Real-Time Systems

Lecture 13: Regions and Zones

2013-06-18

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:

- Started location reachability decidability (by region construction)

This Lecture:

- **Educational Objectives:** Capabilities for following tasks/questions.
  - 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?
  - What's a zone? In contrast to a region?
  - Motivation for having zones?
  - What's a DBM? Who needs to know DBMs?
- **Content:**
  - Region automaton cont'd
  - Reachability Problems for Extended Timed Automata
  - Zones
  - Difference Bound Matrices

The Location Reachability Problem Cont'd

The Region Automaton

one region

Definition 4.29. [*Region Automaton*] The **region automaton** $\mathcal{R}(\mathcal{A})$ of the timed automaton $\mathcal{A}$ is the labelled transition system

$$\mathcal{R}(\mathcal{A}) = (Conf(\mathcal{R}(\mathcal{A})), B_{?!}, \{\xrightarrow{\alpha}_{\mathcal{R}(\mathcal{A})} \mid \alpha \in B_{?!}\}, C_{ini})$$

representative of $[\nu]$

where

- $Conf(\mathcal{R}(\mathcal{A})) = \{\langle \ell, [\nu] \rangle \mid \ell \in L, \nu : X \to \mathsf{Time}, \nu \models I(\ell)\}$,
- for each $\alpha \in B_{?!}$,

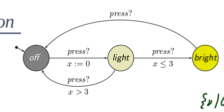
  representative of $[\nu']$

  $$\langle \ell, [\nu] \rangle \xrightarrow{\alpha}_{\mathcal{R}(\mathcal{A})} \langle \ell', [\nu'] \rangle \text{ if and only if } \langle \ell, \nu \rangle \xRightarrow{\alpha} \langle \ell', \nu' \rangle$$

  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



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

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

$\langle \text{bright}, [x = 0] \rangle \xRightarrow{\text{press}} \cdots$ $\{\nu \mid 0 < \nu(x) < 1\}$

$\langle \text{bright}, [x = 0.1] \rangle \xRightarrow{\text{press}} \cdots$ no "..." in between

$\langle \text{bright}, [x = 1.0] \rangle \xRightarrow{\text{press}} \cdots$

$\cdots$

$\langle \text{bright}, [x = 3.0] \rangle \xRightarrow{\text{press}} \cdots$

$\langle \text{bright}, [x = 3.001] \rangle \xRightarrow{\text{press}} \cdots$ $\{\nu \mid \nu(x) > 3\}$

$\langle \text{off}, [x = 0] \rangle \xRightarrow{\text{press}} \cdots$

$\cdots$

$\langle \text{off}, [x = 2.9] \rangle \xRightarrow{\text{press}} \cdots$

$\langle \text{off}, [x = 3.0] \rangle \xRightarrow{\text{press}} \cdots$

$\langle \text{off}, [x = 3.001] \rangle \xRightarrow{\text{press}} \cdots$

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**:

$\langle \ell, \nu_1 \rangle \xRightarrow{\alpha} \langle \ell', \nu_1' \rangle$

$\exists \nu_2' \bullet \langle \ell, \nu_2 \rangle \xRightarrow{\alpha} \langle \ell', \nu_2' \rangle$

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 $\langle \ell, \nu_2 \rangle \xRightarrow{\alpha} \langle \ell', \nu_2' \rangle$.

region equivalence

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

magnitude of X
(number of elements in X)

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]

↳ $|Conf(\mathcal{R}(A))| \leq |L| \cdot (2c+2)^{|X|} \cdot (4c+3)^{\frac{1}{2}|X|(|X|-1)}$

Observations Regarding the Number of Regions

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

  ↳ thus $\mathcal{R}(A)$ is finite

- Note: the upper bound is a **worst case**, not an **exact bound**.

  e.g. if $c_x < c_y$, 4.28 still works with $c = \max\{c_x, c_y\}$

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

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

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

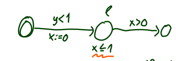
$$\langle \ell_{ini}, \nu_{ini} \rangle \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle \xrightarrow{\lambda_3} \dots \xrightarrow{\lambda_n} \langle \ell_n, \nu_n \rangle = \langle \ell, \nu \rangle$$

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.

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

The Delay Operation

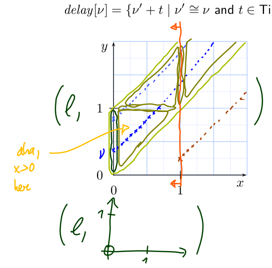


- Let $[\nu]$ be a clock region.
- We set

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

$\varphi \equiv x > 0$ in ℓ? YES

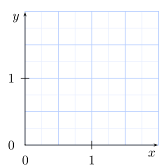
$\varphi' \equiv x > 5$ in ℓ? NO



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

$delay[x = y = 0] =$

Zones

(Presentation following [Fränzle, 2007])

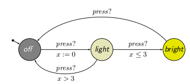
Recall: Number of Regions

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**.

- In the desk lamp controller,



many ~~all~~ regions are reachable in $\mathcal{R}(\mathcal{L})$, but we convinced ourselves that it's **actually** only important whether $\nu(x) \in [0,3]$ or $\nu(x) \in (3, \infty)$.

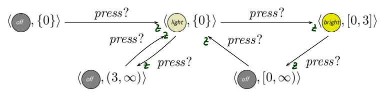
So: seems there are even **equivalence classes** of undistinguishable regions.

Wanted: Zones instead of Regions

- In $\mathcal{R}(\mathcal{L})$ we have transitions: region automaton
  - $\langle \text{light}, \{0\}\rangle \xrightarrow{press?}_{\mathcal{R}} \langle \text{bright}, \{0\}\rangle$, $\langle \text{light}, \{0\}\rangle \xrightarrow{press?}_{\mathcal{R}} \langle \text{bright}, (0,1)\rangle$,
  - $\langle l, \{0\}\rangle \xrightarrow{pr?}_{\mathcal{R}} \langle b, \{1\}\rangle$, $\langle l, \{0\}\rangle \xrightarrow{pr?}_{\mathcal{R}} \langle b, (1,2)\rangle$, $\langle l, \{0\}\rangle \xrightarrow{pr?}_{\mathcal{R}} \langle b, \{2\}\rangle$
  - $\langle \text{light}, \{0\}\rangle \xrightarrow{press?}_{\mathcal{R}} \langle \text{bright}, (2,3)\rangle$, $\langle \text{light}, \{0\}\rangle \xrightarrow{press?}_{\mathcal{R}} \langle \text{bright}, \{3\}\rangle$
- Which seems to be a complicated way to write just:

$$\langle \text{light}, \{0\}\rangle \xrightarrow{press?} \langle \text{bright}, [0,3]\rangle$$

- Can't we **constructively** abstract $\mathcal{L}$ to:

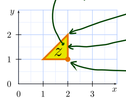


What is a Zone?

Definition. A **(clock) zone** is a set $z \subseteq (X \to \text{Time})$ of valuations of clocks X such that there exists $\varphi \in \Phi(X)$ with

$$\nu \in z \text{ if and only if } \nu \models \varphi.$$

Example:



is a clock zone by

$$\varphi = (x \leq 2) \wedge (x > 1) \wedge (y \geq 1) \wedge (y < 2) \wedge (x - y \geq 0)$$

What is a Zone?

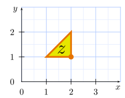
Definition. A **(clock) zone** is a set $z \subseteq (X \to \text{Time})$ of valuations of clocks X such that there exists $\varphi \in \Phi(X)$ with

$$\nu \in z \text{ if and only if } \nu \models \varphi.$$

valuations of X

simple clock constraints (for simplicity $c \in \mathbb{N}_0$)

Example:

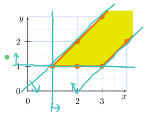


is a clock zone by

$$\varphi = (x \leq 2) \wedge (x > 1) \wedge (y \geq 1) \wedge (y < 2) \wedge (x - y \geq 0)$$

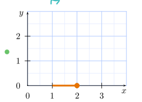
- Note: Each clock constraint φ is a **symbolic representation** of a zone.
- But: There's no one-on-one correspondence between clock constraints and zones. The zone $z = \emptyset$ corresponds to $(x > 1 \wedge x < 1)$, $(x > 2 \wedge x < 2)$, ...

More Examples: Zone or Not?

- 

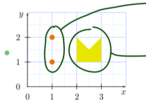
YES by

$(x \geq 1) \wedge (y \geq 1) \wedge (x - y \geq 0) \wedge (x - y \leq 2)$

- 

YES by

$(x > 1) \wedge (x \leq 2) \wedge (y = 0)$

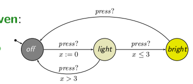
- 

$(x = y = 1) \vee (x = 1 \wedge y = 2)$ ↝ NO — not in simple clock constraints

NO

Zone-based Reachability

Given:

- 

and initial configuration $\langle \text{off}, \{0\}\rangle$

Assume a function

$$\text{Post}_e : (L \times \text{Zones}) \to (L \times \text{Zones})$$

an edge of the automaton

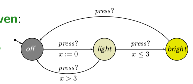
such that $\text{Post}_e(\langle \ell, z\rangle)$ yields the configuration $\langle \ell', z'\rangle$ such that

- zone z' denotes exactly those clock valuations ν'
- which are reachable from a configuration $\langle \ell, \nu\rangle$, $\nu \in z$,
- by taking edge $e = (\ell, \alpha, \varphi, Y, \ell') \in E$.

firstly delaying

Zone-based Reachability

Given:

- 

and initial configuration $\langle \text{off}, \{0\}\rangle$

Assume a function

$$\text{Post}_e : (L \times \text{Zones}) \to (L \times \text{Zones})$$

an edge of the automaton

such that $\text{Post}_e(\langle \ell, z\rangle)$ yields the configuration $\langle \ell', z'\rangle$ such that

- zone z' denotes exactly those clock valuations ν'
- which are reachable from a configuration $\langle \ell, \nu\rangle$, $\nu \in z$,
- by $\underbrace{\text{taking edge } e = (\ell, \alpha, \varphi, Y, \ell') \in E.}_{\text{firstly delaying}}$

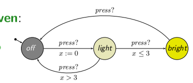
Then $\ell \in L$ is reachable in $\mathcal{A}$ if and only if

$$\text{Post}_{e_n}(\ldots(\text{Post}_{e_1}(\langle \ell_{ini}, z_{ini}\rangle))\ldots)$$

for some $e_1, \ldots, e_n \in E$.

Zone-based Reachability: In Other Words

Given:

-  and initial configuration $\langle$off$, \{0\}\rangle$

Wanted: A procedure to compute the set

- $\langle$light$, \{0\}\rangle$
- $\langle$bright$, [0, 3]\rangle$
- $\langle$off$, [0, \infty)\rangle$

already reached configuration

- Set $R := \{\langle \ell_{ini}, z_{ini} \rangle\} \subset L \times$ Zones   if $z_{ini} \models I(\ell_{ini})$
- Repeat
  - pick
    - a pair $\langle \ell, z \rangle$ from R and
    - an edge $e \in E$ with source ℓ

    such that $\mathrm{Post}_e(\langle \ell, z \rangle)$ is not already subsumed by R
  - add $\mathrm{Post}_e(\langle \ell, z \rangle)$ to R

  until no more such $\langle \ell, z \rangle \in R$ and $e \in E$ are found.

Stocktaking: What's Missing?

- Set $R := \{\langle \ell_{ini}, z_{ini} \rangle\} \subset L \times$ Zones
- Repeat
  - pick
    - a pair $\langle \ell, z \rangle$ from R and
    - an edge $e \in E$ with source ℓ

    such that $\mathrm{Post}_e(\langle \ell, z \rangle)$ is not already **subsumed** by R
  - add $\mathrm{Post}_e(\langle \ell, z \rangle)$ to R

  until no more such $\langle \ell, z \rangle \in R$ and $e \in E$ are found.

Missing:

- Algorithm to effectively compute $\mathrm{Post}_e(\langle \ell, z \rangle)$ for given configuration $\langle \ell, z \rangle \in L \times$ Zones and edge $e \in E$.
- Decision procedure for whether configuration $\langle \ell', z' \rangle$ is **subsumed** by a given subset of $L \times$ Zones.

Note: Algorithm in general **terminates only if** we apply **widening** to zones, that is, roughly, to take maximal constants c_x into account (not in lecture).

What is a Good "Post"?

- If z is given by a constraint $\varphi \in \Phi(X)$, then the zone component z' of $\mathrm{Post}_e(\ell, z) = \langle \ell', z' \rangle$ should also be a constraint from $\Phi(X)$. (Because sets of clock valuations are soo unhandily...)

Good news: the following operations can be carried out by manipulating φ.

- The **elapse time** operation:

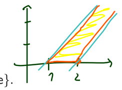
$$\uparrow : \Phi(X) \to \Phi(X)$$

Given a constraint φ, the constraint $\uparrow(\varphi)$, or $\varphi \uparrow$ in postfix notation, is supposed to denote the set of clock valuations

$$\{\nu + t \mid \nu \models \varphi, t \in \mathsf{Time}\}.$$

In other symbols: we **want**

$$[\![\uparrow(\varphi)]\!] = [\![\varphi \uparrow]\!] = \{\nu + t \mid \nu \in [\![\varphi]\!], t \in \mathsf{Time}\}.$$



To this end: remove all upper bounds $x \le c$, $x < c$ from φ and add diagonals.

Good News Cont'd

Good news: the following operations can be carried out by manipulating φ.

- **elapse time** $\varphi \uparrow$ with

$$[\![\varphi \uparrow]\!] = \{\nu + t \mid \nu \models \varphi, t \in \mathsf{Time}\}$$

- **zone intersection** $\varphi_1 \wedge \varphi_2$ with

$$[\![\varphi_1 \wedge \varphi_2]\!] = \{\nu \mid \nu \models \varphi_1 \text{ and } \nu \models \varphi_2\}$$

- **clock hiding** $\exists x.\varphi$ with

$$[\![\exists x.\varphi]\!] = \{\nu \mid \text{there is } t \in \mathsf{Time} \text{ such that } \nu[x := t] \models \varphi\}$$

- **clock reset** $\varphi[x := 0]$ with

$$[\![\varphi[x := 0]]\!] = [\![x = 0 \wedge \exists x.\varphi]\!]$$

This is Good News...

...because given $\langle \ell, z \rangle = \langle \ell, \varphi_0 \rangle$ and $e = (\ell, \alpha, \varphi, \{y_1, \dots, y_n\}, \ell') \in E$ we have

$$\mathrm{Post}_e(\langle \ell, z \rangle) = \langle \ell', \varphi_5 \rangle$$

where

- $\varphi_1 = \varphi_0 \uparrow$

  let **time elapse** starting from φ_0: φ_1 represents all valuations reachable by waiting in ℓ for an arbitrary amount of time.
- $\varphi_2 = \varphi_1 \wedge I(\ell)$

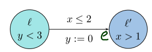
  intersect with invariant of ℓ: φ_2 represents the reachable "good" valuations.
- $\varphi_3 = \varphi_2 \wedge \varphi$

  intersect with guard: φ_3 are the reachable "good" valuations where e is enabled.
- $\varphi_4 = \varphi_3[y_1 := 0] \dots [y_n := 0]$

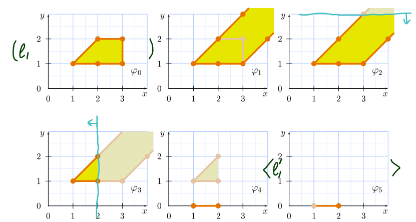
  reset clocks: φ_4 are all possible outcomes of taking e from φ_3
- $\varphi_5 = \varphi_4 \wedge I(\ell')$

  intersect with invariant of ℓ': φ_5 are the "good" outcomes of taking e from φ_3

Example



- $\varphi_1 = \varphi_0 \uparrow$ — let **time elapse**.
- $\varphi_2 = \varphi_1 \wedge I(\ell)$ — **intersect with invariant** of ℓ
- $\varphi_3 = \varphi_2 \wedge \varphi$ — **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \dots [y_n := 0]$ — **reset clocks**
- $\varphi_5 = \varphi_4 \wedge I(\ell')$ — **intersect with invariant** of ℓ'



Difference Bound Matrices

- Given a finite set of clocks X, a **DBM** over X is a mapping

disjoint union

$$M : (X \dot\cup \{x_0\} \times X \dot\cup \{x_0\}) \to (\{<, \leq\} \times \mathbb{Z} \cup \{(<, \infty)\})$$

- $M(x, y) = (\sim, c)$ encodes the conjunct $x - y \sim c$ (x and y can be x_0).

M:

	x_0	x	y
x_0			
x			
y			

$\in \{<, \leq\} \times \mathbb{Z} \cup \{(<, \infty)\}$

Difference Bound Matrices

- Given a finite set of clocks X, a **DBM** over X is a mapping

$$M : (X \dot\cup \{x_0\} \times X \dot\cup \{x_0\}) \to (\{<, \leq\} \times \mathbb{Z} \cup \{(<, \infty)\})$$

- $M(x, y) = (\sim, c)$ encodes the conjunct $x - y \sim c$ (x and y can be x_0).

- If M and N are DBM encoding φ_1 and φ_2 (representing zones z_1 and z_2), then we can efficiently compute $M \uparrow$, $M \wedge N$, $M[x := 0]$ such that
  - all three are again DBM,
  - $M \uparrow$ encodes $\varphi_1 \uparrow$,
  - $M \wedge N$ encodes $\varphi_1 \wedge \varphi_2$, and
  - $M[x := 0]$ encodes $\varphi_1[x := 0]$.
- And there is a **canonical form** of DBM — canonisation of DBM can be done in cubic time (**Floyd-Warshall** algorithm).

- Thus: we can define our 'Post' on DBM, and let our algorithm run on DBM.

Pros and cons

- **Zone-based** reachability analysis usually is explicit wrt. discrete locations:
  - maintains a list of location/zone pairs or
  - maintains a list of location/DBM pairs
  - **confined wrt. size of discrete state space**
  - **avoids blowup by number of clocks and size of clock constraints through symbolic representation of clocks**

- **Region-based** analysis provides a finite-state abstraction, amenable to finite-state symbolic MC
  - **less dependent on size of discrete state space**
  - **exponential in number of clocks**

References

References

[Fränzle, 2007] Fränzle, M. (2007). Formale methoden eingebetteter systeme. Lecture, Summer Semester 2007, Carl-von-Ossietzky Universität Oldenburg.

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