Real-Time Systems

Lecture 14: Regions and Zones

2017-12-21

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Content

Zones

(Presentation following Fränzle (2007))

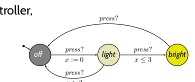
Recall: Number of Regions

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 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 $\nu(x) \in (3,\infty)$.

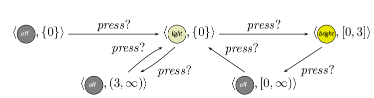
  So: it seems like there are even **equivalence classes** of **undistinguishable regions** in certain timed automata.

Wanted: Zones instead of Regions

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

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

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



Content

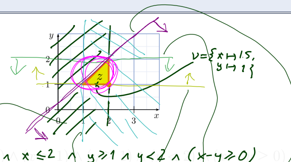
- **Motivation**:
  Sometimes, regions seem too fine-grained
- **Definition**
  - **Examples**: Zone or Not Zone
- **Zone**-based **Reachability Analysis**
  - The **basic algorithm**.
  - Building blocks:
    - **Post**-operator,
    - **subsumption check**
  - A **symbolic Post**-operator
- **Difference-Bounds-Matrices** (DBMs)
- **Discussion**: **Zones** vs. **Regions**

What is a Zone?

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

$\nu \in z$ if and only if $\nu \models \varphi$.

Example:



is a clock zone by

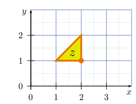
$\varphi = x > 1 \wedge x \leq 2 \wedge y \geq 1 \wedge y < 2 \wedge (x - y \geq 0)$

What is a Zone?

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

$\nu \in z$ if and only if $\nu \models \varphi$.

Example:



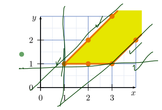
is a clock zone by

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

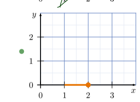
- Note: Each clock constraint φ is a **symbolic representation** of a zone.
- But: There's no one-on-one correspondence between clock constraints and zones. The zone $z = \emptyset$ corresponds to $(x > 1 \wedge x < 1)$, $(x > 2 \wedge x < 2)$, …

More Examples: Zone or Not?

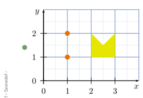
z is a zone iff there is $\varphi \in \Phi(X)$ s.t. $z = \{\nu \mid \nu \models \varphi\}$.

- 

YES
$x \geq 1 \wedge x - y \geq 0 \wedge x - y \leq 2 \wedge y \geq 1$

- 

YES
$y \geq 0 \wedge y \leq 0 \wedge x > 1 \wedge x \leq 2$ $(\sim (1,2) \cup \{2\})$

- 

NO
(not convex)

z is zone
$\Rightarrow \exists r_1, \dots, r_n$ regions $\cdot$ $z = \bigcup_{i=1}^{n} r_i$

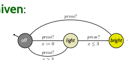
$z = \bigcup_{i=1}^{n} r_i$, r_i region
$\not\Rightarrow$ z is zone

Content

- **Motivation**:
  Sometimes, regions seem too fine-grained
- **Definition**
  - **Examples**: Zone or Not Zone
- **Zone**-based **Reachability Analysis**
  - The **basic algorithm**.
  - Building blocks:
    - **Post**-operator,
    - **subsumption check**
  - A **symbolic Post**-operator
- **Difference-Bounds-Matrices** (DBMs)
- **Discussion**: **Zones** vs. **Regions**

Zone-based Reachability Analysis

Given:

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

Assume a function

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

such that $\mathrm{Post}_e(\langle \ell, z \rangle)$ 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 $e = (\ell, \alpha, \varphi, Y, \ell') \in E$.

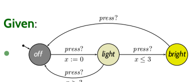
Then $\ell \in L$ is reachable in $\mathcal{A}$ if and only if

$$\mathrm{Post}_{e_n}(\dots(\mathrm{Post}_{e_1}(\langle \ell_{ini}, z_{ini} \rangle))\dots) = \langle \ell, z \rangle$$

for some $e_1, \dots, e_n \in E$ and some z.

Zone-based Reachability: In Other Words

Given:

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

$\langle \ell_1, \{0\} \rangle$
- $\mathrm{Post}_{e_1}(\langle \ell, \{0\} \rangle)$: $\langle \ell_2, [0,\infty) \rangle$
- $\mathrm{Post}_{e_2}(\langle \ell, \{0\} \rangle)$: $\langle \ell_2, \underbrace{\{0\}}_{z_2} \rangle$

subsumption example

Wanted: A procedure to compute the set

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

$X = \{x, y\}$

$x \geq 1$, $x := 0$, $x \leq 1$

- $\langle \ell, \{(0,0)\} \rangle$
- $\langle \ell, \{(0,1)\} \rangle$
- $\langle \ell, \{(0,2)\} \rangle$

does not terminate

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

Note: The 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"?

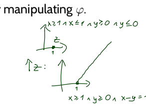
- If z is given by a constraint $\varphi \in \Phi(X)$, (write: $z = [\![\varphi]\!]$) then the zone component z' of $\mathrm{Post}_e(\ell, z) = \langle \ell', z' \rangle$ should also be a constraint from $\Phi(X)$.

  (We want to **manipulate constraints**, not those unhandy sets of clock valuations.)

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

(1) The **elapse time** operation:

$$\begin{aligned} \uparrow \;:\; & \text{Zones} \to \text{Zones} \\ & z \mapsto \{\nu + t \mid t \in \text{Time}\} \end{aligned}$$



can be carried out **symbolically** as follows:

- Let $z = [\![\varphi]\!]$.
- Obtain φ' by removing all upper bounds $x \leq c$, $x < c$, from φ and adding diagonals.
- Then $[\![\varphi']\!] = z\uparrow$.

This procedure defines $\uparrow : \Phi(X) \to \Phi(X)$ (a function on **clock constraints!**), such that $[\![\varphi\uparrow]\!] = z\uparrow$ if $z = [\![\varphi]\!]$.

Good News Cont'd

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

(1) **elapse time:** $\varphi\uparrow$ with $[\![\varphi\uparrow]\!] = z\uparrow$ if $z = [\![\varphi]\!]$.

(2) **zone intersection:** if $z_1 = [\![\varphi_1]\!]$ and $z_2 = [\![\varphi_2]\!]$, then $[\![\varphi_1 \wedge \varphi_2]\!] = z_1 \cap z_2$.

(3) **clock reset:**

$$\begin{aligned} \cdot[\,\cdot := 0] \;:\; & \text{Zones} \times X \to \text{Zones} \\ & (z, x) \mapsto \{\nu[x := 0] \mid \nu \in z\} \end{aligned}$$

can be carried out **symbolically** by setting

$$\begin{aligned} \cdot[\,\cdot := 0] \;:\; & \Phi \times X \to \Phi \\ & (\varphi, x) \mapsto \underline{(x = 0) \wedge (\exists\, x.\varphi)} \end{aligned}$$

$x = y \wedge x = 2$

$x = 0 \wedge x = y \wedge x = 2$ ☹

$x = 0 \wedge (\exists \tilde{x} \cdot \tilde{x} = y \wedge \tilde{x} = 2)$

using **clock hiding** (existential quantification):

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

This is Good News...

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

$$\mathrm{Post}_e(\langle \ell, z \rangle) = \langle \ell', [\![\varphi_5]\!] \rangle \qquad (\textbf{symbolical: } \mathrm{Post}_e(\langle \ell, \varphi_0 \rangle) = \langle \ell', \varphi_5 \rangle)$$

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 "good" valuations reachable from φ_1.
- $\varphi_3 = \varphi_2 \wedge \varphi$

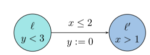
  intersect with guard: in φ_3 are the reachable "good" valuations where e is enabled.
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$

  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.

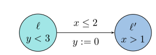
Example

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



Example

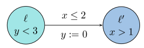
- $\varphi_1 = \varphi_0\uparrow$ — **let time elapse.**
- $\varphi_2 = \varphi_1 \wedge I(\ell)$ — **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \wedge \varphi$ — **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$ — **reset clocks**
- $\varphi_5 = \varphi_4 \wedge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



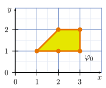
$$\begin{aligned} \varphi_0 &= 1 \leq y \leq 2 \\ &\wedge 1 \leq x \leq 3 \wedge x \geq y \end{aligned}$$

Example

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

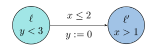


$\varphi_0 = 1 \le y \le 2 \land 1 \le x \le 3 \land x \ge y$

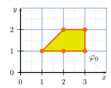


Example

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



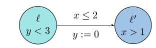
$\varphi_0 = 1 \le y \le 2 \land 1 \le x \le 3 \land x \ge y$



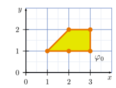
$\varphi_1 = 1 \le y \land 1 \le x \land x \ge y \land x \le y + 2$

Example

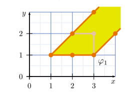
- $\varphi_1 = \varphi_0 \uparrow$ **let time elapse.**
- $\varphi_2 = \varphi_1 \land I(\ell)$ **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \land \varphi$ **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \dots [y_n := 0]$ **reset clocks**
- $\varphi_5 = \varphi_4 \land I(\ell')$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



$\varphi_0 = 1 \le y \le 2 \land 1 \le x \le 3 \land x \ge y$

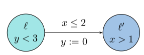


$\varphi_1 = 1 \le y \land 1 \le x \land x \ge y \land x \le y + 2$

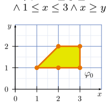


Example

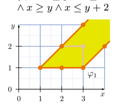
- $\varphi_1 = \varphi_0 \uparrow$ **let time elapse.**
- $\varphi_2 = \varphi_1 \land I(\ell)$ **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \land \varphi$ **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \dots [y_n := 0]$ **reset clocks**
- $\varphi_5 = \varphi_4 \land I(\ell')$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



$\varphi_0 = 1 \le y \le 2 \land 1 \le x \le 3 \land x \ge y$



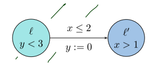
$\varphi_1 = 1 \le y \land 1 \le x \land x \ge y \land x \le y + 2$



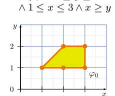
$\varphi_2 = 1 \le y < 3 \land 1 \le x \land x \ge y \land x \le y + 2$

Example

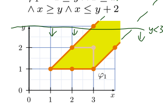
- $\varphi_1 = \varphi_0 \uparrow$ **let time elapse.**
- $\varphi_2 = \varphi_1 \land I(\ell)$ **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \land \varphi$ **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \dots [y_n := 0]$ **reset clocks**
- $\varphi_5 = \varphi_4 \land I(\ell')$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



$\varphi_0 = 1 \le y \le 2 \land 1 \le x \le 3 \land x \ge y$

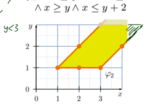


$\varphi_1 = 1 \le y \land 1 \le x \land x \ge y \land x \le y + 2$



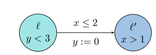
$y<3$

$\varphi_2 = 1 \le y < 3 \land 1 \le x \land x \ge y \land x \le y + 2$

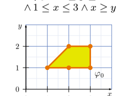


Example

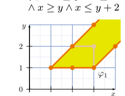
- $\varphi_1 = \varphi_0 \uparrow$ **let time elapse.**
- $\varphi_2 = \varphi_1 \land I(\ell)$ **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \land \varphi$ **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \dots [y_n := 0]$ **reset clocks**
- $\varphi_5 = \varphi_4 \land I(\ell')$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



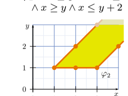
$\varphi_0 = 1 \le y \le 2 \land 1 \le x \le 3 \land x \ge y$



$\varphi_1 = 1 \le y \land 1 \le x \land x \ge y \land x \le y + 2$



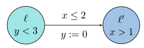
$\varphi_2 = 1 \le y < 3 \land 1 \le x \land x \ge y \land x \le y + 2$



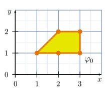
$\varphi_3 = 1 \le y < 3 \land 1 \le x \le 2 \land x \ge y \land x \le y + 2$

Example

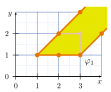
- $\varphi_1 = \varphi_0 \uparrow$ **let time elapse.**
- $\varphi_2 = \varphi_1 \wedge I(\ell)$ **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \wedge \varphi$ **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$ **reset clocks**
- $\varphi_5 = \varphi_4 \wedge I(\ell')$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



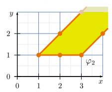
$\varphi_0 = 1 \le y \le 2$
$\wedge 1 \le x \le 3 \wedge x \ge y$



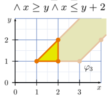
$\varphi_1 = 1 \le y \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



$\varphi_2 = 1 \le y < 3 \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$

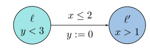


$\varphi_3 = 1 \le y < 3$
$\wedge 1 \le x \le 2$
$\wedge x \ge y \wedge x \le y + 2$

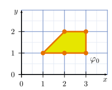


Example

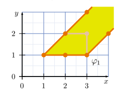
- $\varphi_1 = \varphi_0 \uparrow$ **let time elapse.**
- $\varphi_2 = \varphi_1 \wedge I(\ell)$ **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \wedge \varphi$ **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$ **reset clocks**
- $\varphi_5 = \varphi_4 \wedge I(\ell')$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



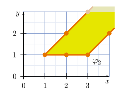
$\varphi_0 \equiv 1 \le y \le 2$
$\wedge 1 \le x \le 3 \wedge x \ge y$



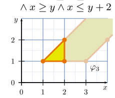
$\varphi_1 \equiv 1 \le y \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



$\varphi_2 \equiv 1 \le y < 3 \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



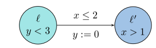
$\varphi_3 \equiv 1 \le y < 3$
$\wedge 1 \le x \le 2$
$\wedge x \ge y \wedge x \le y + 2$



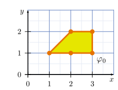
$\varphi_4 \equiv y = 0 \wedge$
$\exists y.1 \le y < 3 \wedge 1 \le x \le$
$2 \wedge x \ge y \wedge x \le y + 2$

Example

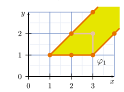
- $\varphi_1 = \varphi_0 \uparrow$ **let time elapse.**
- $\varphi_2 = \varphi_1 \wedge I(\ell)$ **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \wedge \varphi$ **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$ **reset clocks**
- $\varphi_5 = \varphi_4 \wedge I(\ell')$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



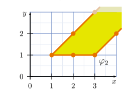
$\varphi_0 = 1 \le y \le 2$
$\wedge 1 \le x \le 3 \wedge x \ge y$



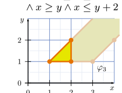
$\varphi_1 = 1 \le y \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



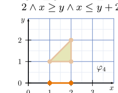
$\varphi_2 = 1 \le y < 3 \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



$\varphi_3 = 1 \le y < 3$
$\wedge 1 \le x \le 2$
$\wedge x \ge y \wedge x \le y + 2$

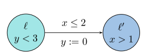


$\varphi_4 = y = 0 \wedge$
$\exists y.1 \le y < 3 \wedge 1 \le x \le$
$2 \wedge x \ge y \wedge x \le y + 2$

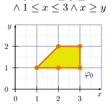


Example

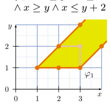
- $\varphi_1 = \varphi_0 \uparrow$ **let time elapse.**
- $\varphi_2 = \varphi_1 \wedge I(\ell)$ **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \wedge \varphi$ **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$ **reset clocks**
- $\varphi_5 = \varphi_4 \wedge I(\ell')$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



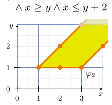
$\varphi_0 = 1 \le y \le 2$
$\wedge 1 \le x \le 3 \wedge x \ge y$



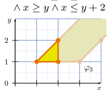
$\varphi_1 = 1 \le y \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



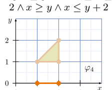
$\varphi_2 = 1 \le y < 3 \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



$\varphi_3 = 1 \le y < 3$
$\wedge 1 \le x \le 2$
$\wedge x \ge y \wedge x \le y + 2$



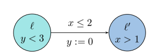
$\varphi_4 = y = 0 \wedge$
$\exists y.1 \le y < 3 \wedge 1 \le x \le$
$2 \wedge x \ge y \wedge x \le y + 2$



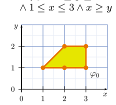
$\varphi_5 \equiv x > 1 \wedge y = 0 \wedge$
$\exists y.1 \le y < 3 \wedge 1 \le x \le$
$2 \wedge x \ge y \wedge x \le y + 2$

Example

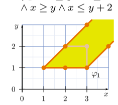
- $\varphi_1 = \varphi_0 \uparrow$ **let time elapse.**
- $\varphi_2 = \varphi_1 \wedge I(\ell)$ **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \wedge \varphi$ **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$ **reset clocks**
- $\varphi_5 = \varphi_4 \wedge I(\ell')$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



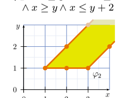
$\varphi_0 \equiv 1 \le y \le 2$
$\wedge 1 \le x \le 3 \wedge x \ge y$



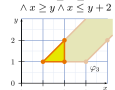
$\varphi_1 \equiv 1 \le y \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



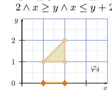
$\varphi_2 \equiv 1 \le y < 3 \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



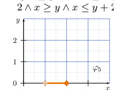
$\varphi_3 \equiv 1 \le y < 3$
$\wedge 1 \le x \le 2$
$\wedge x \ge y \wedge x \le y + 2$



$\varphi_4 \equiv y = 0 \wedge$
$\exists y.1 \le y < 3 \wedge 1 \le x \le$
$2 \wedge x \ge y \wedge x \le y + 2$

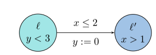


$\varphi_5 \equiv x > 1 \wedge y = 0 \wedge$
$\exists y.1 \le y < 3 \wedge 1 \le x \le$
$2 \wedge x \ge y \wedge x \le y + 2$

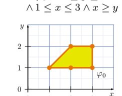


Example

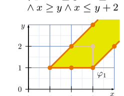
- $\varphi_1 = \varphi_0 \uparrow$ **let time elapse.**
- $\varphi_2 = \varphi_1 \wedge I(\ell)$ **intersect with invariant of ℓ**
- $\varphi_3 = \varphi_2 \wedge \varphi$ **intersect with guard**
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$ **reset clocks**
- $\varphi_5 = \varphi_4 \wedge I(\ell')$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



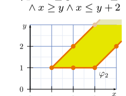
$\varphi_0 \equiv 1 \le y \le 2$
$\wedge 1 \le x \le 3 \wedge x \ge y$



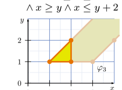
$\varphi_1 \equiv 1 \le y \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



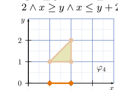
$\varphi_2 \equiv 1 \le y < 3 \wedge 1 \le x$
$\wedge x \ge y \wedge x \le y + 2$



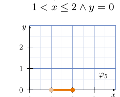
$\varphi_3 \equiv 1 \le y < 3$
$\wedge 1 \le x \le 2$
$\wedge x \ge y \wedge x \le y + 2$



$\varphi_4 \equiv y = 0 \wedge$
$\exists y.1 \le y < 3 \wedge 1 \le x \le$
$2 \wedge x \ge y \wedge x \le y + 2$



$\varphi_5 \iff$
$1 < x \le 2 \wedge y = 0$



Content

- **Motivation**:
  Sometimes, regions seem too fine-grained
- **Definition**
  - **Examples**: Zone or Not Zone
- **Zone**-based **Reachability Analysis**
  - The **basic algorithm**.
  - Building blocks:
    - **Post**-operator,
    - **subsumption check**
  - A **symbolic Post**-operator
- **Difference-Bounds-Matrices** (DBMs)
- **Discussion**: **Zones** vs. **Regions**

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).

$$\begin{array}{c|ccc} & x_0 & \underbrace{x \quad\quad y}_{=X} \\ \hline x_0 & & & (\leq, 5) \\ x & & & (<, 27) \\ y & & & \end{array}$$

$M(x_0, y) = x_0 - y \leq -5$
$= -y \leq -5$
$= y \geq 5$

$M(x, y) = (<, 27)$

$x - y < 27$

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 **DBMs 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.

Content

- **Motivation**:
  Sometimes, regions seem too fine-grained
- **Definition**
  - **Examples**: Zone or Not Zone
- **Zone**-based **Reachability Analysis**
  - The **basic algorithm**.
  - Building blocks:
    - **Post**-operator,
    - **subsumption check**
  - A **symbolic Post**-operator
- **Difference-Bounds-Matrices** (DBMs)
- **Discussion**: **Zones** vs. **Regions**

Pros and cons

- **Zone-based**
  reachability analysis usually is explicit wrt. discrete locations:
  - maintains a list of location/zone pairs (or 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 model-checking
  - **less dependent on size of discrete state space**
  - **exponential in number of clocks**

Content

- **Motivation**:
  Sometimes, regions seem too fine-grained
- **Definition**
  - **Examples**: Zone or Not Zone
- **Zone**-based **Reachability Analysis**
  - The **basic algorithm**.
  - Building blocks:
    - **Post**-operator,
    - **subsumption check**
  - A **symbolic Post**-operator
- **Difference-Bounds-Matrices** (DBMs)
- **Discussion**: **Zones** vs. **Regions**

Tell Them What You've Told Them...

- A **zone** is **a set of clock valuations**
  which can be characterised by a **clock constraint**.
- Each **zone** is a union of **regions**,
  not every union of **regions** is a **zone**.
- There is an **effectively computable**
  Post-operation for TA edges on **zones**.
  - based on: **time elapse**, **intersection**, **reset**
  - so there is a **fully symbolic**
    decision procedure for location reachability
    (if we ensure **termination** by **widening**)
  - even more convenient: using DBMs
    - since DBMs have a **normal form**
- For a **given model**, sometimes the **region**-based /
  sometimes the **zone**-based approach is faster.
  Not so many region-based tools are "on the market" these days.

References

References

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

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