Timed Automata

Sayan Mitra

Verifying cyberphysical systems

mitras@illinois.edu

Outline

- Review: Modeling cyber-physical systems
  - Hybrid automata
  - Executions
  - Urgency
  - Zeno
  - Hybrid stability

Hybrid Automaton

$\mathcal{A} = (X, \Theta, A, \mathcal{D}, \mathcal{T})$

- X: set of **state variables**
  - $Q \subseteq val(X)$ set of **states**
- $\Theta \subseteq Q$ set of **start states**
- set of **actions**, A= E ∪ H
- $\mathcal{D} \subseteq Q \times A \times Q$
- $\mathcal{T}$: set of **trajectories** for X which is closed under **prefix, suffix, and concatenation**

Special Classes of Hybrid Automata

Algorithmic analysis of (Alur-Dill's) Timed Automata[1]

Timed Automata are a restricted class of hybrid automata with only clock variables which are amenable to automatic verification

[1] Rajeev Alur and David L. Dill. A theory of timed automata. Theoretical Computer Science, 126:183-235, 1994.

Clocks and Clock Constraints

- A clock variable x is a continuous (analog) variable of type real such that along any trajectory τ of x, for all $t \in \tau.\mathrm{dom}, (\tau \downarrow x)(t) = t$.
- For a set X of clock variables, the set $\Phi(X)$ of integral clock constraints are expressions defined by the syntax:

  $g ::= x \leq q \mid x \geq q \mid \neg g \mid g_1 \wedge g_2$

  where $x \in X$ and $q \in \mathbb{Z}$
- Examples: x = 10; $x \in [2, 5)$; true are valid clock constraints
- What do clock constraints look like geometrically in state space?
- Semantics of clock constraints $[\![g]\!]$

Integral Timed Automata

Definition. An integral timed automaton is a HIOA $\mathcal{A} = \langle V, \Theta, A, \mathcal{D}, \mathcal{T} \rangle$ where

- V = X ∪ $\{l\}$, where X is a set of n clocks and l is a discrete state variable of finite type L
- A is a finite set
- $\mathcal{D}$ is a set of transitions such that
  - The guards are described by clock constraings $\Phi(X)$
  - $\langle x, l \rangle - a \rightarrow \langle x', l' \rangle$ implies either $x' = x$ or $x = 0$
- $\mathcal{T}$ set of clock trajectories for the clock variables in X

Example: Light switch

Math Formulation

automaton Switch

variables

internal x, y:Real := 0, loc: {on,off} := off

transitions

internal push

pre x ≥ 2

eff if loc = on **then** x := 0

else x,y := 0; loc := on

internal pop

pre y = 15 /\ loc = on

eff x := 0

loc := off

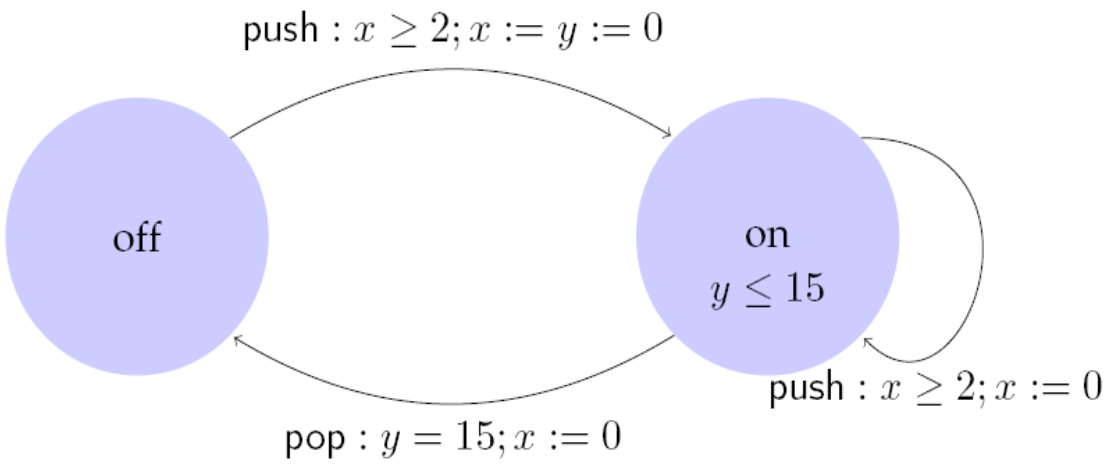
trajectories

invariant loc = on => y ≤ 15

evolve d(x) = 1; d(y) = 1

Description

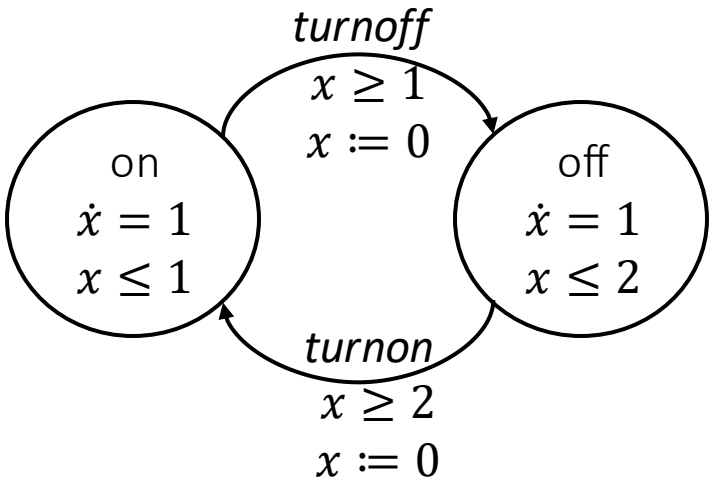
Switch can be turned on whenever at least 2 time units have elapsed since the last turn on. Switches off automatically 15 time units after the last on.



Control State (Location) Reachability Problem

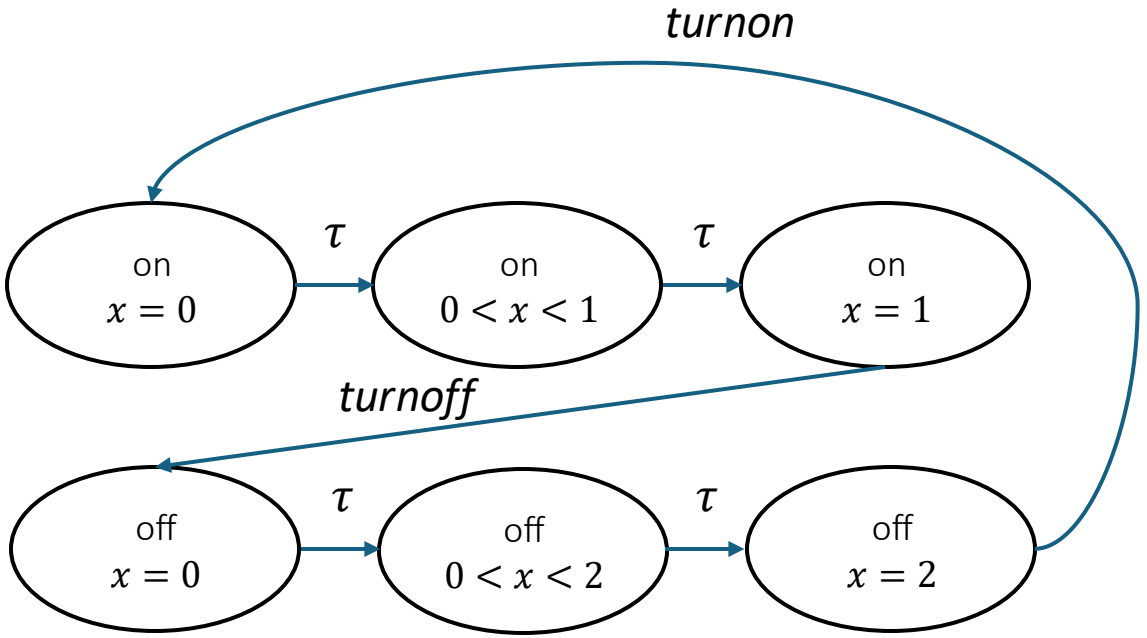
- Given an ITA $\mathcal{A}$, check if a particular (discrete) control state is reachable from the initial states
- Why is control state reachability (CSR) good enough?
- This problem is decidable [Alur Dill]
- Key idea:
  - Construct a finite automaton that is **a time-abstract bisimilar** to the ITA (behaves identically with respect to control state reachability)
  - Check reachability of FSM

An equivalence relation with a finite quotient



- Under what conditions do two states x_1 and x_2 of the automaton $\mathcal{A}$ behave identically with respect to control state reachability (CSR)?
  - When do they satisfy the same set of clock constraints?
  - When would they continue to satisfy the same set of clock constraints?

- If x_1 and x_2 are equivalent, $x_1 \equiv x_2$, from the point of view of reachability then we can use this equivalence relation on the state space Q to define a new automaton $\mathcal{A}'$ with state space $\mathrm{Q}' =$ quotient space $\mathrm{Q}/\equiv$ (which is smaller than Q) and perform reachability analysis on $\mathcal{A}'$



An equivalence relation with a finite quotient

- Under what conditions do two states x_1 and x_2 of the automaton $\mathcal{A}$ behave identically with respect to control state reachability (CSR)?
  - When do they satisfy the same set of clock constraints?
  - When would they continue to satisfy the same set of clock constraints?

- $x_1.\mathbf{loc} = x_2.\mathbf{loc}$ and
- x_1 and x_2 satisfy the same set of clock constraints
  - For each clock $\mathbf{y}$ $\text{int}(x_1.\mathbf{y}) = \text{int}(x_2.\mathbf{y})$ or $\text{int}(x_1.\mathbf{y}) \geq \mathbf{c}_{\mathcal{A}\mathbf{y}}$ and $\text{int}(x_2.\mathbf{y}) \geq \mathbf{c}_{\mathcal{A}\mathbf{y}}$. ($\mathbf{c}_{\mathcal{A}\mathbf{y}}$ **is the maxium clock guard** of $\mathbf{y}$)
  - For each clock $\mathbf{y}$ with $x_1.\mathbf{y} \leq \mathbf{c}_{\mathcal{A}\mathbf{y}}$, $\text{frac}(x_1.\mathbf{y}) = 0$ iff $\text{frac}(x_2.\mathbf{y}) = 0$
  - For any two clocks $\mathbf{y}$ and $\mathbf{z}$ with $x_1.\mathbf{y} \leq \mathbf{c}_{\mathcal{A}\mathbf{y}}$ and $x_1.\mathbf{z} \leq \mathbf{c}_{\mathcal{A}\mathbf{z}}$, $\text{frac}(x_1.\mathbf{y}) \leq \text{frac}(x_1.\mathbf{z})$ iff $\text{frac}(x_2.\mathbf{y}) \leq \text{frac}(x_2.\mathbf{z})$
- Lemma. This is a equivalence relation on val(V) the states of $\mathcal{A}$

- The partition of val(V) induced by this relation is are called clock regions

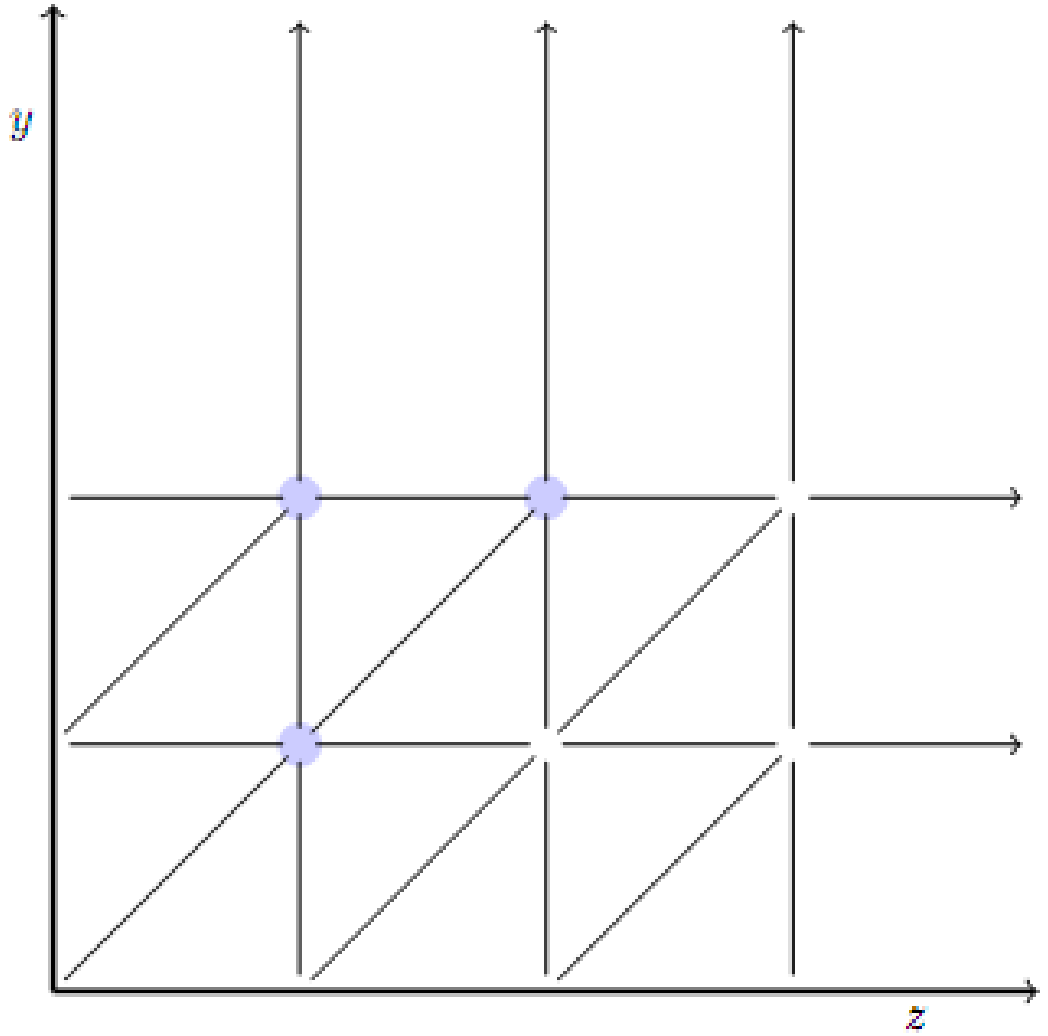
Geometry of clock regions

Example of Two Clocks

X = {y,z}

$c_{\mathcal{A}y}$ = 2

$c_{\mathcal{A}z}$ = 3



Complexity

Lemma. The number of clock regions is bounded by $|X|!\, 2^{|X|} \prod_{z \in X} (2c_{\mathcal{A}z} + 2)$.

Finite

Region automaton $\mathrm{R}(\mathcal{A})$

Given an ITA $\mathcal{A} = \langle V, \Theta, \mathcal{D}, \mathcal{T} \rangle$, we construct the corresponding **Region Automaton** $\mathrm{R}(\mathcal{A}) = \langle Q_R, \Theta_R, D_R \rangle$ such that (i) $\mathrm{R}(\mathcal{A})$ visits the same set of locations (but does not have timing information) and (ii) $\mathrm{R}(\mathcal{A})$ is finite state machine.

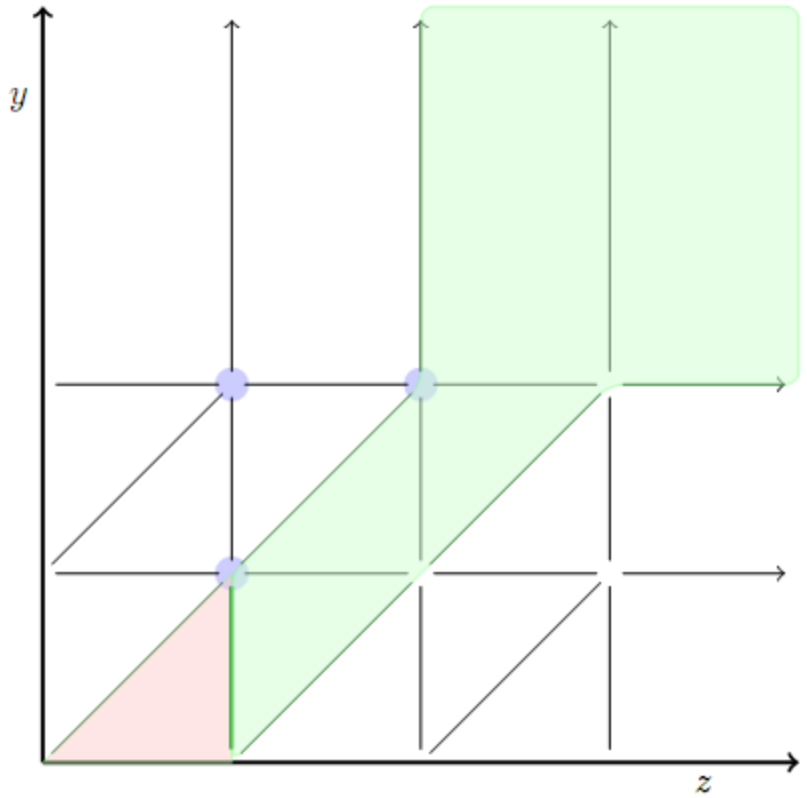
- ITA (clock constants) defines a set of clock regions, say $\mathrm{C}_{\mathcal{A}}$. The set of states $Q_R = C_{\mathcal{A}} \times L$
- $Q_0 \subseteq Q$is the set of states contain initial set Θ of $\mathcal{A}$
- D: We add the transitions between Q (regions)
  - **Time successors**: Consider two clock regions γ and γ', we say that γ' is a time successor of γ if there exits a trajectory of ITA starting from γ that ends in γ'
  - **Discrete transitions**: Same as the ITA

Theorem. A location of ITA $\mathcal{A}$ is reachable iff it is also reachable in $\mathrm{R}(\mathcal{A})$.

(we say that $\mathrm{R}(\mathcal{A})$ is *time abstract bisimilar* to $\mathcal{A}$)

Proof. For each execution of $\mathcal{A}$ inductively construct the corresponding execution of $R(\mathcal{A})$ and vice versa.

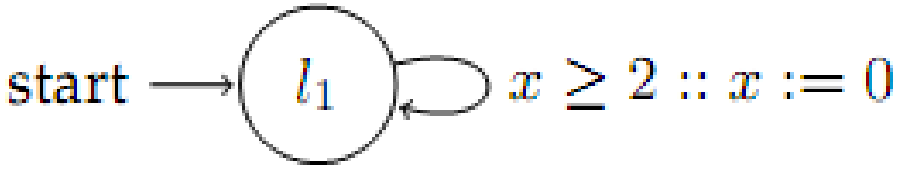
Time successors



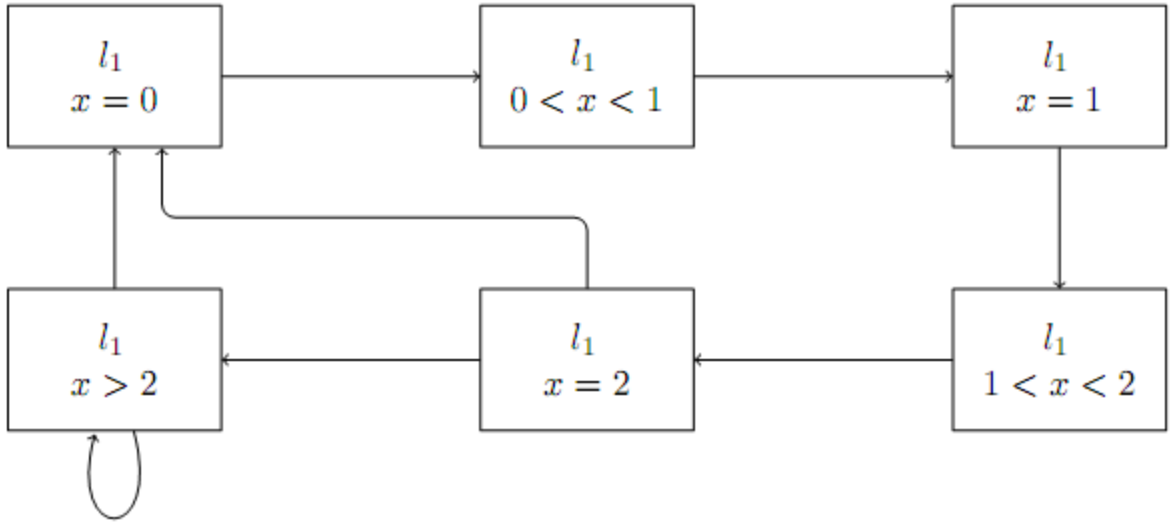
The clock regions in blue are time successors of the clock region in red.

Example 1: Region Automata

ITA

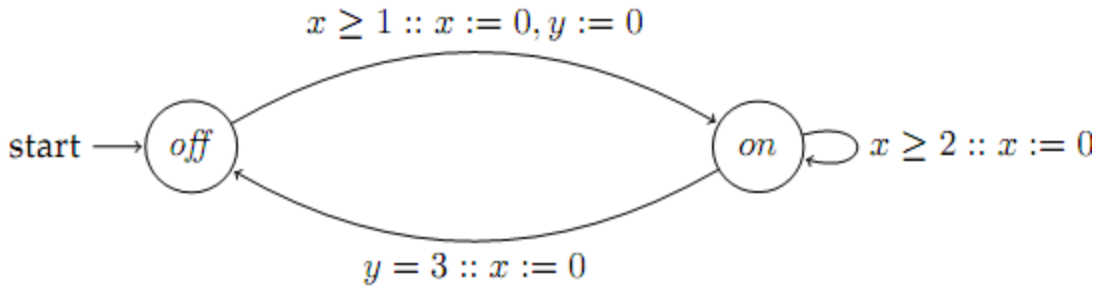


Corresponding FA

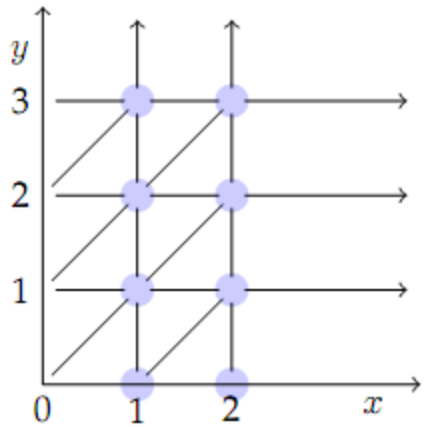


Example 2

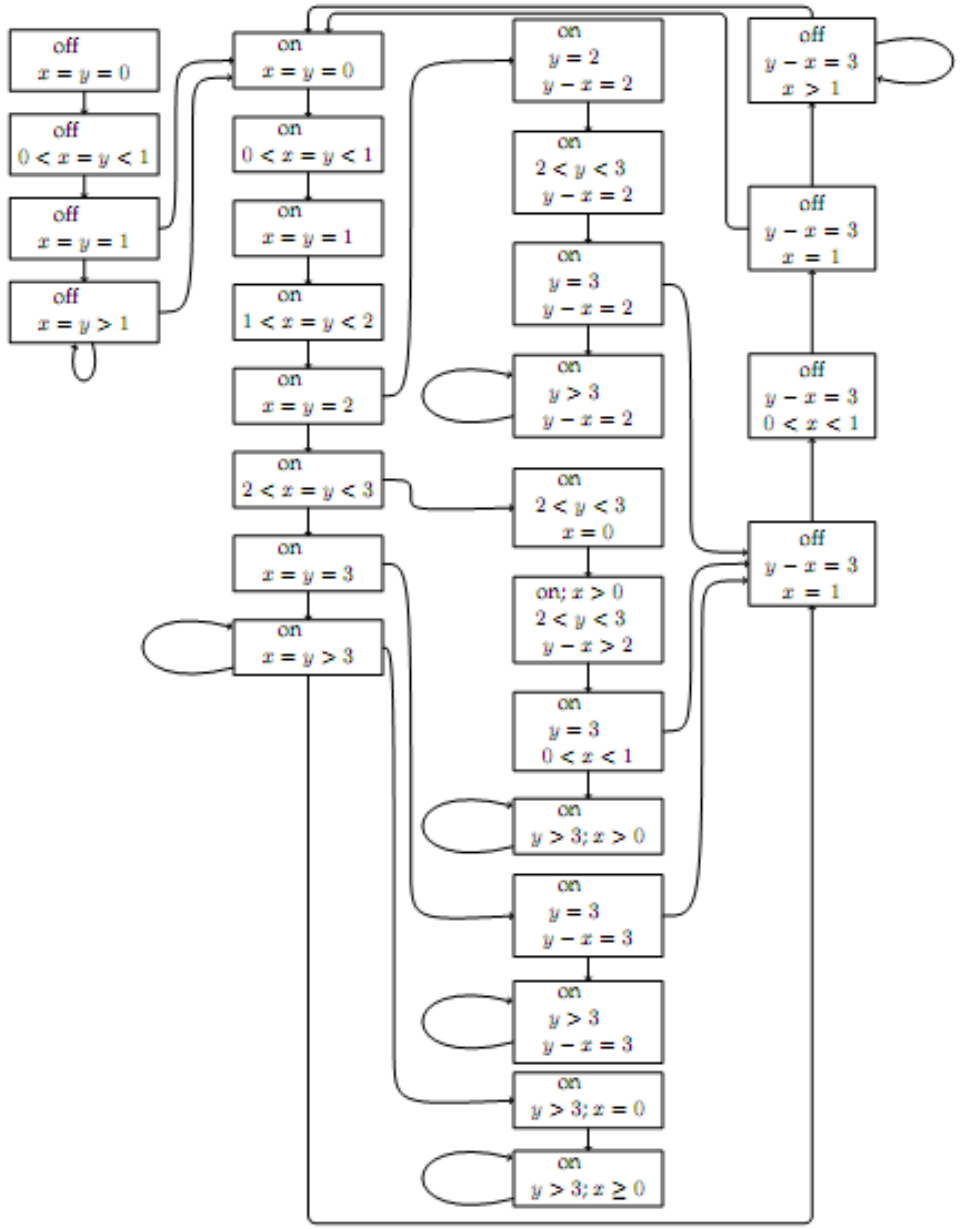
ITA



Clock Regions



Corresponding FA



$$|X|!\, 2^{|X|} \prod_{z \in X} (2c_{\mathcal{A}z} + 2)$$

Drastically increasing with the number of clocks

Summary

- Big idea: If two states behave similarly---simulate each other---then a verification algorithm need not keep track of them separately
  - Related to notion of **indistinguishability**
- Big savings in memory and computation
- Many generalizations
  - Simulation (one way containment) as opposed to bi-simulation
  - Simulation-like relations for stability (not just reachability) and other kinds of requirements
  - Computing simulations offline vs online