INRIA

A Game Approach to Determinize Timed Automata

Amélie Stainer

Supervisors: Nathalie Bertrand and Thierry Jéron
Vertecs

June 24, 2010

Introduction

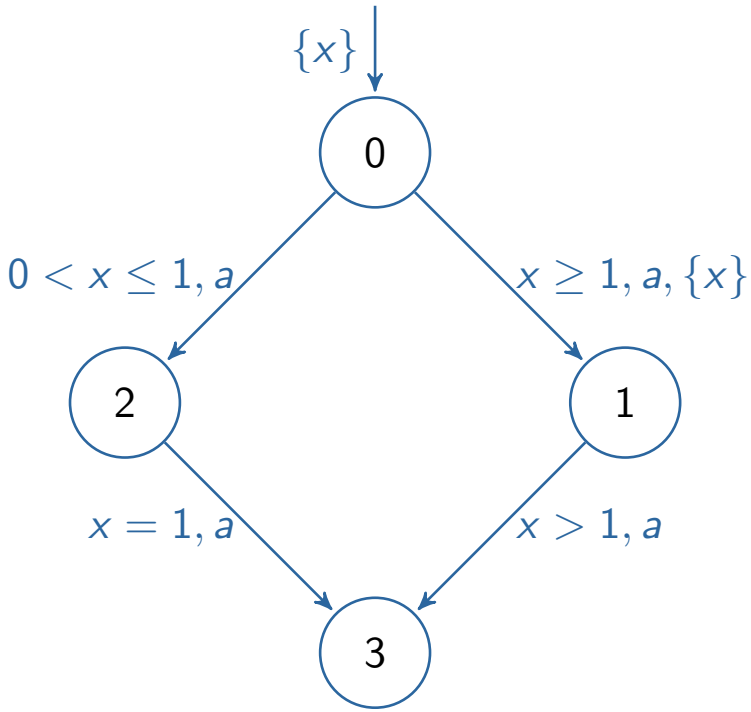
- Timed automata (TA) [AD94]
  - model for real-time systems
  - finite automata equipped with continuous clocks
- Timed automata determinization
  - useful for test generation
  - allows to decide traces inclusion

[AD94] Alur and Dill. A theory of Timed Automata. 1994

INRIA

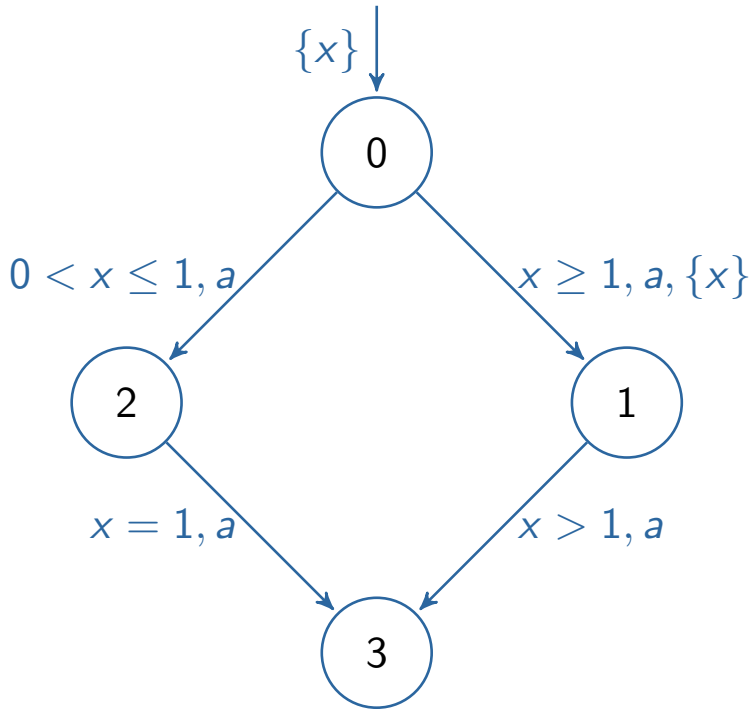
Timed automata

Example (A timed automaton $\mathcal{A}$)



Traces

Example (Some traces of $\mathcal{A}$)



Traces of $\mathcal{A}$:

- 1.a.0.a.12
- 2.a.1.a.2

Remark : $\mathcal{A}$ is not deterministic.

Determinizability

- TAs are not determinizable in general.
- determinizability problem for TAs is undecidable.

$\Rightarrow$ 2 approaches :

- exact determinization which does not always terminate [BBBB09]
  - mappings from original clocks to new ones
- deterministic over-approximation which terminates [KT09]
  - relations between original clocks and new ones
  - fixed resetting policy

[BBBB09] Baier, Bertrand, Bouyer and Brihaye. When are Timed Automata Determinizable? 2009
[KT09] Krichen and Tripakis. Conformance testing for real-time systems. 2009

Our approach

- Goal : combine both methods.
  - fixed ressources (number of clocks and maximal constant),
  - determinization if possible,
  - deterministic over-approximation otherwise.
- Method :
  - same computation as [KT09]
  - a game to choose when to reset clocks
  - inspired by the approach [BCD05] to decide diagnosability of TAs

In each state, we store relations between original clocks and clocks of our construction.

[BCD05] Bouyer, Chevalier, D'Souza. Fault diagnosis using timed automata. 2005

Game definition

- A finite turn-based safety game
- 2 players : Spoiler and Determinizator
  - Spoiler chooses an action and when to fire it.
  - Determinizator chooses the resets.
    $\Rightarrow$ a choice of Spoiler + a choice of Determinizator = one transition
- Determinizator wants to avoid states built by over-approximation.

Properties of the game

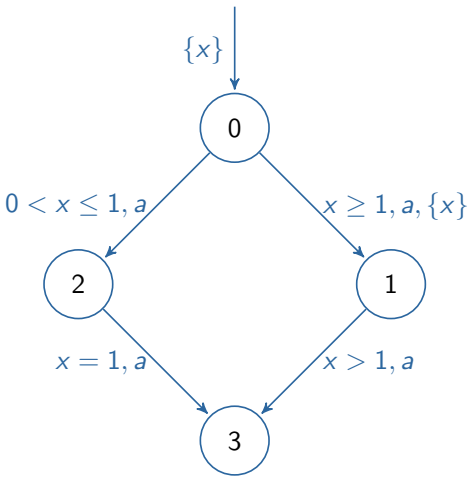
- A strategy S for Determinizator corresponds to a deterministic timed automaton $Aut(S)$ (by merging of each choice of Spoiler with the next choice of Determinizator).

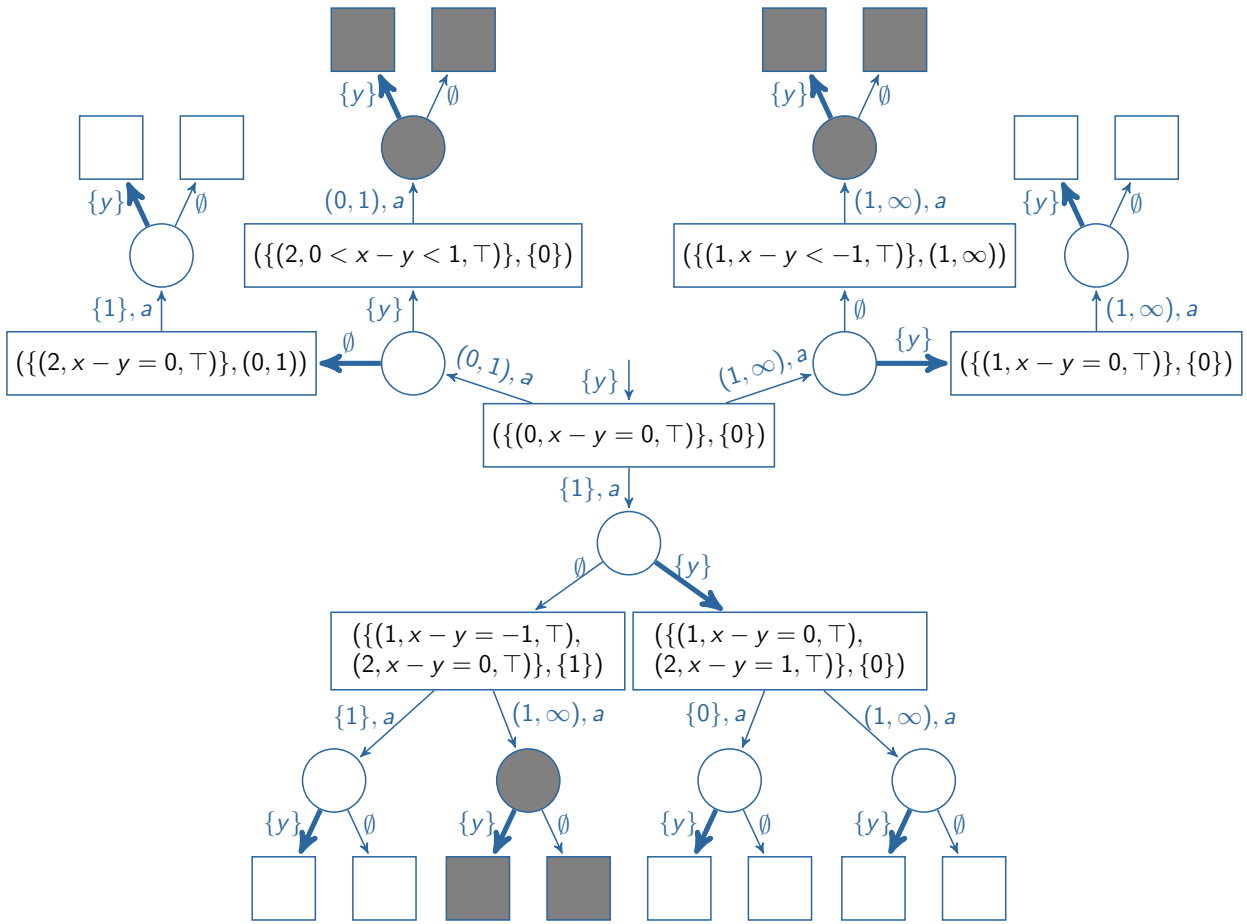
Properties

- A strategy for Determinizator yields an over-approximation.
- A winning strategy for Determinizator yields an exact determinization.

An example

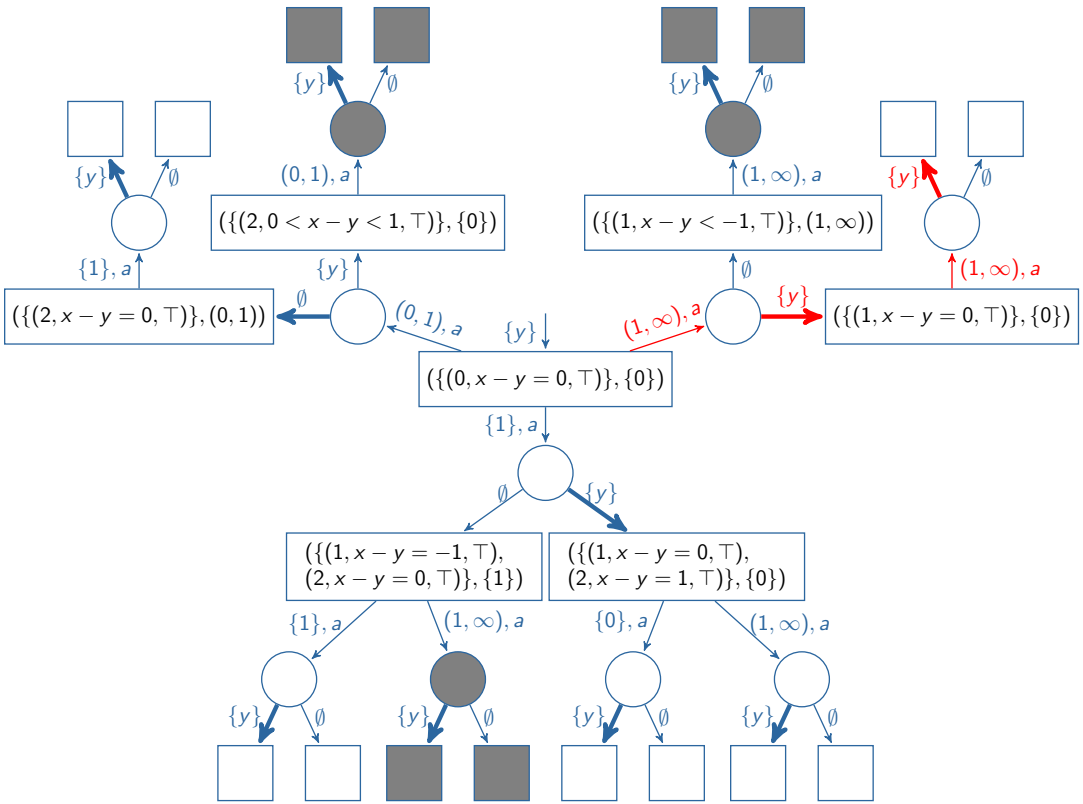
Example (Game corresponding to $\mathcal{A}$ and a winning strategy for Determinizator)

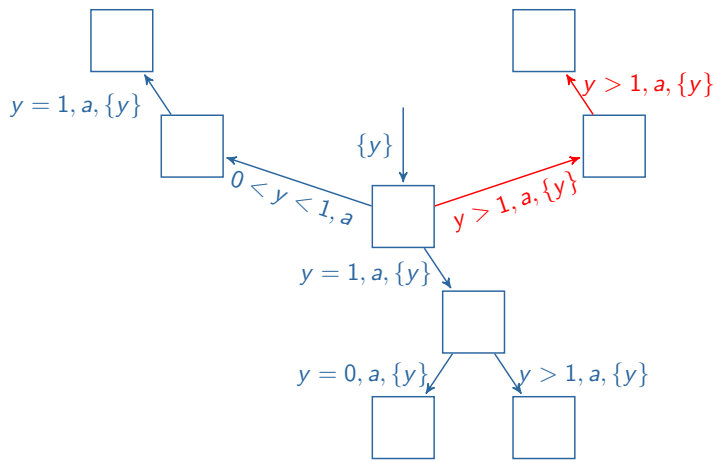




An example

Example (DTA corresponding to a winning strategy)





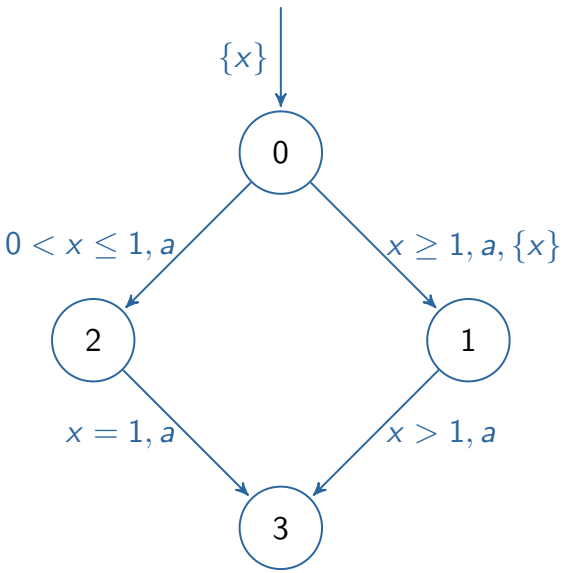
States of the game

- a state = the set of all configurations in which we can be and a region over the clocks of the construction
- a configuration = (ℓ, R, b)
  - ℓ is a location of the original automaton
  - R is a clocks relation

    a conjunction of elementary relations of the form $(x_{\mathcal{A}} - y_{\mathcal{B}} \sharp c)$, where $\sharp \in \{>, <, =\}$
  - b is a marker

    $b = \begin{cases} \top & \text{if exactitude is sure} \\ \bot & \text{if there possibly is an approximation} \end{cases}$

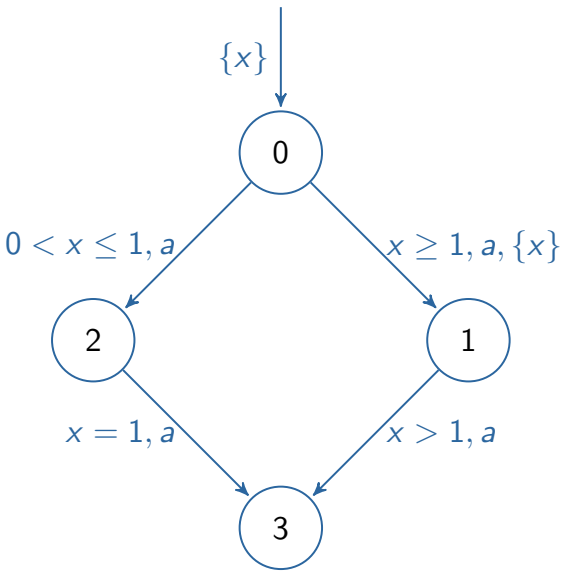
The game construction

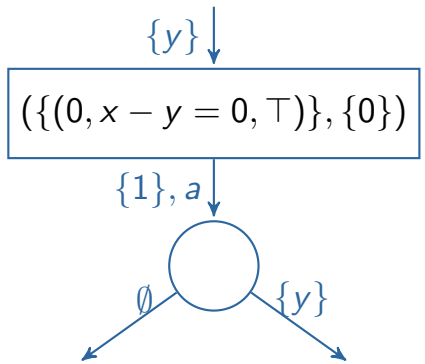


$\{y\}\downarrow$

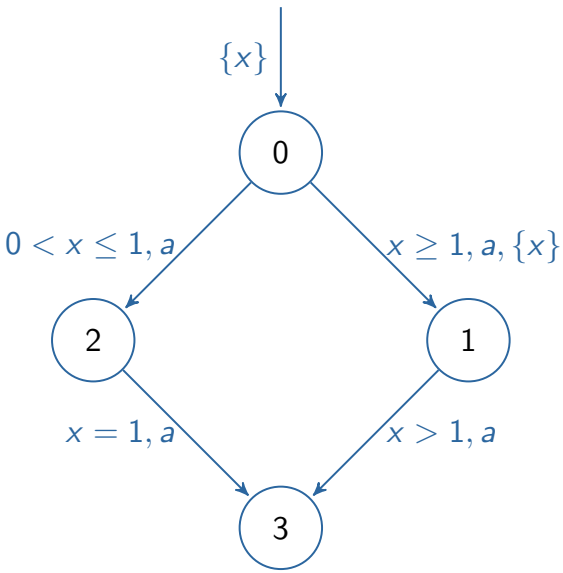
$(\{(0, x - y = 0, \top)\}, \{0\})$

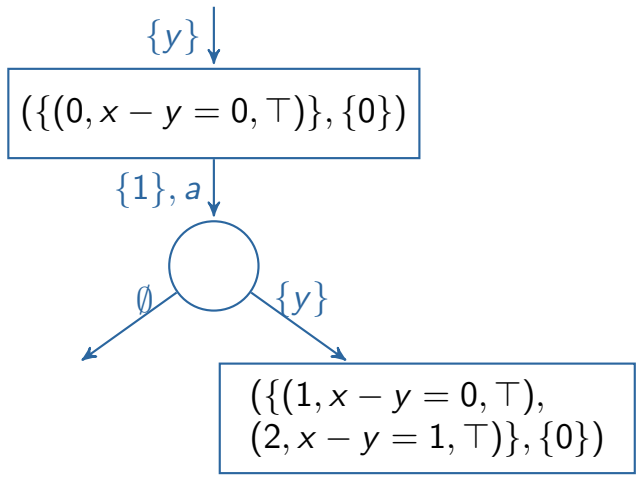
The game construction



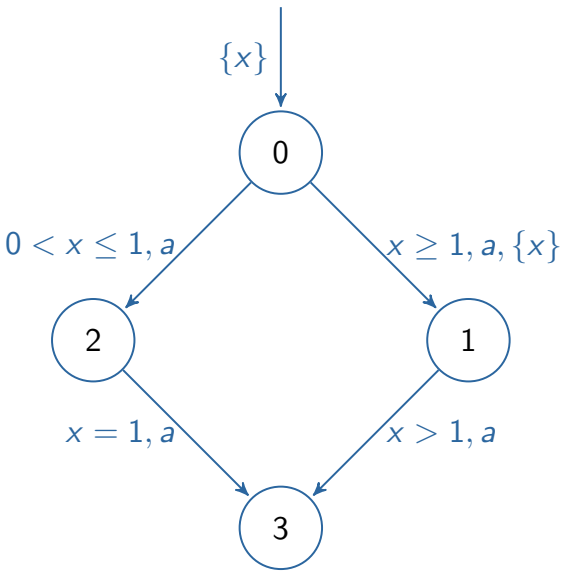


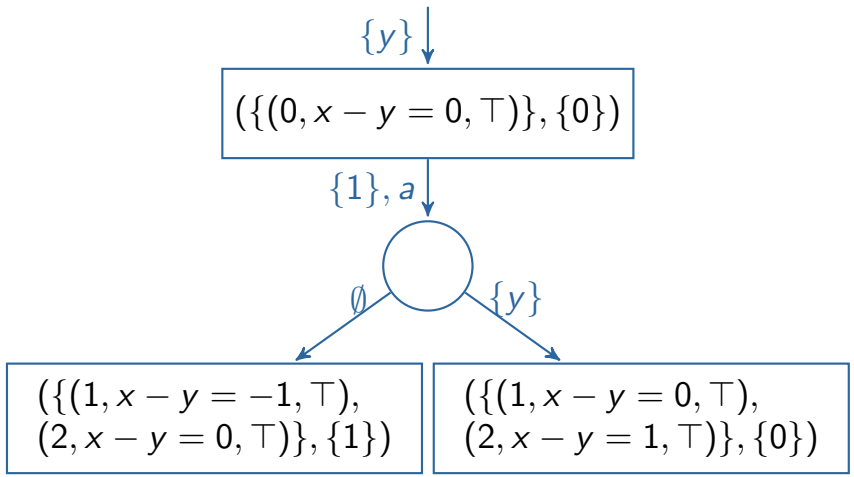
The game construction



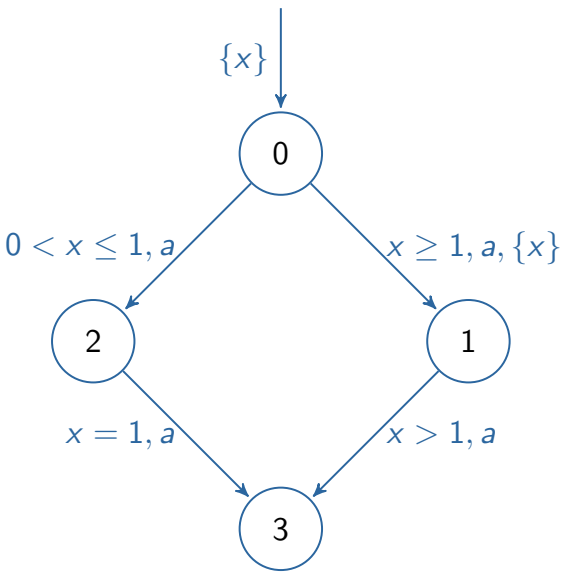


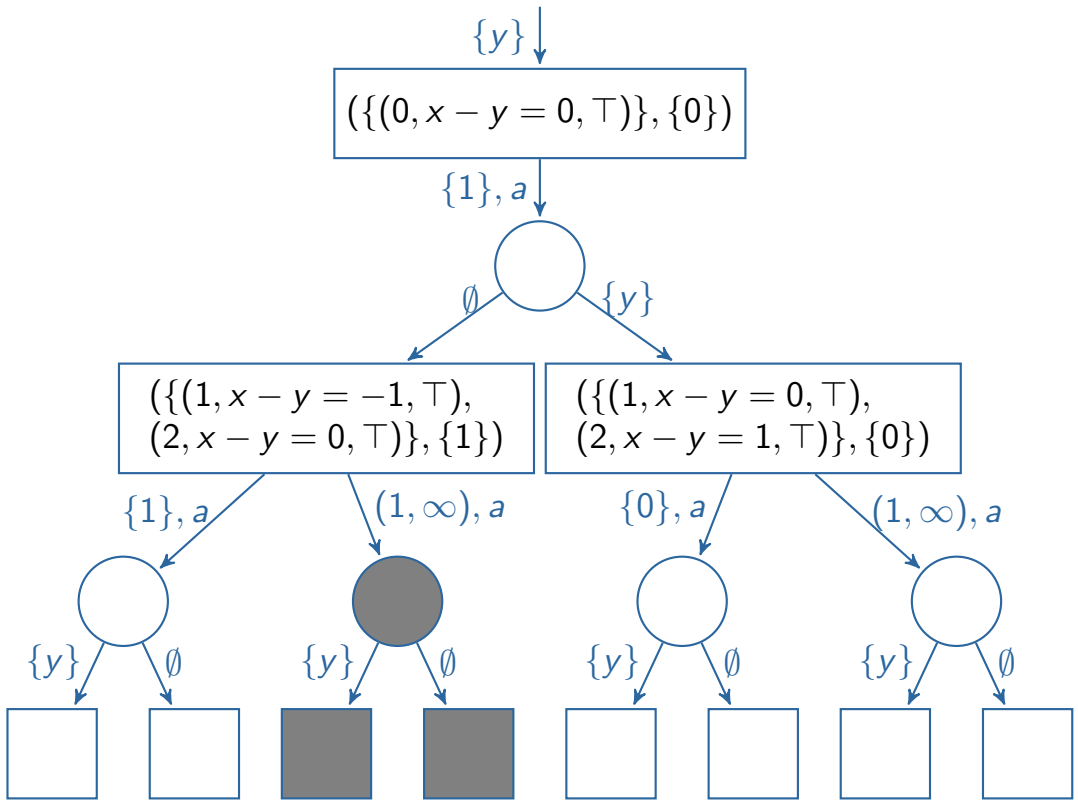
The game construction





The game construction





Comparison

- More precise than the over-approximation [KT09]:
  $\Rightarrow$ preserves the determinism
- Strictly more general than the determinization procedure [BBBB09]:
  - our relations between clocks are more flexible than the mapping
    $\Rightarrow$ extension of the class of timed automata we can treat
  - marking of each configuration
    $\Rightarrow$ dealing with some traces inclusion

Conclusion

- Contribution: a determinization algorithm for timed automata
  - exact if possible and always terminating
  - improving two existing methods with the same order of size for the result
- Paralell work:
  - adaptation to extended models (accepting locations, urgency, invariants...)
  - application to testing of timed automata by partial under-approximation
- Submission to FSTTCS'2010