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
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

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

is an upper bound on the number of regions.

• In the desk lamp controller,

many regions are reachable in $R(L)$, 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:
  - $\langle \text{light}, \{0\} \rangle \xrightarrow{\text{press}} \langle \text{light}, \{0\} \rangle$, $\langle \text{light}, \{0\} \rangle \xrightarrow{\text{press}} \langle \text{light}, (0, 1) \rangle$
  - ..., $\langle \text{light}, \{0\} \rangle \xrightarrow{\text{press}} \langle \text{light}, (2, 3) \rangle$, $\langle \text{light}, \{0\} \rangle \xrightarrow{\text{press}} \langle \text{light}, \{3\} \rangle$

- Which seems to be a complicated way to write just:
  $$\langle \text{light}, \{0\} \rangle \xrightarrow{\text{press}} \langle \text{bright}, [0, 3] \rangle$$

- Can’t we constructively abstract $\mathcal{L}$ to:
  $$\langle \text{off}, \{0\} \rangle \langle \text{light}, \{0\} \rangle \langle \text{bright}, [0, 3] \rangle \langle \text{off}, (3, \infty) \rangle \langle \text{off}, [0, \infty) \rangle$$

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 \( z \subseteq (X \rightarrow \text{Time}) \) 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

\[
\varphi = (x \leq 2) \land (x > 1) \land (y \geq 1) \land (y < 2) \land (x - y \geq 0)
\]

• Note: Each clock constraint \( \varphi \) 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 \((x > 1 \land x < 1), (x > 2 \land x < 2), \ldots\)
More Examples: Zone or Not?

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

### 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
Given:

- and initial configuration \( \langle \text{off}, \{0\} \rangle \)

Assume a function

\[ \text{Post}_e : (L \times \text{Zones}) \rightarrow (L \times \text{Zones}) \]

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 \( v' \)
  - which are reachable from a configuration \( \langle \ell , v \rangle , v \in z \),
  - by taking edge \( e = (\ell , \alpha , \phi , Y, \ell') \in E \).

Then \( \ell \in L \) is reachable in \( A \) if and only if

\[ \text{Post}_{e_1} (\ldots (\text{Post}_{e_1}(\langle \ell_{\text{ini}}, z_{\text{ini}} \rangle) \ldots)) = \langle \ell, z \rangle \]

for some \( e_1, \ldots , e_n \in E \) and some \( z \).

Zone-based Reachability: In Other Words

Given:

- and initial configuration \( \langle \text{off}, \{0\} \rangle \)

Wanted: A procedure to compute the set

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

Set \( R := \{ \langle \ell_{\text{ini}}, z_{\text{ini}} \rangle \} \subset L \times \text{Zones} \)

Repeat

- pick a pair \( \langle \ell , z \rangle \) from \( R \) and an edge \( e \in E \) with source \( \ell \)
  - such that \( \text{Post}_e (\langle \ell , z \rangle) \) is not already subsumed by \( R \)
  - add \( \text{Post}_e (\langle \ell , z \rangle) \) to \( R \)

until no more such \( \langle \ell , z \rangle \in R \) and \( e \in E \) are found.
Stocktaking: What’s Missing?

• Set \( R := \{(\ell_{\text{ini}}, z_{\text{ini}})\} \subset L \times \text{Zones} \)
• Repeat
  • pick a pair \( (\ell, z) \) from \( R \) and
  • an edge \( e \in E \) with source \( \ell \) such that \( \text{Post}_e((\ell, z)) \) is not already subsumed by \( R \)
  • add \( \text{Post}_e((\ell, z)) \) to \( R \)
until no more such \( (\ell, z) \in R \) and \( e \in E \) are found.

Missing:
• Algorithm to effectively compute \( \text{Post}_e((\ell, z)) \)
  for a given configuration \( (\ell, z) \in L \times \text{Zones} \) and an edge \( e \in E \).
• Decision procedure for whether configuration \( (\ell', z') \) is subsumed by a given subset of \( L \times \text{Zones} \).

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: \( z = \llbracket \varphi \rrbracket \))
then the zone component \( z' \) of \( \text{Post}_e(\ell, z) = (\ell', z') \)
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 \( \varphi \).

(1) The elapse time operation:

\[ \uparrow : \text{Zones} \rightarrow \text{Zones} \]
\[ z \mapsto \{ \nu + t \mid t \in \text{Time} \} \]

can be carried out symbolically as follows:
• Let \( z = \llbracket \varphi \rrbracket \).
• Obtain \( \varphi' \) by removing all upper bounds \( x \leq c, x < c \) from \( \varphi \) and adding diagonals.
• Then \( \llbracket \varphi' \rrbracket = z \uparrow \).
This procedure defines \( \uparrow : \Phi(X) \rightarrow \Phi(X) \) (a function on clock constraints!),
such that \( \llbracket \varphi \uparrow \rrbracket = z \uparrow \) if \( z = \llbracket \varphi \rrbracket \).
Good news: the following operations can be carried out by manipulating $\varphi$.

1. **elapse time**: $\varphi \uparrow$ with $\llbracket \varphi \uparrow \rrbracket = z \uparrow$ if $z = \llbracket \varphi \rrbracket$.

2. **zone intersection**: if $z_1 = \llbracket \varphi_1 \rrbracket$ and $z_2 = \llbracket \varphi_2 \rrbracket$, then $\llbracket \varphi_1 \land \varphi_2 \rrbracket = z_1 \cap z_2$.

3. **clock reset**:

   $\cdot \llbracket \cdot := 0 \rrbracket : \text{Zones} \times X \rightarrow \text{Zones}$

   
   
   \[
   (z, x) \mapsto \{ \nu | x := 0 \mid \nu \in z \}
   \]

   can be carried out symbolically by setting

   $\cdot \llbracket \cdot := 0 \rrbracket : \Phi \times X \rightarrow \Phi$

   
   
   \[
   (\varphi, x) \mapsto \left( \begin{array}{c}
   \begin{array}{c}
   x = y \land x = z
   
   x = 0 \land \exists x \cdot \nu \in \text{Time} \Rightarrow
   
   \exists x \cdot \nu \in \varphi
   
   \end{array}
   
   \end{array}
   \right)
   \]

   using clock hiding (existential quantification):

   $\llbracket \exists x \cdot \varphi \rrbracket = \{ \nu | \text{there is } t \in \text{Time such that } \nu[x := t] = \varphi \}$

This is Good News...

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

\[
\text{Post}_e(\langle \ell, z \rangle) = \langle \ell', \llbracket \varphi_5 \rrbracket \rangle \quad \text{(symbolical: Post}_e(\langle \ell, \varphi_0 \rangle) = \langle \ell', \varphi_5 \rangle)\]

where

- $\varphi_1 = \varphi_0 \uparrow$
  
  let time elapse starting from $\varphi_0$:
  
  $\varphi_1$ represents all valuations reachable by waiting in $\ell$ for an arbitrary amount of time.

- $\varphi_2 = \varphi_1 \land I(\ell)$
  
  intersect with invariant of $\ell$: $\varphi_2$ represents the “good” valuations reachable from $\varphi_1$.

- $\varphi_3 = \varphi_2 \land \varphi$
  
  intersect with guard: in $\varphi_3$ are the reachable “good” valuations where $e$ is enabled.

- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$
  
  reset clocks: $\varphi_4$ are all possible outcomes of taking $e$ from $\varphi_3$.

- $\varphi_5 = \varphi_4 \land I(\ell')$
  
  intersect with invariant of $\ell'$: $\varphi_5$ are the “good” outcomes of taking $e$ from $\varphi_3$. 
Example

- $\varphi_1 = \varphi_0 \uparrow$
- $\varphi_2 = \varphi_1 \land I(\ell)$ intersect with invariant of $\ell$
- $\varphi_3 = \varphi_2 \land \varphi$
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$ reset clocks
- $\varphi_5 = \varphi_4 \land I(\ell')$ intersect with invariant of $\ell'$

let time elapse.

$\ell$

$y < 3$

$y := 0$

$\ell'$

$x \leq 2$

$x > 1$

$\varphi_0 = 1 \leq y \leq 2$

$\land 1 \leq x \leq 3 \land x \geq y$
Example

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

\[ \varphi_0 = 1 \leq y \leq 2 \land 1 \leq x \leq 3 \land x \geq y \]
Example

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

\[
\begin{align*}
\varphi_0 &= 1 \leq y \leq 2 \\
&\land 1 \leq x \leq 3 \land x \geq y
\end{align*}
\]

\[
\begin{align*}
\varphi_1 &= 1 \leq y \land 1 \leq x \\
&\land x \geq y \land x \leq y + 2
\end{align*}
\]
Example

- $\varphi_1 = \varphi_0 \uparrow$
- $\varphi_2 = \varphi_1 \land I(\ell)$ intersect with invariant of $\ell$
- $\varphi_3 = \varphi_2 \land \varphi$
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$ reset clocks
- $\varphi_5 = \varphi_4 \land I(\ell')$ intersect with invariant of $\ell'$

let time elapse.

$\varphi_0 = 1 \leq y \leq 2 \land 1 \leq x \leq 3 \land x \geq y$

$\varphi_1 = 1 \leq y \leq 3 \land 1 \leq x$ $\land x \geq y \land x \leq y + 2$

$\varphi_2 = 1 \leq y < 3 \land 1 \leq x$ $\land x \geq y \land x \leq y + 2$
Example

- $\varphi_1 = \varphi_0 \uparrow$
- $\varphi_2 = \varphi_1 \land I(\ell)$ intersect with invariant of $\ell$
- $\varphi_3 = \varphi_2 \land \varphi$
- $\varphi_4 = \varphi_3 [y_1 := 0] \ldots [y_n := 0]$ reset clocks
- $\varphi_5 = \varphi_4 \land I(\ell')$ intersect with invariant of $\ell'$

\[
\begin{align*}
\varphi_0 &= 1 \leq y \leq 2 \\
&\quad \land 1 \leq x \leq 3 \land x \geq y
\end{align*}
\]

\[
\begin{align*}
\varphi_1 &= 1 \leq y \land 1 \leq x \\
&\quad \land x \geq y \land x \leq y + 2
\end{align*}
\]

\[
\begin{align*}
\varphi_2 &= 1 \leq y < 3 \land 1 \leq x \\
&\quad \land x \geq y \land x \leq y + 2
\end{align*}
\]
Example

- $\phi_1 = \phi_0 \uparrow$
- $\phi_2 = \phi_1 \land I(\ell)$ intersect with invariant of $\ell$
- $\phi_3 = \phi_2 \land \varphi$
- $\phi_4 = \phi_3[y_1 := 0] \ldots [y_n := 0]$ reset clocks
- $\phi_5 = \phi_4 \land I(\ell')$ intersect with invariant of $\ell'$

let time elapse.

$\ell$

$y < 3$

$y := 0$

$\ell'$

$x > 1$
Example

- $\varphi_1 = \varphi_0 \uparrow$
- $\varphi_2 = \varphi_1 \land I(\ell)$  
  intersect with invariant of $\ell$
- $\varphi_3 = \varphi_2 \land \varphi$
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$  
  reset clocks
- $\varphi_5 = \varphi_4 \land I(\ell')$  
  intersect with invariant of $\ell'$

let time elapse.

Example

- $\varphi_1 = \varphi_0 \uparrow$
- $\varphi_2 = \varphi_1 \land I(\ell)$  
  intersect with invariant of $\ell$
- $\varphi_3 = \varphi_2 \land \varphi$
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$  
  reset clocks
- $\varphi_5 = \varphi_4 \land I(\ell')$  
  intersect with invariant of $\ell'$

Example

- $\varphi_1 = \varphi_0 \uparrow$
- $\varphi_2 = \varphi_1 \land I(\ell)$  
  intersect with invariant of $\ell$
- $\varphi_3 = \varphi_2 \land \varphi$
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$  
  reset clocks
- $\varphi_5 = \varphi_4 \land I(\ell')$  
  intersect with invariant of $\ell'$

Example

- $\varphi_1 = \varphi_0 \uparrow$
- $\varphi_2 = \varphi_1 \land I(\ell)$  
  intersect with invariant of $\ell$
- $\varphi_3 = \varphi_2 \land \varphi$
- $\varphi_4 = \varphi_3[y_1 := 0] \ldots [y_n := 0]$  
  reset clocks
- $\varphi_5 = \varphi_4 \land I(\ell')$  
  intersect with invariant of $\ell'$
**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
  $M : (X \cup \{x_0\}) \times (X \cup \{x_0\}) \rightarrow (\{<, \le\} \times \mathbb{Z}) \cup \{(<, \infty)\}$

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

![Difference Bound Matrices Diagram](image)

\[ M(x_0, y) = x_0 - y \leq -5 \]
\[ = y \leq -5 \]
\[ = y > 5 \]
\[ M(x_2, z) = (c, z) \]
\[ x - y < \varepsilon_f \]
Given a finite set of clocks $X$, a **DBM** over $X$ is a mapping

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

- $M(x, y) = (\sim, c)$ encodes the conjunct $x - y \sim c$ \,(x and y can be $x_0$).
- If $M$ and $N$ are **DBMs encoding** $\varphi_1$ and $\varphi_2$ (representing zones $z_1$ and $z_2$), then we can efficiently compute $M \uparrow, M \land N, M[x := 0]$ such that
  - all three are again **DBM**.
  - $M \uparrow$ encodes $\varphi_1 \uparrow$,
  - $M \land N$ encodes $\varphi_1 \land \varphi_2$, 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
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
