Computing Optimal Reachability Costs in Priced Timed Pushdown Automata

LATA 2014, March 10-14, Madrid

Jari Stenman
Uppsala University

March 12, 2014

Joint work with Parosh Aziz Abdulla and Mohamed Faouzi Atig

Introduction

Quantitative extensions of discrete systems:

- Time
- Costs
- Probabilities

Time and cost:

- Real-time systems with resource consumption

Introduction

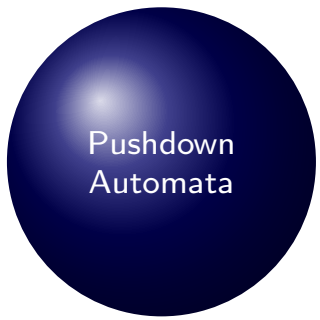
Quantitative extensions of discrete systems:

- Time
- Costs
- Probabilities

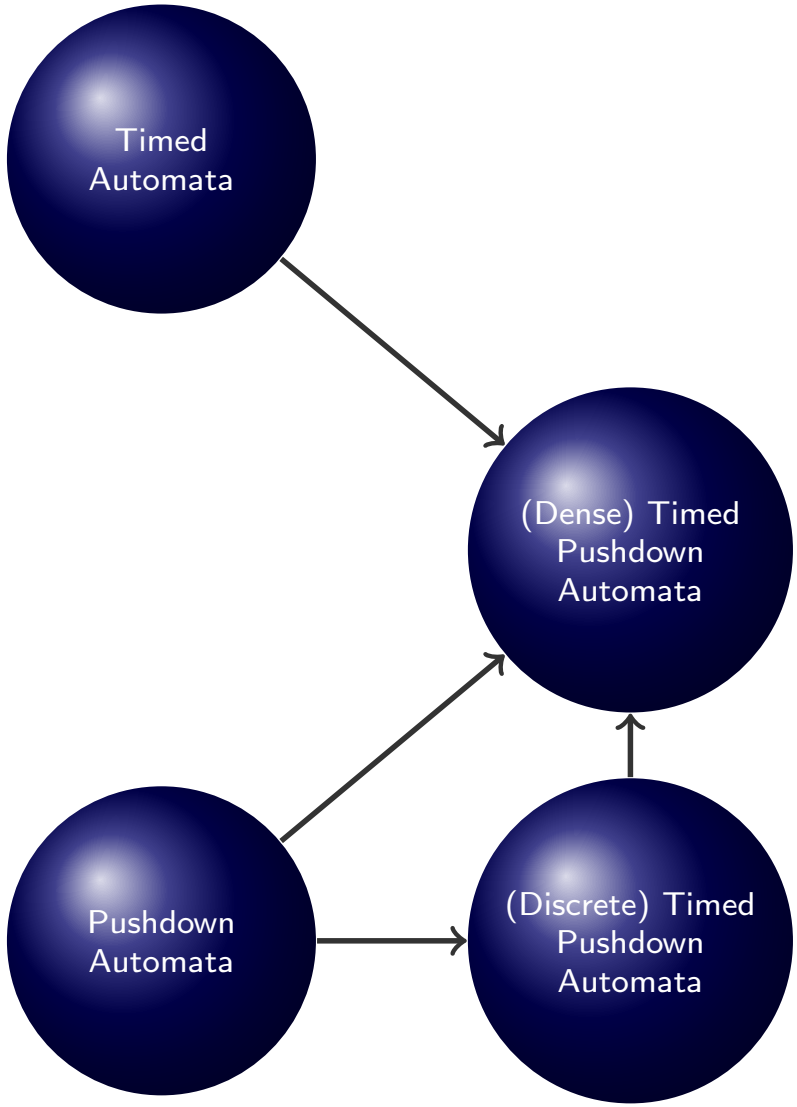
Time and cost:

- Real-time systems with resource consumption
- Priced Timed Automata
- Priced Timed Petri Nets

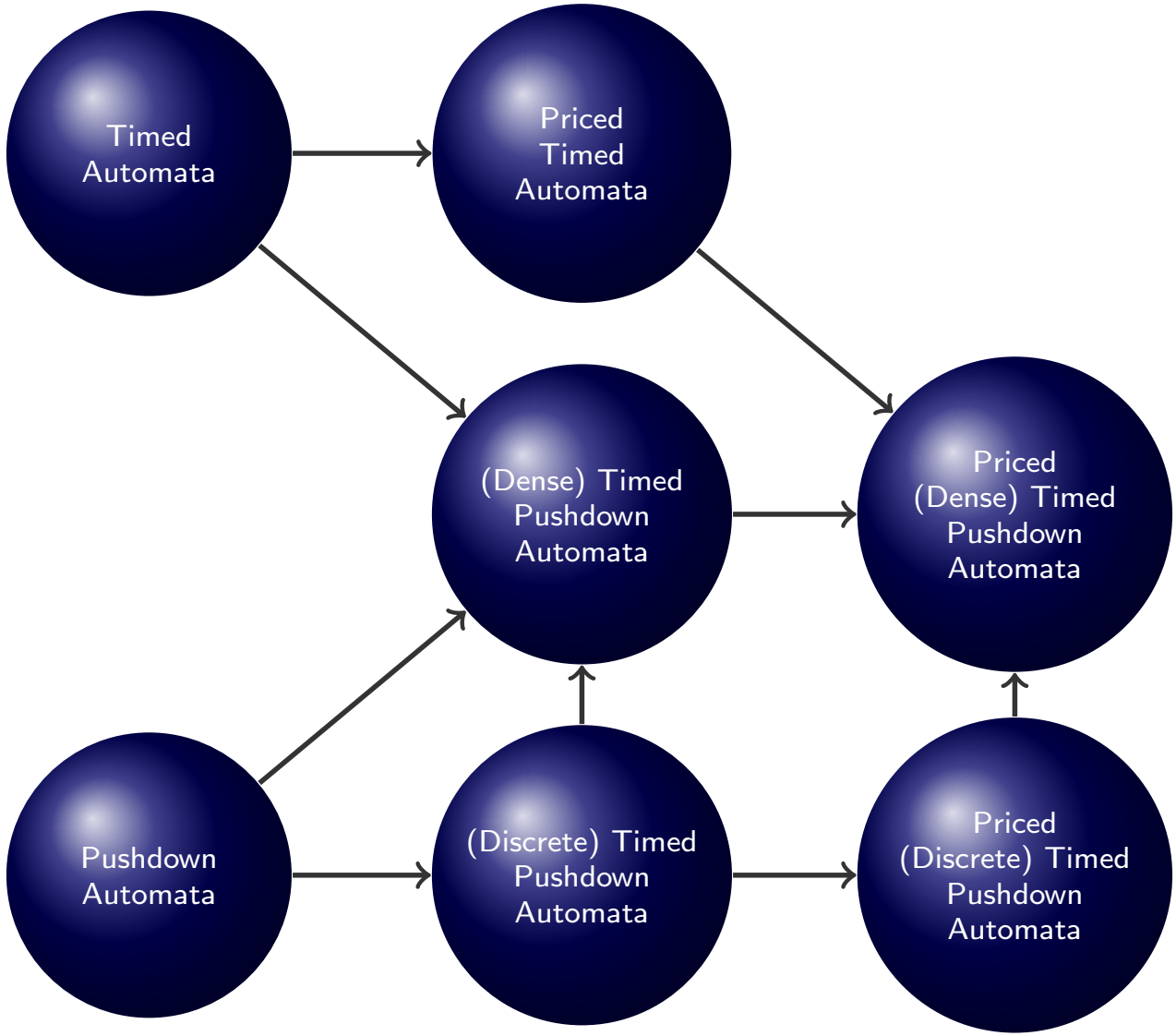
Background



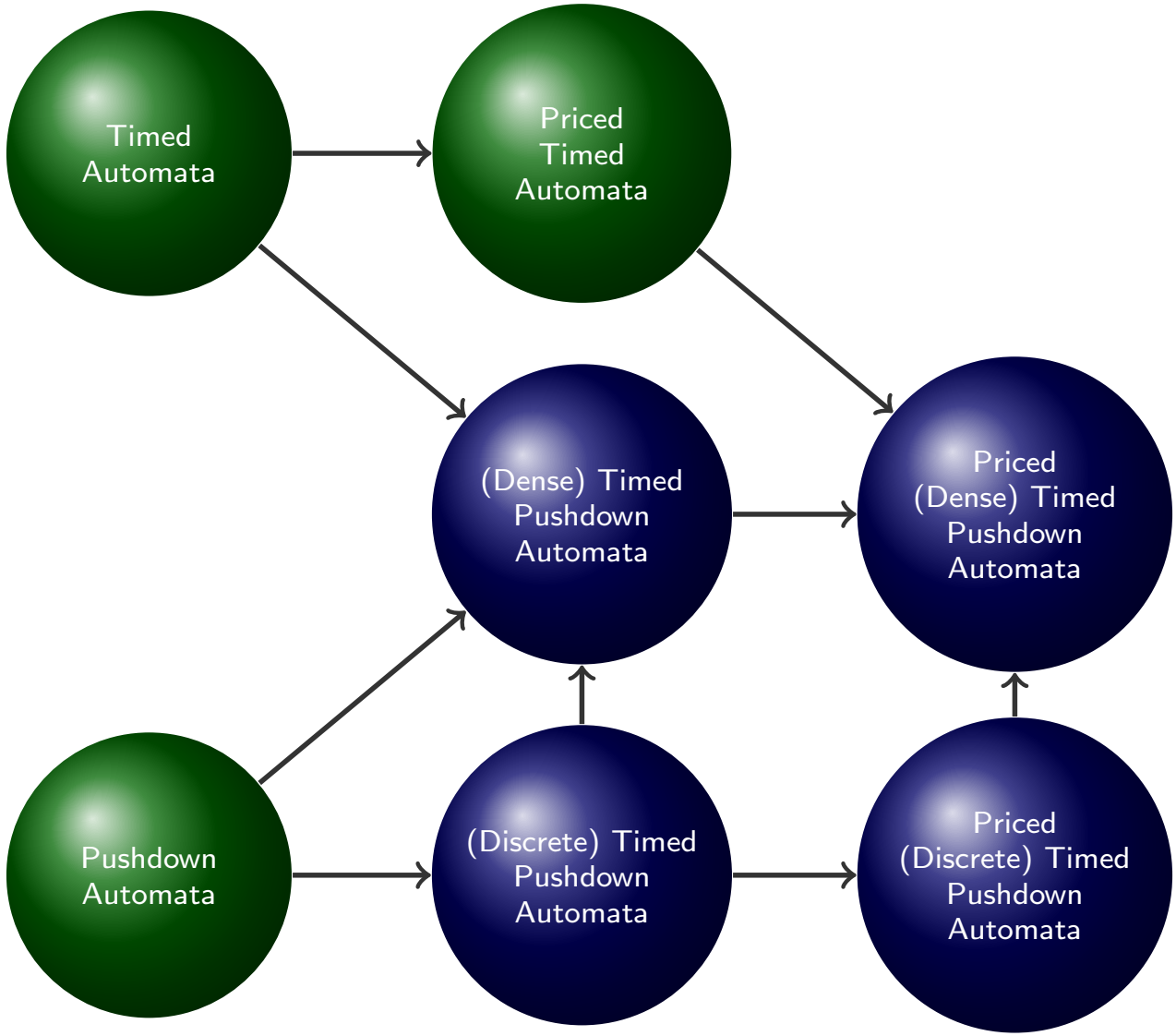
Background


Timed
Automata
(Dense) Timed
Pushdown
Automata
Pushdown
Automata
(Discrete) Timed
Pushdown
Automata

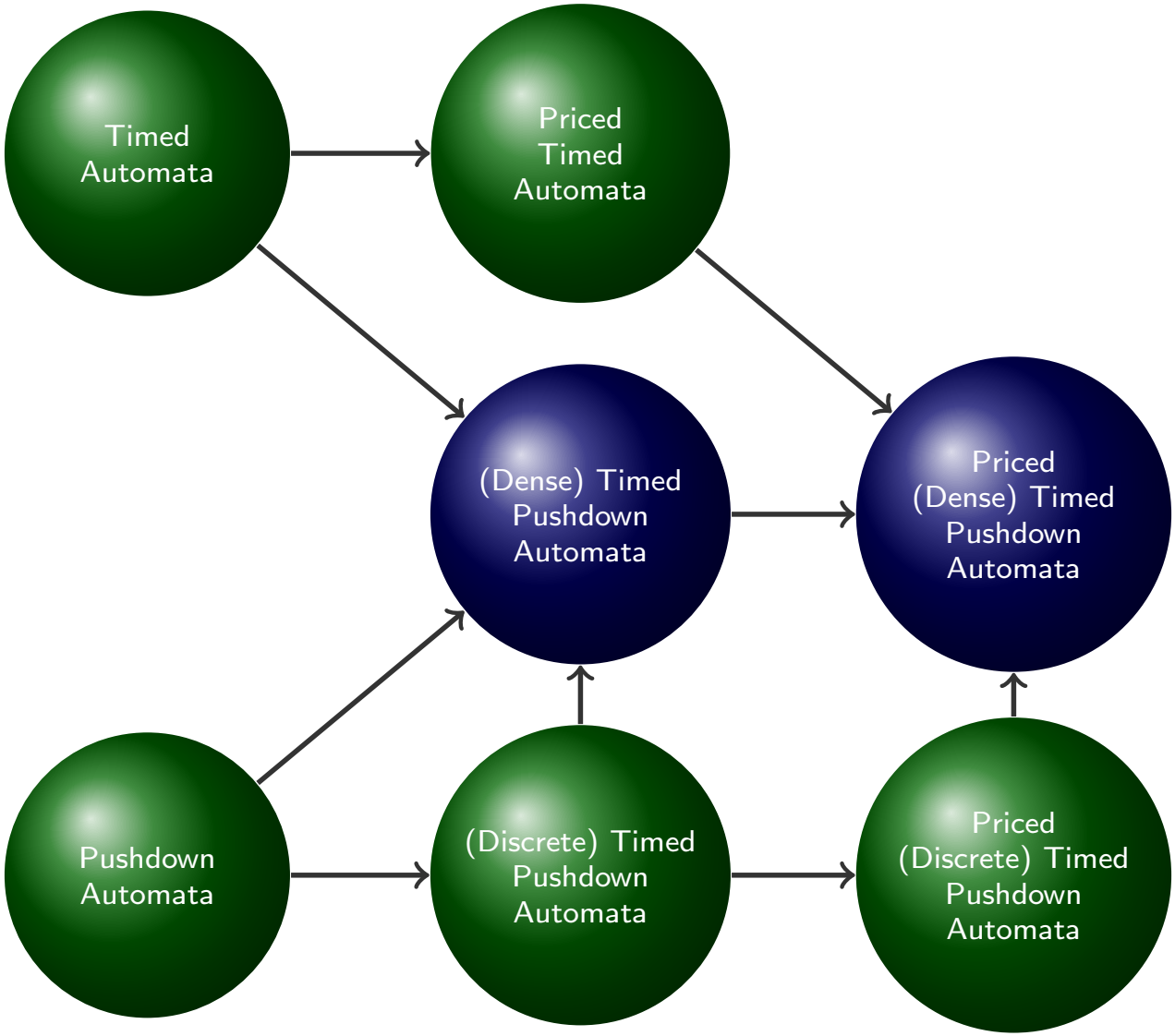
Background


Timed Automata
Priced Timed Automata
(Dense) Timed Pushdown Automata
Priced (Dense) Timed Pushdown Automata
Pushdown Automata
(Discrete) Timed Pushdown Automata
Priced (Discrete) Timed Pushdown Automata

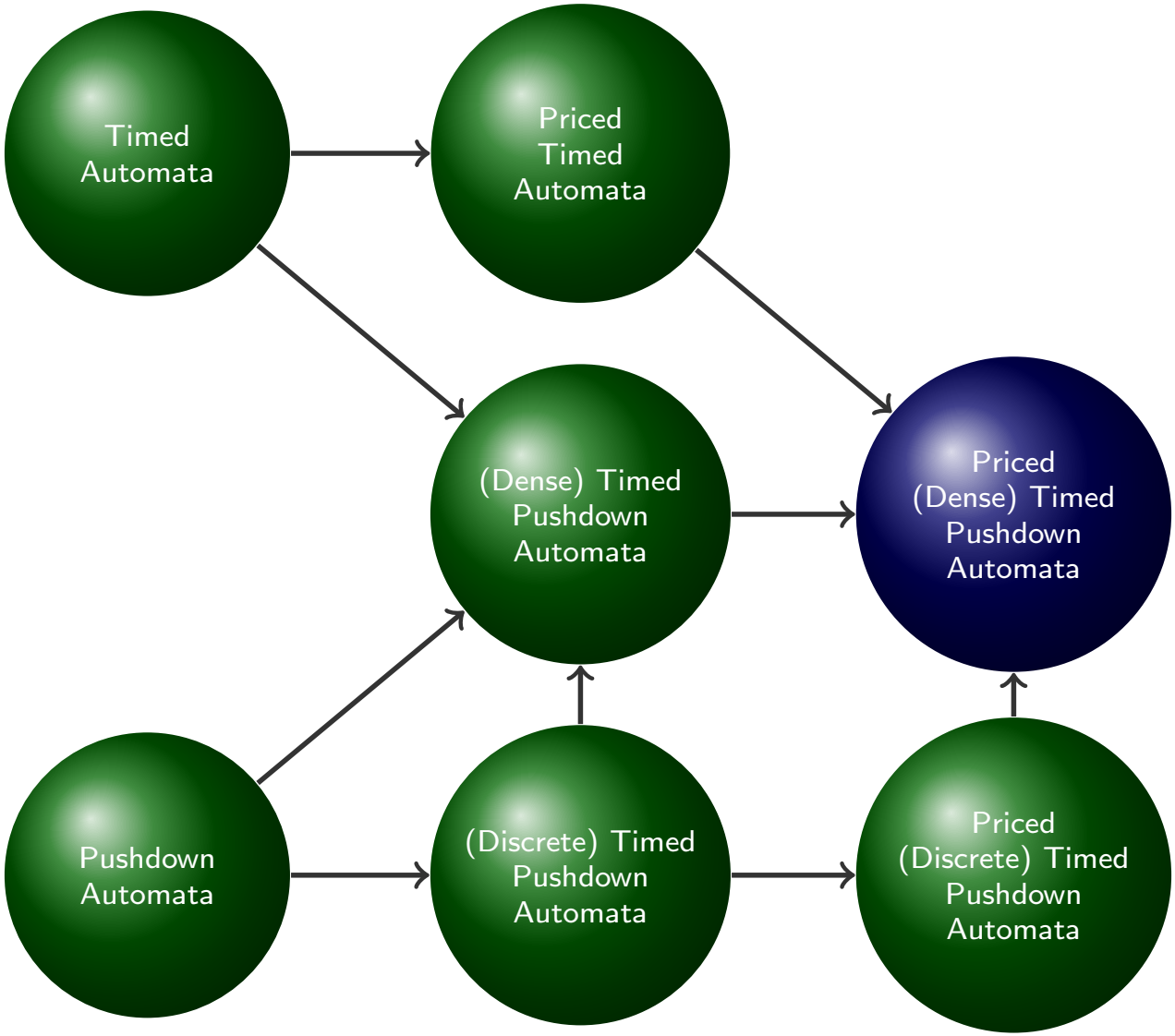
Background


Timed Automata
Priced Timed Automata
(Dense) Timed Pushdown Automata
Priced (Dense) Timed Pushdown Automata
Pushdown Automata
(Discrete) Timed Pushdown Automata
Priced (Discrete) Timed Pushdown Automata

Background


Timed Automata
Priced Timed Automata
(Dense) Timed Pushdown Automata
Priced (Dense) Timed Pushdown Automata
Pushdown Automata
(Discrete) Timed Pushdown Automata
Priced (Discrete) Timed Pushdown Automata

Background


Timed Automata
Priced Timed Automata
(Dense) Timed Pushdown Automata
Priced (Dense) Timed Pushdown Automata
Pushdown Automata
(Discrete) Timed Pushdown Automata
Priced (Discrete) Timed Pushdown Automata

Outline

Outline

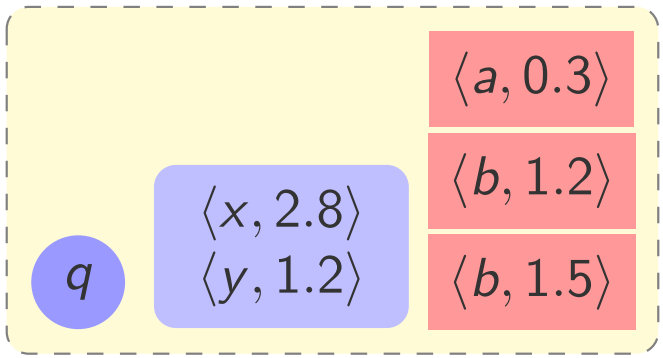
Timed Pushdown Automata

Definition

A *timed pushdown automaton* is a tuple (Q, X, Γ, Δ), where

- Q finite set of states
- X finite set of clocks
- Γ finite stack alphabet
- Δ finite set of transition rules

TPDA Configuration



TPDA Computations

Transition Rules:

$(q_1, \mathbf{op}, q_2)$ where **op** is one of the following:

nop	No operation
push(a)	Pushes a with initial age 0
pop(a, I)	Pops $\langle a, h \rangle$ if $h \in I$
$x \in I\,?$	Allowed if $v(x) \in I$
$x \leftarrow 0$	Sets $v(x)$ to 0

TPDA Computations

Transition Rules:

$(q_1, \mathbf{op}, q_2)$ where **op** is one of the following:

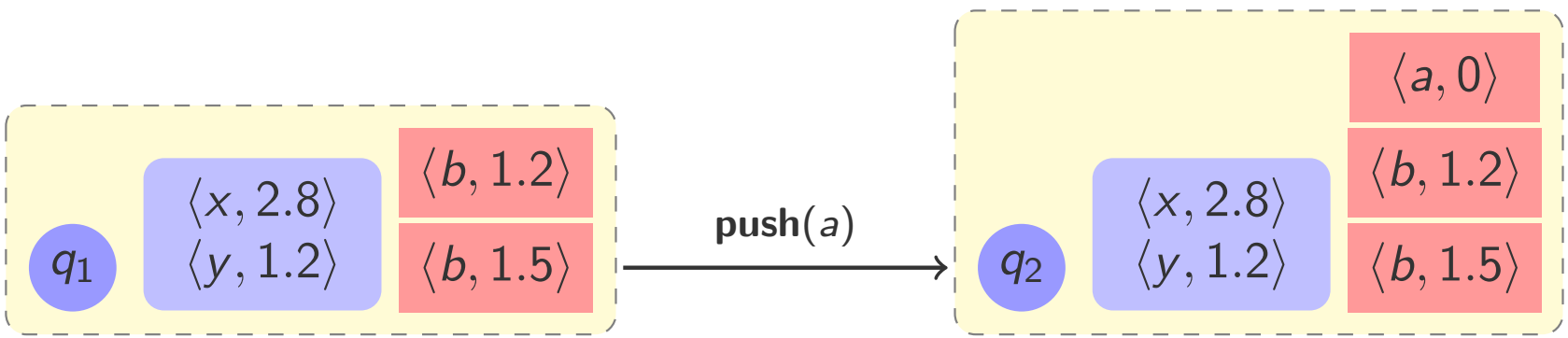
nop	No operation
push(a)	Pushes a with initial age 0
pop(a, I)	Pops $\langle a, h \rangle$ if $h \in I$
$x \in I\,?$	Allowed if $v(x) \in I$
$x \leftarrow 0$	Sets $v(x)$ to 0

TPDA Computations

Transition Rules:

$(q_1, \mathbf{op}, q_2)$ where **op** is one of the following:

nop	No operation
push(a)	Pushes a with initial age 0
pop(a, I)	Pops $\langle a, h \rangle$ if $h \in I$
$x \in I\,?$	Allowed if $v(x) \in I$
$x \leftarrow 0$	Sets $v(x)$ to 0

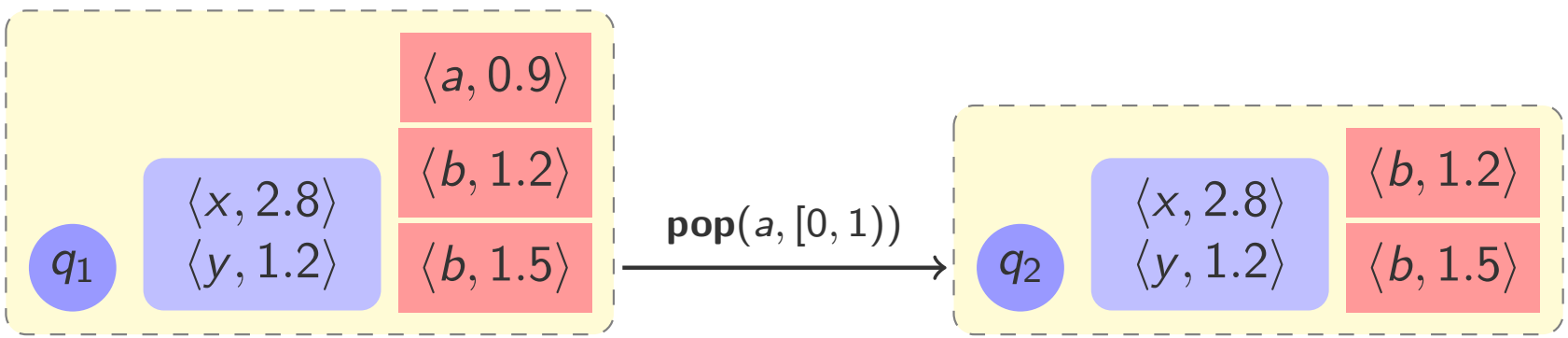


TPDA Computations

Transition Rules:

$(q_1, \mathbf{op}, q_2)$ where **op** is one of the following:

nop	No operation
push(a)	Pushes a with initial age 0
pop(a, I)	Pops $\langle a, h \rangle$ if $h \in I$
$x \in I\,?$	Allowed if $v(x) \in I$
$x \leftarrow 0$	Sets $v(x)$ to 0

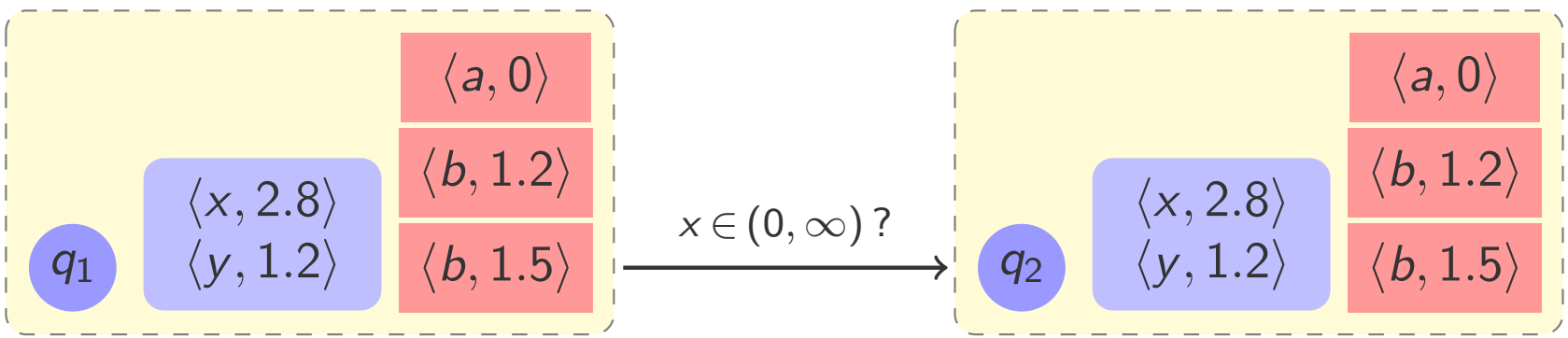


TPDA Computations

Transition Rules:

$(q_1, \mathbf{op}, q_2)$ where **op** is one of the following:

nop	No operation
push(a)	Pushes a with initial age 0
pop(a, I)	Pops $\langle a, h \rangle$ if $h \in I$
$x \in I\,?$	Allowed if $v(x) \in I$
$x \leftarrow 0$	Sets $v(x)$ to 0

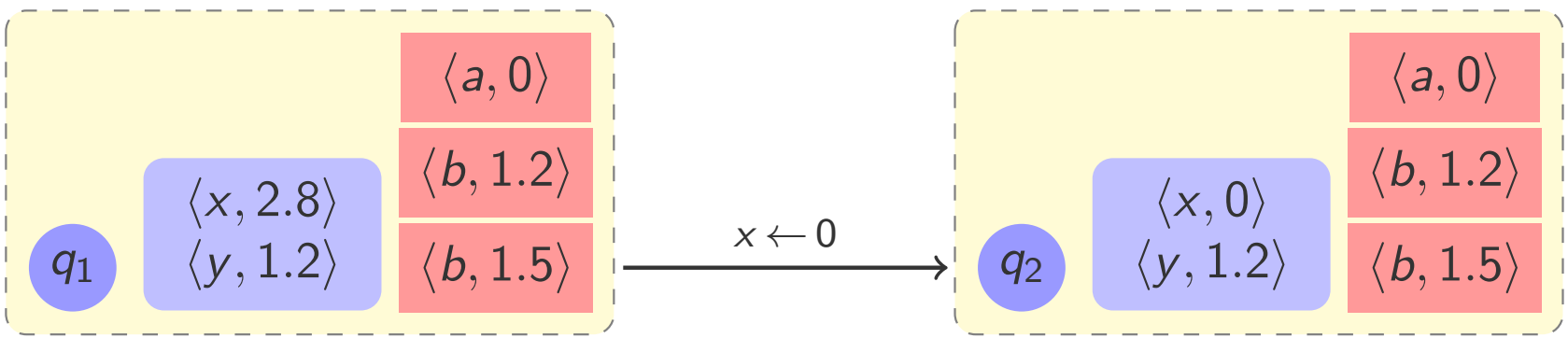


TPDA Computations

Transition Rules:

$(q_1, \mathbf{op}, q_2)$ where **op** is one of the following:

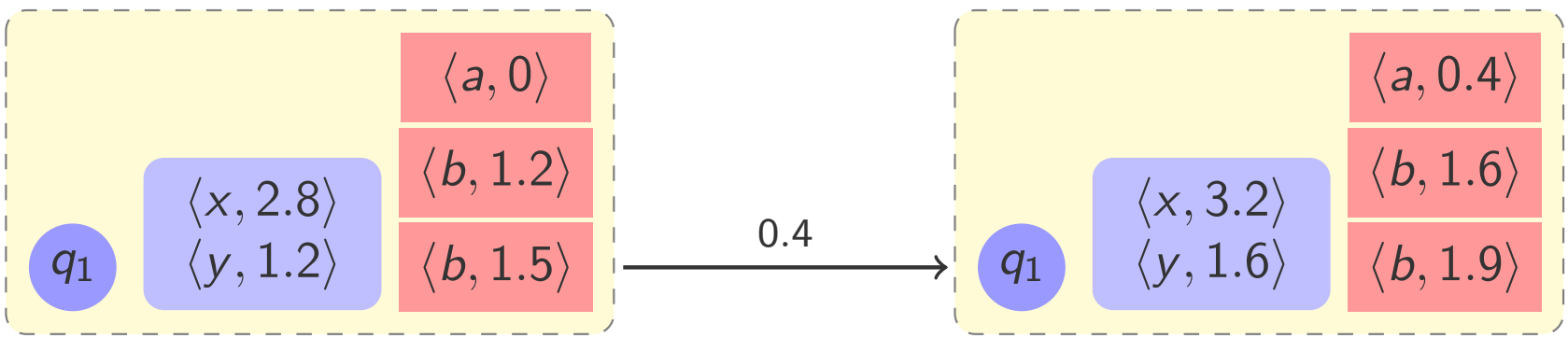
nop	No operation
push(a)	Pushes a with initial age 0
pop(a, I)	Pops $\langle a, h \rangle$ if $h \in I$
$x \in I$?	Allowed if $v(x) \in I$
$x \leftarrow 0$	Sets $v(x)$ to 0



TPDA Computations

Timed Transitions

At any time, the automaton may take a timed transition.



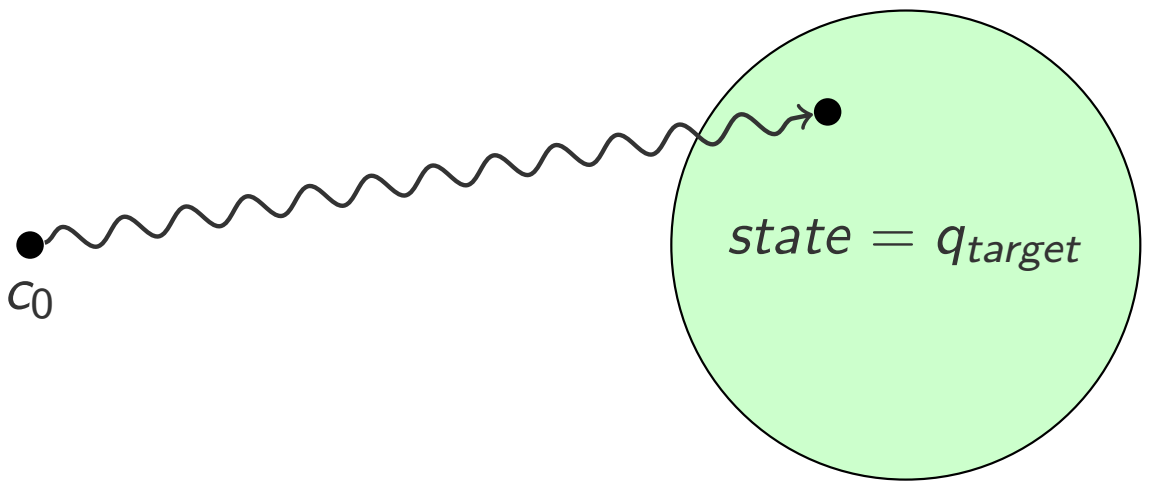
Reachability

Definition (The Reachability Problem)

Given:

- Model M
- Initial configuration c_0
- Target state q_{target}

Decide if there is a computation from c_0 to some configuration in which $state = q_{target}$



Reachability for TPDA

Theorem (LICS 2012)

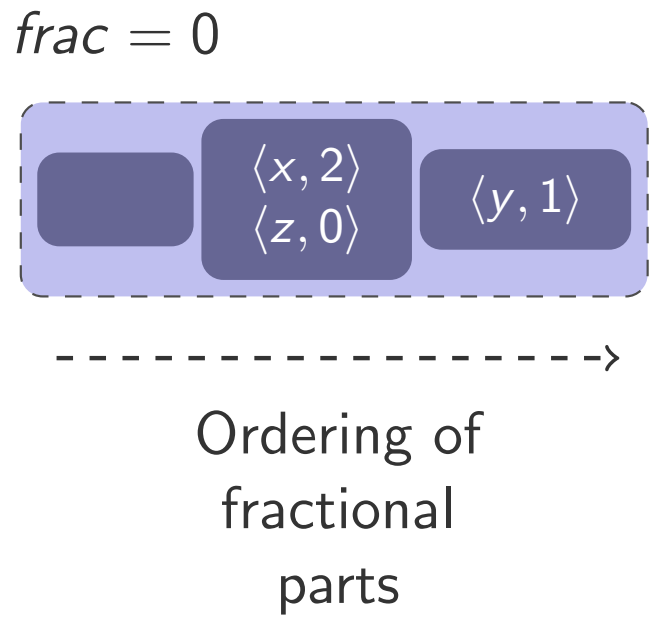
Reachability for TPDA is ExpTime*-complete:*

- *Membership: Construct simulating PDA (size exponential in size of TPDA)*
- *Hardness: Reduction from checking emptiness of intersection of PDA with n regular languages $L(A_1), \ldots, L(A_n)$*

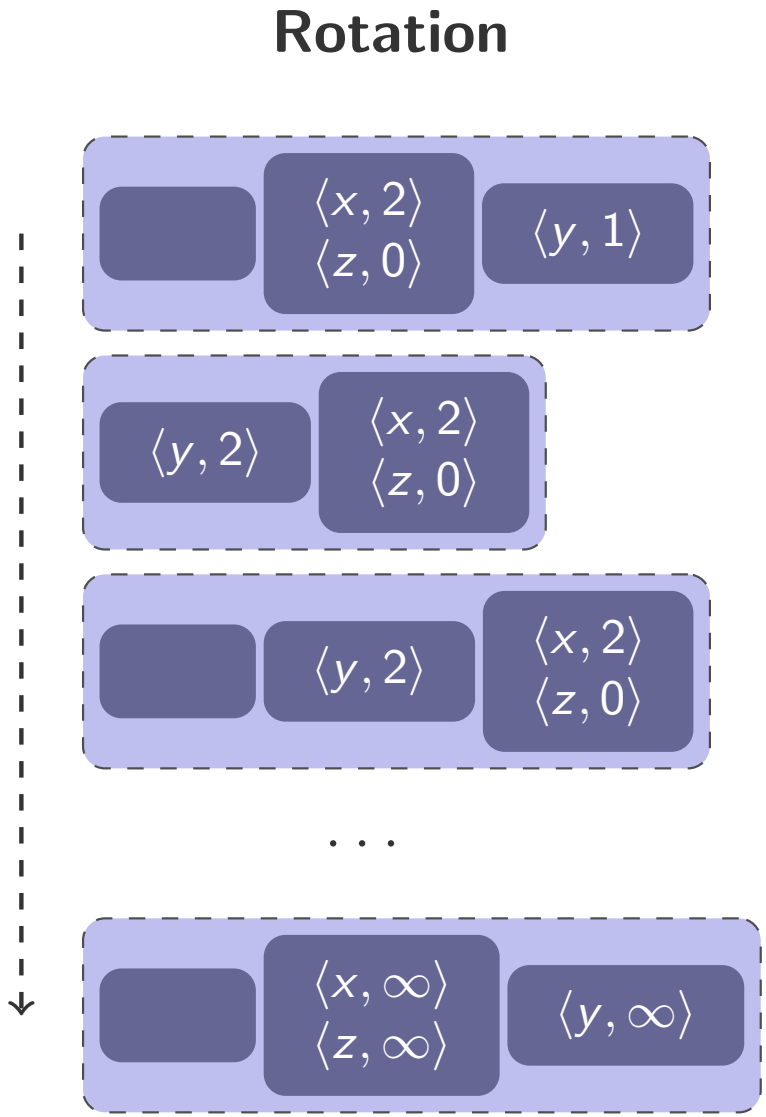
Regions

Equivalence

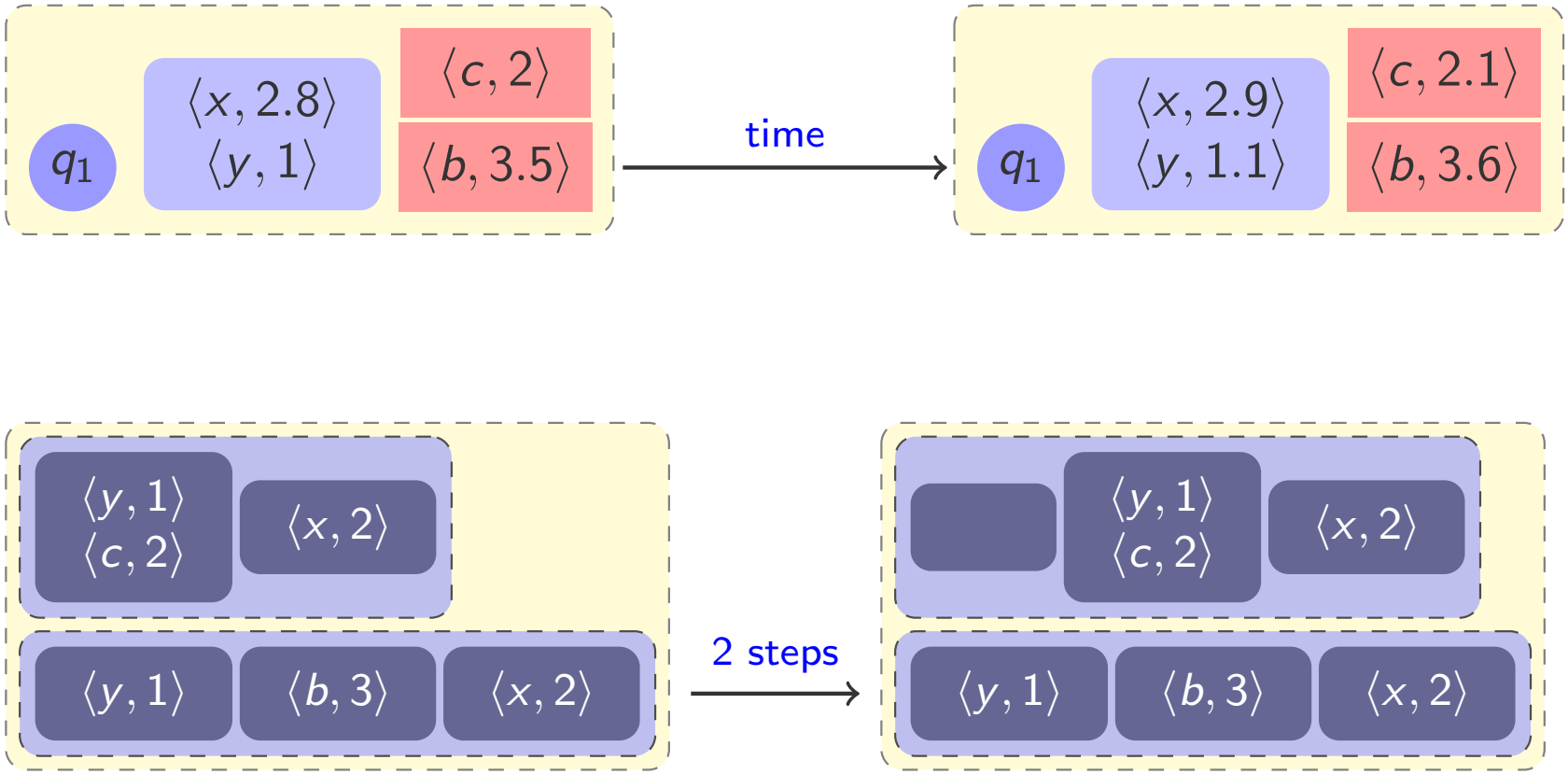
- identical integral values up to c_{max}
- fractional parts either both 0 or both positive
- identical ordering of fractional parts



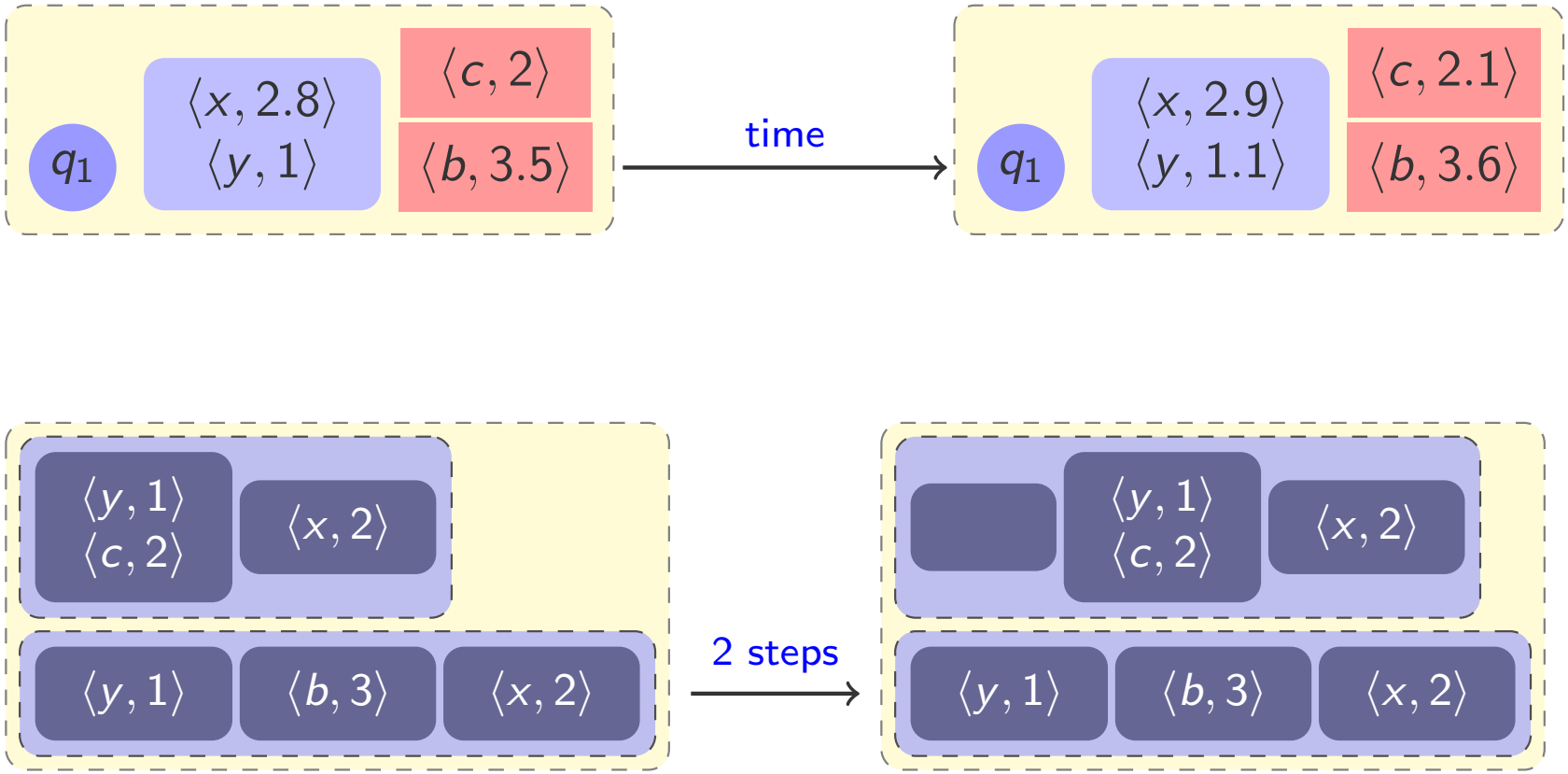
Simulating Time



Simulating a TPDA



Simulating a TPDA



Caveat

We need some additional information that relates regions.
Relation to preceeding region enough $\implies$ finite stack alphabet.

Outline

Priced Timed Pushdown Automata

Definition

A *priced timed pushdown automaton* is a tuple (Q, X, Γ, Δ), where

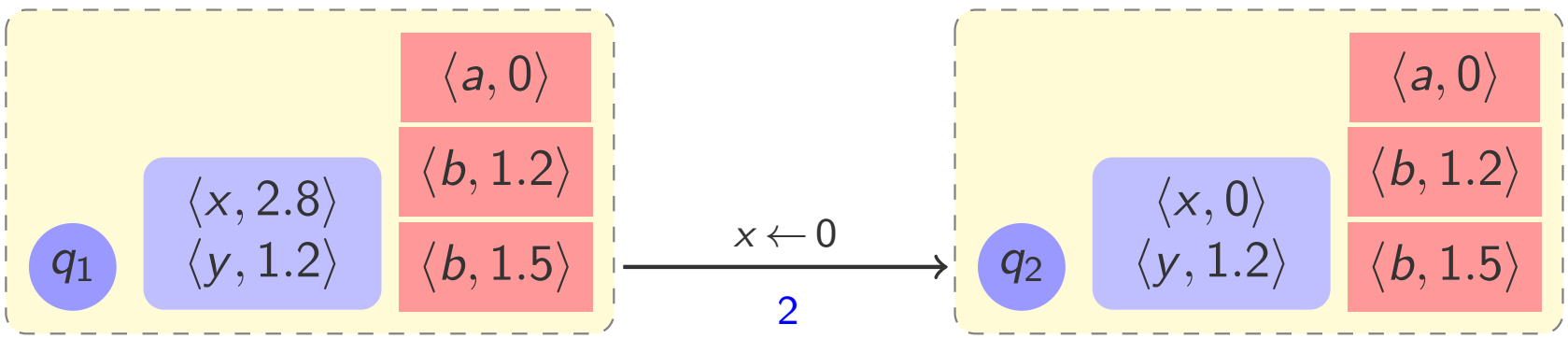
- Q finite set of states
- X finite set of clocks
- Γ finite stack alphabet
- Δ finite set of transition rules
- *Cost* A cost function $\Delta \cup \Gamma \to \mathbb{N}$

Priced Timed Pushdown Automata

Definition

A *priced timed pushdown automaton* is a tuple (Q, X, Γ, Δ), where

- Q finite set of states
- X finite set of clocks
- Γ finite stack alphabet
- Δ finite set of transition rules
- *Cost* A cost function $\Delta \cup \Gamma \to \mathbb{N}$

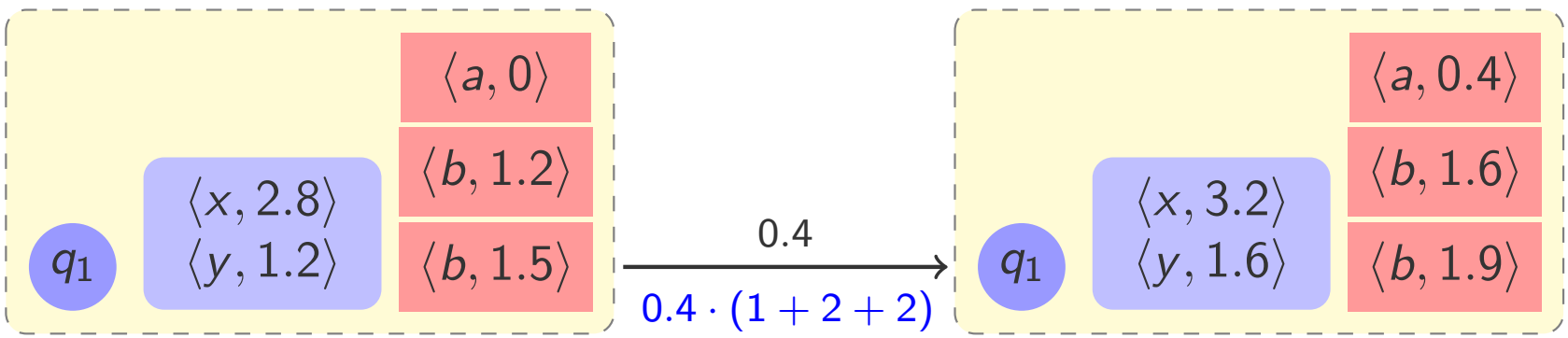


Priced Timed Pushdown Automata

Definition

A *priced timed pushdown automaton* is a tuple (Q, X, Γ, Δ), where

- Q finite set of states
- X finite set of clocks
- Γ finite stack alphabet
- Δ finite set of transition rules
- *Cost* A cost function $\Delta \cup \Gamma \to \mathbb{N}$



Optimal-Cost Reachability

Definition

$Cost_{opt}(T, q)$ is the infimum of the costs of all computations to the state q in T (∞ if q not reachable).

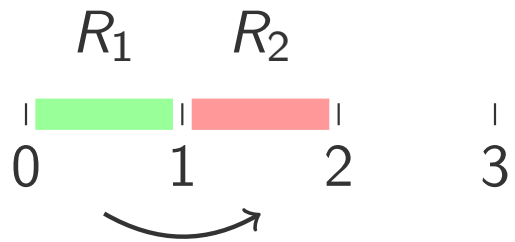
Theorem

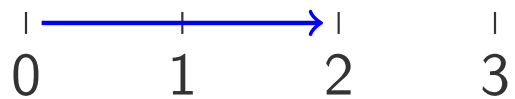
$Cost_{opt}(T, q)$ is computable.

Computing the Optimal Cost

Idea

Translate TPDA T to PDA P, add costs to transitions of P.





Outline

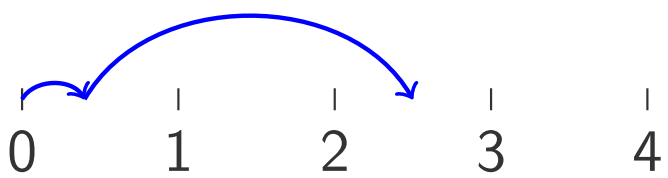
Form of Computations

Definition

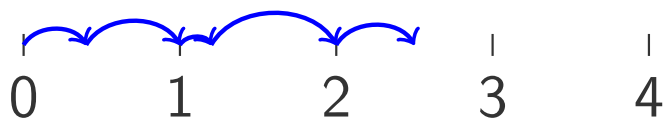
A computation is *detailed* if each timed transition either:

- "Pushes out" items with fractional part 0 without affecting any integral parts.
- Makes at most the items with highest fractional part become integral.

Not Detailed



Detailed



Form of Computations

Definition

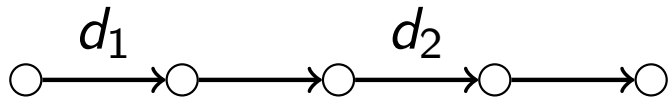
A computation is in δ-*form* for a $\delta : 0 < \delta < \frac{1}{10}$ if all values along the computations are within δ of some integer.

Definition

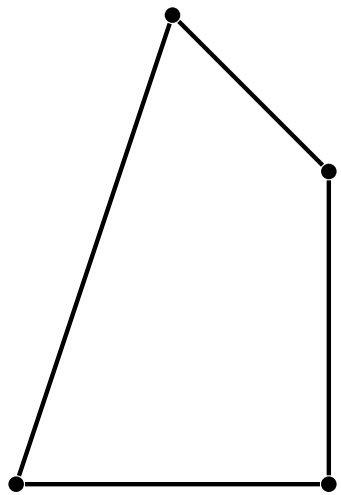
A detailed computation in δ-form consists of *long* (length in $(1 - \delta : 1)$) and *short* (length in $(0 : \delta)$) timed transitions.

Finding the Optimum

Adapt previous technique (Boyer et al, 2007, Abdulla & Mayr, 2011):
Fix the "structure" of a computation and vary the delays.

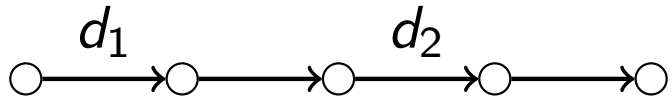


Feasible delays give rise to polyhedron with integer vertices.

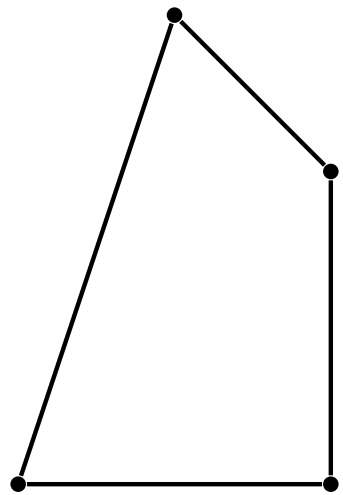


Finding the Optimum

Adapt previous technique (Boyer et al, 2007, Abdulla & Mayr, 2011):
Fix the "structure" of a computation and vary the delays.



Feasible delays give rise to polyhedron with integer vertices.



Cost of computation linear over polyhedron

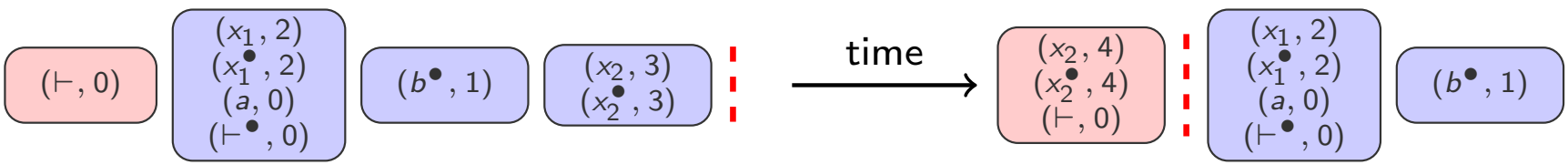
Simulation of PTPDA

Build Priced PDA that simulates detailed PTPDA computations in δ-form for arbitrarily small δ.

Costs

Short timed transitions Pay nothing

Long timed transitions Pay full stack cost



Simulation of PTPDA

We have seen that optimal-cost reachability for PTPDA reduces to optimal-cost reachability for PPDA.

Lemma (LATA 2011)

Optimal-cost reachability for PPDA is computable.

Outline

Conclusions

- Priced extension of dense-timed pushdown automata.
- Optimal reachability cost is computable.

Conclusions

- Priced extension of dense-timed pushdown automata.
- Optimal reachability cost is computable.

Thank you for your attention!