Recursive Timed Automata (for CS 713 -Special Topics in Automata and Logics)

Presentation · April 2019

1 author:

Meet Taraviya
Indian Institute of Technology Bombay

Recursive Timed Automata

CS 713 - Special Topics in Automata and Logics

Ashutosh Trivedi and Dominik Wojtczak

May 20, 2019

Outline

1. Recursive State Machines
   - Labelled Transition Systems
   - Games on Labelled Transition Systems
   - Recursive State Machines
   - Semantics of RSM
2. Recursive Timed Automata
   - Definition
   - RTA semantics
3. Undecidability of RTA Termination with ≥ 3 clocks
   - Two counter Minsky machines
   - Gadgets DB, HF, P3O, P2O, $\text{HALT}^{\mathcal{A}}$
4. Decidability of Reachability on Glitch-free RTA
   - Glitch-free RTA
   - Region Equivalence

Outline

1. Recursive State Machines
   - Labelled Transition Systems
   - Games on Labelled Transition Systems
   - Recursive State Machines
   - Semantics of RSM
2. Recursive Timed Automata
   - Definition
   - RTA semantics
3. Undecidability of RTA Termination with ≥ 3 clocks
   - Two counter Minsky machines
   - Gadgets DB, HF, P3O, P2O, HALT$^{\mathcal{A}}$
4. Decidability of Reachability on Glitch-free RTA
   - Glitch-free RTA
   - Region Equivalence

Labelled Transition Systems

A labelled transition system (LTS) is a tuple $\mathcal{L} = (S, A, X)$, where

- S is the set of states,
- A is the set of actions, and
- $X : S \times A \to S$ is the transition function.

Outline

1. Recursive State Machines
   - Labelled Transition Systems
   - Games on Labelled Transition Systems
   - Recursive State Machines
   - Semantics of RSM
2. Recursive Timed Automata
   - Definition
   - RTA semantics
3. Undecidability of RTA Termination with ≥ 3 clocks
   - Two counter Minsky machines
   - Gadgets DB, HF, P3O, P2O, HALT$^{\mathcal{A}}$
4. Decidability of Reachability on Glitch-free RTA
   - Glitch-free RTA
   - Region Equivalence

Games on Labelled Transition Systems

A game arena G is a tuple $(\mathcal{L}, S_{\text{Ach}}, S_{\text{Tor}})$, where

- $\mathcal{L} = (S, A, X)$ is an LTS,
- $S_{\mathrm{Ach}} \subseteq S$ is the set of states controlled by player Achilles, and
- $S_{\text{Tor}} \subseteq S$ is the set of states controlled by player Tortoise

A strategy of player Achilles is a partial function $\alpha : FRuns^{\mathcal{L}} \to A$ such that for a run $r \in FRuns^{\mathcal{L}}$ we have that $\alpha(r)$ is defined if $\text{last}(r) \in S_{\text{Ach}}$, and $\alpha(r) \in A(\text{last}(r))$ for every such r. A strategy of player Tortoise is defined analogously. For an initial state s and a set of final states F, the value $\text{Val}_F^{\mathcal{L}}(s)$ of the reachability game is defined the number of transitions that Tortoise can ensure before the game visits a state in F irrespective of the strategy of Achilles.
Achilles wins the reachability game if $\text{Val}_F^{\mathcal{C}}(s) < \infty$

Outline

1. Recursive State Machines
   - Labelled Transition Systems
   - Games on Labelled Transition Systems
   - Recursive State Machines
   - Semantics of RSM
2. Recursive Timed Automata
   - Definition
   - RTA semantics
3. Undecidability of RTA Termination with ≥ 3 clocks
   - Two counter Minsky machines
   - Gadgets DB, HF, P3O, P2O, HALT$^{\mathcal{A}}$
4. Decidability of Reachability on Glitch-free RTA
   - Glitch-free RTA
   - Region Equivalence

Recursive State Machines

A recursive state machine $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k)$ is a tuple of components, where for each $1 \leq i \leq k$ component $\mathcal{M}_i = (N_i, En_i, Ex_i, B_i, Y_i, A_i, X_i)$ consists of:

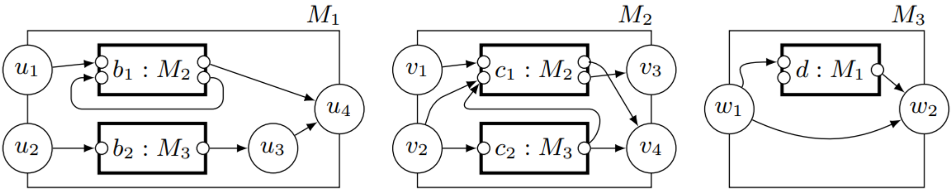
- a finite set N_i of nodes, including the set En_i entry nodes and the set Ex_i of exit nodes.
- a finite set B_i of boxes.
- boxes-to-components mapping $Y_i : B_i \to \{1, 2, \ldots, k\}$ that assigns every box to *a* component. To each box $b \in B_i$ we associate a set of call ports Call(b), and a set of return ports Ret(b) :

$$\text{Call}(b) = \left\{(b, en) : \text{en} \in En_{Y_i(b)}\right\}$$

$$\text{Ret}(b) = \left\{(b, ex) : \text{ex} \in Ex_{Y_i(b)}\right\}$$

  $Q_i = N_i \cup Call_i \cup \text{Ret}_i$
- a finite set A_i of actions.
- a transition function $X_i : Q_i \times A_i \to Q_i$ with a condition that call ports and exit nodes do not have any outgoing transitions.

Example



$$\mathcal{M}_i = (N_i, En_i, Ex_i, B_i, Y_i, A_i, X_i)$$

Outline

1. Recursive State Machines
   - Labelled Transition Systems
   - Games on Labelled Transition Systems
   - Recursive State Machines
   - Semantics of RSM
2. Recursive Timed Automata
   - Definition
   - RTA semantics
3. Undecidability of RTA Termination with ≥ 3 clocks
   - Two counter Minsky machines
   - Gadgets DB, HF, P3O, P2O, HALT$^{\mathcal{A}}$
4. Decidability of Reachability on Glitch-free RTA
   - Glitch-free RTA
   - Region Equivalence

Semantics of RSM

Let $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k)$ be an RSM where the component $\mathcal{M}_i$ is $(N_i, En_i, Ex_i, B_i, Y_i, A_i, X_i)$. The semantics of $\mathcal{M}$ is the countable labelled transition system$[\mathcal{M}] = (S_\mathcal{M}, A_\mathcal{M}, X_\mathcal{M})$ where:

- $S_\mathcal{M} \subseteq B^* \times Q$ is the set of states;
- $A_\mathcal{M} = \cup_{i=1}^{k} A_i$ is the set of actions;
- $X_\mathcal{M} : S_\mathcal{M} \times A_\mathcal{M} \to S_\mathcal{M}$ is the transition function such that for $s = (\langle \kappa \rangle, q) \in S_\mathcal{M}$ and $a \in A_\mathcal{M}$, we have that $s' = X_\mathcal{M}(s, a)$ if and only if one of the following holds:
  - the vertex q is a call port, i.e. $q = (b, en) \in Call$, and $s' = (\langle \kappa, b \rangle, en)$;
  - the vertex q is an exit node, i.e. $q = ex \in Ex$ and $s' = (\langle \kappa' \rangle, (b, ex))$ where $(b, ex) \in \mathrm{Ret}(b)$ and $\kappa = (\kappa', b)$;
  - the vertex q is any other kind of vertex, and $s' = (\langle \kappa \rangle, q')$ and $q' \in X(q, a)$.

Games on RSM: $([\mathcal{M}], [Q_{\mathrm{Ach}}]_\mathcal{M}, [\mathbb{Q}_{\mathrm{Tor}}]_\mathcal{M})$

Complexity results for Reachability in RSMs

# Players	1-box RSMs	1-exit RSMs	multi-exit RSMs
1	NLOGSPACE-complete [12]	PTIME-complete [3]	PTIME-complete [3]
2	PSPACE-complete [20,17]	PTIME-complete [22,15]	EXPTIME-complete [22]

Table 1. Complexity results for reachability objective for RSMs

Outline

1 Recursive State Machines
- Labelled Transition Systems
- Games on Labelled Transition Systems
- Recursive State Machines
- Semantics of RSM

2 Recursive Timed Automata
- Definition
- RTA semantics

3 Undecidability of RTA Termination with ≥ 3 clocks
- Two counter Minsky machines
- Gadgets DB, HF, P3O, P2O, HALT$^{\mathcal{A}}$

4 Decidability of Reachability on Glitch-free RTA
- Glitch-free RTA
- Region Equivalence

Recursive Timed Automata

A recursive timed automaton $\mathcal{T} = (\mathcal{C}, (\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k))$ is a pair made of a set of clocks $\mathcal{C}$ and a collection of components $(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k)$. Each component $\mathcal{T}_i = (N_i, En_i, Ex_i, B_i, Y_i, A_i, X_i, P_i, \mathrm{In}\, v_i, E_i, \rho_i)$ consists of:

- $N_i, En_i, Ex_i, B_i, Y_i, A_i, X_i$ as in RSM
- pass-by-value mapping $P_i : B_i \to 2^C$ that assigns every box the set of clocks that that are passed by value to the component mapped to the box; (The rest of the clocks are assumed to be passed by reference.)
- the invariant condition $Inv_i : Q_i \to \mathcal{Z}$
- the action enabledness function $E_i : Q_i \times A_i \to \mathcal{Z}$; and
- the clock reset function $\rho_i : A_i \to 2^{\mathcal{C}}$.

Example

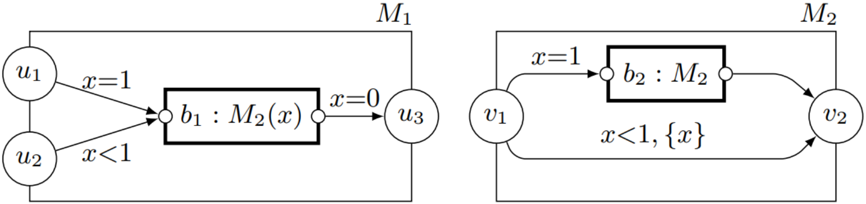


Fig. 2. Example recursive timed automaton

$$\mathcal{T}_i = (N_i, En_i, Ex_i, B_i, Y_i, A_i, X_i, P_i, \mathrm{In}\, v_i, E_i, \rho_i)$$

Outline

1. Recursive State Machines
   - Labelled Transition Systems
   - Games on Labelled Transition Systems
   - Recursive State Machines
   - Semantics of RSM
2. Recursive Timed Automata
   - Definition
   - RTA semantics
3. Undecidability of RTA Termination with ≥ 3 clocks
   - Two counter Minsky machines
   - Gadgets DB, HF, P3O, P2O, HALT$^{\mathcal{A}}$
4. Decidability of Reachability on Glitch-free RTA
   - Glitch-free RTA
   - Region Equivalence

Let $\mathcal{T} = (\mathcal{C}, (\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k))$ be an RTA where each component is of the form $\mathcal{T}_i = (N_i, En_i, Ex_i, B_i, Y_i, A_i, X_i, P_i, \mathrm{Inv}_i, E_i, \rho_i)$. The semantics of $\mathcal{T}$ is a labelled transition system $[\![\mathcal{T}]\!] = (S_\mathcal{T}, A_\mathcal{T}, X_\mathcal{T})$ where:

- $S_\mathcal{T} \subseteq (B \times V)^* \times Q \times V$, the set of states, is such that $(\langle\kappa\rangle, q, \nu) \in S_\mathcal{T}$ if $\nu \in \mathrm{In}\, v(q)$
- $A_\mathcal{T} = \mathbb{R}_\oplus \times A$ is the set of timed actions;
- $X_\mathcal{T} : S_\mathcal{T} \times A_\mathcal{T} \to S_\mathcal{T}$ is the transition function such that for $(\langle\kappa\rangle, q, \nu) \in S_\mathcal{T}$ and $(t, a) \in A_\mathcal{T}$, we have $(\langle\kappa'\rangle, q', \nu') = X_\mathcal{T}((\langle\kappa\rangle, q, \nu), (t, a))$ if and only if the following condition holds:
  - if the vertex q is a call port, i.e. $q = (b, en) \in \mathrm{Call}$ then $t = 0$, the context $\langle\kappa'\rangle = \langle\kappa, (b, \nu)\rangle$, $q' = en$, and $\nu' = \nu$
  - if the vertex q is an exit node, i.e. $q = ex \in Ex$, $\langle\kappa\rangle = \langle\kappa'', (b, \nu'')\rangle$ and let $(b, ex) \in \mathrm{Ret}(b)$, then $t = 0$; $\langle\kappa'\rangle = \langle\kappa''\rangle$; $q' = (b, ex)$; and $\nu' = \nu[P(b) := \nu'']$
  - if vertex q is any other kind of vertex, then $\nu + t' \in \mathrm{Inv}(q)$ for all $t' \in [0, t]$; $\nu + t \in E(q, a)$; and $\langle\kappa'\rangle = \langle\kappa\rangle$, $q' \in X(q, a)$, and $\nu' = (\nu + t)[\rho(a) := \mathbf{0}]$

Undecidability of RTA Termination with ≥ 3 clocks

Theorem

Termination problem is undecidable for recursive timed automata with at least three clocks. Moreover, termination game problem is undecidable for recursive timed automata with at least two clocks.

Outline

1. Recursive State Machines
   - Labelled Transition Systems
   - Games on Labelled Transition Systems
   - Recursive State Machines
   - Semantics of RSM
2. Recursive Timed Automata
   - Definition
   - RTA semantics
3. Undecidability of RTA Termination with ≥ 3 clocks
   - Two counter Minsky machines
   - Gadgets DB, HF, P3O, P2O, HALT$^{\mathcal{A}}$
4. Decidability of Reachability on Glitch-free RTA
   - Glitch-free RTA
   - Region Equivalence

Two counter Minsky machines

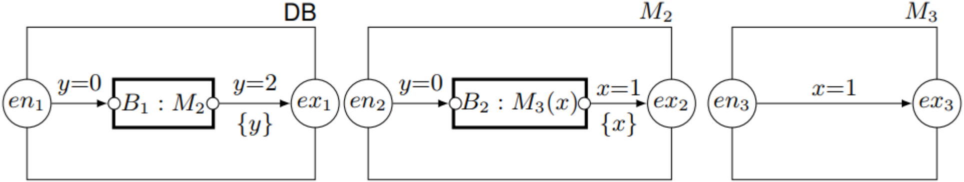
A Minsky machine $\mathcal{A}$ is a tuple (L, C, D) where: $L = \{\ell_0, \ell_1, \ldots, \ell_n\}$ is the set of states including the distinguished terminal state ℓ_n ; $C = \{c_1, c_2\}$ is the set of two counters; $D = \{\delta_0, \delta_1, \ldots, \delta_{n-1}\}$ is the set of transitions of the following type:

- (increment c)$\delta_i : c := c + 1;$ goto ℓ_k
- (test-and-decrement c)δ_i : if $(c > 0)$ then $(c := c - 1;$ goto ℓ_k else goto ℓ_m

where $c \in C, \delta_i \in D$ and $\ell_k, \ell_m \in L$.
Proof idea: A configuration (ℓ_i, c, d) of a Minsky machine corresponds to the configuration $(\langle \varepsilon \rangle, \ell_i, \nu)$ such that $\nu(x) = 2^{-c} \cdot 3^{-d}$ and $\nu(y) = 0$.

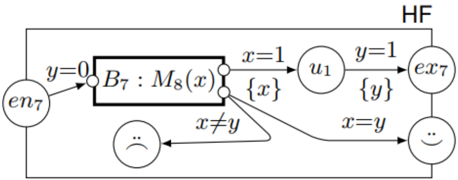
Outline

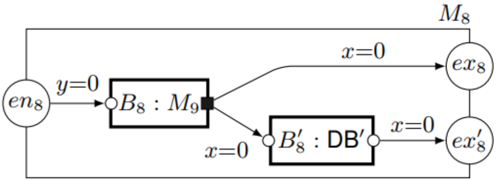
1. Recursive State Machines
   - Labelled Transition Systems
   - Games on Labelled Transition Systems
   - Recursive State Machines
   - Semantics of RSM
2. Recursive Timed Automata
   - Definition
   - RTA semantics
3. Undecidability of RTA Termination with ≥ 3 clocks
   - Two counter Minsky machines
   - Gadgets DB, HF, P3O, P2O, HALT$^{\mathcal{A}}$
4. Decidability of Reachability on Glitch-free RTA
   - Glitch-free RTA
   - Region Equivalence



$$
\begin{aligned}
& ((\langle\kappa\rangle, (b, en_1), (x_0, 0)) \rightsquigarrow ((\langle\kappa, b\rangle, en_1, (x_0, 0)) \\
\xrightarrow{y=0}_{0}\; & ((\langle\kappa, b\rangle, (B_1, en_2), (x_0, 0)) \rightsquigarrow ((\langle\kappa, b, B_1\rangle, en_2, (x_0, 0)) \\
\xrightarrow{y=0}_{0}\; & ((\langle\kappa, b, B_1\rangle, (B_2, en_3), (x_0, 0)) \rightsquigarrow ((\langle\kappa, b, B_1, B_2(x_0, 0)\rangle, en_3, (x_0, 0)) \\
\xrightarrow{x=1}_{(1-x_0)}\; & ((\langle\kappa, b, B_1, B_2(x_0, 0)\rangle, ex_3, (1, 1 - x_0)) \rightsquigarrow ((\langle\kappa, b, B_1\rangle, (B_2, ex_3), (x_0, 1 - x_0)) \\
\xrightarrow{x=1,\{x\}}_{(1-x_0)}\; & ((\langle\kappa, b, B_1\rangle, ex_2, (0, 2 - 2 \cdot x_0)) \rightsquigarrow ((\langle\kappa, b\rangle, (B_1, ex_2), (0, 2 - 2 \cdot x_0)) \\
\xrightarrow{y=2,\{y\}}_{(2\cdot x_0)}\; & ((\langle\kappa, b\rangle, ex_1, (2 \cdot x_0, 0)).
\end{aligned}
$$

HF



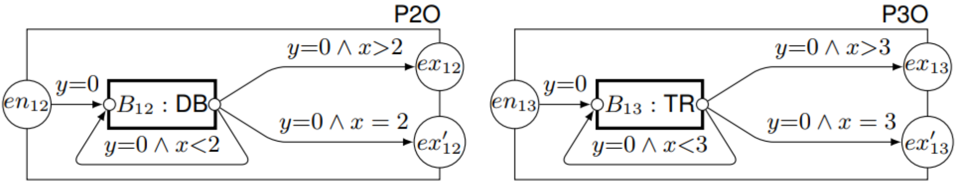


Proof. The main observation here is that, in component HF, starting from the configuration $(\langle\kappa\rangle, (b, en_7), (x_0, 0))$ Achilles has a strategy to terminate only if he chooses to delay the time by $\frac{x_0}{2}$ in component M_9 (called via box B_8). The evolution of the run from $(\langle\kappa\rangle, (b, en_7), (x_0, 0))$ to $(\langle\kappa, b, B_7(x_0, 0), B_8\rangle, en_9, (x_0, 0))$ is straightforward. Now, in component M_9 Achilles can wait for an arbitrary amount of time before taking a transition to ex_9 and resetting clock x. Let us assume that he waits for t time units, and hence $(\langle\kappa, b, B_7(x_0, 0)\rangle, (B_8, ex_9), (0, t))$ is reached which is controlled by Tortoise. Now Tortoise has a choice between making a transition to ex_8 (believing that $t = \frac{x_0}{2}$) or invoking the component B_8' (when suspecting that $t \neq \frac{x_0}{2}$).

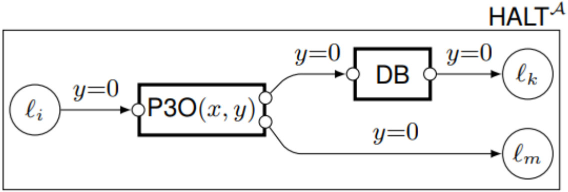
If Tortoise believes that $t = \frac{x_0}{2}$ then he makes a transition to ex_8 and thus the system reaches the configuration $(\langle\kappa, b\rangle, (B_7, ex_8), (x_0, t))$ giving rise to the following run:

$$(\langle\kappa, b\rangle, (B_7, ex_8), (x_0, t)) \xrightarrow{x=1,\{x\}}_{(1-x_0)} (\langle\kappa, b\rangle, u_1, (0, 1 - x_0 + t))$$
$$\xrightarrow{y=1,\{y\}}_{(x_0-t)} (\langle\kappa, b\rangle, ex_7, (x_0 - t, 0)) \rightsquigarrow (\langle\kappa\rangle, (b, ex_7), (x_0 - t, 0)).$$

P2O, P3O


P2O
en_{12}
$y{=}0$
B_{12} : DB
$y{=}0 \wedge x{<}2$
$y{=}0 \wedge x{>}2$
ex_{12}
$y{=}0 \wedge x = 2$
ex'_{12}
P3O
en_{13}
$y{=}0$
B_{13} : TR
$y{=}0 \wedge x{<}3$
$y{=}0 \wedge x{>}3$
ex_{13}
$y{=}0 \wedge x = 3$
ex'_{13}

HALT



Outline

1. Recursive State Machines
   - Labelled Transition Systems
   - Games on Labelled Transition Systems
   - Recursive State Machines
   - Semantics of RSM
2. Recursive Timed Automata
   - Definition
   - RTA semantics
3. Undecidability of RTA Termination with ≥ 3 clocks
   - Two counter Minsky machines
   - Gadgets DB, HF, P3O, P2O, HALT$^{\mathcal{A}}$
4. Decidability of Reachability on Glitch-free RTA
   - Glitch-free RTA
   - Region Equivalence

Glitch-free RTA

We say that a recursive timed automaton is glitch-free if for every box either all clocks are passed by value or none is passed by value, i.e. for each $b \in B$ we have that either $P(b) = \mathcal{C}$ or $P(b) = \emptyset$.
Any general recursive timed automaton with one clock is trivially glitch-free.

Outline

1. Recursive State Machines
   - Labelled Transition Systems
   - Games on Labelled Transition Systems
   - Recursive State Machines
   - Semantics of RSM
2. Recursive Timed Automata
   - Definition
   - RTA semantics
3. Undecidability of RTA Termination with ≥ 3 clocks
   - Two counter Minsky machines
   - Gadgets DB, HF, P3O, P2O, HALT$^{\mathcal{A}}$
4. Decidability of Reachability on Glitch-free RTA
   - Glitch-free RTA
   - Region Equivalence

Region Equivalence

For every $\mathrm{RTA}\mathcal{T}$ we define regional equivalence relation $\mathcal{E}_R \subseteq S_\mathcal{T} \times S_\mathcal{T}$. For configurations $s = (\langle\kappa\rangle, q, \nu)$ and $s' = (\langle\kappa'\rangle, q', \nu')$, $[s] = [s']$ if:

- $q = q', [\nu] = [\nu']$, and
- $\kappa = (b_1, \nu_1), (b_2, \nu_2), \ldots, (b_n, \nu_n)$ and $\kappa' = (b'_1, \nu'_1), (b'_2, \nu'_2), \ldots, (b'_n, \nu'_n)$ are such that for every $1 \leq i \leq n$ we have $[\nu_i] = [\nu'_i]$ and $b_i = b'_i$.

Bisimulation of Region Equivalence

Given: $s = (\langle\kappa\rangle, q, \nu)$ and $s' = (\langle\kappa'\rangle, q', \nu')$ such that $[s] = [s']$, timed action $(t, a) \in X_T$ such that $X_T(s, (t, a)) = s_a \, (= (\kappa_a, (q_a, \nu_a)))$
Find: (t', a) such that $X_T(s', (t', a)) = s'_a \, (= (\kappa'_a, (q'_a, \nu'_a)))$ and $[s_a] = [s'_a]$

- $q = (b, en) \in Call$: $t = 0$, $\langle\kappa_a\rangle = \langle\kappa, (b, \nu)\rangle, q_a = en$, and $\nu_a = \nu$. $q' = q$ is also a call port. So, $t' = 0$, and $\langle\kappa'_a\rangle = \langle\kappa', (b, \nu')\rangle, q'_a = en$, and $\nu'_a = \nu_a$. Trivially, $[s_a] = [s'_a]$.
- $q = ex \in Ex$: Let $\langle\kappa\rangle = \langle\kappa_*, (b, \nu_*)\rangle$. So, $t = 0$; context $\langle\kappa_a\rangle = \langle\kappa_*\rangle; q_a = (b, ex)$; and $\nu_a = \nu[P(b) := \nu_*]$. Let the context $\langle\kappa'\rangle$ be $\langle\kappa'_*, (b, \nu'_*)\rangle$. $q' = q$ is an exit node. So, $t' = 0, \langle\kappa'_a\rangle = \langle\kappa'_*\rangle$ and $\nu'_a = \nu'[P(b) := \nu'_*]$.
  - $P(b) = \mathcal{C}$: In this case $\nu_a = \nu_*$ and $\nu'_a = \nu'_*$, and since $[\nu_*] = [\nu'_*]$ we get that $[\nu_a] = [\nu'_a]$.
  - $P(b) = \emptyset$. In this case $\nu_a = \nu$ and $\nu'_a = \nu'$, and since $[\nu] = [\nu']$ we get that $[\nu_a] = [\nu'_a]$.
- For other q: Follows by classical region equivalence relation.

Decidability of Reachability on Glitch-free RTA

Theorem

Reachability (termination) problems and games on glitch-free RTA $\mathcal{T}$ can be reduced to solving reachability (termination) problems and games, respectively, on the corresponding region abstraction $\mathcal{T}^{\mathrm{RG}}$.

Complexity results for Reachability in Glitch-free RTAs

# Players	RTAs with 1 clock	RTAs with at least 2 clocks
1	PTIME-complete	EXPTIME-complete
2	EXPTIME-complete	2EXPTIME

Table 2. Complexity results for glitch-free RTAs

# Players	1-box RTAs with 1 global clock	1-box RTAs with at least 2 global clocks
1	PTIME-complete	PSPACE (PSPACE-complete for 3+ clocks)
2	PSPACE-complete	EXPSPACE (and EXPTIME-hard)

Table 3. Complexity results for 1-box RTAs with only global clocks

# Players	1-exit RTAs with 1 local clock	1-exit RTAs with at least 2 local clocks
1	PTIME-complete	EXPTIME-complete
2	PTIME-complete	EXPTIME-complete

Table 4. Complexity results for 1-exit RTAs with only local clocks