Determinization and Expressiveness of Integer Reset Timed Automata with Silent Transitions

P. Vijay Suman

(Joint work with Paritosh K. Pandya)

Tata Institute Of Fundamental Research, Mumbai

January, 2009

Workshop on Automata, Concurrency and Timed Systems

Outline

Timed Automata

- Timed Automata (TA): Finite State Automata + Clocks running at the rate of global time
- Extensions of timed automata with Periodic Guards [CG00] and Silent Transitions [BPGD98] have been introduced to model periodic behaviours
- Expressiveness: TA $<$ Per-TA $<$ ϵ-TA
- TA are closed under union and intersection but not closed under determinization and complementation [AD94]
- Though reachability is decidable for all the above classes, language inclusion is undecidable [AD94]

Timed Automata

- Timed Automata (TA): Finite State Automata + Clocks running at the rate of global time
- Extensions of timed automata with Periodic Guards [CG00] and Silent Transitions [BPGD98] have been introduced to model periodic behaviours
- Expressiveness: TA < Per-TA < ϵ-TA
- TA are closed under union and intersection but not closed under determinization and complementation [AD94]
- Though reachability is decidable for all the above classes, language inclusion is undecidable [AD94]

Timed Automata

- Timed Automata (TA): Finite State Automata + Clocks running at the rate of global time
- Extensions of timed automata with Periodic Guards [CG00] and Silent Transitions [BPGD98] have been introduced to model periodic behaviours
- Expressiveness: TA $<$ Per-TA $<$ ϵ-TA
- TA are closed under union and intersection but not closed under determinization and complementation [AD94]
- Though reachability is decidable for all the above classes, language inclusion is undecidable [AD94]

Timed Automata

- Timed Automata (TA): Finite State Automata + Clocks running at the rate of global time
- Extensions of timed automata with Periodic Guards [CG00] and Silent Transitions [BPGD98] have been introduced to model periodic behaviours
- Expressiveness: TA $<$ Per-TA $<$ ϵ-TA
- TA are closed under union and intersection but not closed under determinization and complementation [AD94]
- Though reachability is decidable for all the above classes, language inclusion is undecidable [AD94]

Timed Automata

- Timed Automata (TA): Finite State Automata + Clocks running at the rate of global time
- Extensions of timed automata with Periodic Guards [CG00] and Silent Transitions [BPGD98] have been introduced to model periodic behaviours
- Expressiveness: TA $<$ Per-TA $<$ ϵ-TA
- TA are closed under union and intersection but not closed under determinization and complementation [AD94]
- Though reachability is decidable for all the above classes, language inclusion is undecidable [AD94]

Integer Resets

- Integer Reset Timed Automata (IRTA): Clock resets happen at integral time points. Each resetting edge has an atomic constraint of the form $x = c$
  - Fractional values of all the clocks remain equal
  - Also equal to the fractional value of the global time
- Extensions of IRTA with periodic constraints and silent transitions can be defined. However clock resets happen at integral time points: In case of Per-IRTA, resetting edge consists an atomic constraint of the form $\exists k \in \mathbb{N} : x \in [a + kp, a + kp]$
- Notion of *global but sparse time* as used in Time Triggered Architecture and Distributed Business Processes can be naturally modeled as ϵ-IRTA [SPKM08]

Integer Resets

- Integer Reset Timed Automata (IRTA): Clock resets happen at integral time points. Each resetting edge has an atomic constraint of the form $x = c$
  - Fractional values of all the clocks remain equal
  - Also equal to the fractional value of the global time
- Extensions of IRTA with periodic constraints and silent transitions can be defined. However clock resets happen at integral time points: In case of Per-IRTA, resetting edge consists an atomic constraint of the form $\exists k \in \mathbb{N} : x \in [a + kp, a + kp]$
- Notion of *global but sparse time* as used in Time Triggered Architecture and Distributed Business Processes can be naturally modeled as ϵ-IRTA [SPKM08]

Integer Resets

- Integer Reset Timed Automata (IRTA): Clock resets happen at integral time points. Each resetting edge has an atomic constraint of the form $x = c$
  - Fractional values of all the clocks remain equal
  - Also equal to the fractional value of the global time
- Extensions of IRTA with periodic constraints and silent transitions can be defined. However clock resets happen at integral time points: In case of Per-IRTA, resetting edge consists an atomic constraint of the form $\exists k \in \mathbb{N} : x \in [a + kp, a + kp]$
- Notion of *global but sparse time* as used in Time Triggered Architecture and Distributed Business Processes can be naturally modeled as ϵ-IRTA [SPKM08]

Integer Resets

- Integer Reset Timed Automata (IRTA): Clock resets happen at integral time points. Each resetting edge has an atomic constraint of the form $x = c$
  - Fractional values of all the clocks remain equal
  - Also equal to the fractional value of the global time
- Extensions of IRTA with periodic constraints and silent transitions can be defined. However clock resets happen at integral time points: In case of Per-IRTA, resetting edge consists an atomic constraint of the form $\exists k \in \mathbb{N} : x \in [a + kp, a + kp]$
- Notion of *global but sparse time* as used in Time Triggered Architecture and Distributed Business Processes can be naturally modeled as ϵ-IRTA [SPKM08]

Integer Resets

- Integer Reset Timed Automata (IRTA): Clock resets happen at integral time points. Each resetting edge has an atomic constraint of the form $x = c$
  - Fractional values of all the clocks remain equal
  - Also equal to the fractional value of the global time
- Extensions of IRTA with periodic constraints and silent transitions can be defined. However clock resets happen at integral time points: In case of Per-IRTA, resetting edge consists an atomic constraint of the form $\exists k \in \mathbb{N} : x \in [a + kp, a + kp]$
- Notion of *global but sparse time* as used in Time Triggered Architecture and Distributed Business Processes can be naturally modeled as ϵ-IRTA [SPKM08]

Q & A

- Is language inclusion question $L(\mathcal{A}) \subseteq L(\mathcal{B})$ decidable? YES, if $\mathcal{A}$ is an ϵ-TA and $\mathcal{B}$ is an ϵ-IRTA
- Expressiveness: IRTA $<$ Per-IRTA $<$ ϵ-IRTA
- Can we get ϵ-free and deterministic representaions for ϵ-IRTA? We introduce a new variant of timed automata called *Generalized Reset Timed Automata (GRTA)*. Every ϵ-IRTA can be effectively translated into a 1-Clock Deterministic Per-GRTA
- Can ϵ-IRTA be effectively reduced to 1-Clock Deterministic ϵ-IRTA? YES

Q & A

- Is language inclusion question $L(\mathcal{A}) \subseteq L(\mathcal{B})$ decidable? YES, if $\mathcal{A}$ is an ϵ-TA and $\mathcal{B}$ is an ϵ-IRTA
- Expressiveness: IRTA $<$ Per-IRTA $<$ ϵ-IRTA
- Can we get ϵ-free and deterministic representaions for ϵ-IRTA? We introduce a new variant of timed automata called *Generalized Reset Timed Automata (GRTA)*. Every ϵ-IRTA can be effectively translated into a 1-Clock Deterministic Per-GRTA
- Can ϵ-IRTA be effectively reduced to 1-Clock Deterministic ϵ-IRTA? YES

Q & A

- Is language inclusion question $L(\mathcal{A}) \subseteq L(\mathcal{B})$ decidable? YES, if $\mathcal{A}$ is an ϵ-TA and $\mathcal{B}$ is an ϵ-IRTA
- Expressiveness: IRTA $<$ Per-IRTA $<$ ϵ-IRTA
- Can we get ϵ-free and deterministic representaions for ϵ-IRTA? We introduce a new variant of timed automata called *Generalized Reset Timed Automata (GRTA)*. Every ϵ-IRTA can be effectively translated into a 1-Clock Deterministic Per-GRTA
- Can ϵ-IRTA be effectively reduced to 1-Clock Deterministic ϵ-IRTA? YES

Q & A

- Is language inclusion question $L(\mathcal{A}) \subseteq L(\mathcal{B})$ decidable? YES, if $\mathcal{A}$ is an ϵ-TA and $\mathcal{B}$ is an ϵ-IRTA
- Expressiveness: IRTA $<$ Per-IRTA $<$ ϵ-IRTA
- Can we get ϵ-free and deterministic representaions for ϵ-IRTA? We introduce a new variant of timed automata called *Generalized Reset Timed Automata (GRTA)*. Every ϵ-IRTA can be effectively translated into a 1-Clock Deterministic Per-GRTA
- Can ϵ-IRTA be effectively reduced to 1-Clock Deterministic ϵ-IRTA? YES

Outline

$\delta\checkmark$-Representation of Timed Words

- Each integer time stamp i (except 0) is marked by a $\checkmark$. All the events occurring at time point i appear immediately after this $\checkmark$
- Entering an interval $(i, i+1)$ is marked by a δ. All the events occurring in the time interval $(i, i+1)$ appear immediately after this δ
- **The mapping** f: example timed words and their mappings:
  $\rho_1 = \langle (a, 1.2), (b, 3.5), (c, 4), (d, 4.5), (e, 4.5), (f, 5.6), (g, 5.8)) \rangle$
  $f(\rho_1) = \delta\checkmark\delta a\checkmark\delta\checkmark\delta b\checkmark c\delta d\ e\checkmark\delta fg$
  $\rho_2 = \langle (a, 0), (b, 0), (c, 0.5), (c, 0.6), (d, 2) \rangle$
  $\rho_3 = \langle (a, 0), (b, 0), (c, 0.3), (c, 0.7), (d, 2) \rangle$
  $f(\rho_2) = f(\rho_3) = ab\delta cc\checkmark\delta\checkmark\ d$

$\delta\checkmark$-Representation of Timed Words

- Each integer time stamp i (except 0) is marked by a $\checkmark$. All the events occurring at time point i appear immediately after this $\checkmark$
- Entering an interval $(i, i+1)$ is marked by a δ. All the events occurring in the time interval $(i, i+1)$ appear immediately after this δ
- **The mapping f**: example timed words and their mappings:
  $\rho_1 = \langle (a, 1.2), (b, 3.5), (c, 4), (d, 4.5), (e, 4.5), (f, 5.6), (g, 5.8) \rangle$
  $f(\rho_1) = \delta\checkmark\delta a\checkmark\delta\checkmark\delta b\checkmark c\delta d\ e\checkmark\delta fg$
  $\rho_2 = \langle (a, 0), (b, 0), (c, 0.5), (c, 0.6), (d, 2) \rangle$
  $\rho_3 = \langle (a, 0), (b, 0), (c, 0.3), (c, 0.7), (d, 2) \rangle$
  $f(\rho_2) = f(\rho_3) = ab\delta cc\checkmark\delta\checkmark d$

$\delta\checkmark$-Representation of Timed Words

- Each integer time stamp i (except 0) is marked by a $\checkmark$. All the events occurring at time point i appear immediately after this $\checkmark$
- Entering an interval $(i, i+1)$ is marked by a δ. All the events occurring in the time interval $(i, i+1)$ appear immediately after this δ
- **The mapping** f: example timed words and their mappings:
  $\rho_1 = \langle (a, 1.2), (b, 3.5), (c, 4), (d, 4.5), (e, 4.5), (f, 5.6), (g, 5.8) \rangle$
  $f(\rho_1) = \delta\checkmark\delta a\checkmark\delta\checkmark\delta b\checkmark c\delta d\ e\checkmark\delta fg$
  $\rho_2 = \langle (a, 0), (b, 0), (c, 0.5), (c, 0.6), (d, 2) \rangle$
  $\rho_3 = \langle (a, 0), (b, 0), (c, 0.3), (c, 0.7), (d, 2) \rangle$
  $f(\rho_2) = f(\rho_3) = ab\delta cc\checkmark\delta\checkmark d$

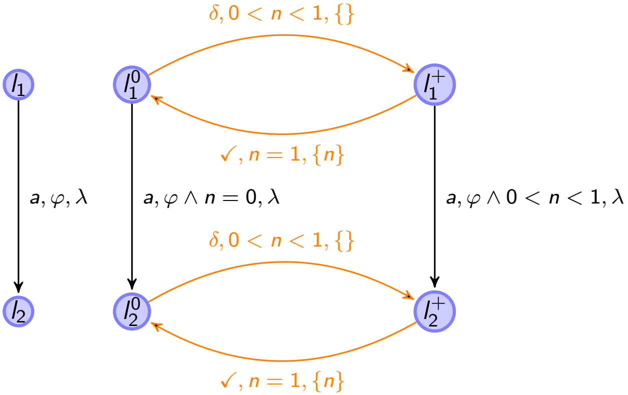
Construction

- For every ϵ-TA $\mathcal{A}$, one can effectively construct a finite state automaton $\mathcal{B}$ such that $L(\mathcal{B}) = f(L(\mathcal{A}))$.
- Apply following.

- Then apply region construction

Construction

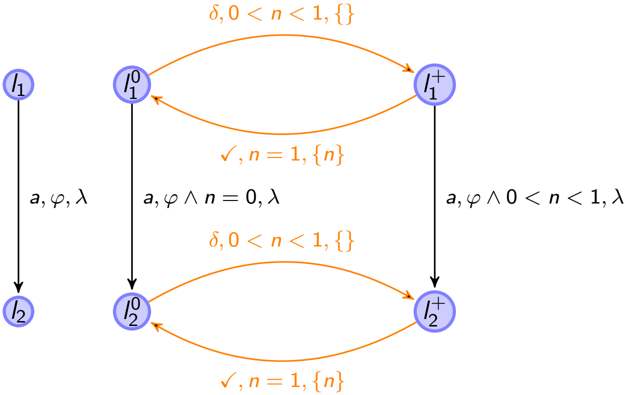
- For every ϵ-TA $\mathcal{A}$, one can effectively construct a finite state automaton $\mathcal{B}$ such that $L(\mathcal{B}) = f(L(\mathcal{A}))$.
- Apply following.



- Then apply region construction

Construction

- For every ϵ-TA $\mathcal{A}$, one can effectively construct a finite state automaton $\mathcal{B}$ such that $L(\mathcal{B}) = f(L(\mathcal{A}))$.
- Apply following.



- Then apply region construction

Key Properties of f

Theorem

Language inclusion problem $L(\mathcal{A}) \subseteq L(\mathcal{B})$?, where $\mathcal{A}$ is an ϵ-TA and $\mathcal{B}$ is an ϵ-IRTA, can be decided (EXPSPACE) in the $\delta\checkmark$-world [SPKM08]

- If $\mathcal{A}$ is an ϵ-IRTA and $f(\rho) = f(\rho')$ then $\rho \in L(\mathcal{A})$ iff $\rho' \in L(\mathcal{A})$ [SPKM08]
- If $\mathcal{A}$ is an...
  - ϵ-TA: $f^{-1}(f(L(\mathcal{A}))) \supseteq L(\mathcal{A})$
  - ϵ-IRTA: $f^{-1}(f(L(\mathcal{A}))) = L(\mathcal{A})$
- Therefore $L(\mathcal{A}) \subseteq L(\mathcal{B})$ iff $f(L(\mathcal{A})) \subseteq f(L(\mathcal{B}))$

Key Properties of f

Theorem

Language inclusion problem $L(\mathcal{A}) \subseteq L(\mathcal{B})$?, where $\mathcal{A}$ is an ϵ-TA and $\mathcal{B}$ is an ϵ-IRTA, can be decided (EXPSPACE) in the $\delta\checkmark$-world [SPKM08]

- If $\mathcal{A}$ is an ϵ-IRTA and $f(\rho) = f(\rho')$ then $\rho \in L(\mathcal{A})$ iff $\rho' \in L(\mathcal{A})$ [SPKM08]
- If $\mathcal{A}$ is an...
  - ϵ-TA: $f^{-1}(f(L(\mathcal{A}))) \supseteq L(\mathcal{A})$
  - ϵ-IRTA: $f^{-1}(f(L(\mathcal{A}))) = L(\mathcal{A})$
- Therefore $L(\mathcal{A}) \subseteq L(\mathcal{B})$ iff $f(L(\mathcal{A})) \subseteq f(L(\mathcal{B}))$

Key Properties of f

Theorem

Language inclusion problem $L(\mathcal{A}) \subseteq L(\mathcal{B})$?, where $\mathcal{A}$ is an ϵ-TA and $\mathcal{B}$ is an ϵ-IRTA, can be decided (EXPSPACE) in the $\delta\checkmark$-world [SPKM08]

- If $\mathcal{A}$ is an ϵ-IRTA and $f(\rho) = f(\rho')$ then $\rho \in L(\mathcal{A})$ iff $\rho' \in L(\mathcal{A})$ [SPKM08]
- If $\mathcal{A}$ is an...
  - ϵ-TA: $f^{-1}(f(L(\mathcal{A}))) \supseteq L(\mathcal{A})$
  - ϵ-IRTA: $f^{-1}(f(L(\mathcal{A}))) = L(\mathcal{A})$
- Therefore $L(\mathcal{A}) \subseteq L(\mathcal{B})$ iff $f(L(\mathcal{A})) \subseteq f(L(\mathcal{B}))$

Key Properties of f

Theorem

Language inclusion problem $L(\mathcal{A}) \subseteq L(\mathcal{B})$?, where $\mathcal{A}$ is an ϵ-TA and $\mathcal{B}$ is an ϵ-IRTA, can be decided (EXPSPACE) in the $\delta\checkmark$-world [SPKM08]

- If $\mathcal{A}$ is an ϵ-IRTA and $f(\rho) = f(\rho')$ then $\rho \in L(\mathcal{A})$ iff $\rho' \in L(\mathcal{A})$ [SPKM08]
- If $\mathcal{A}$ is an...
  - ϵ-TA: $f^{-1}(f(L(\mathcal{A}))) \supseteq L(\mathcal{A})$
  - ϵ-IRTA: $f^{-1}(f(L(\mathcal{A}))) = L(\mathcal{A})$
- Therefore $L(\mathcal{A}) \subseteq L(\mathcal{B})$ iff $f(L(\mathcal{A})) \subseteq f(L(\mathcal{B}))$

$\delta\checkmark$-Regularity

- $\delta\checkmark$-Regular Language: a regular subset of $(\Sigma^* \parallel ((\delta\checkmark)^* + \delta(\checkmark\delta)^*)).\Sigma$
- In a $\delta\checkmark$-word, when labels from Σ are ignored...
  - Pattern if nonempty starts with a δ
  - δ's and $\checkmark$'s strictly alternate
- $\delta\checkmark$-Automaton: Deterministic finite state automaton accepting a $\delta\checkmark$-regular language
- Questions: Where do we end up if we go back to the timed world? How about ϵ-freeness?

$\delta\checkmark$-Regularity

- $\delta\checkmark$-Regular Language: a regular subset of $(\Sigma^* \mid\mid ((\delta\checkmark)^* + \delta(\checkmark\delta)^*)).\Sigma$
- In a $\delta\checkmark$-word, when labels from Σ are ignored...
  - Pattern if nonempty starts with a δ
  - δ's and $\checkmark$'s strictly alternate
- $\delta\checkmark$-Automaton: Deterministic finite state automaton accepting a $\delta\checkmark$-regular language
- Questions: Where do we end up if we go back to the timed world? How about ϵ-freeness?

$\delta\checkmark$-Regularity

- $\delta\checkmark$-Regular Language: a regular subset of $(\Sigma^* \mid\mid ((\delta\checkmark)^* + \delta(\checkmark\delta)^*)).\Sigma$
- In a $\delta\checkmark$-word, when labels from Σ are ignored...
  - Pattern if nonempty starts with a δ
  - δ's and $\checkmark$'s strictly alternate
- $\delta\checkmark$-Automaton: Deterministic finite state automaton accepting a $\delta\checkmark$-regular language
- Questions: Where do we end up if we go back to the timed world? How about ϵ-freeness?

$\delta\checkmark$-Regularity

- $\delta\checkmark$-Regular Language: a regular subset of $(\Sigma^* \mid\mid ((\delta\checkmark)^* + \delta(\checkmark\delta)^*)).\Sigma$
- In a $\delta\checkmark$-word, when labels from Σ are ignored...
  - Pattern if nonempty starts with a δ
  - δ's and $\checkmark$'s strictly alternate
- $\delta\checkmark$-Automaton: Deterministic finite state automaton accepting a $\delta\checkmark$-regular language
- Questions: Where do we end up if we go back to the timed world? How about ϵ-freeness?

Outline

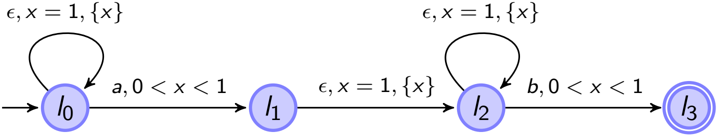
Per-IRTA Are NOT Sufficient

- Cannot go from $\delta\checkmark$-automata to Per-IRTA: Per-IRTA $\subsetneq$ ϵ-IRTA
- The following is an ϵ-IRTA

  Language accepted $L = \{\langle(a, \tau_1), (b, \tau_2)\rangle : \tau_1$ and τ_2 are non-integral, and are separated by at least one integer$\}$
- There exists no Per-IRTA accepting L:
  - Say there is a Per-IRTA $\mathcal{A}$ for L and k is the max. constant
  - Let $k < k_1 < k_2$. Therefore $\langle(a, k_1 + .5), (b, k_2 + .5)\rangle$ has an accepting run over $\mathcal{A}$
  - Since no clock was reset on the a-transition, k_1 and k_2 can be chose such that $\langle(a, k_2 + .5), (b, k_2 + .5)\rangle$ also has an accepting run via the same edges in $\mathcal{A}$

Per-IRTA Are NOT Sufficient

- Cannot go from $\delta\checkmark$-automata to Per-IRTA: Per-IRTA $\subsetneq$ ϵ-IRTA
- The following is an ϵ-IRTA

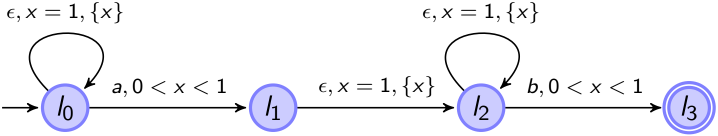


Language accepted $L = \{\langle (a, \tau_1), (b, \tau_2) \rangle : \tau_1$ and τ_2 are non-integral, and are separated by at least one integer$\}$

- There exists no Per-IRTA accepting L:
  - Say there is a Per-IRTA $\mathcal{A}$ for L and k is the max. constant
  - Let $k < k_1 < k_2$. Therefore $\langle (a, k_1 + .5), (b, k_2 + .5) \rangle$ has an accepting run over $\mathcal{A}$
  - Since no clock was reset on the a-transition, k_1 and k_2 can be chose such that $\langle (a, k_2 + .5), (b, k_2 + .5) \rangle$ also has an accepting run via the same edges in $\mathcal{A}$

Per-IRTA Are NOT Sufficient

- Cannot go from $\delta\checkmark$-automata to Per-IRTA: Per-IRTA $\subsetneq$ ϵ-IRTA
- The following is an ϵ-IRTA

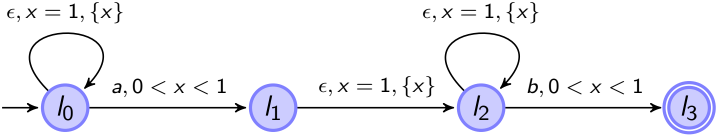


Language accepted $L = \{\langle (a, \tau_1), (b, \tau_2) \rangle : \tau_1$ and τ_2 are non-integral, and are separated by at least one integer$\}$

- There exists no Per-IRTA accepting L:
  - Say there is a Per-IRTA $\mathcal{A}$ for L and k is the max. constant
  - Let $k < k_1 < k_2$. Therefore $\langle (a, k_1 + .5), (b, k_2 + .5) \rangle$ has an accepting run over $\mathcal{A}$
  - Since no clock was reset on the a-transition, k_1 and k_2 can be chose such that $\langle (a, k_2 + .5), (b, k_2 + .5) \rangle$ also has an accepting run via the same edges in $\mathcal{A}$

Per-IRTA Are NOT Sufficient

- Cannot go from $\delta\checkmark$-automata to Per-IRTA: Per-IRTA $\subsetneq$ ϵ-IRTA
- The following is an ϵ-IRTA

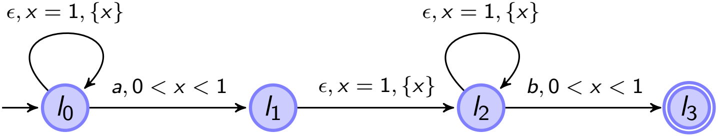


Language accepted $L = \{\langle (a, \tau_1), (b, \tau_2) \rangle : \tau_1$ and τ_2 are non-integral, and are separated by at least one integer$\}$

- There exists no Per-IRTA accepting L:
  - Say there is a Per-IRTA $\mathcal{A}$ for L and k is the max. constant
  - Let $k < k_1 < k_2$. Therefore $\langle (a, k_1 + .5), (b, k_2 + .5) \rangle$ has an accepting run over $\mathcal{A}$
  - Since no clock was reset on the a-transition, k_1 and k_2 can be chose such that $\langle (a, k_2 + .5), (b, k_2 + .5) \rangle$ also has an accepting run via the same edges in $\mathcal{A}$

Per-IRTA Are NOT Sufficient

- Cannot go from $\delta\checkmark$-automata to Per-IRTA: Per-IRTA $\subsetneq$ ϵ-IRTA
- The following is an ϵ-IRTA

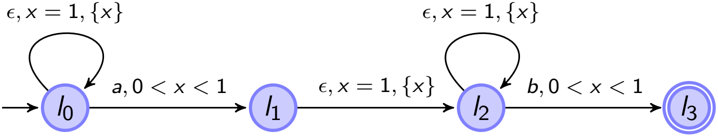


Language accepted $L = \{\langle(a, \tau_1), (b, \tau_2)\rangle : \tau_1$ and τ_2 are non-integral, and are separated by at least one integer$\}$

- There exists no Per-IRTA accepting L:
  - Say there is a Per-IRTA $\mathcal{A}$ for L and k is the max. constant
  - Let $k < k_1 < k_2$. Therefore $\langle(a, k_1 + .5), (b, k_2 + .5)\rangle$ has an accepting run over $\mathcal{A}$
  - Since no clock was reset on the a-transition, k_1 and k_2 can be chose such that $\langle(a, k_2 + .5), (b, k_2 + .5)\rangle$ also has an accepting run via the same edges in $\mathcal{A}$

Per-IRTA Are NOT Sufficient

- Cannot go from $\delta\checkmark$-automata to Per-IRTA: Per-IRTA $\subsetneq$ ϵ-IRTA
- The following is an ϵ-IRTA



  Language accepted $L = \{\langle (a, \tau_1), (b, \tau_2) \rangle : \tau_1$ and τ_2 are non-integral, and are separated by at least one integer$\}$
- There exists no Per-IRTA accepting L:
  - Say there is a Per-IRTA $\mathcal{A}$ for L and k is the max. constant
  - Let $k < k_1 < k_2$. Therefore $\langle (a, k_1 + .5), (b, k_2 + .5) \rangle$ has an accepting run over $\mathcal{A}$
  - Since no clock was reset on the a-transition, k_1 and k_2 can be chose such that $\langle (a, k_2 + .5), (b, k_2 + .5) \rangle$ also has an accepting run via the same edges in $\mathcal{A}$

GRTA

- Generalized Resets: Reset only the integer part of the clock; keeps the fractional value unchanged
- GRTA: Variant of timed automata wherein all the resets are generalized
- Per-GRTA: GRTA wherein the atomic constraints can be periodic. We will translate $\delta\checkmark$-automata to Per-GRTA.

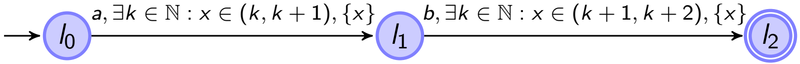
  Per-GRTA for L:

GRTA

- Generalized Resets: Reset only the integer part of the clock; keeps the fractional value unchanged
- GRTA: Variant of timed automata wherein all the resets are generalized
- Per-GRTA: GRTA wherein the atomic constraints can be periodic. We will translate $\delta\checkmark$-automata to Per-GRTA. Per-GRTA for L:

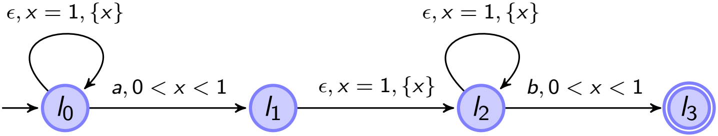
GRTA

- Generalized Resets: Reset only the integer part of the clock; keeps the fractional value unchanged
- GRTA: Variant of timed automata wherein all the resets are generalized
- Per-GRTA: GRTA wherein the atomic constraints can be periodic. We will translate $\delta\checkmark$-automata to Per-GRTA.
  Per-GRTA for L:



Construction by Example

- ϵ-IRTA

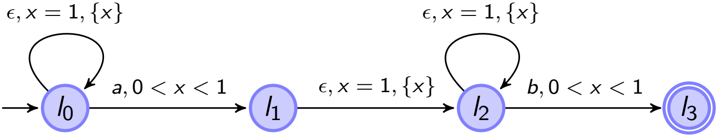


- $\delta\checkmark$-Automaton (equivalent to the one obtained by our construction)

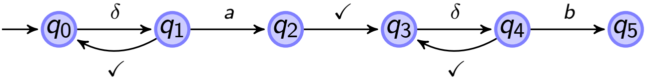
- The idea is to encode the time distances represented by $\delta\checkmark$-paths into clock constraints. Deterministic, 1-Clock Per-GRTA

Construction by Example

- ϵ-IRTA



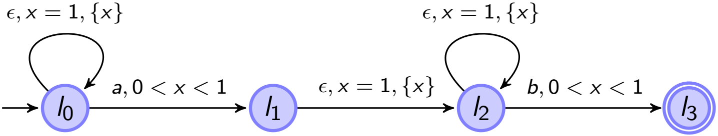
- $\delta\checkmark$-Automaton (equivalent to the one obtained by our construction)



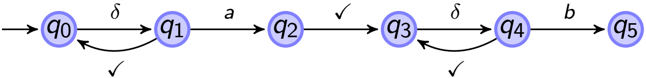
- The idea is to encode the time distances represented by $\delta\checkmark$-paths into clock constraints. Deterministic, 1-Clock Per-GRTA

Construction by Example

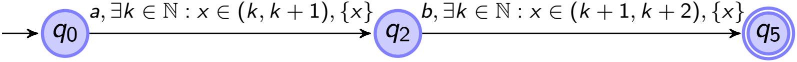
- ϵ-IRTA



- $\delta\checkmark$-Automaton (equivalent to the one obtained by our construction)



- The idea is to encode the time distances represented by $\delta\checkmark$-paths into clock constraints. Deterministic, 1-Clock Per-GRTA



Hence. . .

Theorem

For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic Per-GRTA $\mathcal{B}$ with at most double exponential blowup in number of locations

- For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic ϵ-IRTA $\mathcal{B}$ with at most double exponential blowup in number of locations (via $\delta\checkmark$-automata):
  - Replace each $\checkmark$-transition by an ϵ-transition with constraint $x = 1$; Reset x
  - Replace each δ-transition by an ϵ-transition with constraint $0 < x < 1$
  - Each Σ-transition that has to occur at integral time point is guarded by $x = 0$
  - Each Σ-transition that has to occur at nonintegral time point is guarded by $0 < x < 1$
- $\delta\checkmark$-Regularity characterizes the timed languages accepted by ϵ-IRTA

Hence. . .

Theorem

For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic Per-GRTA $\mathcal{B}$ with at most double exponential blowup in number of locations

- For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic ϵ-IRTA $\mathcal{B}$ with at most double exponential blowup in number of locations (via $\delta\checkmark$-automata):
  - Replace each $\checkmark$-transition by an ϵ-transition with constraint $x = 1$; Reset x
  - Replace each δ-transition by an ϵ-transition with constraint $0 < x < 1$
  - Each Σ-transition that has to occur at integral time point is guarded by $x = 0$
  - Each Σ-transition that has to occur at nonintegral time point is guarded by $0 < x < 1$
- $\delta\checkmark$-Regularity characterizes the timed languages accepted by ϵ-IRTA

Hence...

Theorem

For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic Per-GRTA $\mathcal{B}$ with at most double exponential blowup in number of locations

- For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic ϵ-IRTA $\mathcal{B}$ with at most double exponential blowup in number of locations (via $\delta\checkmark$-automata):
  - Replace each $\checkmark$-transition by an ϵ-transition with constraint $x = 1$; Reset x
  - Replace each δ-transition by an ϵ-transition with constraint $0 < x < 1$
  - Each Σ-transition that has to occur at integral time point is guarded by $x = 0$
  - Each Σ-transition that has to occur at nonintegral time point is guarded by $0 < x < 1$
- $\delta\checkmark$-Regularity characterizes the timed languages accepted by ϵ-IRTA

Hence. . .

Theorem

For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic Per-GRTA $\mathcal{B}$ with at most double exponential blowup in number of locations

- For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic ϵ-IRTA $\mathcal{B}$ with at most double exponential blowup in number of locations (via $\delta\checkmark$-automata):
  - Replace each $\checkmark$-transition by an ϵ-transition with constraint $x = 1$; Reset x
  - Replace each δ-transition by an ϵ-transition with constraint $0 < x < 1$
  - Each Σ-transition that has to occur at integral time point is guarded by $x = 0$
  - Each Σ-transition that has to occur at nonintegral time point is guarded by $0 < x < 1$
- $\delta\checkmark$-Regularity characterizes the timed languages accepted by ϵ-IRTA

Hence. . .

Theorem

For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic Per-GRTA $\mathcal{B}$ with at most double exponential blowup in number of locations

- For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic ϵ-IRTA $\mathcal{B}$ with at most double exponential blowup in number of locations (via $\delta\checkmark$-automata):
  - Replace each $\checkmark$-transition by an ϵ-transition with constraint $x = 1$; Reset x
  - Replace each δ-transition by an ϵ-transition with constraint $0 < x < 1$
  - Each Σ-transition that has to occur at integral time point is guarded by $x = 0$
  - Each Σ-transition that has to occur at nonintegral time point is guarded by $0 < x < 1$
- $\delta\checkmark$-Regularity characterizes the timed languages accepted by ϵ-IRTA

Hence. . .

Theorem

For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic Per-GRTA $\mathcal{B}$ with at most double exponential blowup in number of locations

- For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic ϵ-IRTA $\mathcal{B}$ with at most double exponential blowup in number of locations (via $\delta\checkmark$-automata):
  - Replace each $\checkmark$-transition by an ϵ-transition with constraint $x = 1$; Reset x
  - Replace each δ-transition by an ϵ-transition with constraint $0 < x < 1$
  - Each Σ-transition that has to occur at integral time point is guarded by $x = 0$
  - Each Σ-transition that has to occur at nonintegral time point is guarded by $0 < x < 1$
- $\delta\checkmark$-Regularity characterizes the timed languages accepted by ϵ-IRTA

Hence. . .

Theorem

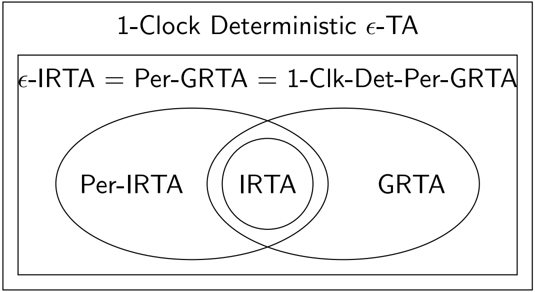
For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic Per-GRTA $\mathcal{B}$ with at most double exponential blowup in number of locations

- For every ϵ-IRTA $\mathcal{A}$ we can effectively construct a language equivalent 1-clock, deterministic ϵ-IRTA $\mathcal{B}$ with at most double exponential blowup in number of locations (via $\delta\checkmark$-automata):
    - Replace each $\checkmark$-transition by an ϵ-transition with constraint $x = 1$; Reset x
    - Replace each δ-transition by an ϵ-transition with constraint $0 < x < 1$
    - Each Σ-transition that has to occur at integral time point is guarded by $x = 0$
    - Each Σ-transition that has to occur at nonintegral time point is guarded by $0 < x < 1$
- $\delta\checkmark$-Regularity characterizes the timed languages accepted by ϵ-IRTA

Outline

Relative Expressive Power

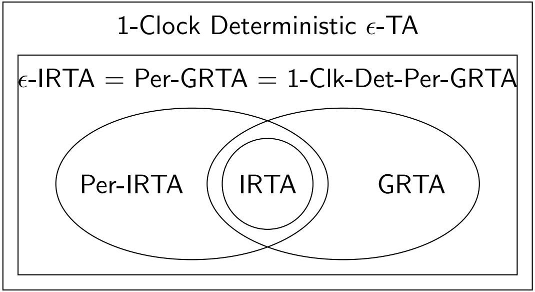
- (i) IRTA $\subsetneq$ Per-IRTA, (ii) GRTA $\not\subseteq$ Per-IRTA, (iii) IRTA $\subsetneq$ GRTA and (iv) Per-IRTA $\not\subseteq$ GRTA (See Full Version of the Paper)



- What are the characterizations if the restriction on clock valuations is just that their fractional values are equal, but need not be equal to the fractional part of the global time?

Relative Expressive Power

- (i) IRTA $\subsetneq$ Per-IRTA, (ii) GRTA $\not\subseteq$ Per-IRTA, (iii) IRTA $\subsetneq$ GRTA and (iv) Per-IRTA $\not\subseteq$ GRTA (See Full Version of the Paper)



- What are the characterizations if the restriction on clock valuations is just that their fractional values are equal, but need not be equal to the fractional part of the global time?

One Clock Timed Automata

- 1C-TA are well behaved
  - $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is decidable if $\mathcal{B}$ is 1C-TA [OW04]
  - Reachability is NLOGSPACE-complete for 1C-TA [AOQW07]
- Equi-Fractional Timed Automata: A timed automaton is an EFTA iff for each resetting edge either
- A 1-clock timed automaton is an EFTA by definition

Theorem

For every EFTA a language equivalent 1-Clock EFTA can be effectively constructed: EFTA = 1C-TA

One Clock Timed Automata

- 1C-TA are well behaved
  - $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is decidable if $\mathcal{B}$ is 1C-TA [OW04]
  - Reachability is NLOGSPACE-complete for 1C-TA [AOQW07]
- Equi-Fractional Timed Automata: A timed automaton is an EFTA iff for each resetting edge either
  - has an atomic constraint of the form $x = c$, or
  - resets all the clocks
- A 1-clock timed automaton is an EFTA by definition

Theorem

For every EFTA a language equivalent 1-Clock EFTA can be effectively constructed: EFTA = 1C-TA

One Clock Timed Automata

- 1C-TA are well behaved
  - $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is decidable if $\mathcal{B}$ is 1C-TA [OW04]
  - Reachability is NLOGSPACE-complete for 1C-TA [AOQW07]
- Equi-Fractional Timed Automata: A timed automaton is an EFTA iff for each resetting edge either
  - has an atomic constraint of the form $x = c$ or,
  - resets all the clocks
- A 1-clock timed automaton is an EFTA by definition

Theorem

For every EFTA a language equivalent 1-Clock EFTA can be effectively constructed: EFTA = 1C-TA

One Clock Timed Automata

- 1C-TA are well behaved
  - $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is decidable if $\mathcal{B}$ is 1C-TA [OW04]
  - Reachability is NLOGSPACE-complete for 1C-TA [AOQW07]
- Equi-Fractional Timed Automata: A timed automaton is an EFTA iff for each resetting edge either
  - has an atomic constraint of the form $x = c$ or,
  - resets all the clocks
- A 1-clock timed automaton is an EFTA by definition

Theorem

For every EFTA a language equivalent 1-Clock EFTA can be effectively constructed: EFTA = 1C-TA

One Clock Timed Automata

- 1C-TA are well behaved
  - $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is decidable if $\mathcal{B}$ is 1C-TA [OW04]
  - Reachability is NLOGSPACE-complete for 1C-TA [AOQW07]
- Equi-Fractional Timed Automata: A timed automaton is an EFTA iff for each resetting edge either
  - has an atomic constraint of the form $x = c$ or,
  - resets all the clocks
- A 1-clock timed automaton is an EFTA by definition

Theorem

For every EFTA a language equivalent 1-Clock EFTA can be effectively constructed: EFTA = 1C-TA

One Clock Timed Automata

- 1C-TA are well behaved
  - $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is decidable if $\mathcal{B}$ is 1C-TA [OW04]
  - Reachability is NLOGSPACE-complete for 1C-TA [AOQW07]
- Equi-Fractional Timed Automata: A timed automaton is an EFTA iff for each resetting edge either
  - has an atomic constraint of the form $x = c$ or,
  - resets all the clocks
- A 1-clock timed automaton is an EFTA by definition

Theorem

For every EFTA a language equivalent 1-Clock EFTA can be effectively constructed: EFTA = 1C-TA

One Clock Timed Automata

- 1C-TA are well behaved
  - $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is decidable if $\mathcal{B}$ is 1C-TA [OW04]
  - Reachability is NLOGSPACE-complete for 1C-TA [AOQW07]
- Equi-Fractional Timed Automata: A timed automaton is an EFTA iff for each resetting edge either
  - has an atomic constraint of the form $x = c$ or,
  - resets all the clocks
- A 1-clock timed automaton is an EFTA by definition

Theorem

For every EFTA a language equivalent 1-Clock EFTA can be effectively constructed: EFTA = 1C-TA

One Clock Timed Automata

- 1C-TA are well behaved
  - $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is decidable if $\mathcal{B}$ is 1C-TA [OW04]
  - Reachability is NLOGSPACE-complete for 1C-TA [AOQW07]
- Equi-Fractional Timed Automata: A timed automaton is an EFTA iff for each resetting edge either
  - has an atomic constraint of the form $x = c$ or,
  - resets all the clocks
- A 1-clock timed automaton is an EFTA by definition

Theorem

For every EFTA a language equivalent 1-Clock EFTA can be effectively constructed: EFTA = 1C-TA

Reduction

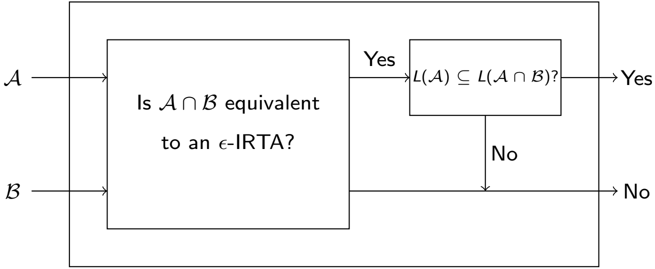
- Is it decidable to determine whether there exists an ϵ-IRTA equivalent to a given ϵ-TA? NO
- The problem $L(\mathcal{A}) \subseteq L(\mathcal{B})$? (equivalently $L(\mathcal{A} \cap \mathcal{B}) = L(\mathcal{A})$?), where $\mathcal{A}$ is an ϵ-IRTA and $\mathcal{B}$ is an ϵ-TA, is undecidable
- We reduce this question to the question above

Reduction

- Is it decidable to determine whether there exists an ϵ-IRTA equivalent to a given ϵ-TA? NO
- The problem $L(\mathcal{A}) \subseteq L(\mathcal{B})$? (equivalently $L(\mathcal{A} \cap \mathcal{B}) = L(\mathcal{A})$?), where $\mathcal{A}$ is an ϵ-IRTA and $\mathcal{B}$ is an ϵ-TA, is undecidable
- We reduce this question to the question above

Reduction

- Is it decidable to determine whether there exists an ϵ-IRTA equivalent to a given ϵ-TA? NO
- The problem $L(\mathcal{A}) \subseteq L(\mathcal{B})$? (equivalently $L(\mathcal{A} \cap \mathcal{B}) = L(\mathcal{A})$?), where $\mathcal{A}$ is an ϵ-IRTA and $\mathcal{B}$ is an ϵ-TA, is undecidable
- We reduce this question to the question above



Future Directions

- Lower bounds on the size of the translated automata
- Does the theory extend to infinite behaviours?
- Direct translation from ϵ-IRTA to Per-GRTA

Future Directions

- Lower bounds on the size of the translated automata
- Does the theory extend to infinite behaviours?
- Direct translation from ϵ-IRTA to Per-GRTA

Future Directions

- Lower bounds on the size of the translated automata
- Does the theory extend to infinite behaviours?
- Direct translation from ϵ-IRTA to Per-GRTA

Thank You

Rajeev Alur and David L. Dill.
A theory of timed automata.
Theoretical Computer Science, 126(2):183–235, 1994.

P.A. Abdulla, J. Ouaknine, K. Quaas, and J. Worrell.
Zone-based universality analysis for single-clock timed automata.
In *Proc. FSEN'07, IPM International Symposium on Fundamentals of Software Engineering*, Lecture Notes in Computer Science 1790, pages 98–112. Springer-Verlag, 2007.

Beatrice Berard, Antoine Petit, Paul Gastin, and Volker Diekert.
Characterization of the expressive power of silent transitions in timed automata.
Fundamenta Informaticae, 36(2-3):145–182, 1998.

Christian Choffrut and Massimiliano Goldwurm.
Timed automata with periodic clock constraints.

Journal of Automata, Languages and Combinatorics, 5(4):371–404, 2000.

Joel Ouaknine and James Worrell.
On the language inclusion problem for timed automata: Closing a decidability gap.
In *LICS '04: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, pages 54–63, Washington, DC, USA, 2004. IEEE Computer Society.

P. Vijay Suman, Paritosh K. Pandya, Shankara Narayanan Krishna, and Lakshmi Manasa.
Timed automata with integer resets: Language inclusion and expressiveness.
In *FORMATS*, pages 78–92. Springer, 2008.
LNCS 5215.