Timed Push-Down Automata

Parosh Aziz Abdulla <parosh@it.uu.se>

Department of Information Technology
Uppsala University
Sweden

(Joint work with Mohamed Faouzi Atig and Jari Stenman)
Background

Finite-State Systems

Stochastic Games

Markov Decision Processes

• Classical Model Checking:
  • hardware verification

Infinite-State Systems

• Unbounded Data:
  • stacks, counters, queues, clocks, . . .

• Unbounded Control:
  • multi-threaded programs, parameterized systems, . . .

Finite-State + Quantities

• Time (Timed Automata)
• Probabilities (Markov Chains)
• Weights/Costs

Infinite-State + Quantities

Existing Work:

• Probabilistic/Weighted Pushdown Systems
• Timed/Probabilistic Petri Nets
• Weighted/Probabilistic Timed Automata

This Presentation:

• Timed Push-Down Systems
Background

- Classical Model Checking:
  - hardware verification

Finite-State Systems

Existing Work:
- Probabilistic/Weighted Pushdown Systems
- Timed/Probabilistic Petri Nets
- Weighted/Probabilistic Timed Automata

This Presentation:
- Timed Push-Down Systems
Background

Finite-State Systems

Infinite-State Systems

• Classical Model Checking:
  - hardware verification

• Unbounded Data:
  - stacks, counters, queues, clocks, . . .

• Unbounded Control:
  - multi-threaded programs, parameterized systems, . . .

Finite-State + Quantities

- Time (Timed Automata)
- Probabilities (Markov Chains)
- Weights/Costs

Infinite-State + Quantities

- Probabilistic/Weighted Pushdown Systems
- Timed/Probabilistic Petri Nets
- Weighted/Probabilistic Timed Automata

Existing Work:

- Probabilistic/Weighted Pushdown Systems
- Timed/Probabilistic Petri Nets
- Weighted/Probabilistic Timed Automata

This Presentation:

- Timed Push-Down Systems

Timed Push-Down Automata
Background

• Classical Model Checking:
  • hardware verification

• Infinite-State Systems
  • Unbounded Data:
    • stacks, counters, queues, clocks, . . .
  • Unbounded Control:
    • multi-threaded programs, parameterized systems, . . .

Finite-State Systems

Infinite-State Systems

Existing Work:
• Probabilistic/Weighted Pushdown Systems
• Timed/Probabilistic Petri Nets
• Weighted/Probabilistic Timed Automata

This Presentation:
• Timed Push-Down Systems

Timed Push-Down Automata
Background

Finite-State + Quantities

Finite-State Systems

Infinite-State Systems

Existing Work:
- Probabilistic/Weighted Pushdown Systems
- Timed/Probabilistic Petri Nets
- Weighted/Probabilistic Timed Automata

This Presentation:
- Timed Push-Down Systems

Stochastic Games
Markov Decision Processes

Finite-State Systems

Infinite-State Systems

• Classical Model Checking:
  - hardware verification
• Unbounded Data:
  - stacks, counters, queues, clocks, ...
• Unbounded Control:
  - multi-threaded programs, parameterized systems, ...

Finite-State + Quantities

- Time (Timed Automata)
- Probabilities (Markov Chains)
- Weights/Costs

Infinite-State + Quantities

Timed Push-Down Automata
Background

Finite-State + Quantities

• Time (Timed Automata)
• Probabilities (Markov Chains)
• Weights/Costs

Finite-State Systems

Infinite-State Systems
Background

Finite-State + Quantities

Infinite-State + Quantities

Finite-State Systems

Infinite-State Systems

Existing Work:

- Probabilistic/Weighted Pushdown Systems
- Timed/Probabilistic Petri Nets
- Weighted/Probabilistic Timed Automata

This Presentation:

- Timed Push-Down Systems
Background

Existing Work:
- Probabilistic/Weighted Pushdown Systems
- Timed/Probabilistic Petri Nets
- Weighted/Probabilistic Timed Automata

Finite-State + Quantities

Infinite-State + Quantities

Finite-State Systems

Infinite-State Systems

This Presentation:
- Timed Push-Down Systems
Background

This Presentation:
- Timed Push-Down Systems

Finite-State + Quantities → Infinite-State + Quantities

Finite-State Systems → Infinite-State Systems

Finite-State Systems

Infinite-State Systems

Existing Work:
- Probabilistic/Weighted Pushdown Systems
- Timed/Probabilistic Petri Nets
- Weighted/Probabilistic Timed Automata

This Presentation:
- Timed Push-Down Systems
Background

Markov Decision Processes

Finite-State + Quantities

Infinite-State + Quantities

Finite-State Systems

Infinite-State Systems

Existing Work:
- Probabilistic/Weighted Pushdown Systems
- Timed/Probabilistic Petri Nets
- Weighted/Probabilistic Timed Automata

This Presentation:
- Timed Push-Down Systems

Timed Push-Down Automata
Background

Stochastic Games

Markov Decision Processes

Finite-State + Quantities

Finite-State Systems

Infinite-State + Quantities

Infinite-State Systems

Existing Work:
• Probabilistic/Weighted Pushdown Systems
• Timed/Probabilistic Petri Nets
• Weighted/Probabilistic Timed Automata

This Presentation:
• Timed Push-Down Systems

Timed Push-Down Automata
Related Models
Related Models

- Timed Automata
- Priced Timed Automata
- Dense-Timed Push-Down Automata
- Priced Dense-Timed Push-Down Automata
- Push-Down Automata
- Priced Discrete-Timed Push-Down Automata
Definition

A pushdown automaton is a tuple $(Q, \Gamma, \Delta)$, where

- $Q$ finite set of states
- $\Gamma$ finite stack alphabet
- $\Delta$ finite set of transition rules

Configurations consist of:

- A state $q \in Q$
- A word (stack content) $w$ over $\Gamma$
Related Models

- Timed Automata
- Priced Timed Automata
- Dense-Timed Push-Down Automata
- Priced Dense-Timed Push-Down Automata
- Push-Down Automata
- Priced Discrete-Timed Push-Down Automata
Pushdown Automata

Definition

A pushdown automaton is a tuple \((Q, \Gamma, \Delta)\), where

- \(Q\) finite set of states
- \(\Gamma\) finite stack alphabet
- \(\Delta\) finite set of transition rules

Configurations consist of:

- A state \(q \in Q\)
- A word (stack content) \(w\) over \(\Gamma\)
Pushdown Automata

1. \( q_1 \xrightarrow{nop} q_2 \)
2. \( q_1 \xrightarrow{push(b)} q_2 \)
3. \( q_1 \xrightarrow{pop(b)} q_2 \)
Extending Pushdown Automata

Model

- extend PDA by adding:
  - clocks
  - ages to the stack symbols
- make decisions based on:
  - clock values
  - ages of stack symbols

Challenge

- One clock per stack symbol
- Unbounded number of clocks
TPDA Configurations

TPDA configuration =
- PDA configuration
- clock valuation $\nu : X \rightarrow \mathbb{R}^\geq 0$
- ages of stack symbols

PDA Configuration  TPDA Configuration

\begin{align*}
q & \quad \begin{array}{c} c \\ b \\ a \end{array} \\
& \quad q \begin{array}{c} x_1 = 1.2 \\
& \quad \begin{array}{c} x_2 = 2 \\
& \quad \begin{array}{c} (c, 0.3) \\
& \quad \begin{array}{c} (b, 0.5) \\
& \quad \begin{array}{c} (a, 1.6) 
\end{array}
\end{array}
\end{array}
\end{array}
\end{align*}
TPDA Configurations

TPDA configuration =

- PDA configuration
- clock valuation $\nu : X \rightarrow \mathbb{R}_{\geq 0}$
- ages of stack symbols

#### PDA Configuration

```
q
  a  b  c
```

#### TPDA Configuration

```
q
  x_1 = 1.2
  x_2 = 2
  (a, 1.6)
  (b, 0.5)
  (c, 0.3)
```
TPDA Configurations

TPDA configuration =
- PDA configuration
- clock valuation $\nu : X \rightarrow \mathbb{R}_{\geq 0}$
- ages of stack symbols

PDA Configuration  TPDA Configuration

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>1.2</td>
</tr>
<tr>
<td>$b$</td>
<td>0.5</td>
</tr>
<tr>
<td>$a$</td>
<td>1.6</td>
</tr>
</tbody>
</table>

$q_0 a b c q_0 x_1 = 1.2 x_2 = 2 $(b, 0.5) (a, 1.6)
TPDA Transitions

- **nop** transition:
  - $q_1$: $x_1 = 1.2$, $x_2 = 2$
  - $q_2$: $x_1 = 1.2$, $x_2 = 2$

- **push** transition:
  - $q_1$: $x_1 = 1.2$, $x_2 = 2$
  - $q_2$: $x_1 = 1.2$, $x_2 = 2$

- **pop** transition:
  - $q_1$: $x_1 = 1.2$, $x_2 = 2$
  - $q_2$: $x_1 = 1.2$, $x_2 = 2$
TPDA Transitions

$q_1 \ x_1 = 1.2 \ x_2 = 2 \ (a, 2.3) \ x_1 \in [1..2]?$

$q_2 \ x_1 = 1.2 \ x_2 = 2 \ (a, 2.3)$

$q_1 \ x_1 = 1.2 \ x_2 = 2 \ (a, 2.3) \ x_1 \leftarrow [0..0]$

$q_2 \ x_1 = 0 \ x_2 = 2 \ (a, 2.3)$

$q_1 \ x_1 = 0 \ x_2 = 2 \ (a, 2.3) \ time$

$q_1 \ x_1 = 0.4 \ x_2 = 2.4 \ (a, 2.7)$
Timed Push-Down Automata

TPDA Computation

$q_1 \quad x_1 = 0 \quad x_2 = 0 \quad (a, 0) \xrightarrow{time} \quad q_1 \quad x_1 = 1.32 \quad x_2 = 1.32 \quad (a, 2.3)$
TPDA Computation

\[
q_2 \quad \begin{align*}
  x_1 &= 1.32 \\
  x_2 &= 1.32
\end{align*} \quad (a, 0) \quad \xrightarrow{\text{push}(a, (0 : 1))} \quad q_3 \quad \begin{align*}
  x_1 &= 1.32 \\
  x_2 &= 1.32
\end{align*} \quad (a, 0.9)
\]
TPDA Computation

$q_3 \overset{x_1 = 1.32}{\rightarrow} (a, 0.9) \overset{\text{time}}{\rightarrow} q_3 \overset{x_1 = 1.62}{\rightarrow} (a, 1.2)$
TPDA Computation

\[ q_3 \xrightarrow{x_1 = 1.62, x_2 = 1.62} (a, 1.2) \]

\[ push(b, (1 : 5]) \]

\[ q_4 \xrightarrow{x_1 = 1.62, x_2 = 1.62} (b, 5.0) \]

\[ (a, 1.2) \]
TPDA Computation

\[ q_4 \quad x_1 = 1.62 \quad x_2 = 1.62 \quad (b, 5.0) \]

\[ (a, 1.2) \]

\[ \text{time} \]

\[ q_4 \quad x_1 = 2.0 \quad x_2 = 2.0 \quad (b, 5.38) \]

\[ (a, 1.58) \]
TPDA Computation

$q_4$: $x_1 = 2.0$, $x_2 = 2.0$ (a, 1.58)

$q_5$: $x_1 = 0.1$, $x_2 = 2.0$ (a, 1.58)

$x_1 \leftarrow (0 : 1)$
TPDA Computation

$q_5$: $x_1 = 0.1$, $x_2 = 2.0$  
$(a, 1.58)$  

$(b, 5.38)$  

$q_6$: $x_1 = 0.1$, $x_2 = 2.0$  
$(a, 1.58)$  

$test(x_2 \in (1:2])$
TPDA Computation

\[ q_6 \xrightarrow{\text{push}(c, [0 : 0])} q_7 \]

- \( q_6 \):
  - \( x_1 = 0.1 \)
  - \( x_2 = 2.0 \)
  - \( (a, 1.58) \)
  - \( (b, 5.38) \)

- \( q_7 \):
  - \( x_1 = 0.1 \)
  - \( x_2 = 2.0 \)
  - \( (a, 1.58) \)
  - \( (b, 5.38) \)
  - \( (c, 0) \)
Timed Push-Down Automata

TPDA Computation

\[
q_7 \quad x_1 = 0.1 \\
(0, 1.58)
\]

\[
q_7 \quad x_2 = 2.0 \\
(b, 5.38)
\]

\[
q_7 \quad x_1 = 0.12 \\
(a, 1.6)
\]

\[
q_7 \quad x_2 = 2.02 \\
(b, 5.4)
\]
TPDA Computation

\[ q_7 \quad x_1 = 0.12 \quad x_2 = 2.02 \]

\[ (c, 0) \quad (b, 5.4) \quad (a, 1.6) \]

\[ pop(c, [0 : 1]) \]

\[ q_8 \quad x_1 = 0.12 \quad x_2 = 2.02 \]

\[ (b, 5.4) \quad (a, 1.6) \]
TPDA Computation

$q_8 \xrightarrow{x_2 \leftarrow [0:0]} q_9$

$x_1 = 0.12$
$x_2 = 2.02$
$(a, 1.6)$

$(b, 5.4)$

$x_1 = 0.12$
$x_2 = 0$
$(a, 1.6)$

$(b, 5.4)$
TPDA Computation

$q_9 \xrightarrow{x_1 = 0.12, x_2 = 0} (b, 5.4) \xrightarrow{pop(b, [0: \infty))} q_{10} \xrightarrow{x_1 = 0.12, x_2 = 0} (a, 1.6)$
Reachability

Definition (The Reachability Problem)

Given:
- TPDA $P$
- Initial configuration $c_0$
- Target state $q_{target}$

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

$c_0$
Main Result

Reachability Problem for TPDA is decidable

- Reduction from reachability problem for TPDA to reachability problem for PDA
- Simulate TPDA with PDA

Challenge:

- All components need to be finite, despite
  - Continuous time
  - Unboundedly many clocks

Method:

- Symbolic representation: Regions
- Simulation of TPDA by PDA
Regions

Definition

A region is a word over $2^\Gamma \times [0..\text{max}]$.

$\frac{frac}{= 0}$

Ordering of fractional parts

- [Region] = clock valuations satisfying it
- Finitely many regions
Resetting Clock Values:

\[ \text{frac} = 0 \]

\[ \begin{align*}
(a, 1) & \quad (b, 2) \\
(d, 0) & \\
\end{align*} \]

\[ \text{Reset}(b) \]

\[ \begin{align*}
(b, 0) & \quad (a, 1) \\
(d, 0) & \\
\end{align*} \]
Passage of time: Rotation

Time

(a, 1)  (b, 2)
(d, 0)
Regions

Passage of time: Rotation

(a, 1)  (b, 2)  (d, 0)
(b, 3)  (d, 1)  (a, 1)
Passage of time: Rotation
Passage of time: Rotation

Time

\begin{align*}
(a, 1) & \quad (b, 2) \\
(b, 3) & \quad (d, 1) \\
(a, 2) & \quad (b, 3) \\
\end{align*}
Simulating a TPDA

Main Ideas:

- extend regions to TPDA
- store regions in the stack
- relate each stack symbol to global clocks
Simulating a TPDA

TPDA Configuration

PDA Configuration

$q \quad x_1 = 1.2 \quad x_2 = 0.3 \quad (a, 1.6)$

$q \quad (x_1, 1) \quad (x_2, 0) \quad (a, 1)$
Simulating Push

\[ x_1 = 1.2 \]
\[ x_2 = 0.3 \]
\[ (a, 1.6) \]

\[ push(b, [0..1]) \]

\[ x_1 = 1.2 \]
\[ x_2 = 0.3 \]
\[ (a, 1.6) \]

\[ (b, 0.1) \]

\[ push(b, [0..1]) \]

\[ (x_1, 0) \]
\[ (x_2, 0) \]
\[ (a, 1) \]

\[ (b, 0) \]
\[ (x_1, 1) \]
\[ (x_2, 0) \]

\[ (x_1, 1) \]
\[ (x_2, 0) \]
\[ (a, 1) \]
Simulating Pop

Five operations are simple:
- **Nop** and **Test** do not modify anything
- **Push** creates new topmost region
- **Reset** and **Time** modify topmost region

The difficult operation is **pop**:

Which new topmost region when popping \( b \)?

\[
\begin{align*}
& (x, 0) \quad (a, 2) \quad \text{OR} \quad (x, 0) \quad (a, 2) \quad \text{OR} \quad (a, 2) \quad (x, 0)
\end{align*}
\]
Simulating Pop

\( \text{push}(b, [0..1]) \)

\( x \leftarrow [0..0] \)

\( \text{time} \)
Simulating Pop

\[ \text{push}(b, [0..1]) \]

\[ x \leftarrow [0..0] \]

\[ \text{time} \]
Simulating Pop

Lost information: \( \text{frac}(x) \neq \text{frac}(a) \)
Simulating a TPDA

- In general: relate clocks in topmost region with symbols that lie arbitrarily deep in the stack.

- Can we do this in a finite way?
Shadow Items

A little bit of information about previous region.

Regions contain

- Plain items: $X \cup \Gamma \cup \{\vdash\}$
- Shadow items: $X^\bullet \cup \Gamma^\bullet \cup \{\vdash^\bullet\}$

$\vdash$ is a reference clock which is (almost) always 0
Simulating Pop with Shadow Items

\[ \begin{align*} & (\cdot, 0) \quad (\cdot, 0) \quad (a, 2) \quad (c, 0) \quad (x, 1) \quad (x, 1) \\ & (\cdot, 0) \quad (\cdot, 0) \quad (a, 2) \quad (c, 0) \quad (x, 1) \quad (x, 1) \end{align*} \]

\[ \begin{align*} & (b, 0) \quad (\cdot, 0) \quad (a, 2) \quad (c, 0) \quad (x, 1) \quad (x, 1) \\ & (\cdot, 0) \quad (\cdot, 0) \quad (a, 2) \quad (c, 0) \quad (x, 1) \quad (x, 1) \end{align*} \]

\[ \begin{align*} & (b, 0) \quad (\cdot, 0) \quad (a, 2) \quad (c, 0) \quad (x, 1) \quad (x, 1) \\ & (\cdot, 0) \quad (\cdot, 0) \quad (a, 2) \quad (c, 0) \quad (x, 1) \quad (x, 1) \end{align*} \]

\[ \begin{align*} & (b, 0) \quad (\cdot, 0) \quad (a, 2) \quad (c, 0) \quad (x, 1) \quad (x, 1) \\ & (\cdot, 0) \quad (\cdot, 0) \quad (a, 2) \quad (c, 0) \quad (x, 1) \quad (x, 1) \end{align*} \]

---

Simulating Pop with Shadow Items

- \( \text{push}(b, [0..1]) \)

- \( x \leftarrow [0..0] \)

- \( \text{time} \)
Simulating Pop with Shadow Items

- Rotate lower region until matching

![Diagram showing simulation process]
Simulating Pop with Shadow Items

- Rotate lower region until **matching**
- **Plain stack symbol** taken from lower region

![Diagram of simulation process]

((⊢, 0), (⊢•, 0), (a•, 2), (x•, 1))

((⊢, 0), (a•, 2), (x•, 1))

((怛, 0), (b, 0), (x, 0), (怛•, 0))

((怛, 0), (怛•, 0), (a•, 2), (x•, 1))

((怛, 0), (怛•, 0), (a•, 2), (x•, 1))

((怛, 0), (怛•, 0), (a•, 2), (x•, 1))
Simulating Pop with Shadow Items

- Rotate lower region until matching
- Plain stack symbol taken from lower region
- Plain clock symbols taken from upper region
Simulating Pop with Shadow Items

- **Rotate lower region until** matching
- **Plain stack symbol** taken from lower region
- **Plain clock symbols** taken from upper region
- **Shadow items** taken from lower region
Simulating Pop with Shadow Items

- Rotate lower region until matching
- Plain stack symbol taken from lower region
- Plain clock symbols taken from upper region
- Shadow items taken from lower region

Merge:
Example Simulation

\[
q_1 \xrightarrow{push(a, [0..1])} q_2
\]

Timed Push-Down Automata
Example Simulation

Simulation

Example Simulation

Timed Push-Down Automata

48 / 64
Example Simulation

test(x ∈ [0 : 1])
Example Simulation

Simulation

reset\((y, [0 : 0])\)

\(q_3\)
\[
\begin{align*}
x &= 0.2 \\
y &= 0.2 \\
(a, 0.5)
\end{align*}
\]

\(q_3\)
\[
\begin{align*}
x &= 0.2 \\
y &= 0 \\
(a, 0.5)
\end{align*}
\]

Timed Push-Down Automata
Example Simulation

\[ q_3 \quad x = 0.2 \quad y = 0 \quad (a, 0.5) \quad \text{push}(b, (0 : 1)) \quad q_3 \quad x = 0.2 \quad y = 0 \quad (a, 0.5) \]
Example Simulation

Simulation Example

\( q_3 \)

\[
\begin{align*}
x &= 0.2 \\
y &= 0 \\
(a, 0.5)
\end{align*}
\]

\[
\begin{align*}
x &= 1.0 \\
y &= 0.8 \\
(a, 1.3)
\end{align*}
\]

time

\[
\begin{align*}
&\ (y, 0) \\
&\ (y\cdot, 0) \\
&\ (\lnot, 0) \\
&\ (\lnot, 0)
\end{align*}
\]

\[
\begin{align*}
&\ (x, 1) \\
&\ (x\cdot, 1) \\
&\ (\lnot, 0) \\
&\ (\lnot, 0)
\end{align*}
\]

\[
\begin{align*}
&\ (x, 0) \\
&\ (x\cdot, 0) \\
&\ (\lnot, 0) \\
&\ (\lnot, 0)
\end{align*}
\]

\[
\begin{align*}
&\ (x, 0) \\
&\ (x\cdot, 0) \\
&\ (\lnot, 0) \\
&\ (\lnot, 0)
\end{align*}
\]

\[
\begin{align*}
&\ (x, 0) \\
&\ (x\cdot, 0) \\
&\ (\lnot, 0) \\
&\ (\lnot, 0)
\end{align*}
\]

\[
\begin{align*}
&\ (x, 0) \\
&\ (x\cdot, 0) \\
&\ (\lnot, 0) \\
&\ (\lnot, 0)
\end{align*}
\]
Example Simulation

\[ q_3 \]
\[ x = 1.0 \]
\[ y = 0.8 \]
\[ (b, 1.1) \]
\[ (a, 1.3) \]

\[ \text{pop}(b, (1 : 2)) \]

\[ q_3 \]
\[ x = 1.0 \]
\[ y = 0.8 \]
\[ (a, 1.3) \]
Example Simulation

Simulation

\[ q_3 \]

\[ \begin{array}{c}
x = 1.0 \\
y = 0.8 \\
(a, 1.3)
\end{array} \]

\[ \text{pop}(b, (1 : 2)) \]

\[ q_3 \]

\[ \begin{array}{c}
x = 1.0 \\
y = 0.8 \\
(a, 1.3)
\end{array} \]
Example Simulation

Simulation

q3

x = 1.0
y = 0.8

(a, 1.3)

pop(b, (1 : 2))

q3

x = 1.0
y = 0.8

(a, 1.3)
Example Simulation

\[
\begin{align*}
q_3 & : x = 1.0 \quad y = 0.8 \\
(a, 1.3) & \\
\end{align*}
\]

\[
\begin{align*}
\text{time} & \quad \longrightarrow \\
q_3 & : x = 1.2 \quad y = 1.0 \\
(a, 1.5) & \\
\end{align*}
\]

\[
\begin{align*}
(x, 1) & \\
(x^*, 1) & \\
(y^*, 1) & \\
(\text{\textbullet}, 1) & \\
(\neg, 0) & \\
\end{align*}
\]

\[
\begin{align*}
(x, 0) & \\
(y, 0) & \\
(x^*, 0) & \\
(y^*, 0) & \\
(\text{\textbullet}, 0) & \\
(\neg, 0) & \\
\end{align*}
\]
Example Simulation

Simulation

$q_3 \rightarrow x = 1.0 \quad y = 0.8 \quad (a, 1.3)$

$pop(a, (1:2))$

$q_4 \rightarrow x = 1.2 \quad y = 1.0 \quad (a, 1.5)$
Example Simulation

Simulation

$\text{Example Simulation}$

$\begin{align*}
q_3 & \quad x = 1.0 \\
& \quad y = 0.8 \\
& \quad (a, 1.3) \\
\end{align*}$

$\text{pop}(a, (1 : 2))$

$\begin{align*}
q_4 & \quad x = 1.2 \\
& \quad y = 1.0 \\
& \quad (a, 1.5) \\
\end{align*}$
Example Simulation

$q_3: x = 1.0, y = 0.8, (a, 1.3) \rightarrow pop(a, (1 : 2)) \rightarrow q_4: x = 1.2, y = 1.0, (a, 1.5)$

$(y, 1) (\vdash, 0) \rightarrow (x, 1) (\vdash, 1) (y, 1) (\vdash, 1) (\cdot, 1) (\vdash, 1) (\cdot, 1) (\vdash, 1) \rightarrow (y, 1) (\vdash, 0) (\vdash, 1) (\cdot, 1) (\cdot, 1) (\cdot, 1)$
Definition

A computation is *zeno* if it has infinitely many discrete transitions in finite time.
Detecting Zenoness in TPDA

**Mode 1:** No restrictions

**Mode 2:** $x_{control} < 1$
Detecting Zenoness in TPDA

Mode 1: No restrictions

Mode 2: $x_{control} < 1$

Theorem

Zenoness $\iff a^\omega \in Traces(PDA)$
Related Models

- Timed Automata
- Priced Timed Automata
- Dense-Timed Push-Down Automata
- Priced Dense-Timed Push-Down Automata
- Push-Down Automata
- Priced Discrete-Timed Push-Down Automata
Conclusions and Future Work

Conclusions
- Timed and cost extensions of push-down automata
- Reachability problem for TPDA is EXPTIME-complete

Future Work
- Priced Dense-Timed PDA