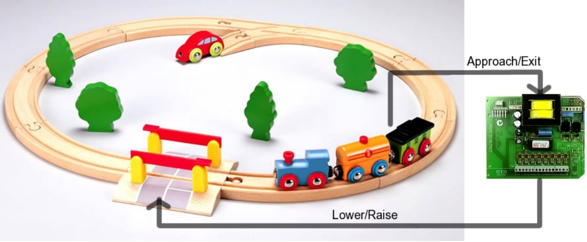
Topics in Timed Automata

B. Srivathsan

RWTH-Aachen

Software modeling and Verification group



Reachability: Does something **bad** happen?

"The gate is still open when the train is 2 minutes away from the crossing"

This problem is PSPACE-complete

A theory of timed automata

R. Alur and D.L. Dill, *TCS'94*

Tools

- UPPAAL:

  Uppsaala university *(Sweden)*, Aalborg university *(Denmark)*

- KRONOS:

  Verimag *(France)*

- RED

  National Taiwan University *(Taiwan)*

- Rabbit

  Brandenburg TU Cottbus *(Germany)*

Tools

- UPPAAL:

  Uppsaala university *(Sweden)*, Aalborg university *(Denmark)*

- KRONOS:

  Verimag *(France)*

- RED

  National Taiwan University *(Taiwan)*

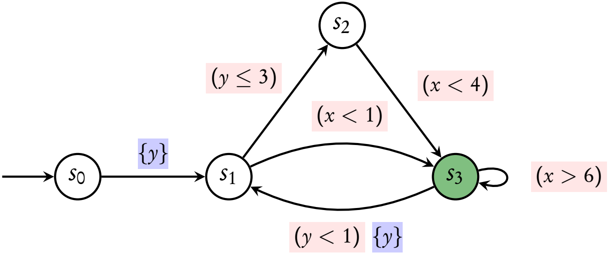
- Rabbit

  Brandenburg TU Cottbus *(Germany)*

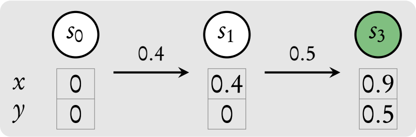
and still research on for efficient algorithms . . .

Lecture 6:
Reachability

Timed Automata



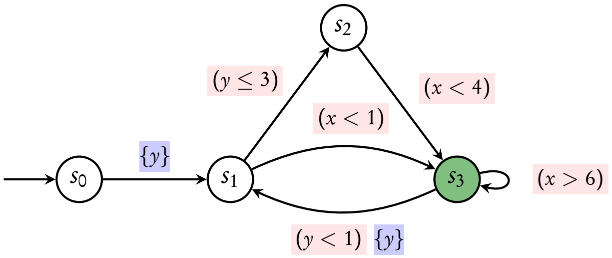
Run: finite sequence of transitions



- **accepting** if ends in green state

Reachability problem

Given a TA, does it **have** an **accepting** run

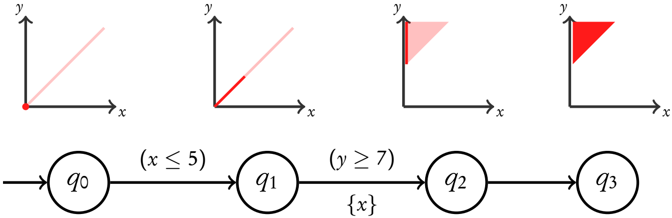


Theorem [AD94]

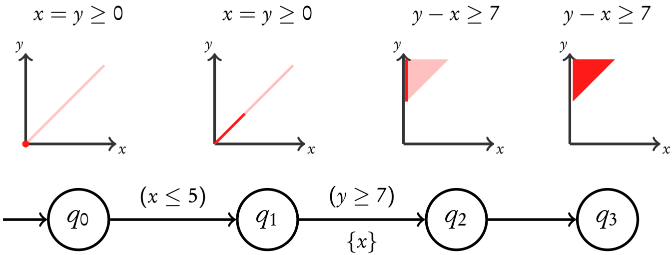
This problem is **PSPACE-complete**

first solution based on Regions

Key idea: Maintain **sets of valuations** reachable along a path

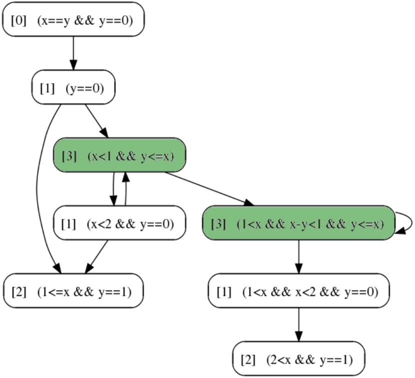


Key idea: Maintain **sets of valuations** reachable along a path



Easy to describe **convex** sets

Zones and zone graph



- Zone: set of valuations defined by conjunctions of constraints:

$$x \sim c$$
$$x - y \sim c$$

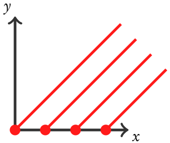
e.g. $(x - y \geq 1) \wedge (y < 2)$

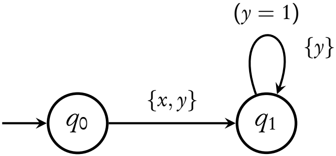
- Representation: by DBM [Dil89]

Sound and complete [DT98]

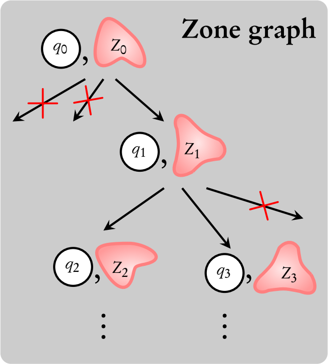
Zone graph preserves state **reachability**

Problem of non-termination



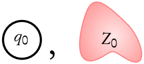


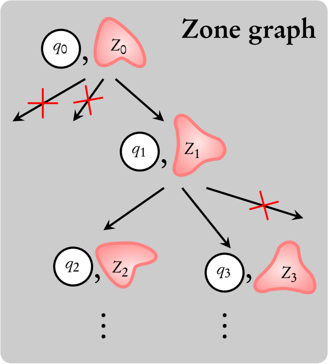
Abstractions



potentially infinite...

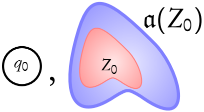
Abstractions

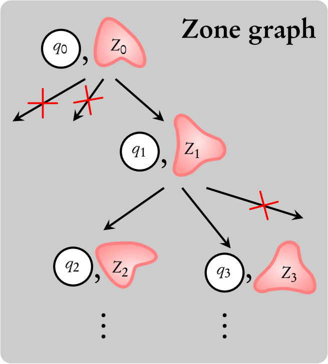




potentially infinite...

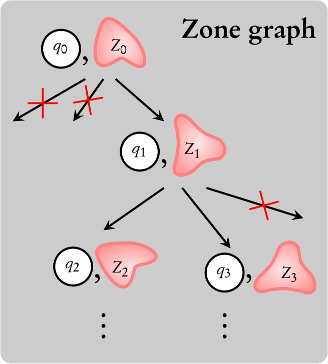
Abstractions



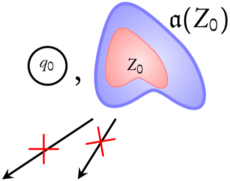


potentially infinite...

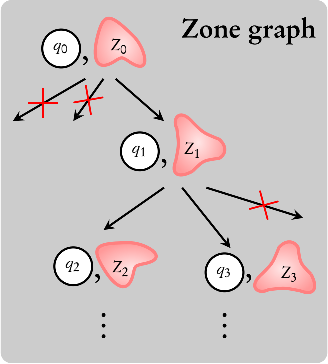
Abstractions



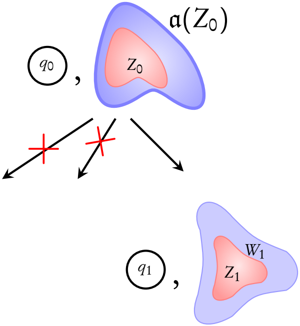
potentially infinite...



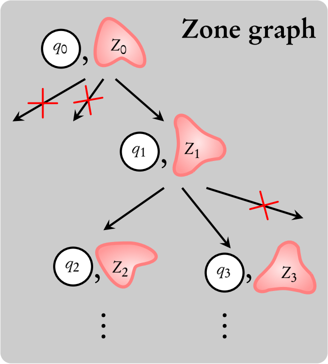
Abstractions



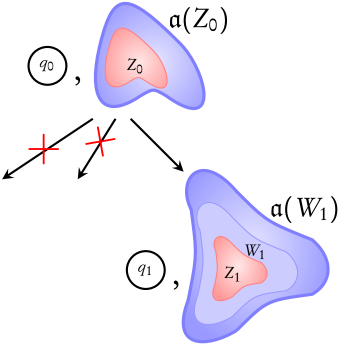
potentially infinite...



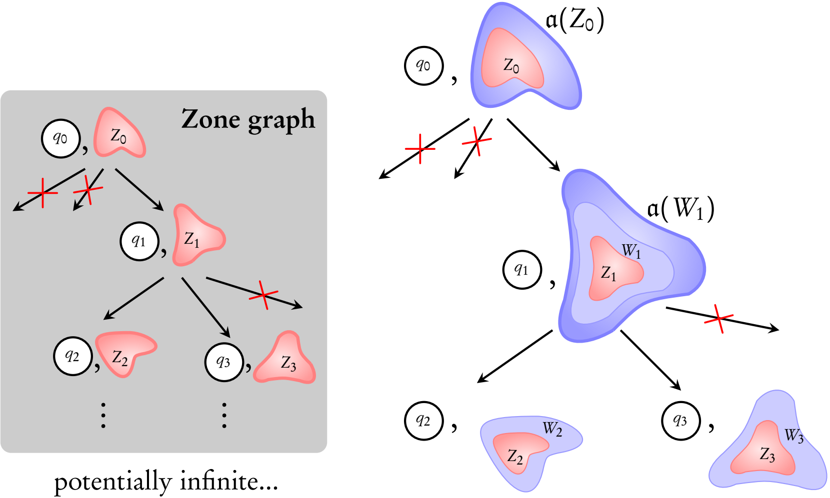
Abstractions



potentially infinite...

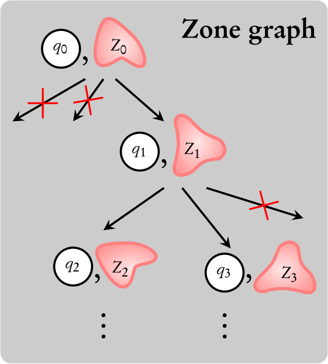


Abstractions

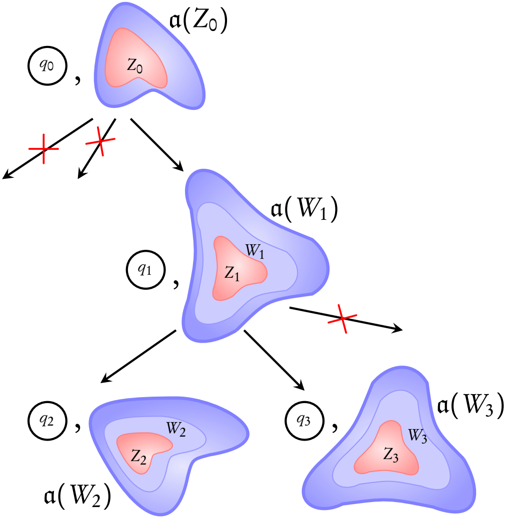


potentially infinite...

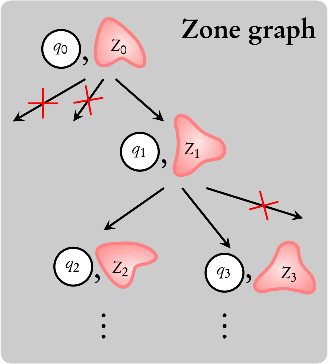
Abstractions



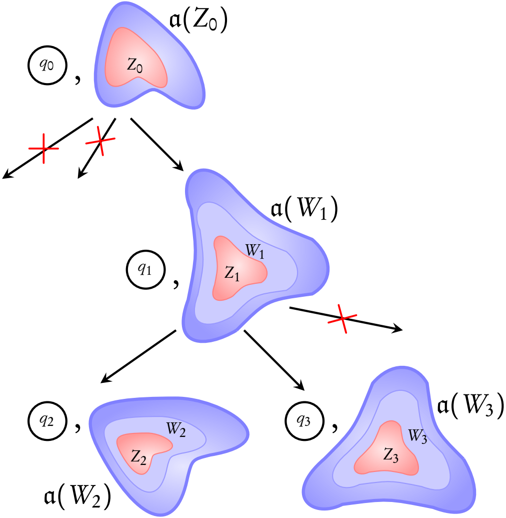
potentially infinite...



Abstractions

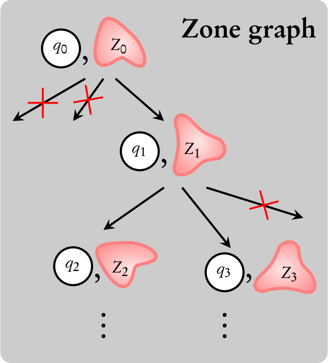


potentially infinite...

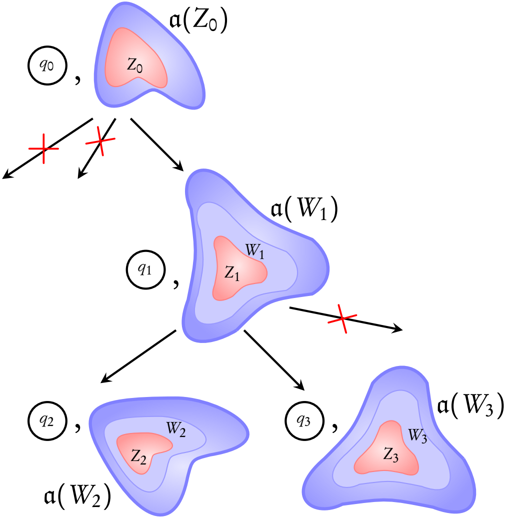


Find $\mathfrak{a}$ such that number of **abstracted** sets is **finite**

Abstractions



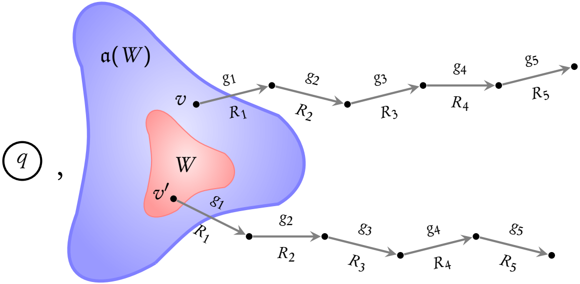
potentially infinite...



Coarser the abstraction, **smaller** the abstracted graph

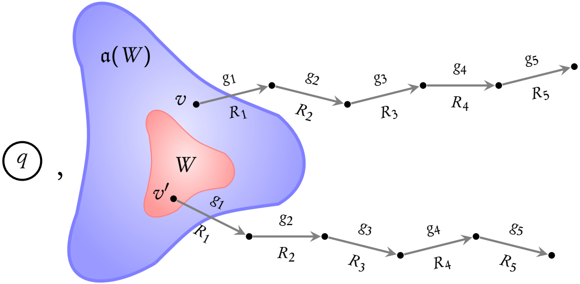
Condition 1: Abstractions should have **finite range**

Condition 2: Abstractions should be sound ⇒ $\mathfrak{a}(W)$ can contain only valuations **simulated** by W



Condition 1: Abstractions should have **finite range**

Condition 2: Abstractions should be sound ⇒ $\mathfrak{a}(W)$ can contain only valuations **simulated** by W

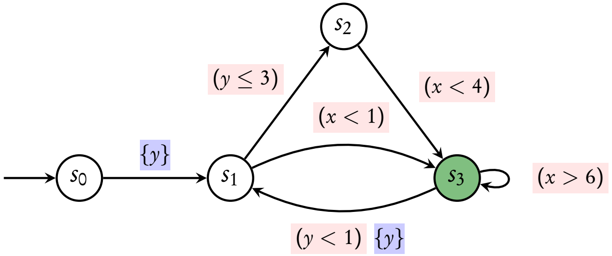


Question: Why not add **all** the valuations **simulated** by W?

Bounds and abstractions

Theorem [LS00]

Coarsest simulation relation is **EXPTIME-hard**



Bounds and abstractions

Theorem [LS00]

Coarsest simulation relation is **EXPTIME-hard**

$(y \leq 3)$ $(x < 4)$

$(x < 1)$

$(x > 6)$

$(y < 1)$

Bounds and abstractions

Theorem [LS00]

Coarsest simulation relation is **EXPTIME-hard**

$(y \leq 3)$

$(x < 4)$

$(x < 1)$

$(x > 6)$

$(y < 1)$

M-bounds [AD94]

$$M(x) = 6, \ M(y) = 3$$

$$v \preccurlyeq_M v'$$

Bounds and abstractions

Theorem [LS00]

Coarsest simulation relation is **EXPTIME-hard**

$(y \leq 3)$

$(x < 4)$

$(x < 1)$

$(x > 6)$

$(y < 1)$

M-bounds [AD94]

$$M(x) = 6, \; M(y) = 3$$

$$v \preccurlyeq_M v'$$

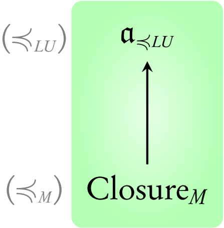
LU-bounds [BBLP04]

$$L(x) = 6, \; L(y) = -\infty$$

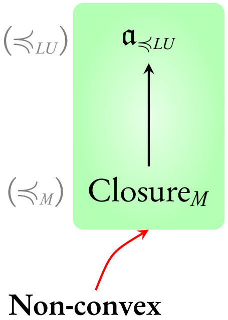
$$U(x) = 4, \; U(y) = 3$$

$$v \preccurlyeq_{LU} v'$$

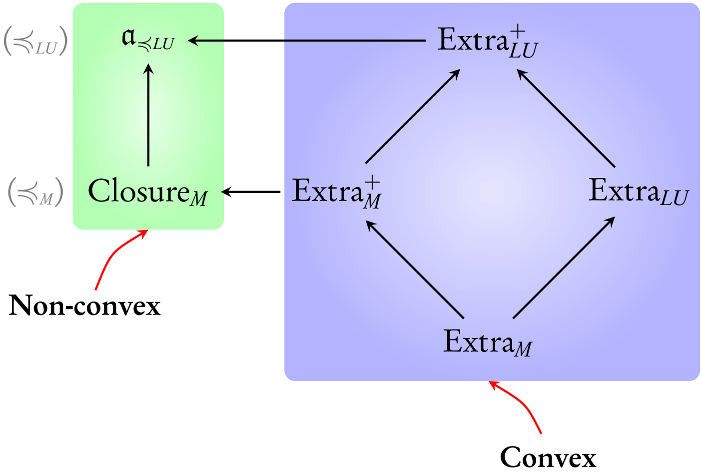
Abstractions in literature [BBLP04, Bou04]



Abstractions in literature [BBLP04, Bou04]



Abstractions in literature [BBLP04, Bou04]



Only convex abstractions used in **implementations!**

Timed automata

↓

Zone graph

↓

Problem of non-termination → Use finite abstractions

Zones
are
efficient

↓

Bounds as parameters

↓

Restriction to convex abstractions

Non-convex abstr.
are
coarser

Timed automata

↓

Zone graph

↓

Problem of non-termination ⟶ Use finite abstractions

↓

Bounds as parameters

↓

Restriction to convex abstractions

Zones are **efficient**

Non-convex abstr. are **coarser**

Question: Can we benefit from both together?

In this lecture...

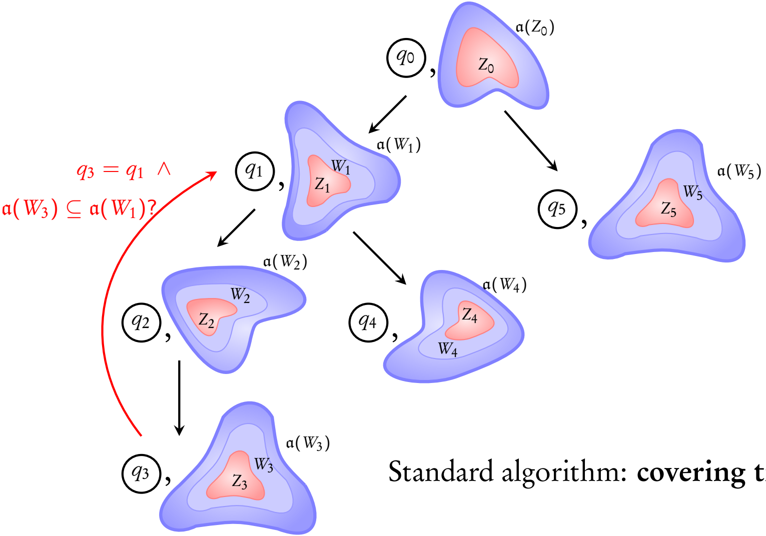
Efficient use of the **non-convex** Closure approximation

Using non-convex approximations for efficient analysis of timed automata

F. Herbreteau, D. Kini, B. Srivathsan, I. Walukiewicz. *FSTTCS'11*

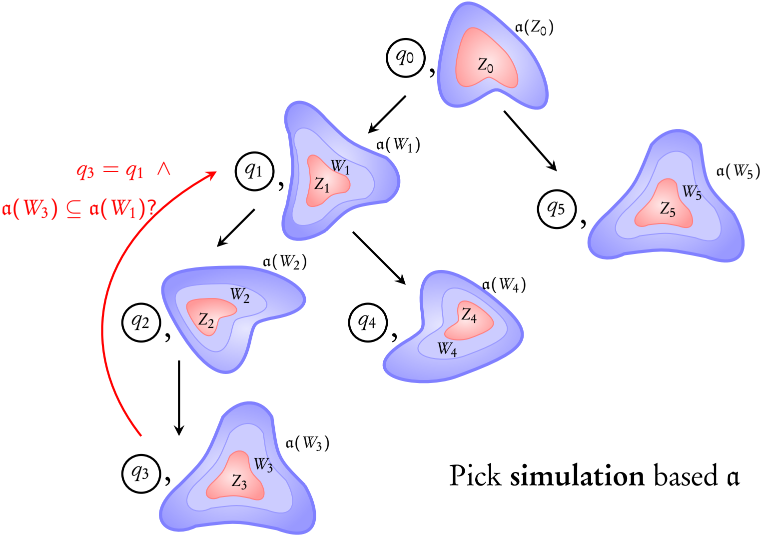
Observation 1: We can use abstractions **without storing** them

Using non-convex abstractions



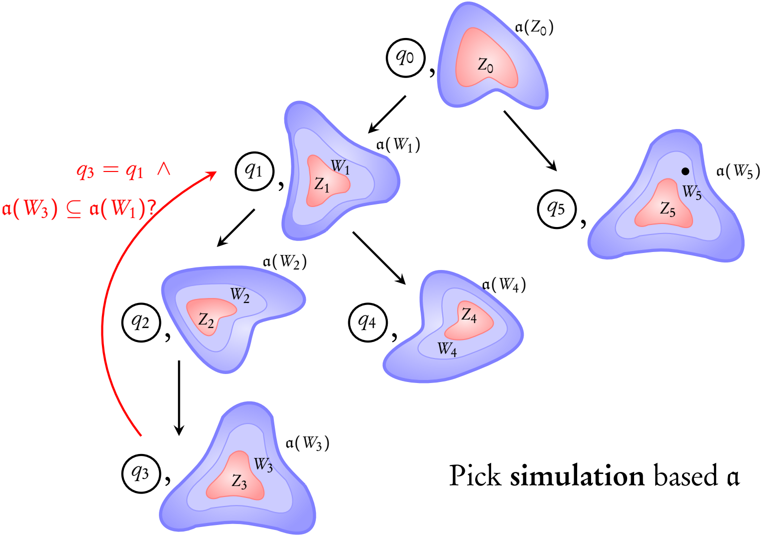
Standard algorithm: **covering tree**

Using non-convex abstractions



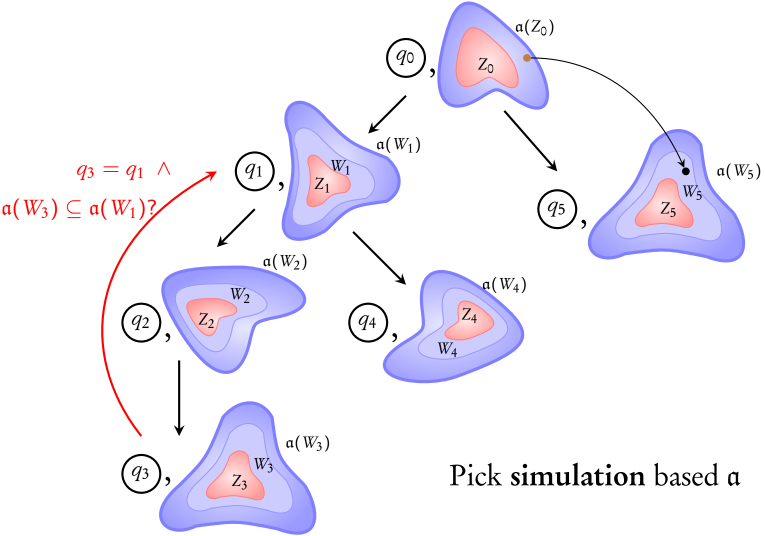
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



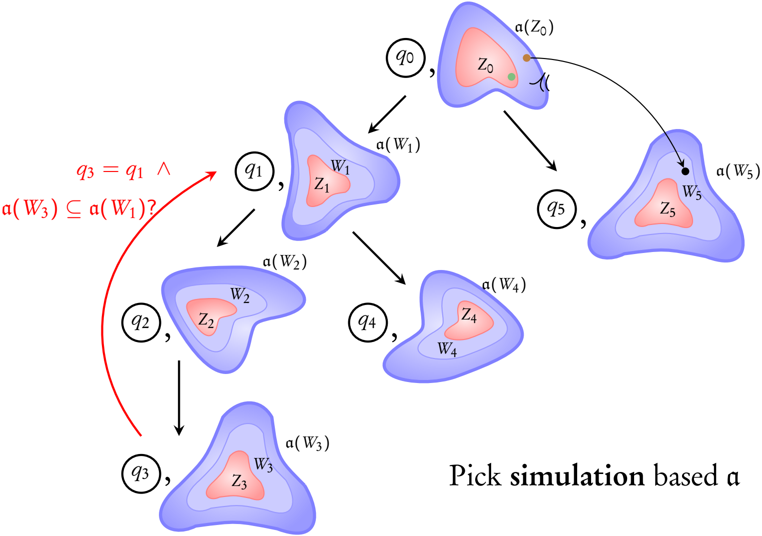
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



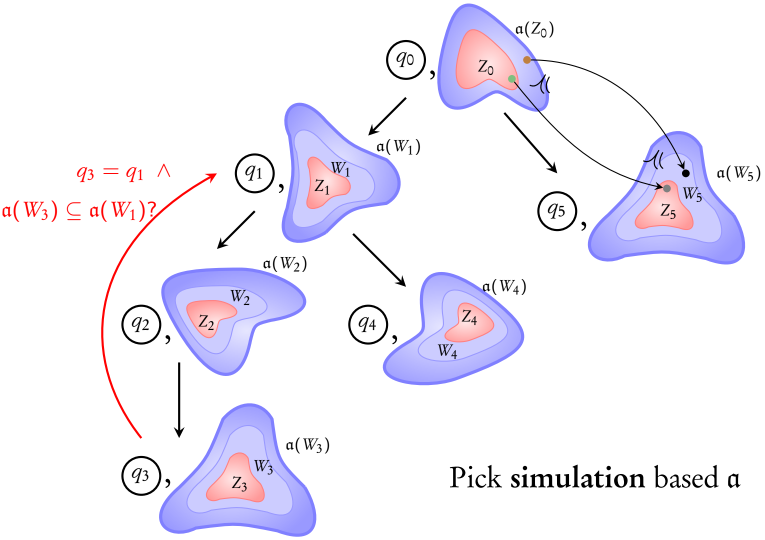
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



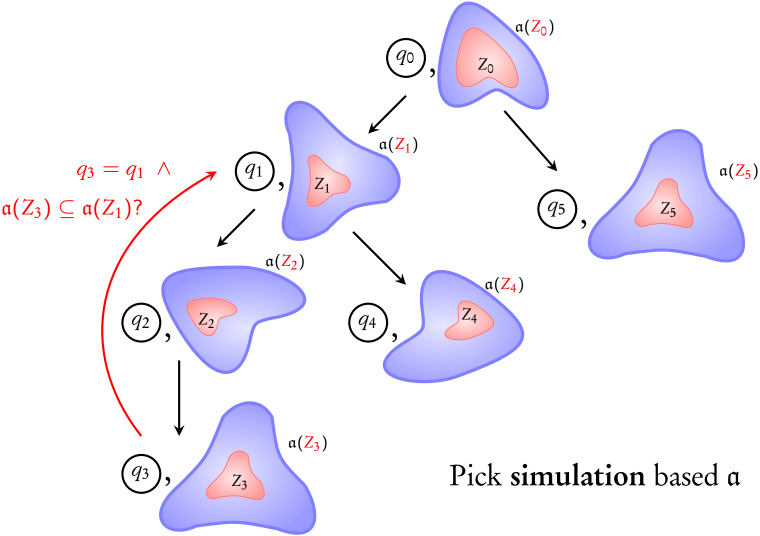
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



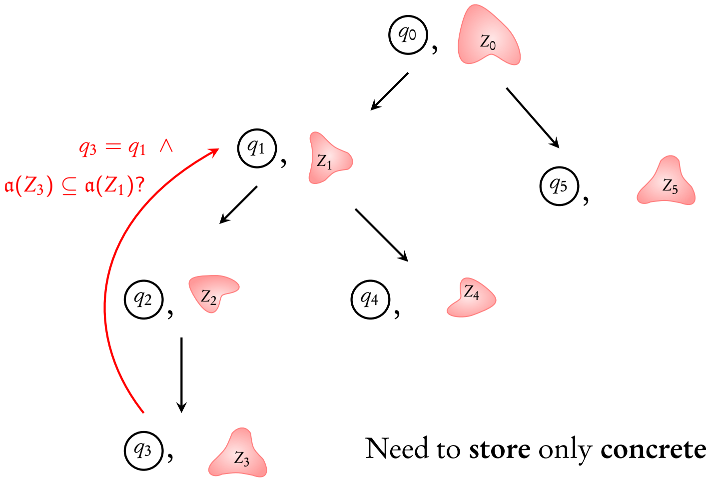
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



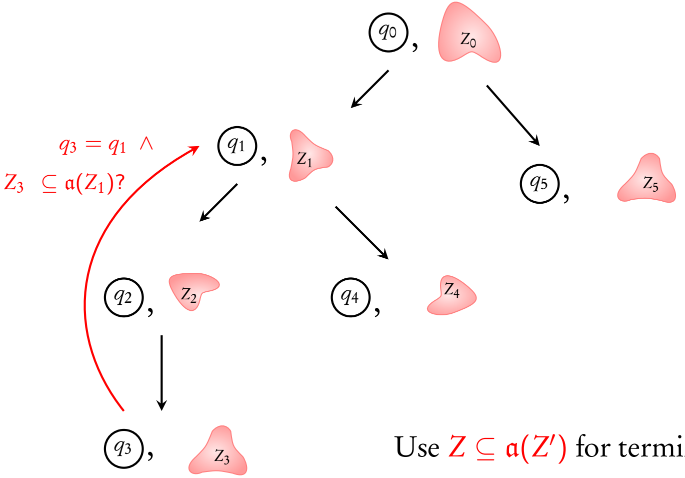
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



Need to **store** only **concrete** semantics

Using non-convex abstractions



Use $Z \subseteq \mathfrak{a}(Z')$ for termination

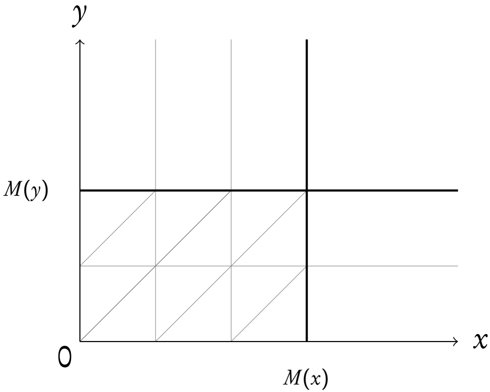
Observation 1: We can use abstractions **without storing** them

Observation 2: We can do the **inclusion** test **efficiently**

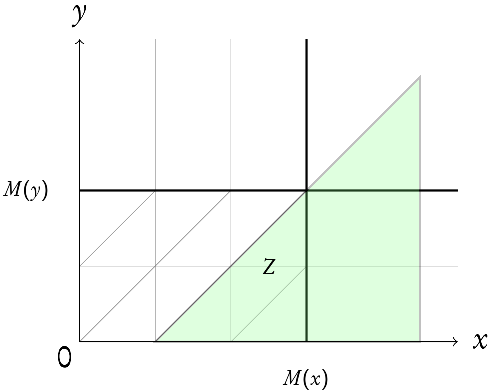
Coming next...

The inclusion test $Z \subseteq \text{Closure}_M(Z')$

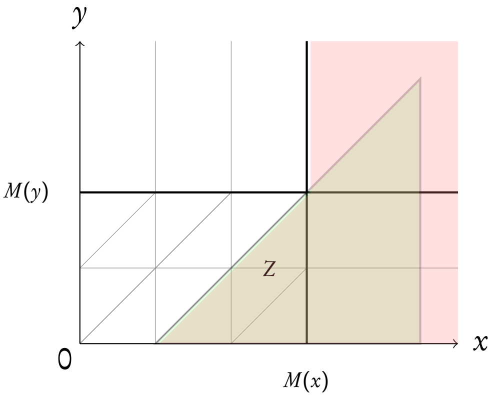
What is Closure$_M$?



What is Closure$_M$**?**

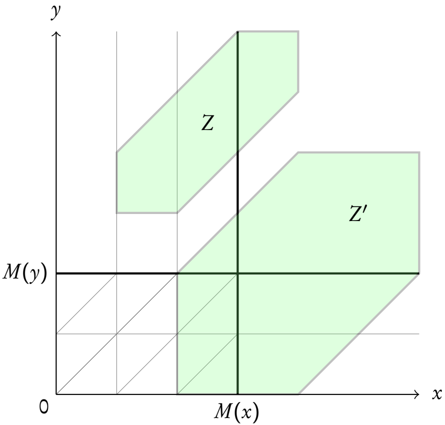


What is Closure$_M$?

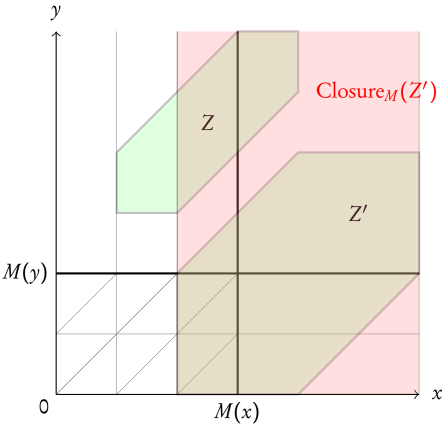


Closure$_M$(Z): set of regions that Z intersects

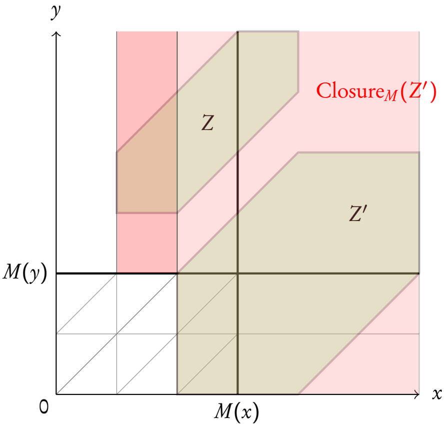
$Z \subseteq \text{Closure}_M(Z')$?



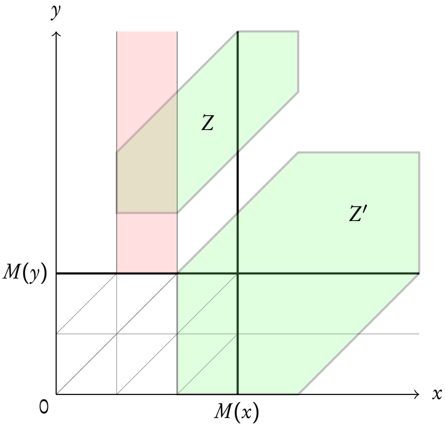
$Z \subseteq \mathrm{Closure}_M(Z')$?



$Z \subseteq \text{Closure}_M(Z')$?

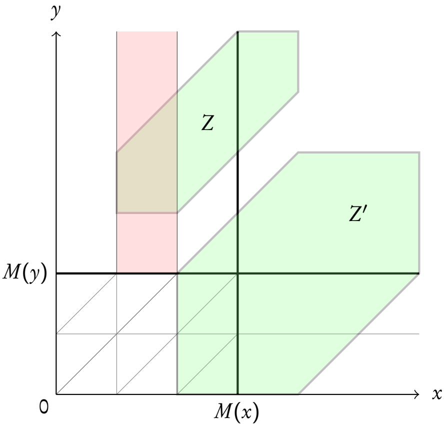


Z ⊆ $\text{Closure}_M(Z')$?



$Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.$ R **intersects Z**, R does **not intersect Z′**

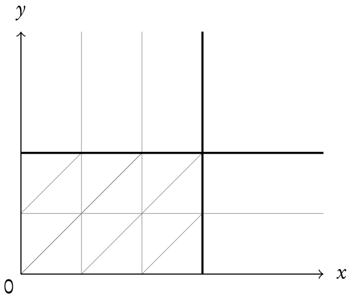
$Z \subseteq \text{Closure}_M(Z')$?



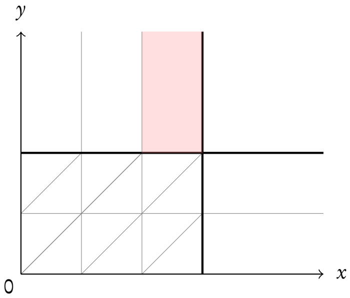
$Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ **intersects Z**, R does **not intersect Z′**

Coming next: Steps to the **efficient algorithm** for $Z \not\subseteq \text{Closure}_M(Z')$

Step 1: Representing regions and zones



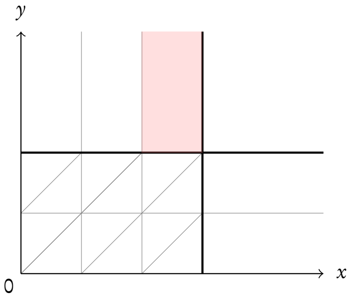
Step 1: Representing regions and zones



$x < 3$ $\quad y < \infty$

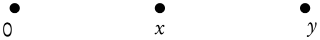
$x > 2$ $\quad y > 2$

Step 1: Representing regions and zones

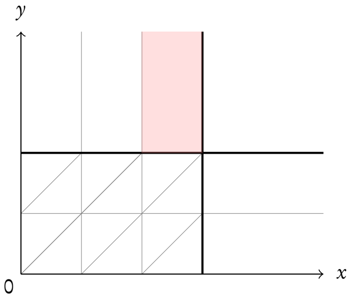


$x < 3$ $y < \infty$

$x > 2$ $y > 2$

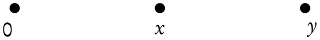


Step 1: Representing regions and zones

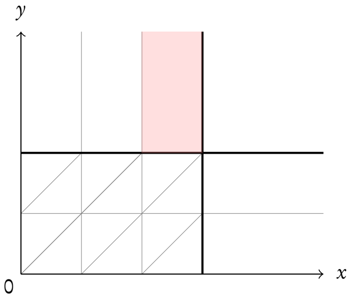


$x < 3$ $\quad$ $y < \infty$

$x > 2$ $\quad$ $y > 2$

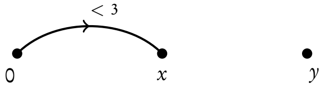


Step 1: Representing regions and zones

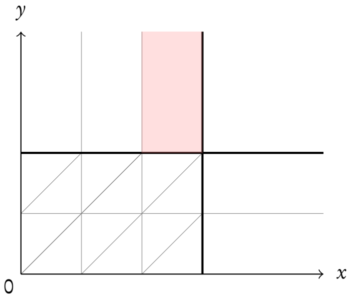


$x - 0 < 3$ $\quad$ $y < \infty$

$x > 2$ $\quad$ $y > 2$

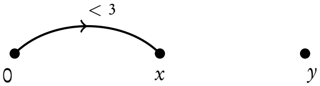


Step 1: Representing regions and zones

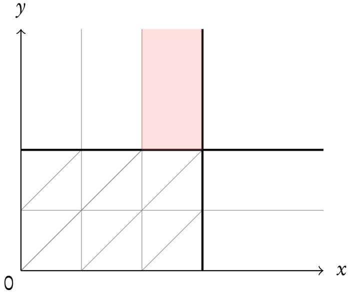


$x - 0 < 3$ $\quad$ $y < \infty$

$x > 2$ $\quad$ $y > 2$

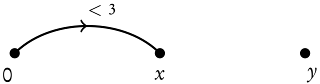


Step 1: Representing regions and zones

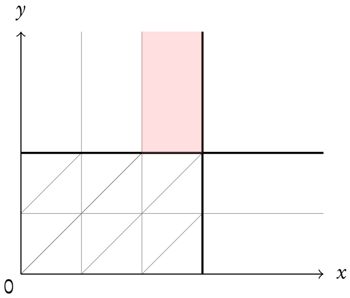


$$x - 0 < 3 \qquad y < \infty$$

$$0 - x < -2 \qquad y > 2$$

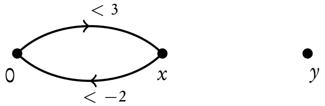


Step 1: Representing regions and zones

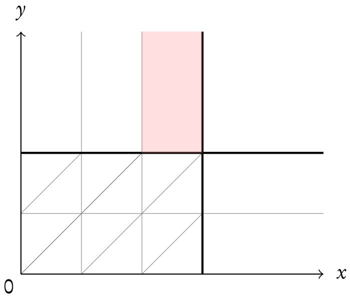


$x - 0 < 3$ $\quad y < \infty$

$0 - x < -2$ $\quad y > 2$

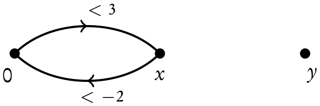


Step 1: Representing regions and zones

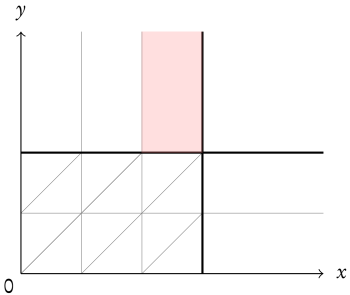


$x - 0 < 3$ $\quad$ $y < \infty$

$0 - x < -2$ $\quad$ $y > 2$

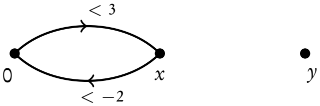


Step 1: Representing regions and zones

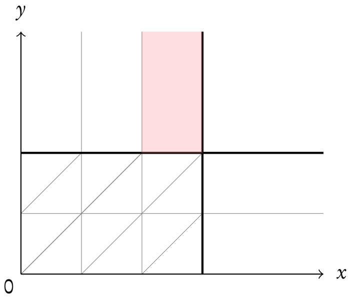


$x - 0 < 3$ $\quad$ $y - 0 < \infty$

$0 - x < -2$ $\quad$ $0 - y < -2$

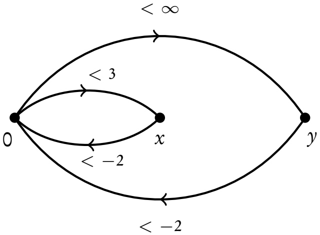


Step 1: Representing regions and zones

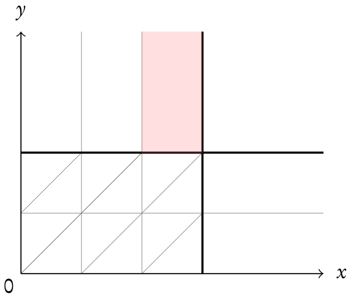


$x - 0 < 3$ $\quad$ $y - 0 < \infty$

$0 - x < -2$ $\quad$ $0 - y < -2$

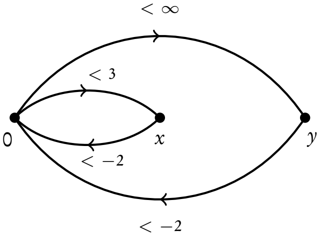


Step 1: Representing regions and zones

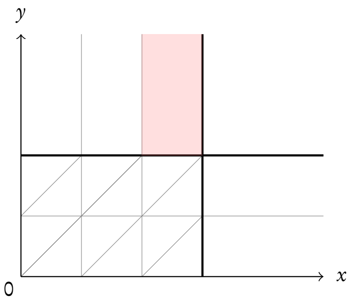


$$x - 0 < 3 \qquad y - 0 < \infty$$

$$0 - x < -2 \qquad 0 - y < -2$$

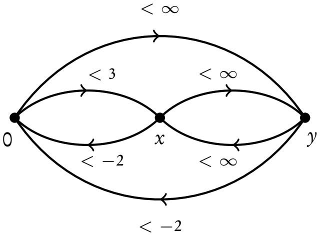


Step 1: Representing regions and zones

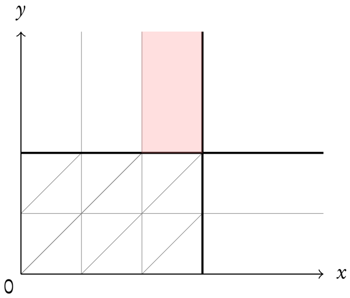


$x - 0 < 3 \qquad y - 0 < \infty$

$0 - x < -2 \qquad 0 - y < -2$

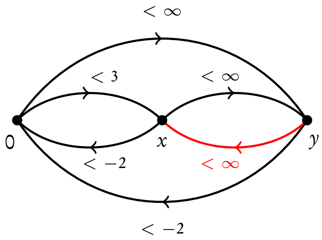


Step 1: Representing regions and zones



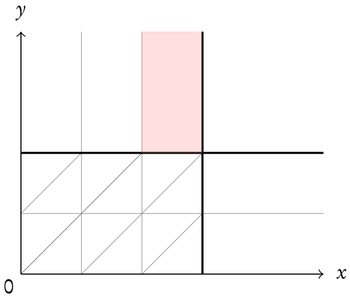
$x - 0 < 3$ $\quad y - 0 < \infty$

$0 - x < -2$ $\quad 0 - y < -2$



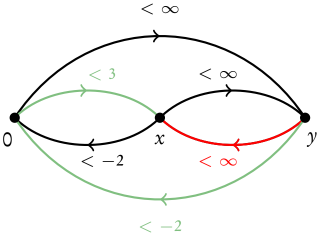
Need a **canonical** representation

Step 1: Representing regions and zones



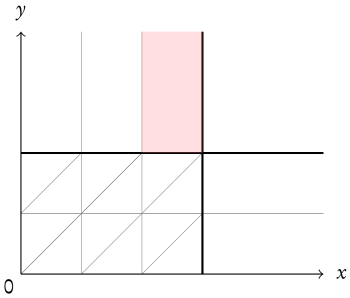
$x - 0 < 3$ $\quad$ $y - 0 < \infty$

$0 - x < -2$ $\quad$ $0 - y < -2$



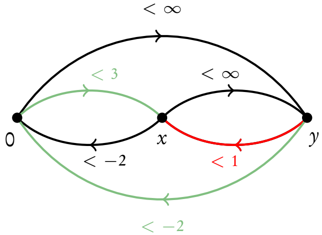
Shortest path should be given by the **direct edge**

Step 1: Representing regions and zones



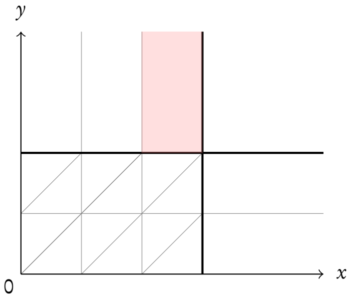
$x - 0 < 3$ $\quad$ $y - 0 < \infty$

$0 - x < -2$ $\quad$ $0 - y < -2$



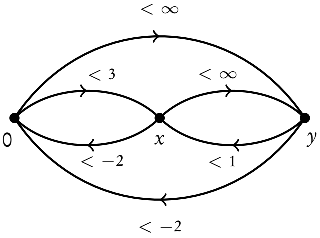
Shortest path should be given by the **direct edge**

Step 1: Representing regions and zones



$x - 0 < 3$ $\quad$ $y - 0 < \infty$

$0 - x < -2$ $\quad$ $0 - y < -2$



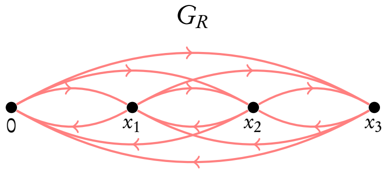
For every zone Z, **canonical distance graph** G_Z

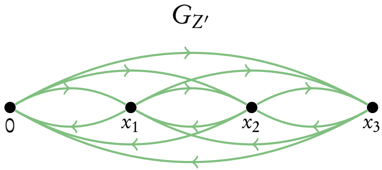
Step 2: When is $R \cap Z'$ empty?

Inspired by an observation made in [Bou04]

Step 2: When is $R \cap Z'$ empty?

Inspired by an observation made in [Bou04]

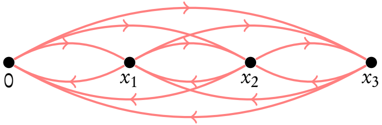




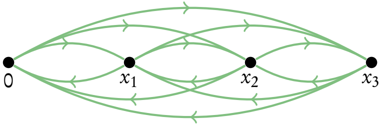
Step 2: When is $R \cap Z'$ empty?

Inspired by an observation made in [Bou04]

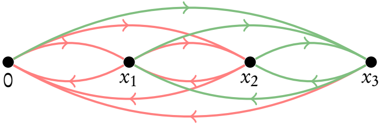
G_R



$G_{Z'}$



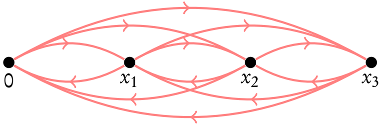
$\min(G_R, G_{Z'})$



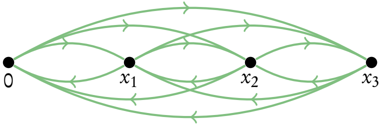
Step 2: When is $R \cap Z'$ empty?

Inspired by an observation made in [Bou04]

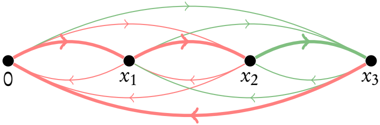
G_R



$G_{Z'}$



$\min(G_R, G_{Z'})$



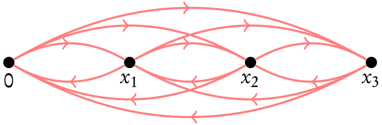
Lemma

$R \cap Z'$ is **empty** $\quad \Leftrightarrow \quad$ $\min(G_R, G_{Z'})$ has a **negative cycle**

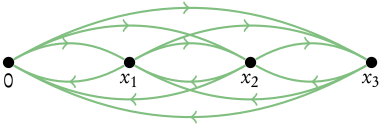
Step 2: When is $R \cap Z'$ empty?

Inspired by an observation made in [Bou04]

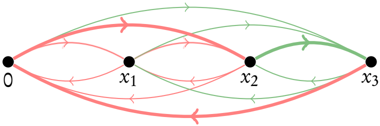
G_R



$G_{Z'}$



$\min(G_R, G_{Z'})$



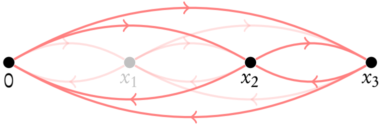
Lemma

$R \cap Z'$ is **empty** $\Leftrightarrow$ $\min(G_R, G_{Z'})$ has a **negative cycle** involving **at most 2 clocks!**

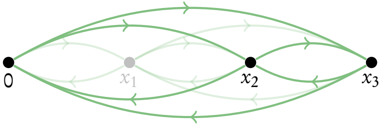
Step 2: When is $R \cap Z'$ empty?

Inspired by an observation made in [Bou04]

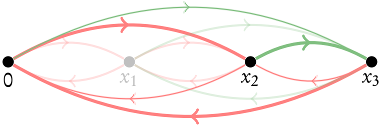
G_R



$G_{Z'}$



$\min(G_R, G_{Z'})$



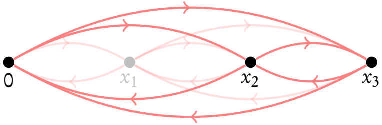
Lemma

$R \cap Z'$ is **empty** $\Leftrightarrow$ $\min(G_R, G_{Z'})$ has a **negative cycle** involving **at most 2 clocks!**

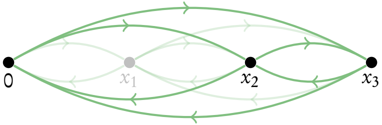
Step 2: When is $R \cap Z'$ empty?

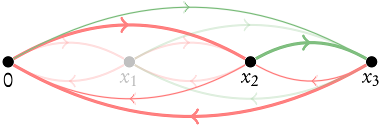
Inspired by an observation made in [Bou04]

$G_{\mathsf{Proj}_{x_2 x_3}(R)}$



$G_{\mathsf{Proj}_{x_2 x_3}(Z')}$





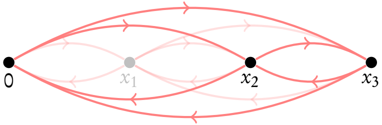
Lemma

$R \cap Z'$ is **empty** $\Leftrightarrow$ $\min(G_R, G_{Z'})$ has a **negative cycle** involving **at most 2 clocks!**

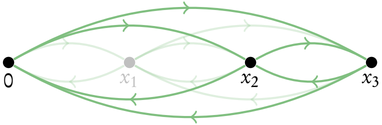
Step 2: When is $R \cap Z'$ empty?

Inspired by an observation made in [Bou04]

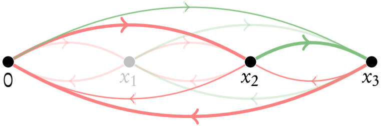
$G_{\mathsf{Proj}_{x_2x_3}(R)}$



$G_{\mathsf{Proj}_{x_2x_3}(Z')}$



$\min(G_{\mathsf{Proj}_{x_2x_3}(R)}, G_{\mathsf{Proj}_{x_2x_3}(Z')})$



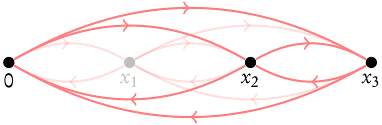
Lemma

$R \cap Z'$ is **empty** $\Leftrightarrow$ $\min(G_R, G_{Z'})$ has a **negative cycle** involving **at most 2 clocks!**

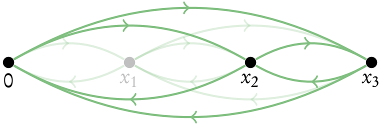
Step 2: When is $R \cap Z'$ empty?

Inspired by an observation made in [Bou04]

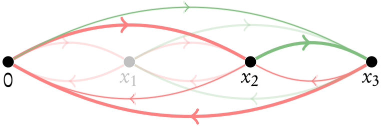
$G_{\mathsf{Proj}_{x_2 x_3}(R)}$



$G_{\mathsf{Proj}_{x_2 x_3}(Z')}$



$\min(G_{\mathsf{Proj}_{x_2 x_3}(R)}, G_{\mathsf{Proj}_{x_2 x_3}(Z')})$



Lemma

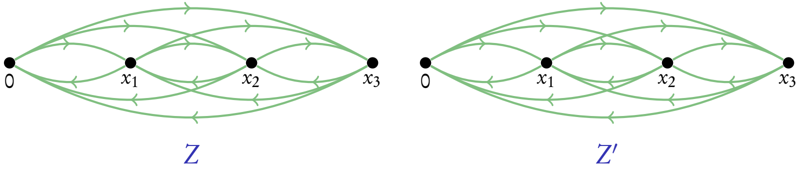
$R \cap Z'$ is **empty** $\Leftrightarrow$ $\exists\, x, y.\ \mathsf{Proj}_{\mathbf{xy}}(R) \cap \mathsf{Proj}_{\mathbf{xy}}(Z')$ is **empty**

Step 3: Reduction to two clocks

Recall: $Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ intersects Z, R does not intersect Z'

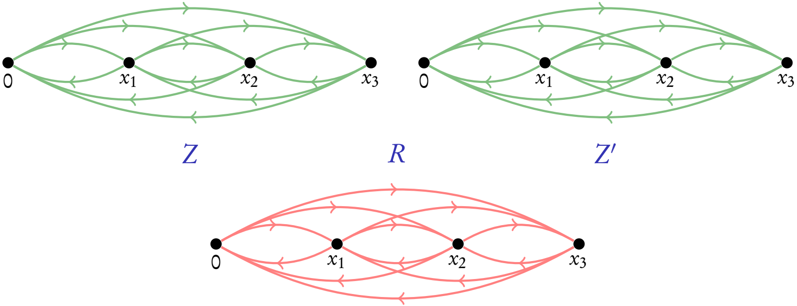
Step 3: Reduction to two clocks

Recall: $Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ intersects Z, R does not intersect Z'



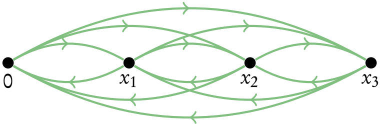
Step 3: Reduction to two clocks

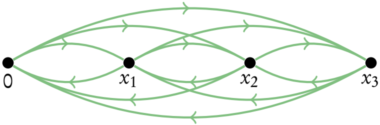
Recall: $Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ intersects Z, R does not intersect Z'



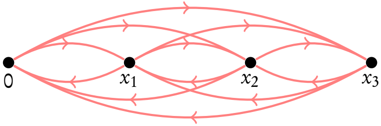
Step 3: Reduction to two clocks

Recall: $Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ intersects Z, R does not intersect Z'



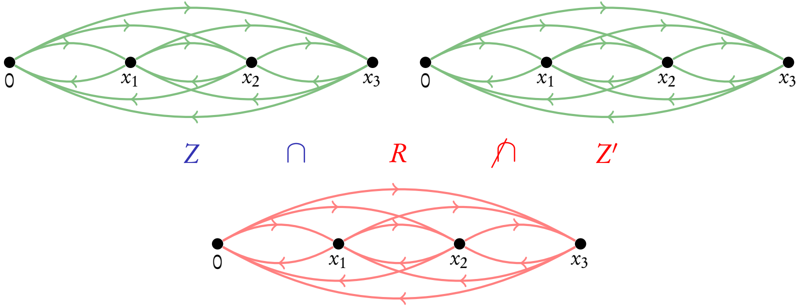


$$Z \quad \cap \quad R \quad \not\cap \quad Z'$$



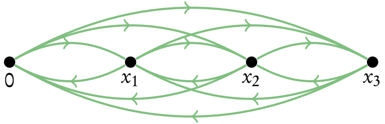
Step 3: Reduction to two clocks

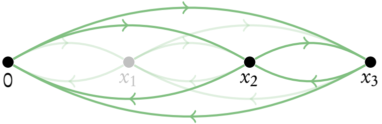
Recall: $Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ intersects Z, R does not intersect Z'



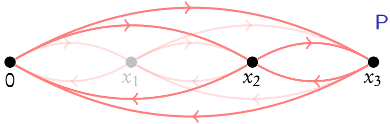
Step 3: Reduction to two clocks

Recall: $Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ intersects Z, R does not intersect Z'





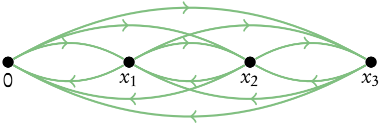
$Z \quad \cap \quad R \quad \not\cap \quad Z'$

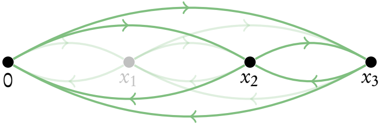


$\text{Proj}_{x_2 x_3}(R) \not\cap \text{Proj}_{x_2 x_3}(Z')$

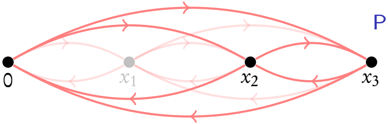
Step 3: Reduction to two clocks

Recall: $Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ intersects Z, R does not intersect Z'





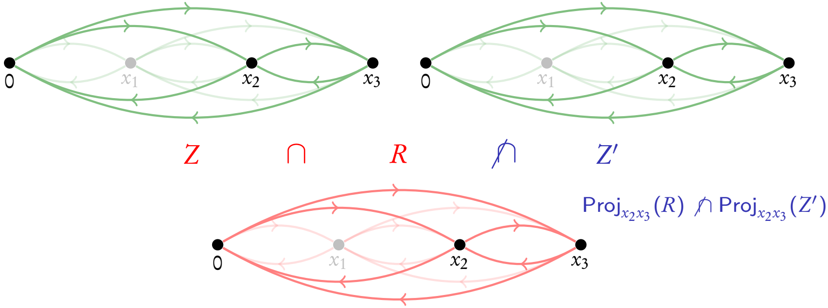
$Z \quad \cap \quad R \quad \not\cap \quad Z'$



$\text{Proj}_{x_2x_3}(R) \not\cap \text{Proj}_{x_2x_3}(Z')$

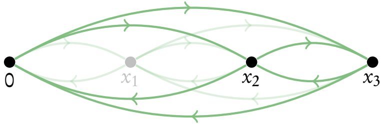
Step 3: Reduction to two clocks

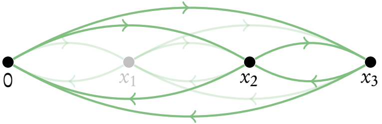
Recall: $Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ intersects Z, R does not intersect Z'



Step 3: Reduction to two clocks

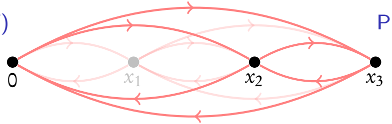
Recall: $Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ intersects Z, R does not intersect Z'





$Z \quad \cap \quad R \quad \not\cap \quad Z'$

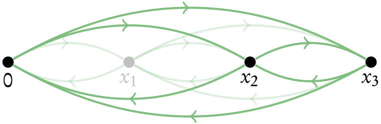
$\text{Proj}_{x_2x_3}(R) \cap \text{Proj}_{x_2x_3}(Z)$

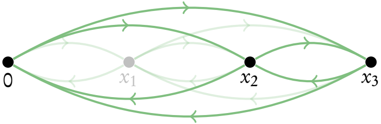


$\text{Proj}_{x_2x_3}(R) \not\cap \text{Proj}_{x_2x_3}(Z')$

Step 3: Reduction to two clocks

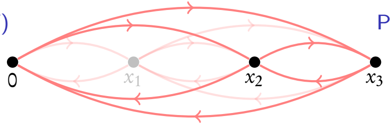
Recall: $Z \not\subseteq \text{Closure}_M(Z') \Leftrightarrow \exists R.\ R$ intersects Z, R does not intersect Z'





$Z \quad \cap \quad R \quad \not\cap \quad Z'$

$\text{Proj}_{x_2x_3}(R) \cap \text{Proj}_{x_2x_3}(Z)$



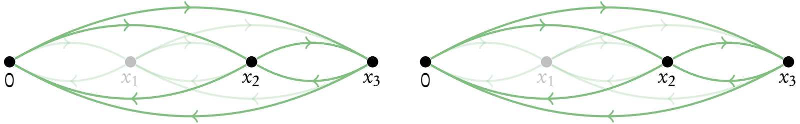
$\text{Proj}_{x_2x_3}(R) \not\cap \text{Proj}_{x_2x_3}(Z')$

Theorem

$Z \not\subseteq \text{Closure}_\alpha(Z')$ if and only if there **exist 2 clocks** x, y s.t.

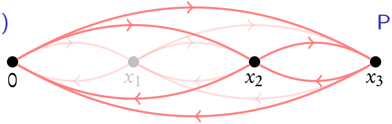
$$\mathbf{Proj_{xy}}(Z) \not\subseteq \text{Closure}_M(\mathbf{Proj}_{xy}(Z'))$$

Step 3: Reduction to two clocks



$Z \quad \cap \quad R \quad \not\cap \quad Z'$

$\mathsf{Proj}_{x_2x_3}(R) \cap \mathsf{Proj}_{x_2x_3}(Z)$



$\mathsf{Proj}_{x_2x_3}(R) \not\cap \mathsf{Proj}_{x_2x_3}(Z')$

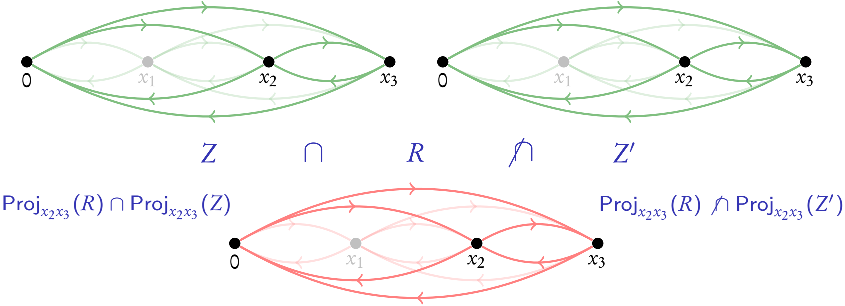
Theorem

$Z \not\subseteq \text{Closure}_\alpha(Z')$ if and only if there **exist 2 clocks** x, y s.t.

$$\mathbf{Proj_{xy}}(Z) \not\subseteq \text{Closure}_M(\mathbf{Proj}_{xy}(Z'))$$

Slightly **modified edge-edge comparison** is enough

Step 3: Reduction to two clocks



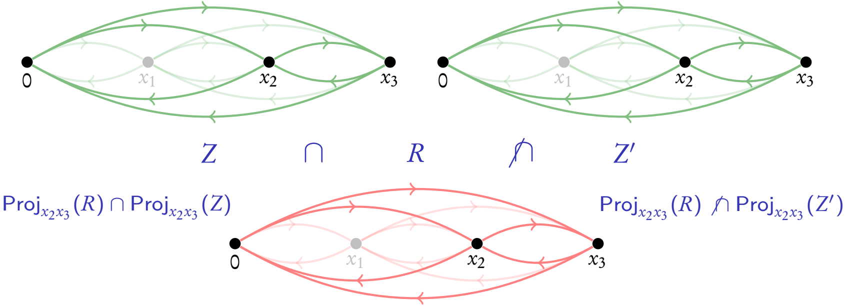
Theorem

$Z \not\subseteq \mathrm{Closure}_\alpha(Z')$ if and only if there **exist 2 clocks** x, y s.t.

$$\mathbf{Proj_{xy}}(Z) \not\subseteq \mathrm{Closure}_M(\mathbf{Proj}_{xy}(Z'))$$

Complexity: $\mathcal{O}(|X|^2)$, where X is the set of clocks

Step 3: Reduction to two clocks



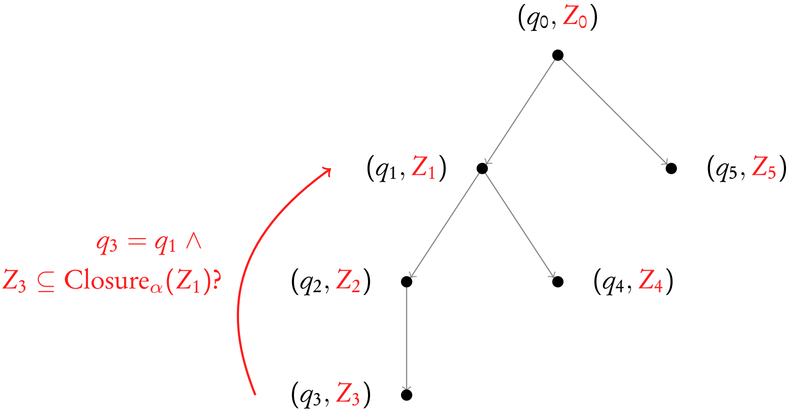
Theorem

$Z \not\subseteq \text{Closure}_\alpha(Z')$ if and only if there **exist 2 clocks** x, y s.t.

$$\mathbf{Proj_{xy}}(Z) \not\subseteq \text{Closure}_M(\mathbf{Proj}_{xy}(Z'))$$

Same complexity as $Z \subseteq Z'$!

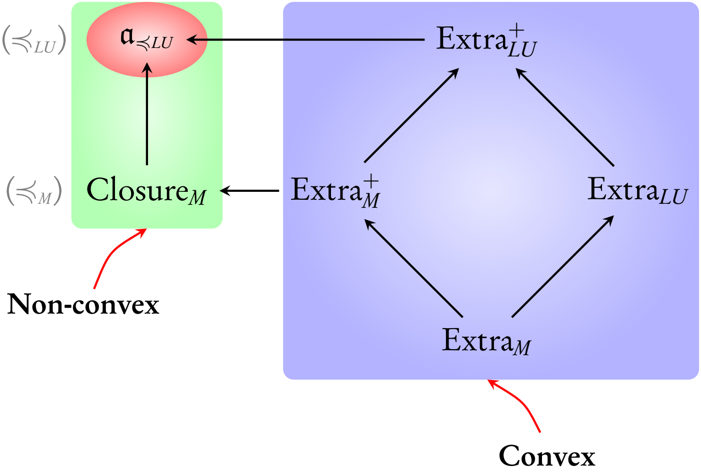
So what do we have now...



Efficient algorithm for $Z \subseteq \text{Closure}_\alpha(Z')$

Overall algorithm

- **Store** concrete semantics : zones
- Compute $ZG(\mathcal{A})$: $Z \subseteq \text{Closure}_{\alpha'}(Z')$ for **termination**



Next lecture: $\mathfrak{a}_{\preccurlyeq LU}$, optimality and benchmarks

References I

R. Alur and D.L. Dill.
A theory of timed automata.
Theoretical Computer Science, 126(2):183–235, 1994.

G. Behrmann, P. Bouyer, E. Fleury, and K. G. Larsen.
Static guard analysis in timed automata verification.
In *TACAS'03*, volume 2619 of *LNCS*, pages 254–270. Springer, 2003.

G. Behrmann, P. Bouyer, K. Larsen, and R. Pelánek.
Lower and upper bounds in zone based abstractions of timed automata.
Tools and Algorithms for the Construction and Analysis of Systems, pages 312–326, 2004.

P. Bouyer.
Forward analysis of updatable timed automata.
Form. Methods in Syst. Des., 24(3):281–320, 2004.

D. Dill.
Timing assumptions and verification of finite-state concurrent systems.
In *AVMFSS*, volume 407 of *LNCS*, pages 197–212. Springer, 1989.

C. Daws and S. Tripakis.
Model checking of real-time reachability properties using abstractions.
In *TACAS'98*, volume 1384 of *LNCS*, pages 313–329. Springer, 1998.

François Laroussinie and Ph. Schnoebelen.
The state explosion problem from trace to bisimulation equivalence.
In *Proceedings of the Third International Conference on Foundations of Software Science and Computation Structures*, FOSSACS '00, pages 192–207. Springer-Verlag, 2000.