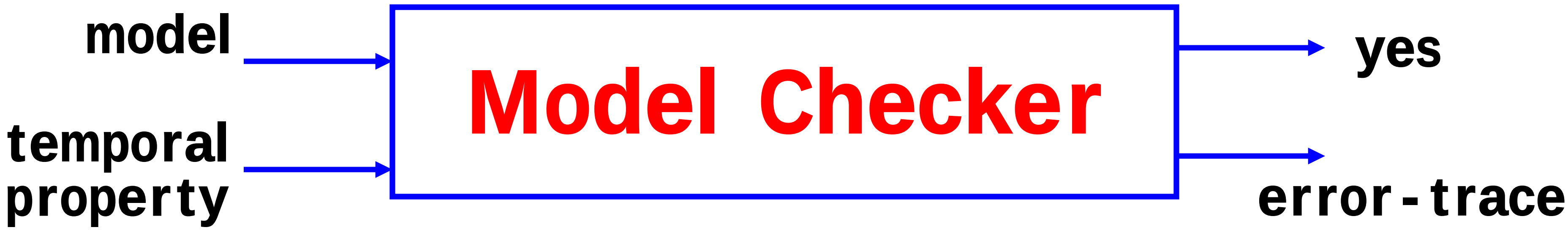
Lecture 2.
Decidability and Verification



Advantages

Automated formal verification, Effective debugging tool

Moderate industrial success

In-house groups: Intel, Microsoft, Lucent, Motorola...

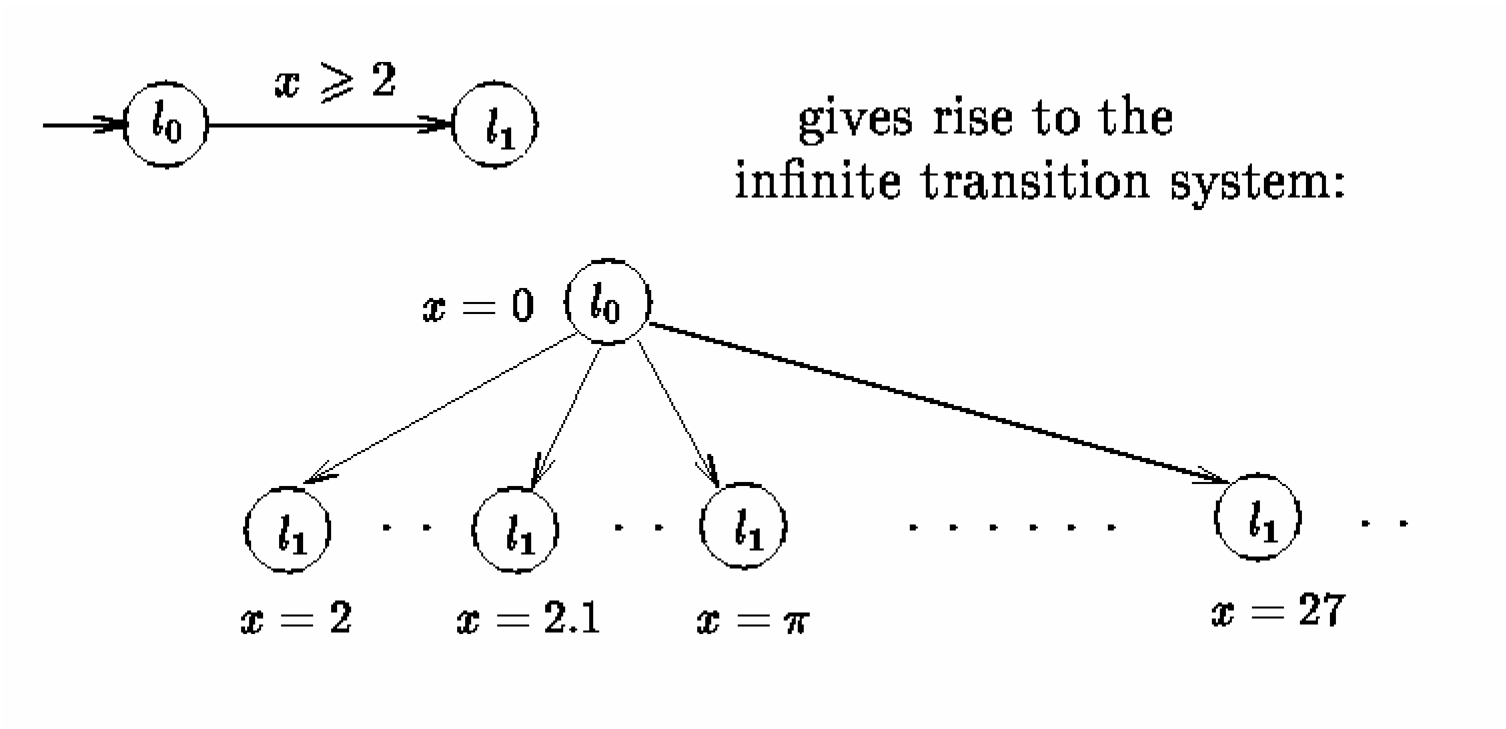
Commercial model checkers: FormalCheck by Cadence

Obstacles

Scalability is still a problem (about 100 state vars)

Effective use requires great expertise

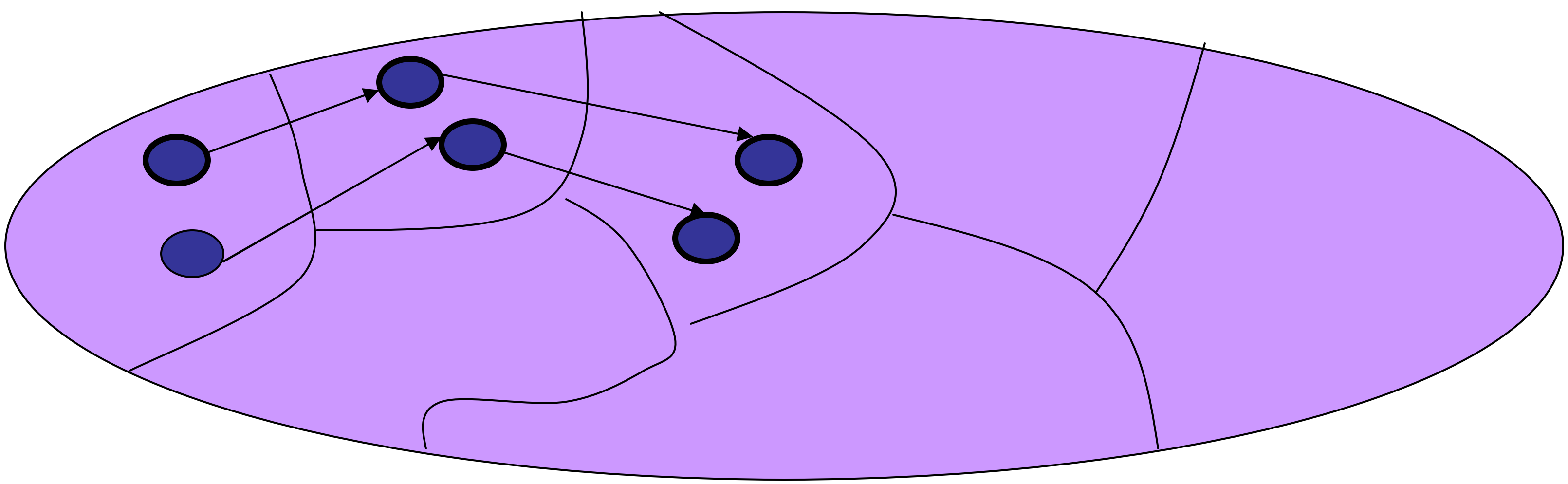
Model Checking of Hybrid Systems



Is finite state analysis possible?
Is reachability problem decidable?

Finite Partitioning

Goal: To partition state-space into finitely many equivalence classes so that equivalent states exhibit similar behaviors



Talk Outline

- ➲ Preliminaries: Transition Systems
- ❑ Timed Automata and Region Graphs
- ❑ Equivalences and Finite Quotients
- ❑ Decidable Problems

Labeled Transition System T

- Set Q of states
- Set I of initial states
- Set L of labels
- Set → of labeled transitions of the form
  q –a-> q’

Partitions and Quotients

- Let $T=(Q,I,L,\rightarrow)$ be a transition system and $\cong$ be a partitioning of Q (i.e. an equivalence relation on Q)
- Quotient $T/\cong$ is transition system:
  1. States are equivalence classes of $\cong$
  2. A state P is initial if it contains a state in I
  3. Set of labels is L
  4. Transitions: P –a-> P' if q-a->q' for some q in P and some q' in P'

Language Equivalence

- Language of T: Set of possible finite strings over L that can be generated starting from initial states
- T and T' are language-equivalent iff they generate the same language
- Roughly speaking, language equivalent systems satisfy the same set of "safety" properties

Bisimulation

- Relation $\cong$ on QXQ' is a bisimulation iff whenever $q \cong q'$ then

  if q-a->u then for some u', $u \cong u'$ and q'-a->u', and

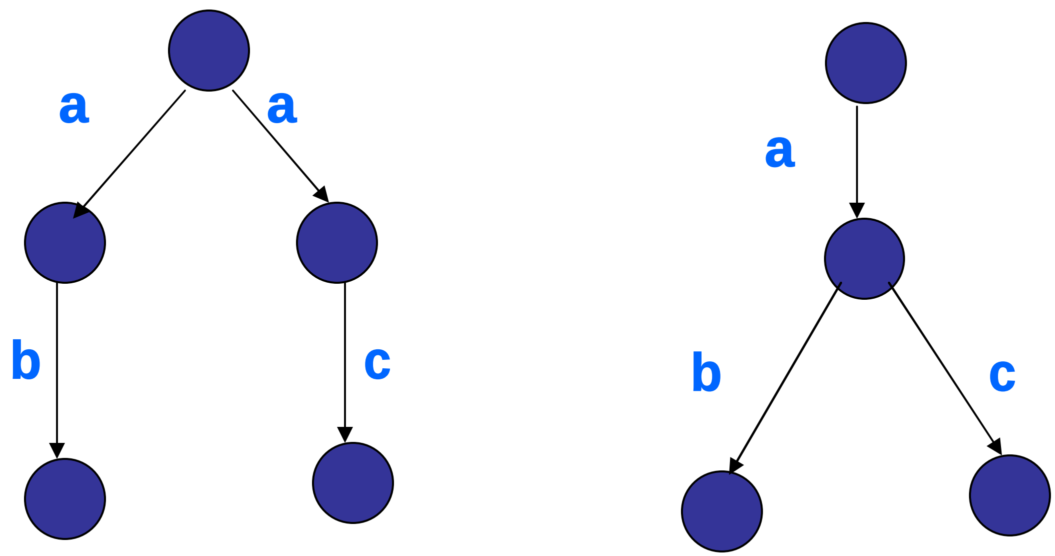
  if q'-a->u' then for some u, $u \cong u'$ and q-a->u.

- Transition systems T and T' are bisimilar if there exists bisimulation $\cong$ on QXQ' such that

  For every q in I, there is q' in I', $q \cong q'$ and vice versa

- Many equivalent characterizations (e.g. game-theoretic)
- Roughly speaking, bisimilar systems satisfy the same set of branching-time properties (including safety)

Bisimulation Vs Language equivalence



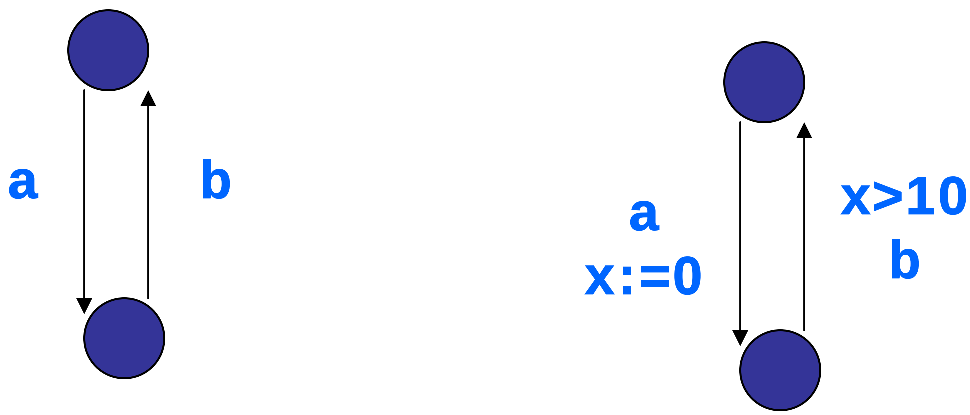
Language equivalent but not bisimilar

Bisimilarity -> Language equivalence

Timed Vs Time-Abstract Relations

- Transition system associated with a timed/hybrid automaton:
  - Labels on continuous steps are delays in R
  - Actual delays are suppressed (all continuous steps have same label): **Time-abstract**
- Two versions of language equivalence and two versions of bisimulation
- Time-abstract relations enough to capture untimed properties (e.g. reachability, safety)

Time-abstract Vs Timed



Time-abstract equivalent but not timed equivalent

Timed equivalence -> Time-abstract equivalence

Talk Outline

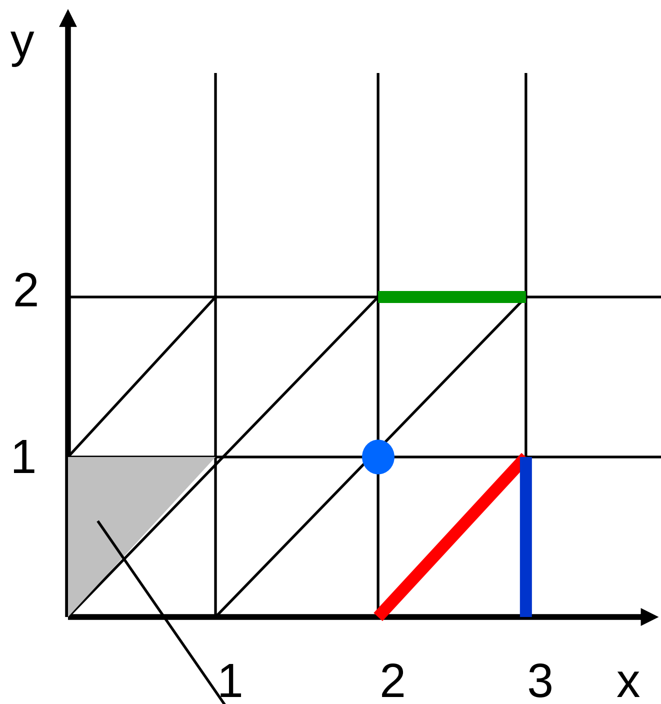
- ✓ Preliminaries: Transition Systems
- ➲ Timed Automata and Region Graphs
- ❑ Equivalences and Finite Quotients
- ❑ Decidable Problems

Timed Automata (Recap)

- Only continuous variables are timers
- Invariants and Guards: x<const, x>=const
- Actions: x:=0
- Can express lower and upper bounds on delays

Regions

Finite partitioning of state space

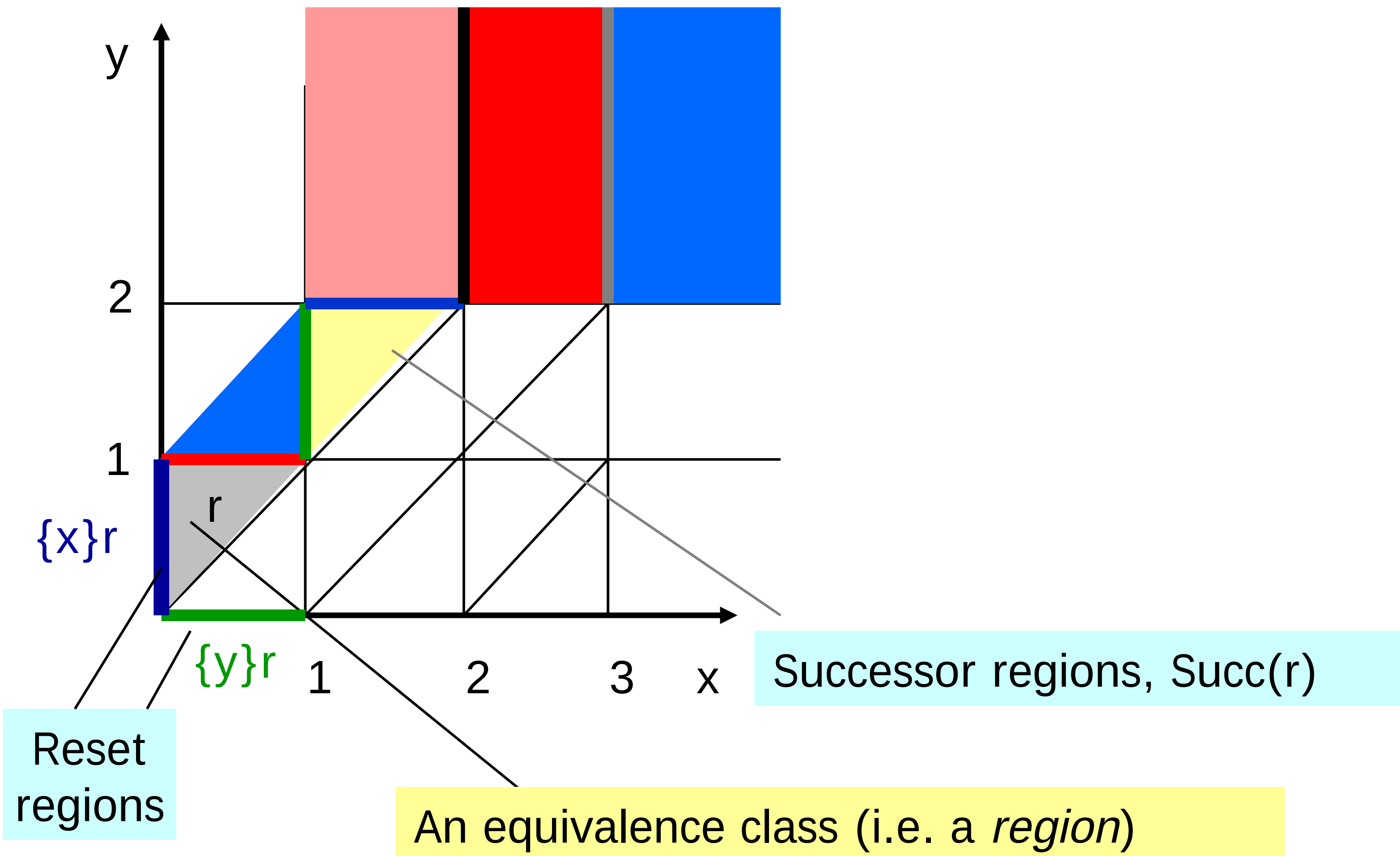


Definition

$w \cong w'$ iff they satisfy the same set of constraints of the form
$x_i < c$, $x_i = c$, $x_i - x_j < c$, $x_i - x_j = c$
for $c <=$ largest const relevant to x_i

An equivalence class (i.e. a *region*)
in fact there is only a *finite* number of regions!!

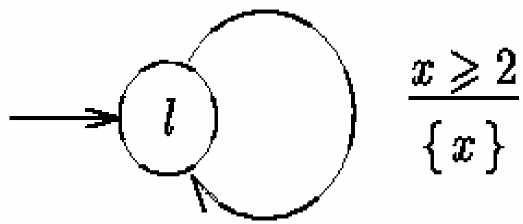
Region Operations



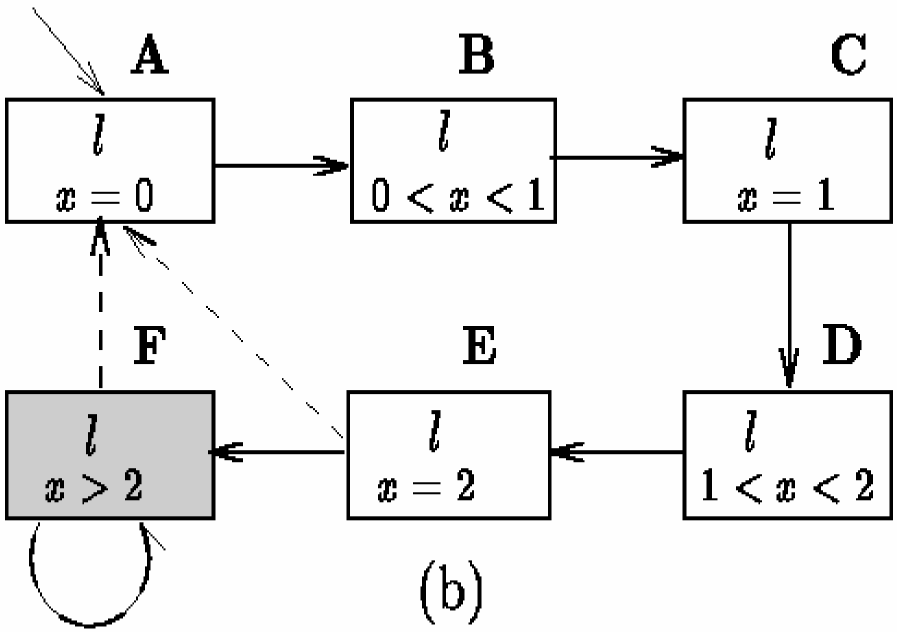
Properties of Regions

- The region equivalence relation $\cong$ is a **time-abstract bisimulation**:
  - Action transitions: If $w \cong v$ and (l,w) -a-> (l',w') for some w', then $\exists$ v' $\cong$ w' s.t. (l,v) -a-> (l',v')
  - Delay transitions: If $w \cong v$ then for all real numbers d, there exists d' s.t. $w+d \cong v+d'$
- If $w \cong v$ then (l,w) and (l,v) satisfy the same temporal logic formulas

Region graph of a simple timed automata



(a)



(b)

Region Graphs (Summary)

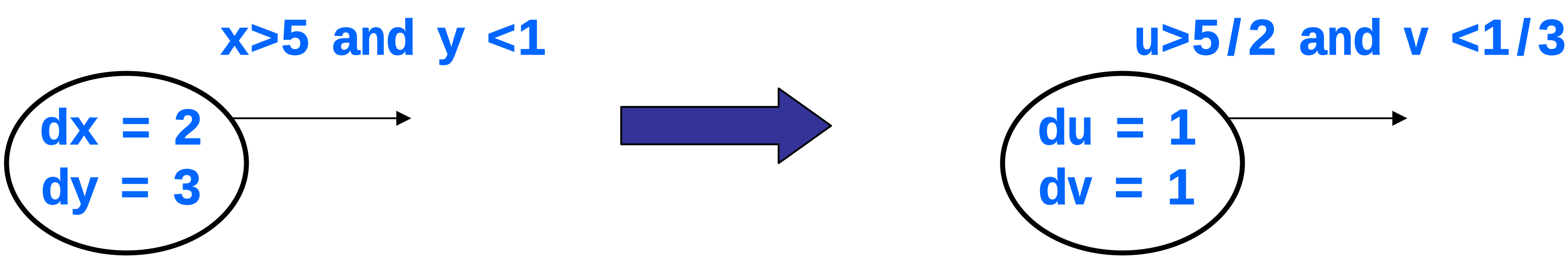
- Finite quotient of timed automaton that is time-abstract bisimilar
- Number of regions: (# of locations) times (product of all constants) times (factorial of number of clocks)
- Precise complexity class of reachability problem: PSPACE (basically, exponential dependence of clocks/constants unavoidable)

Talk Outline

- ✓ Preliminaries: Transition Systems
- ✓ Timed Automata and Region Graphs
- ➲ Equivalences and Finite Quotients
- ❑ Decidable Problems

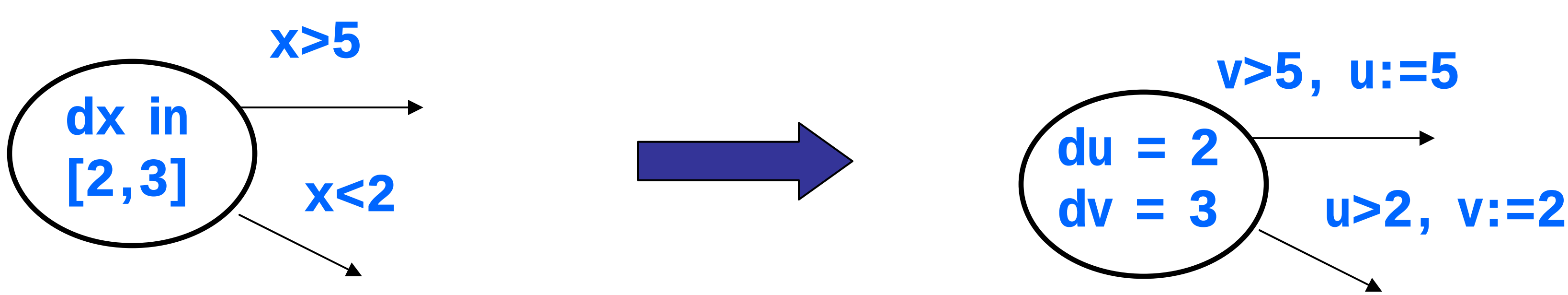
Multi-rate Automata

- Modest extension of timed automata
  - Dynamics of the form dx = const (rate of a clock is same in all locations)
  - Guards and invariants: x < const, x > const
  - Resets: x := const
- Simple translation to timed automata that gives time-abstract bisimilar system by scaling

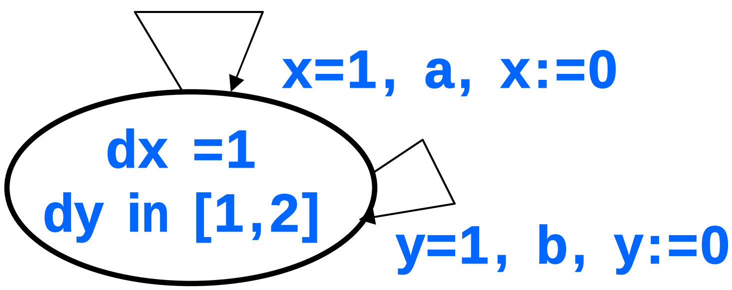


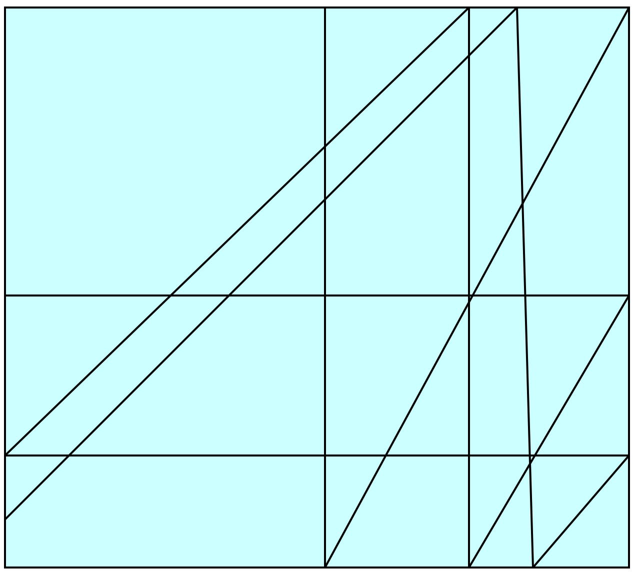
Rectangular Automata

- Interesting extension of timed automata
  - Dynamics of the form dx in const interval (rate-bounds of a clock same in all locations)
  - Guards/invariants/resets as before
- Translation to multi-rate automata that gives time-abstract language-equiv system



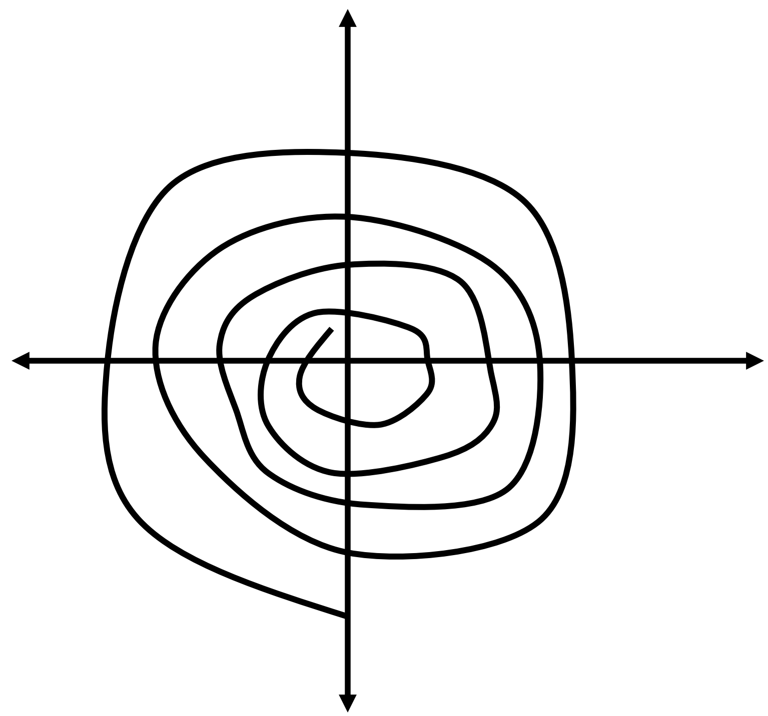
Rectangular Automata may not have finite bismilar quotients!





Continuous systems

- Given an initial partitioning P of R^k and continuous dynamics $dX=F(X)$, is there a refinement of P that is time-abstract bisimilar to original system?
- Counter-example: Spiral. Initial partition: $-5<=x<0$ and $0<x<=+5$



O-minimal Structures

- A structure over R is order-minimal if every definable subset is a finite union of points and open intervals
- O-minimal structures
  - R with <, +, -, 0, 1 (polyhedral sets)
  - R with <, +, -, *, e^x, 0, 1 (semialgebraic sets and exponential trajectories)
  - And many more such as sub-analytic

O-minimal Hybrid Systems

- Guards, Flows, Invariants definable in the same o-minimal structure
- Edges reset all variables (to constants or intervals)
- Thm: O-minimal hybrid systems have finite time-abstract bisimilarity quotients

Talk Outline

- ✓ Preliminaries: Transition Systems
- ✓ Timed Automata and Region Graphs
- ✓ Equivalences and Finite Quotients
- ➲ Decidable Problems

Decidable Problems

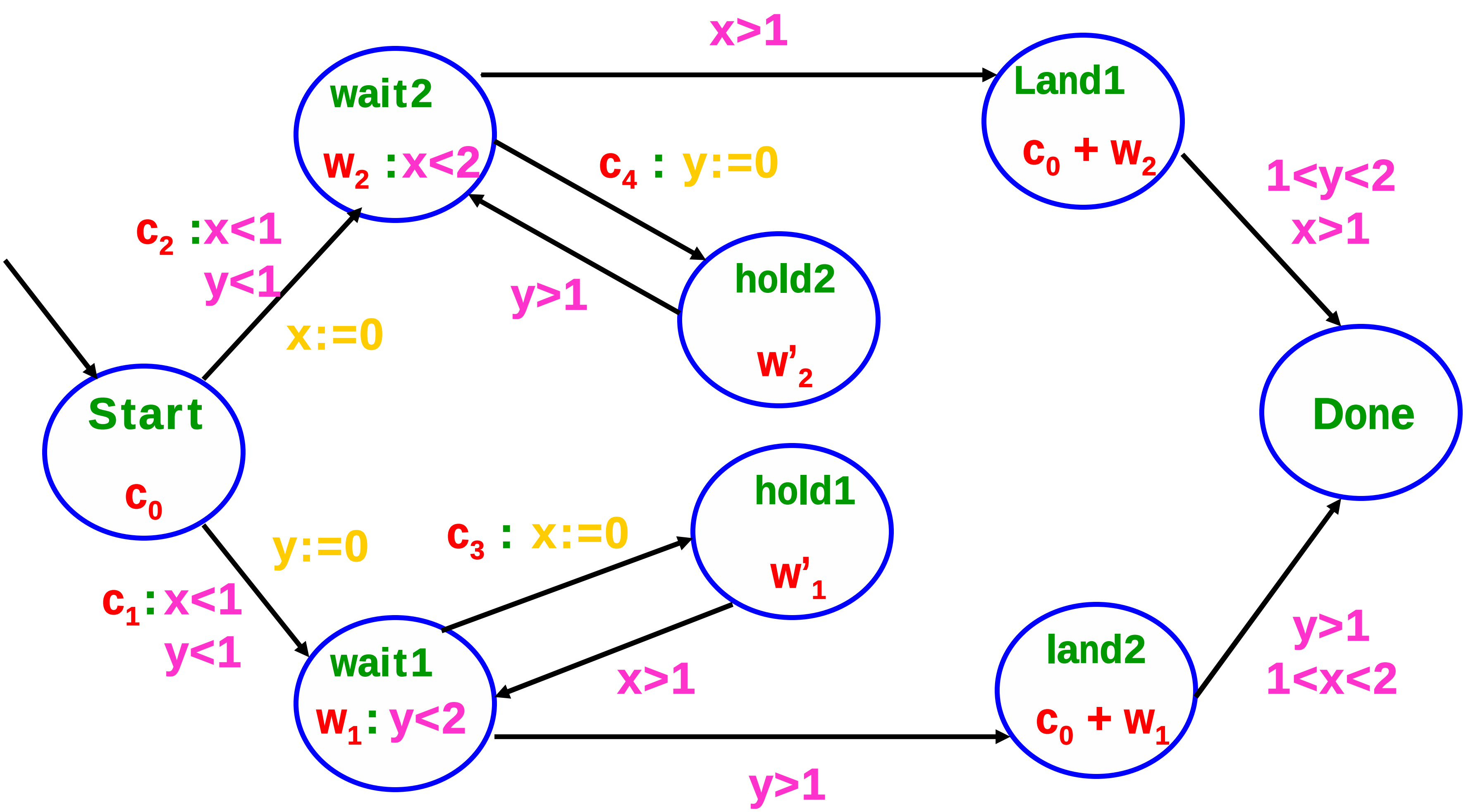
- Model checking branching-time properties of timed automata
- Reachability in rectangular automata
- Timed bisimilarity: are given two timed automata bisimilar?
- Optimization: Compute shortest paths (e.g. minimum time reachability) in timed automata with costs on locations and edges
- Controller synthesis: Computing winning strategies in timed automata with controllable and uncontrollable transitions

Undecidable Reachability Problems

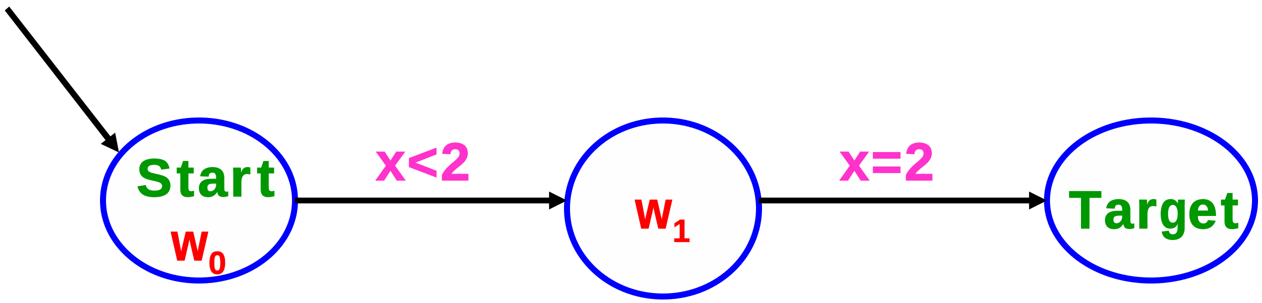
- Timed automata + linear expressions as guards
- Multi-rate automata with comparisons among clocks as guards
- Timed automata + stop-watches (i.e. clocks that can have rates 0 or 1)

Many such results
Proofs by encoding Turing machines/2-counter machines
Sharp boundary for decidability understood

Air-traffic Control Problem as Weigted timed automaton

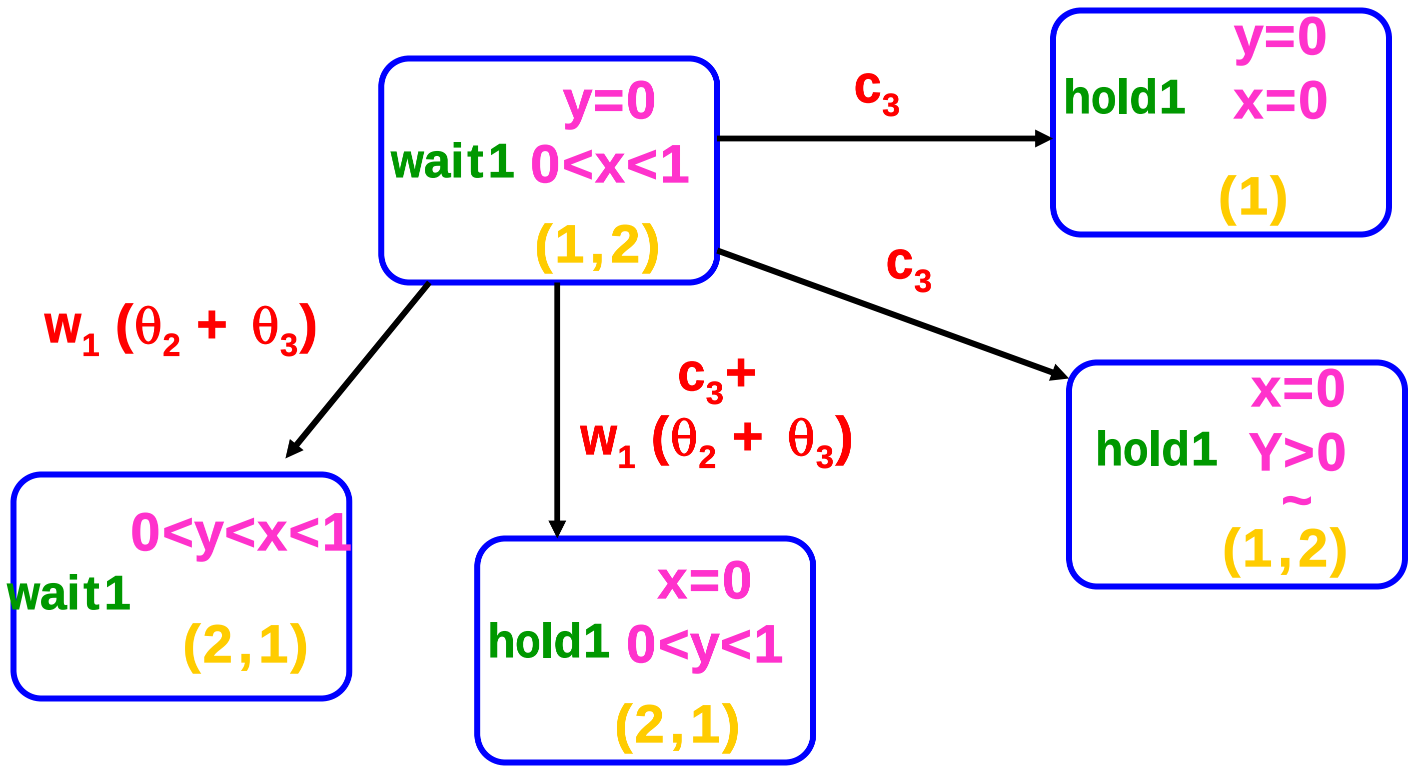


Shortest Paths in WTA



- Optimum solution may only be a limit
- Region graph construction not enough
- Algorithm
  1. Reduce to Parametric Shortest Path Problem on graphs (PSP)
  2. Solve PSP

From WTA to Weighted Graphs



- **Augmented Region Automaton**
  - Regions are split in *boundary* sub-regions

Summary

- Decidability only when simple dynamics or decoupled dynamics
- Theory of equivalences useful in understanding structural properties