Lecture 20: Timed automata and its reachability (cont.)

Huan Zhang
huan@huan-zhang.com
Review: Integral Timed Automata (ITA)

- **Definition.** A *integral timed automaton* $\mathcal{A} = \langle V, \Theta, A, D, T \rangle$ where
  - $V = X \cup \{l\}$, where $X$ is a set of $n$ clocks and $l$ is a discrete state variable of finite type $L$. The *state space is* $\text{val}(V) \times L$
  - $A$ is a finite set
  - $D$ is a set of transitions such that
    - The preconditions are described by clock constraints $\Phi(X)$
    - $\langle x, l \rangle_a \rightarrow \langle x', l' \rangle$ implies either $x' = x$ or $x' = 0$ (time is reset to 0, or no change)
  - $T$ set of clock trajectories for the clock variables in $X$
Review: Control-state Reachability of ITA: construct a bisimilar FA
Corresponding FA

\[ |X|! 2^{|X|} \prod_{z \in X} (2c_{\mathcal{AZ}} + 2) \]

Drastically increasing with the number of clocks
Rational Timed Automata: Decidable

Definition. A \textit{rational timed automaton} is a HA $\mathcal{A} = \langle V, \Theta, A, \mathcal{D}, \mathcal{T} \rangle$ where

- $V = X \cup \{loc\}$, where $X$ is a set of $n$ clocks and $l$ is a discrete state variable of finite type $L$
- $A$ is a finite set
- $\mathcal{D}$ is a set of transitions such that
  - The guards are described by \textit{rational} clock constraints $\Phi(X)$
  - $\langle x, l \rangle \xrightarrow{a} \langle x', l' \rangle$ implies either $x' = x$ or $x = 0$
- $\mathcal{T}$ set of clock trajectories for the clock variables in $X$

Convert to ITA by multiply clocks by a factor $q$
Multi-Rate Automaton: decidable

- **Definition.** A **multirate automaton** is $\mathcal{A} = \langle V, Q, \Theta, A, \mathcal{D}, \mathcal{T} \rangle$ where
  - $V = X \cup \{ \text{loc} \}$, where $X$ is a set of $n$ **continuous variables** and $\text{loc}$ is a discrete state variable of finite type $L$
  - $A$ is a finite set of actions
  - $\mathcal{D}$ is a set of transitions such that
    - The guards are described by **rational** clock constraints $\Phi(X)$
    - $\langle x, l \rangle \rightarrow a \rightarrow \langle x', l' \rangle$ implies either $x' = c$ or $x' = x$
  - $\mathcal{T}$ set of trajectories such that
    - for each variable $x \in X \exists k$ such that $\tau \in \mathcal{T}$, $t \in \tau. \text{dom}$
      \[ \tau(t).x = \tau(0).x + k \cdot t \]

*Convert to RTA by multiply clocks by a factor $q$*

Slides adapted from Prof. Sayan Mitra’s slides in Fall 2021
Example: Multi-rate to rational TA

Slides adapted from Prof. Sayan Mitra's slides in Fall 2021
Rectangular HA: undecidable

**Definition.** A **rectangular hybrid automaton (RHA)** is a HA $\mathcal{A} = \langle V, A, T, D \rangle$ where

- $V = X \cup \{loc\}$, where $X$ is a set of $n$ continuous variables and $loc$ is a discrete state variable of finite type $L$
- $A$ is a finite set
- $\mathcal{T} = \bigcup_{\ell} T_\ell$ set of trajectories for $X$
  - For each $\tau \in T_\ell$, $x \in X$ either (i) $d(x) = k_\ell$ or (ii) $d(x) \in [k_{\ell 1}, k_{\ell 2}]$
  - Equivalently, (i) $\tau(t)\mid x = \tau(0)\mid x + k_\ell t$
  - (ii) $\tau(0)\mid x + k_{\ell 1} t \leq \tau(t)\mid x \leq \tau(0)\mid x + k_{\ell 2} t$
- $\mathcal{D}$ is a set of transitions such that
  - Guards are described by rational clock constraints
  - $\langle x, l \rangle \rightarrow_\alpha \langle x', l' \rangle$ implies $x' = x$ or $x' \in [c_1, c_2]$
Reachability of Rectangular HA

• Is this problem decidable? No

• We will see that the control state reachability (CSR) problem for rectangular hybrid automata (RHA) is undecidable

• This implies that automatic verification of invariants and safety properties is also impossible for this class of models

• The result was shown by Henzinger et al. [1995] through a reduction from the Halting problem of two counter machines
General reductions: Using known hard problem $B$ to show hardness of $A$

Given $B$ is known to be hard
Suppose (for the sake of contradiction) $A$ is solvable
If we can construct a reduction $f: B \rightarrow A$ (from $B$ to $A$) then $B$ becomes easy, which is a contradiction
Reduction from Halting Problem for 2CM

2CM Halting problem: p

Translator f(p)

CSR for RHA: f(p)

CSR algorithm for Rectangular HA

2 CM Halting problem decider

Yes

No

Suppose CSR for RHA is decidable.
If we can construct a reduction from 2CM Halting Problem to CSR for RHA then 2CM Halting problem is also decidable.
Counter Machines

An n-counter machine is an elementary computer with n-unbounded counters and a finite program written in a minimalistic assembly language. More precisely: A 2-counter machine (2CM) is a discrete transition system with the following components:

• Two **nonnegative** integer counters C and D. Both are initialized to 0.

• A finite program with one of these instructions at each location (or line):
  • INCC, INCD: increments counter C (or D)
  • DECC, DECD: decrements counter C (or D), provided it is not 0,
  • JNZC, JNZD [label]: moves the program control to line *label* provided that counter C (or D) is not zero.
Example 2CM for multiplication

A 2-counter machine for multiplying 2x3 is shown below.

INCC;
INCC; % C = 2
INCD; % LOOP
INCD;
INCD;
INCD;
DECC;
JNZC 3; % Jump to LOOP
% HALT

Exercise: Show that any k-counter machine can be simulated by a 2CM.
Halting problem for 2CM

- A **configuration** of a 2CM is a triple \((pc, C, D)\)
  - \(pc\) is the program counter that stores the next line to be executed
  - \(C, D\) are values of the counter

- A sequence of configurations \((pc_0, D_0, C_0), (pc_1, D_1, C_1), \ldots\) is an **execution** if the \(i\)th configuration goes to the \((i+1)\)st configuration in the sequence executing the instruction in line \(pci\)

- Given a 2CM \(M\) and a special halting location \((pc_{halt})\), the Halting problem requires us to decide whether all executions of \(M\) reach the halting location

- Theorem [Minsky 67]. The Halting problem for 2CMs is undecidable.
Reduction from 2CM to CSR-RHS

We have to construct a function (reduction) that maps instances of 2CM-Halt to instances of CSR-RHA
Reduction from 2CM to CSR-RHS

- Program counter pc
- Counters C, D
- Instructions (program)
- Halting location

- Locations, sequence of locations
- Clocks c, d that can go at some constant rates $k_1, k_2, \ldots$
- Transitions: widgets
- Particular location / control state (to which we will check CSR)
Idea of reduction (an RHA compiler)

• Two clocks \( (k_2 > k_1) \)
  • \( c = k_1 \left( \frac{k_2}{k_1} \right)^C \)
  • \( d = k_1 \left( \frac{k_2}{k_1} \right)^D \)

• INCC
  • \( k_1 \left( \frac{k_2}{k_1} \right)^{C+1} = c \left( \frac{k_2}{k_1} \right) \)

• DECC
  • \( k_1 \left( \frac{k_2}{k_1} \right)^{C-1} = c \left( \frac{k_1}{k_2} \right) \)

• checking nonzero:
  • \( c > k_1 \)
A widget that preserves the value of clock \( c \)

[precondition] \[ c = Wk_1 \]

{effect} \[ c := 0 \]

Transitions and clock guards for this control state

\[
\begin{aligned}
\dot{a} &= 1 \\
\dot{c} &= k_1 \\
a &\leq W \\
c &\leq k_1W
\end{aligned}
\]

\[ a = W \]
A widget for checking JNZC (c < k₁)
A widget implementing INCC

First outgoing transition sets $z = k_2 c$ and the next outgoing transition sets $c = z \ast \left(\frac{1}{k_1}\right)$

**Exercise:** Show the widget for DECC
Suppose CSR for RHA is decidable
If we can construct a reduction from 2CM Halting Problem to CSR for RHA then 2CM Halting problem is also decidable
**Theorem:** CSR for RHA is undecidable
Initialized Rectangular HA

Definition. An initialized rectangular hybrid automaton (IRHA) is a RHA $\mathcal{A}$ where

- $V = X \cup \{loc\}$, where $X$ is a set of $n$ continuous variables and $\{loc\}$ is a discrete state variable of finite type $\ell$
- $A$ is a finite set
- $\mathcal{T} = \bigcup_\ell \mathcal{T}_\ell$ set of trajectories for $X$
  - For each $\tau \in \mathcal{T}_\ell$, $x \in X$ either (i) $d(x) = k_\ell$ or (ii) $d(x) \in [k_{\ell_1}, k_{\ell_2}]$
  - Equivalently, (i) $\tau(t)[x = \tau(0)[x + k_\ell t$
    (ii) $\tau(0)[x + k_{\ell_1} t \leq \tau(t)[x \leq \tau(0)[x + k_{\ell_2} t$
- $\mathcal{D}$ is a set of transitions such that
  - Guards are described by rational clock constraints
  - $\langle x, l \rangle \rightarrow_\alpha \langle x', l' \rangle$ implies if dynamics $d(x)$ changes from $\ell$ to $\ell'$ then $x' \in [c_1, c_2]$, otherwise $x' = x$ if $d(x)$ is not changed
Example IRHA

c := 0; d := 1

\[ \begin{align*}
\dot{c} & \in [1,3] \\
\dot{d} & \in [-3, -2] \\
\end{align*} \]

\[ \begin{align*}
c & \geq 0 \land d \leq 2 \\
d & := 1 \\
\end{align*} \]

dynamics for d changes, so d must be reset

\[ \begin{align*}
\dot{c} & \in [1,3] \\
\dot{d} & \in [1,2] \\
\end{align*} \]

dynamics for c changes

\[ \begin{align*}
c & \leq 5 \land d \leq -3 \\
c & := 4 \\
\end{align*} \]

\[ \begin{align*}
d & \leq -5 \\
d & := -4 \\
\end{align*} \]

dynamics for d changes

\[ \begin{align*}
\dot{c} & \in [-4, -2] \\
\dot{d} & \in [-3, -2] \\
\end{align*} \]

dynamics for c changes

\[ \begin{align*}
c & \geq -3 \land d \leq -2 \\
c & \in [-1, -2] \\
\end{align*} \]

dynamics for c changes

\[ \begin{align*}
d & \leq -2 \\
\end{align*} \]

Slides adapted from Prof. Sayan Mitra’s slides in Fall 2021
CSR Decidable for IRHA?

• Given an IRHA, check if a particular location is reachable from the initial states
• Is this problem decidable? Yes
• Key idea:
  • Construct a 2n-dimensional initialized multi-rate automaton that is bisimilar to the given IRHA
  • Construct a ITA that is bisimilar to the Singular TA

Slides adapted from Prof. Sayan Mitra’s slides in Fall 2021
From IRHA to Singular HA conversion

For every variable create two variables---tracking the upper and lower bounds

<table>
<thead>
<tr>
<th>IRHA</th>
<th>MRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x )</td>
<td>( x_u; x_l )</td>
</tr>
<tr>
<td>Evolve: ( d(x) \in [a_1, b_1] )</td>
<td>Evolve: ( d(x_u) = a_1; d(x_l) = b_1 )</td>
</tr>
<tr>
<td>Eff: ( x' \in [a_1, b_1] )</td>
<td>Eff: ( x_u = a_1; x_l = b_1 )</td>
</tr>
<tr>
<td>( x' = c )</td>
<td>( x_l = x_u = c )</td>
</tr>
<tr>
<td>Guard: ( x \geq 5 )</td>
<td>( x_l \geq 5 )</td>
</tr>
<tr>
<td></td>
<td>( x_l &lt; 5 \land x_u \geq 5 ) Eff ( x_l = 5 )</td>
</tr>
</tbody>
</table>

Slides adapted from Prof. Sayan Mitra’s slides in Fall 2021
Example IRHA

\[ c := 0; d := 1 \]

---

\[ \begin{align*}
  v1 & \quad \dot{c} \in [1,3] \\
  & \quad \dot{d} \in [-3, -2]
\end{align*} \]

\[ c \geq 0 \land d \leq 2 \]
\[ d := 1 \]

---

\[ \begin{align*}
  v2 & \quad \dot{c} \in [-4, -2] \\
  & \quad \dot{d} \in [-3, -2]
\end{align*} \]

\[ c \leq 5 \land d \leq -3 \]
\[ c := 4 \]

---

\[ \begin{align*}
  v3 & \quad \dot{c} \in [-4, -2] \\
  & \quad \dot{d} \in [1,2]
\end{align*} \]

\[ c \geq 3 \land d \leq -2 \]
\[ c \in [-1, -2] \]

---

\[ \begin{align*}
  v4 & \quad \dot{c} \in [1,3] \\
  & \quad \dot{d} \in [1,2]
\end{align*} \]

\[ c \geq 0 \land d \leq 2 \]
\[ d := 1 \]

---

\[ c := 1 \land d \leq 5 \land d \leq -2 \]
\[ c := 4 \land d \leq -5 \]

---

Slides adapted from Prof. Sayan Mitra’s slides in Fall 2021
Initialized Singular HA

\[ c_l, c_u := 0; d_l, d_u := 1 \]

\[ \begin{align*}
    v1 & : \quad \begin{align*}
        c_l &= 1 \\
        c_u &= 3 \\
        d_l &= -3 \\
        d_u &= -2
    \end{align*} \\

    v2 & : \quad \begin{align*}
        c_l &= -4 \\
        c_u &= -2 \\
        d_l &= -3 \\
        d_u &= -2
    \end{align*} \\

    v3 & : \quad \begin{align*}
        c_l &= -4 \\
        c_u &= -2 \\
        d_l &= 1 \\
        d_u &= 2
    \end{align*} \\

    v4 & : \quad \begin{align*}
        c_l &= 1 \\
        c_u &= 3 \\
        d_l &= 1 \\
        d_u &= 2
    \end{align*}
\]

Slides adapted from Prof. Sayan Mitra’s slides in Fall 2021
Transitions

\[ v1 \]
\[ c_l = 1 \]
\[ c_u = 3 \]
\[ \dot{d}_l = -3 \]
\[ d_u = -2 \]

From IRHA:
\[ c \leq 5 \land d \leq -3 \]
\[ c := 4 \]

\[ \begin{align*}
    c_l &\leq 5 \\
    c_l, c_u &:= 4 \\
    d_u &\leq -3 \\
    \text{no reset} \\
    d_u &> -3 \land d_l \leq -3 \\
    d_u &:= -3
\end{align*} \]

Slides adapted from Prof. Sayan Mitra’s slides in Fall 2021
Initialized Singular HA

\[ c_l, c_u := 0; d_l, d_u := 1 \]

**v1**
- \( \dot{c}_l = 1 \)
- \( \dot{c}_u = 3 \)
- \( \dot{d}_l = -3 \)
- \( \dot{d}_u = -2 \)

**v4**
- \( \dot{c}_l = 1 \)
- \( \dot{c}_u = 3 \)
- \( \dot{d}_l = 1 \)
- \( \dot{d}_u = 2 \)

**v2**
- \( \dot{c}_l = -4 \)
- \( \dot{c}_u = -2 \)
- \( \dot{d}_l = -3 \)
- \( \dot{d}_u = -2 \)

**v3**
- \( \dot{c}_l = -4 \)
- \( \dot{c}_u = -2 \)
- \( \dot{d}_l = 1 \)
- \( \dot{d}_u = 2 \)

**v1**
- \( c_l \geq 0 \land d_l \leq 2 \)
- \( d_l, d_u := 1 \)

**v4**
- \( c_l \leq 0 \land c_u \geq 0 \land d_l \leq 2 \)
- \( c_l := 0d_l, d_u := 1 \)

**v2**
- \( c_l \leq 5 \land d_u \leq -3 \)
- \( c_l, c_u := 4 \)
- \( c_l \leq 5 \land d_l \leq -3 \land d_u > -3 \)
- \( c_l, c_u := 4 \land d_u := -3 \)

**v3**
- \( c_u \geq -3 \land d_u \leq -2 \)
- \( c_l := -2c_u := -1 \)
- \( c_u \geq -3 \land d_l \leq -2 \land d_u > -2 \)
- \( c_l := -2c_u := -1 \land d_u = 2 \)
Practical reachability

Algorithm: BasicReach
1. **Input:** $A = (V, \Theta, A, D, T), d > 0$
   - $R_t, \text{Reach} := \text{val}(V)$
2. $R_t := \Theta$
3. $\text{Reach} := \emptyset$
4. While ($R_t \notin \text{Reach}$)
   - $\text{Reach} := \text{Reach} \cup R_t$
5. $R_t := R_t \cup \text{Post}_D(R_t)$
6. $R_t := \text{Post}_{T(d)}(R_t)$
7. **Output:** Reach

Algorithm: Post$_D$
1. \text{\textbackslash\textbackslash computes post of all transitions}
2. **Input:** $A, D, S_{in}$
3. $S_{out} := \emptyset$
4. For each $a \in A$
5.  - For each $(g_1, g_2) \in S_{in}$
6.  - If $[g_1, g_2] \cap [g_{eq1}, g_{eq2}] \neq \emptyset$
7.  - $S_{out} := S_{out} \cup \langle g_{eq1}, g_{eq2} \rangle$
8. **Output:** $S_{out}$

Algorithm: Post$_{T(d)}$
1. \text{\textbackslash\textbackslash computes post of all trajectories}
2. **Input:** $A, T, S_{in}, d$
3. $S_{out} := \emptyset$
4. For each $t \in L$
5.  - For each $(g_1, g_2) \in S_{in}$
6.  - $P := \cup_{g \in \delta} [g_1, g_2] \oplus [f_{g1}, f_{g2}]$
7.  - $S_{out} := S_{out} \cup \text{Approx}(P)$
8. **Output:** $S_{out}$

Tools:
- SpaceEX
- CORA
- C2E2
- Flow*
- DryVR

Lecture Slides by Sayan Mitra mitras@illinois.edu
Data structures critical for reachability

• Hyperrectangles
  \[ ([g_1; g_2]) = \{ x \in \mathbb{R}^n \mid ||x - g_1||_\infty \leq ||g_2 - g_1||_\infty \} = \Pi_i[g_{1i}, g_{2i}] \]

• Polyhedra

• Zonotopes [Girard 2005]

• Ellipsoids [Kurzhanskiy 2001]

• Support functions [Guernic et al. 2009]

• Generalized star set [Duggirala and Viswanathan 2018]
Reachability in practice

Reach\((X_0, T)\)

\(x(t)\)

Unsafe set \(X_0\)
Verify no collision with **uncertainties**: speeds in [70, 85] mph and acceleration range of NPC
For a different user model C2E2 finds a corner case

Verify no collision with uncertainties like speeds in [70, 85] mph and bigger acceleration range of NPC