Zones and Difference Bound Matrices

Lecture #18 of Advanced Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

E-mail: `katoen@cs.rwth-aachen.de`

January 15, 2006

TCTL model checking

- TCTL model-checking problem: $TA \models \Phi$ for non-Zeno TA

$$\underbrace{TA \models \Phi}_{\text{timed automaton}} \quad \text{iff} \quad \underbrace{TS(TA) \models \Phi}_{\text{infinite transition system}}$$

  - timelocks in TA are irrelevant as their presence can be checked

- Idea: consider a finite quotient of $TS(TA)$ wrt. a bisimulation
  - $TS(TA)/\cong$ is a *region* transition system and denoted $RG(TA)$
  - dependence on Φ is ignored

- Transform TCTL formula Φ into an "equivalent" CTL-formula $\widehat{\Phi}$

- Then: $TA \models_{\text{TCTL}} \Phi \quad \text{iff} \quad \underbrace{RG(TA)}_{\text{finite transition system}} \models_{\text{CTL}} \widehat{\Phi}$

Clock equivalence

Impose an equivalence, denoted $\cong$, on the clock valuations such that:

(A) Equivalent clock valuations satisfy the same clock constraints g in *TA* and Φ:

$$\eta \cong \eta' \;\Rightarrow\; (\eta \models g \quad \text{iff} \quad \eta' \models g)$$

- no diagonal clock constraints are considered
- all the constraints in *TA* and Φ are thus either of the form $x \leqslant c$ or $x < c$

(B) Time-divergent paths emanating from equivalent states are "equivalent"

- this property guarantees that equivalent states satisfy the same path formulas

(C) The number of equivalence classes under $\cong$ is finite

Clock equivalence

- Correctness criteria (A) and (B) are ensured if equivalent states:
  - agree on the integer parts of all clock values, and
  - agree on the ordering of the fractional parts of all clocks

$\Rightarrow$ This yields a denumerable infinite set of equivalence classes

- Observe that:
  - if clocks exceed the maximal constant with which they are compared their precise value is not of interest

$\Rightarrow$ The number of equivalence classes is then finite (C)

Clock equivalence

Clock valuations $\eta, \eta' \in \textit{Eval}(C)$ are *equivalent*, denoted $\eta \cong \eta'$, if:

(1) for any $x \in C$: $(\eta(x) > c_x) \wedge (\eta'(x) > c_x)$ or $(\eta(x) \leqslant c_x) \wedge (\eta'(x) \leqslant c_x)$

(2) for any $x \in C$: if $\eta(x), \eta'(x) \leqslant c_x$ then:

$$\lfloor \eta(x) \rfloor = \lfloor \eta'(x) \rfloor \quad \text{and} \quad \mathit{frac}(\eta(x)) = 0 \text{ iff } \mathit{frac}(\eta_2(x)) = 0$$

(3) for any $x, y \in C$: if $\eta(x), \eta'(x) \leqslant c_x$ and $\eta(y), \eta'(y) \leqslant c_y$, then:

$$\mathit{frac}(\eta(x)) \leqslant \mathit{frac}(\eta(y)) \quad \text{iff} \quad \mathit{frac}(\eta'(x)) \leqslant \mathit{frac}(\eta'(y)).$$

$$s \cong s' \quad \text{iff} \quad \ell = \ell' \quad \text{and} \quad \eta \cong \eta'$$

Regions

- The *clock region* of $\eta \in \textit{Eval}(C)$, denoted $[\eta]$, is defined by:

$$[\eta] \;=\; \{\, \eta' \in \textit{Eval}(C) \mid \eta \cong \eta' \,\}$$

- The *state region* of $s = \langle \ell, \eta \rangle \in \textit{TS}(\textit{TA})$ is defined by:

$$[s] \;=\; \langle \ell, [\eta] \rangle \;=\; \{\, \langle s, \eta' \rangle \mid \eta' \in [\eta] \,\}$$

Canonical representation of regions

- Each clock region can be uniquely represented
- For each clock x a term of the form (where $n \in \mathbb{N}$ and $n < c_x$):
  - $x = n$, or
  - $n < x < n{+}1$, or
  - $x > c_x$
- For each pair of clocks x, y a term of the form:
  - $x - y < 0$, or
  - $x - y = n$, or
  - $n < x - y < n{+}1$, or
  - $x - y > c_x$

Clock equivalence is a bisimulation

Clock equivalence is a bisimulation equivalence over AP'

Unbounded and successor regions

- Clock region $r_\infty = \{ \eta \in \mathit{Eval}(C) \mid \forall x \in C.\, \eta(x) > c_x \}$ is *unbounded*
- r' is the *successor (clock) region* of r, denoted $r' = \mathit{succ}(r)$, if either:
  1. $r = r_\infty$ and $r = r'$, or
  2. $r \neq r_\infty$, $r \neq r'$ and for all $\eta \in r$:

$$\exists d \in \mathbb{R}_{>0}.\ (\eta{+}d \in r' \quad \text{and} \quad \forall 0 \leqslant d' \leqslant d.\, \eta{+}d' \in r \cup r')$$

- The *successor region*: $\mathit{succ}(\langle \ell, r \rangle) \ = \ \langle \ell, \mathit{succ}(r) \rangle$
- Note: the location invariants are ignored so far!

Region automaton

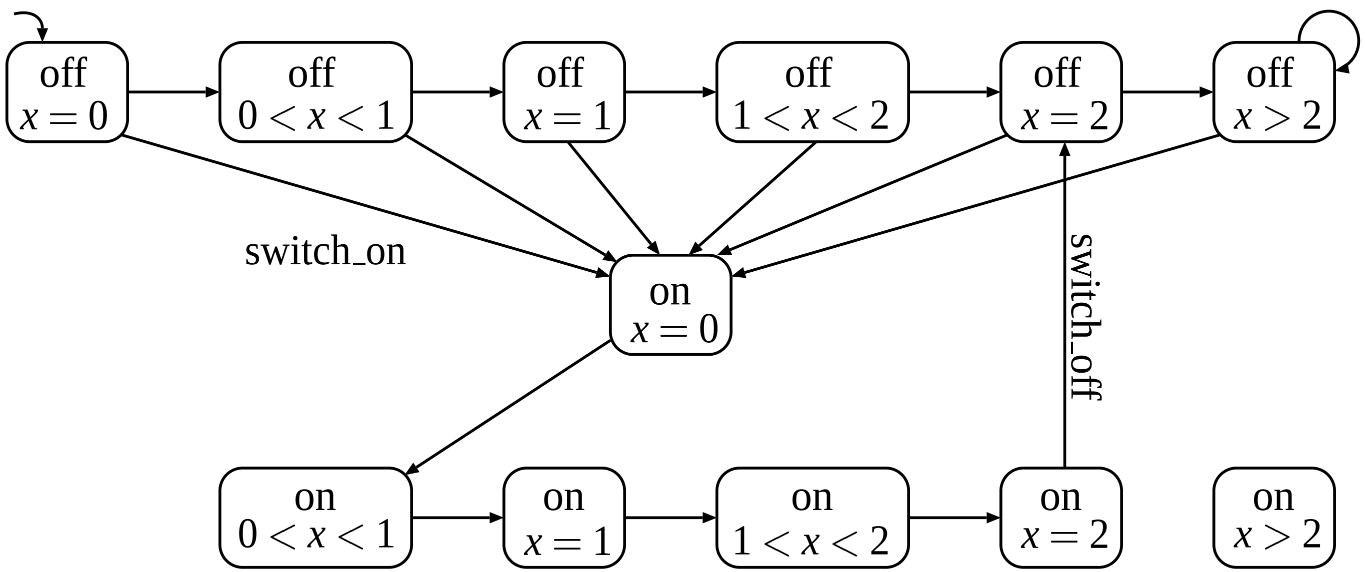
For non-Zeno *TA* with $\mathit{TS}(\mathit{TA}) = (S, \mathit{Act}, \rightarrow, I, \mathit{AP}, L)$ let:

$$\mathit{RG}(\mathit{TA}, \Phi) = (S', \mathit{Act} \cup \{ \tau \}, \rightarrow', I, \mathit{AP}', L') \quad \text{with}$$

- $S' = S/\cong = \{ [s] \mid s \in S \}$ and $I' = \{ [s] \mid s \in I \}$, the state regions
- $L'(\langle \ell, r \rangle) = L(\ell) \cup \{ g \in \mathit{AP}' \setminus \mathit{AP} \mid r \models g \}$
- $\rightarrow'$ is defined by: $\dfrac{\ell \overset{g:\alpha,D}{\rightsquigarrow} \ell' \quad r \models g \quad \text{reset } D \text{ in } r \models \mathit{Inv}(\ell')}{\langle \ell, r \rangle \xrightarrow{\alpha}{}' \langle \ell', \text{reset } D \text{ in } r \rangle}$ and

$$\frac{r \models \mathit{Inv}(\ell) \quad \mathit{succ}(r) \models \mathit{Inv}(\ell)}{\langle \ell, r \rangle \xrightarrow{\tau}{}' \langle \ell, \mathit{succ}(r) \rangle}$$

Example: simple light switch



Number of regions

The *number of clock regions* is bounded from below and above by:

$$|C|! \cdot \prod_{x \in C} c_x \;\leqslant\; \big| \underbrace{\textit{Eval}(C)/\!\cong}_{\text{number of regions}} \big| \;\leqslant\; |C|! \cdot 2^{|C|-1} \cdot \prod_{x \in C} (2c_x + 2)$$

where for the upper bound it is assumed that $c_x \geqslant 1$ for any $x \in C$

the number of state regions is $|\textit{Loc}|$ times larger

a more compact representation is obtained by zones

Zones

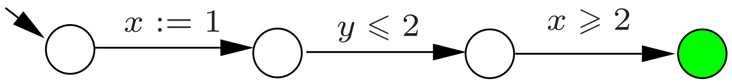
- Clock constraints are *conjunctions* of atomic constraints
  - $x \prec c$ and $x - y \prec c$ for $\prec \in \{ <, \leqslant, =, \geqslant, > \}$
  - restrict to *TA* with *only conjunctive clock constraints*
  - and (as before) assume no difference clock constraints

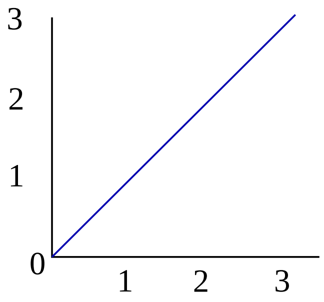
- A *clock zone* is the set of clock valuations that satisfy a clock constraint
  - a clock zone for g is the maximal set of clock valuations satisfying g

- Clock zone of g: $[\![g]\!] = \{ \eta \in \mathit{Eval}(C) \mid \eta \models g \}$
  - use z, z' and so on to range over zones

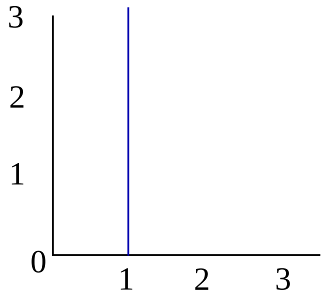
- The *state zone* of $s = \langle \ell, \eta \rangle \in \mathit{TS}(\mathit{TA})$ is $\langle \ell, z \rangle$ with $\eta \in z$

Zone automaton: intuition

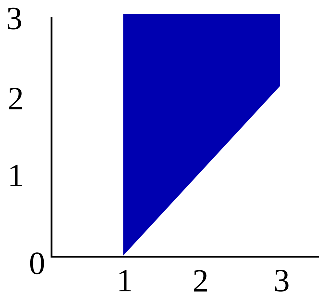




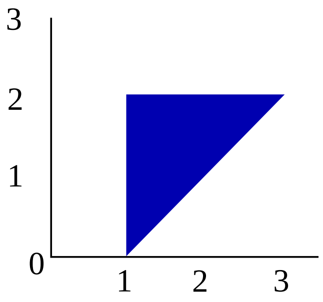
leaving initial



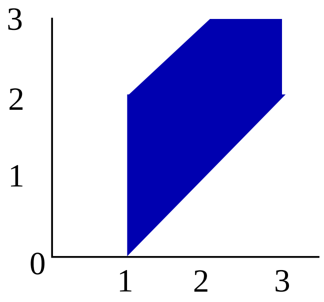
entering first



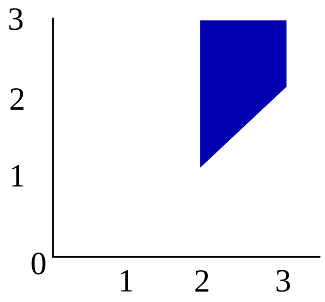
leaving first



entering second



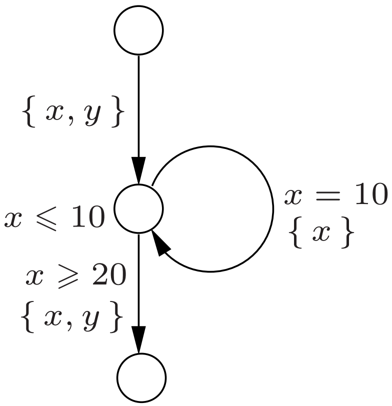
leaving second

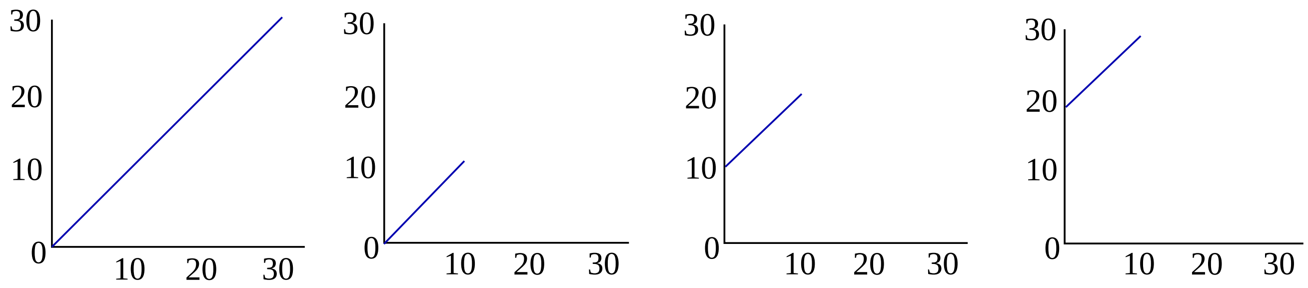


entering third

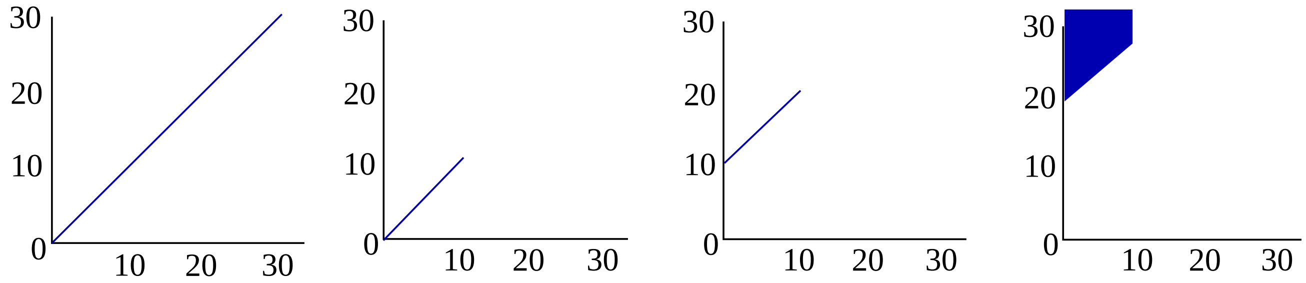
Normalization: intuition

symbolic semantics has infinitely many zones:





normalization yields a finite zone graph:

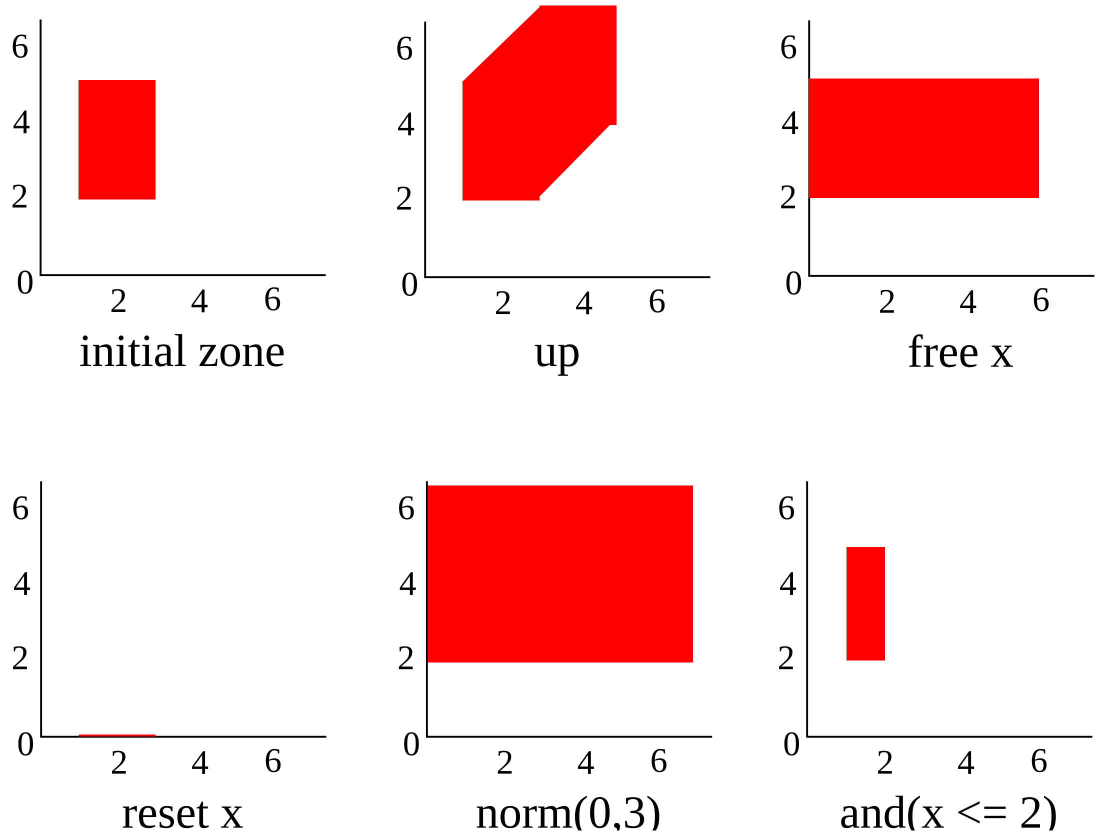


more about normalization later.....

Successor and reset zones

- z' is the *successor* (clock) zone of z, denoted $z' = z^{\uparrow}$, if:
  - $z^{\uparrow} \;=\; \{\, \eta + d \mid \eta \in z, d \in \mathbb{R}_{>0} \,\}$
- z' is the zone obtained from z by *resetting* clocks D:
  - *reset* D *in* $z \;=\; \{\,$ *reset* D *in* $\eta \mid \eta \in z \,\}$

Some operations on zones



Zone automaton

For non-Zeno *TA* (without difference clock constraints) let:

$$ZG(TA, \Phi) = (S, Act \cup \{ \tau \}, \rightarrow, I, AP', L) \quad \text{with}$$

- $S = Loc \times Zone(C)$ and $I = \{ \langle \ell, z_0 \rangle \mid \ell \in Loc_0 \}$
- $L(\langle \ell, z \rangle) = L(\ell) \cup \{ g \mid g \in z \}$
- $\rightarrow$ is defined by: $\langle \ell, z \rangle \xrightarrow{\tau} \langle \ell, z^{\uparrow} \wedge Inv(\ell) \rangle$ and

$$\frac{\ell \overset{g:\alpha,D}{\rightsquigarrow} \ell'}{\langle \ell, z \rangle \xrightarrow{\alpha} \langle \ell', \textit{reset } D \textit{ in } (z \wedge g) \ \wedge \ Inv(\ell') \rangle}$$

Correctness (1)

For timed automaton *TA* and any initial state $\langle \ell, \eta_0 \rangle$:

- Soundness:

$$\underbrace{\langle \ell, \underbrace{\{\,\eta_0\,\}}_{z_0} \rangle \longrightarrow^* \langle \ell', z' \rangle}_{\text{in } ZG(TA)} \quad \text{implies} \quad \underbrace{\langle \ell, \eta_0 \rangle \longrightarrow^* \langle \ell', \eta' \rangle}_{\text{in } TS(TA)} \text{ for all } \eta' \in z'$$

- Completeness:

$$\underbrace{\langle \ell, \eta_0 \rangle \longrightarrow^* \langle \ell', \eta' \rangle}_{\text{in } TS(TA)} \quad \text{implies} \quad \underbrace{\langle \ell, \{\,\eta_0\,\} \rangle \longrightarrow^* \langle \ell', z' \rangle}_{\text{in } ZG(TA)} \text{ for some } z' \text{ with } \eta' \in z'$$

Example

Zone normalization

- To obtain a finite representation, *zone normalization* is employed
- For zone z, $norm(z) \;=\; \{\, \eta \mid \eta \cong \eta', \eta' \in z \,\}$
  - where $\cong$ is the clock equivalence
- There can only be finitely many normalized zones
- $\langle \ell, z \rangle \longrightarrow_{norm} \; \langle \ell', norm(z') \rangle$ if $\langle \ell, z \rangle \longrightarrow \; \langle \ell', z' \rangle$

Correctness (2)

For timed automaton *TA* and any initial state $\langle \ell, \eta \rangle$:

- Soundness:

$$\langle \ell, \{ \eta_0 \} \rangle \longrightarrow^*_{norm} \langle \ell', z' \rangle \quad \text{implies} \quad \langle \ell, \eta_0 \rangle \longrightarrow^* \langle \ell', \eta' \rangle$$

  - for all $\eta' \in z'$ such that $\forall x.\, \eta'(x) \leqslant c_x$

- Completeness:

$$\langle \ell, \eta_0 \rangle \longrightarrow^* \langle \ell', \eta' \rangle \text{ with } \forall x.\, \eta'(x) \leqslant c_x \quad \text{implies} \quad \langle \ell, \{ \eta_0 \} \rangle \longrightarrow^*_{norm} \langle \ell', z' \rangle$$

  - for some z' such that $\eta' \in z'$

- Finiteness: the transition relation $\longrightarrow_{norm}$ is finite

Example

Forward reachability algorithm

```
PASSED := ∅;                                              // explored states so far
WAIT := { (ℓ0, z0) };                                     // states to be explored
while WAIT ≠ ∅                                            // still states to go
do select and remove (ℓ, z) from WAIT;
    if (ℓ = goal ∧ z ∩ z_goal ≠ ∅) then return "reachable"! fi ;
    if ¬(∃(ℓ, z') ∈ PASSED. z ⊆ z')                       // no "super"state explored yet
    then add (ℓ, z) to PASSED                             // (ℓ, z) is a new state
        foreach (ℓ', z') with (ℓ, z) →_norm (ℓ', z')
        do add (ℓ', z') to WAIT;                          // add symbolic successors
    fi
od
return "not reachable"!
```

Representing zones

- Let $\mathbf{0}$ be a clock with constant value 0; let $C_0 = C \cup \{ \mathbf{0} \}$
- Any zone $z \in$ *Zone*(C) can be written as:
  - conjunction of constraints $x - y < n$ or $x - y \leqslant n$ for $n \in \mathbb{Z}$, $x, y \in C_0$
  - when $x - y \preceq n$ and $x - y \preceq m$ take only $x - y \preceq \min(n, m)$

  $\Rightarrow$ this yields at most $|C_0| \cdot |C_0|$ constraints
- Example:

$$x - \mathbf{0} < 20 \;\wedge\; y - \mathbf{0} \leqslant 20 \;\wedge\; y - x \leqslant 10 \;\wedge\; x - y \leqslant -10 \;\wedge\; \mathbf{0} - z < 5$$

- Store each such constraint in a matrix
  - this yields a *difference bound matrix*

Difference bound matrices

- Zone z over C is represented by DBM **Z** of cardinality $|C{+}1|\cdot|C{+}1|$
  - for $C = x_1, \ldots, x_n$, let $C_0 = \{ x_0, x_1, \ldots, x_n \}$ with $x_0 = \mathbf{0}$
  - $\mathbf{Z}(i, j) = (c, \prec)$ if and only if $x_i - x_j \prec c$
- Definition of **Z** for zone z:
  - for $x_i - x_j \prec c$ let $\mathbf{Z}(i, j) = (c, \prec)$
  - if $x_i - x_j$ is unbounded in z, set $\mathbf{Z}(i, j) = \infty$
  - $\mathbf{Z}(0, i) = (\leqslant, 0)$ and $\mathbf{Z}(i, i) = (\leqslant, 0)$
- Operations on bounds:
  - $(c, \preceq) < \infty$, $(c, <) < (c, \leqslant)$, and $(c, \preceq) < (c', \preceq')$ if $c < c'$
  - $c + \infty = \infty$, $(c, \leqslant) + (c', \leqslant) = (c{+}c', \leqslant)$ and $(c, <) + (c', \leqslant) = (c{+}c', <)$

Example

The need for canonicity

Canonical DBMs

- A zone z is in *canonical form* if and only if:
  - no constraint in z can be strengthened without reducing $[\![z]\!] = \{ \eta \mid \eta \in z \}$
- For each zone z: $\exists$ a *unique* and *equivalent* zone in canonical form
- Represent zone z by a *weighted digraph* $G = (V, E, w)$ where
  - $V = C_0$ is the set of vertices
  - $(x_i, x_j) \in E$ whenever $x_j - x_i \preceq c$ is a constraint in z
  - $w(x_i, x_j) = (\preceq, c)$ whenever $x_j - x_i \preceq c$ is a constraint in z
- Zone z is in *canonical form* if and only if DBM **Z** satisfies:
  - $\mathbf{Z}(i, j) \leqslant \mathbf{Z}(i, k) + \mathbf{Z}(k, j)$ for any $x_i, x_j, x_k \in C_0$
- Compute canonical zone?
  - use *Floyd-Warshall*'s all-pairs SP algorithm (time $\mathcal{O}(|C_0|^3)$)

Example

Minimal constraint systems

- A zone may contain *redundant* constraints
  - e.g., in $x-y < 2$, $y-z < 5$, and $x-z < 7$, constraint $x-z < 7$ is redundant
- Reduce memory usage: consider *minimal* constraint systems
  - e.g., $x-y \leqslant 0$, $y - z \leqslant 0$, $z - x \leqslant 0$, $x-\mathbf{0} \leqslant 3$, and $\mathbf{0}-x < -2$
  - is a minimal representation of a zone in canonical form with 12 constraints
- For each zone: $\exists$ a unique and equivalent minimal constraint system
- Determining minimal representations of canonical zones:
  - $x_i \xrightarrow{(n,\preceq)} x_j$ is redundant if an alternative path from x_i to x_j has weight at most $(n, \preceq)$
  - it suffices to consider alternative paths of length two

zero cycles require a special treatment

Main operations on DBMs (1)

- *Nonemptiness*: is $[\![\mathbf{Z}]\!] \neq \varnothing$?
  - search for negative cycles in the graph representation of **Z**, or
  - mark **Z** when upper bound of some clock is set to value $<$ its lower bound
- *Inclusion test*: is $[\![\mathbf{Z}]\!] \subseteq [\![\mathbf{Z}']\!]$?
  - for DBMs in canonical form, test whether $\mathbf{Z}(i,j) \leqslant \mathbf{Z}'(i,j)$, for all $i, j \in C_0$
- *Delay*: determine $\mathbf{Z}^{\uparrow}$
  - remove the upper bounds on any clock, i.e.,
  - $\mathbf{Z}^{\uparrow}(i,0) = \infty$ and $\mathbf{Z}^{\uparrow}(i,j) = \mathbf{Z}(i,j)$ for $j \neq 0$

Main operations on DBMs (2)

- *Conjunction*: $z \wedge (x_i - x_j \preceq n)$
  - if $(n, \preceq) < \mathbf{Z}(i,j)$ then $\mathbf{Z}(i,j) := (n, \preceq)$ else do nothing
  - put $\mathbf{Z}$ back into canonical form (in time $\mathcal{O}(|C_0|^2)$ using that only $\mathbf{Z}(i,j)$ changed)
- *Clock reset*: $x_i := d$
  - $\mathbf{Z}(i,j) := (d, \leqslant) + \mathbf{Z}(0,j)$ and $\mathbf{Z}(j,i) := \mathbf{Z}(j,0) + (-d, \leqslant)$
- *Normalization*
  - remove all bounds $x - y \preceq m$ for which $(m, \preceq) > (c_x, \leqslant)$, and
  - set all bounds $x - y \preceq m$ with $(m, \preceq) < (-c_y, <)$ to $(-c_y, <)$
  - put the DBM back into canonical form (Floyd-Warshall)