Real-Time Systems

Lecture 12: Location Reachability (or: The Region Automaton)

2013-06-12

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:

- Networks of Timed Automata
- Uppaal Demo

This Lecture:

- **Educational Objectives:** Capabilities for following tasks/questions.
  - What are decidable problems of TA?
  - How can we show this? What are the essential premises of decidability?
  - What is a region? What is the region automaton of this TA?
  - What's the time abstract system of a TA? Why did we consider this?
  - What can you say about the complexity of Region-automaton based reachability analysis?
- **Content:**
  - ~~Timed Transition System of network of timed automata~~
  - Location Reachability Problem
  - Constructive, region-based decidability proof

The Location Reachability Problem

The Location Reachability Problem

Given: A timed automaton $\mathcal{A}$ and one of its control locations ℓ.
Question: Is ℓ **reachable**?

That is, is there a transition sequence of the form

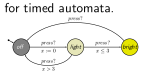
$$\langle \ell_{ini}, \nu_0 \rangle \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle \xrightarrow{\lambda_3} \dots \xrightarrow{\lambda_n} \langle \ell_n, \nu_n \rangle \cancel{= \langle \ell, \nu \rangle}, \ell_n = \ell$$

in the labelled transition system $\mathcal{T}(\mathcal{A})$?

- **Note:** Decidability is not **soo** obvious, recall that
  - clocks range over real numbers, thus infinitely many configurations,
  - at each configuration, uncountably many transitions $\xrightarrow{t}$ 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 The Location Reachability Problem

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



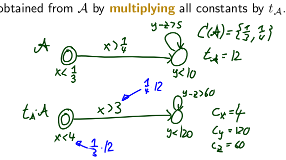
Approach: Constructive proof.

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

Without Loss of Generality: Natural Constants

Recall: Simple clock constraints are $\varphi ::= x \sim c \mid x - y \sim c \mid \varphi \wedge \varphi$ with $x, y \in X$, $c \in \mathbb{Q}_0^+$, and $\sim \in \{<, >, \leq, \geq\}$.

- Let $C(\mathcal{A}) = \{c \in \mathbb{Q}_0^+ \mid c$ 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 $t_\mathcal{A}$.



Without Loss of Generality: Natural Constants

Recall: Simple clock constraints are $\varphi ::= x \sim c \mid x - y \sim c \mid \varphi \wedge \varphi$ with $x, y \in X$, $c \in \mathbb{Q}_0^+$, 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 $t_\mathcal{A}$.
- Then:
  - $C(t_\mathcal{A} \cdot \mathcal{A}) \subset \mathbb{N}_0$.
  - A location ℓ is reachable in $t_\mathcal{A} \cdot \mathcal{A}$ $\underbrace{\text{if and only if}}$ ℓ is reachable in $\mathcal{A}$.
- That is: we can **without loss of generality** in the following consider only timed automata $\mathcal{A}$ with $C(\mathcal{A}) \subset \mathbb{N}_0$.

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

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** $\mathcal{U}(\mathcal{A})$ — abstracts from uncountably many delay transitions, still infinite-state.

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

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

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

✘ **Lem. 4.28**: $\mathcal{R}(\mathcal{A})$ is **finite**.

Helper: Relational Composition

Recall: $\mathcal{T}(\mathcal{A}) = (Conf(\mathcal{A}), \mathsf{Time} \cup B_{?!}, \{\xrightarrow{\lambda} \mid \lambda \in \mathsf{Time} \cup B_{?!}\}, C_{ini})$

- Note: The $\xrightarrow{\lambda}$ are binary relations on configurations.

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

$$\langle \ell_1, \nu_1 \rangle \xrightarrow{\lambda_1} \circ \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle$$

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

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

Helper: Relational Composition

Recall: $\mathcal{T}(\mathcal{A}) = (Conf(\mathcal{A}), \mathsf{Time} \cup B_{?!}, \{\xrightarrow{\lambda} \mid \lambda \in \mathsf{Time} \cup B_{?!}\}, C_{ini})$

- Note: The $\xrightarrow{\lambda}$ are binary relations on configurations.

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

$$\langle \ell_1, \nu_1 \rangle \underbrace{\xrightarrow{\lambda_1} \circ \xrightarrow{\lambda_2}} \langle \ell_2, \nu_2 \rangle$$

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

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

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

$$\forall t_1, t_2 \in \mathsf{Time} : \underbrace{\xrightarrow{t_1} \circ \xrightarrow{t_2}} = \xrightarrow{t_1 + t_2}$$

Time-abstract Transition System

Definition 4.19. [*Time-abstract transition system*]
Let $\mathcal{A}$ be a timed automaton.
The **time-abstract transition system** $\mathcal{U}(\mathcal{A})$ is obtained from $\mathcal{T}(\mathcal{A})$ (Def. 4.4) by taking

$$\mathcal{U}(\mathcal{A}) = (Conf(\mathcal{A}), B_{?!}, \{\xRightarrow{\alpha} \mid \alpha \in B_{?!}\}, C_{ini})$$

where

$$\xRightarrow{\alpha} \subseteq Conf(\mathcal{A}) \times Conf(\mathcal{A})$$

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

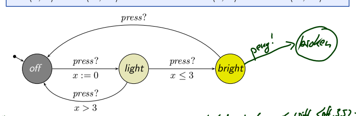
$$\langle \ell, \nu \rangle \xRightarrow{\alpha} \langle \ell', \nu' \rangle$$

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

$$\langle \ell, \nu \rangle \xrightarrow{t} \circ \xrightarrow{\alpha} \langle \ell', \nu' \rangle.$$

Example

$\langle \ell, \nu \rangle \xRightarrow{\alpha} \langle \ell', \nu' \rangle$ iff $\exists\, t \in \mathsf{Time} \bullet \langle \ell, \nu \rangle \xrightarrow{t} \circ \xrightarrow{\alpha} \langle \ell', \nu' \rangle$



$\langle off, x{=}3 \rangle \overset{?}{\Longrightarrow} \langle off, x{=}3.5 \rangle$ NO, $t = 0.5$ is a delay candidate, but no α with $\langle off, 3.5 \rangle \xrightarrow{\alpha} \langle off, 3.5 \rangle$

$\langle off, x{=}4 \rangle \overset{press?}{\Longrightarrow} \langle light, x{=}0 \rangle$ YES, any $t \in \mathbb{R}_0^+$ works, $\alpha = press?$

$\langle off, x{=}4 \rangle \overset{?}{\Longrightarrow} \langle light, x{=}5 \rangle$ NO, $\langle off, x{=}t \rangle \xrightarrow{t'} \circ \xrightarrow{\alpha} \langle light, x = t'' \rangle$ implies $\alpha = press?$ and $t'' = 0$

$\langle off, x{=}7 \rangle \overset{!}{\Longrightarrow} \langle bright, x{=}3 \rangle$ NO, cannot go from off to bright with <u>one</u> action transition

$\langle bright, x{=}13 \rangle \overset{peng!}{\Longrightarrow} \langle broken, x{=}13 \rangle$ YES, $t = 0$ and $\alpha = peng!$

$\langle broken, x{=}13 \rangle \overset{?}{\Longrightarrow} \langle broken, x{=}27 \rangle$ NO, no outgoing edge from broken

Location Reachability is preserved in $\mathcal{U}(\mathcal{A})$

Lemma 4.20. For all locations ℓ of a given timed automaton $\mathcal{A}$ the following holds:

ℓ is reachable in $\mathcal{T}(\mathcal{A})$ if and only if ℓ is reachable in $\mathcal{U}(\mathcal{A})$.

$\xrightarrow{\lambda}$ $\quad \overset{\alpha}{\Longrightarrow}$

Proof:

"$\Leftarrow$": easy

"$\Rightarrow$": ℓ reachable in $\mathcal{T}(\mathcal{A})$

$n_0 \geq 0$, i.e. sequence could be empty

$\Leftrightarrow$ there is $\langle \ell_0, \nu_0 \rangle \xrightarrow{t_{0_1}} \langle \ell_0, \nu_{0_1} \rangle \to \dots \xrightarrow{t_{0_{n_0}}} \langle \ell_0, \nu_{0_{n_0}} \rangle \xrightarrow{\alpha_1} \langle \ell_1, \nu_1 \rangle$

$t_1 := \sum_{i=1}^{n_0} t_{0_{n_i}}$

$\xrightarrow{t_{1_1}} \langle \ell_1, \nu_{1_1} \rangle \to \dots$

$\vdots$

$\xrightarrow{t_{n_1}} \langle \ell_{n-1}, \nu_{n-1_1} \rangle \to \dots \langle \ell_{n-1}, \nu_{n-1_{n_{n-1}}} \rangle \xrightarrow{\alpha_n} \langle \ell_n, \nu_n \rangle$, $\ell_n = \ell$, $n_i \in \mathbb{N}_0$

$\Rightarrow \langle \ell_0, \nu_0 \rangle \overset{\alpha_1}{\Longrightarrow} \langle \ell_1, \nu_1 \rangle \dots \overset{\alpha_n}{\Longrightarrow} \langle \ell_n, \nu_n \rangle$, $\ell_n = \ell$

by $\xrightarrow{t_1} 0 \xrightarrow{\alpha_1}$

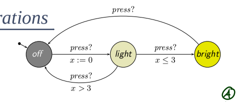
Decidability of The Location Reachability Problem

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

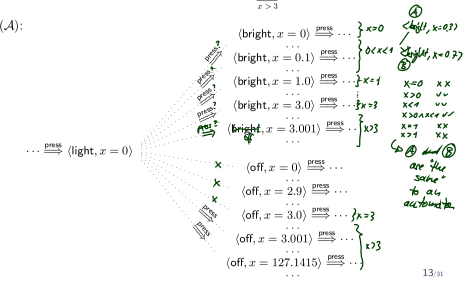
Approach: Constructive proof.

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

Indistinguishable Configurations

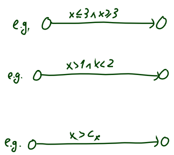


$\mathcal{U}(\mathcal{A})$:



Distinguishing Clock Valuations: One Clock

- Assume $\mathcal{A}$ with only a single clock, i.e. $X = \{x\}$ (**recall**: $C(\mathcal{A}) \subset \mathbb{N}$.)
  - $\mathcal{A}$ **could detect**, for a given ν, whether $\nu(x) \in \{0, \dots, c_x\}$.
  - $\mathcal{A}$ **cannot distinguish** ν_1 and ν_2 if $\nu_i(x) \in (k, k+1)$, $i = 1, 2$, and $k \in \{0, \dots, c_x - 1\}$.
  - $\mathcal{A}$ **cannot distinguish** ν_1 and ν_2 if $\nu_i(x) > c_x$, $i = 1, 2$.



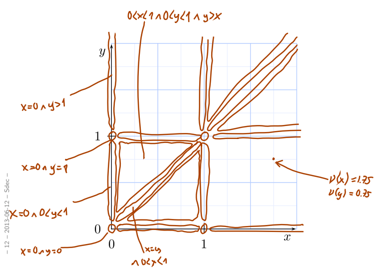
- If $c_x \geq 1$, there are $(2c_x + 2)$ **equivalence classes**:

$$\{\{0\}, (0,1), \{1\}, (1,2), \dots, \{c_x\}, (c_x, \infty)\}$$

If $\nu_1(x)$ and $\nu_2(x)$ are in the **same** equivalence class, then ν_1 and ν_2 are **indistinguishable** by $\mathcal{A}$.

Distinguishing Clock Valuations: Two Clocks

- $X = \{x, y\}$, $c_x = 1$, $c_y = 1$.



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** $frac(q) \in [0, 1)$

such that

$$q = \lfloor q \rfloor + frac(q).$$

An Equivalence-Relation on Valuations

Definition. Let X be a set of clocks, $c_x \in \mathbb{N}_0$ for each clock $x \in X$, and ν_1, ν_2 clock valuations of X.

We set $\nu_1 \cong \nu_2$ iff the following **four** conditions are satisfied.

(1) For all $x \in X$,
$$\lfloor \nu_1(x) \rfloor = \lfloor \nu_2(x) \rfloor \text{ or **both** } \nu_1(x) > c_x \text{ and } \nu_2(x) > c_x.$$

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

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

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

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

Example: Regions

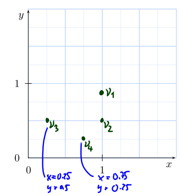
(1) $\forall x \in X : \lfloor \nu_1(x) \rfloor = \lfloor \nu_2(x) \rfloor \vee (\nu_1(x) > c_x \wedge \nu_2(x) > c_x)$

(2) $\forall x \in X : \nu_1(x) \le c_x \implies (frac(\nu_1(x)) = 0 \iff frac(\nu_2(x)) = 0)$

(3) $\forall x, y \in X : \lfloor \nu_1(x) - \nu_1(y) \rfloor = \lfloor \nu_2(x) - \nu_2(y) \rfloor \vee (|\nu_1(x) - \nu_1(y)| > c \wedge |\nu_2(x) - \nu_2(y)| > c)$

(4) $\forall x, y \in X : -c \le \nu_1(x) - \nu_1(y) \le c \implies (frac(\nu_1(x) - \nu_1(y)) = 0 \iff frac(\nu_2(x) - \nu_2(y)) = 0)$

$c_x = 1$
$c_y = 1$



- $\nu_1 \cong \nu_2$:
  - (1) ✓✓
  - (2) ✓✓
  - (3) ✓
  - (4) ✓
- $\nu_3 \not\cong \nu_4$:
  - (1) ✓✓
  - (2) ✓✓
  - (3) $\nu_3(x) - \nu_3(y) = -0.25$ $\lfloor -0.25 \rfloor = -1$; $\nu_4(x) - \nu_4(y) = 0.5$ $\lfloor 0.5 \rfloor = 0$ $\neq$

Regions

Proposition. $\cong$ is an **equivalence relation**.

Definition 4.27. For a given valuation ν we denote by $[\nu]$ the equivalence class of ν. We call equivalence classes of $\cong$ **regions**.

$\{\nu' \mid \nu' \cong \nu\}$

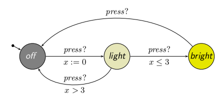
The Region Automaton

Definition 4.29. [*Region Automaton*] The **region automaton** $\mathcal{R}(\mathcal{A})$ of the timed automaton $\mathcal{A}$ is the labelled transition system
$$\mathcal{R}(\mathcal{A}) = (Conf(\mathcal{R}(\mathcal{A})), B_{?!}, \{\xrightarrow{\alpha}_{R(\mathcal{A})} \mid \alpha \in B_{?!}\}, C_{ini})$$
where

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

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

Example: Region Automaton



$\mathcal{U}(\mathcal{A})$:

$\cdots \xRightarrow{\text{press}} \langle \text{light}, x = 0 \rangle$

- $\xRightarrow{\text{press}} \langle \text{bright}, x = 0 \rangle \xRightarrow{\text{press}} \cdots$
- $\xRightarrow{\text{press}} \langle \text{bright}, x = 0.1 \rangle \xRightarrow{\text{press}} \cdots$
- $\xRightarrow{\text{press}} \langle \text{bright}, x = 1.0 \rangle \xRightarrow{\text{press}} \cdots$
- $\xRightarrow{\text{press}} \langle \text{bright}, x = 1.31415 \rangle \xRightarrow{\text{press}} \cdots$
- $\xRightarrow{\text{press}} \langle \text{off}, x = 2.0 \rangle \xRightarrow{\text{press}} \cdots$
- $\xRightarrow{\text{press}} \langle \text{off}, x = 3.0 \rangle \xRightarrow{\text{press}} \cdots$
- $\xRightarrow{\text{press}} \langle \text{off}, x = 127 \rangle \xRightarrow{\text{press}} \cdots$

Remark

Remark 4.30. That a configuration $\langle \ell, [\nu] \rangle$ is reachable in $\mathcal{R}(\mathcal{A})$ represents the fact, that all $\langle \ell, \nu \rangle$ are reachable.

IAW: in $\mathcal{A}$, we can observe ν when location ℓ has **just been entered**.

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

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** $\mathcal{U}(\mathcal{A})$ — abstracts from uncountably many delay transitions, still infinite-state.
- ✔ **Lem. 4.20**: location reachability of $\mathcal{A}$ is **preserved** in $\mathcal{U}(\mathcal{A})$.
- ✔ **Def. 4.29**: **region automaton** $\mathcal{R}(\mathcal{A})$ — equivalent configurations collapse into regions
- ✘ **Lem. 4.32**: location reachability of $\mathcal{U}(\mathcal{A})$ is **preserved** in $\mathcal{R}(\mathcal{A})$.
- ✘ **Lem. 4.28**: $\mathcal{R}(\mathcal{A})$ is **finite**.

Region Automaton Properties

Lemma 4.32. [*Correctness*] For all locations ℓ of a given timed automaton $\mathcal{A}$ the following holds:

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

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 ν_2' with $\nu_1' \sim \nu_2'$ and $\langle \ell, \nu_2 \rangle \xRightarrow{\alpha} \langle \ell', \nu_2' \rangle$.

Lemma 4.26. [*Bisimulation*] $\cong$ 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** $\mathcal{U}(\mathcal{A})$ — abstracts from uncountably many delay transitions, still infinite-state.
- ✔ **Lem. 4.20**: location reachability of $\mathcal{A}$ is **preserved** in $\mathcal{U}(\mathcal{A})$.
- ✔ **Def. 4.29**: **region automaton** $\mathcal{R}(\mathcal{A})$ — equivalent configurations collapse into regions
- ✔ **Lem. 4.32**: location reachability of $\mathcal{U}(\mathcal{A})$ is **preserved** in $\mathcal{R}(\mathcal{A})$.
- ✘ **Lem. 4.28**: $\mathcal{R}(\mathcal{A})$ is **finite**.

The Number of Regions

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

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

is an **upper bound** on the **number of regions**.

Proof: [Olderog and Dierks, 2008]

Observations Regarding the Number of Regions

- 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**, not an **exact bound**.

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** $\mathcal{U}(\mathcal{A})$ — abstracts from uncountably many delay transitions, still infinite-state.
- ✔ **Lem. 4.20**: location reachability of $\mathcal{A}$ is **preserved** in $\mathcal{U}(\mathcal{A})$.
- ✔ **Def. 4.29**: **region automaton** $\mathcal{R}(\mathcal{A})$ — equivalent configurations collapse into regions
- ✔ **Lem. 4.32**: location reachability of $\mathcal{U}(\mathcal{A})$ is **preserved** in $\mathcal{R}(\mathcal{A})$.
- ✔ **Lem. 4.28**: $\mathcal{R}(\mathcal{A})$ is **finite**.

Putting It All Together

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

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

$$\langle \ell_{ini}, [\nu_{ini}] \rangle \xrightarrow{\alpha}_{\mathcal{R}(\mathcal{A})} \langle \ell_1, [\nu_1] \rangle \xrightarrow{\alpha}_{\mathcal{R}(\mathcal{A})} \dots \xrightarrow{\alpha}_{\mathcal{R}(\mathcal{A})} \langle \ell_n, [\nu_n] \rangle$$

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

Putting It All Together

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

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

$$\langle \ell_{ini}, [\nu_{ini}] \rangle \xrightarrow{\alpha}_{\mathcal{R}(\mathcal{A})} \langle \ell_1, [\nu_1] \rangle \xrightarrow{\alpha}_{\mathcal{R}(\mathcal{A})} \dots \xrightarrow{\alpha}_{\mathcal{R}(\mathcal{A})} \langle \ell_n, [\nu_n] \rangle$$

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

So we have

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

The Constraint Reachability Problem

- **Given:** A timed automaton $\mathcal{A}$, one of its control locations ℓ, and a clock constraint φ.
- **Question:** Is a configuration $\langle \ell, \nu \rangle$ **reachable** where $\nu \models \varphi$, i.e. is there a transition sequence of the form

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

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

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

The Constraint Reachability Problem

- **Given:** A timed automaton $\mathcal{A}$, one of its control locations ℓ, and a clock constraint φ.
- **Question:** Is a configuration $\langle \ell, \nu \rangle$ **reachable** where $\nu \models \varphi$, i.e. is there a transition sequence of the form

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

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

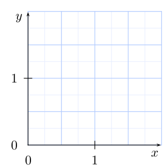
- **Note:** we just observed that $\mathcal{R}(\mathcal{A})$ loses some information about the clock valuations that are possible in/from a region.

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

The Delay Operation

- Let $[\nu]$ be a clock region.
- We set

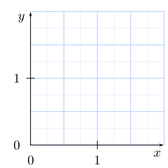
$$delay[\nu] = \{\nu' + t \mid \nu' \cong \nu \text{ and } t \in \mathsf{Time}\}.$$



The Delay Operation

- Let $[\nu]$ be a clock region.
- We set

$$delay[\nu] = \{\nu' + t \mid \nu' \cong \nu \text{ and } t \in \mathsf{Time}\}.$$



- **Note**: $delay[\nu]$ can be represented as a **finite** union of regions.
  For example, with our two-clock example we have

$delay[x = y = 0] =$

References

References

[Olderog and Dierks, 2008] Olderog, E.-R. and Dierks, H. (2008). *Real-Time Systems - Formal Specification and Automatic Verification*. Cambridge University Press.