A Unifying Approach to Decide Relations for Timed Automata and their Game Characterization

Shibashis Guha[1], Krishna S[2], Chinmay Narayan[1], S. Arun-Kumar[1]

[1]Department of Computer Science & Engineering
Indian Institute of Technology Delhi

[2]Department of Computer Science & Engineering
Indian Institute of Technology Bombay

August 26, 2013

Overview

- Contribution, Related Work
- Timed Automata and Definitions
- Relations over Timed Automata
- Our Approach for Deciding the Relations
- Game Characterization
- Future Work and Discussions

Contribution, Related Work

Related Work

- Deciding timed bisimulation using a product construction on regions : Shown in EXPTIME (Čerāns, ’92)
- Deciding timed bisimulation using a product construction on zones : Shown in EXPTIME, running time improvement over product construction on regions. (Weise and Lenzkes, ’97)
- EXPTIME hardness for any relation between timed simulation and timed bisimulation (Laroussinie and Schnoebelen, ’00)
- Time abstracted bisimulation : Shown in EXPTIME (Larsen and Yi, ’97)
- Different quotienting of zone graph for different time abstracted relations : (Tripakis and Yovine, ’01)

Our Contribution

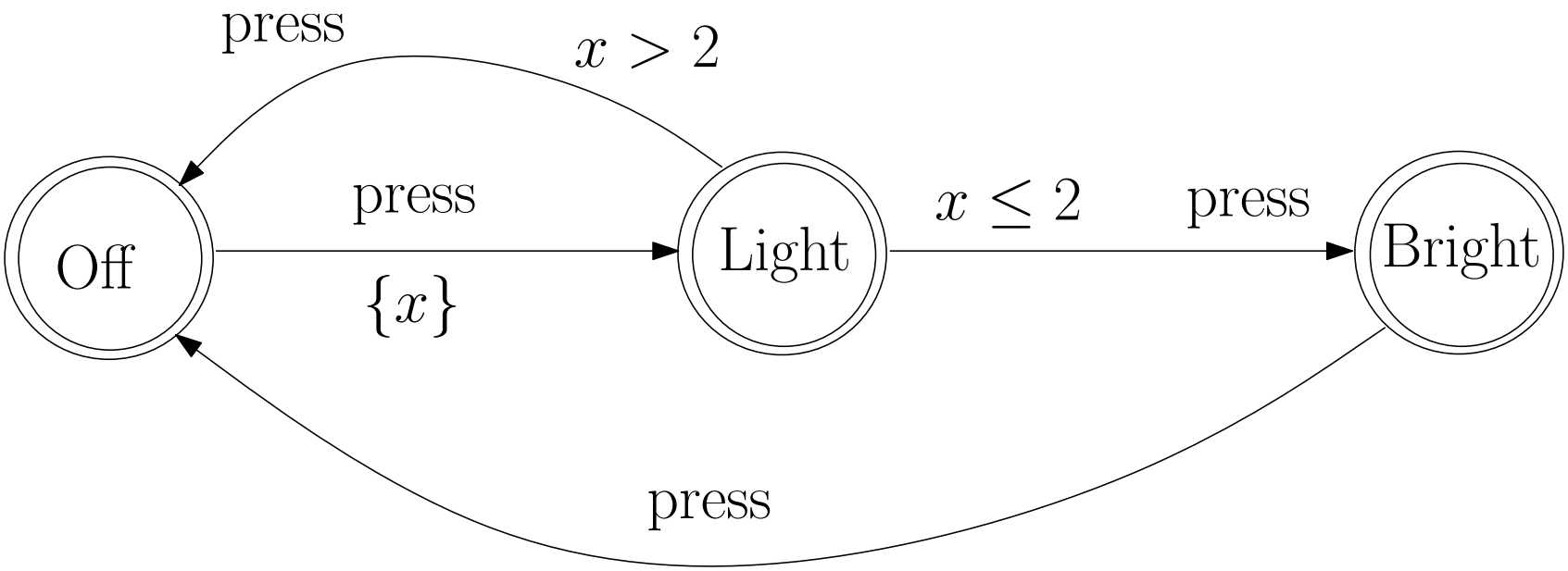
- Unifying approach for deciding several relations
- Common zone graph construction
- Game semantics for these relations
- Infinite hierarchy of these games can help in studying and defining several new relations

Timed Automata and Related Definitions

Timed Automata

- Formalism to model *real time* systems : automated analysis of real time systems
- Traditional *model checking* tools do not admit modelling of time.
- Some behavioural equivalences involving real time can also be decided
- Introduced by Alur and Dill in 1990 (Automata for Modelling Real Time Systems), 1994 (A Theory of Timed Automata).
- Subsequent *theory* and *tool* research - Tools: Uppaal and Kronos.

An Example



Non-deterministic automaton with finite number of *real valued* clock variables

Timed automaton Semantics: Timed Labeled Transition System (TLTS)

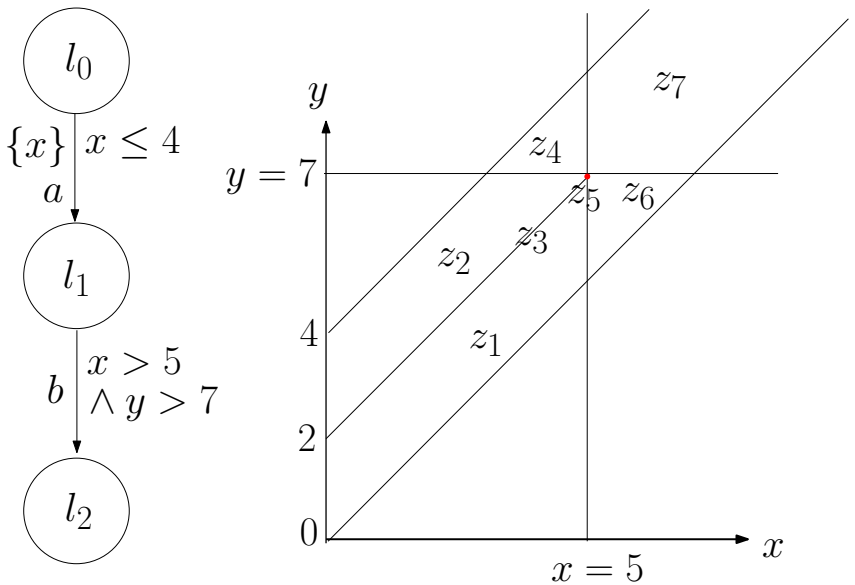
- Infinite transition graph structure
- Nodes are timed automaton states or configurations; tuple (ℓ, v)
- Two types of transitions

  Discrete transitions: $a \in Act$: $(\ell, v) \xrightarrow{a} (\ell', v')$ if there is an edge $(\ell \xrightarrow{g,a,r} \ell') \in E$ and $v \models g, v' = v[r]$

  Delay transitions: $d \in \mathbb{R}_{\geq 0}$: $(\ell, v) \xrightarrow{d} (\ell, v + d)$.

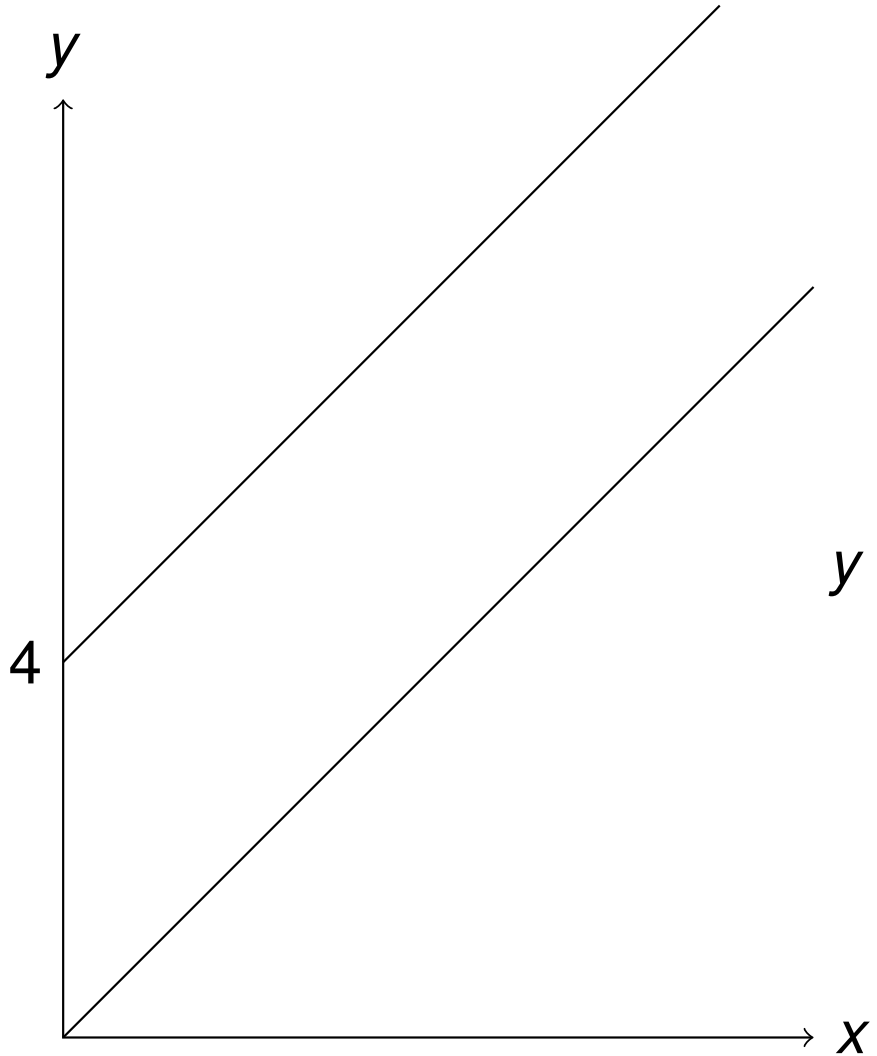
A Few Definitions

- **zone**: A zone $z = \{v \in \mathbb{R}^{|C|}_{\geq 0} \mid v \models \gamma\}$, where γ is of the form $\gamma ::= x \sim c \mid x - y \sim c \mid g \wedge g$, where $c \in \mathbb{Z}$, $x, y \in C$ and $\sim \in \{\leq, <, =, >, \geq\}$.



- $z_1 \cup z_2 \cup z_3 \cup z_5$ is a zone
- $z_1 \cup z_2 \cup z_3 \cup z_5 \cup z_4 \cup z_6$ is NOT a zone
- $z\uparrow$ denotes the future of the zone z.
  $z\uparrow = \{v + d \mid v \in z, d \geq 0\}$ is the set of all valuations reachable from z by time elapse.

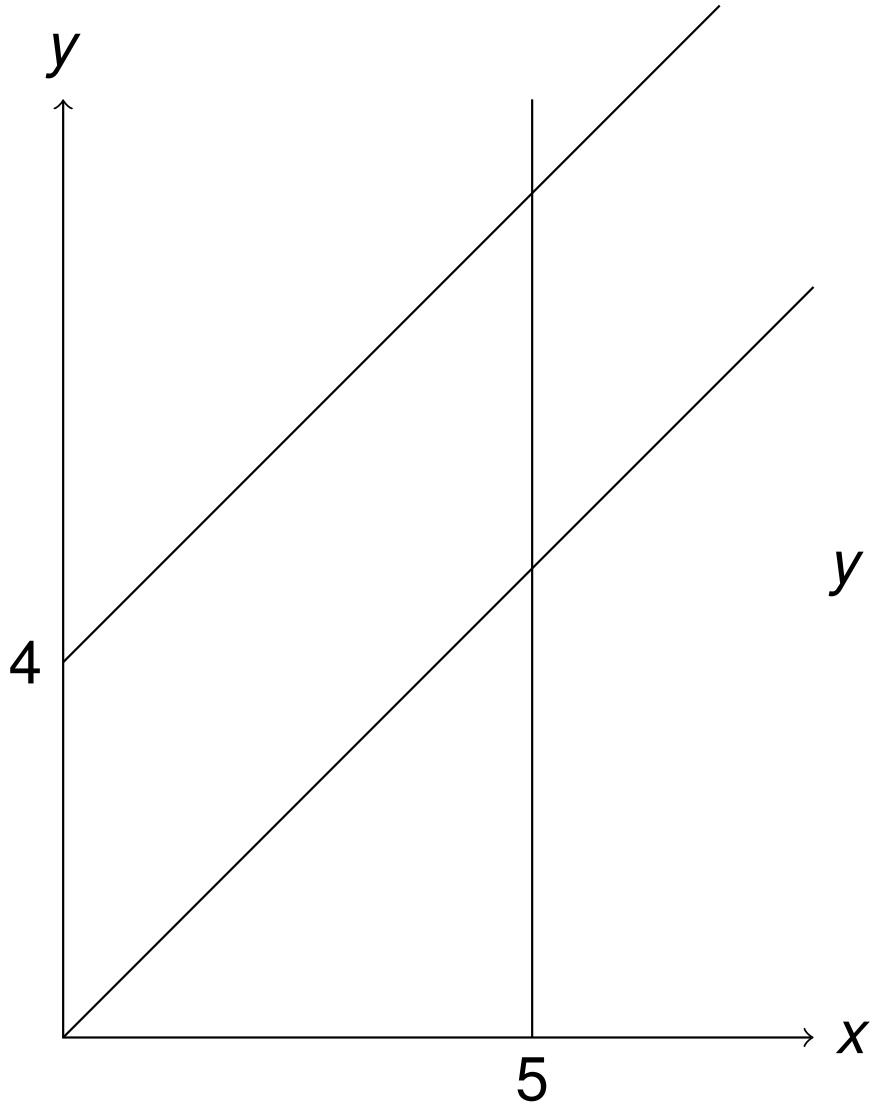
Canonical Decomposition and Prestability



$$y - x \geq 4 \ \wedge \ x \geq 0 \ \wedge \ y \geq 0$$

$$x > 5 \ \wedge \ y > 7$$

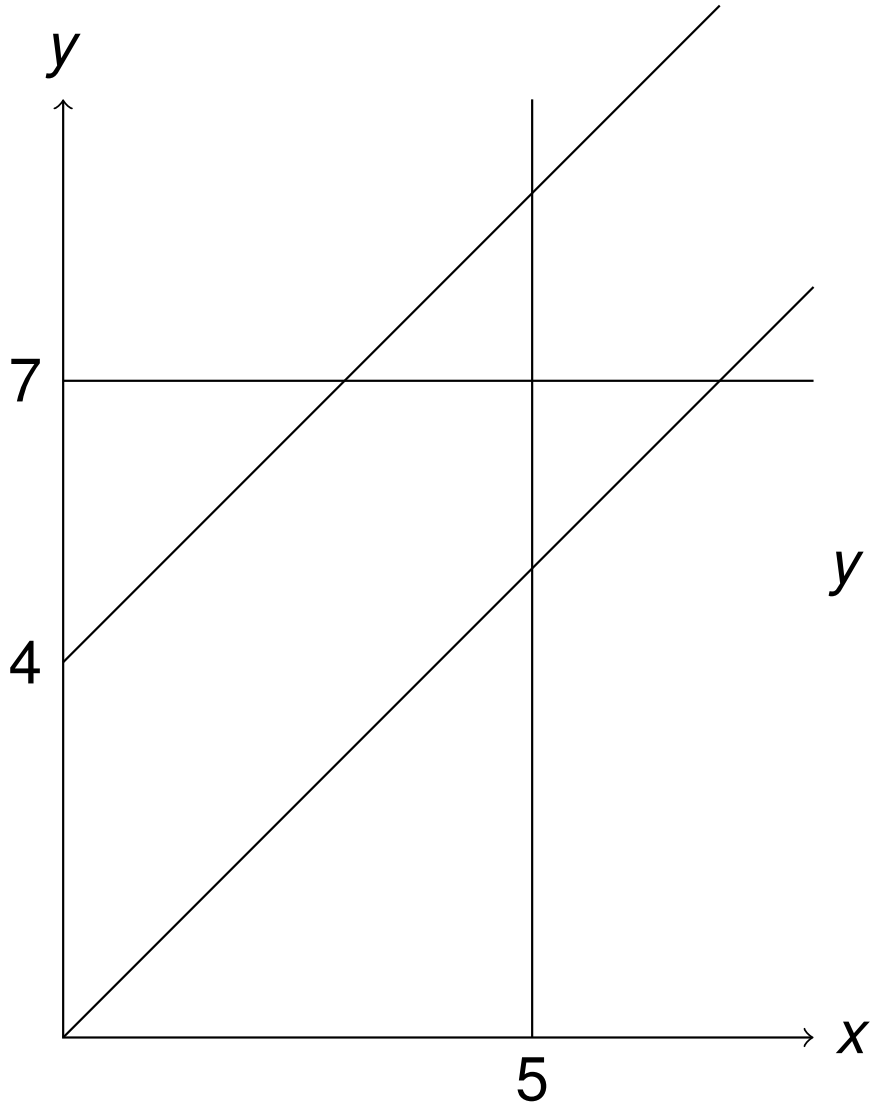
Canonical Decomposition and Prestability



$$y - x \geq 4 \ \wedge \ x \geq 0 \ \wedge \ y \geq 0$$

$$x > 5 \ \wedge \ y > 7$$

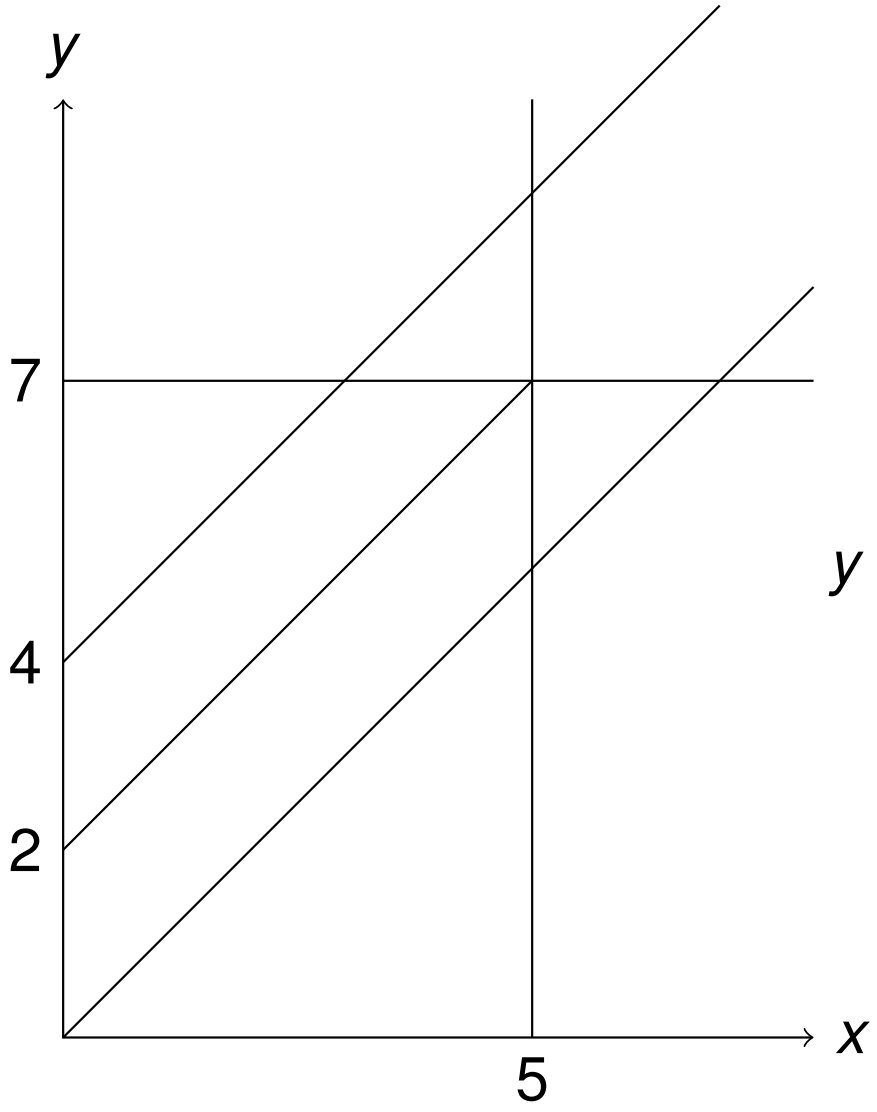
Canonical Decomposition and Prestability



$$y - x \geq 4 \ \wedge \ x \geq 0 \ \wedge \ y \geq 0$$

$$x > 5 \ \wedge \ y > 7$$

Canonical Decomposition and Prestability



$$y - x \geq 4 \ \wedge \ x \geq 0 \ \wedge \ y \geq 0$$

$$x > 5 \ \wedge \ y > 7$$

Zone Graph

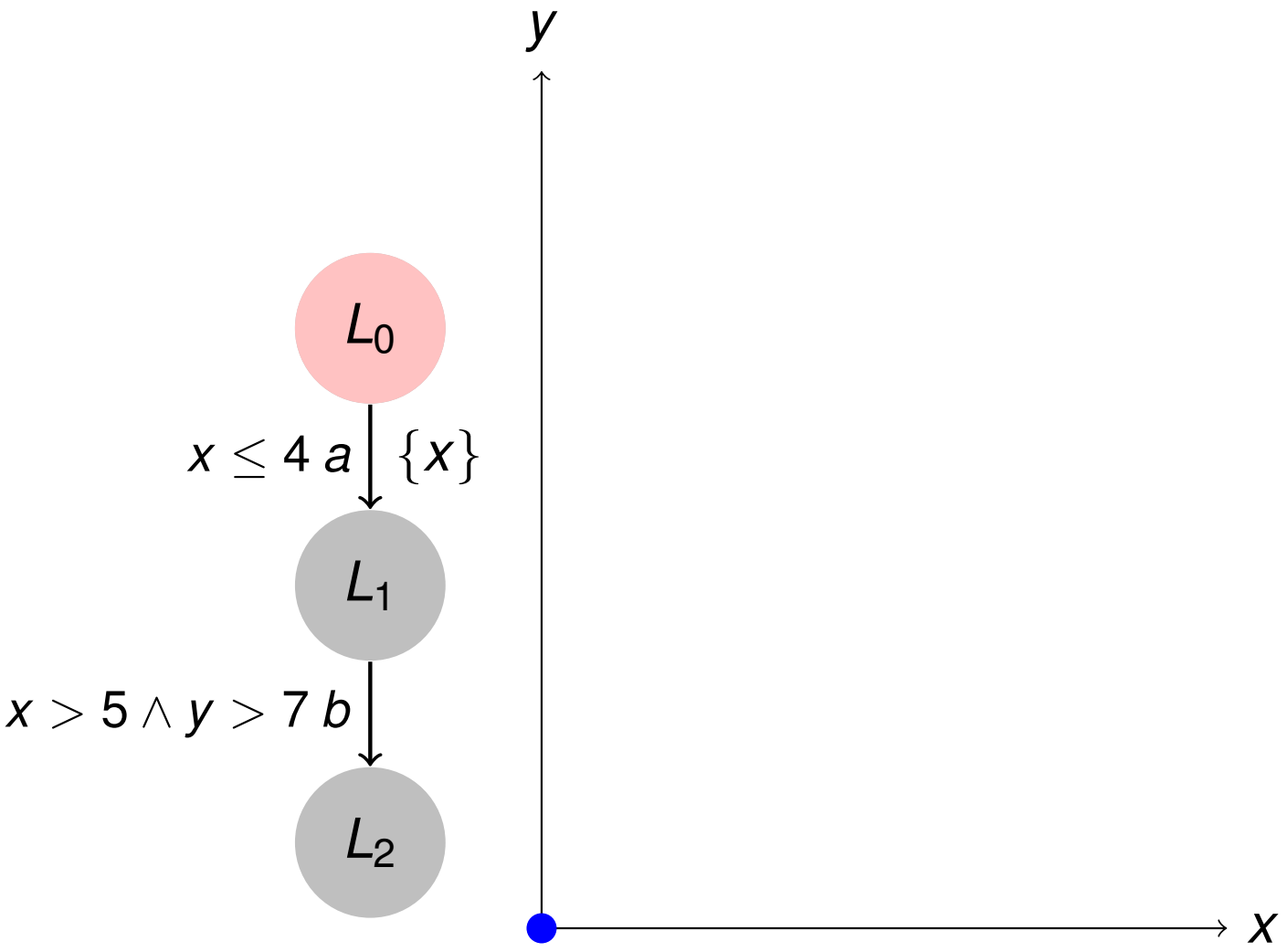
- Forward computation of the timed automaton
- $s_0 = (l_0, z_0 \uparrow)$ is the initial node.
- If $l_1 \xrightarrow{g,a,R} l_2$, then
  we have a transition $(l_1, z_1) \xrightarrow{a} (l_2, z_2)$ where
  $z_2 = (z_1 \uparrow \cap g)_{R \leftarrow \overline{0}}$

  or z_2 is a set of valuations we obtain by letting time elapse from z_1 : $(l_1, z_1) \xrightarrow{\varepsilon} (l_1, z_2)$.
- An ε transition appears due to canonical decomposition
- Abstract a new zone if necessary.

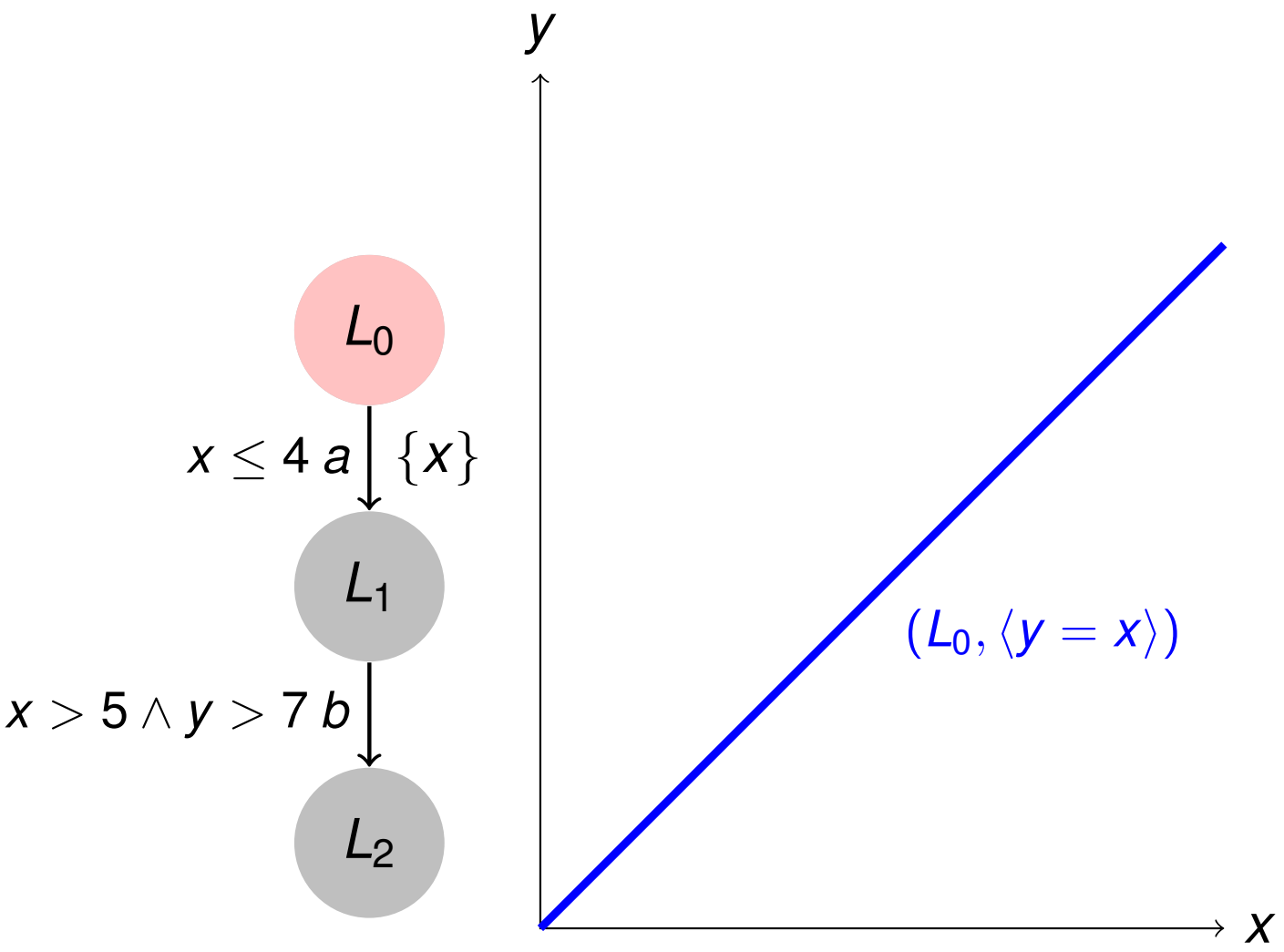
Zone Graph

- If the guards are zones (i.e. convex), then only zones are produced.
- Canonically decompose z based on guards on its outgoing transitions.
- pre-stabilize zones.

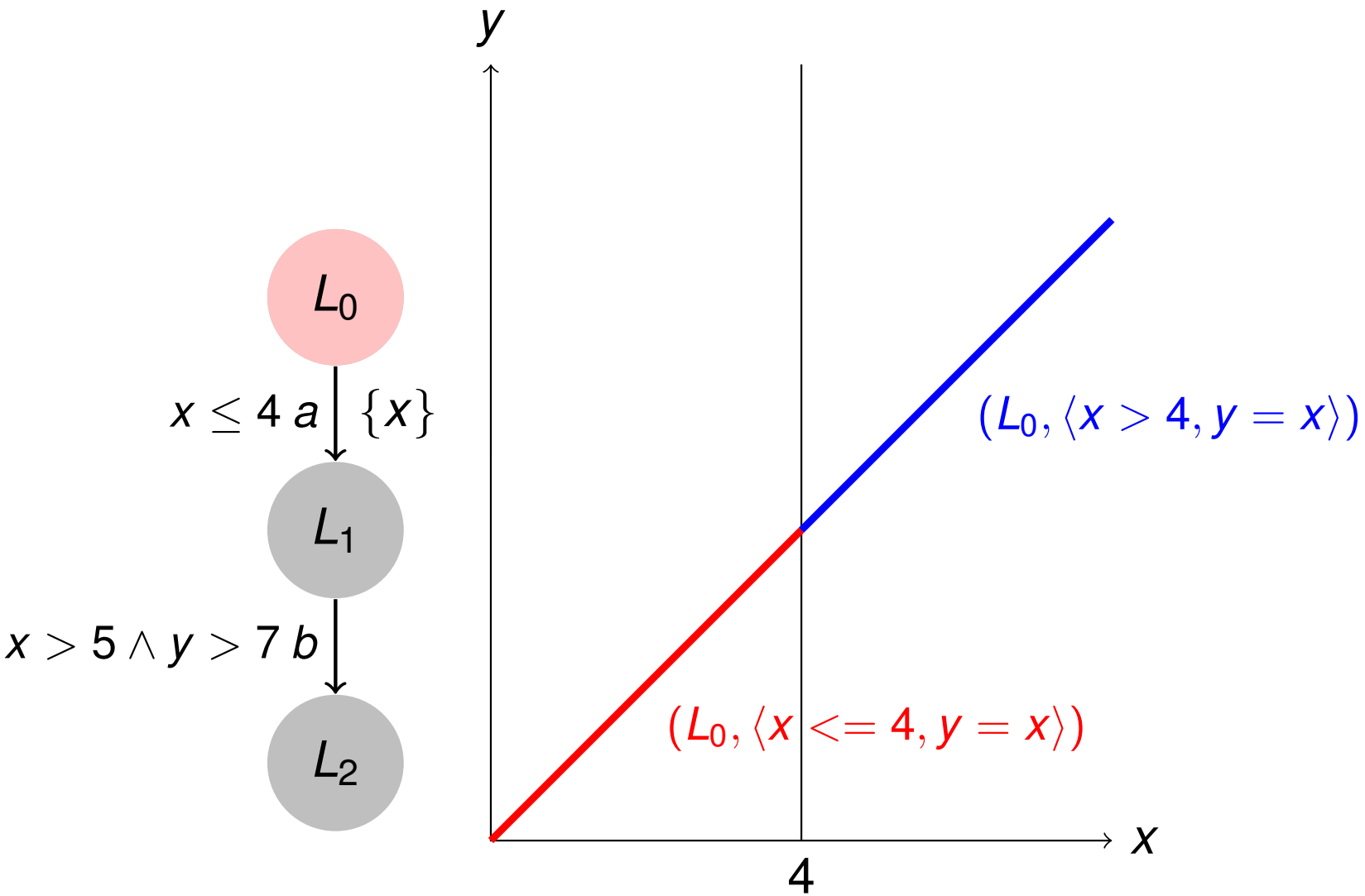
Zone Graph Construction



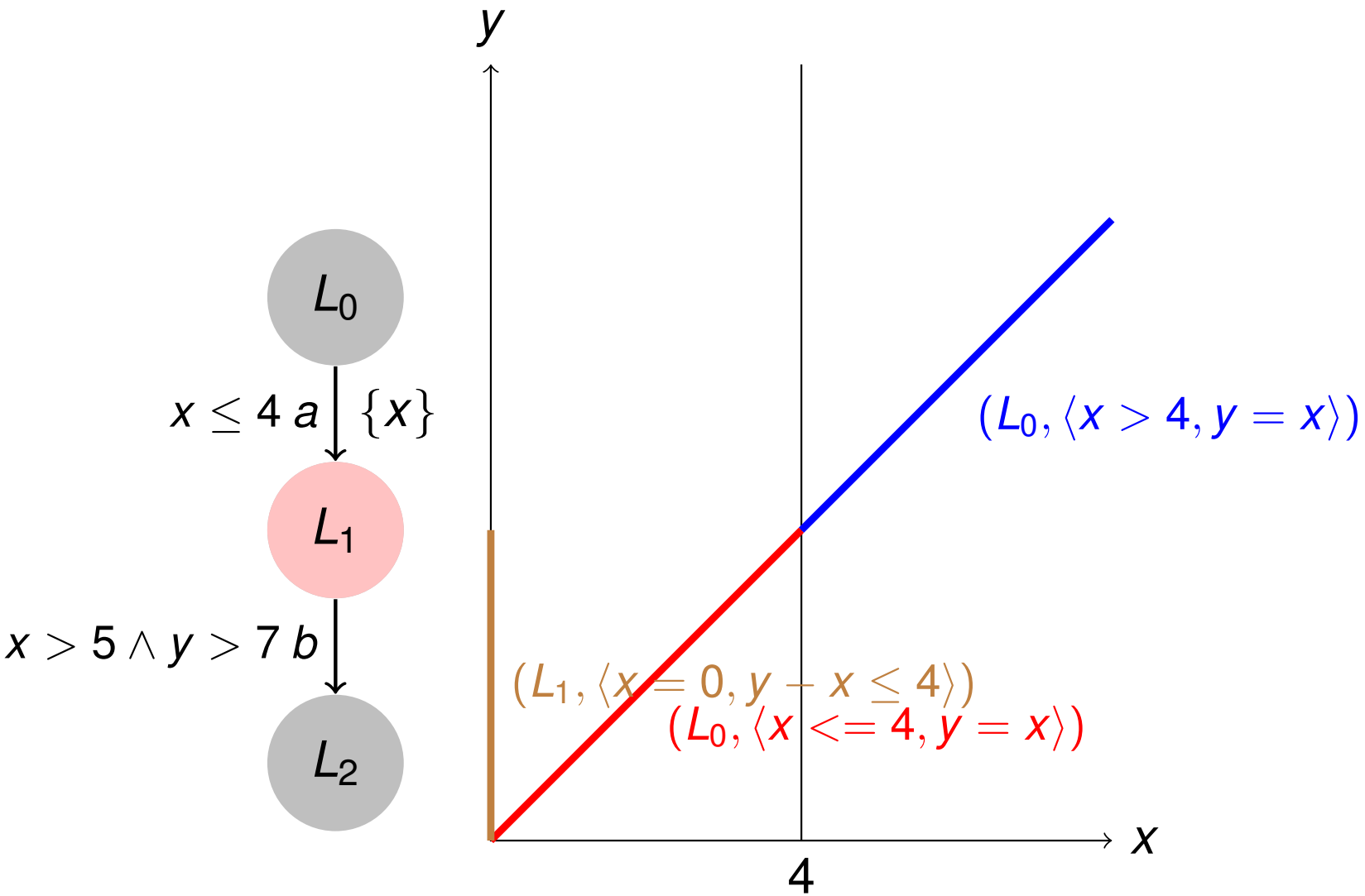
Zone Graph Construction



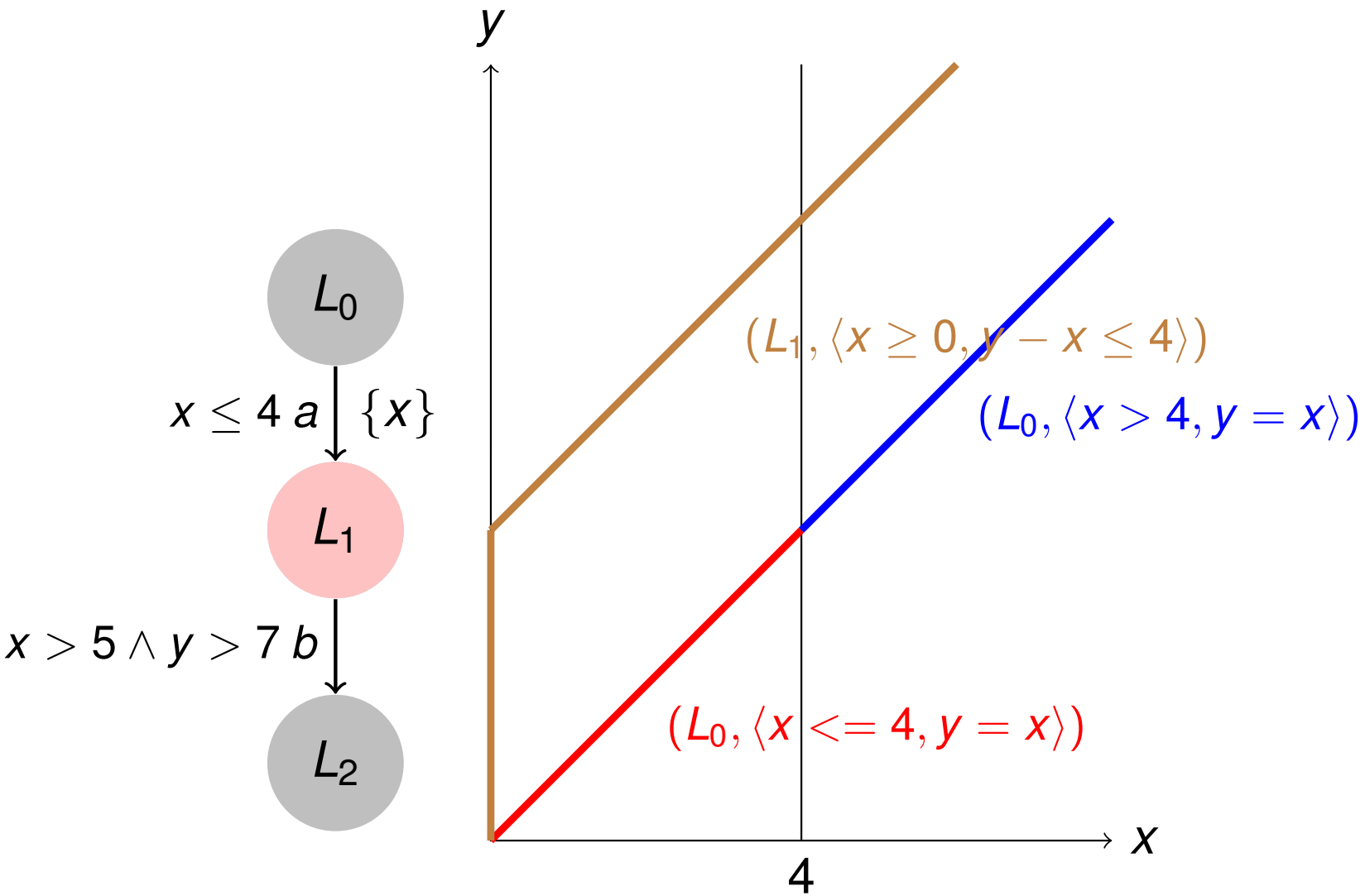
Zone Graph Construction



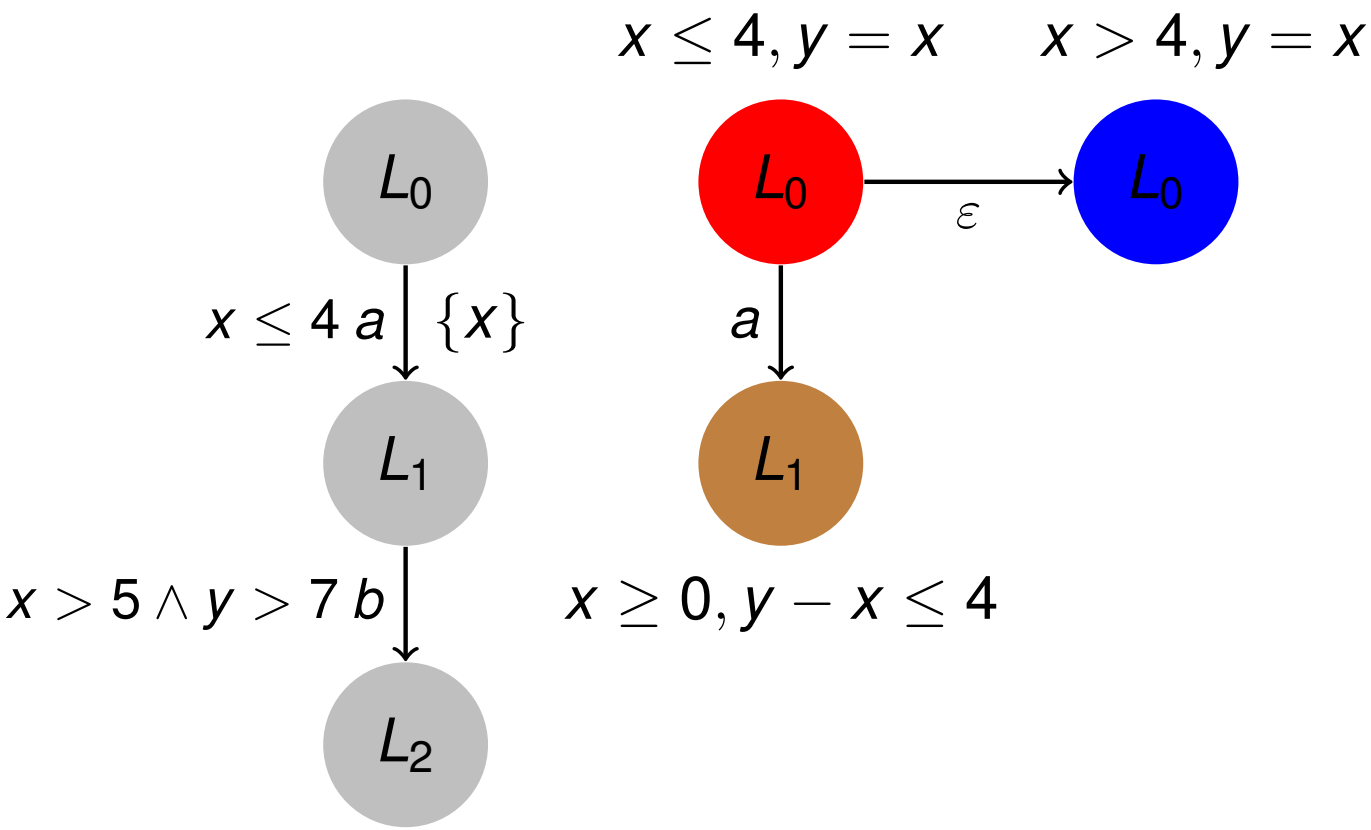
Zone Graph Construction



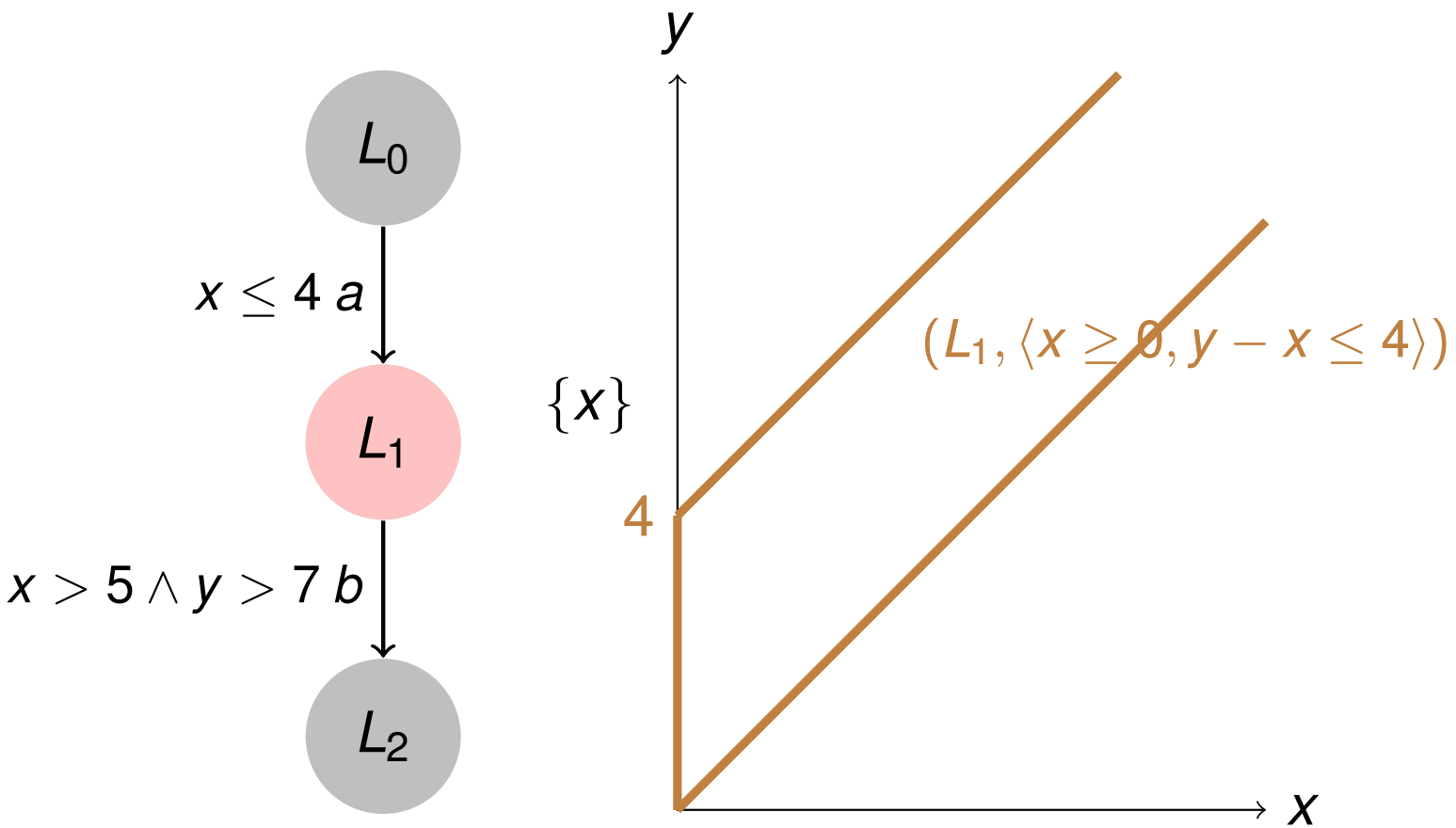
Zone Graph Construction



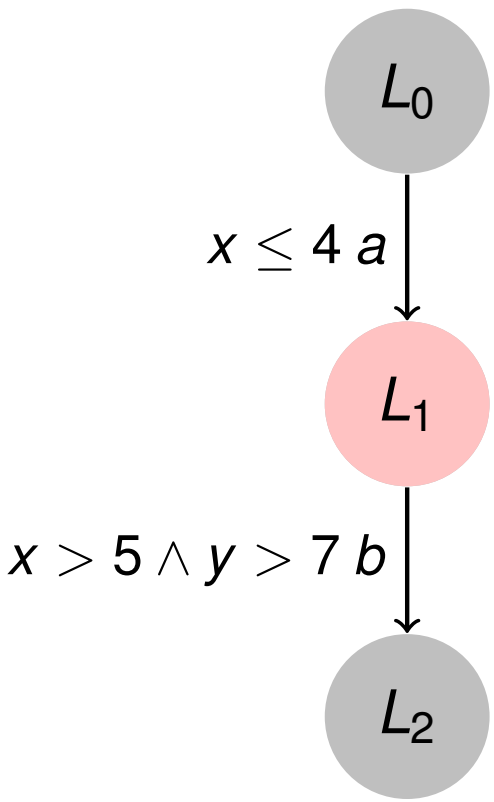
Zone Graph Construction

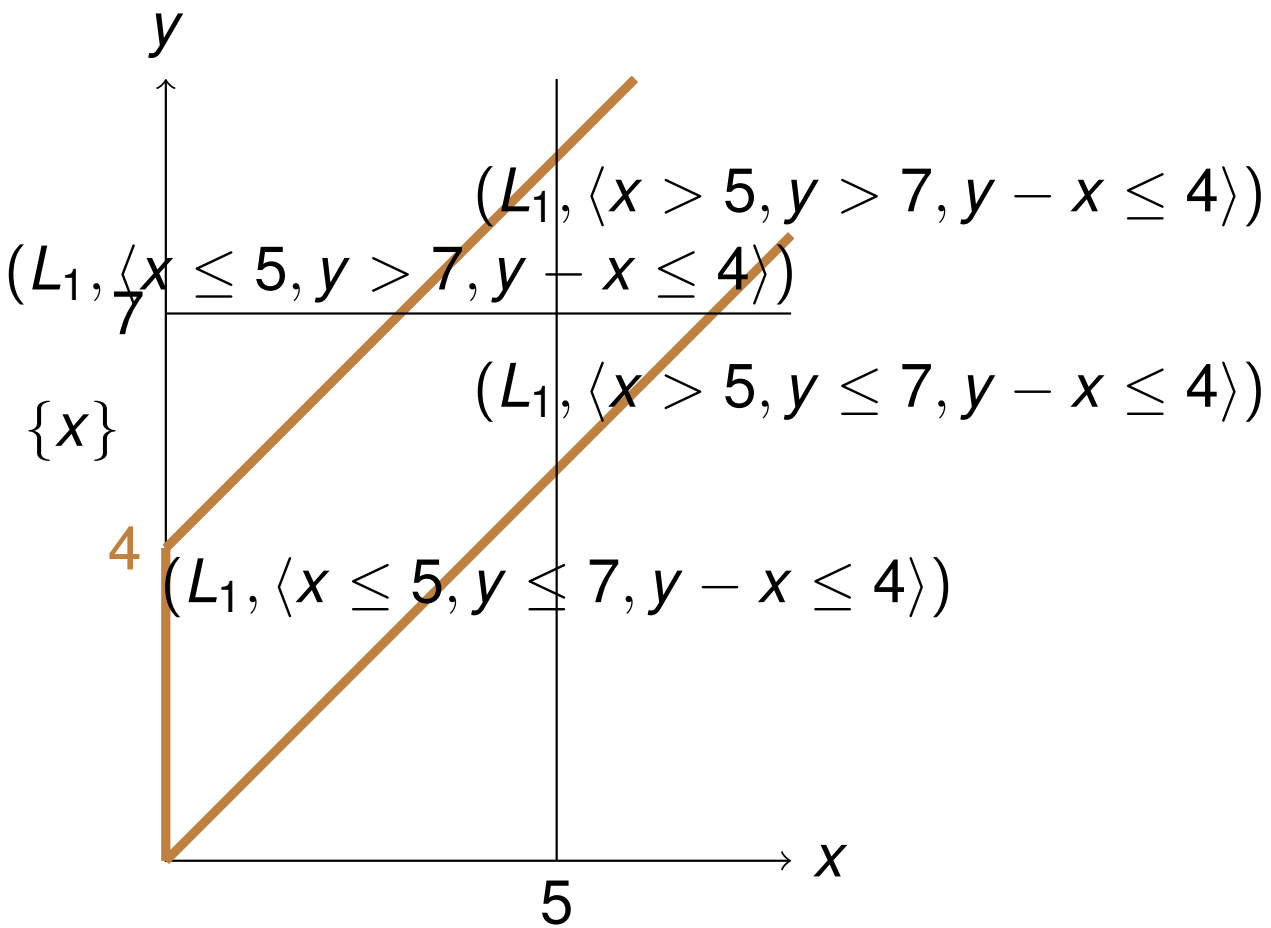


Zone Graph Construction

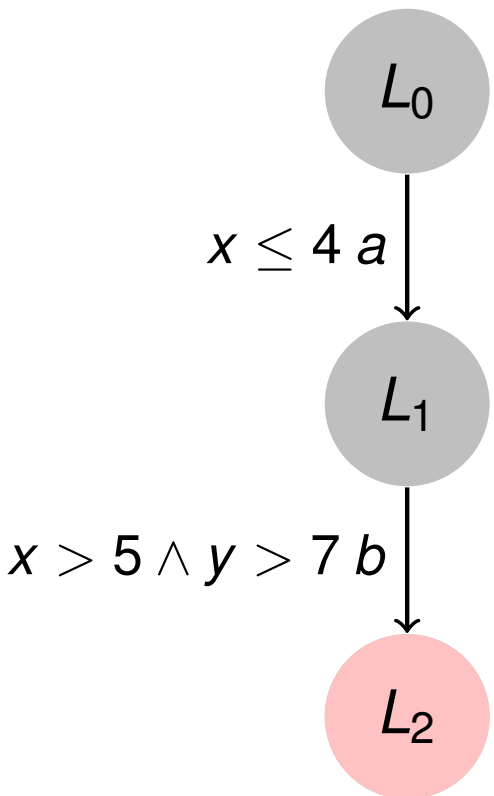


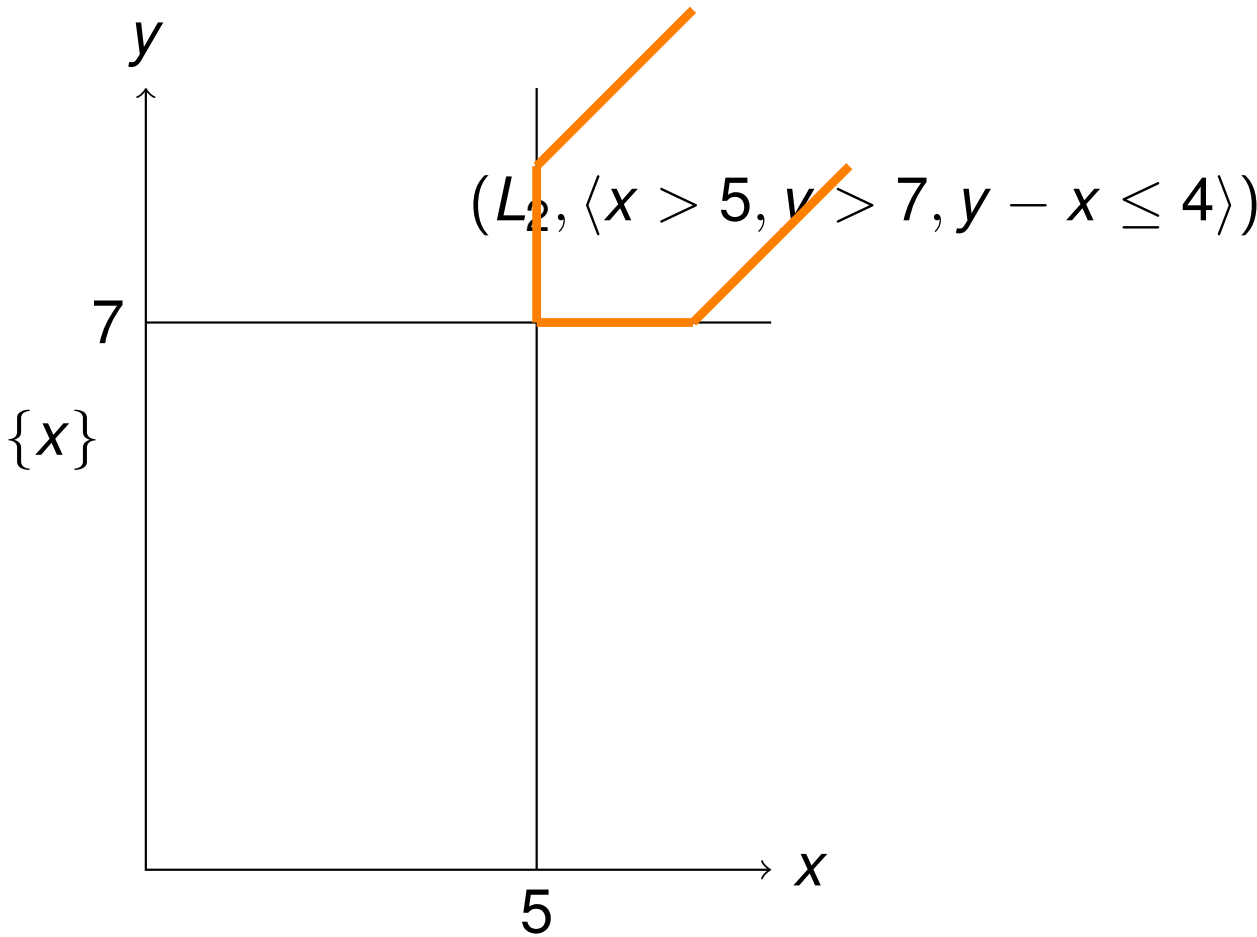
Zone Graph Construction



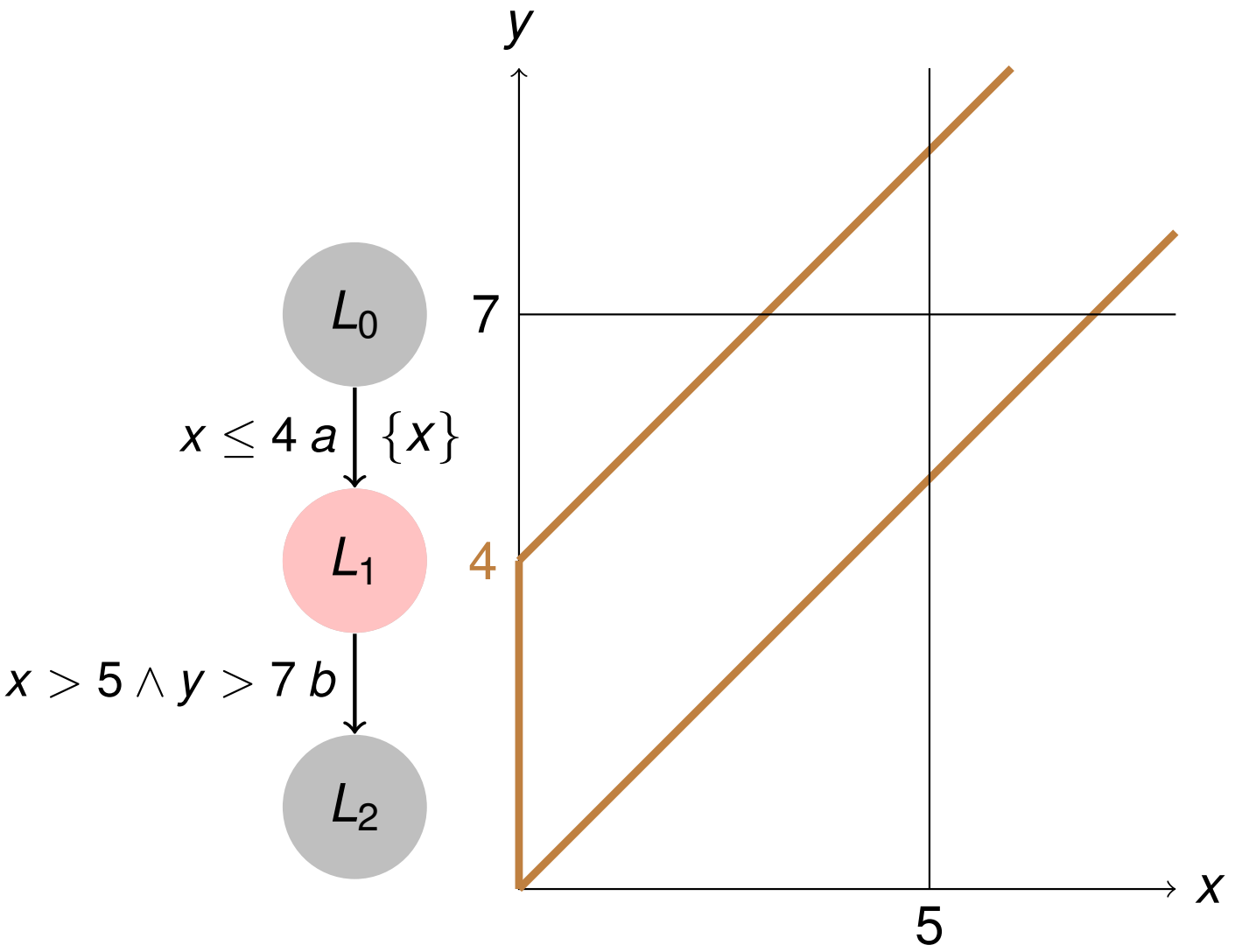


Zone Graph Construction

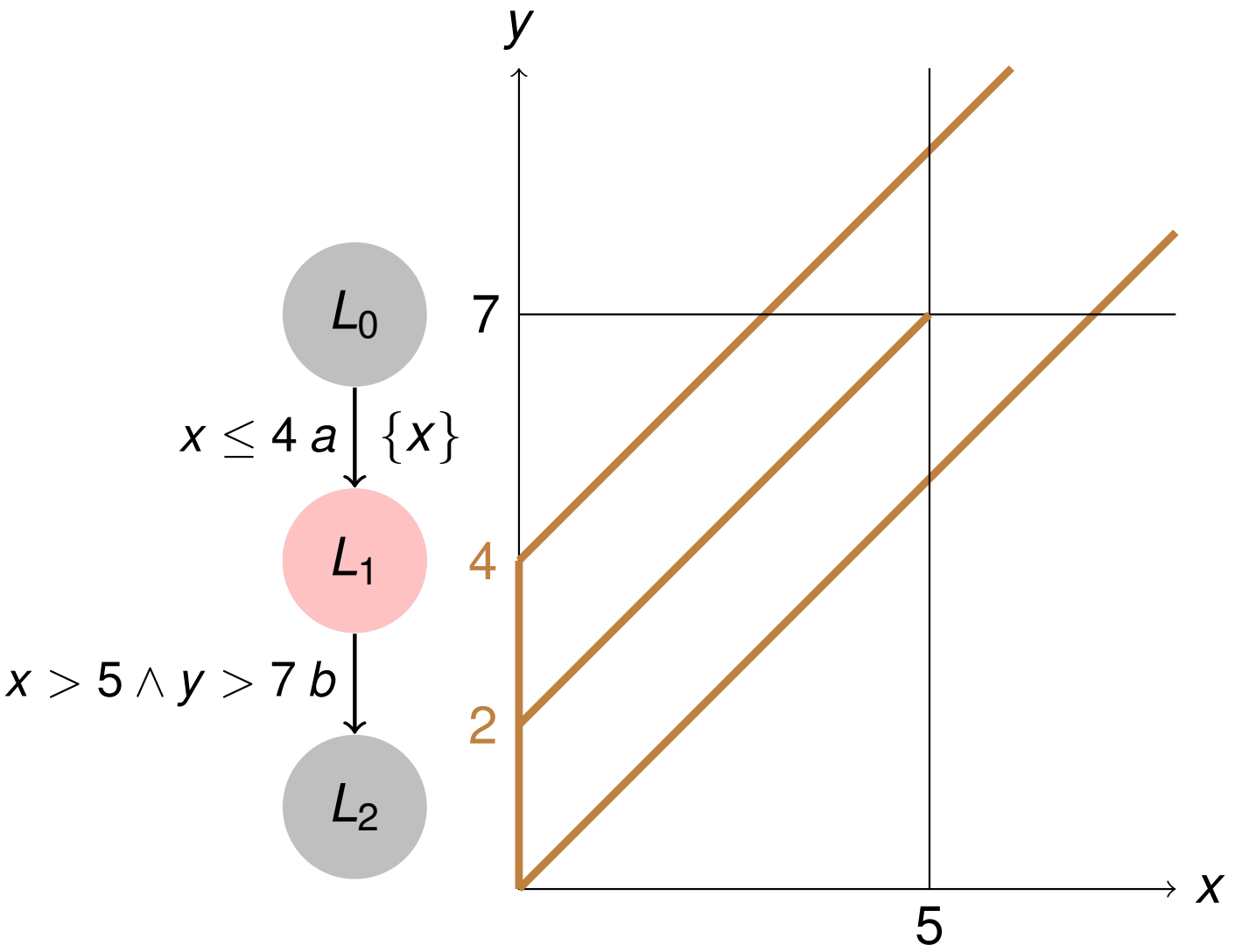




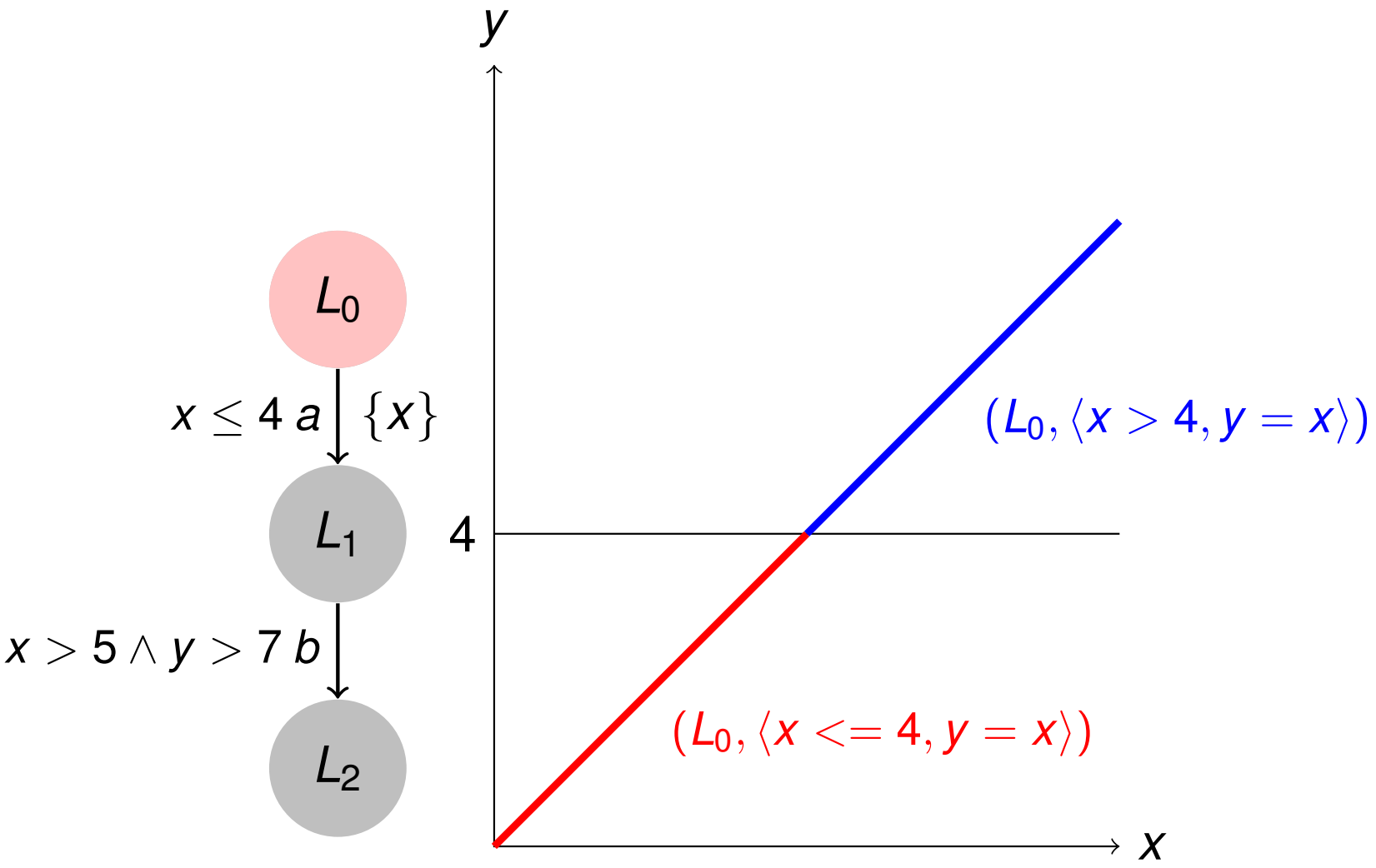
Prestabilizing Zones



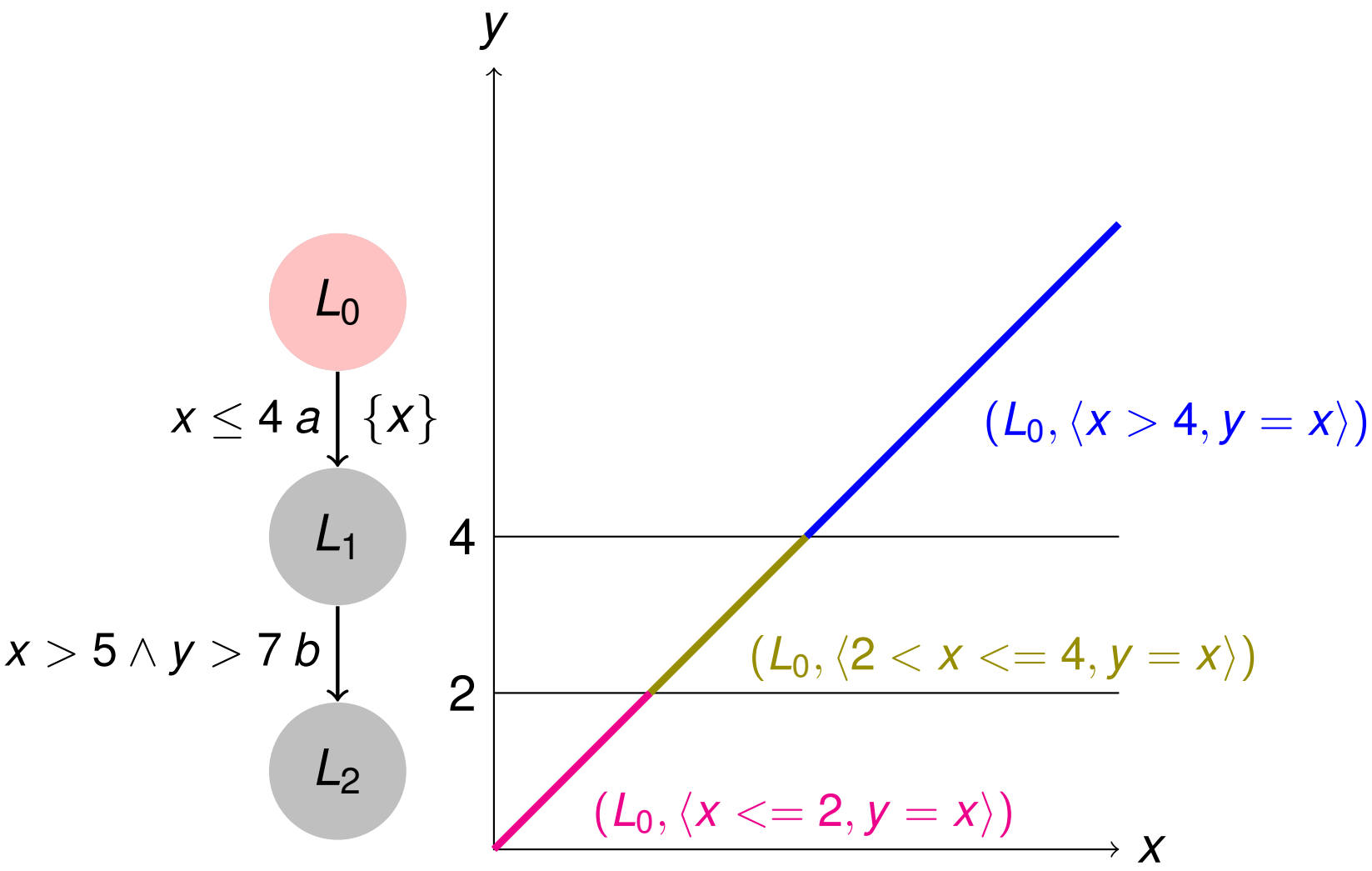
Prestabilizing Zones



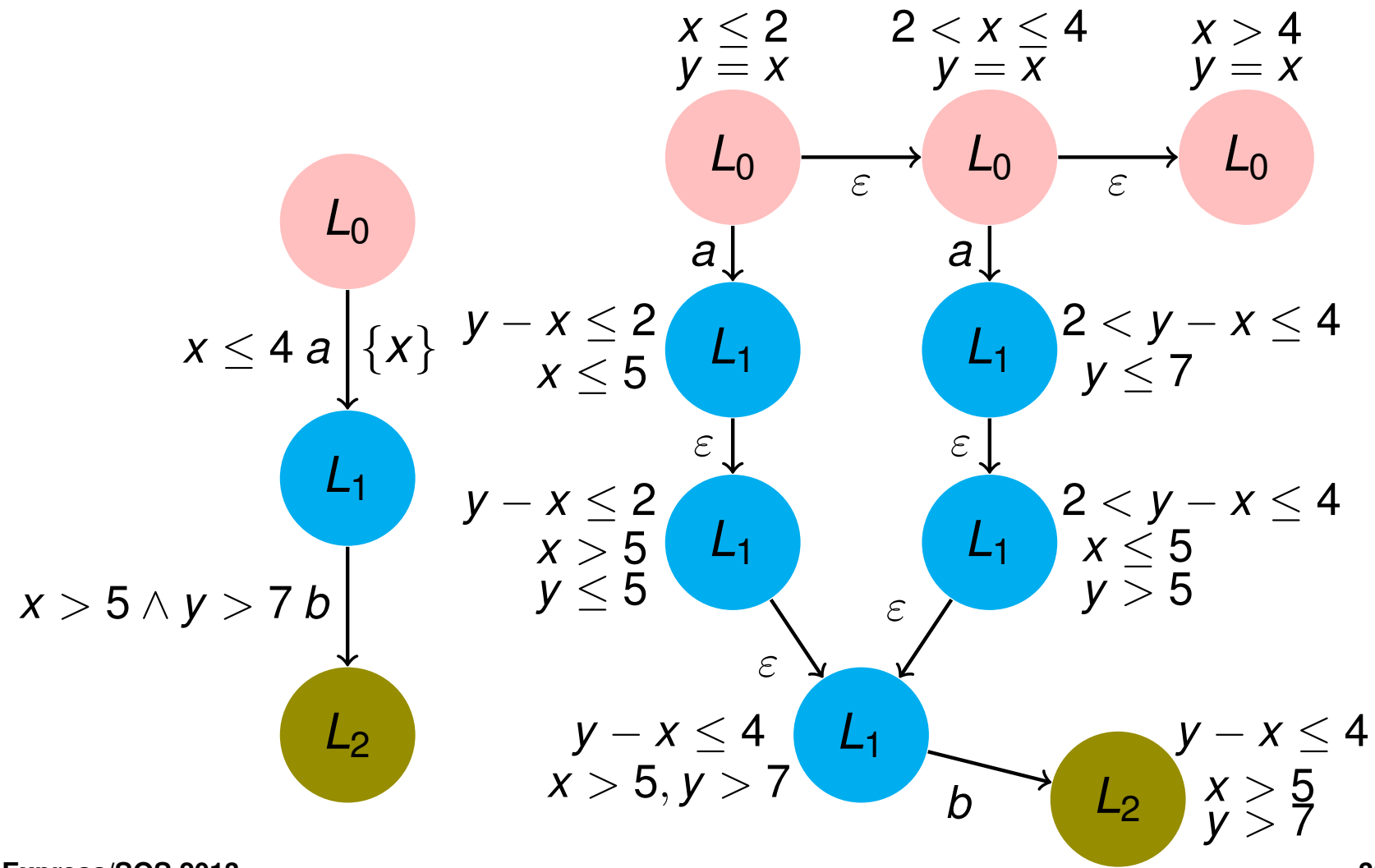
Prestabilizing Zones



Prestabilizing Zones



Zone Construction



Abstraction: Location Dependent Maximal Constants

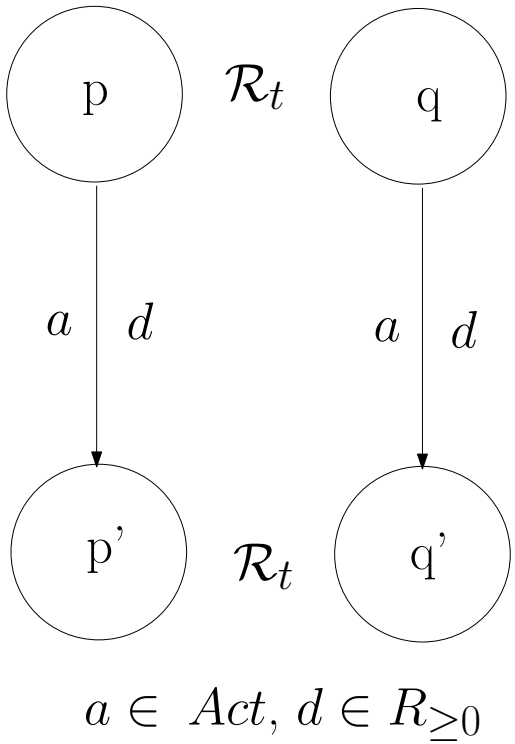
- *Static Guard Analysis in Timed Automata Verification* Behrmann et. al. 03
- For each clock $x \in C$ and each locaion $l \in L$, determine max_x^l
- $max_x^l \leq c_x$
- Thus the number of nodes reduced compared to region graph abstraction.

Relations over Timed Automata

Timed Equivalences

Timed Bisimulation

p and q are two timed states.

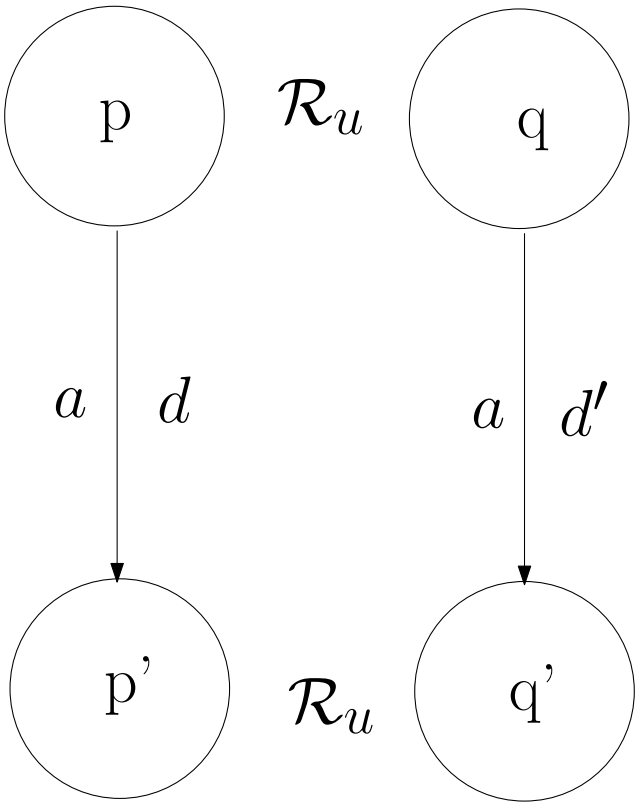


$a \in Act,\ d \in R_{\geq 0}$

Infinitely many equivalence classes

Timed Equivalences

Time Abstracted Bisimulation

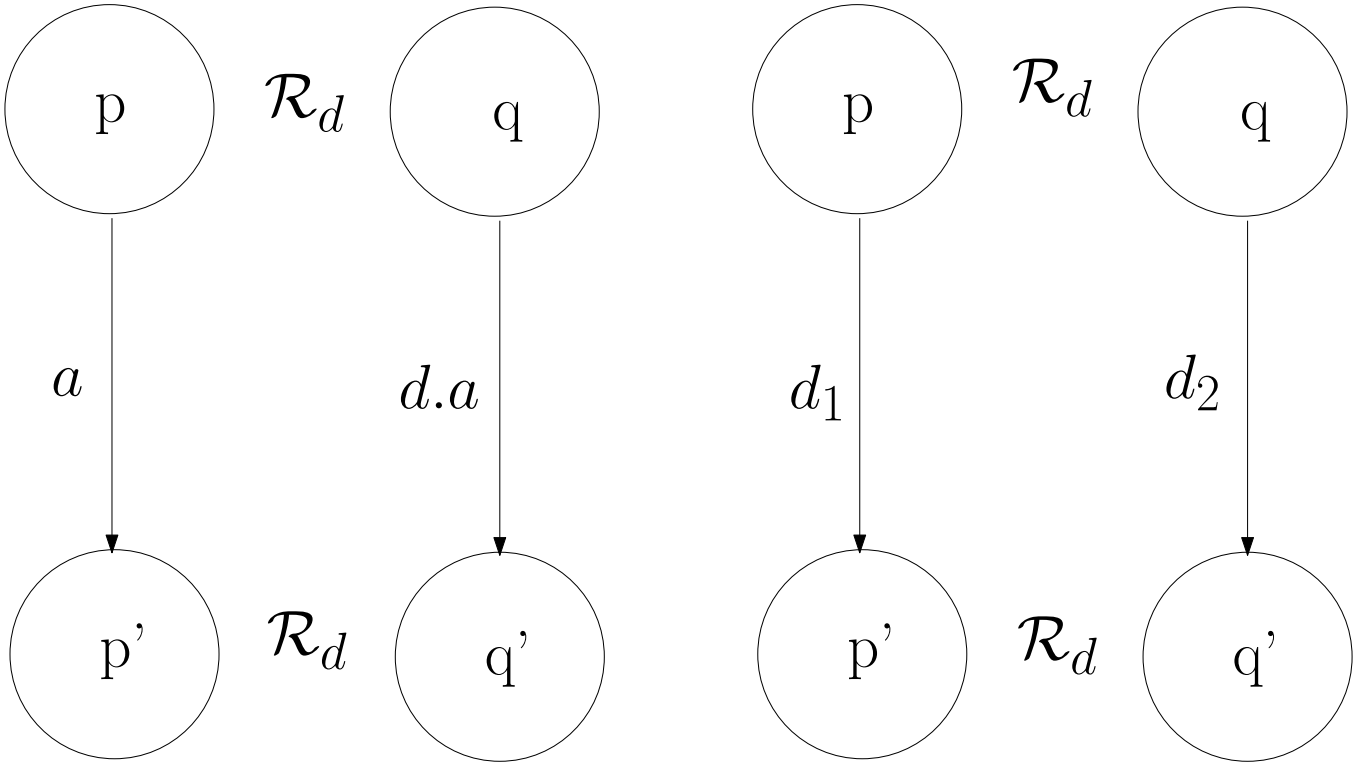


$$a \in Act,\ d, d' \in R_{\geq 0}$$

Timed Equivalences

Time Abstracted Delay Bisimulation

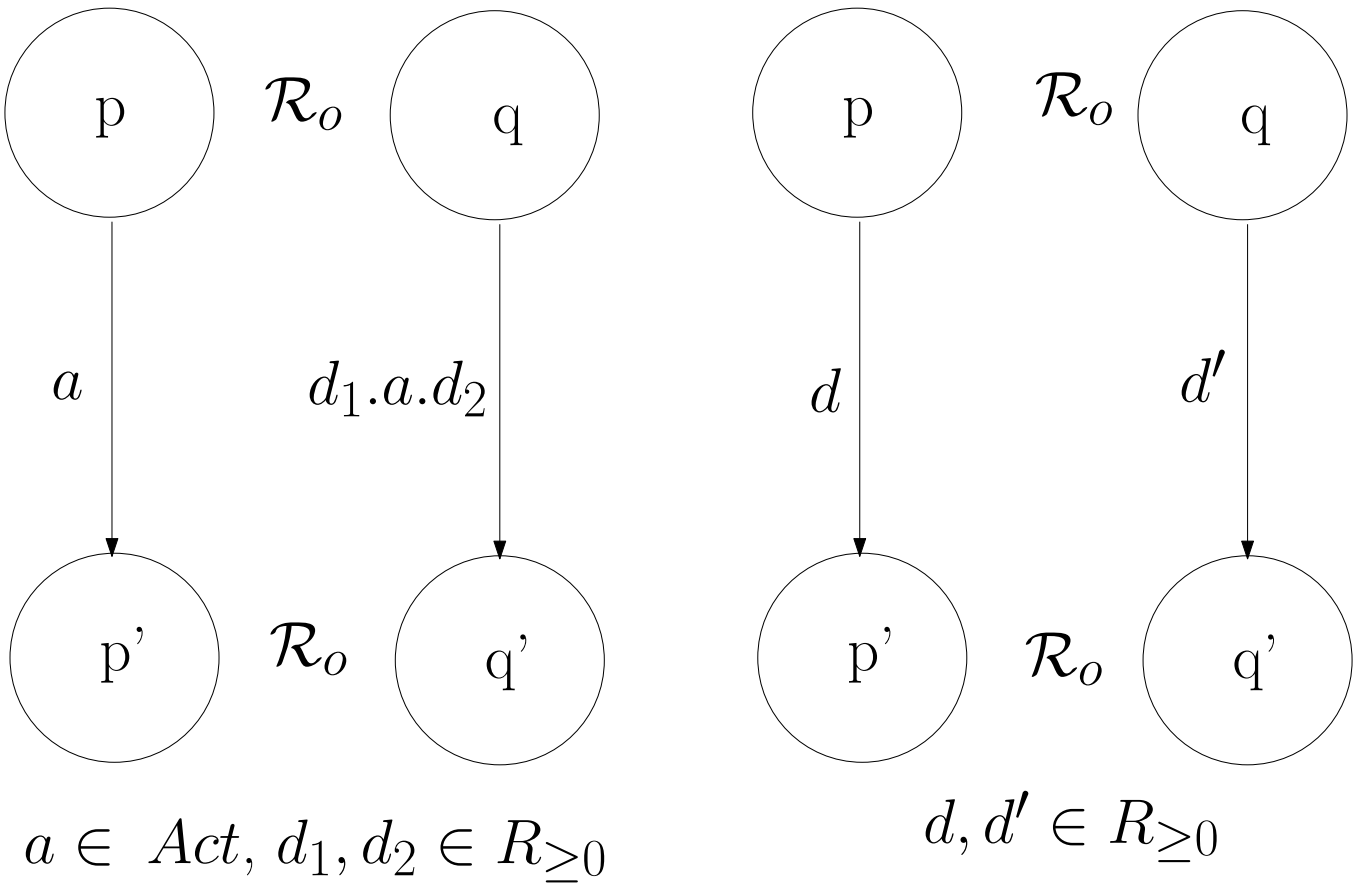
p and q are two timed valuations.



$a \in Act, d \in R_{\geq 0}$ $\qquad$ $d_1, d_2 \in R_{\geq 0}$

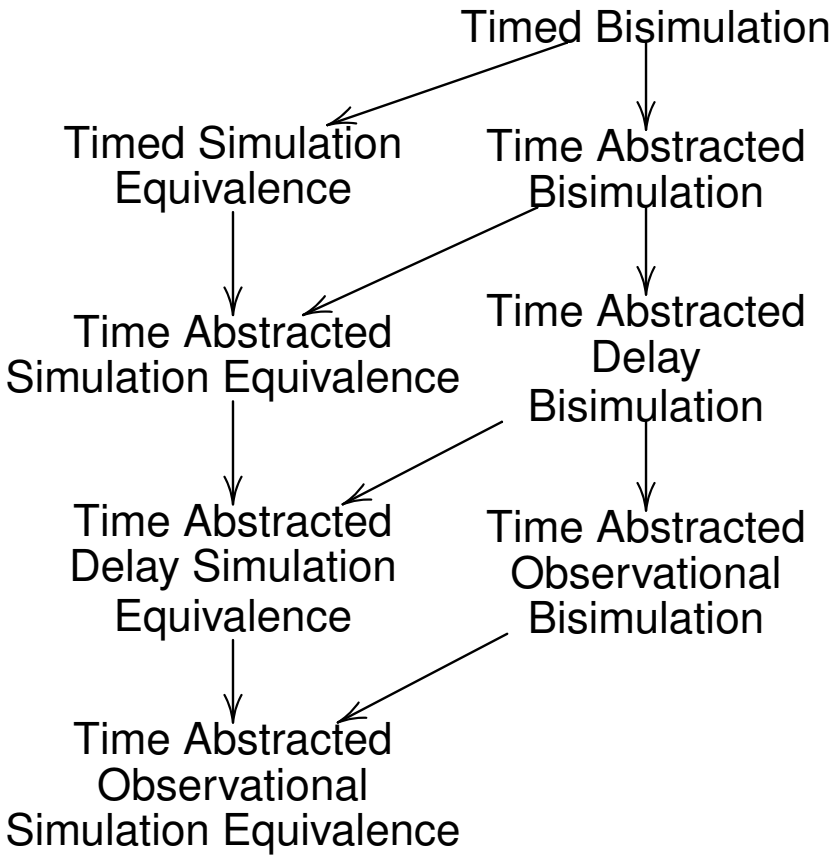
Timed Equivalences

Time Abstracted Observational Bisimulation



Simulation Preorder and Equivalences

For every bisimulation relation defined above, we can also have a simulation equivalence and a simulation preorder



Deciding the Relation using *common* Zone Graph

Deciding Timed Bisimulation

- Infinitely many equivalence classes.

  (i) Not all delays to be checked

  (ii) if p and q are initial states, or states with integer clock valuations, then consider delays of the form n, $n + \delta$ or $n - \delta$, where $n \in \{0, 1, \ldots, C\}$, $C = max(C_A, C_B)$.

  (iii) for fractional clock valuations with small modification.

- Delays mentioned above enabling movement from one corner point to another of the same zone or a delay successor zone are called *corner point delays* or *cp-delays*.

Corner Point Bisimulation (CP-bisimulation)

- Similar to timed bisimulation
- Consider only corner point delays instead of all possible delays
- Timed bisimulation implies corner point bisimulation

CP-bisimulation

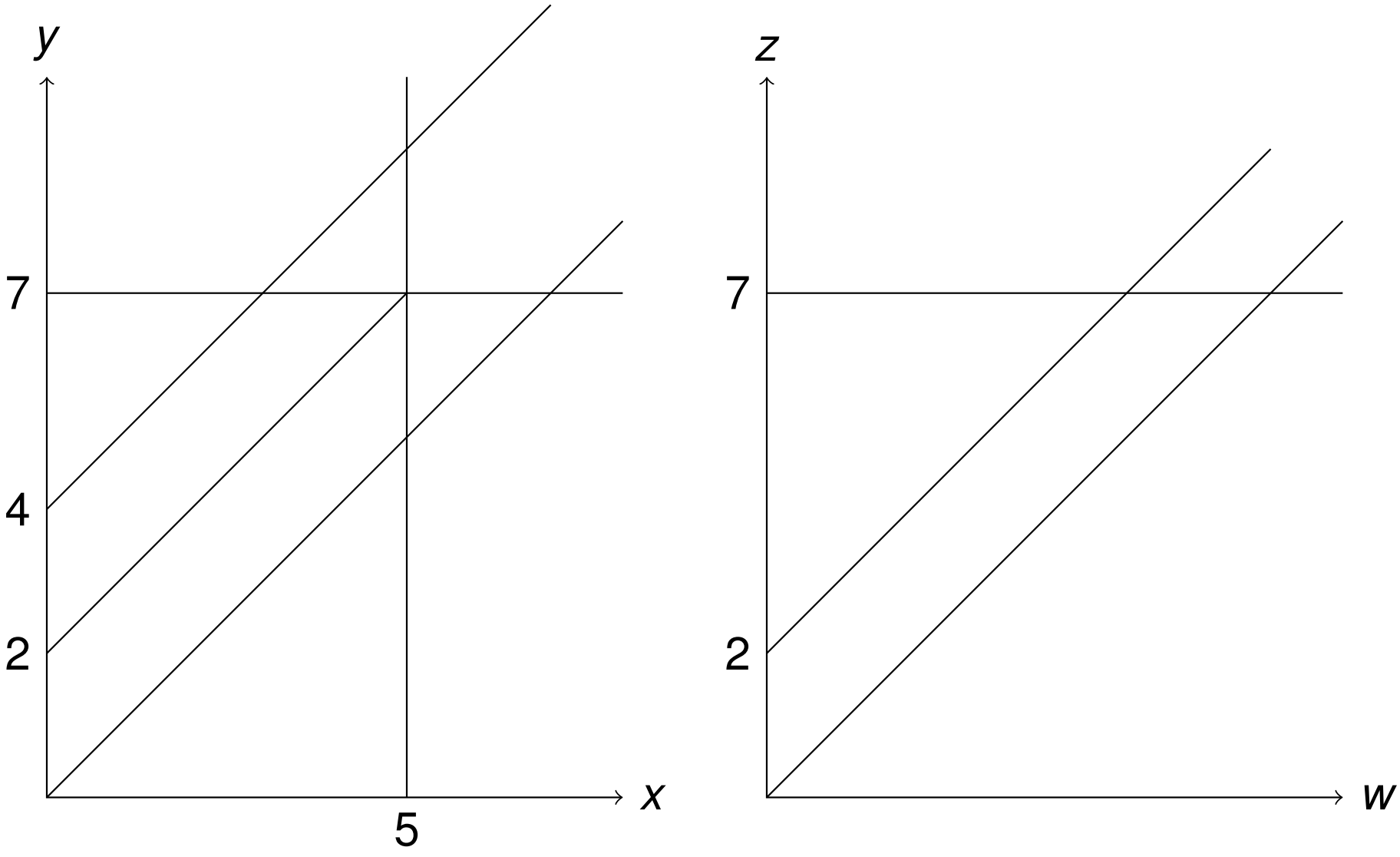
Lemma

For checking whether the timed automata states p and q are related through corner point simulation or corner point bisimulation relation, there are only finitely many pairs of states that need to be considered.

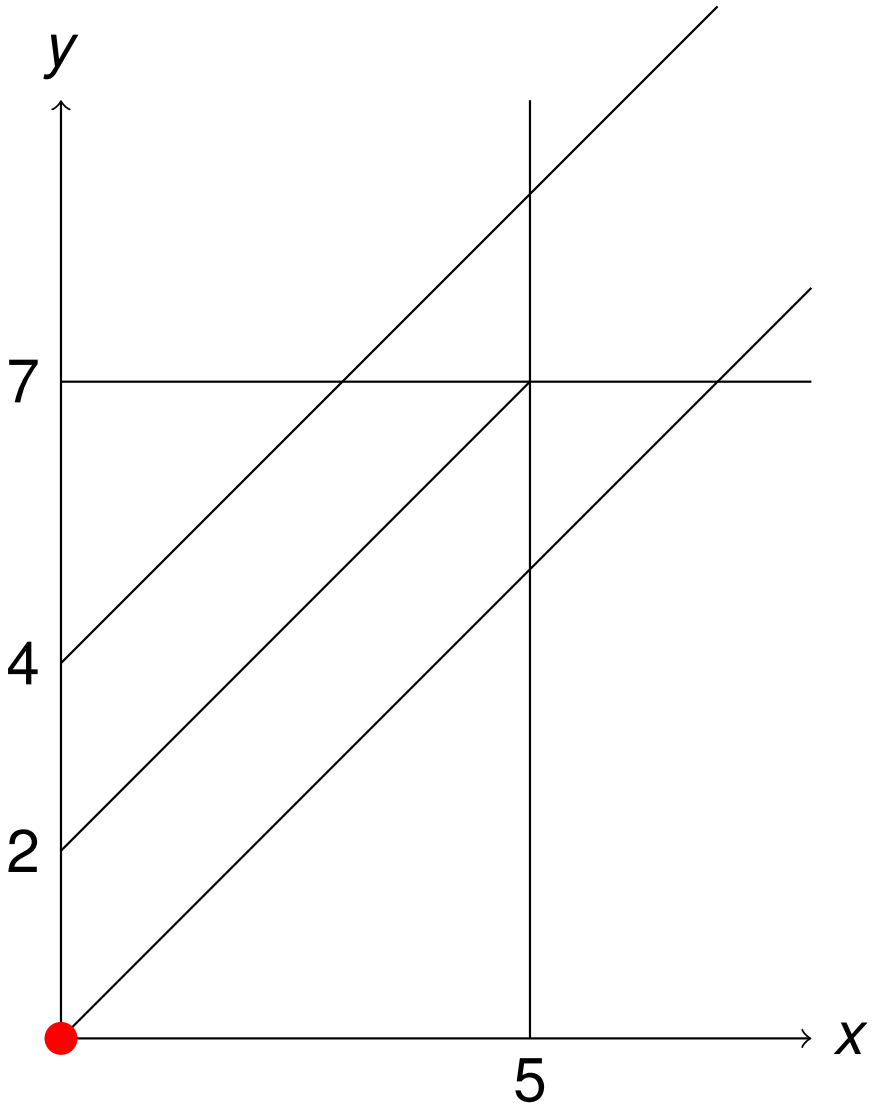
Theorem

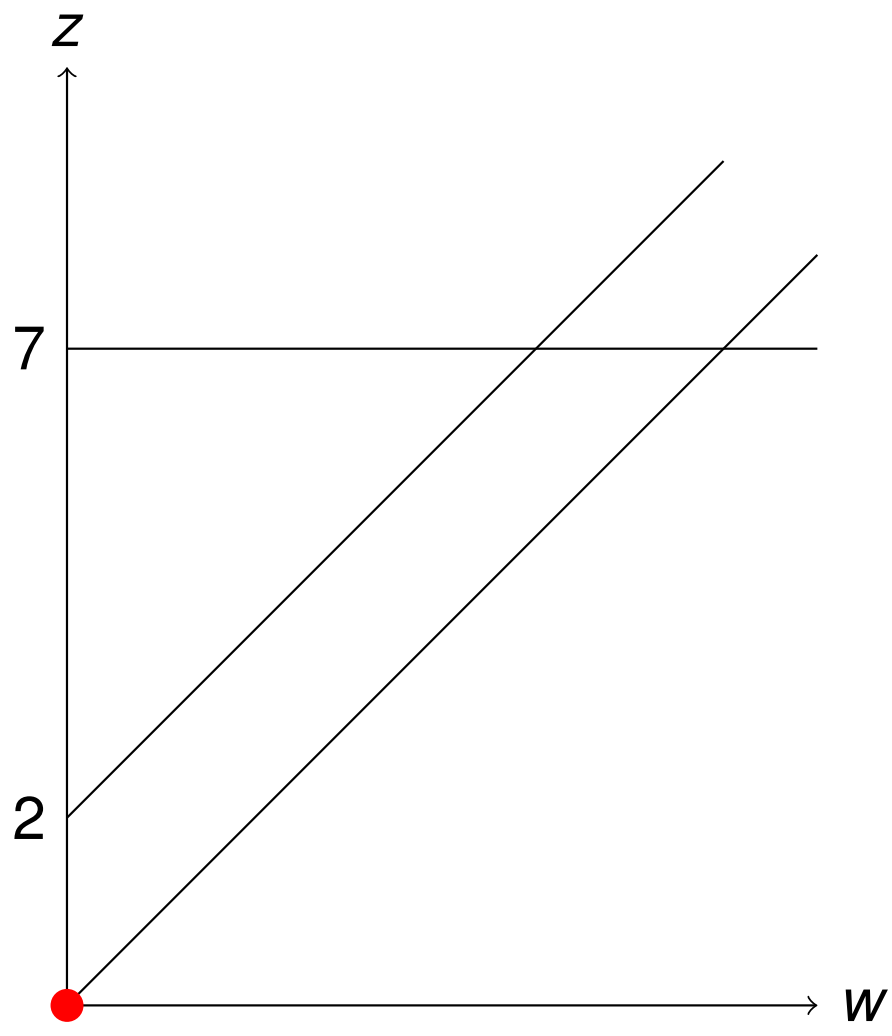
Corner point simulation and corner point bisimulation relations are decidable.

CP bisimulation (Can be a game)

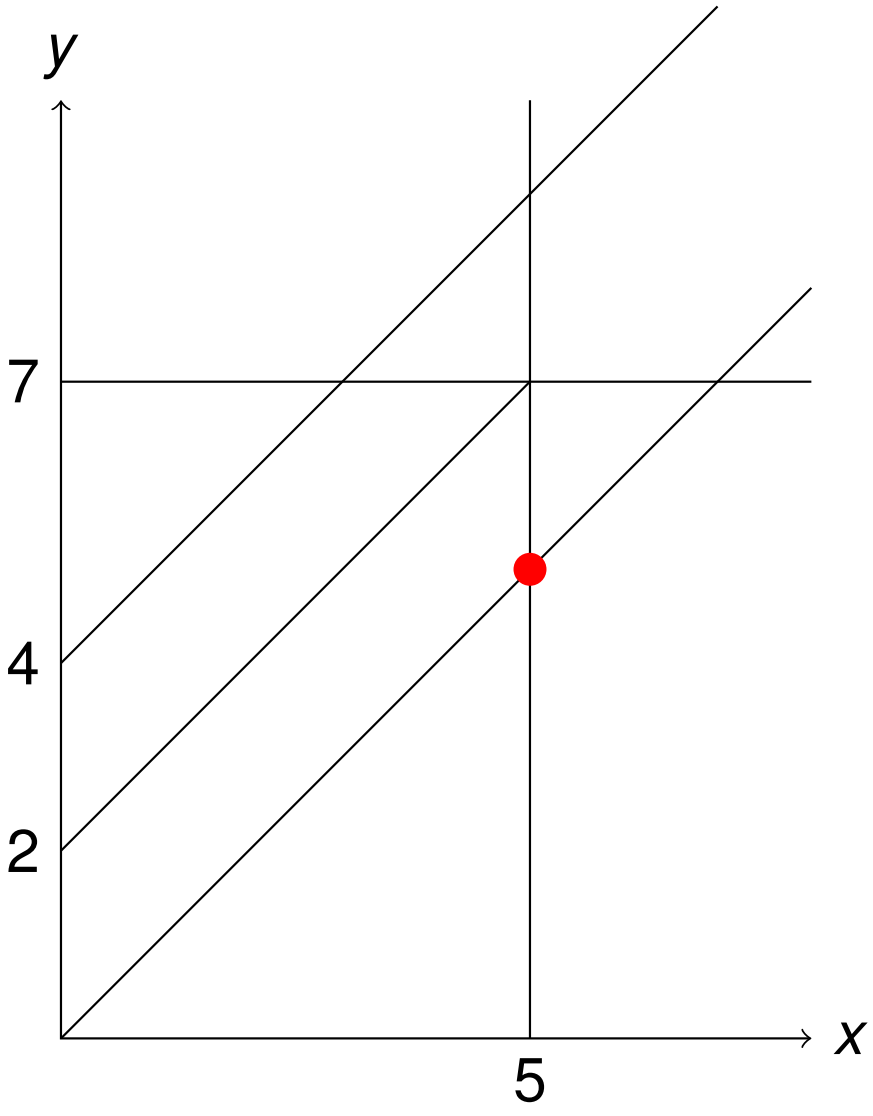


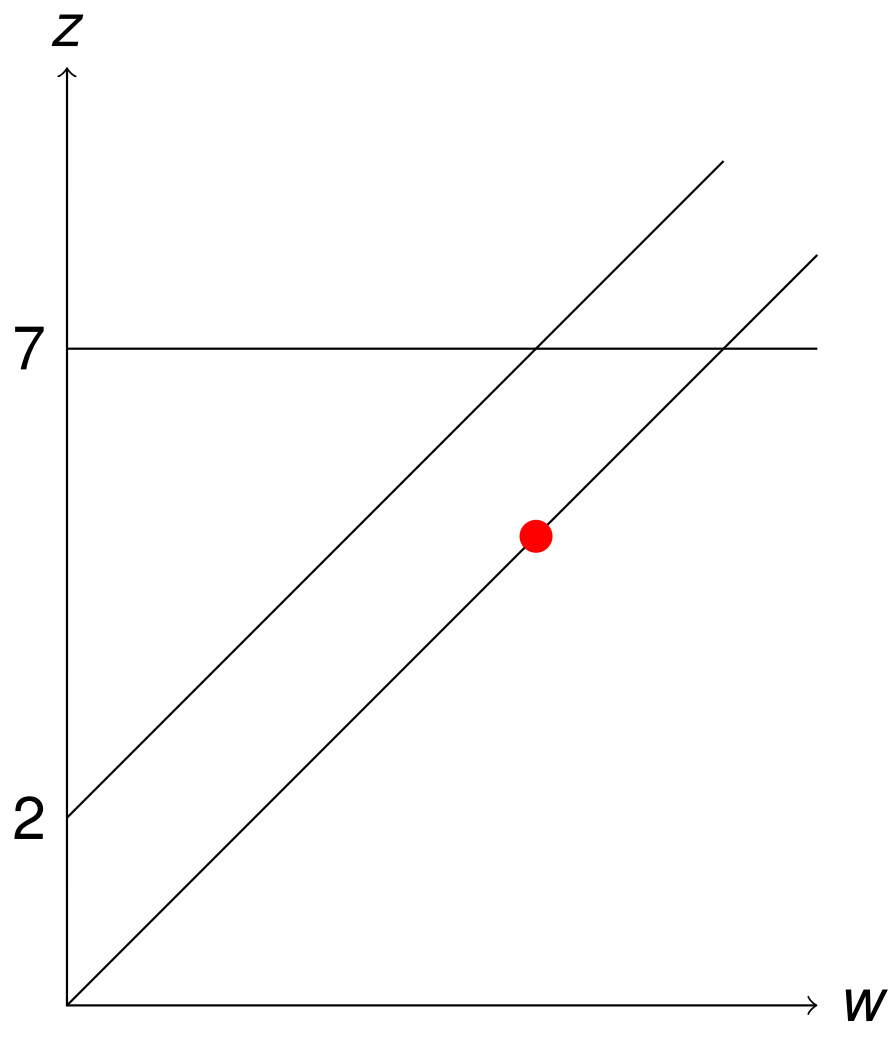
CP bisimulation (Can be a game)



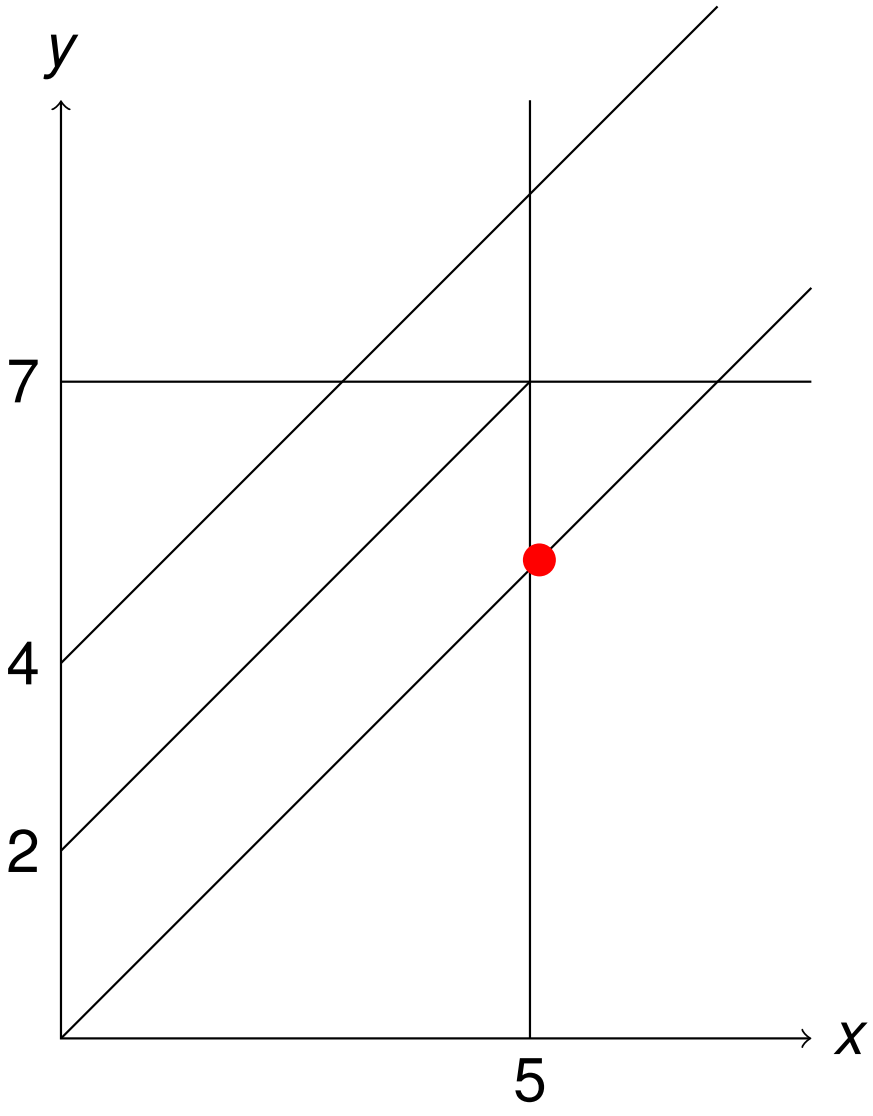


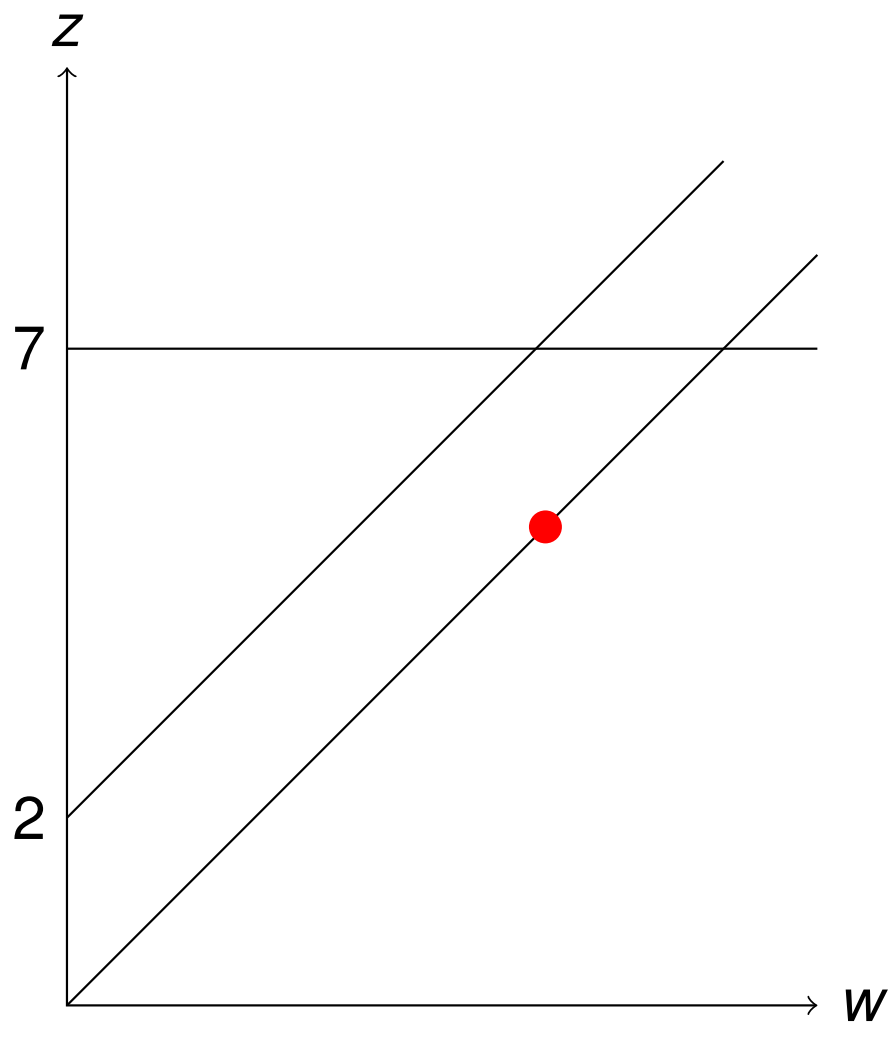
CP bisimulation (Can be a game)



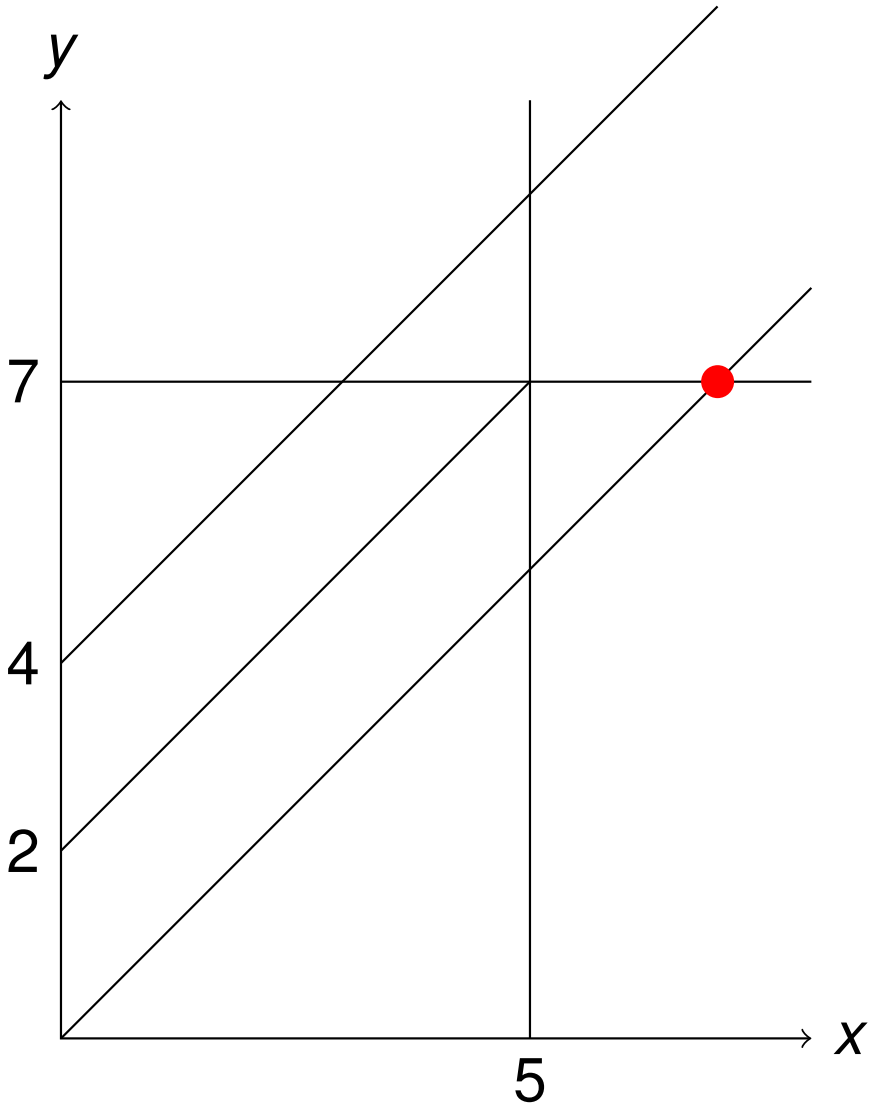


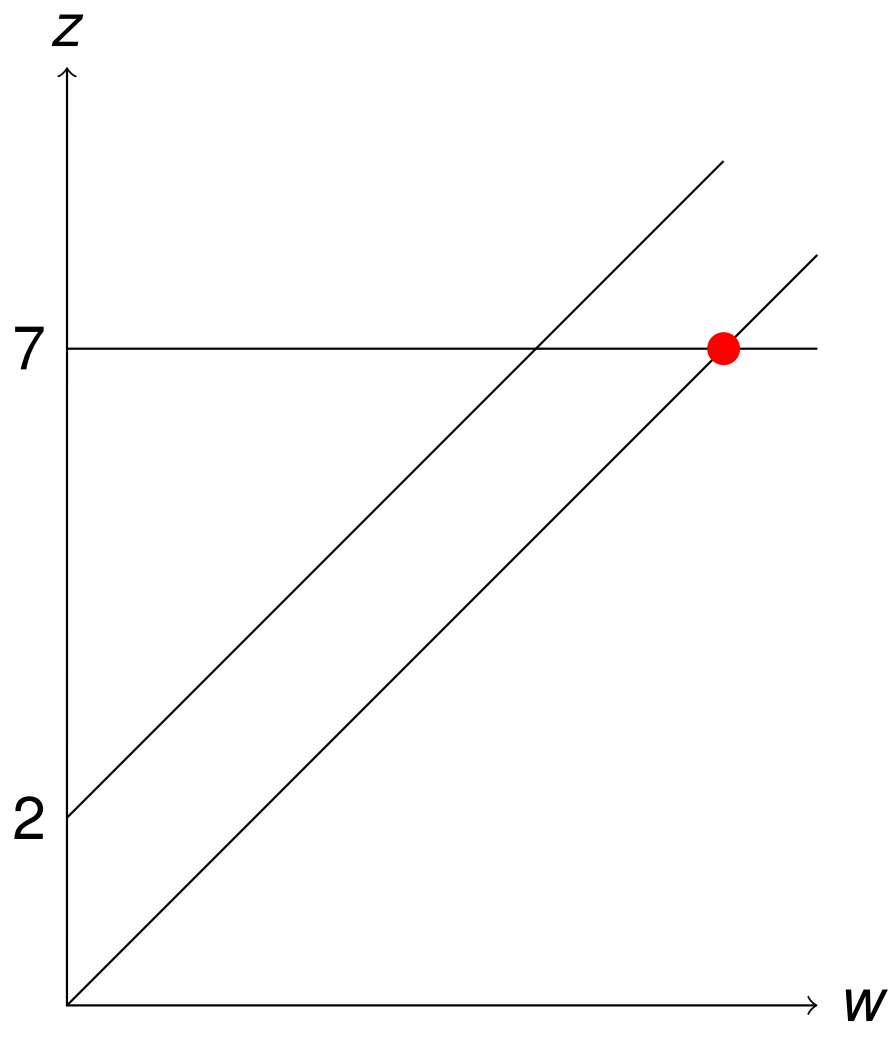
CP bisimulation (Can be a game)



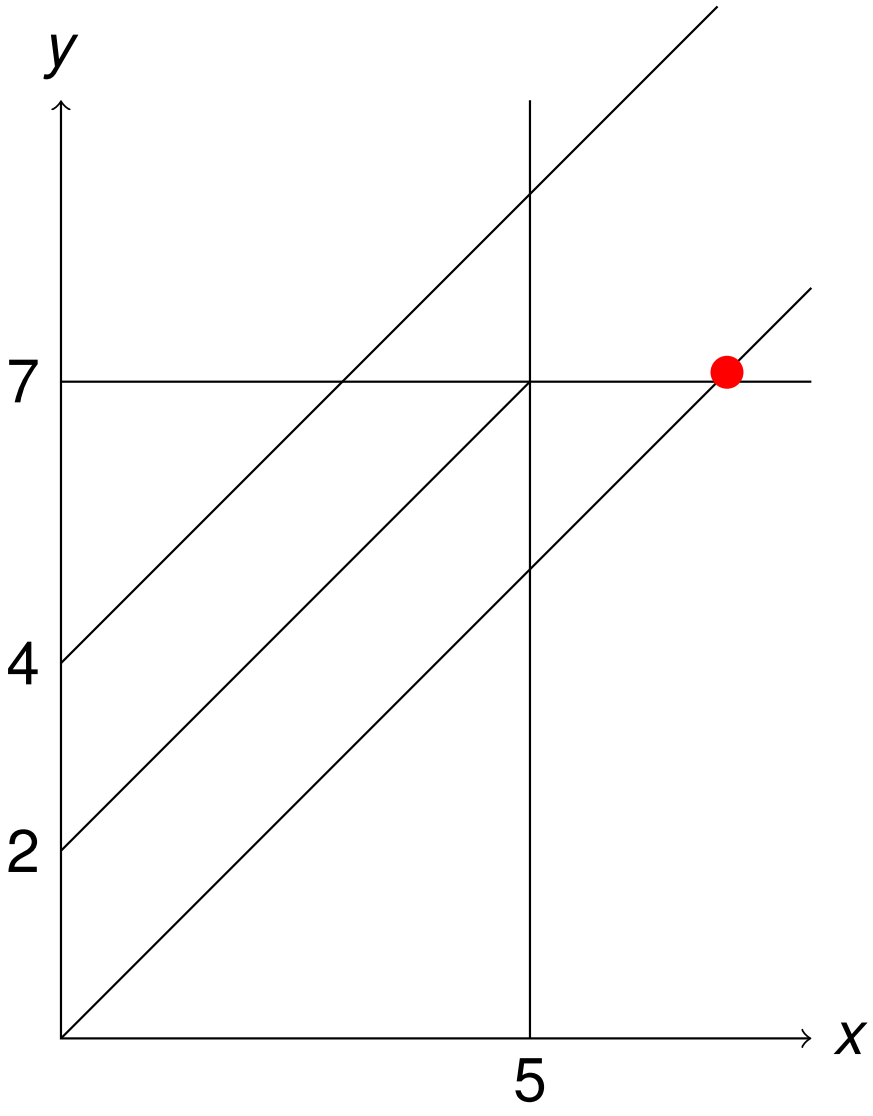


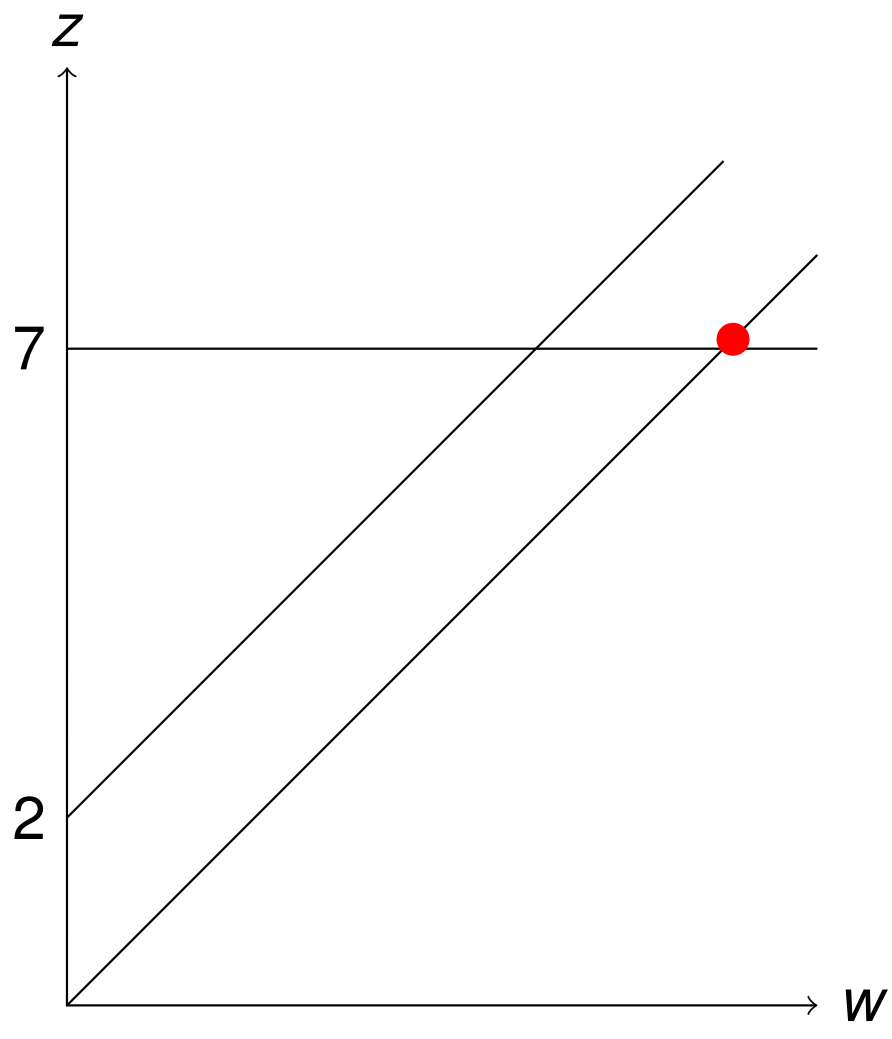
CP bisimulation (Can be a game)



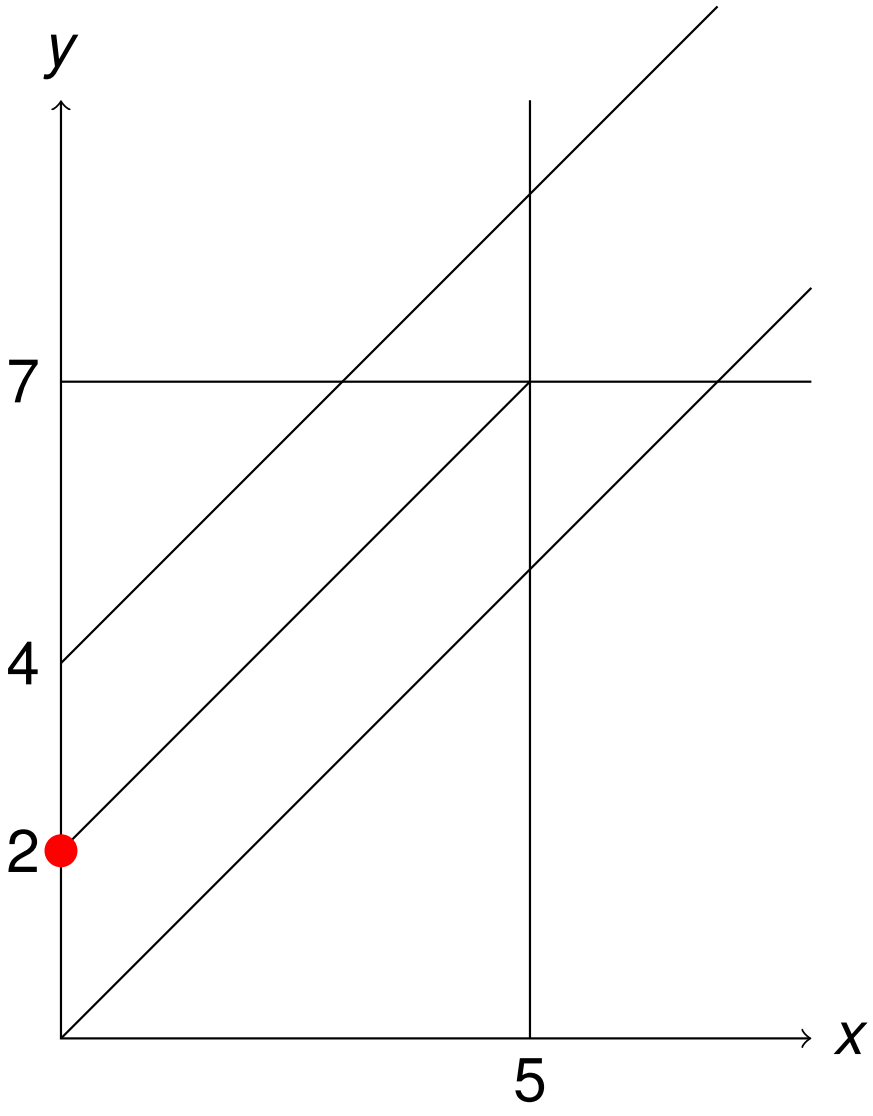


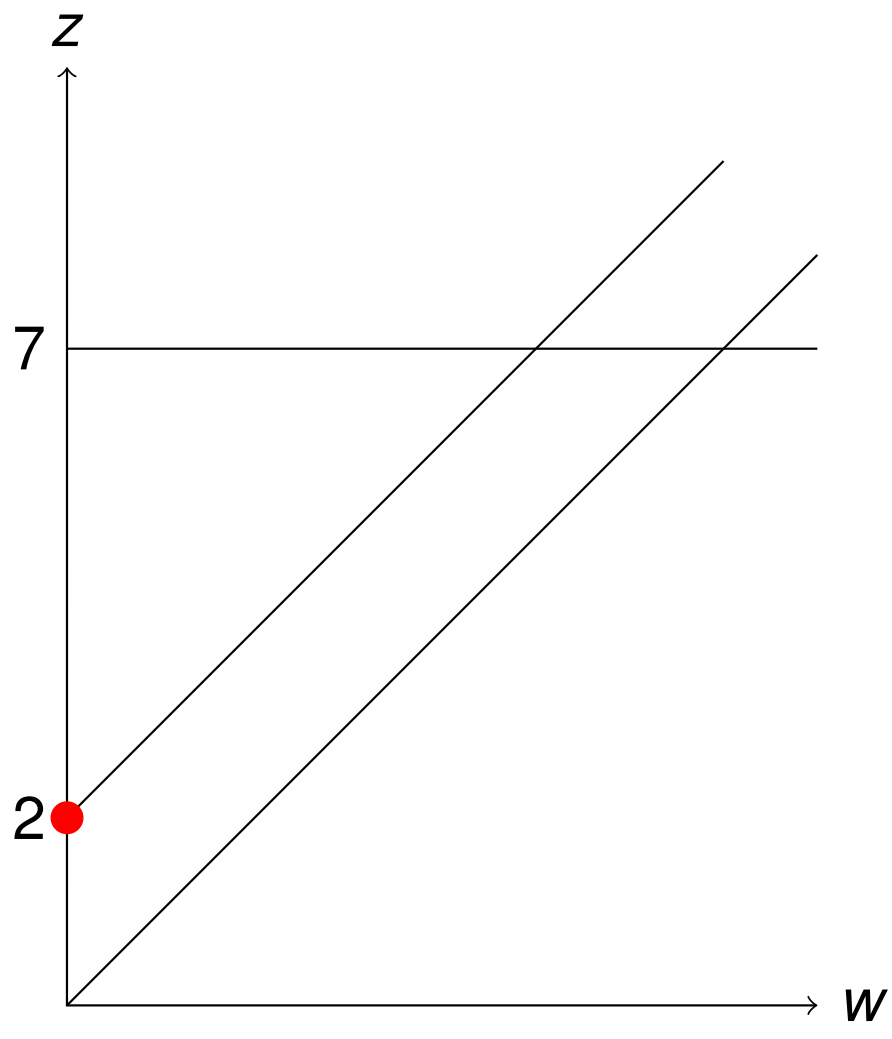
CP bisimulation (Can be a game)



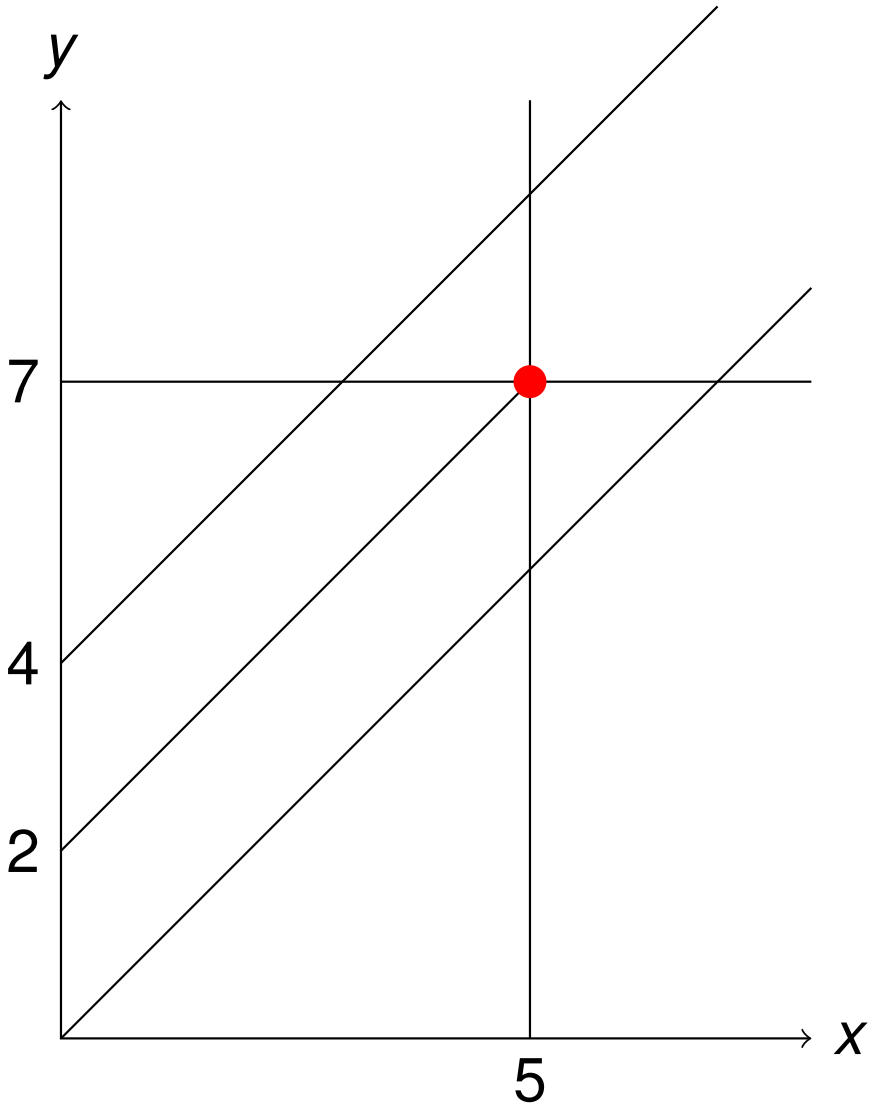


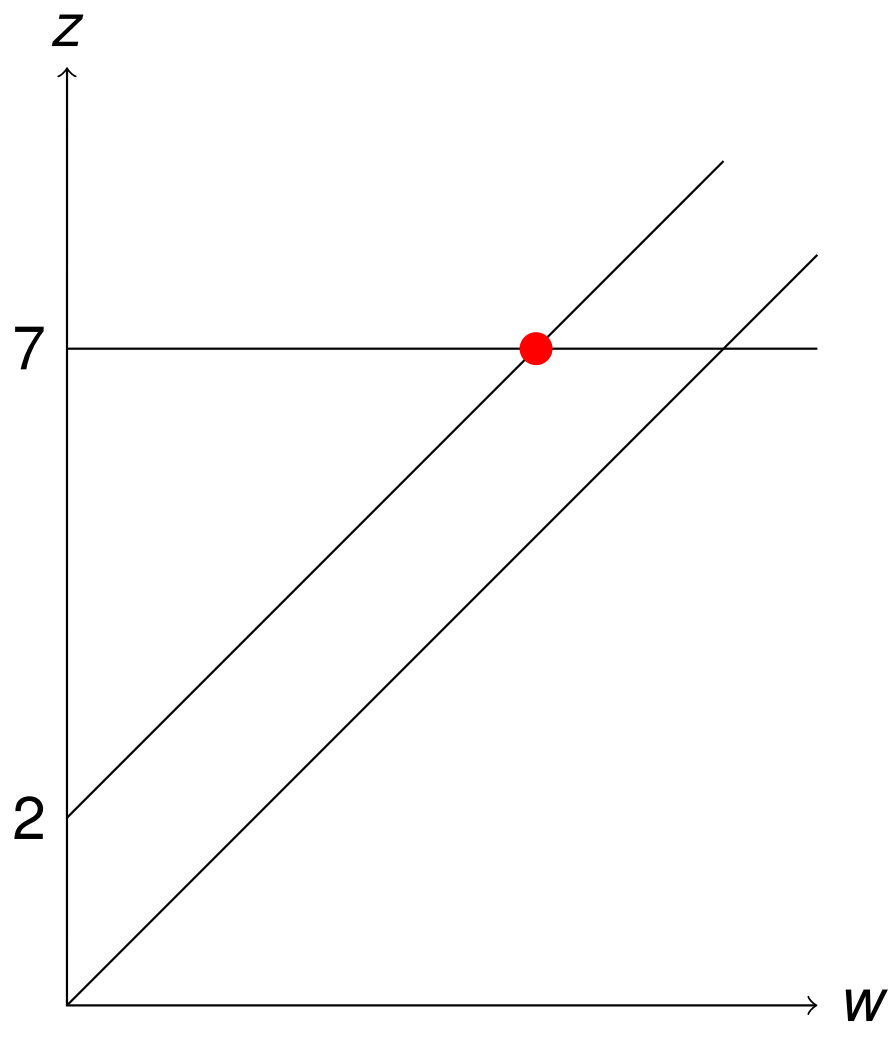
CP bisimulation (Can be a game)



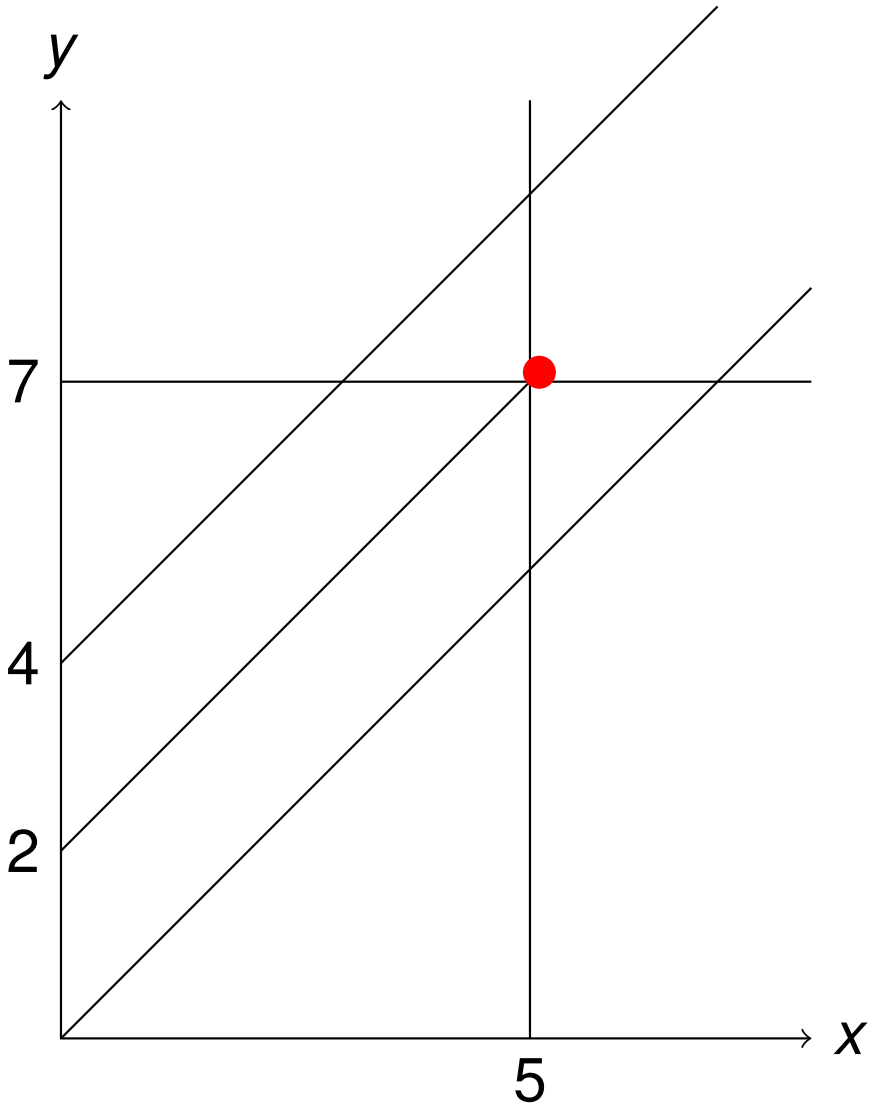


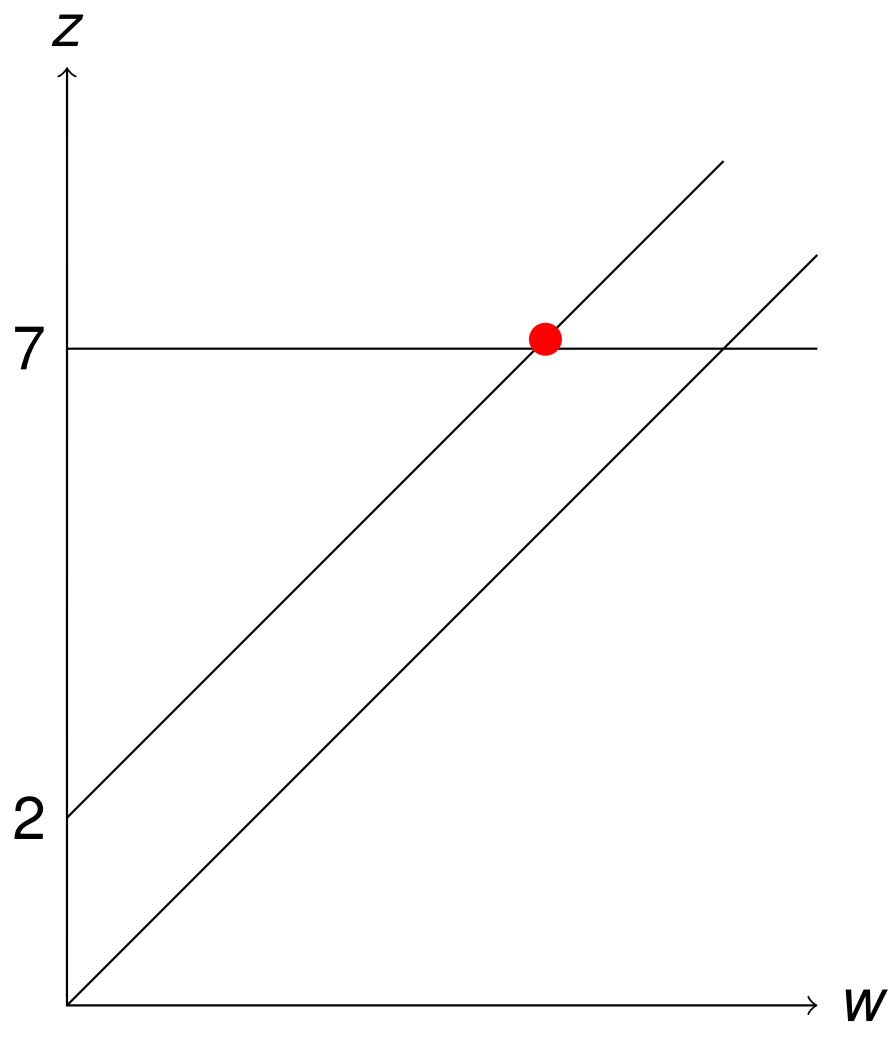
CP bisimulation (Can be a game)





CP bisimulation (Can be a game)





CP-bisimulation

Lemma

For two timed automata states p and q, $p \sim_t q \Rightarrow p\mathcal{R}q$, where $\mathcal{R}$ is a corner point bisimulation relation.

Lemma

For two timed automata states p and q, $p\mathcal{R}q \implies p \sim_t q$, where $\mathcal{R}$ is a corner point bisimulation relation.

Theorem

For two timed automata states p and q, p and q are timed bisimilar if and only if p and q are cp-bisimilar.

Timed HML Distinguishing formula

- Abstract syntax

  $\phi ::= \mathtt{tt} \mid \mathtt{ff} \mid \phi \wedge \psi \mid \phi \vee \psi \mid \langle a \rangle \phi \mid [a]\phi \mid \exists \phi \mid \forall \phi \mid x \ \underline{\mathtt{in}} \ \phi \mid g$

  where $a \in Act$, $x \in D$, set of formula clocks and $g \in \mathcal{B}(D)$.

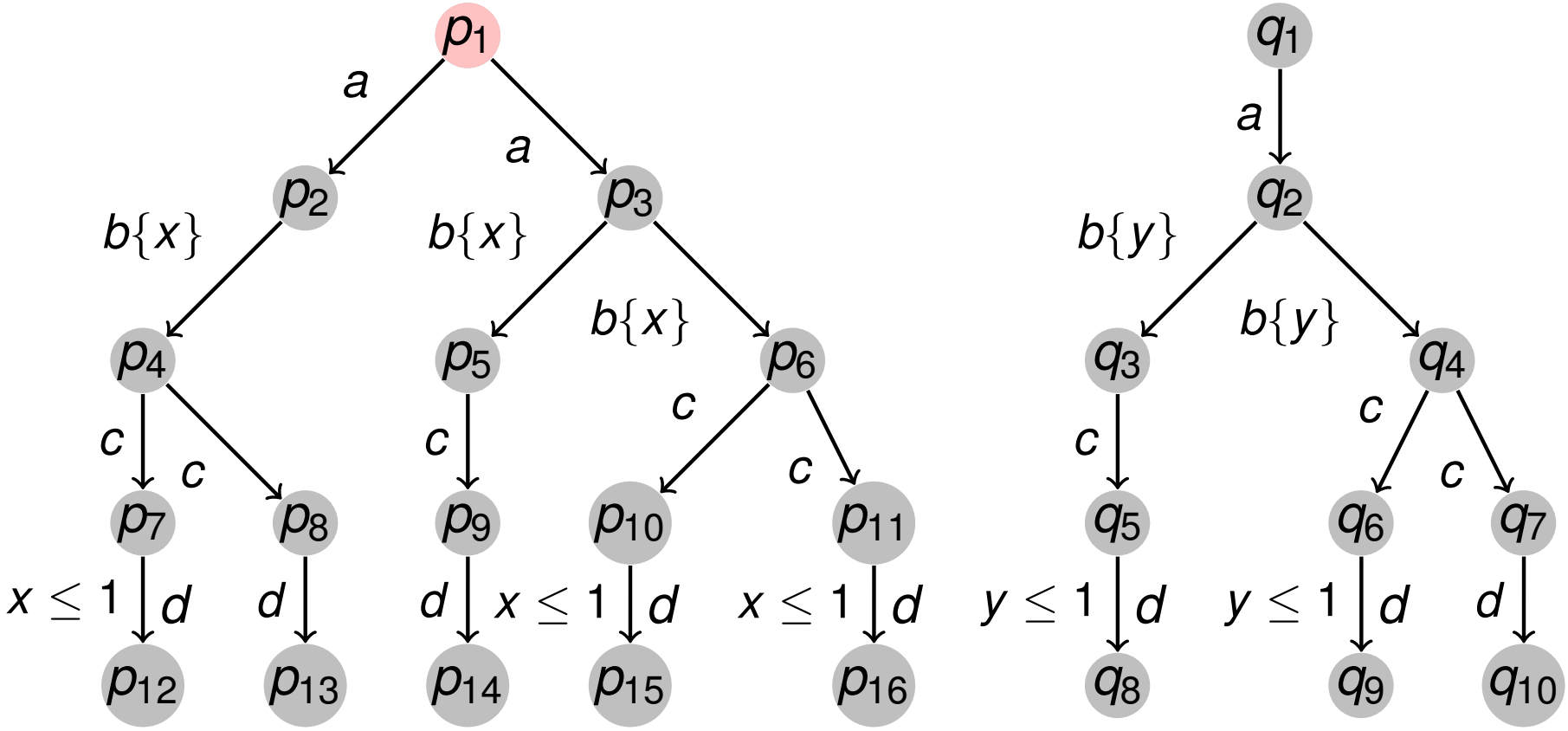
- Work primarily to generate a *characteristic formula*

- Two timed automata states satisfy the same set of formula if and only if they are timed bisimilar.

Generating Distinguishing formula

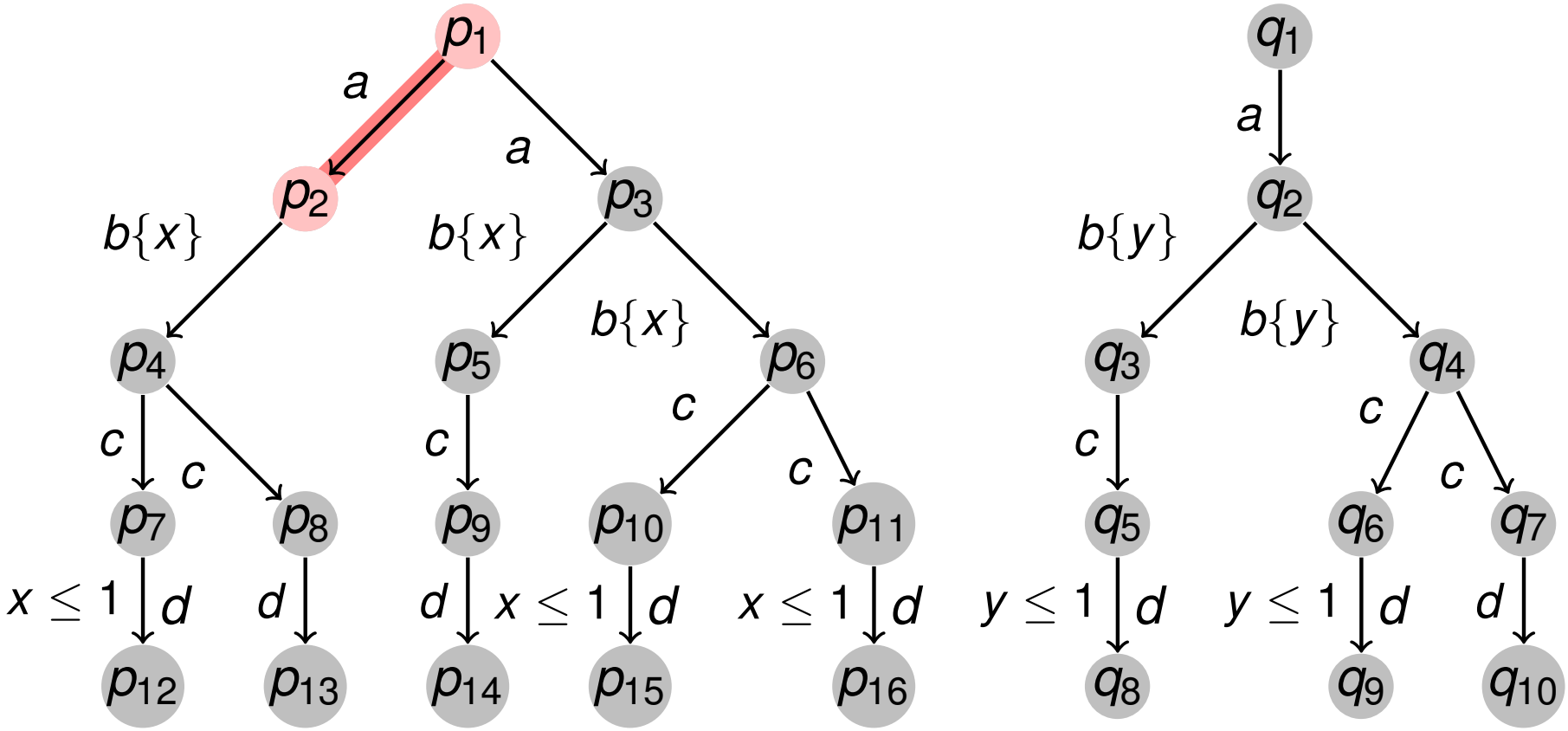
- Region based distinguishing formula synthesis in Epsilon tool
- Generate formula that will be satisfied by the timed automata that is chosen by the challenger in the first round in the cp-bisimulation game.
- The formula generated is based on the visible actions and the corner-point delays and hence is not region based.

Generating Distinguishing Formula



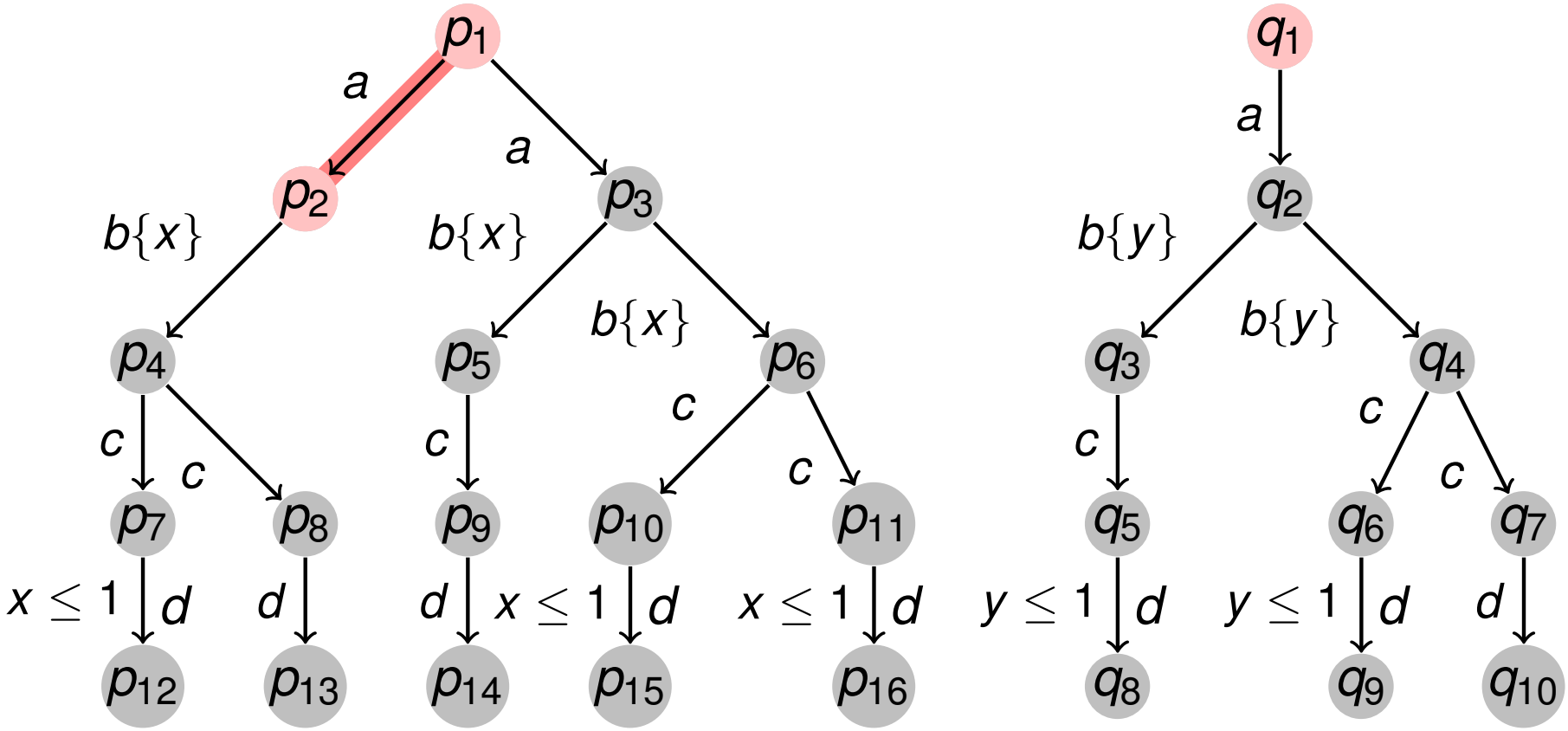
x_1 in ( )

Generating Distinguishing Formula



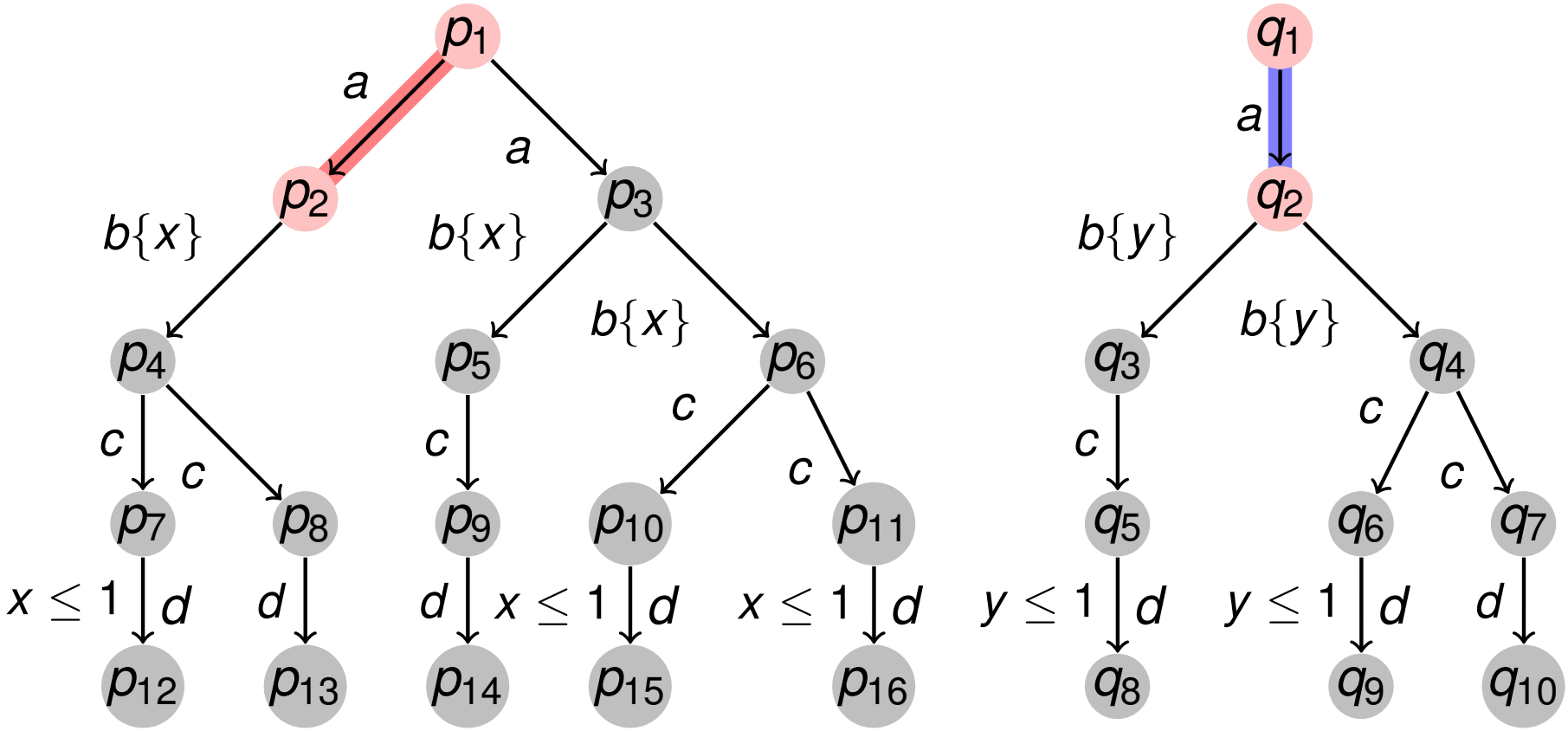
x_1 in $(\langle a \rangle)$

Generating Distinguishing Formula



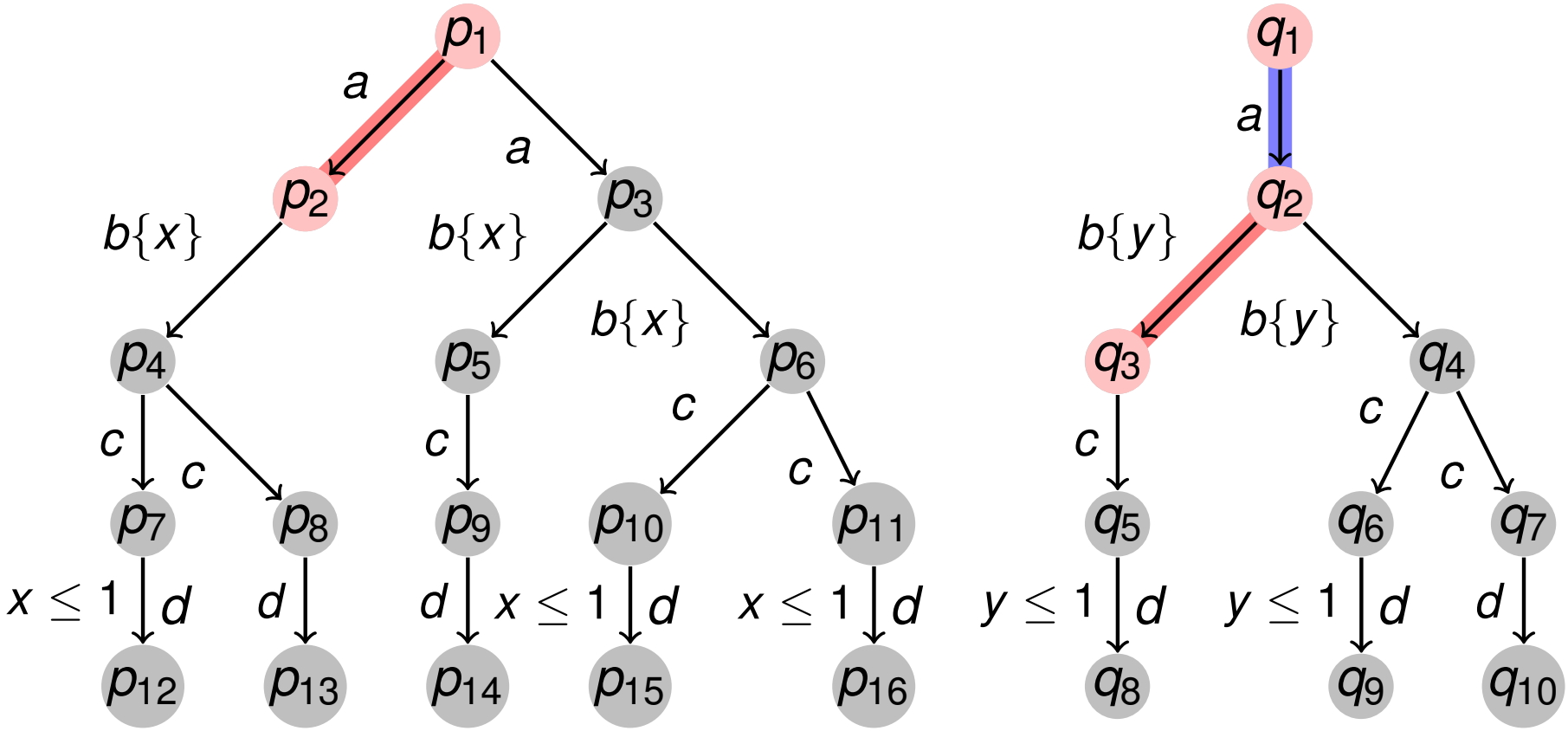
x_1 in $(\langle a \rangle)$

Generating Distinguishing Formula



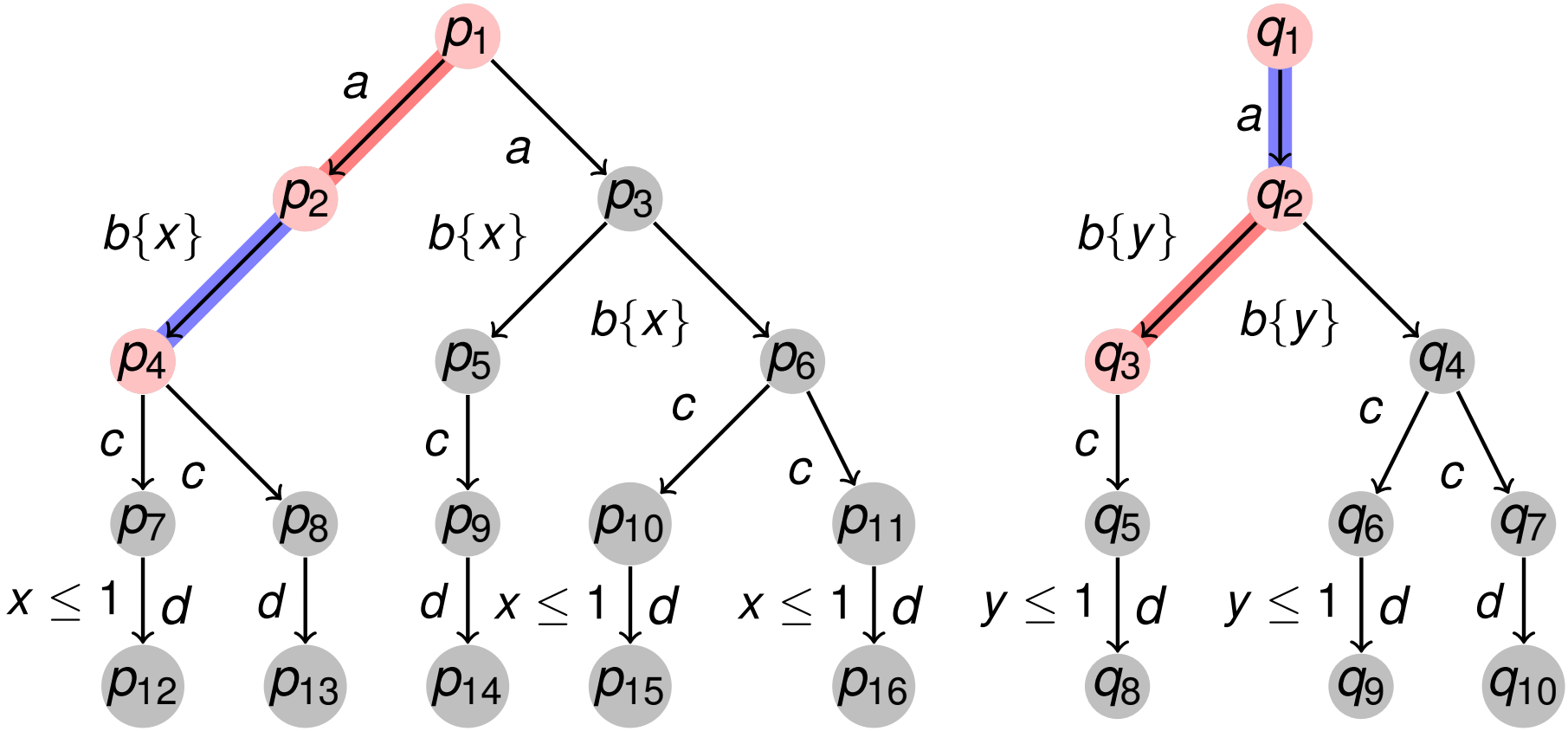
x_1 in $(\langle a \rangle)$

Generating Distinguishing Formula



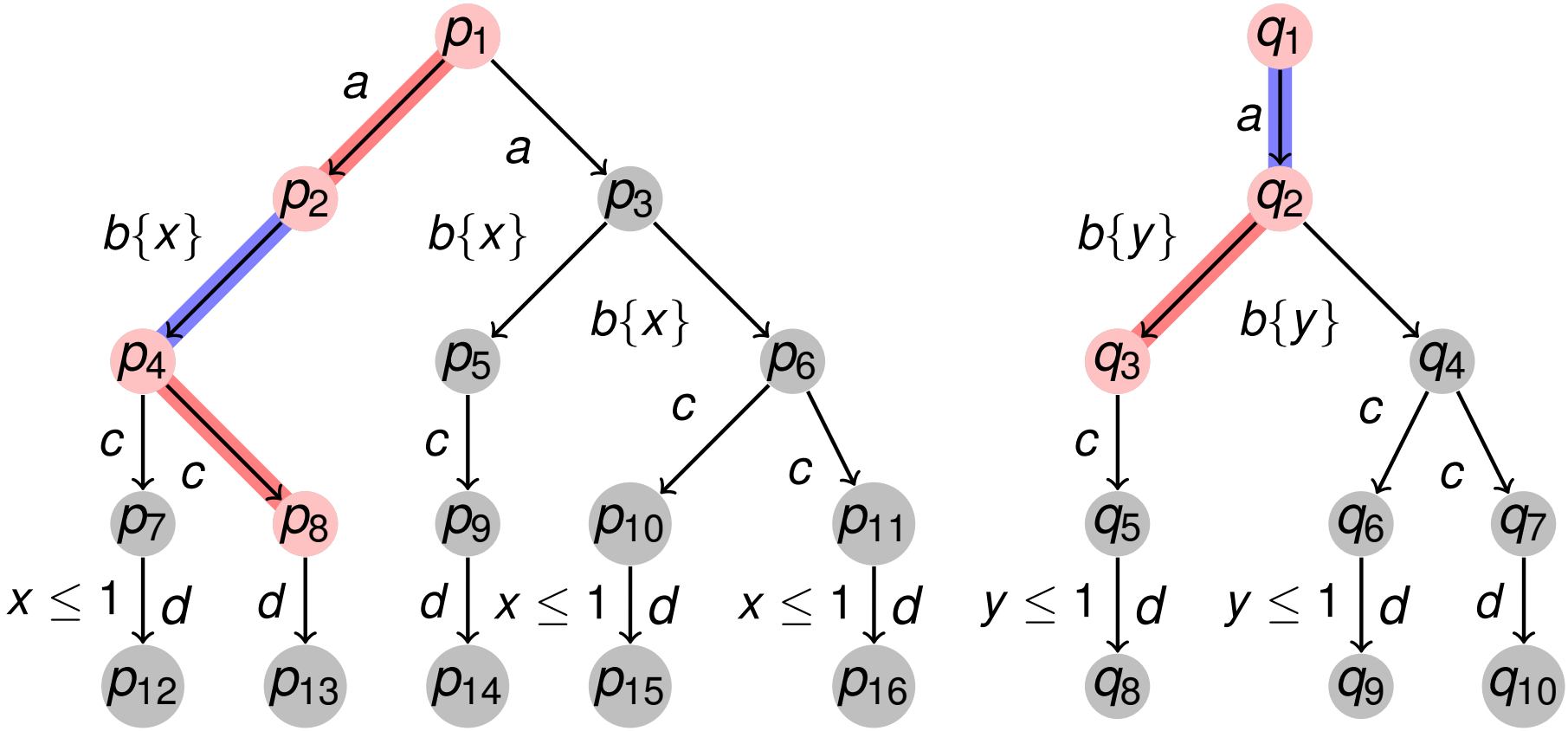
x_1 in $(\langle a \rangle\ [b]\ x_2)$

Generating Distinguishing Formula



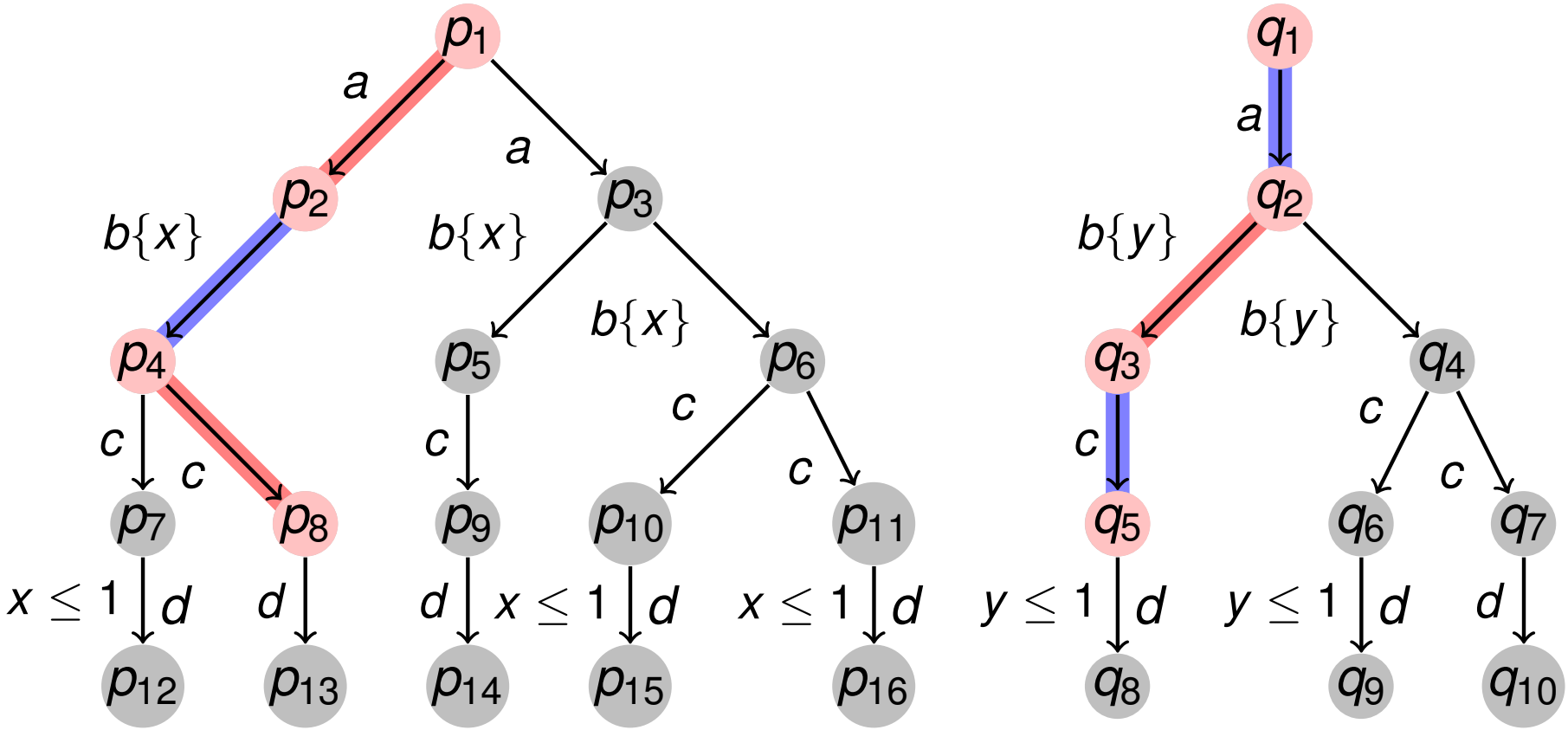
$$x_1 \ \underline{\texttt{in}} \ (\langle a \rangle \ [b] \ x_2)$$

Generating Distinguishing Formula



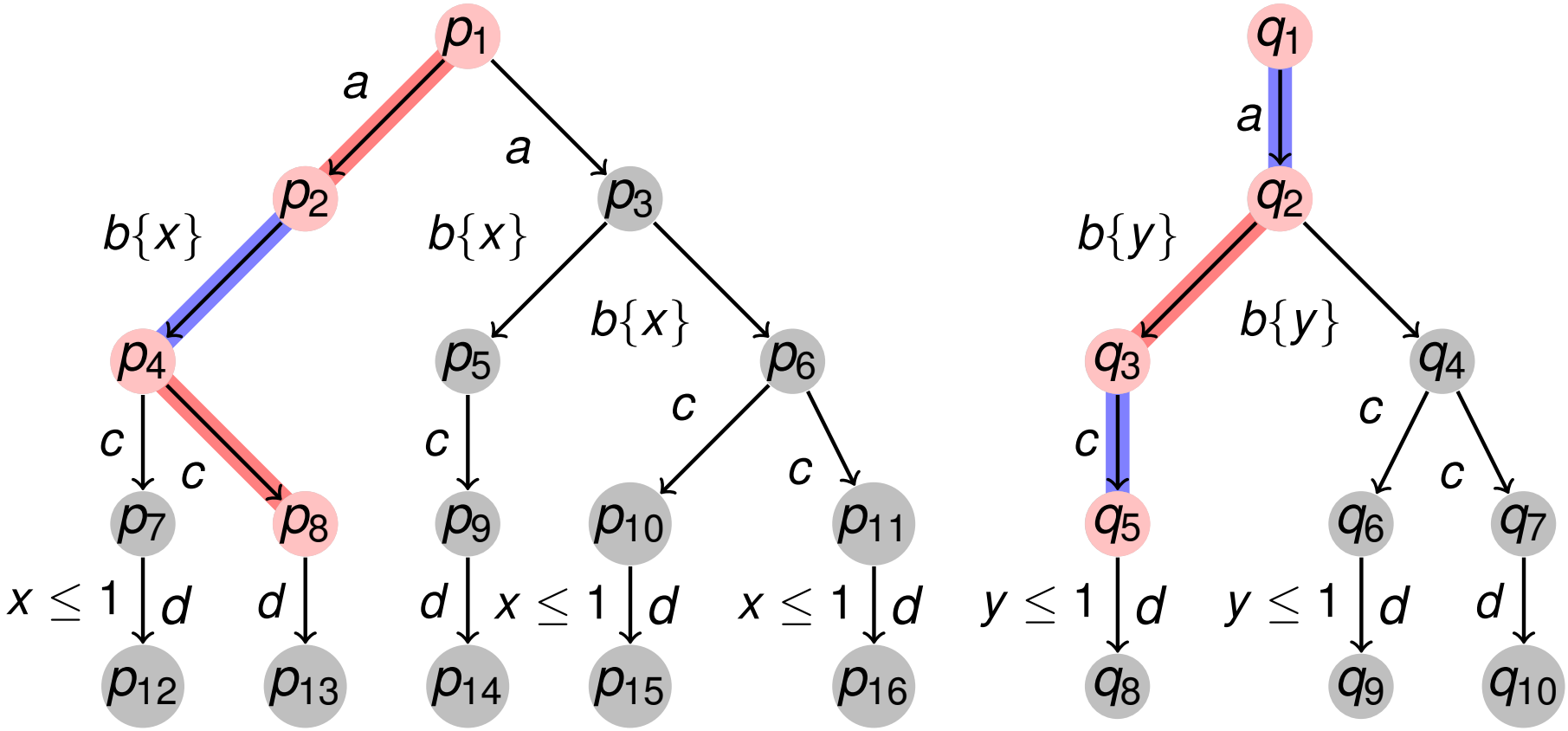
$$x_1 \ \underline{\texttt{in}} \ (\langle a \rangle \ [b] \ x_2 \ \underline{\texttt{in}} \ (\langle c \rangle))$$

Generating Distinguishing Formula



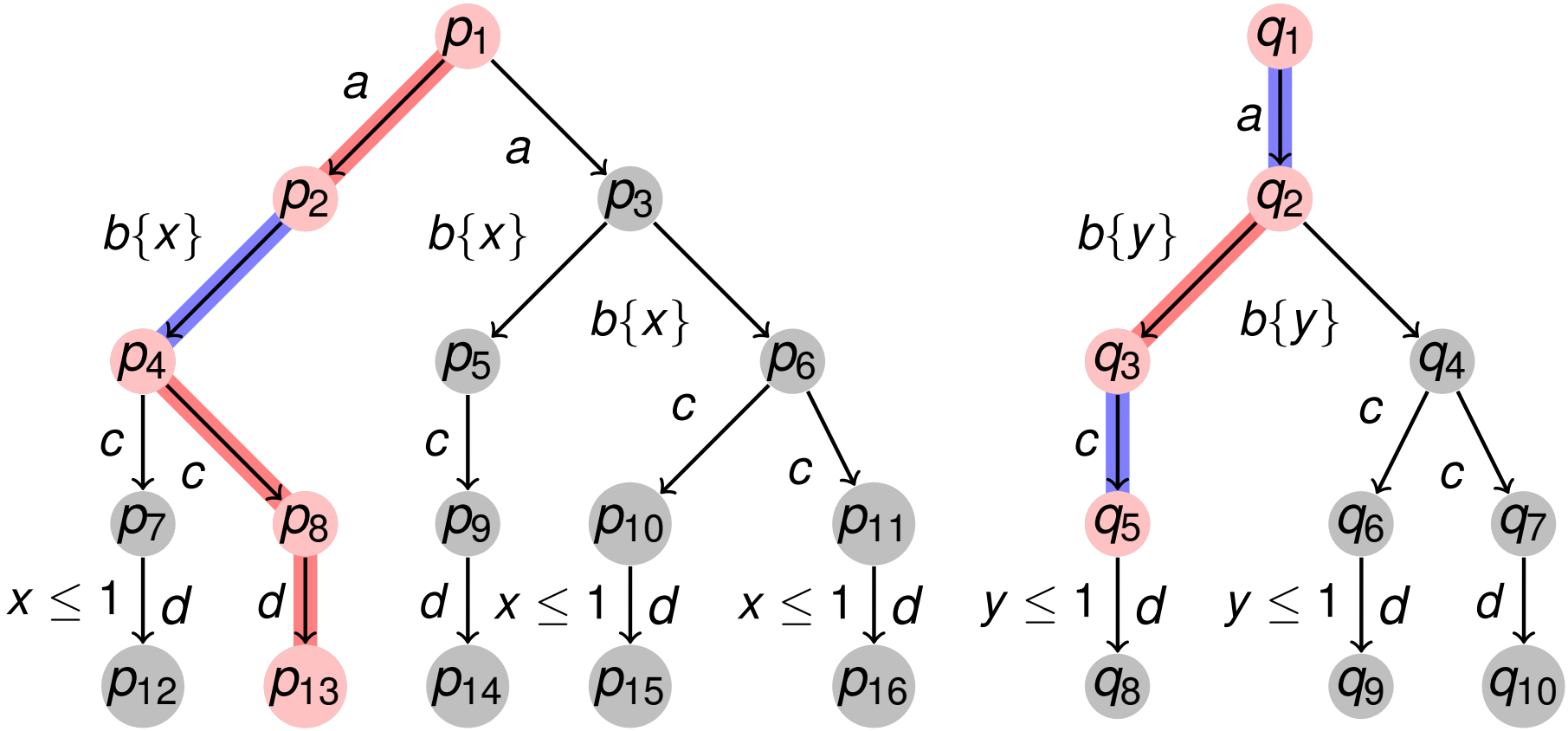
$$x_1 \ \underline{\texttt{in}} \ (\langle a \rangle \ [b] \ x_2 \ \underline{\texttt{in}} \ (\langle c \rangle))$$

Generating Distinguishing Formula



$$x_1 \,\underline{\texttt{in}}\, (\langle a \rangle\, [b]\, x_2 \,\underline{\texttt{in}}\, (\langle c \rangle \exists (1 < x_2 < 2)))$$

Generating Distinguishing Formula



$$x_1 \,\underline{\texttt{in}}\, (\langle a \rangle\, [b]\, x_2 \,\underline{\texttt{in}}\, (\langle c \rangle \exists (1 < x_2 < 2 \wedge \langle d \rangle\, \texttt{tt})))$$

Deciding Time Abstracted Bisimulation

- All the valuations in a zone are time abstracted bisimilar to each other.
- Thus two zone graphs being strongly bisimilar implies that they are time abstracted bisimilar to each other.
- ε is considered to be similar to visible action.

Deciding Time Abstracted Relations

- Let $\mathcal{R} \subseteq S_1 \times S_2$ be a symmetric relation. Two nodes $(s_1, s_2) \in \mathcal{R}$ if and only if
  $\forall a \in Act, \forall s_1' [s_1 \stackrel{a}{\rightarrow} s_1' \Rightarrow \exists s_2' \,.\, s_2 \stackrel{\beta}{\rightarrow} s_2'$ and $(s_1', s_2') \in \mathcal{R}]$
  and $\forall s_1', [s_1 \stackrel{\varepsilon}{\rightarrow} s_1' \Rightarrow \exists s_2' \,.\, s_2 \stackrel{\varepsilon}{\rightarrow} s_2'$ and $(s_1', s_2') \in \mathcal{R}]$

- Two states p and q are strong time abstracted bisimilar iff $Z_{A_1,p}\ \mathcal{R}\ Z_{A_2,q}$ and $\stackrel{\beta}{\rightarrow}$ is the transition $\stackrel{a}{\rightarrow}$,

- p and q are time abstracted delay bisimilar iff $\stackrel{\beta}{\rightarrow}$ is the sequence of transitions $\stackrel{\varepsilon}{\rightarrow}\stackrel{a}{\rightarrow}$

- p and q are time abstracted observational bisimilar iff $\stackrel{\beta}{\rightarrow}$ is $\stackrel{\varepsilon}{\rightarrow}\stackrel{a}{\rightarrow}\stackrel{\varepsilon}{\rightarrow}$.

Complexity

- Time bisimulation through product construction on regions: EXPTIME
- Zone graph is *at least as good as* region graph
- Also we remove product construction
- Our algorithm too decides the relation in EXPTIME: not asymptotically better, but it should have better running time
- Speedup: requires experimental verification

Game Characterization

Game Characterization

- Easy comparison between relations
- Similar to Stirling's bisimulation game for discrete relations
- Games for Van Glabeek's spectrum for discrete relations (Chen and Deng, '08)
- Game can be parameterized using several parameters in a template
- Leads to defining several new relations

Game Template

- $n - \Gamma_k^\alpha$
- n : number of alternations.
- $k \in \mathbb{N} \cup \{\infty\}$: number of rounds; $n \leq k - 1$ when $k \neq \infty$.
- α : an ordered tuple $\langle \alpha_1, \alpha_2 \rangle$.

Hierarchy of Games

Lemma

$\Gamma^{\alpha}_{\infty} \longrightarrow n-\Gamma^{\alpha}_{\infty} \longrightarrow (n-1)-\Gamma^{\alpha}_{\infty}$, *for all* $n > 0$

$\Gamma^{\alpha}_{k} \longrightarrow n-\Gamma^{\alpha}_{k} \longrightarrow (n-1)-\Gamma^{\alpha}_{k}$, *for all* $k > 0$, $n < k$

Lemma

$\Gamma^{\alpha}_{\infty} \longrightarrow \Gamma^{\alpha}_{k} \longrightarrow \Gamma^{\alpha}_{k-1}$, *for all* $k > 0$

$n-\Gamma^{\alpha}_{\infty} \longrightarrow n-\Gamma^{\alpha}_{k} \longrightarrow n-\Gamma^{\alpha}_{k-1}$, *for all* $k > 0$, $n < k$

Lemma

$n - \Gamma_k^{\langle a/d, a/d\rangle} \longrightarrow n - \Gamma_k^{\langle a/\varepsilon, a/\varepsilon\rangle} \longrightarrow n - \Gamma_k^{\langle a/\varepsilon, \varepsilon.a/\varepsilon\rangle} \longrightarrow$
$n - \Gamma_k^{\langle a/\varepsilon, \varepsilon.a.\varepsilon/\varepsilon\rangle}$

By varying n and k, one can create an infinite hierarchy of games.

Game Hierarchy

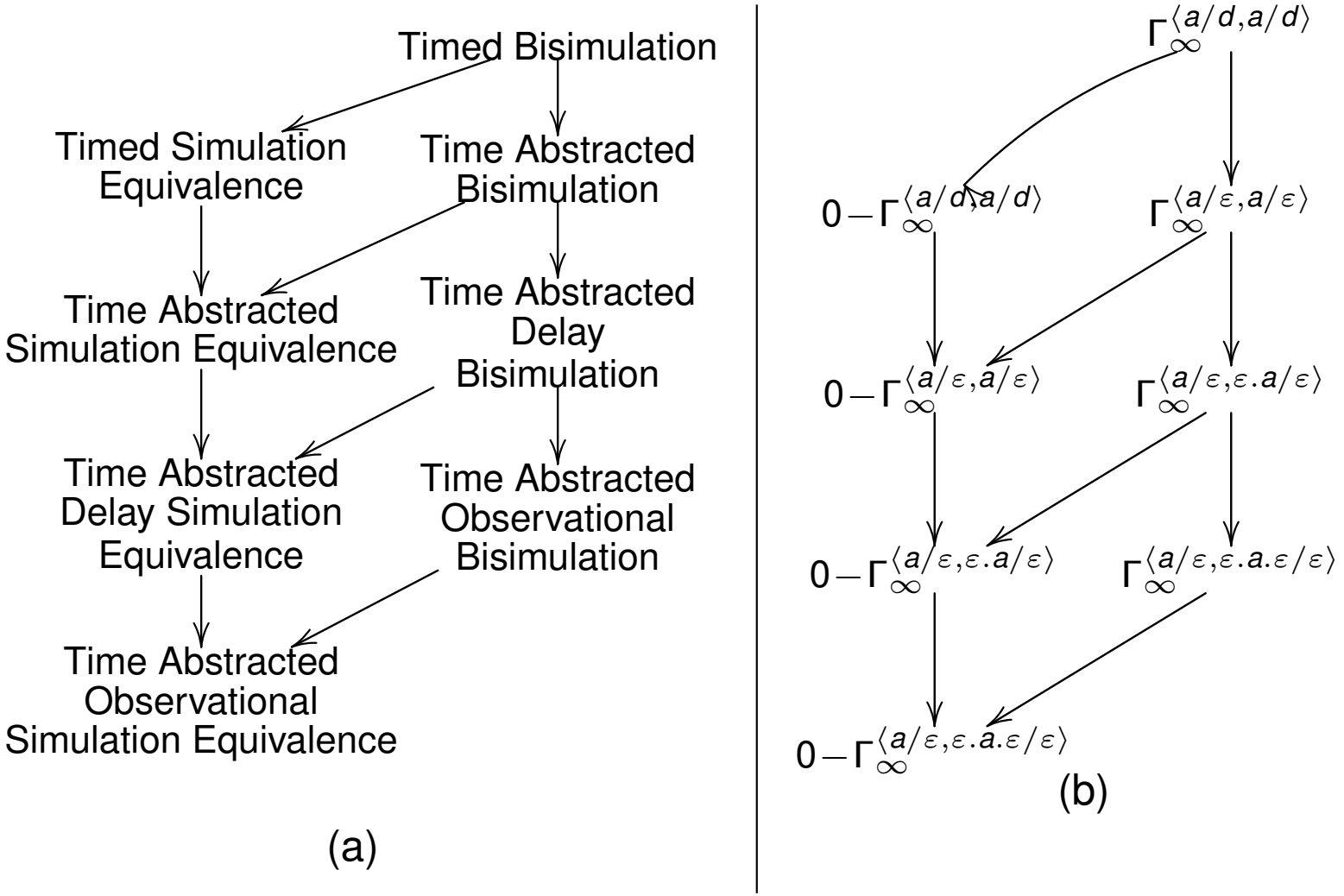
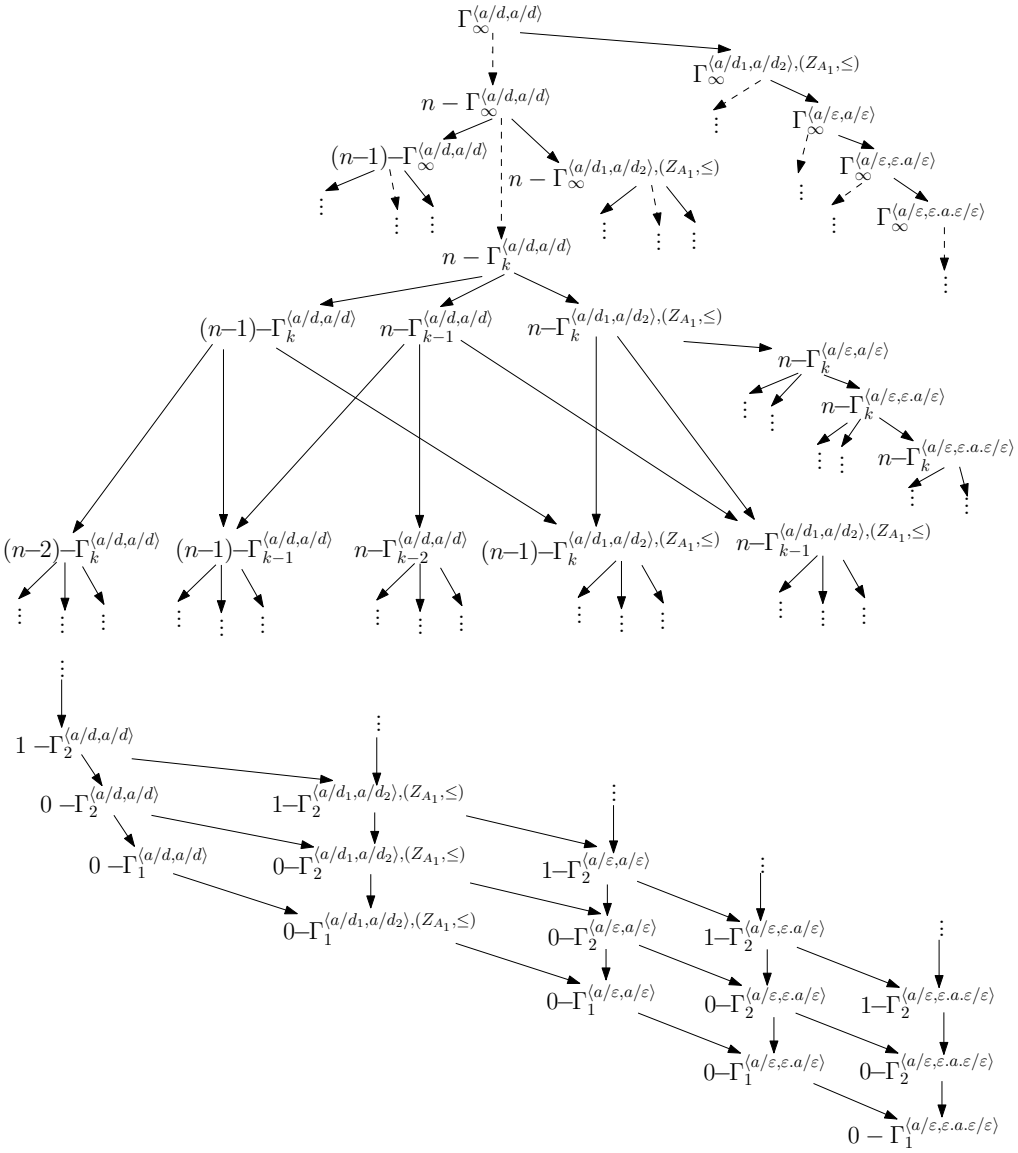


Figure: (a) presents the spectrum of timed relations and (b) shows timed games corresponding to these relations

Infinite Hierarchy of Timed Games



Conclusion and Future Work

Discussion

- Unified zone based approach
- Remove product construction
- Game-theoretic formulation
- Relationships between relation over timed automata
- Study and define new relations

Tool Development (Ongoing)

- Tool for deciding relations over network of timed automata.
- Timed and time-abstracted relations, Equivalences and preorders
- Game hierarchy
- Distinguishing formula for various relations.
- Current Status: Zone graphs, deciding the time abstracted relations
- Benchmarking with similar tools like Epsilon, Standard applications for benchmarking

Future Work

- Timed Hennessy Milner logic based distinguishing formula for other relations
- Counterexample generation : implementation to conform to specification
- Study the weak relations for timed automata

Thank You