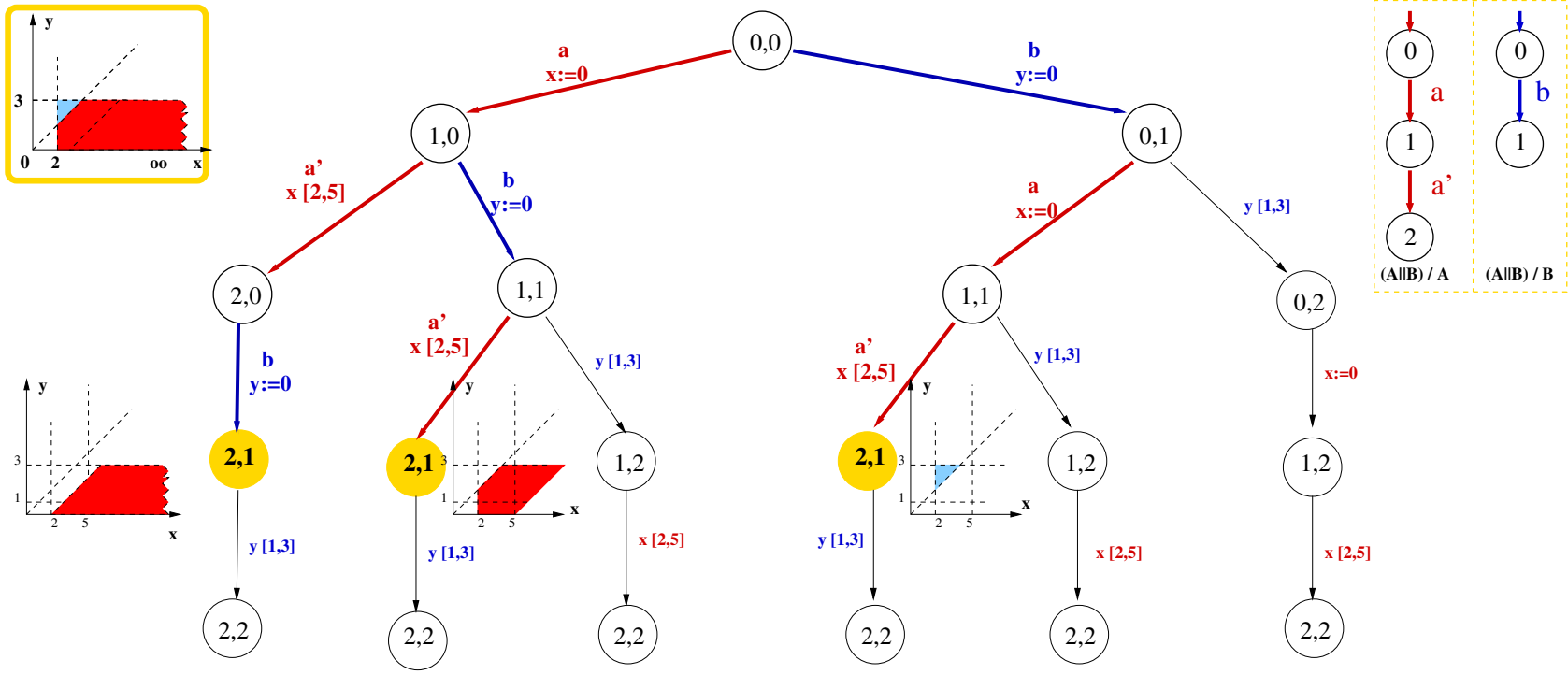
The union of all zones reached by different locally-equivalent runs is convex.

Example



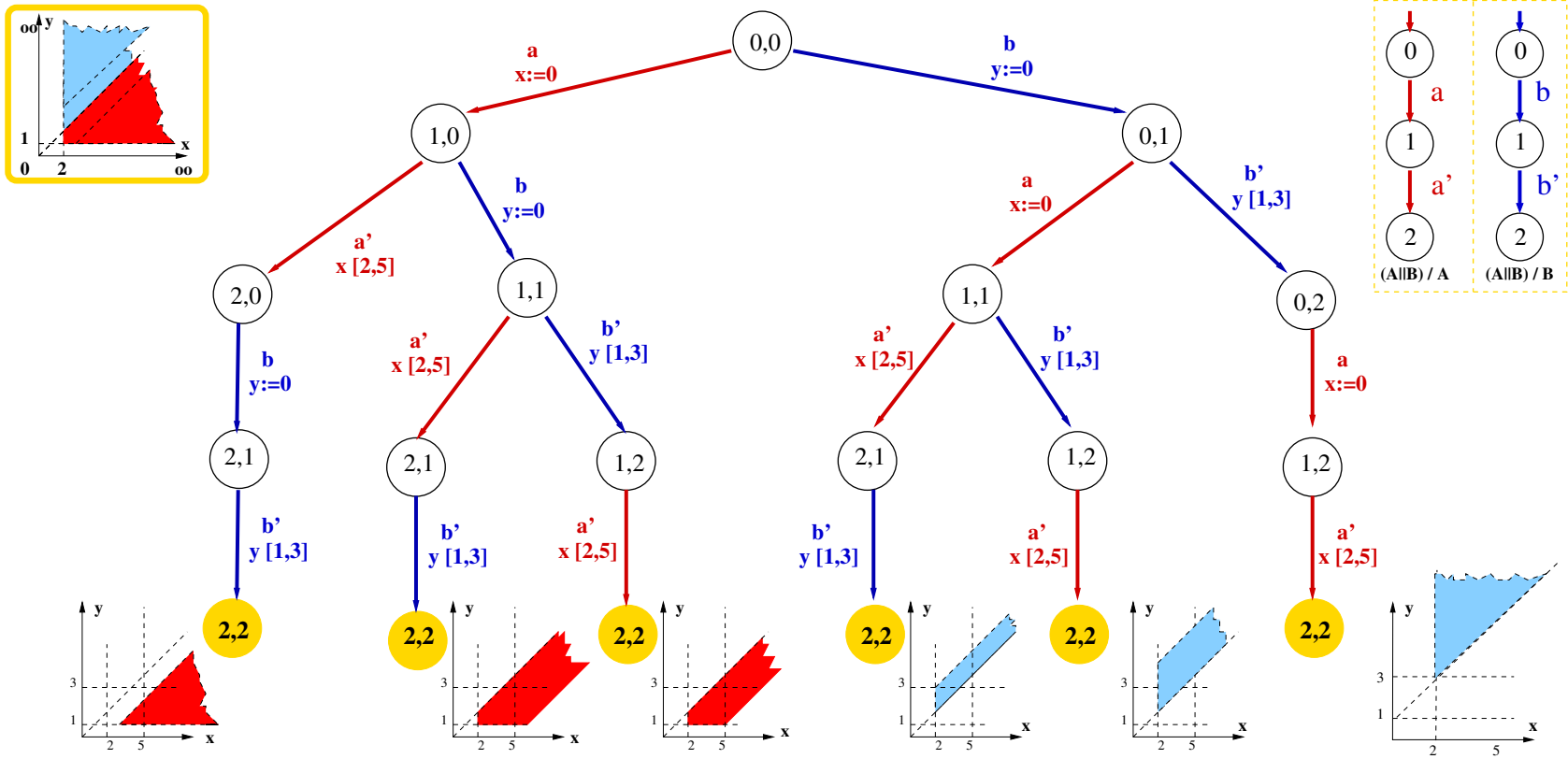
The union of all zones reached by different locally-equivalent runs is convex.

Example



The union of all zones reached by different locally-equivalent runs is convex.

Example



The union of all zones reached by different locally-equivalent runs is convex.

Plan

The Improved Reachability Computation Algorithm

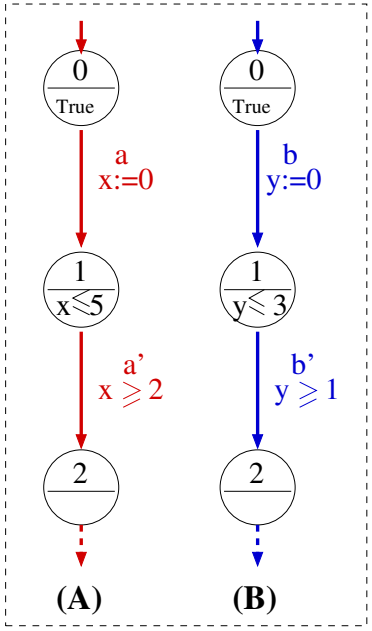
- We generate the graph in a breadth-first manner.
- At each level we merge some symbolic states.
- To recognize states reached by locally equivalent runs we decorate the symbolic states with path information:
  - A *shuffle expression* over Σ is $\alpha = (\alpha^1 \parallel .. \parallel \alpha^n)$ with $\alpha^i \in (\Sigma)^*$.
  - A concatenation of a shuffle expression and a symbol a is defined as $(\alpha^1 \parallel .. \parallel \alpha^n).a = (\beta^1 \parallel .. \parallel \beta^n)$ where $\begin{cases} \beta^i = \alpha^i \text{ if } a \notin \Sigma^i \\ \beta^i = \alpha^i.a \text{ if } a \in \Sigma^i \end{cases}$
- Merging : $\{(q,Z_1,\alpha),\ldots,(q,Z_m,\alpha)\} \Rightarrow (q,\bigcup_i Z_i,\alpha)$.

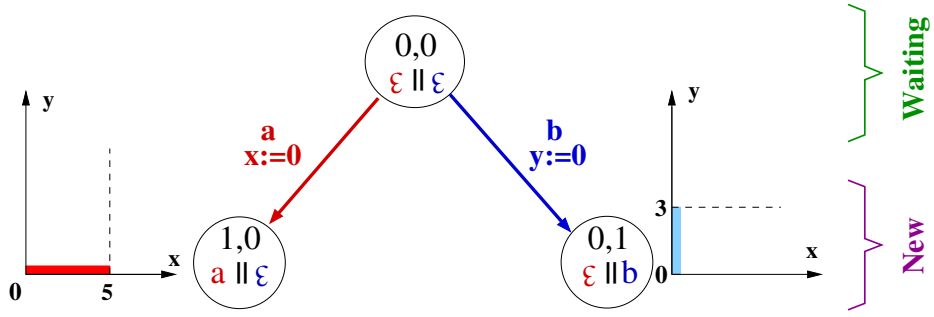
The Improved Reachability Computation Algorithm

Algorithm

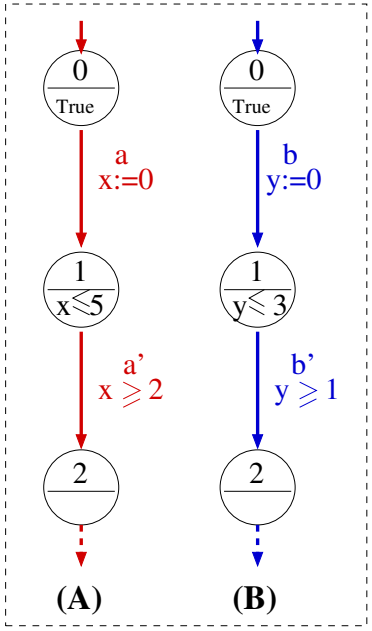
Explored $:= New := \emptyset$
Waiting $:= \{(q_0, Z_0, \varepsilon \parallel .. \parallel \varepsilon)\}$
while *Waiting* $\neq \emptyset$ do
  for each $(q, Z, \alpha) \in$ *Waiting* such that $(q, Z) \notin$ *Explored* **do**
    for each $a \in \Sigma$ **do**
      $New := New \cup \{(Succ^a(q, Z), \alpha.a)\}$
    $Explored := Explored \cup \{(q, Z)\}$
  $Waiting :=$ *Merge*(New)
return(*Explored*)

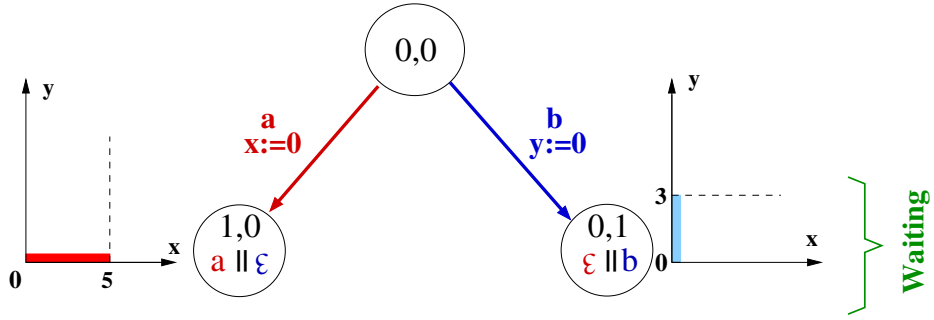
Example


0
True
a
x:=0
1
x≤5
a'
x ≥ 2
2
(A)
0
True
b
y:=0
1
y≤ 3
b'
y ≥ 1
2
(B)

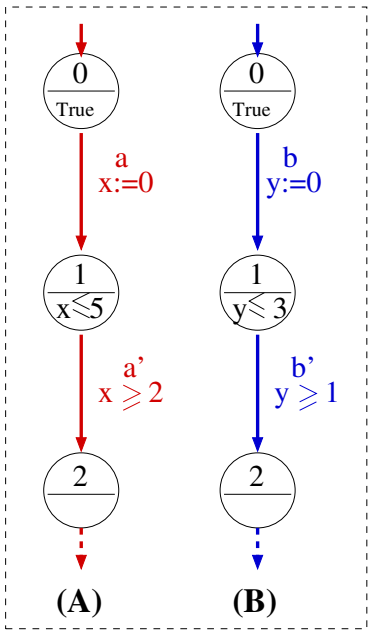

0,0
ε ‖ ε
a
x:=0
b
y:=0
1,0
a ‖ ε
0,1
ε ‖ b
y
x
0
5
3
Waiting
New

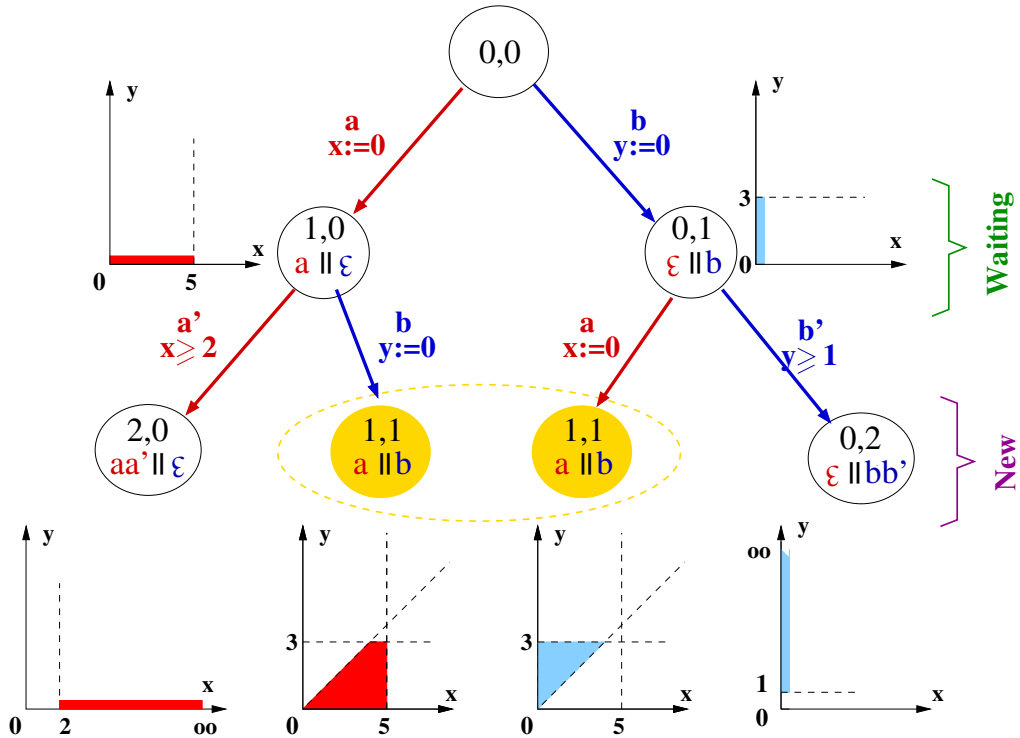
Example


0
True
a
x:=0
1
x≤5
a'
x ≥ 2
2
(A)
0
True
b
y:=0
1
y≤ 3
b'
y ≥ 1
2
(B)

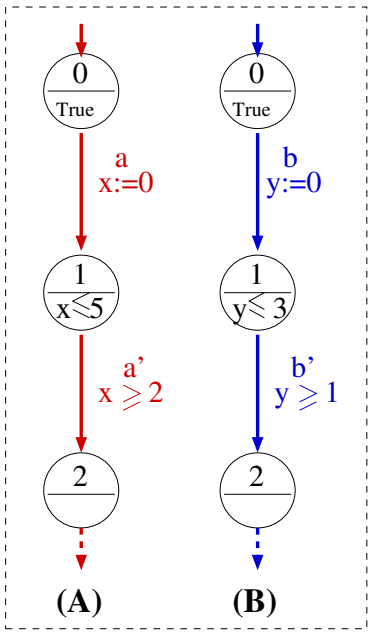

0,0
a
x:=0
b
y:=0
y
x
0
5
1,0
a ∥ ε
0,1
ε ∥ b
y
3
x
0
Waiting

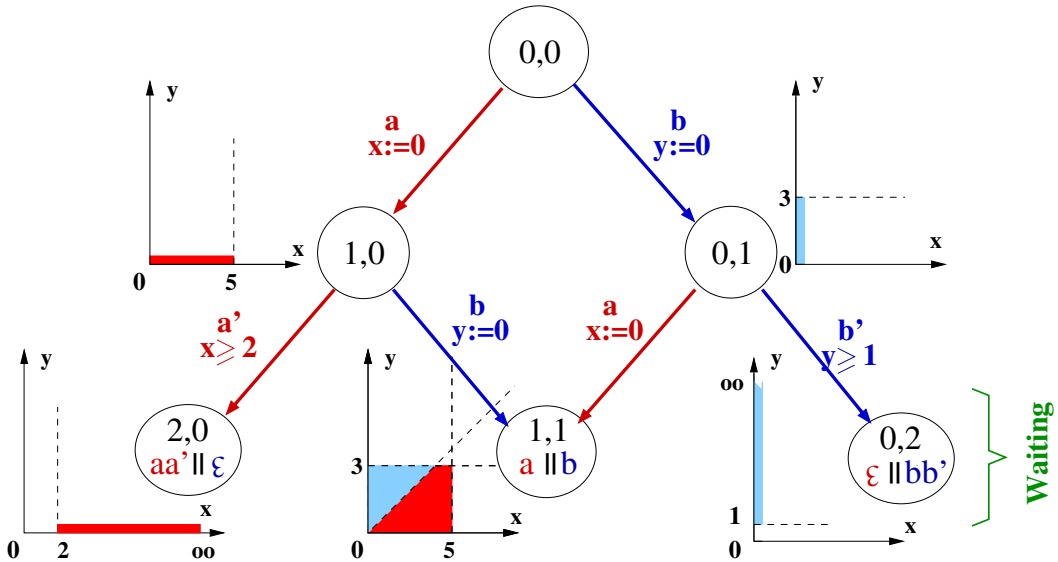
Example


0
True
a
x:=0
1
x≤5
a'
x ≥ 2
2
(A)
0
True
b
y:=0
1
y≤ 3
b'
y ≥ 1
2
(B)

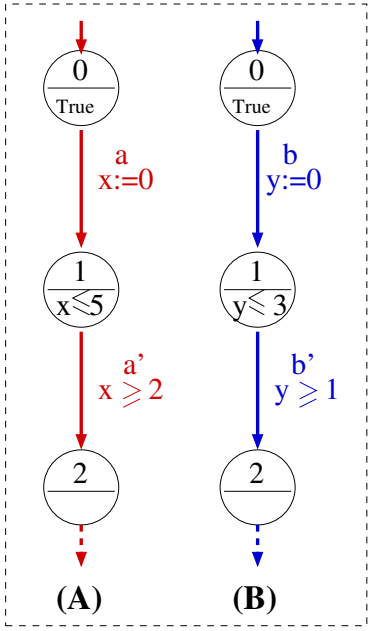

0,0
a
x:=0
b
y:=0
1,0
a ‖ ε
0,1
ε ‖b
a'
x≥ 2
b
y:=0
a
x:=0
b'
y≥ 1
2,0
aa'‖ ε
1,1
a ‖b
1,1
a ‖b
0,2
ε ‖bb'
Waiting
New
y
x
0
5
3
2
oo
1

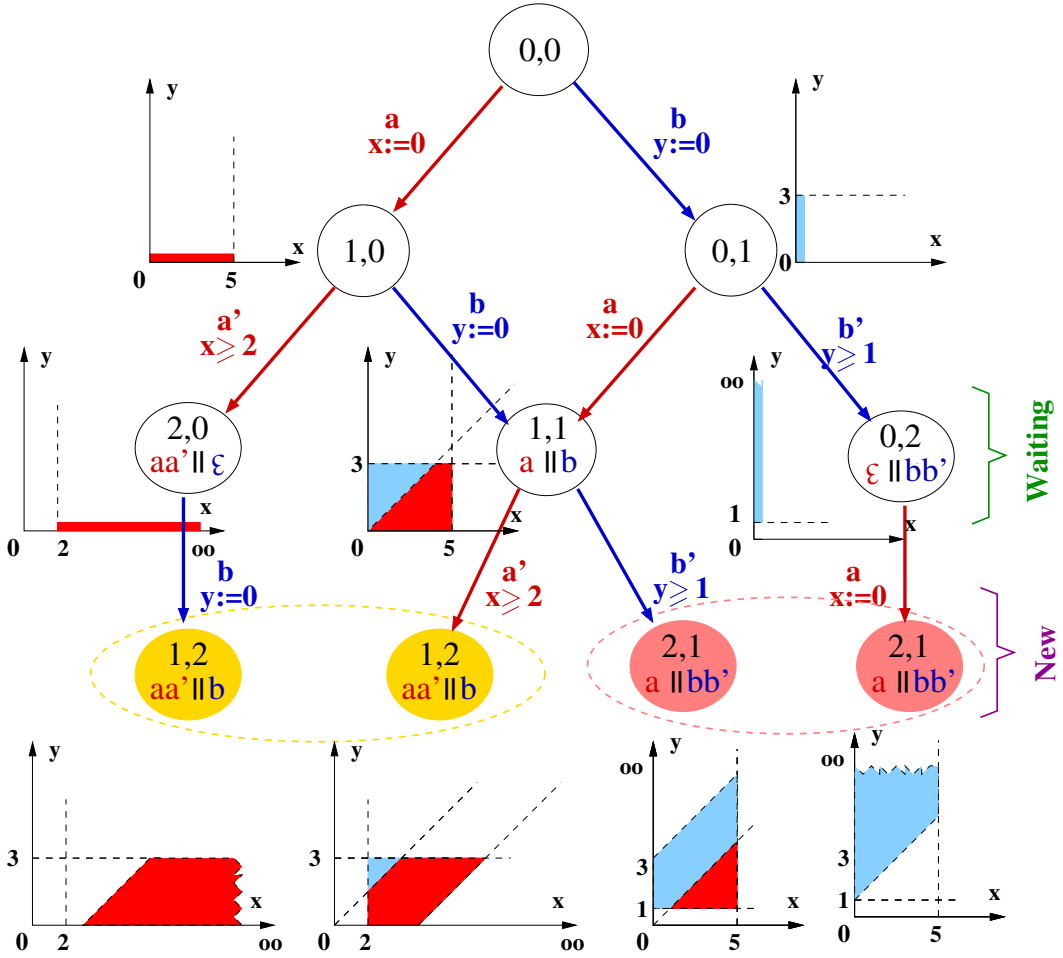
Example


0
True
a
x:=0
1
x≤5
a'
x ≥ 2
2
(A)
0
True
b
y:=0
1
y≤3
b'
y ≥ 1
2
(B)

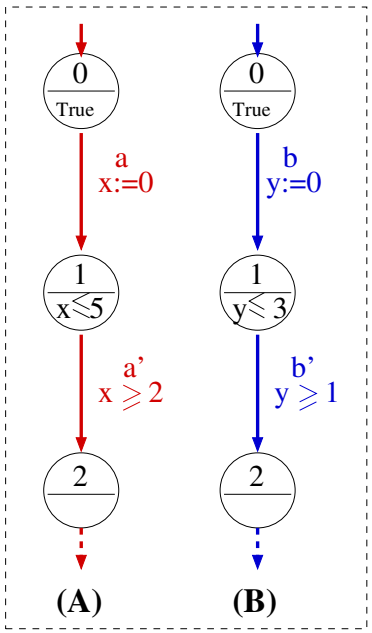

0,0
a
x:=0
b
y:=0
1,0
0,1
a'
x≥ 2
b
y:=0
a
x:=0
b'
y≥ 1
2,0
aa'||ε
1,1
a ||b
0,2
ε ||bb'
Waiting
y
x
0
5
3
2
oo
1

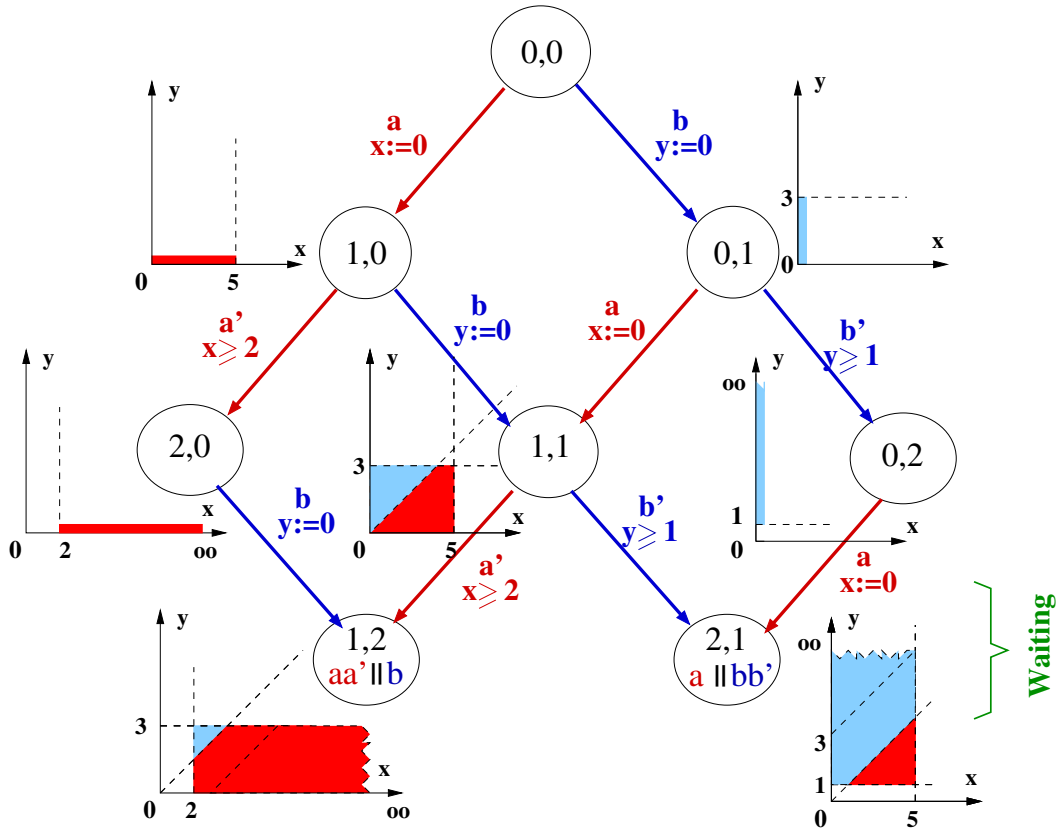
Example


0
True
a
x:=0
1
x≤5
a'
x ≥ 2
2
(A)
0
True
b
y:=0
1
y≤3
b'
y ≥ 1
2
(B)

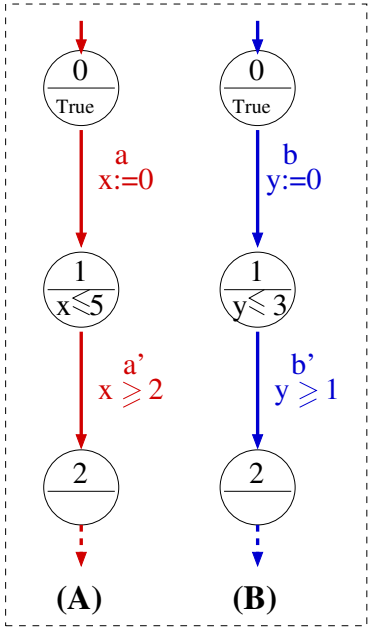

0,0
a
x:=0
b
y:=0
1,0
0,1
a'
x≥ 2
b
y:=0
a
x:=0
b'
y≥ 1
2,0
aa'||ε
1,1
a ||b
0,2
ε ||bb'
Waiting
b
y:=0
a'
x≥ 2
b'
y≥ 1
a
x:=0
1,2
aa'||b
1,2
aa'||b
2,1
a ||bb'
2,1
a ||bb'
New

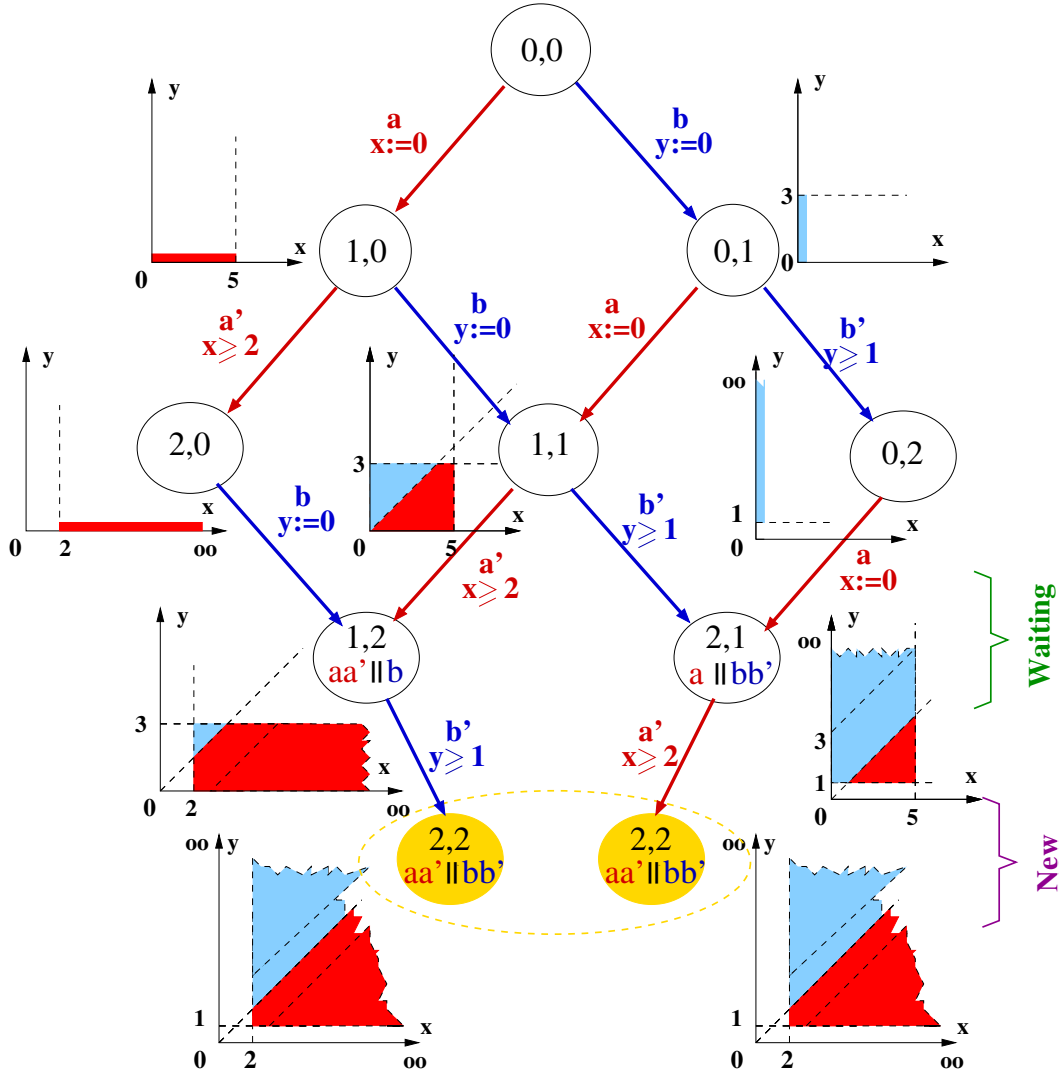
Example


0
True
a
x:=0
1
x≤5
a'
x ≥ 2
2
(A)
0
True
b
y:=0
1
y≤ 3
b'
y ≥ 1
2
(B)

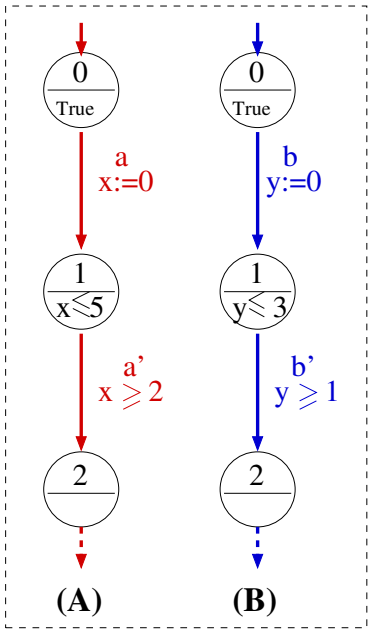

0,0
a
x:=0
b
y:=0
1,0
0,1
a'
x≥ 2
b
y:=0
a
x:=0
b'
y≥ 1
2,0
1,1
0,2
b
y:=0
b'
y≥ 1
a'
x≥ 2
a
x:=0
1,2
aa'||b
2,1
a ||bb'
Waiting
x
y
oo
0
2
3
5
1

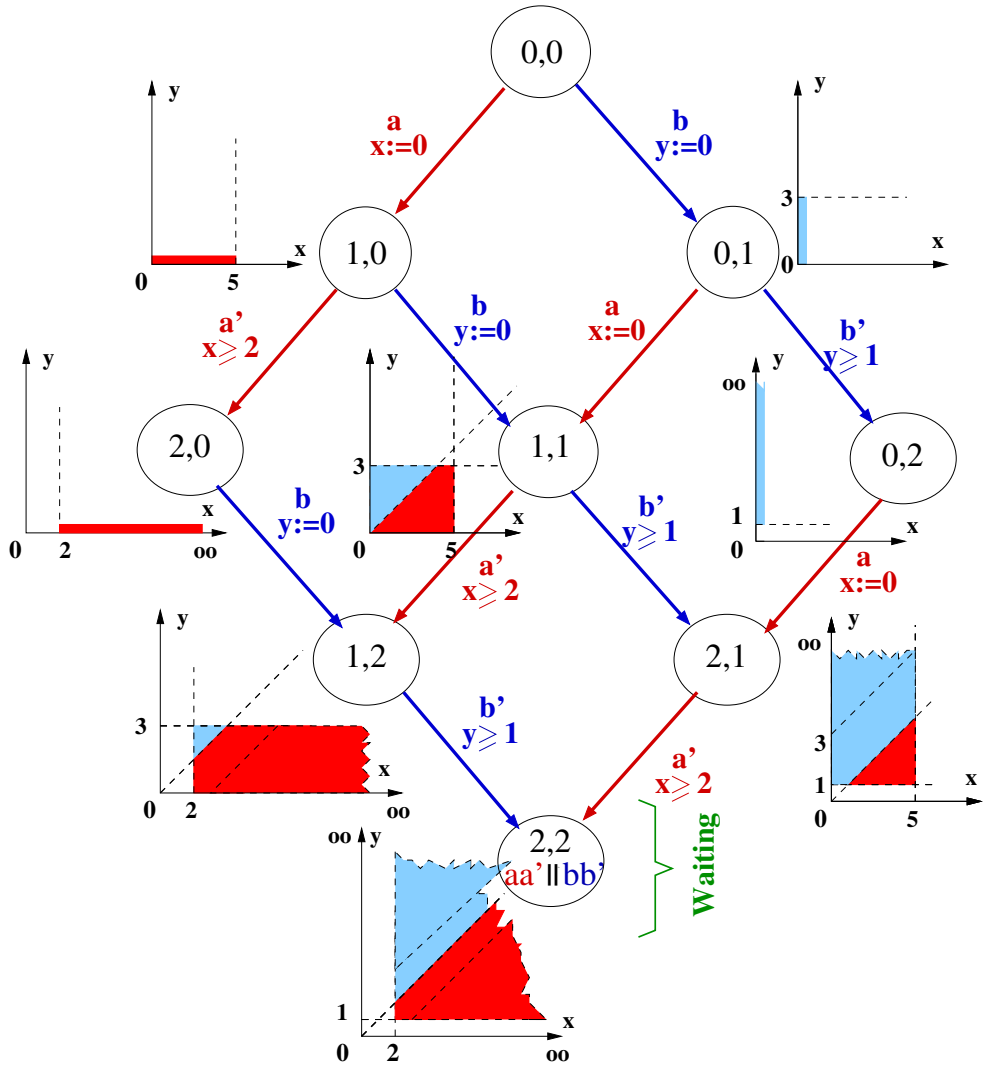
Example


0
True
a
x:=0
1
x≤5
a'
x ≥ 2
2
(A)
0
True
b
y:=0
1
y≤ 3
b'
y ≥ 1
2
(B)


0,0
a
x:=0
b
y:=0
1,0
0,1
a'
x≥ 2
b
y:=0
a
x:=0
b'
y≥ 1
2,0
1,1
0,2
b
y:=0
b'
y≥ 1
a'
x≥ 2
a
x:=0
1,2
aa'||b
2,1
a ||bb'
b'
y≥ 1
a'
x≥ 2
2,2
aa'||bb'
2,2
aa'||bb'
Waiting
New

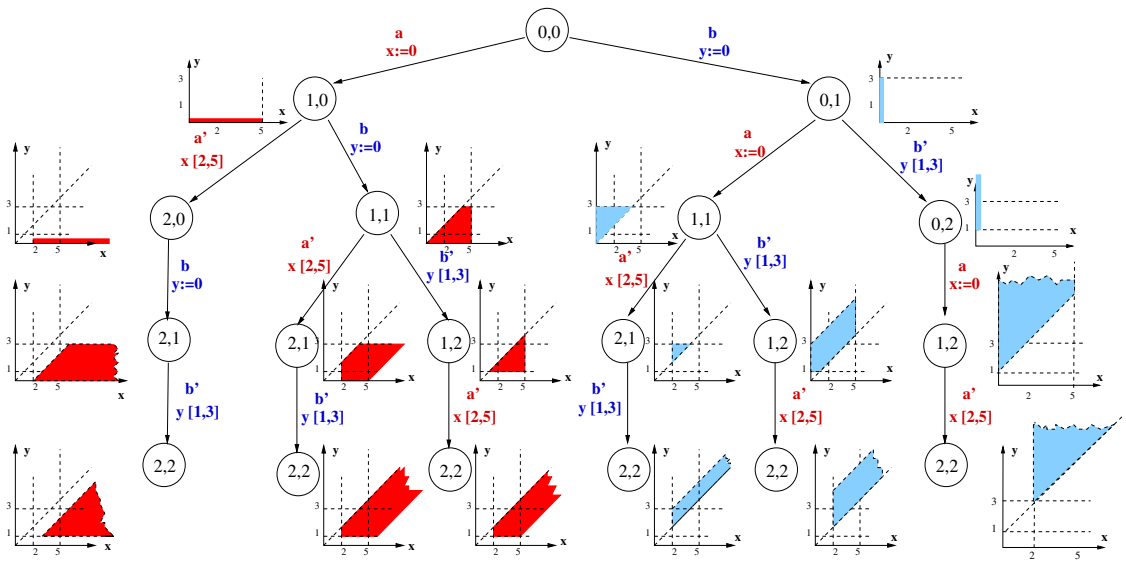
Example


0
True
a
x:=0
1
x≤5
a'
x ≥ 2
2
(A)
0
True
b
y:=0
1
y≤ 3
b'
y ≥ 1
2
(B)

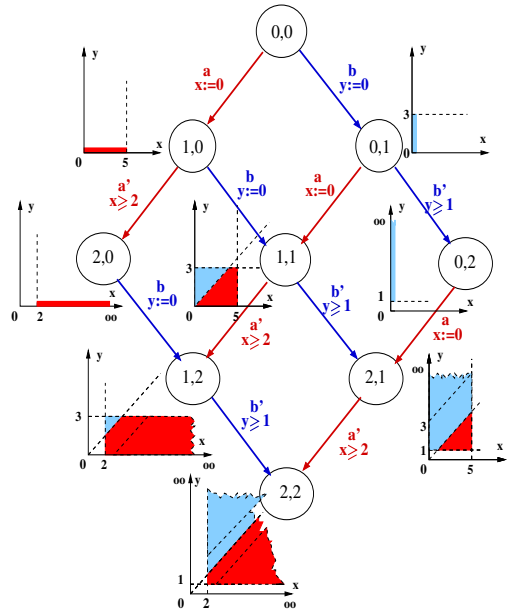

0,0
a
x:=0
b
y:=0
1,0
0,1
a'
x≥ 2
b
y:=0
a
x:=0
b'
y≥ 1
2,0
1,1
0,2
b
y:=0
b'
y≥ 1
a'
x≥ 2
a
x:=0
1,2
2,1
b'
y≥ 1
a'
x≥ 2
2,2
aa'||bb'
Waiting
x
y
oo
0
2
3
5
1

Comparing Standard and Improved Algorithms Results

Standard Algorithm

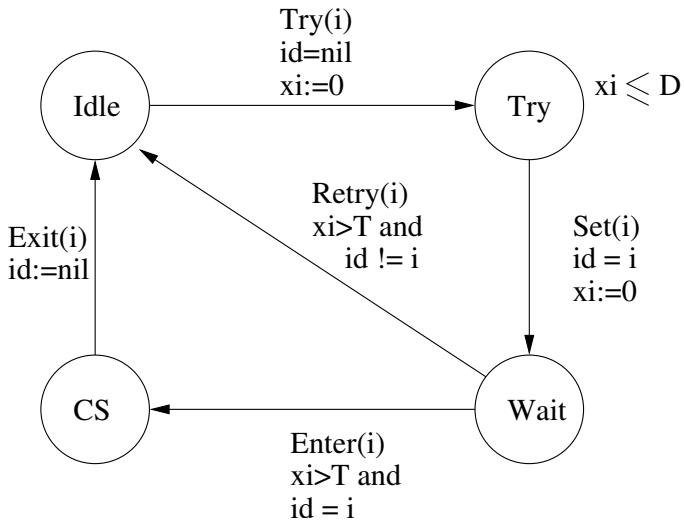


Improved Algorithm



- We have the same reachable state space with much less symbolic states: 9 instead of 19.
- In this example we do not exceed the discrete explosion (3×3).

Experimental Results: The Fischer protocol.



Size	Standard (**states**/time)	Improved (**states**/time)
2	**29**/0.003s	**18**/0.002s
3	**165**/0.01s	**53**/0.01s
4	**1099**/0.07s	**164**/0.03s
5	**8453**/1.07s	**527**/0.04s
6	**74939**/21.06s	**1726**/0.20s
7	**762429**/595.75s	**5693**/1.75s
8	⊥/⊥	**18792**/5.73s
9	⊥/⊥	**61883**/28.42s
10	⊥/⊥	**202994**/367.76s
11	⊥/⊥	**662873**/4489.23s

- The improved algorithm performs exponentially better than the standard one.
- Its performance is similar to UPPAAL or Kronos when the convex-hull approximation is employed.
- Our result shows that convex-hull approximation can be made exact.

Plan

Contribution / Perspectives

We proposed a remedy to that part of the state explosion problem for TA which is due to the interleaving semantics:

- We proved that the union of all zones reached by interleavings of the same set of transitions is convex.
- We improved the reachability algorithm.
- We implemented this algorithm, and showed through examples it's efficiency.

We detected through this study an interesting subset of Timed Automata where the zones have often a rectangular form.

In the context of circuits modeling this result could be specialized and some abstraction techniques could be improved.

Related Work

- T.G. Rokicki, PhD Thesis,

  Representing and Modeling Digital Circuits,
  Stanford University, 1994.

- J. Bengtsson, B. Jonsson, J. Lilius and W. Yi,

  Partial Order Reductions for Timed Systems,
  CONCUR 98, 485-500, 1998.

- D. Lugiez, P. Niebert and S. Zennou,

  A Partial Order Semantics Approach to the Clock Explosion Problem of Timed Automata,
  Theoretical Computer Science 345, 27-59, 2005.

- J. Zhao,

  Partial Order Path Technique for Checking Parallel Timed Automata,
  FTRTFT 02, 417-432, 2002.

- *Convex-hull approximation* : C. Daws and S. Tripakis,

  Model Checking of Real-Time Reachability Properties Using Abstractions,
  TACAS 98, 313-329, 1998.