Regions & zones

A set of configurations may span a collection of regions

Zone

Defined by a conjunction of constraints

$$x - y \; R \; c$$

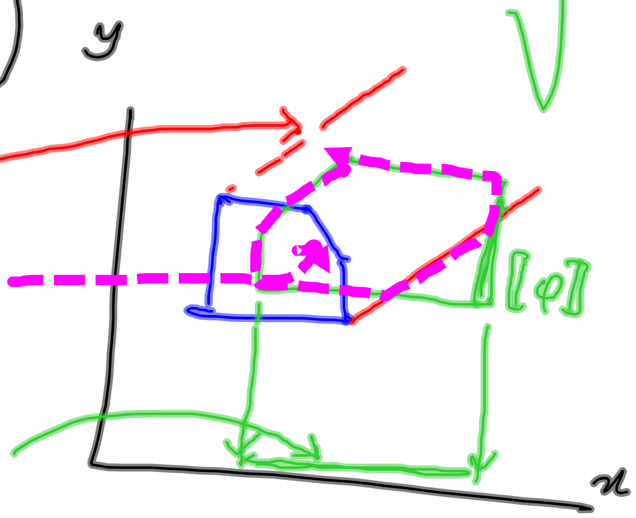
Assume an extra clock $\overline{0}$ which is always 0

Manipulating zones is "efficient"

$s \xrightarrow[R]{a,\varphi} s'$ zone format

$[\![\varphi]\!] = \{v \mid v \models \varphi\}$ is a zone ✓

(s, Z) (s', Z')



let time elapse

identify if φ is true

reset clocks in R

Reset y

Given a representation of zones Z, Z'

upward-closure (Z) at time elapse

$Z \cap Z'$

$[R \leftarrow 0](Z)$

and reverse : backward-closure (Z)

$[\text{free } R]\ (Z)$

Representation of zones

Difference Bound Matrix (DBM)

$$D:\quad \begin{array}{cc} & \begin{array}{ccccc} 0 & c_1 & c_2 & \cdots & c_n \end{array} \\ \begin{array}{c} 0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{array} & \left[\begin{array}{ccccc} \; & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{array} \right] \end{array}$$

$<$, $\leq$

$$D_{ij} = (n, \preccurlyeq)$$

$\in \mathbb{Z} \cup \{\infty\}$

$$c_i - c_j \preccurlyeq n$$

$4 < c_i - c_j \leq 7$

$D_{ij} = (7, \leq)$

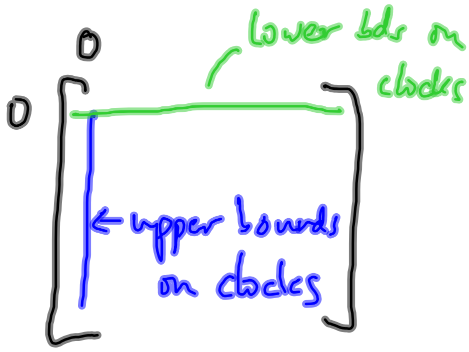
$D_{ji} = (-4, <)$

$c_j - c_i < -4$

$c \leq 20 \qquad c - \overline{0} \leq 20$

$c \geq 7 \qquad \overline{0} - c \leq -7$

Diagonal entries : $c - c \leq 0$



Canonicity

$$x - y \leq 5 \qquad z - x \leq 10$$

$$\Downarrow$$

$$z - y \leq 15$$

$$
\begin{array}{c|c}
 & \begin{array}{ccc} x & y & z \end{array} \\
\begin{array}{c} x \\ y \\ z \end{array} & \left[\begin{array}{ccc} & 5 & \\ & & \\ 10 & 20 & \end{array}\right]
\end{array}
$$

∞

Canonicity

Step 1 : Closed (under implication)

eg $x - y \leq 5$, $z - x \leq 10$, $z - y \leq \infty$

$\downarrow$

$z - y \leq 15$

Represent DBM as a weighted directed graph

Vertices = clocks

$x - y \le c$ $\qquad$ $x \xleftarrow{(c,\le)} y$

$$\begin{tikzcd} x \arrow[d, "{(10,\le)}"'] & y \arrow[l, "{(5,\le)}"'] \arrow[dl, "{(\infty,\le)}" , "{(15,\le)}"'] \\ z & \end{tikzcd}$$

Compute shortest path from y to z

$$(m, <) + (n, \leq) = (m+n, <)$$

$$(m, \precsim_1) \leq (n, \precsim_2) \text{ if } m < n$$

$$\text{or } m = n \ \& \ \precsim_1 = <$$

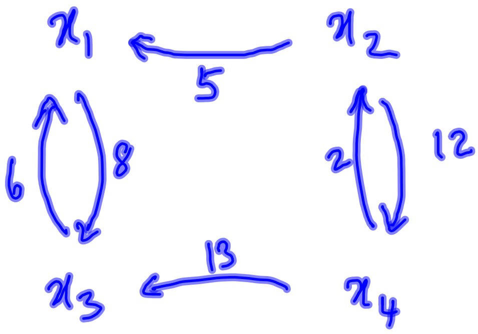
Step 2 : Minimization

Remove redundant constraints

$x - y \leq 5$, $z - x \leq 10$, $\underline{z - y \leq 15}$
redundant

In graph remove edge $x \longrightarrow y$
if there is a path $x \dashrightarrow y$ with same weight.

Check $x \to y$ & $x \to z$, $z \to y$ $\forall z$



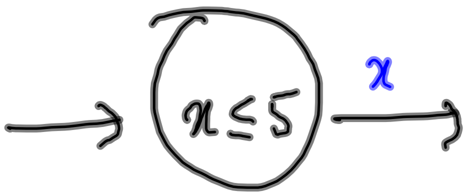
upclose(D) — relax all upper bds to ∞

$Z \cap Z'$ $\quad Z' = \varphi_1 \wedge \varphi_2 \wedge \dots \varphi_k$

$Z \cap [\![\varphi_1]\!]$

-

Invariants on states - progress



Next generalisation

$\dot{x}$ depends on state

timed automaton: $\dot{x} = 1$ $\forall x$, $\forall s$

Hybrid automata

Discrete behaviour : State changes
"Mode" changes

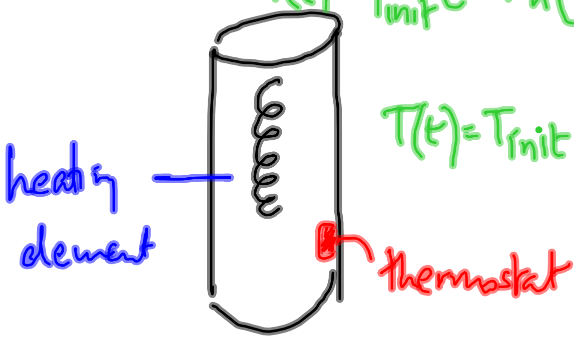
Continuous behaviour : State dependent

Thermostat:

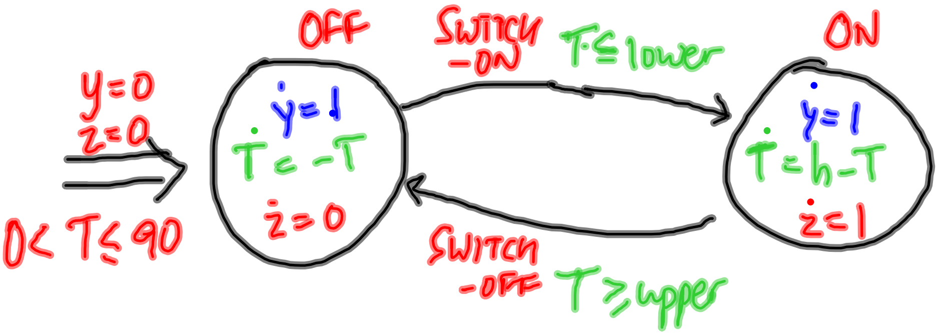
Two states:

$T(t) = T_{init} e^{-kt} + h(1-e^{-kt})$ Heater is on – T increases

$T(t) = T_{init} e^{-kt}$ Heater is off – T decreases



Measuring temperature T



Want to ask questions such as reachability etc

Note that timed automata are a special case

Need to severely restrict hybrid automata for problems to be decidable

Rectangular hybrid automata

"Rectangle" over variables $X = \{x_1, \ldots, x_n\}$

is the set of points in $\mathbb{R}^n$ generated

by n intervals $l_i \lesssim x_i \lesssim u_i$

Specifying a hybrid automaton

$\forall s$: flow(s) flow condition — invariant / defn of $\dot{x}$

init(s) initial condition if system starts at s

Transitions from s to s'

Jump(s,s')
- Precondition = guard
- Postcondition = reset
- Action

stable(x) ⇒ x after transition retains value

Rectangular hybrid automaton

Flow(s), Init(s), Jump(s) all specified as rectangles

Rectangular —

No "interference" across variables

What happens to value of x when its derivative changes?

Retain value

Initialised || Force value to be reset (post(s))

Only initialised rectangular hybrid automata
have a decidable reachability problem