Real-Time Systems

Lecture 13: Location Reachability
(or: The Region Automaton)

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Content

Introduction

• Observables and Evolutions
• Duration Calculus (DC)
• Semantical Correctness Proofs
• DC Decidability
• DC Implementables
• PLC-Automata

Timed Automata (TA), Uppaal
• Networks of Timed Automata
• Region/Zone-Abstraction
• TA model-checking
• Extended Timed Automata
• Undecidability Results

Automatic Verification...

...whether a TA satisfies a DC formula, observer-based

Recent Results:

• Timed Sequence Diagrams, or Quasi-equal Clocks,
  or Automatic Code Generation, or ...
The Location Reachability Problem

...is decidable for TA:

- Normalised Constants
- Time Abstract Transition System

- Regions:
  - Equivalence Classes of Clock Valuations

- The Region Automaton
  - ...is finite
  - ...and effectively constructable.

The Constraint Reachability Problem

...is decidable as well.
Given: A timed automaton $\mathcal{A}$ and one of its locations $\ell$.

Question: Is $\ell$ reachable?

That is, is there a transition sequence of the form

$$(\ell_{ini}, \nu_0) \xrightarrow{\lambda_1} (\ell_1, \nu_1) \xrightarrow{\lambda_2} (\ell_2, \nu_2) \xrightarrow{\lambda_3} \ldots \xrightarrow{\lambda_n} (\ell_n, \nu_n)$$

with $\ell_n = \ell$ in the labelled transition system $T(\mathcal{A})$?

- **Note:** Decidability is not so obvious, recall that
  - clocks range over real numbers, thus infinitely many configurations,
  - at each configuration, uncountably many transitions $\xrightarrow{\ell}$ may originate

- **Consequence:** The timed automata as we consider them here cannot encode a 2-counter machine, and they are strictly less expressive than DC.

---

Decidability of Location Reachability for TA
**Claim:** (Theorem 4.33)

The location reachability problem is **decidable** for timed automata.

**Approach:** Constructive proof.

- Observe: clock constraints are simple — w.l.o.g. assume constants $c \in \mathbb{N}_0$.
- **Def. 4.19:** time-abstract transition system $U(A)$ — abstracts from uncountably many delay transitions, still infinite-state.
- **Lemma 4.20:** location reachability of $A$ is **preserved** in $U(A)$.
- **Def. 4.29:** region automaton $R(A)$ — equivalent configurations collapse into regions
- **Lemma 4.32:** location reachability of $U(A)$ is **preserved** in $R(A)$.
- **Lemma 4.28:** $R(A)$ is finite.

**Without Loss of Generality: Natural Constants**

**Recall:** $\varphi ::= x \sim c \mid x - y \sim c \mid \varphi \land \varphi$, $x, y \in X$, $c \in \mathbb{Q}^+_0$, and $\sim \in \{<, >, \leq, \geq\}$.

- Let $C(A) = \{c \in \mathbb{Q}^+_0 \mid c \text{ appears in } A\}$ — $C(A)$ is finite! (Why?)
- Let $t_A$ be the least common multiple of the denominators in $C(A)$.
- Let $t_A \cdot A$ be the TA obtained from $A$ by multiplying all constants by $t_A$.

- $A:
  \begin{align*}
    & \begin{array}{c}
      x > \frac{4}{3} \\
      y \leq 10 \\
      y - z > \frac{5}{2}
    \end{array} \\
    & \quad x < \frac{3}{5} \\
  \end{align*}

- $\begin{align*}
    C(A) &= \left\{ \frac{1}{5}, \frac{1}{3}, 10, \frac{5}{2} \right\} \\
    t_A &= 15
  \end{align*}$
Recall: $\varphi ::= x \sim c \mid x - y \sim c \mid \varphi \land \varphi, \ x, y \in X, \ c \in \mathbb{Q}_0^+, \text{ and } \sim \in \{<, >, \leq, \geq\}$.

- Let $C(\mathcal{A}) = \{c \in \mathbb{Q}_0^+ \mid c \text{ appears in } \mathcal{A}\}$ – $C(\mathcal{A})$ is finite! (Why?)
- Let $t_\mathcal{A}$ be the least common multiple of the denominators in $C(\mathcal{A})$.
- Let $t_\mathcal{A} \cdot \mathcal{A}$ be the TA obtained from $\mathcal{A}$ by multiplying all constants by $t_\mathcal{A}$.

$\mathcal{A}$:

$x < \frac{1}{3}$

$y < 10$

$y - z > 5$

$C(\mathcal{A}) = \{\frac{1}{3}, \frac{1}{4}, 5, 10\}$

$t_\mathcal{A} = 12$

$t_\mathcal{A} \cdot \mathcal{A}$:

$x < 4$

$y < 120$

$y - z > 60$

$C_\varphi = 4$

$c_\varphi = 120$

$c_\varphi \cdot c_\varphi$
Recall: $\varphi ::= x \sim c \mid x - y \sim c \mid \varphi \land \varphi$, $x, y \in X$, $c \in \mathbb{Q}^+_0$, and $\sim \in \{<, >, \leq, \geq\}$.

- Let $C(A) = \{c \in \mathbb{Q}^+_0 \mid c \text{ appears in } A\}$ — $C(A)$ is finite! (Why?)
- Let $t_A$ be the least common multiple of the denominators in $C(A)$.
- Let $t_A \cdot A$ be the TA obtained from $A$ by multiplying all constants by $t_A$.

Then:
- $C(t_A \cdot A) \subset \mathbb{N}_0$.
- A location $\ell$ is reachable in $t_A \cdot A$ if and only if $\ell$ is reachable in $A$.

That is: we can, without loss of generality, in the following consider only timed automata $A$ with $C(A) \subset \mathbb{N}_0$.

Definition. Let $x$ be a clock of timed automaton $A$ (with $C(A) \subset \mathbb{N}_0$).
We denote by $c_x \in \mathbb{N}_0$ the largest time constant $c$ that appears together with $x$ in a constraint of $A$.

Decidability of The Location Reachability Problem

Claim: (Theorem 4.33)

The location reachability problem is decidable for timed automata.

Approach: Constructive proof.

✔ Observe: clock constraints are simple
- w.l.o.g. assume constants $c \in \mathbb{N}_0$.

✘ Def. 4.19: time-abstract transition system $\mathcal{U}(A)$ — abstracts from uncountably many delay transitions, still infinite-state.

✘ Lemma 4.20: location reachability of $A$ is preserved in $\mathcal{U}(A)$.

✘ Def. 4.29: region automaton $\mathcal{R}(A)$ — equivalent configurations collapse into regions

✘ Lemma 4.32: location reachability of $\mathcal{U}(A)$ is preserved in $\mathcal{R}(A)$.

✘ Lemma 4.28: $\mathcal{R}(A)$ is finite.
Recall: $\mathcal{T}(A) = (\text{Conf}(A), \text{Time} \cup B \cap, \{ \lambda \mid \lambda \in \text{Time} \cup B \cap \}, C_{\text{ini}})$

- Note: The $\lambda \rightarrow$ are binary relations on configurations.

**Definition.** Let $A$ be a TA. For all $\langle \ell_1, \nu_1 \rangle, \langle \ell_2, \nu_2 \rangle \in \text{Conf}(A)$,

$$\langle \ell_1, \nu_1 \rangle \lambda_1 \rightarrow \circ \lambda_2 \rightarrow \langle \ell_2, \nu_2 \rangle$$

if and only if there exists some $\langle \ell', \nu' \rangle \in \text{Conf}(A)$ such that

$$\langle \ell_1, \nu_1 \rangle \lambda_1 \rightarrow \langle \ell', \nu' \rangle \text{ and } \langle \ell', \nu' \rangle \lambda_2 \rightarrow \langle \ell_2, \nu_2 \rangle.$$

**Remark.** The following property of time additivity holds.

$$\forall t_1, t_2 \in \text{Time}: t_1 \circ t_2 = t_1 + t_2$$

---

**Time-abstract Transition System**

**Definition 4.19.** [Time-abstract transition system]

Let $A$ be a timed automaton.

The time-abstract transition system $\mathcal{U}(A)$ is obtained from $\mathcal{T}(A)$ (Def. 4.4) by taking

$$\mathcal{U}(A) = (\text{Conf}(A), B \cap, \{ \alpha \mid \alpha \in B \cap \}, C_{\text{ini}})$$

where

$$\alpha \rightarrow \subseteq \text{Conf}(A) \times \text{Conf}(A)$$

is defined as follows: Let $\langle \ell, \nu \rangle, \langle \ell', \nu' \rangle \in \text{Conf}(A)$ be configurations of $A$ and $\alpha \in B \cap$ an action. Then

$$\langle \ell, \nu \rangle \alpha \rightarrow \langle \ell', \nu' \rangle$$

if and only if there exists $t \in \text{Time}$ such that

$$\langle \ell, \nu \rangle \rightarrow t \circ \alpha \rightarrow \langle \ell', \nu' \rangle.$$
\[ (\ell, \nu) \overset{\alpha}{\Rightarrow} (\ell', \nu') \text{ iff } \exists t \in \text{Time} \cdot (\ell, \nu) \overset{t}{\Rightarrow} \overset{\alpha}{\Rightarrow} (\ell', \nu') \]

**Example**

\[ \begin{array}{c}
\text{press?} \\
\text{light} \\
\text{off} \\
x := 0 \\
x > 3 \\
\text{press?} \\
\text{press?} \\
\text{bright}
\end{array} \]

- \((\text{light}, x = 0) \overset{\text{press?}}{\Rightarrow} (\text{off}, x = 27)\) YES, with \(t = 27\) we have \((l, 0) \overset{27}{\Rightarrow} (l, 27) \overset{\text{press?}}{\Rightarrow} (o, 27)\)
- \((\text{off}, x = 4) \overset{\text{press?}}{\Rightarrow} (\text{light}, x = 0)\) YES, any \(t \in \mathbb{R}^+\) works
- \((\text{off}, x = 4) \overset{\text{press?}}{\Rightarrow} (\text{light}, x = 1)\) NO, \((o, 1) \overset{t}{\Rightarrow} (l, t')\) implies \(t' = 0\)
- \((\text{off}, x = 0) \overset{\text{press?}}{\Rightarrow} (\text{light}, x = 5)\) NO, no \(\alpha\) s.t. \((o, 5) \overset{\alpha}{\Rightarrow} (o, 5)\)
- \((\text{off}, x = 0) \overset{\text{press?}}{\Rightarrow} (\text{bright}, x = 5)\) NO, needs two actions
- \((\text{light}, x = 1) \overset{\text{press?}}{\Rightarrow} (\text{bright}, x = 1)\) YES, with \(t = 0\)

**Location Reachability is preserved in \(U(A)\)**

**Lemma 4.20.** For all locations \(\ell\) of a given timed automaton \(A\) the following holds:

\(\ell\) is \((\lambda \rightarrow \cdot)\)-reachable in \(T(A)\) if and only if \(\ell\) is \((\alpha \rightarrow \cdot)\)-reachable in \(U(A)\).

**Proof:**
- \(\Leftarrow\): easy
- \(\Rightarrow\): \(\ell\) is reachable in \(T(A)\)

\[ \begin{array}{c}
(\ell_0, \nu_0) \overset{t_0}{\Rightarrow} (\ell_1, \nu_1) \overset{t_1}{\Rightarrow} (\ell_2, \nu_2) \overset{t_2}{\Rightarrow} \cdots \overset{t_m}{\Rightarrow} (\ell_m, \nu_m) \overset{\alpha_{m+1}}{\Rightarrow} (\ell, \nu_{m+1})
\end{array} \]
Lemma 4.20. For all locations $\ell$ of a given timed automaton $A$ the following holds:

$$\ell$$ is ($\lambda\rightarrow$-)reachable in $T(A)$ if and only if $\ell$ is ($\alpha\Rightarrow$-)reachable in $U(A)$.

Proof:

- $\leftarrow$: easy
- $\Rightarrow$: $\ell$ is reachable in $T(A)$ if

$$\langle \ell_0, \nu_0 \rangle \xrightarrow{t_0} \langle \ell_1, \nu_1 \rangle \xrightarrow{t_1} \langle \ell_2, \nu_2 \rangle \ldots$$

by $t_m \xrightarrow{\alpha m} \langle \ell, \nu_{m+1} \rangle$

implies $\langle \ell_0, \nu_0 \rangle \xrightarrow{\alpha_1} \langle \ell_1, \nu_1 \rangle \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{m+1}} \langle \ell, \nu_{m+1} \rangle$

Decidability of The Location Reachability Problem

Claim: (Theorem 4.33)

The location reachability problem is decidable for timed automata.

Approach: Constructive proof.

- Observe: clock constraints are simple
- w.l.o.g. assume constants $c \in \mathbb{N}_0$.
- Def. 4.19: time-abstract transition system $U(A)$ — abstracts from uncountably many delay transitions, still infinite-state.
- Lemma 4.20: location reachability of $A$ is preserved in $U(A)$.
- Def. 4.29: region automaton $R(A)$ — equivalent configurations collapse into regions
- Lemma 4.32: location reachability of $U(A)$ is preserved in $R(A)$.
- Lemma 4.28: $R(A)$ is finite.
Distinguishing Clock Valuations: One Clock

- Assume $\mathcal{A}$ with only a single clock, i.e. $X = \{x\}$ (recall: $C(\mathcal{A}) \subseteq \mathbb{N}$).

  - $\mathcal{A}$ could detect, for a given $\nu$, whether $\nu(x) \in \{0, \ldots, c_x\}$.

  - $\mathcal{A}$ cannot distinguish $\nu_1$ and $\nu_2$ if $\nu_i(x) \in (k, k + 1), i = 1, 2$, and $k \in \{0, \ldots, c_x - 1\}$.

  - $\mathcal{A}$ cannot distinguish $\nu_1$ and $\nu_2$ if $\nu_i(x) > c_x, i = 1, 2$.

- If $c_x \geq 1$, there are $(2c_x + 2)$ equivalence classes:
  $$\{\{0\}, (0, 1), \{1\}, (1, 2), \ldots, \{c_x\}, (c_x, \infty)\}$$

If $\nu_1(x)$ and $\nu_2(x)$ are in the same equivalence class, then $\nu_1$ and $\nu_2$ are indistinguishable by $\mathcal{A}$. 
• \( X = \{x, y\}, c_x = 1, c_y = 1. \)

**Helper: Floor and Fraction**

• **Recall:**
  
  Each \( q \in \mathbb{R}_0^+ \) can be split into
  
  • **floor** \( \lfloor q \rfloor \in \mathbb{N}_0 \) and
  
  • **fraction** \( \text{frac}(q) \in [0, 1) \)

  such that

  \[
  q = \lfloor q \rfloor + \text{frac}(q).
  \]

  \[
  \lfloor 3.14 \rfloor = 3
  \]

  \[
  \text{frac}(3.14) = 0.14
  \]
**Definition.** Let $X$ be a set of clocks, $c_x \in \mathbb{N}_0$ for each clock $x \in X$, and $\nu_1, \nu_2$ clock valuations of $X$.

We set $\nu_1 \cong \nu_2$ if and only if the following **four** conditions are satisfied:

1. For all $x \in X$, $|\nu_1(x)| = |\nu_2(x)|$ or both $\nu_1(x) > c_x$ and $\nu_2(x) > c_x$.

2. For all $x \in X$ with $\nu_1(x) \leq c_x$,
   
   \[ \frac{\nu_1(x)}{c} = 0 \text{ if and only if } \frac{\nu_2(x)}{c} = 0. \]

3. For all $x, y \in X$,
   \[ |\nu_1(x) - \nu_1(y)| = |\nu_2(x) - \nu_2(y)| \]
   
   or both $|\nu_1(x) - \nu_2(y)| > c$ and $|\nu_2(x) - \nu_2(y)| > c$.

4. For all $x, y \in X$ with $-c \leq \nu_1(x) - \nu_1(y) \leq c$,
   \[ \frac{\nu_1(x) - \nu_1(y)}{c} = 0 \text{ if and only if } \frac{\nu_2(x) - \nu_2(y)}{c} = 0. \]

Where $c = \max\{c_x, c_y\}$.

---

**Example: Regions**

- $(1)$ $\forall x \in X \bullet |\nu_1(x)| = |\nu_2(x)| \lor (\nu_1(x) > c_x \land \nu_2(x) > c_x)$
- $(2)$ $\forall x \in X \bullet \nu_1(x) \leq c_x \iff (\frac{\nu_1(x)}{c} = 0 \iff \frac{\nu_2(x)}{c} = 0)$
- $(3)$ $\forall x, y \in X \bullet |\nu_1(x) - \nu_1(y)| = |\nu_2(x) - \nu_2(y)|$
  \[ \lor (|\nu_1(x) - \nu_1(y)| > c \land |\nu_2(x) - \nu_2(y)| > c) \]
- $(4)$ $\forall x, y \in X \bullet -c \leq \nu_1(x) - \nu_1(y) \leq c$
  \[ \iff (\frac{\nu_1(x) - \nu_1(y)}{c} = 0 \iff \frac{\nu_2(x) - \nu_2(y)}{c} = 0) \]

\[ \nu_1 \cong \nu_2 \text{ because} \]

- $[\nu_1(x)] = [1] = 1 = [\nu_2(x)]$
- $[\nu_1(y)] = [0.8] = 0 = [0.4] = [\nu_2(y)]$
- $\frac{\nu_1(x)}{c} = 0 = \frac{\nu_2(x)}{c}$
  \[ \frac{\nu_1(y)}{c} = \text{frac}(0.8) = 0.8 \neq 0 \]
  \[ \frac{\nu_2(y)}{c} = \text{frac}(0.4) = 0.4 \neq 0 \]
- $[\nu_1(x) - \nu_1(y)] = [1 - 0.8] = 0$
  \[ = [1 - 0.4] = [\nu_2(x) - \nu_2(y)] \]
  \[ \ldots \]
Example: Regions

1. \( \forall x \in X \bullet [\nu_1(x)] = [\nu_2(x)] \lor (\nu_1(x) > c_x \land \nu_2(x) > c_x) \)

2. \( \forall x \in X \bullet \nu_1(x) \leq c_x \iff (\text{frac}(\nu_1(x)) = 0 \iff \text{frac}(\nu_2(x)) = 0) \)

3. \( \forall x, y \in X \bullet [\nu_1(x) - \nu_1(y)] = [\nu_2(x) - \nu_2(y)] \]
   \( \lor (|\nu_1(x) - \nu_1(y)| > c \land |\nu_2(x) - \nu_2(y)| > c) \)

4. \( \forall x, y \in X \bullet -c \leq \nu_1(x) - \nu_1(y) \leq c \iff (\text{frac}(\nu_1(x) - \nu_1(y)) = 0 \iff \text{frac}(\nu_2(x) - \nu_2(y)) = 0) \)

---

**Regions**

**Proposition.** \( \equiv \) is an equivalence relation.

**Definition 4.27.**
For a given valuation \( \nu \) we denote by \([\nu]\) the equivalence class of \( \nu \).

We call the equivalence classes of \( \equiv \) regions.
Definition 4.29. [Region Automaton] The region automaton \( \mathcal{R}(A) \) of the timed automaton \( A \) is the labelled transition system

\[
\mathcal{R}(A) = ( \text{Conf}(\mathcal{R}(A)), B_\tau, \{ \overset{\alpha}{\to}_{\mathcal{R}(A)} \mid \alpha \in B_\tau \}, C_{\text{ini}} )
\]

where

- \( \text{Conf}(\mathcal{R}(A)) = \{ \langle \ell, [\nu] \rangle \mid \ell \in L, \nu : X \to \text{Time}, \nu \models I(\ell) \} \),
- for each \( \alpha \in B_\tau \), \( \langle \ell, [\nu] \rangle \overset{\alpha}{\to}_{\mathcal{R}(A)} \langle \ell', [\nu'] \rangle \) if and only if \( (\ell, \nu) \overset{\alpha}{\to} (\ell', \nu') \) in \( \mathcal{U}(A) \), and
- \( C_{\text{ini}} = \{ \langle \ell_{\text{ini}}, [\nu_{\text{ini}}] \rangle \} \cap \text{Conf}(\mathcal{R}(A)) \) with \( \nu_{\text{ini}}(X) = \{ 0 \} \).

Proposition. The transition relation of \( \mathcal{R}(A) \) is well-defined, that is, independent of the choice of the representative \( \nu \) of a region \([\nu]\).

Example: Region Automaton

\[\begin{array}{c}
\mathcal{R}(A) = \langle \text{off}, \text{light}, \text{bright} \rangle \\
\end{array}\]

\[
\begin{array}{c}
\text{conf}(\mathcal{R}(A)) = \\
\langle \text{off}, [x = 0] \rangle \overset{\text{press}}{\to} \langle \text{light}, [x = 0] \rangle \\
\langle \text{light}, [x = 0] \rangle \overset{\text{press}}{\to} \langle \text{bright}, [x = 0] \rangle \\
\langle \text{bright}, [x = 0] \rangle \overset{\text{press}}{\to} \langle \text{off}, [x = 0] \rangle \\
\end{array}\]

\[
\begin{array}{c}
\text{conf}(\mathcal{R}(A)) = \\
\langle \text{off}, [x = 3] \rangle \overset{\text{press}}{\to} \langle \text{light}, [x = 3] \rangle \\
\langle \text{light}, [x = 3] \rangle \overset{\text{press}}{\to} \langle \text{bright}, [x = 3] \rangle \\
\langle \text{bright}, [x = 3] \rangle \overset{\text{press}}{\to} \langle \text{off}, [x = 3] \rangle \\
\end{array}\]
Remark 4.30. A configuration $\langle \ell, [\nu] \rangle$ is reachable in $R(A)$ if and only if all $\langle \ell, \nu' \rangle$ with $\nu' \in [\nu]$ are reachable.

In other words: it is possible to enter the configuration $\langle \ell, \nu' \rangle$ with an action transition (possibly some delay before).

The clock values reachable by staying / letting time pass in $\ell$ are not explicitly represented by the regions of $R(A)$.

Decidability of The Location Reachability Problem

Claim: (Theorem 4.33)

The location reachability problem is decidable for timed automata.

Approach: Constructive proof.

✔ Observe: clock constraints are simple – w.l.o.g. assume constants $c \in \mathbb{N}_0$.

✔ Def. 4.19: time-abstract transition system $\mathcal{U}(A)$ – abstracts from uncountably many delay transitions, still infinite-state.

✔ Lemma 4.20: location reachability of $A$ is preserved in $\mathcal{U}(A)$.

✔ Def. 4.29: region automaton $R(A)$ – equivalent configurations collapse into regions

✘ Lemma 4.32: location reachability of $\mathcal{U}(A)$ is preserved in $R(A)$.

✘ Lemma 4.28: $R(A)$ is finite.
Lemma 4.32. [Correctness] For all locations $\ell$ of a given timed automaton $A$ the following holds:

$$\ell \text{ is reachable in } U(A) \text{ if and only if } \ell \text{ is reachable in } R(A).$$

For the Proof:

Definition 4.21. [Bisimulation] An equivalence relation $\sim$ on valuations is a (strong) bisimulation if and only if, whenever

$$\nu_1 \sim \nu_2 \text{ and } \langle \ell, \nu_1 \rangle \xrightarrow{\alpha} \langle \ell', \nu_1' \rangle$$

then there exists $\nu_2'$ with $\nu_1' \sim \nu_2'$ and $\langle \ell, \nu_2 \rangle \xrightarrow{\alpha} \langle \ell', \nu_2' \rangle$.

Lemma 4.26. [Bisimulation] $\equiv$ is a strong bisimulation.

Decidability of The Location Reachability Problem

Claim: (Theorem 4.33) The location reachability problem is decidable for timed automata.

Approach: Constructive proof.

✔ Observe: clock constraints are simple
  - w.l.o.g. assume constants $c \in \mathbb{N}_0$.

✔ Def. 4.19: time-abstract transition system $U(A)$ – abstracts from uncountably many delay transitions, still infinite-state.

✔ Lemma 4.20: location reachability of $A$ is preserved in $U(A)$.

✔ Def. 4.29: region automaton $R(A)$ – equivalent configurations collapse into regions

✔ Lemma 4.32: location reachability of $U(A)$ is preserved in $R(A)$.

✘ Lemma 4.28: $R(A)$ is finite.
Lemma 4.28. Let $X$ be a set of clocks, $c_x \in \mathbb{N}_0$ the maximal constant for each $x \in X$, and $c = \max\{c_x \mid x \in X\}$. Then

$$\left(2c + 2\right)^{|X|} \cdot \left(4c + 3\right)^{\frac{1}{2}|X| - (|X| - 1)}$$

is an upper bound on the number of regions.

Proof: Olderog and Dierks (2008)

• Lemma 4.28 in particular tells us that each timed automaton (in our definition) has finitely many regions.

• Note: the upper bound is a worst case / upper bound, not an exact number.
**Claim:** (Theorem 4.33)

The location reachability problem is **decidable** for timed automata.

**Approach:** Constructive proof.

- ✔ Observe: clock constraints are simple — w.l.o.g. assume constants $c \in \mathbb{N}_0$.
- ✔ **Def. 4.19:** time-abstract transition system $\mathcal{U}(A)$ — abstracts from uncountably many delay transitions, still infinite-state.
- ✔ **Lemma 4.20:** location reachability of $A$ is preserved in $\mathcal{U}(A)$.
- ✔ **Def. 4.29:** region automaton $\mathcal{R}(A)$ — equivalent configurations collapse into regions
- ✔ **Lemma 4.32:** location reachability of $\mathcal{U}(A)$ is preserved in $\mathcal{R}(A)$.
- ✔ **Lemma 4.28:** $\mathcal{R}(A)$ is finite.

---

**Putting It All Together**

Let $A = (L, B, X, I, E, \ell_{ini})$ be a timed automaton and $\ell \in L$ a location.

- $\mathcal{R}(A)$ can be constructed effectively.
- There are finitely many locations in $L$ (by definition).
- There are finitely many regions by Lemma 4.28.
- So $Conf(\mathcal{R}(A))$ is finite (by construction).
- It is decidable whether there exists a sequence

$$
\langle \ell_{ini}, [\nu_{ini}] \rangle \xrightarrow{\alpha_{R(A)}} \langle \ell_1, [\nu_1] \rangle \xrightarrow{\alpha_{R(A)}} \cdots \xrightarrow{\alpha_{R(A)}} \langle \ell_n, [\nu_n] \rangle
$$

such that $\ell_n = \ell$ (reachability in graphs).

Thus we have just shown:

> **Theorem 4.33. [Decidability]**
> The location reachability problem for timed automata is **decidable**.
The Constraint Reachability Problem

• **Given**: Timed automaton $A$, one of its locations $\ell$, and a clock constraint $\varphi$.

• **Question**: Is a configuration $(\ell, \nu)$ reachable where $\nu \models \varphi$, i.e. is there a transition sequence of the form

$$\langle \ell_{\text{ini}}, \nu_{\text{ini}} \rangle \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle \xrightarrow{\lambda_3} \ldots \xrightarrow{\lambda_n} \langle \ell_n, \nu_n \rangle = \langle \ell, \nu \rangle$$

in the labelled transition system $\mathcal{T}(A)$ with $\nu \models \varphi$?

• **Note**: we just observed that $\mathcal{R}(A)$ loses some information about the clock valuations that are possible in $A$ from a region.

Theorem 4.34.
The constraint reachability problem for timed automata is decidable.

The Delay Operation

• Let $[\nu]$ be a clock region.
• We set $\text{delay}[\nu] := \{ \nu' + t \mid \nu' \cong \nu \text{ and } t \in \text{Time} \}$.

• **Note**: $\text{delay}[\nu]$ can be represented as a finite union of regions.

For example, with our two-clock example we have

$$\text{delay}[x = y = 0] = [x = y = 0] \cup [0 < x = y < 1] \cup [x = y = 1] \cup [1 < x = y]$$
• Location Reachability Problem: is location $\ell$ reachable in $\mathcal{A}$?

• Decidability proof: $[\text{AD94}]$
  • normalise constants,
  • construct the Time Abstract Transition System
    • “get rid of” delay transitions,
    • still uncountably many configurations
  • collapse equivalent clock valuations into regions
    • obtain finitely many (abstract) configurations
  • construct the Region Automaton
    • it is finite, \( \checkmark \)
    • and preserves location reachability, from $\ell(\mathcal{A})$

• Thus: there are chances to get automatic verification for TA.
• Result can easily be lifted to constraint reachability.

---

References