Timed Automata

Semantics, Algorithms and Tools

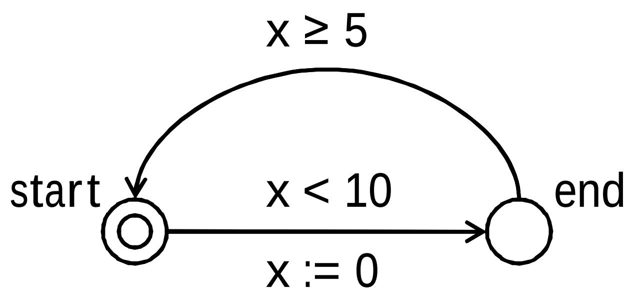
Zhou Huaiyang

Agenda

- **Introduction**
- **Timed Automata**
  - Formal Syntax
  - Operational Semantics
  - Verification Problems
- **Symbolic Semantics & Verification**
  - Regions, Zones, and Symbolic Semantics
  - Zone-Normalization for Automata
  - Symbolic Reachability Analysis
- **DBM**

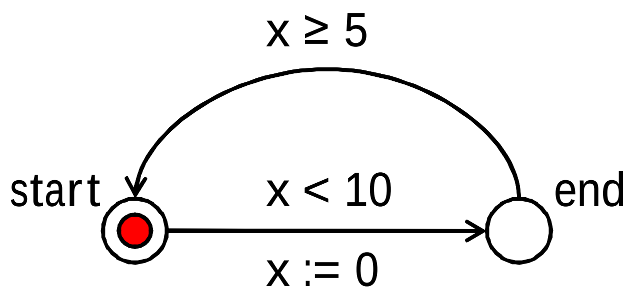
Introduction

- Timed Automata
  - For modeling & verification of real time systems.



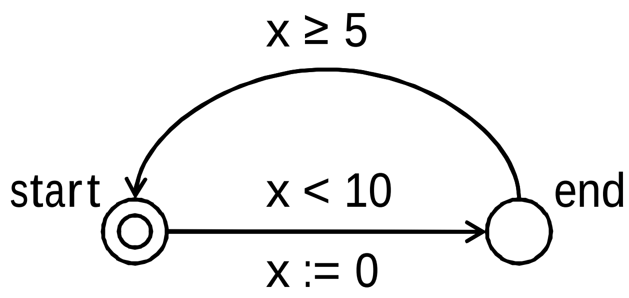
Introduction cont.

- Timed Büchi Automata
  - Büchi-acceptance conditions



Introduction cont.

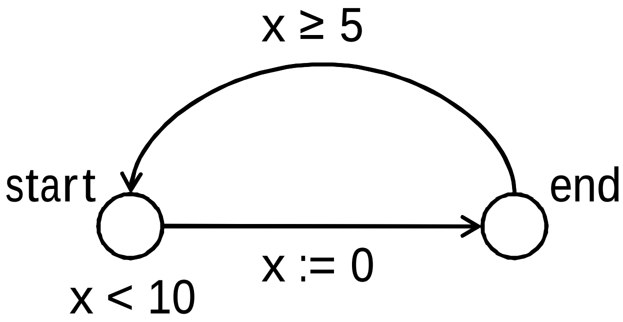
- Timed Safety Automata
  - Local invariant



Timed Automata

- Formal Syntax

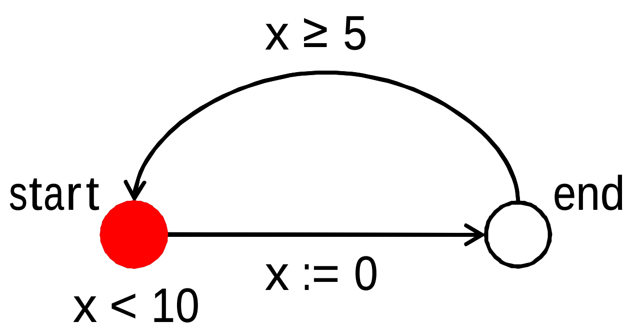
$$\langle \mathcal{N}, \ell_0, \mathcal{E}, \mathcal{I} \rangle$$



- $\ell \xrightarrow{g,\, a,\, r} \ell'$ when $\langle \ell, g, a, r, \ell' \rangle \in \mathcal{E}$

Operational Semantics

- Timed Transition System
  - states: $<l, u>$
  - transitions:
    - $<l, u> \xrightarrow{d} <l, u+d>$
      - if $u \in I(l)$ and $(u+d) \in I(l)$ for $d \in \mathbb{R}_+$
    - $<l, u> \xrightarrow{a} <l', u'>$
      - if $l \xrightarrow{g, a, r} l'$, $u \in g$, $u' = [r \mapsto 0]u$ and $u' \in I(l')$



Verification Problems

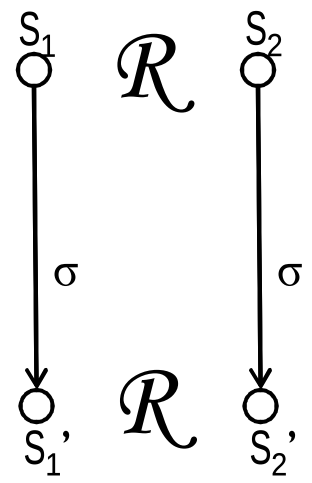
- Timed action: (t, a)
- Timed trace: $\xi=(t_1,a_1)(t_2,a_2)..(t_i,a_i)\ldots$
  - where $t_i \le t_{i+1}$ for all $i > 1$
- Run over a timed trace:
  - $<\ell_0, u_0> \xrightarrow{d_1} \xrightarrow{a_1} <\ell_1, u_1> \xrightarrow{d_2} \xrightarrow{a_2} <\ell_2, u_2> \xrightarrow{d_3} \xrightarrow{a_3} <\ell_3, u_3> \ldots$
    - $t_i = t_{i-1}+d_i$ for all $i \ge 1$
- Timed language $L(\mathcal{A})$:
  - all timed traces ξ for which there exists a run of A over ξ
- Untimed language $L_{untimed}(\mathcal{A})$:
  - e.g. $a_1a_2a_3\ldots$

Verification Problems cont.

- Language Inclusion: check L($\mathcal{A}$) ⊆ L($\mathcal{B}$)
  - Undecidable:
    - Timed automata is not determinizable in general.
    - Timed automata can not be complemented.
    - Essentially due to the arbitrary clock reset.
  - Decidable if:
    - $\mathcal{B}$ is restricted to deterministic class
      - event-clock automata & timed communicating sequential processes
    - Determinizable
      - All the edges labeled with the same action symbol are also labeled with the same set of clocks to reset
- Untimed Language Inclusion: Decidable

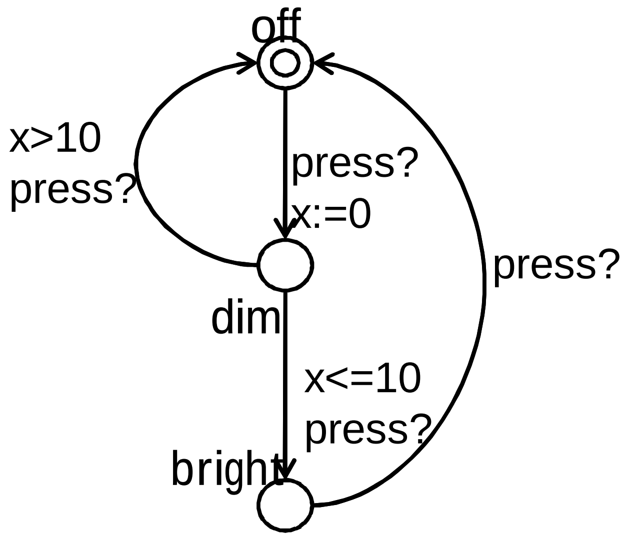
Verification Problems cont.

- Bisimulation $\mathcal{R}$
  - $\sigma \in \Sigma \cup R_+$
- Timed bisimilar iff
  - $(s_0, s_0') \in \mathcal{R}$
- Timed bisimulation
  - decidable.
- Untimed bisimulation
  - decidable



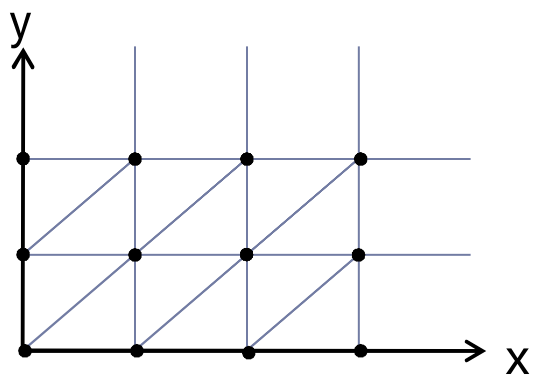
Verification Problems cont.

- Reachability Analysis
  - $<l,u>$ reachable iff
    - $<l_0, u_0> \longrightarrow^* <l, u>$
  - $<l,\Phi>$ reachable if
    - $<l,u>$ reachable for some u satisfying Φ
      - $\Phi \in \mathcal{B}(C)$, the set of clock constraints
  - decidable



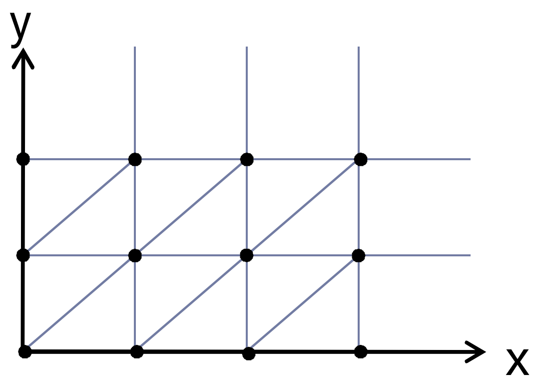
Symbolic Semantics & Verification

- Region Equivalence: $u \cong_k v$, *iff*
  - $\forall x$, either $\lfloor u(x) \rfloor = \lfloor v(x) \rfloor$ or both $u(x) > k(x)$ and $v(x) > k(x)$
  - $\forall x$, if $u(x) \leq k(x)$ then $\{u(x)\} = 0$ *iff* $\{v(x)\} = 0$
  - $\forall x, y$ if $u(x) \leq k(x)$ and $u(y) \leq k(y)$ then $\{u(x)\} \leq \{u(y)\}$ *iff* $\{v(x)\} \leq \{v(y)\}$
- Region: $[u]$
- Basis for finite partitioning:
  - fixed number of clocks
  - $(l, u) \sim (l, v)$



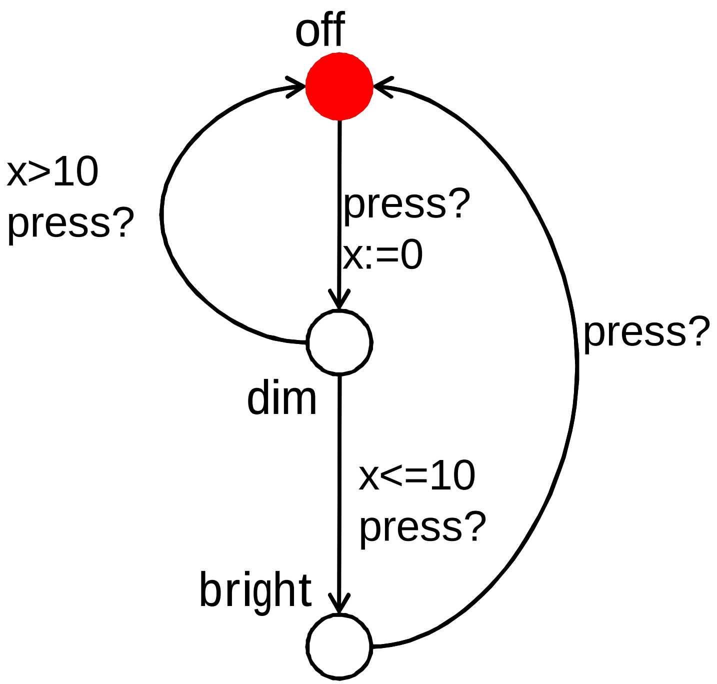
Symbolic Semantics & Verification cont.

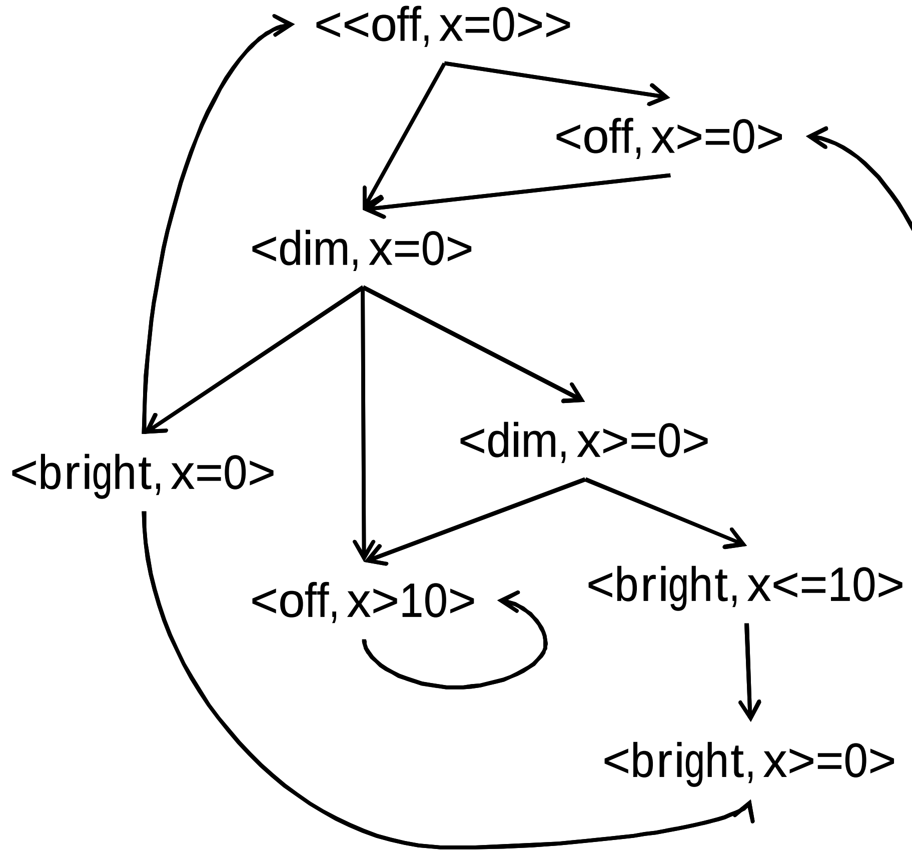
- Transition:
  - $<\ell, [u]> \Rightarrow <\ell, [v]>$
    - if $<\ell, u> \xrightarrow{d} <\ell, v>$ for d $\in \mathcal{R}_+$
  - $<\ell, [u]> \Rightarrow <\ell', [v]>$
    - if $<\ell, u> \xrightarrow{a} <\ell', v>$ for an action a
- $\Rightarrow$ is finite, so region graph is finite.
- Problem: state-space explosion
- Solution: zone



Symbolic Semantics & Verification cont.

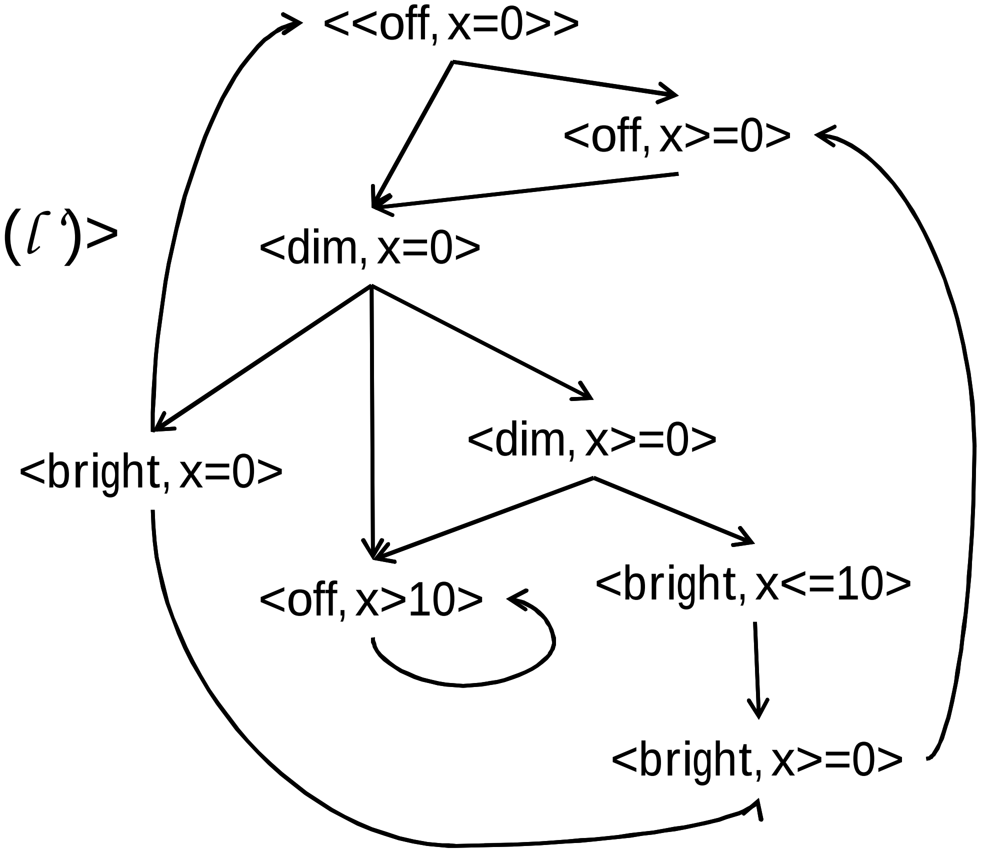
- Zone: [D]
- Symbolic state: <ℓ, D>





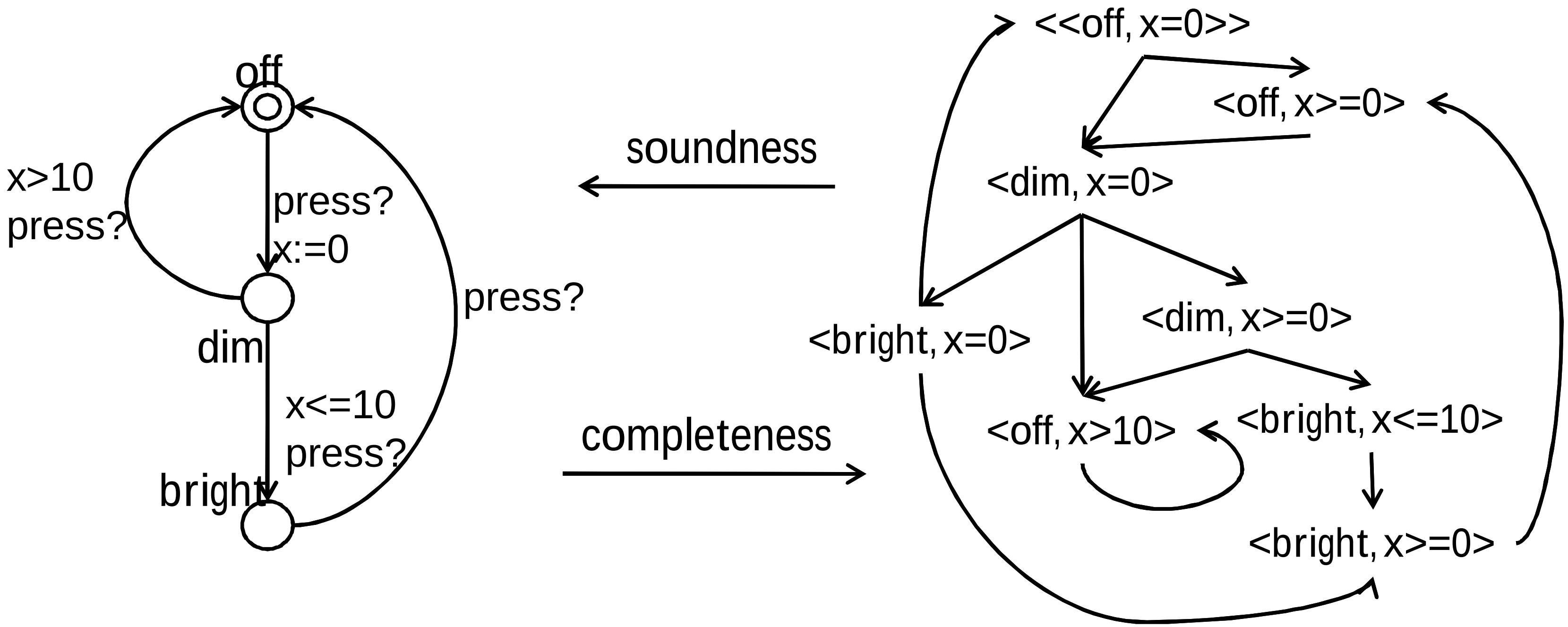
Symbolic Semantics & Verification cont.

- $D^{\uparrow} = \{u + d \mid u \in D,\ d \in \mathcal{R}_+\}$
- $r(D) = \{\, [r \mapsto 0]u \mid u \in D\}$
- Symbolic transition: $\rightsquigarrow$
  - $\langle \ell, D\rangle \rightsquigarrow \langle \ell, D^{\uparrow} \wedge I(\ell)\rangle$
  - $\langle \ell, D\rangle \rightsquigarrow \langle \ell', r(D \wedge g) \wedge I(\ell')\rangle$
    - if $\ell \xrightarrow{g,\,a,\,r} \ell'$



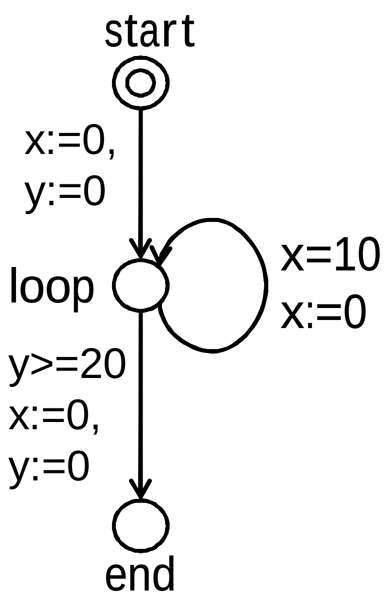
Symbolic Semantics & Verification cont.

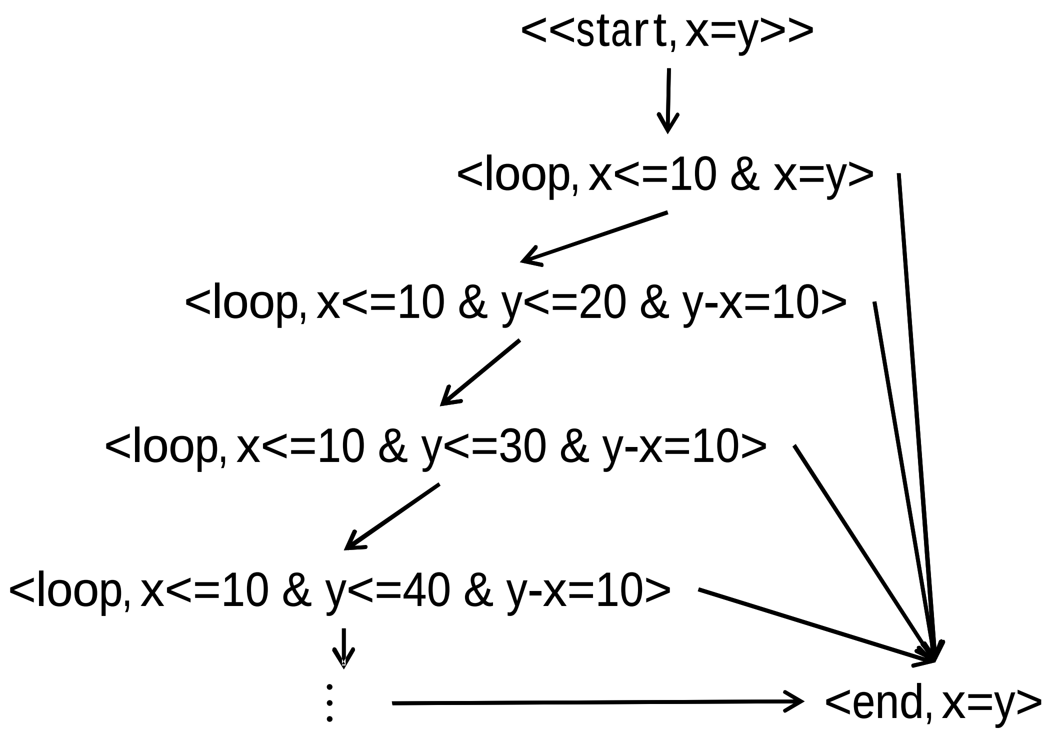
- Theorem 1
  - Soundness: $\langle l_0, \{u_0\}\rangle \rightsquigarrow^* \langle l_f, D_f\rangle$ implies $\langle l_0, u_0\rangle \rightarrow^* \langle l_f, u_f\rangle$ $\forall u_f \in D_f$
  - Completeness: $\langle l_0, u_0\rangle \rightarrow^* \langle l_f, u_f\rangle$ implies $\langle l_0, \{u_0\}\rangle \rightsquigarrow^* \langle l_f, D_f\rangle$ for some D_f such that $u_f \in D_f$



Symbolic Semantics & Verification cont.

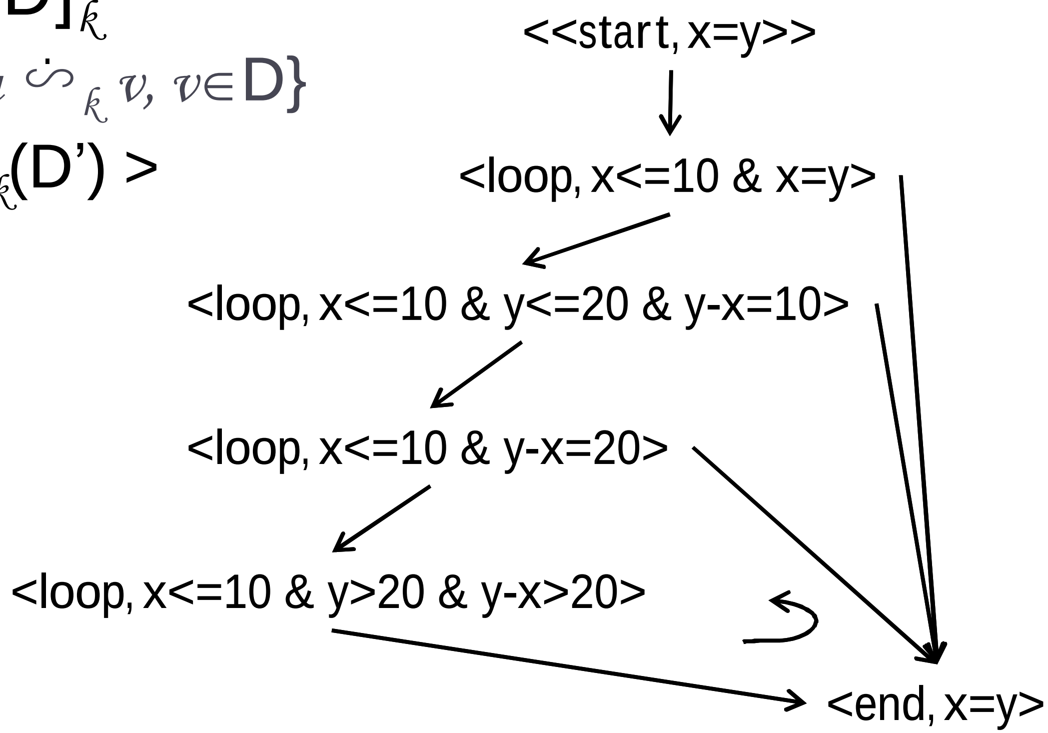
- Problem: $\rightsquigarrow$ infinite
- Solution: normalization





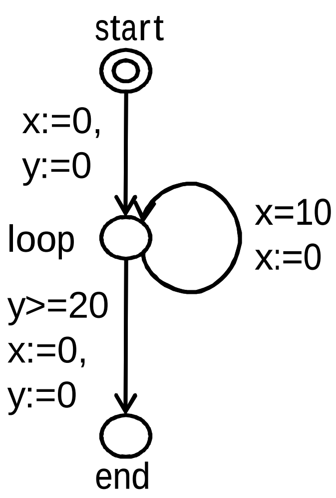
Symbolic Semantics & Verification cont.

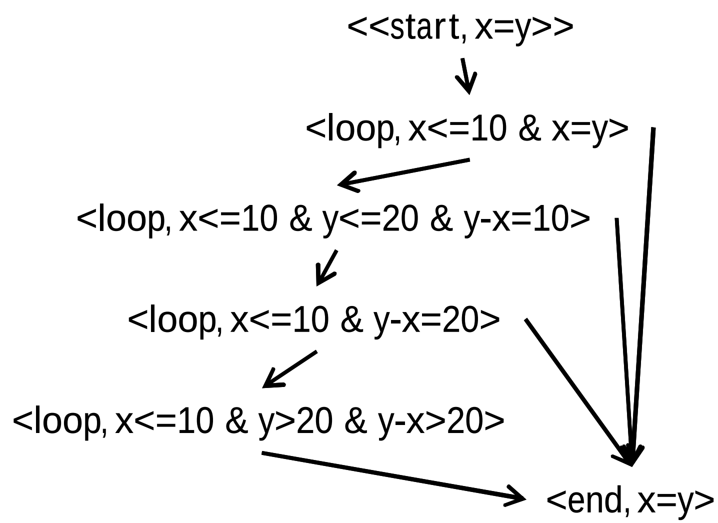
- Diagonal-free automata
  - without difference constraints
- k-Normalization: $[D]_k$
  - $\mathrm{norm}_k(D) = \{ u \mid u \sim_k v,\ v \in D\}$
- $\langle \ell, D\rangle \rightsquigarrow_k \langle \ell, \mathrm{norm}_k(D')\rangle$
  - if $\langle \ell, D\rangle \rightsquigarrow \langle \ell, D'\rangle$



Symbolic Semantics & Verification cont.

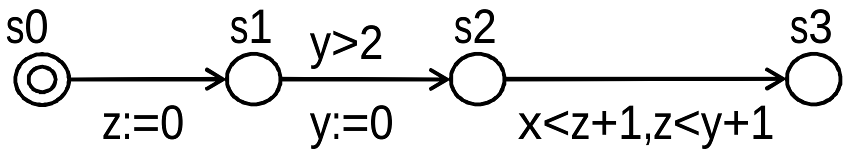
- Theorem 2
  - Soundness: $\langle \ell_0, \{u_0\}\rangle \rightsquigarrow^*_k \langle \ell_f, D_f\rangle$ implies $\langle \ell_0, u_0\rangle \rightarrow^* \langle \ell_f, u_f\rangle$ $\forall u_f \in D_f$ such that $u_f(x) \leq k(x)$ $\forall x$
  - Completeness: $\langle \ell_0, u_0\rangle \rightarrow^* \langle \ell_f, u_f\rangle$ with $u_f(x) \leq k(x)$ $\forall x$ implies $\langle \ell_0, \{u_0\}\rangle \rightsquigarrow^*_k \langle \ell_f, D_f\rangle$ for some D_f such that $u_f \in D_f$
  - Finiteness: $\rightsquigarrow_k$ is finite





Symbolic Semantics & Verification cont.

- Problem: soundness will not hold for TA with difference constraints
- Solution: refined normalization
  - refined region equivalence



s0: x-y=0
y-z=0
y-x=0

s1: x-y=0
z-x<=0
z-y<=0

s2: y-x<-2
y-z<=0
z-x<=0
0-x<-2

s0: x-y=0
y-z=0
y-x=0

s1: x-y=0
z-x<=0
z-y<=0

s2: y-x<-1
y-z<=0
z-x<=0
0-x<-1

Symbolic Semantics & Verification cont.

- Refined Region Equivalence: $u \ \dot{\backsim}_{k,\mathcal{G}}\ v$, *if*
  - $u \ \dot{\backsim}_{k}\ v$
  - $\forall\ g \in \mathcal{G}$, $u \in g$ iff $v \in g$
- $norm_{k,\mathcal{G}}(D) = \{\ u \mid u \ \dot{\backsim}_{k,\mathcal{G}}\ v,\ v \in D\}$
- $\dot{\backsim}_{k,\mathcal{G}}$ induces finitely many equivalence classes
- $<\ell, D> \rightsquigarrow_{k,\mathcal{G}} <\ell, norm_{k,\mathcal{G}}(D') >$
  - if $<\ell, D> \rightsquigarrow <\ell, D'>$

Symbolic Semantics & Verification cont.

- Theorem 3
  - Soundness: $\langle \ell_0, \{u_0\} \rangle \rightsquigarrow^*_{k,\mathcal{G}} \langle \ell_f, D_f \rangle$ implies $\langle \ell_0, u_0 \rangle \longrightarrow^* \langle \ell_f, u_f \rangle$ $\forall u_f \in D_f$ such that $u_f(x) \leq k(x)\ \forall x$
  - Completeness: $\langle \ell_0, u_0 \rangle \longrightarrow^* \langle \ell_f, u_f \rangle$ with $u_f(x) \leq k(x)\ \forall x$ implies $\langle \ell_0, \{u_0\} \rangle \rightsquigarrow^*_{k,\mathcal{G}} \langle \ell_f, D_f \rangle$ for some D_f such that $u_f \in D_f$
  - Finiteness: $\rightsquigarrow_{k,\mathcal{G}}$ is finite
- DONE

Symbolic Semantics & Verification cont.

- Symbolic Reachability Analysis
  - computing state-space
  - searching for states
- Algorithm 1
  - Depth-first search

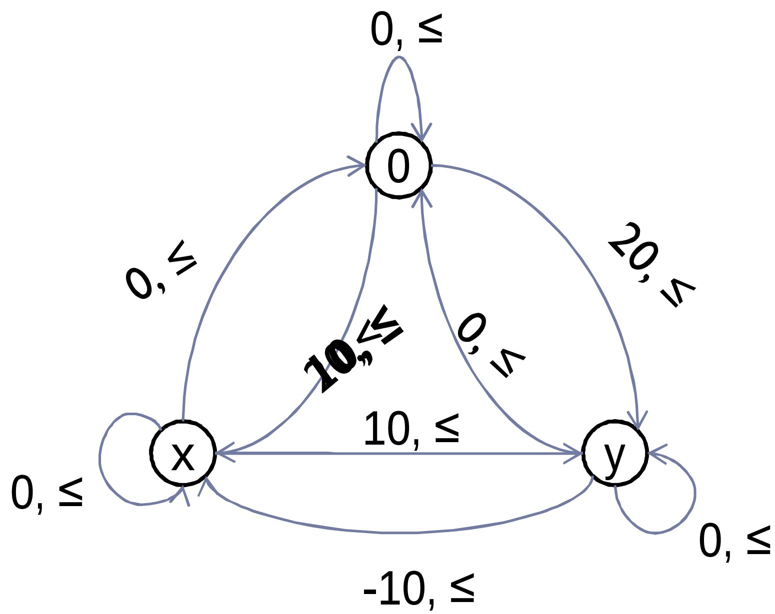
DBM

- Difference Bound Matrix
  - $C_0 = C \cup \{0\}$
  - $D_{xy} = x - y$
  - $\preceq \in \{<, \leq\}$
  - $(n, \preceq) < \infty$
  - $(n, <) < (n, \leq)$
  - Row: lower
  - Column: upper

	0	x	y	z
0	(0,≤)	(0,≤)	(0,≤)	(5,<)
x	(20,<)	(0,≤)	(-10,≤)	∞
y	(20,≤)	(10,≤)	(0,≤)	∞
z	∞	∞	∞	(0,≤)

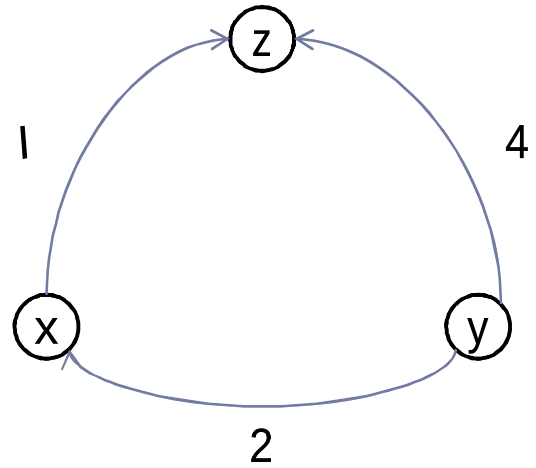
DBM cont.

- Canonical form
  - Tightest constraint on each clock difference
  - Using shortest path algorithm(Floyd-Warshall alg.)
  - Desirable to preserve canonical form



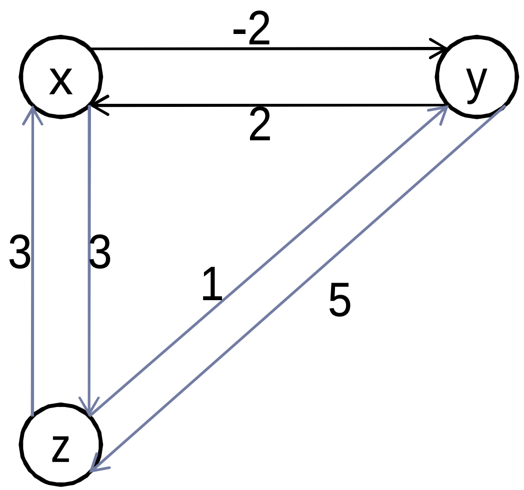
DBM cont.

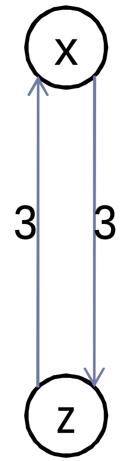
- Minimal Constraint Systems
  - Zero cycle
    - Sum of weights is 0
  - Without zero cycles
    - Safe to remove all redundant edges



DBM cont.

- Minimal Constraint Systems
  - With zero cycles
    - To partition





DBM cont.

- Basic operations
  - Property-checking
    - consistent (D)
    - relation (D, D')
    - satisfied (D, $x_i - x_j \preceq m$)
  - Transformations
    - up(D)
    - down(D)
    - and(D , $x_i - x_j \preceq b$)
    - free(D, x)
    - reset(D, x:=m)
    - copy(D, x:=y)
    - shift(D, x:=x+m)

	0	x	y	z
0	(0,≤)	(0,≤)	(0,≤)	(5,<)
x	(20,<)	(0,≤)	(-10,≤)	∞
y	(20,≤)	(10,≤)	(0,≤)	∞
z	∞	∞	∞	(0,≤)

DBM cont.

- Zone-Normalization
  - $norm_k(D)$
    - remove $x-y \preceq m$ such that $(m, \preceq) > (k(x), \leq)$
    - replace $x-y \preceq m$ such that $(m, \preceq) < (-k(y), <)$ with $(-k(y), <)$
    - NOT preserve the canonical form
    - solution: run Floyd-Warshall algorithm

	0	x	y	z
0	(0,≤)	(0,≤)	(0,≤)	(5,<)
x	(20,<)	(0,≤)	(-10,≤)	∞
y	(20,≤)	(10,≤)	(0,≤)	∞
z	∞	∞	∞	(0,≤)

DBM cont.

- Zone-Normalization
  - $norm_{k,G}(D)$
    - Collect $\mathcal{G}_{unsat}=\{g \mid g \wedge D = 0\} \cup \{\neg g \mid \neg g \wedge D = 0\}$,
    - Compute $norm_{k}(D)$
    - Compute $norm_{k}(D) \wedge \neg \mathcal{G}_{unsat}$
    - Collect $\mathcal{G}_{split}=\{g \mid g \wedge D \neq 0 \; \& \; \neg g \wedge D \neq 0\}$
    - Split D by $\mathcal{G}_{split}$

DBM cont.

- Zones in Memory
  - Storing DBM Elements
    - LSB: (≤ 1) (< 0)
  - Placing DBMs in Memory
    - By row (column)
    - By layer
  - Storing Sparse Zones
    - Nice feature: Check if $D_s \subseteq D_f$
      - not have to compute the full DBM for D_f

Thank you!