Abstractions for timed automata

B. Srivathsan

Chennai Mathematical Institute

Joint work with

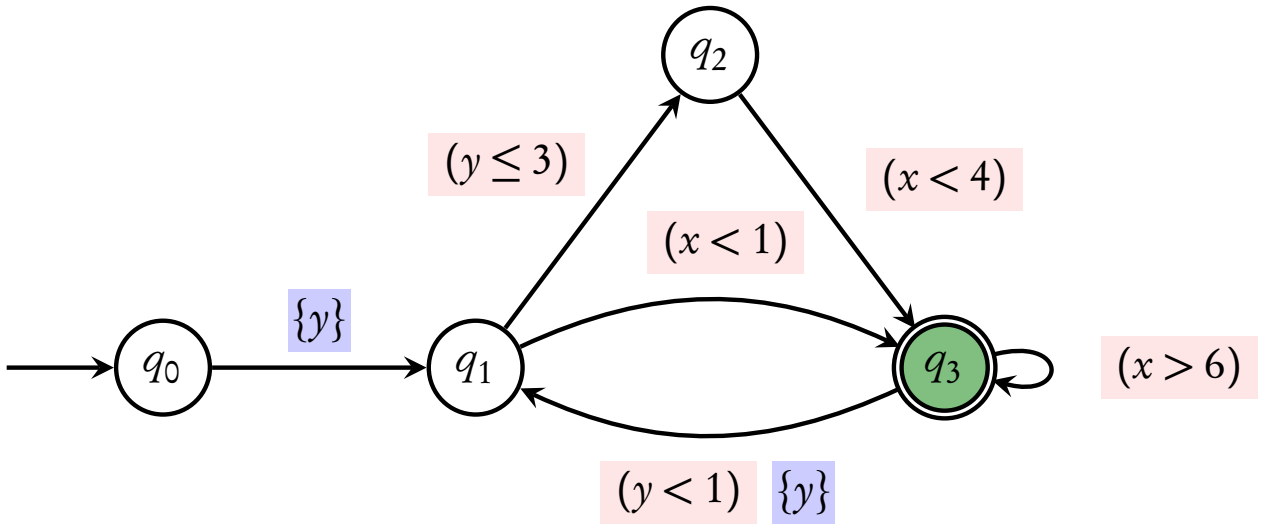
F. Herbreteau, I. Walukiewicz *(LaBRI, Bordeaux)*

Reachability for timed automata

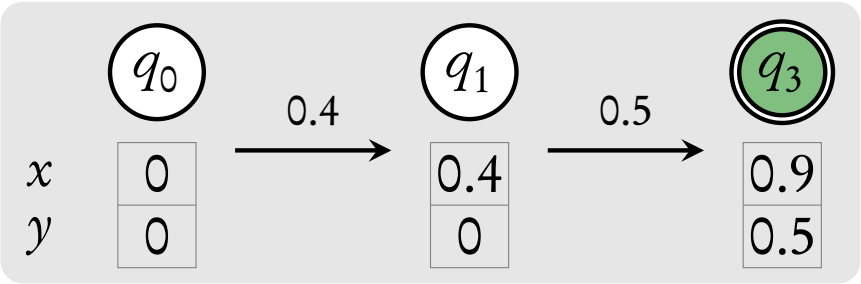
Observation 1

Observation 2

Timed Automata



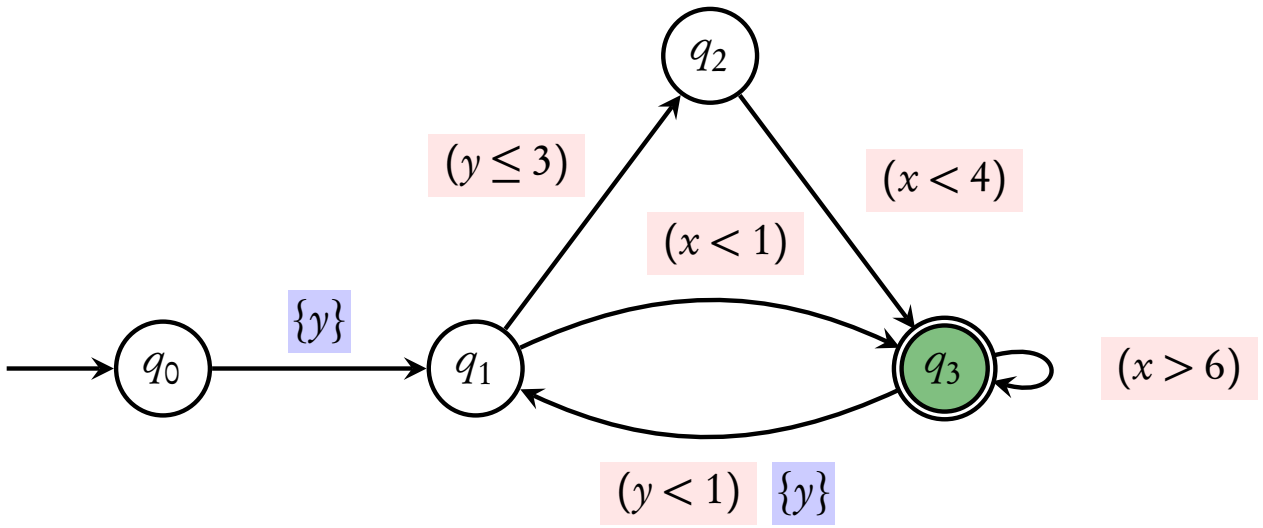
Run: finite sequence of transitions



- **accepting** if ends in green state

Reachability problem

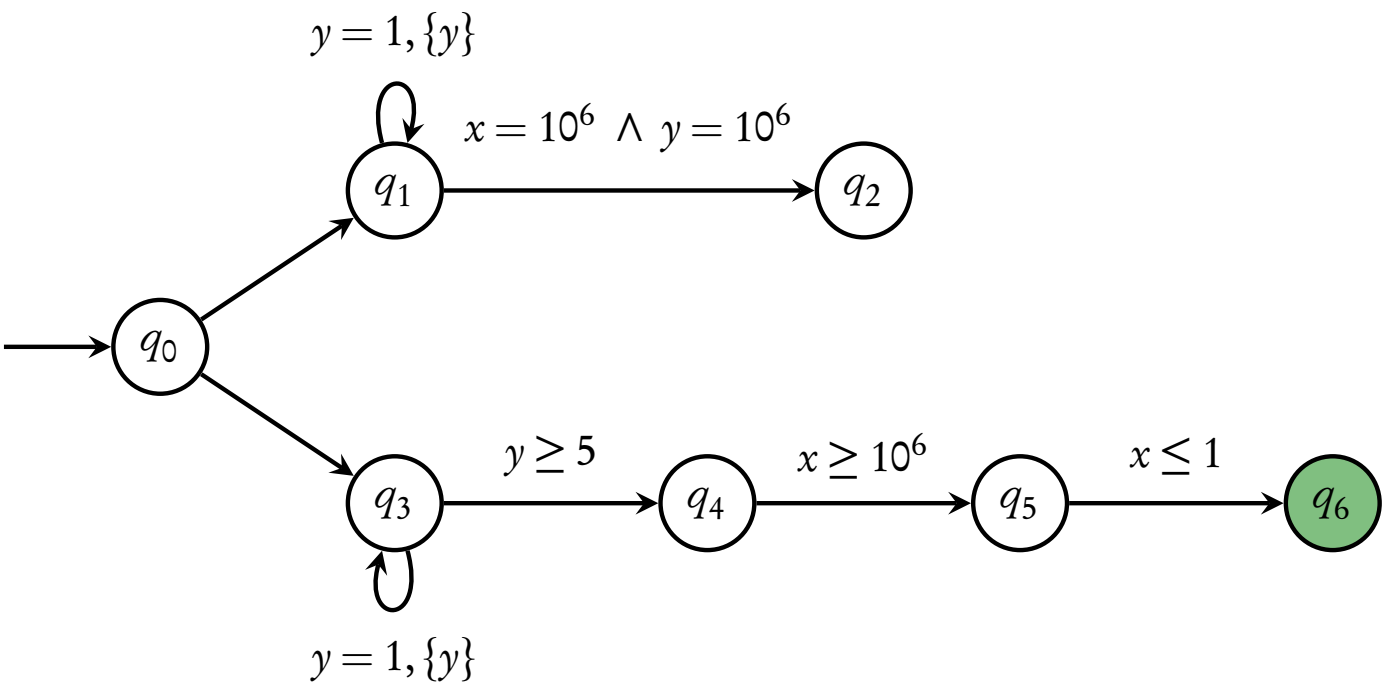
Given a TA, does it **have** an **accepting** run



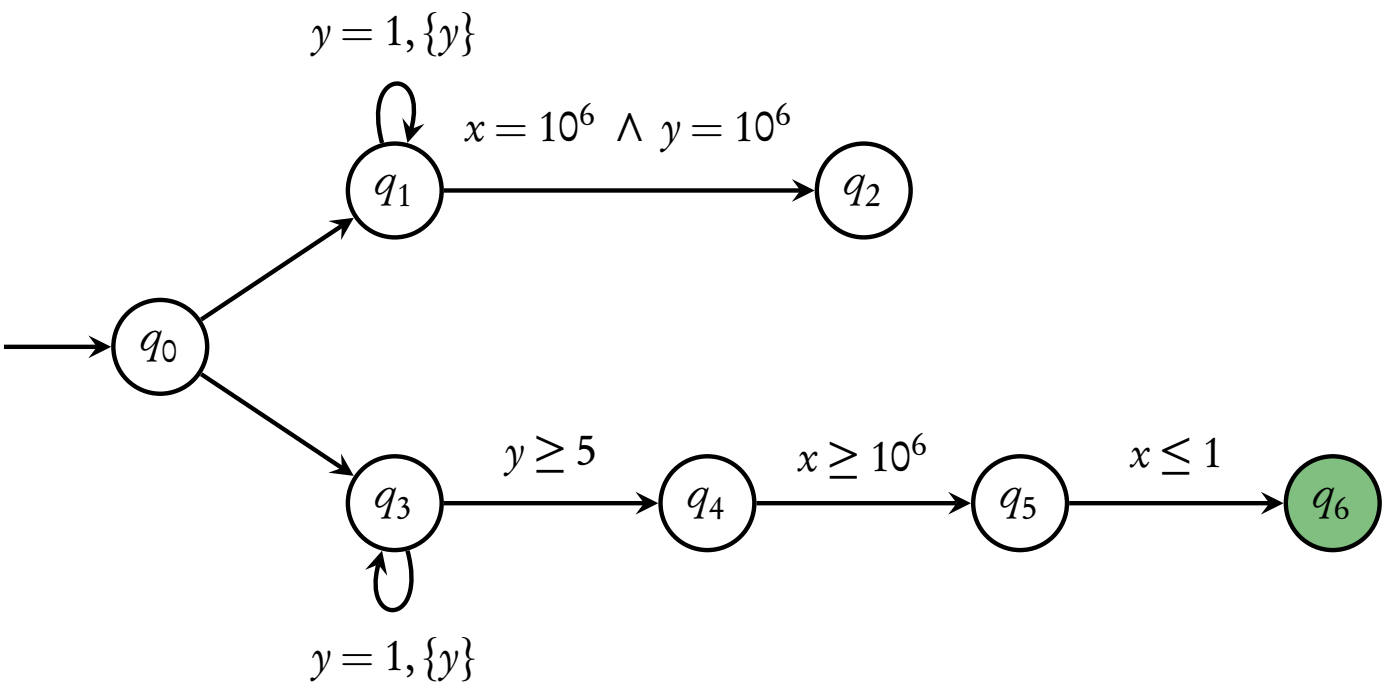
Theorem [Alur, Dill'90]

This problem is **PSPACE-complete**

Is accepting state **reachable**?



Is accepting state **reachable**? **No**



Is accepting state **reachable**? **No**

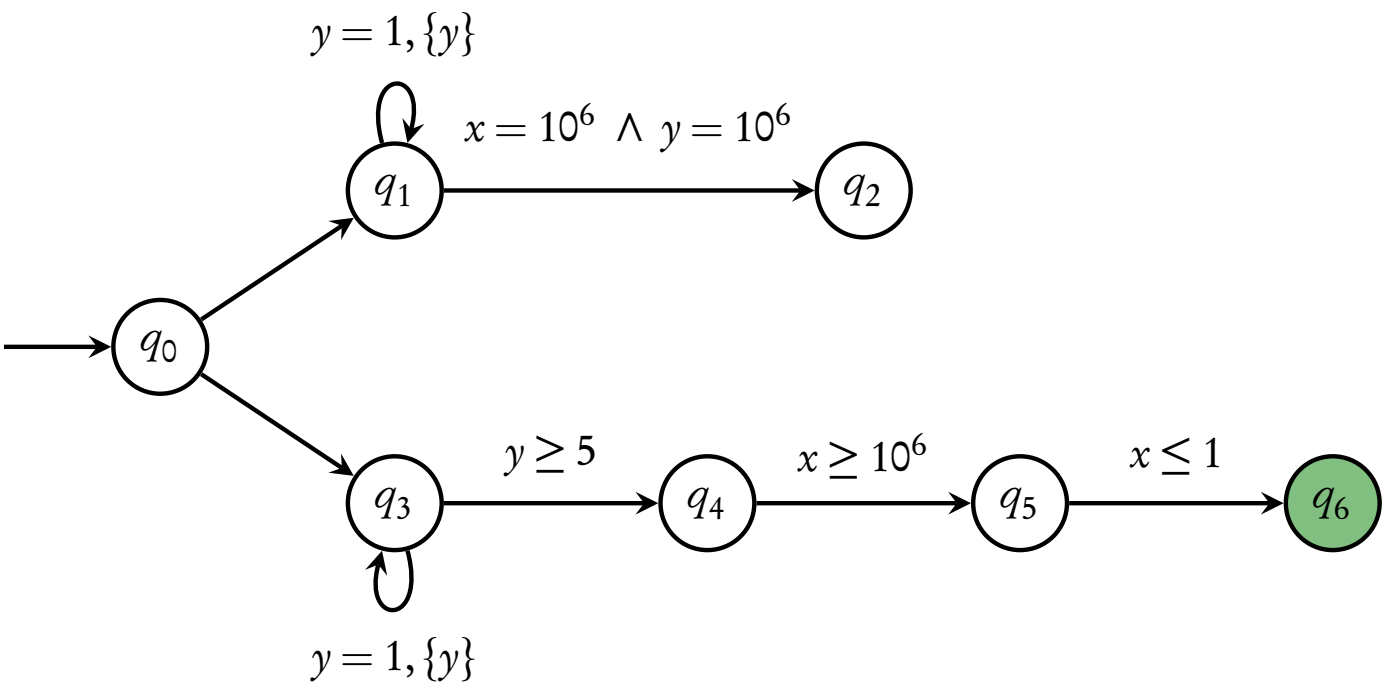
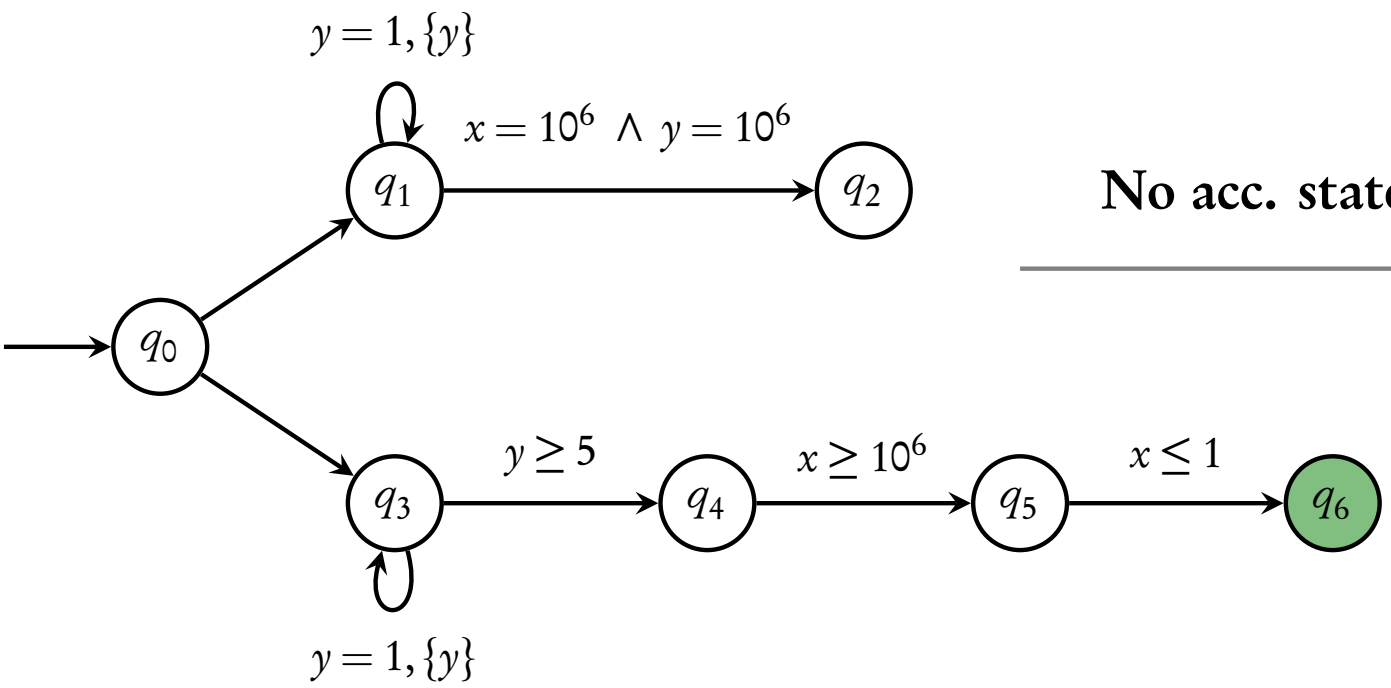


Exhibit a **proof** for unreachability

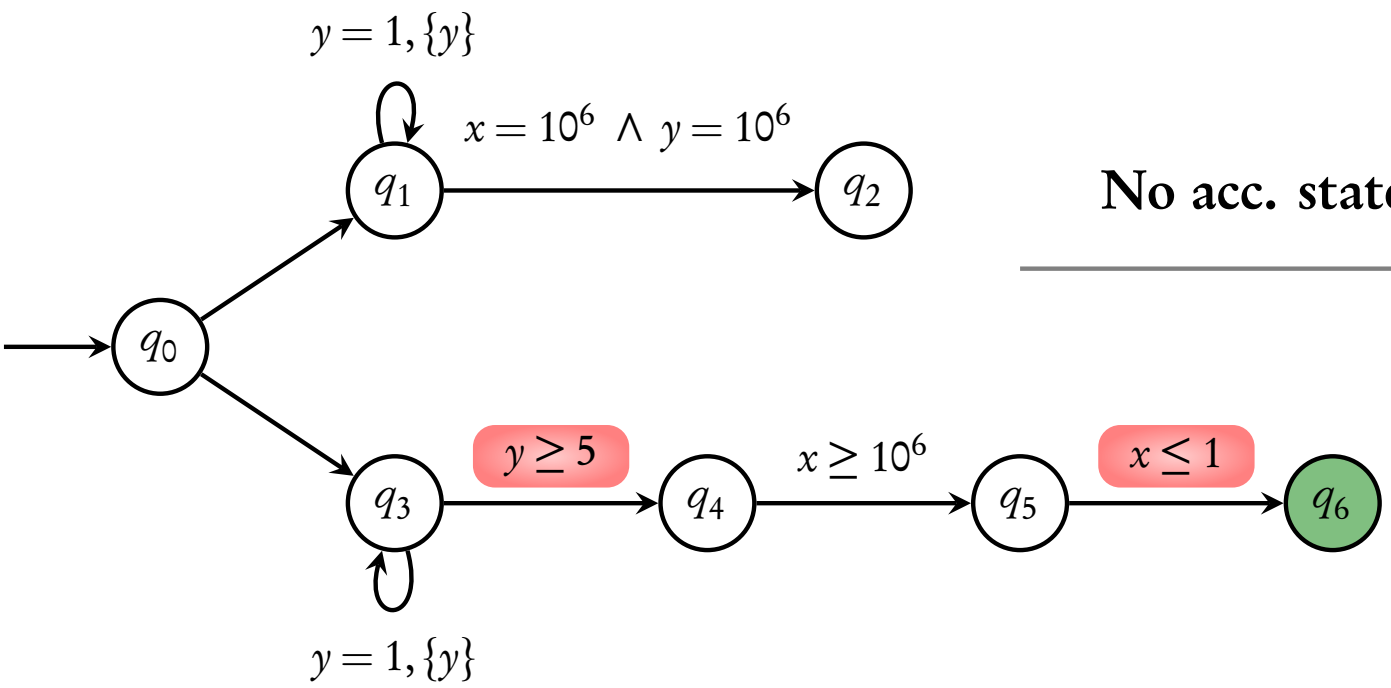
Is accepting state **reachable**? **No**



No acc. state

Exhibit a **proof** for unreachability

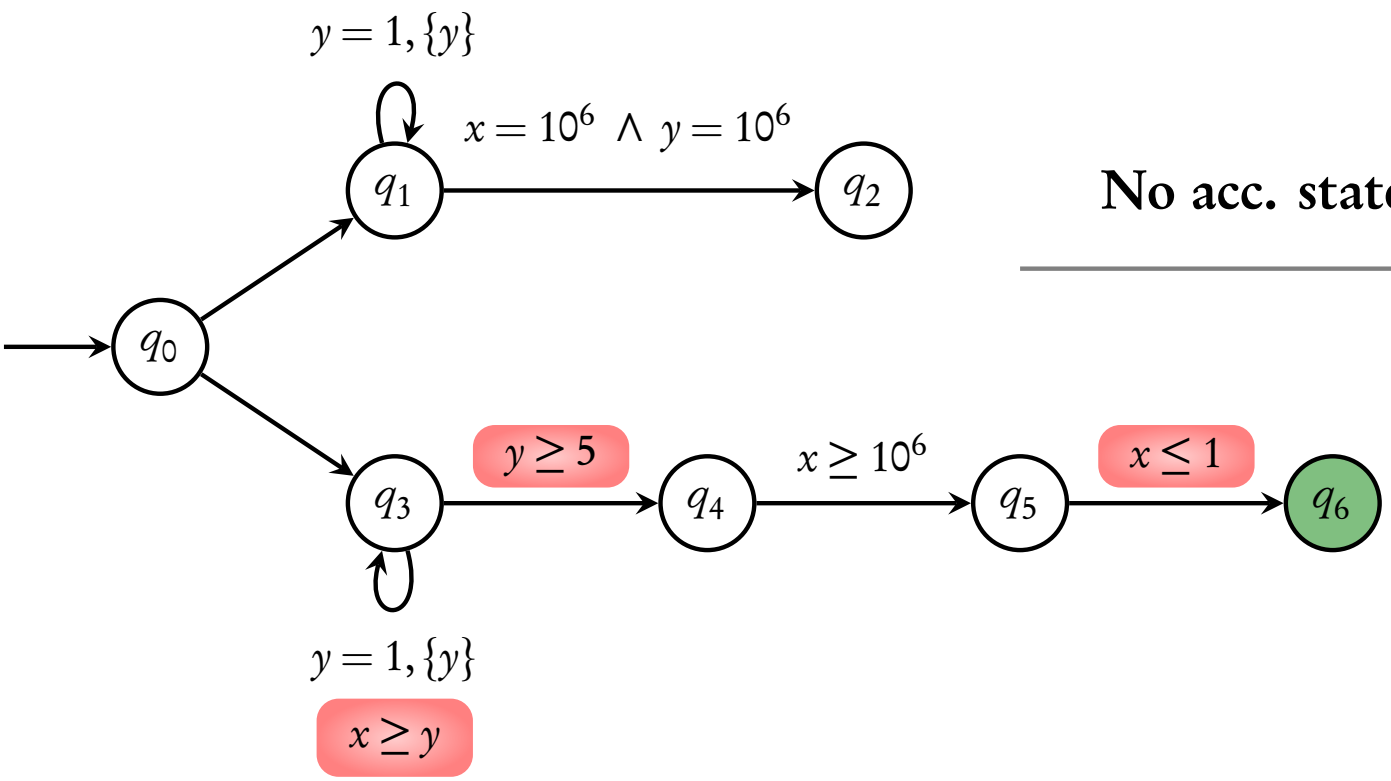
Is accepting state **reachable**? **No**



No acc. state

Exhibit a **proof** for unreachability

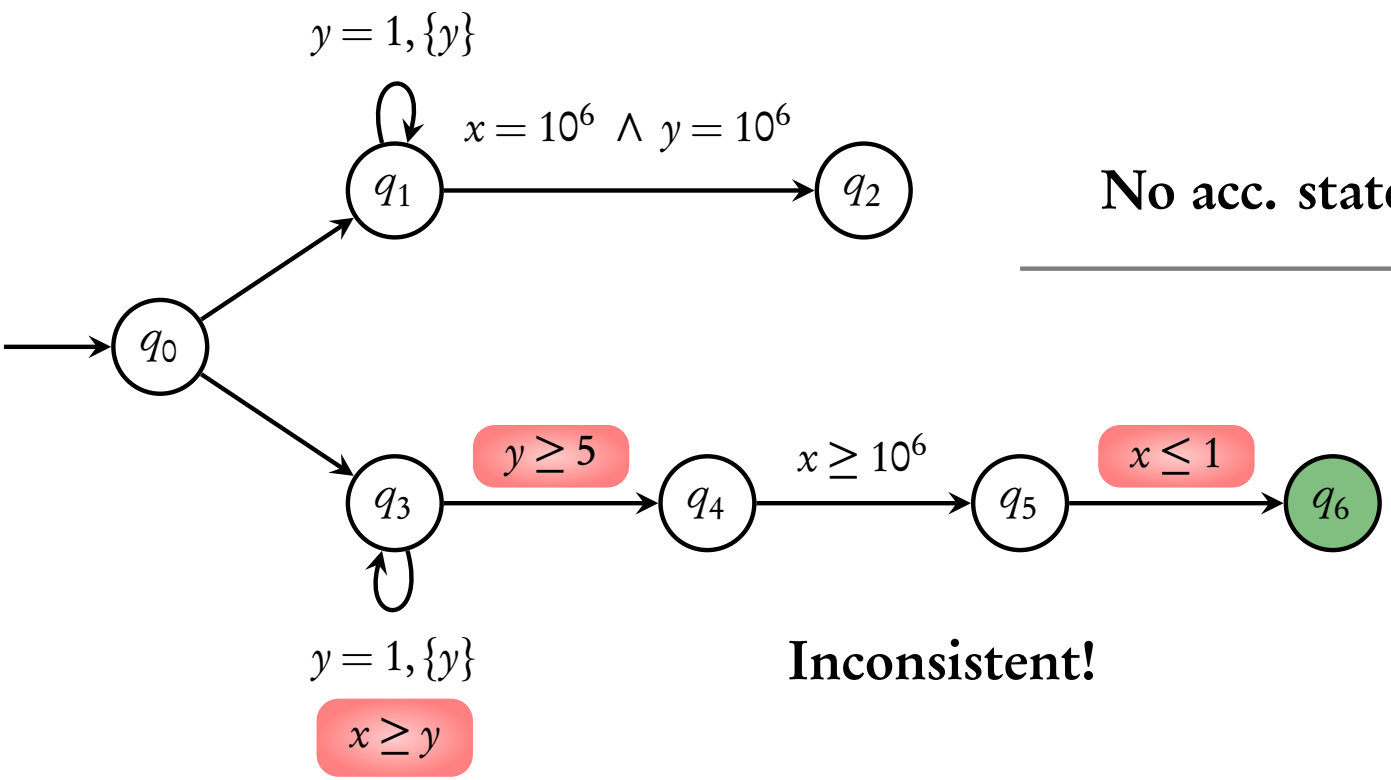
Is accepting state **reachable**? **No**



No acc. state

Exhibit a **proof** for unreachability

Is accepting state **reachable**? **No**



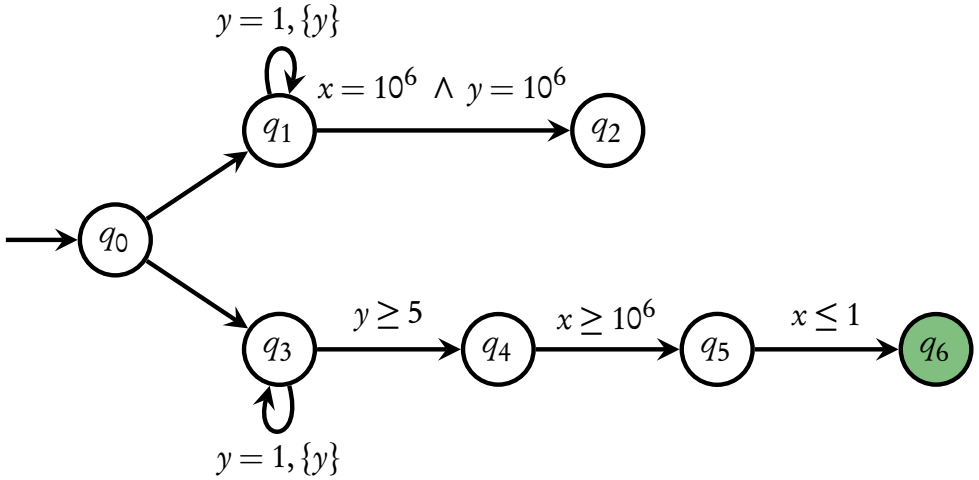
No acc. state

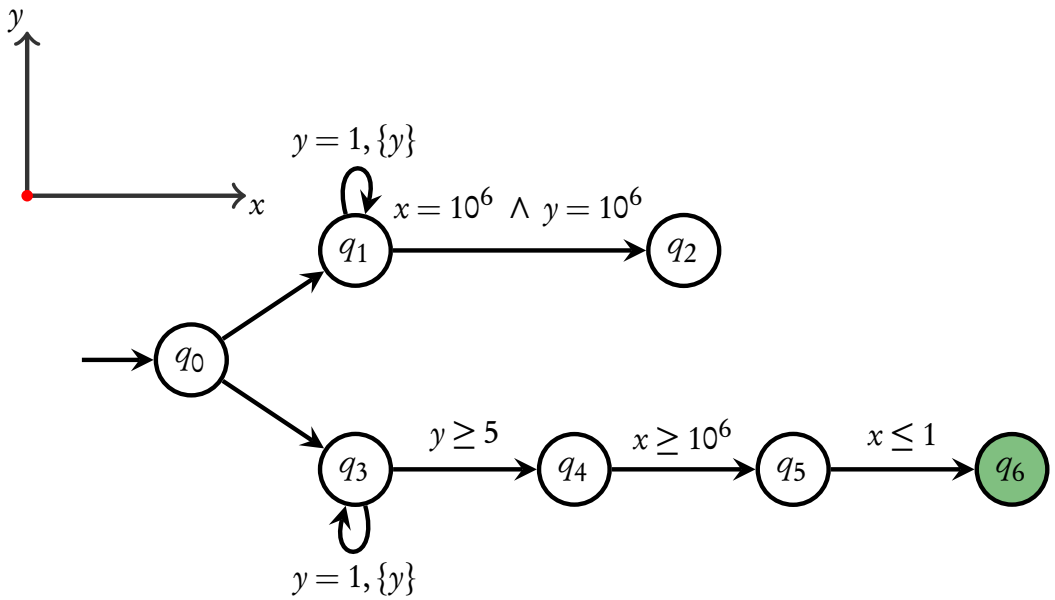
Inconsistent!

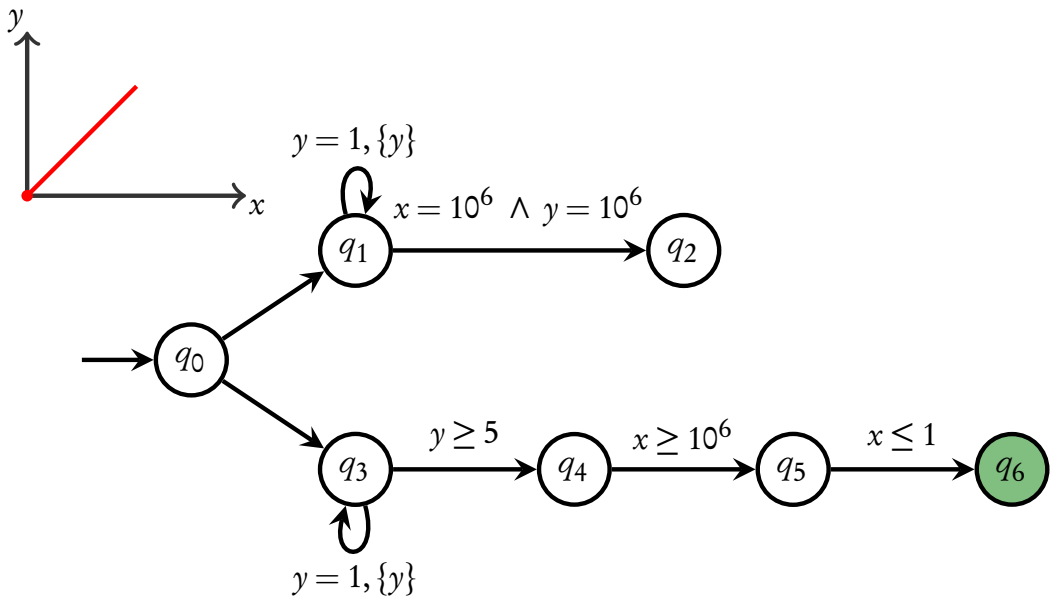
Exhibit a **proof** for unreachability

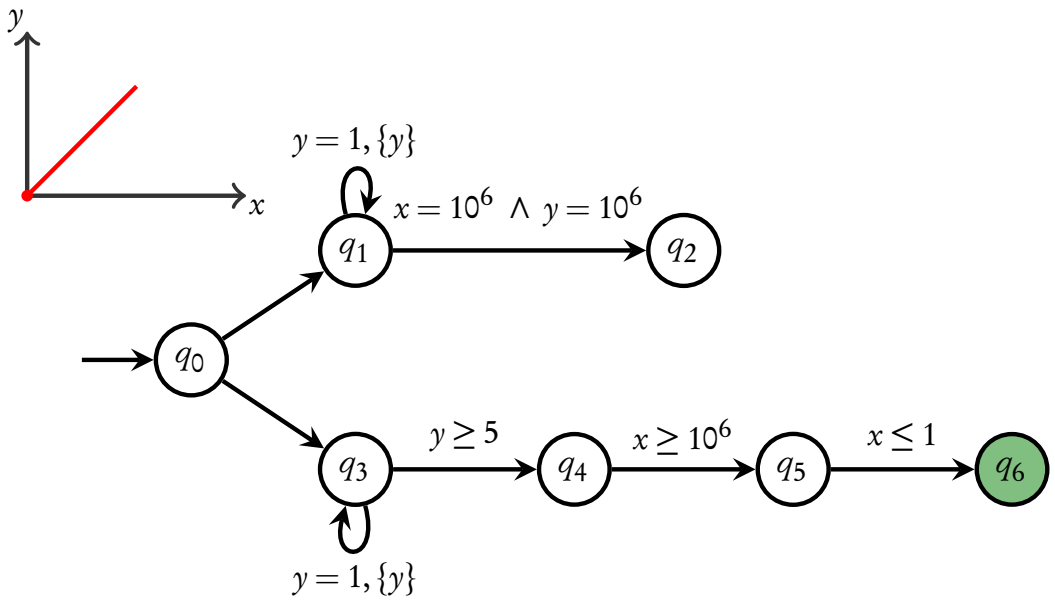
Goal of this talk

Computing **short proofs** for (un)-reachability

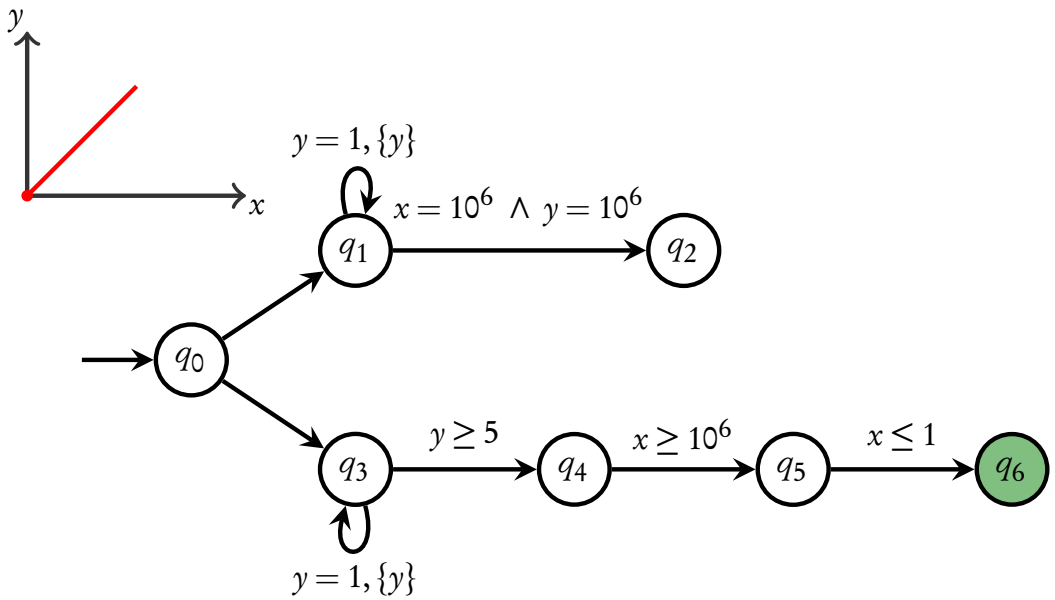

$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

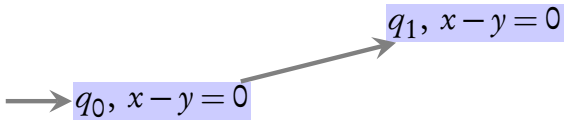

y
x
y = 1, {y}
x = 10^6 ∧ y = 10^6
q1
q2
q0
q3
y ≥ 5
q4
x ≥ 10^6
q5
x ≤ 1
q6
y = 1, {y}

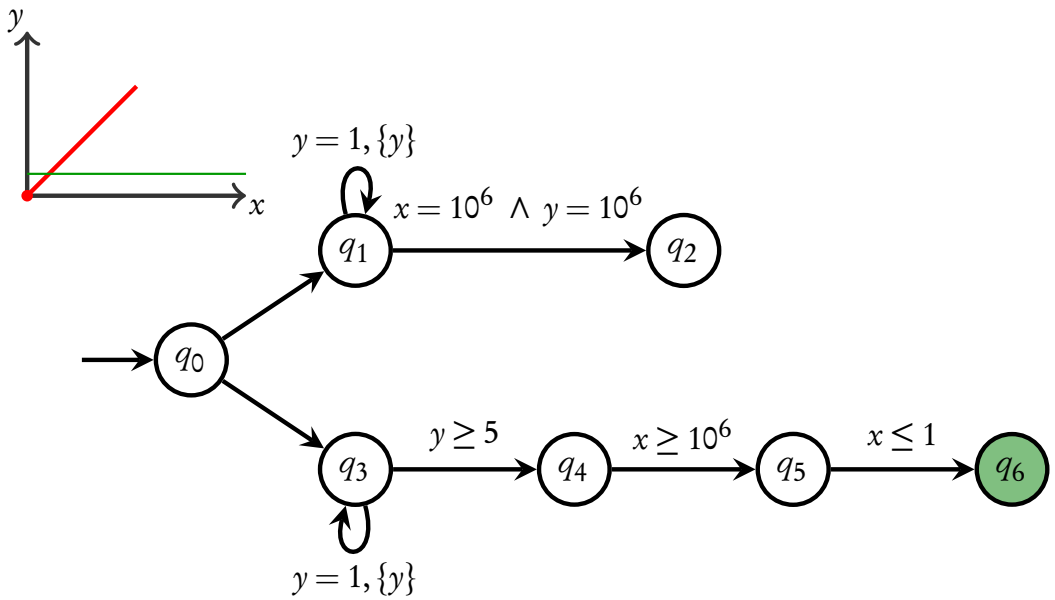

y
x
y = 1, {y}
x = 10^6 ∧ y = 10^6
q1
q2
q0
q3
y ≥ 5
q4
x ≥ 10^6
q5
x ≤ 1
q6
y = 1, {y}

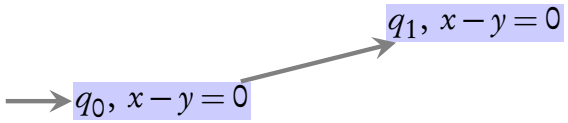


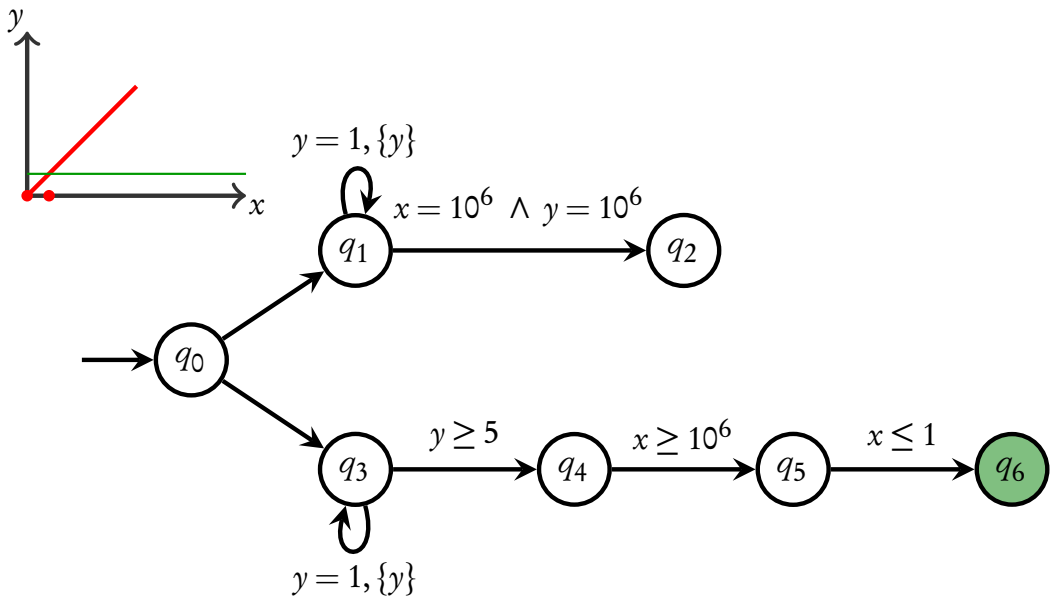
$\longrightarrow q_0,\ x - y = 0$

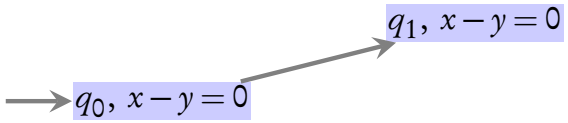

y
x
$y = 1, \{y\}$
$x = 10^6 \wedge y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

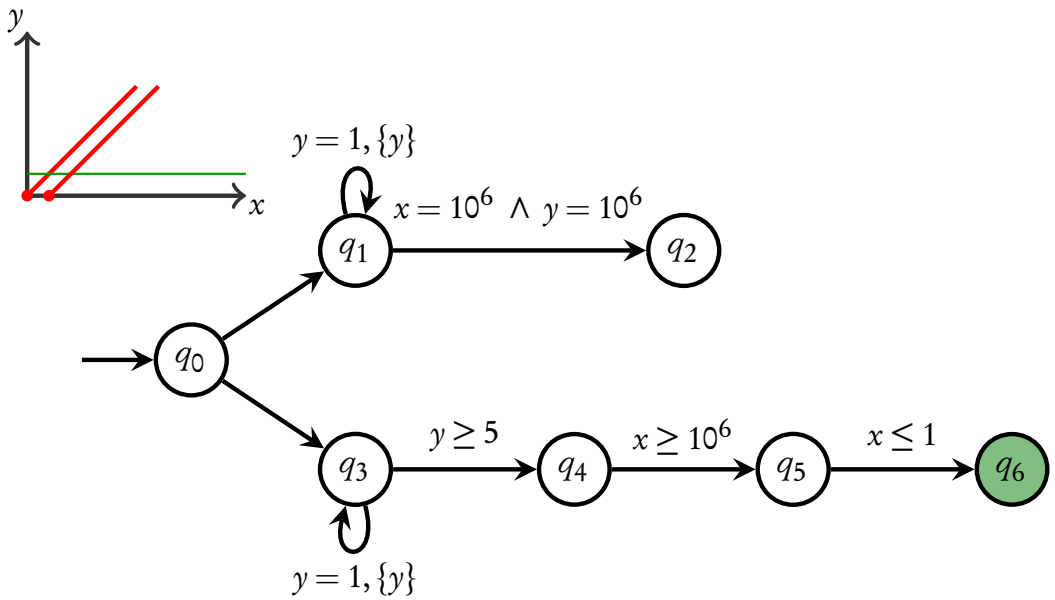

$q_1,\ x - y = 0$
$q_0,\ x - y = 0$

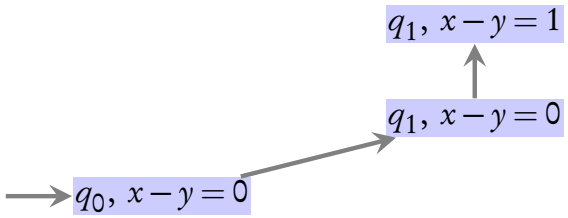

y
x
$y = 1, \{y\}$
$x = 10^6 \wedge y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

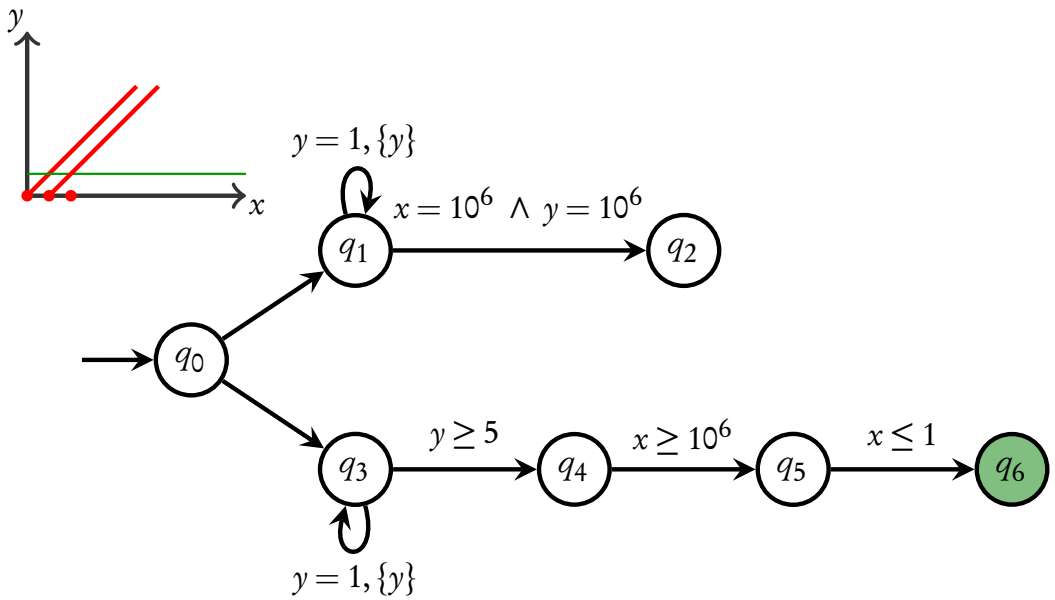

$q_1,\ x - y = 0$
$q_0,\ x - y = 0$

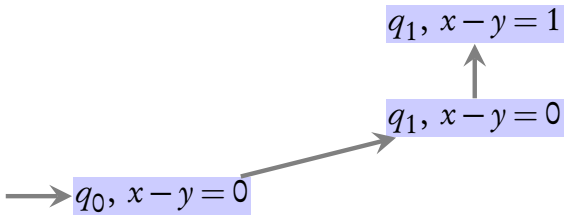

y
x
$y = 1, \{y\}$
$x = 10^6 \wedge y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

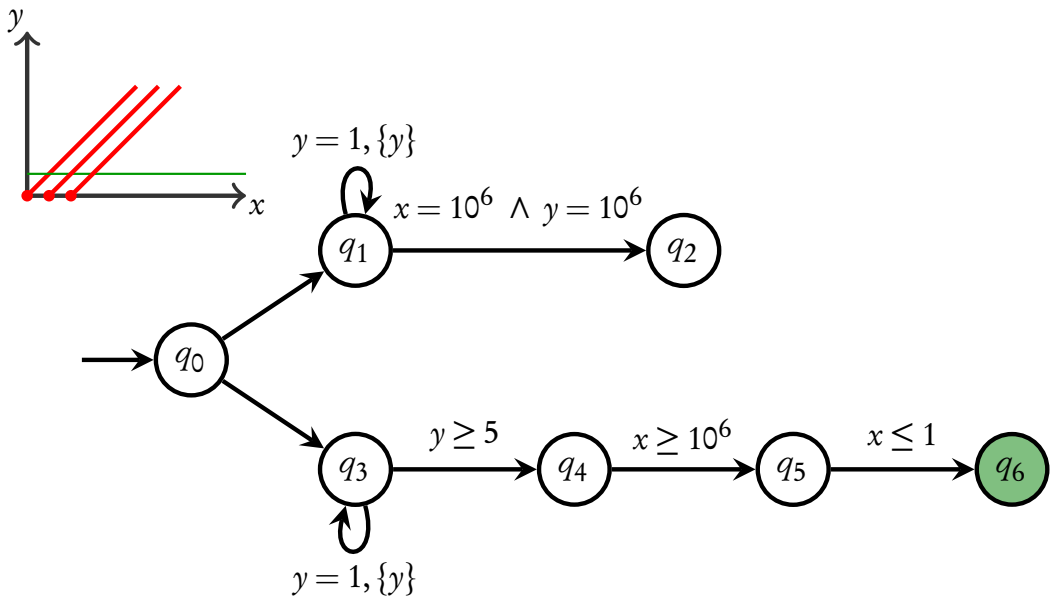

$q_1,\ x - y = 0$
$q_0,\ x - y = 0$

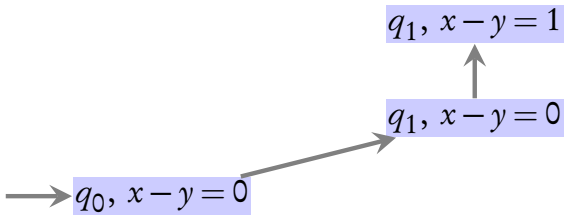

y
x
$y = 1, \{y\}$
$x = 10^6 \wedge y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

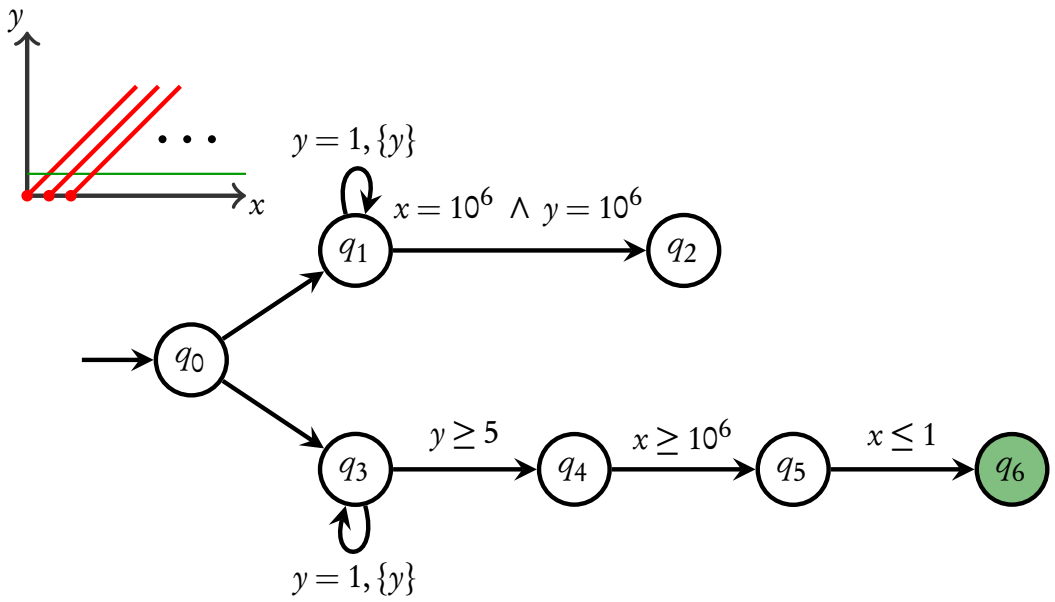

$q_1,\ x - y = 1$
$q_1,\ x - y = 0$
$q_0,\ x - y = 0$

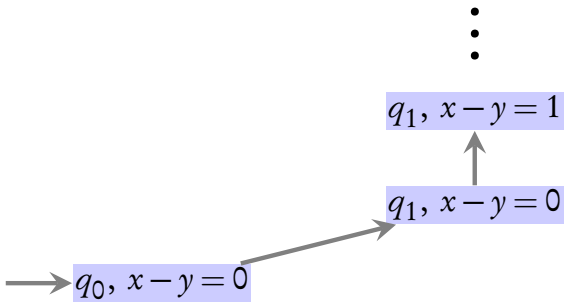

y
x
$y = 1, \{y\}$
$x = 10^6 \wedge y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

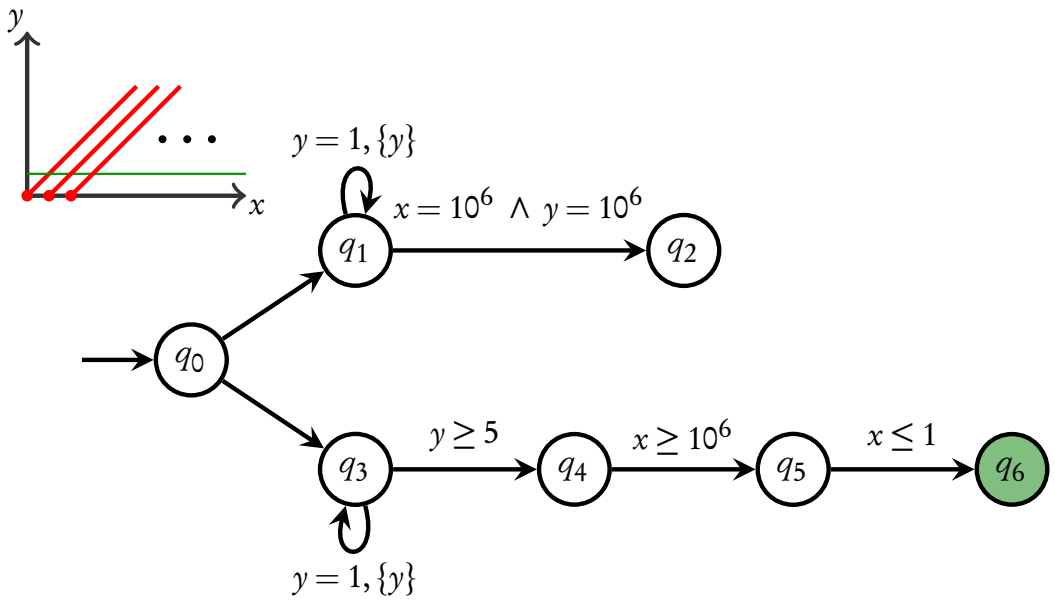

$q_1,\ x - y = 1$
$q_1,\ x - y = 0$
$q_0,\ x - y = 0$

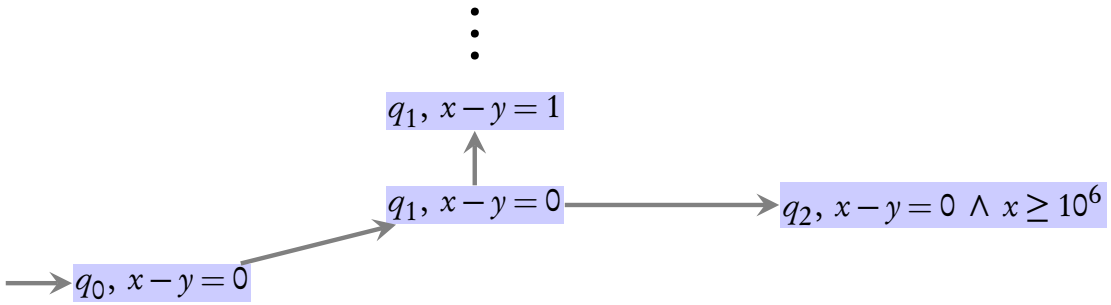

y
x
$y = 1, \{y\}$
$x = 10^6 \wedge y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

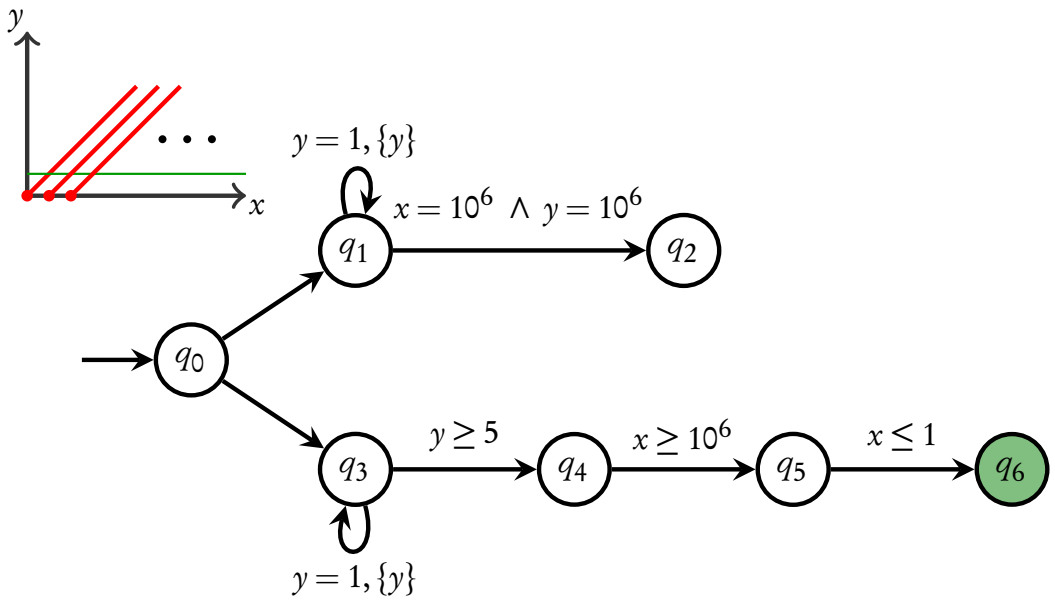

$q_1,\ x - y = 1$
$q_1,\ x - y = 0$
$q_0,\ x - y = 0$

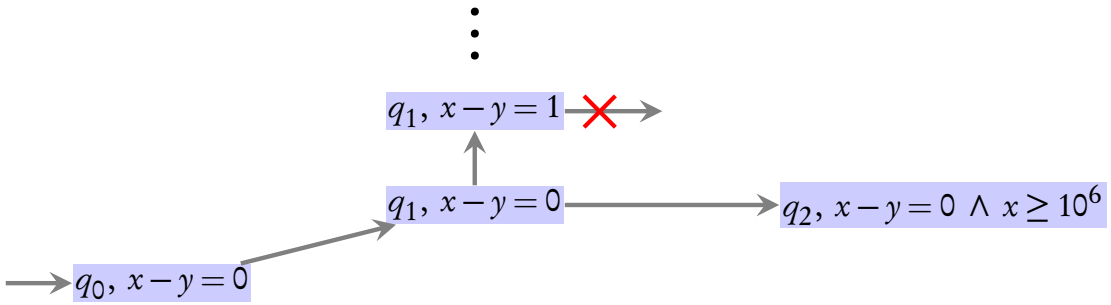

y
x
y = 1, {y}
x = 10^6 ∧ y = 10^6
q1
q2
q0
q3
y ≥ 5
q4
x ≥ 10^6
q5
x ≤ 1
q6
y = 1, {y}

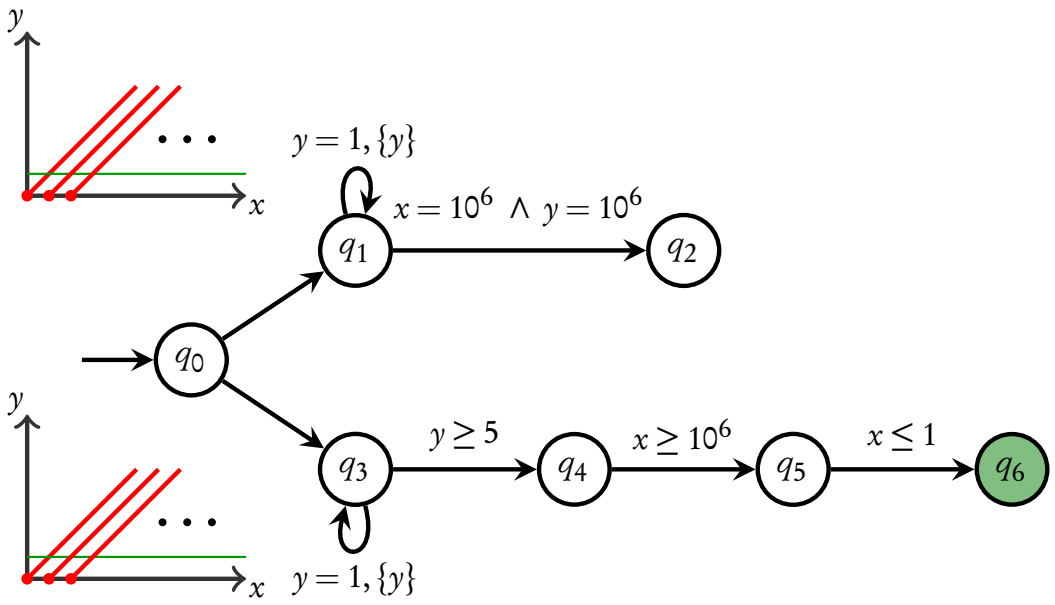

q1, x − y = 1
q1, x − y = 0
q0, x − y = 0

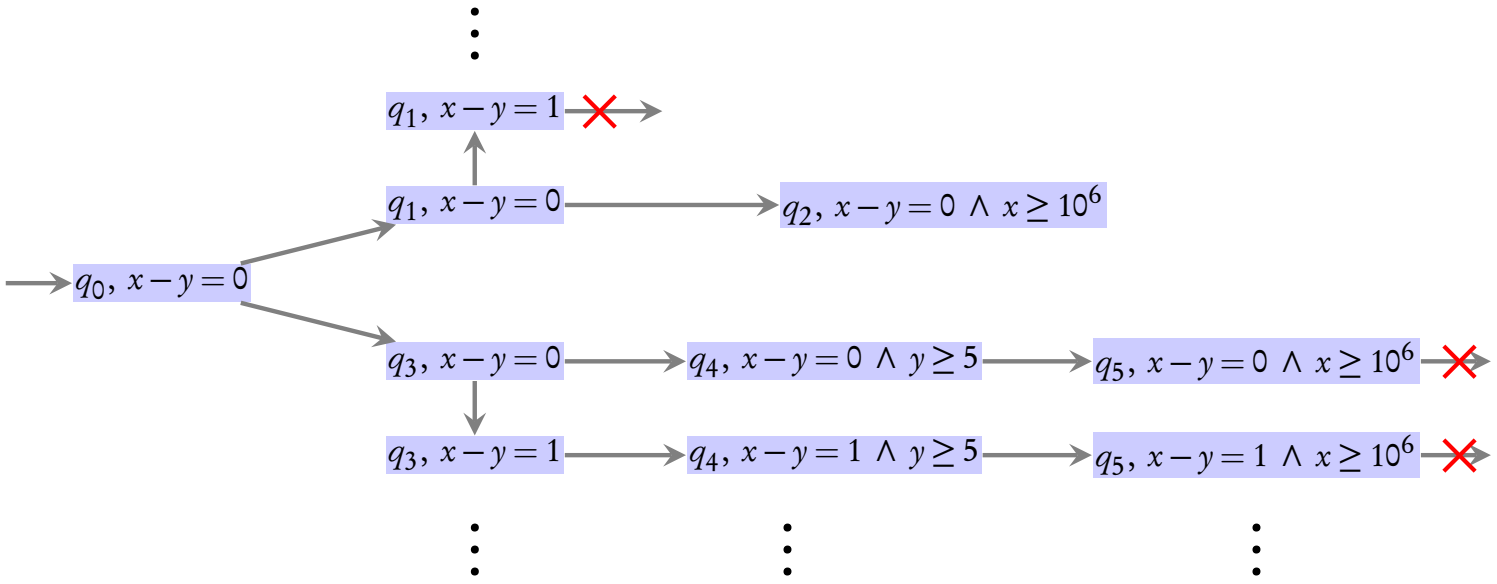

y
x
y = 1, {y}
x = 10^6 ∧ y = 10^6
q1
q2
q0
q3
y ≥ 5
q4
x ≥ 10^6
q5
x ≤ 1
q6
y = 1, {y}

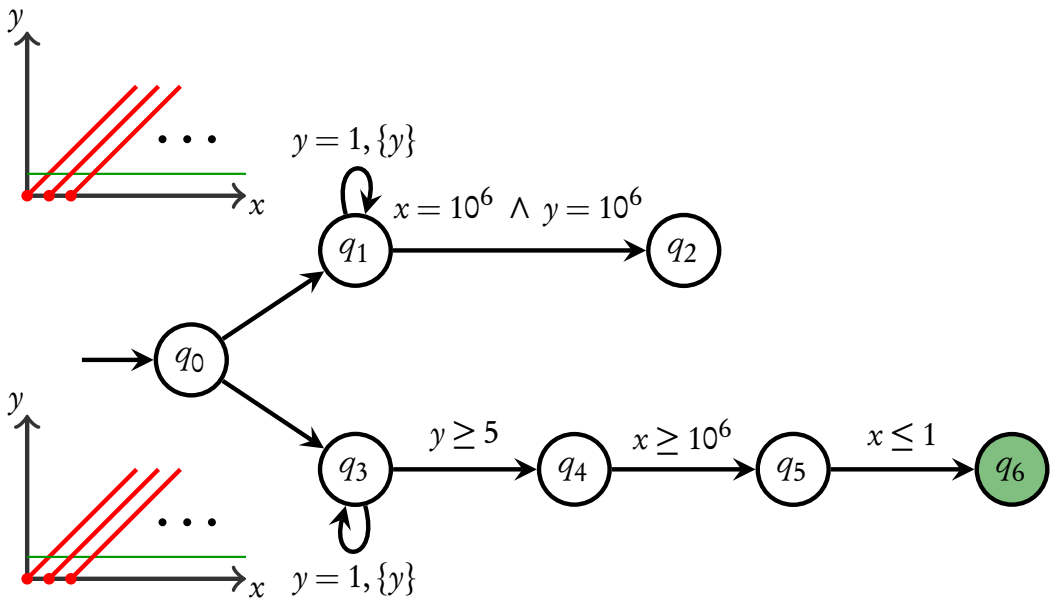

q1, x − y = 1
q1, x − y = 0
q2, x − y = 0 ∧ x ≥ 10^6
q0, x − y = 0

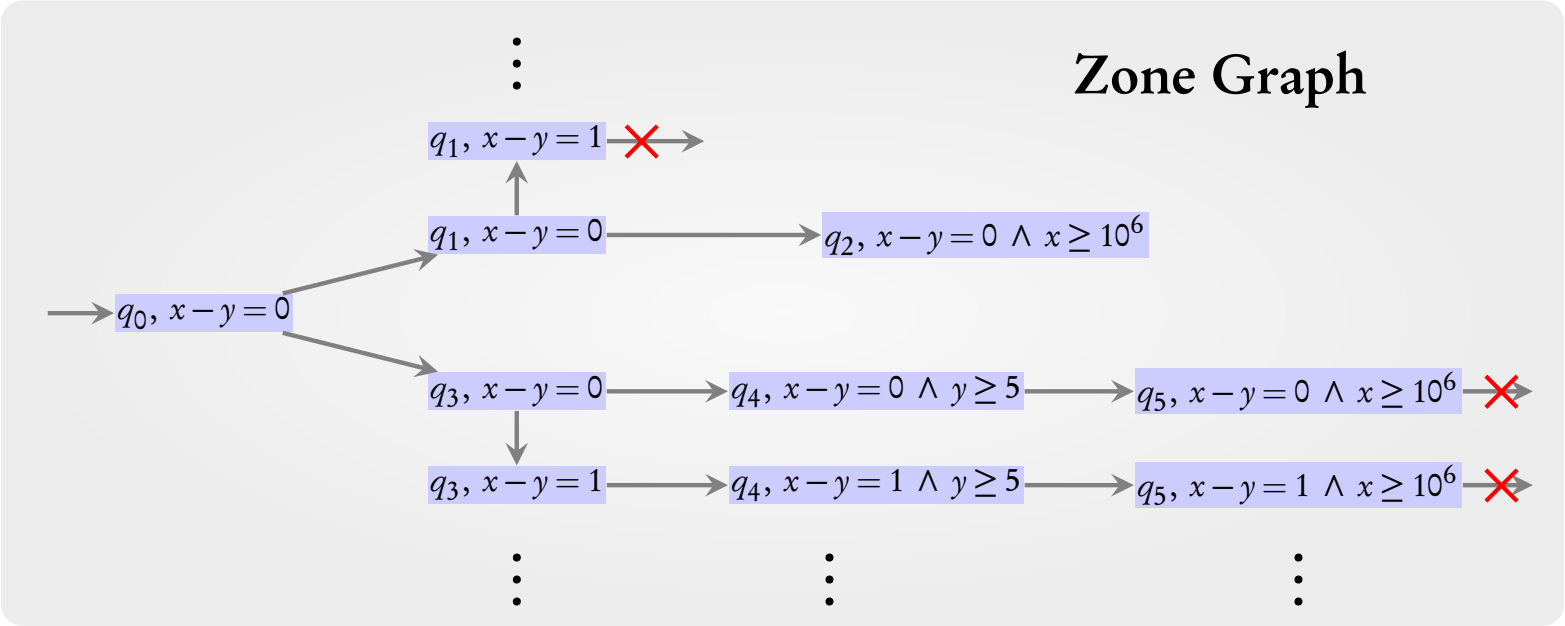

y
x
y = 1, {y}
x = 10^6 ∧ y = 10^6
q1
q2
q0
q3
y ≥ 5
q4
x ≥ 10^6
q5
x ≤ 1
q6
y = 1, {y}


q1, x − y = 1
q1, x − y = 0
q2, x − y = 0 ∧ x ≥ 10^6
q0, x − y = 0


y
x
y = 1, {y}
x = 10^6 ∧ y = 10^6
q1
q2
q0
q3
y ≥ 5
q4
x ≥ 10^6
q5
x ≤ 1
q6
y = 1, {y}


q1, x − y = 1
q1, x − y = 0
q2, x − y = 0 ∧ x ≥ 10^6
q0, x − y = 0
q3, x − y = 0
q4, x − y = 0 ∧ y ≥ 5
q5, x − y = 0 ∧ x ≥ 10^6
q3, x − y = 1
q4, x − y = 1 ∧ y ≥ 5
q5, x − y = 1 ∧ x ≥ 10^6


y
x
y = 1, {y}
x = 10^6 ∧ y = 10^6
q1
q2
q0
y
x
q3
y ≥ 5
q4
x ≥ 10^6
q5
x ≤ 1
q6
y = 1, {y}


Zone Graph
q1, x − y = 1
q1, x − y = 0
q2, x − y = 0 ∧ x ≥ 10^6
q0, x − y = 0
q3, x − y = 0
q4, x − y = 0 ∧ y ≥ 5
q5, x − y = 0 ∧ x ≥ 10^6
q3, x − y = 1
q4, x − y = 1 ∧ y ≥ 5
q5, x − y = 1 ∧ x ≥ 10^6

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 **[Dill'89]**

Sound and complete [Daws, Tripakis'98]

Zone graph preserves state **reachability**

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 **[Dill'89]**

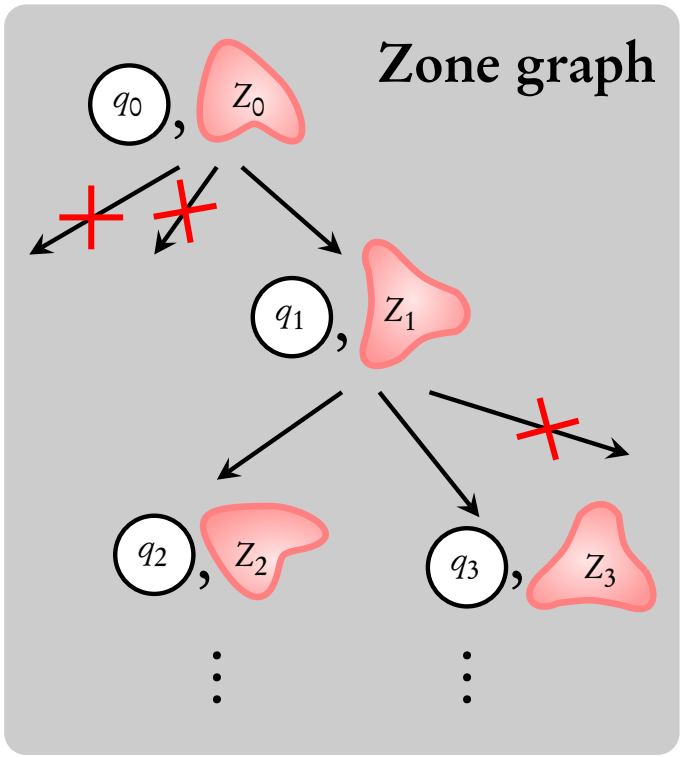
Sound and complete [Daws, Tripakis'98]

Zone graph preserves state **reachability**

But zone graph could be **infinite!**

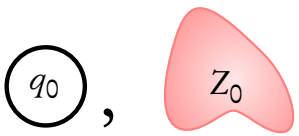
Coming next: Finite abstractions of the zone graph

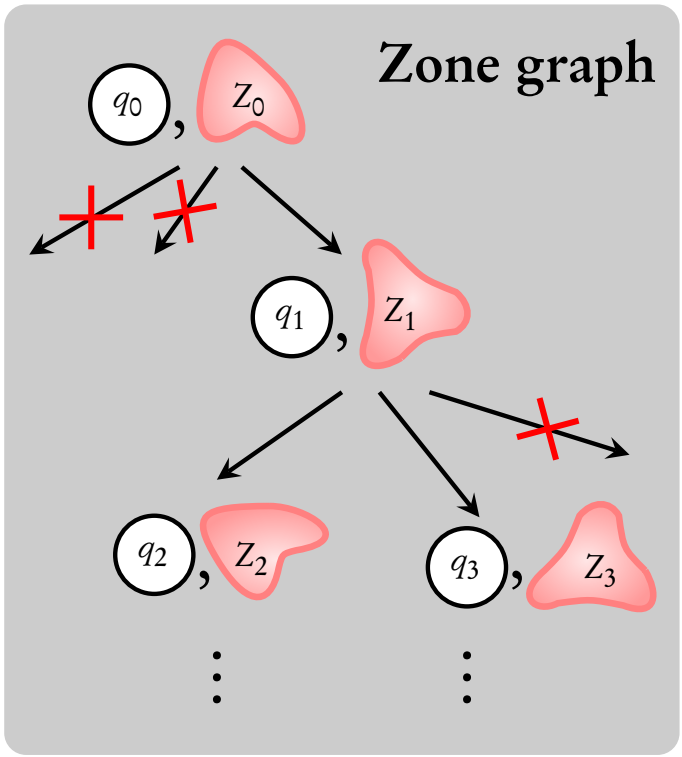
Abstractions



potentially infinite...

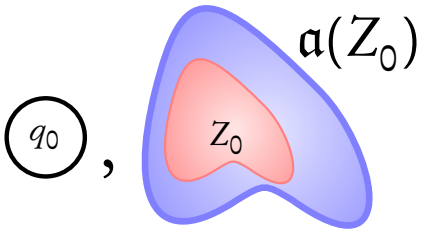
Abstractions

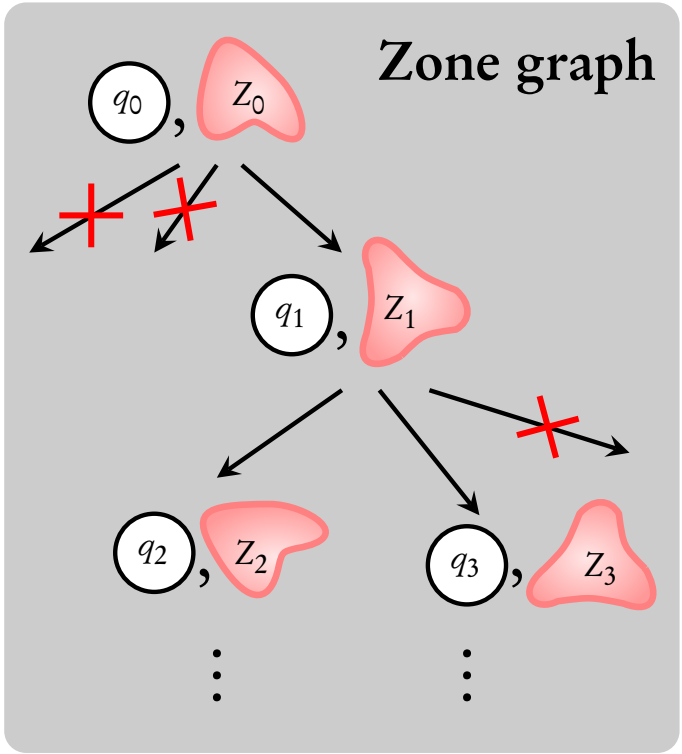




potentially infinite...

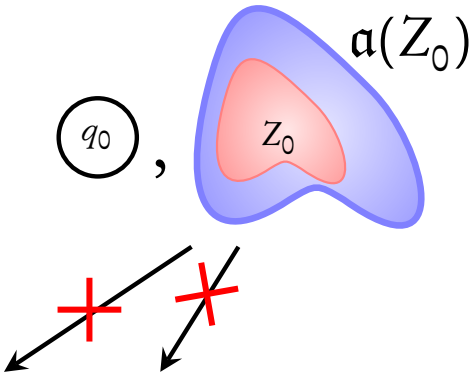
Abstractions

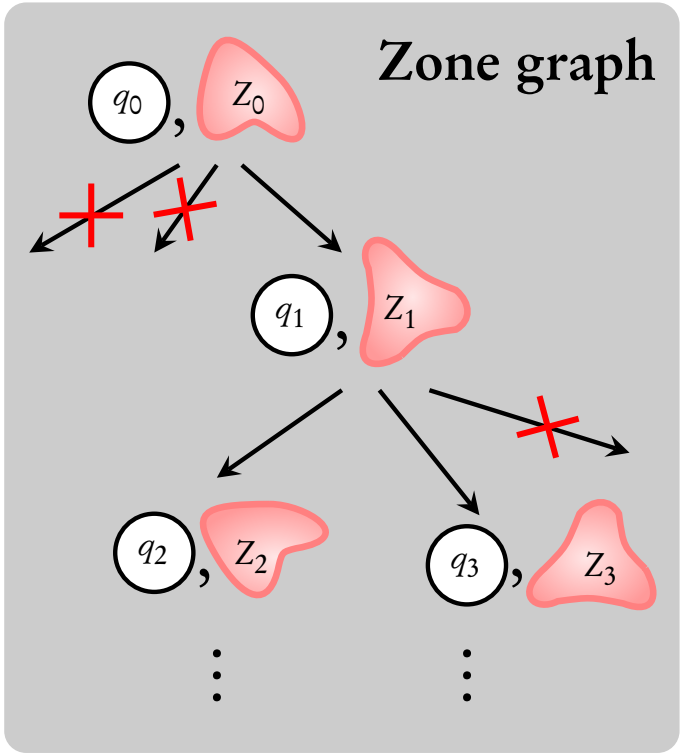




potentially infinite...

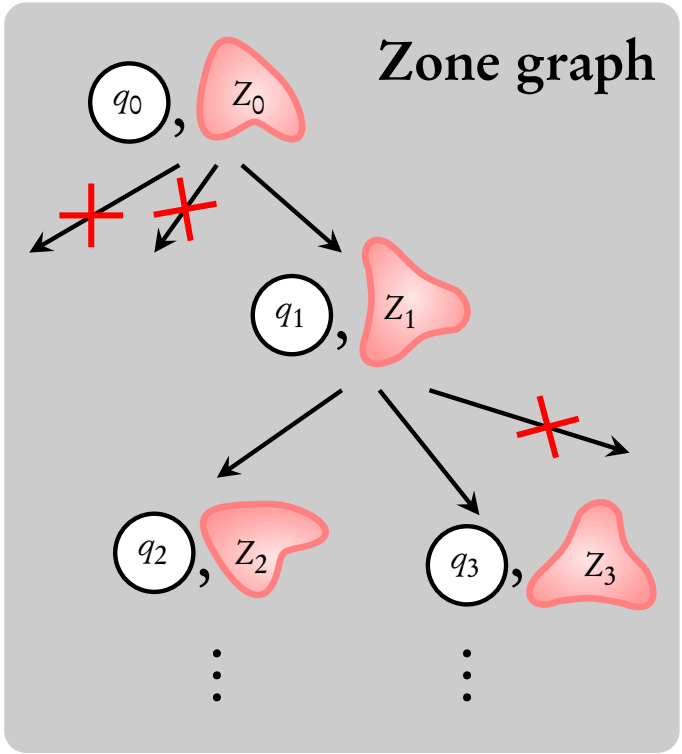
Abstractions



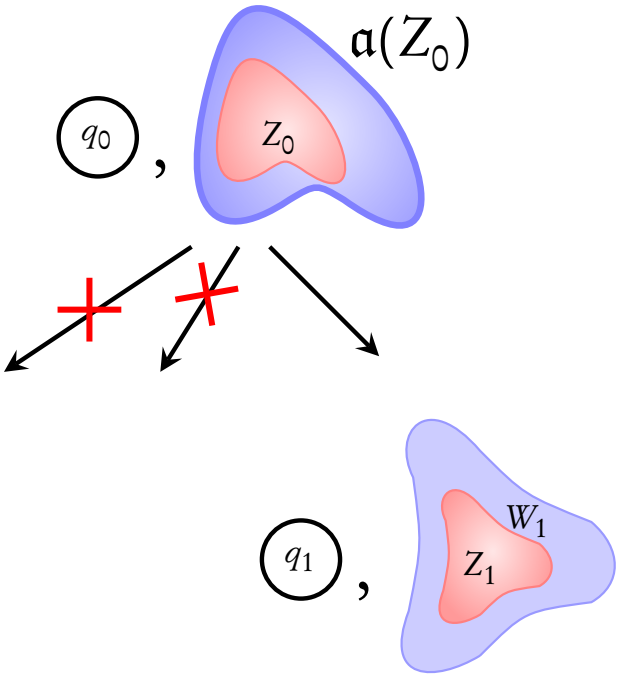


potentially infinite...

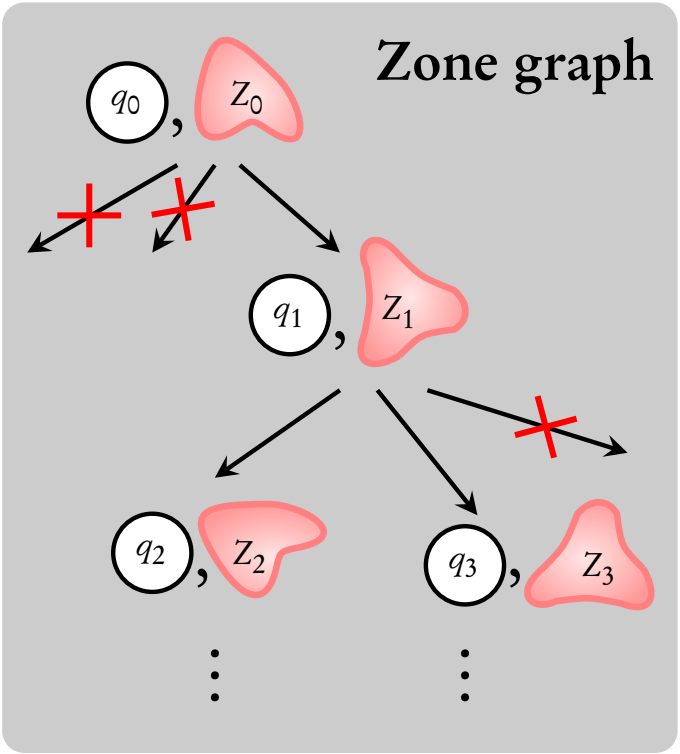
Abstractions



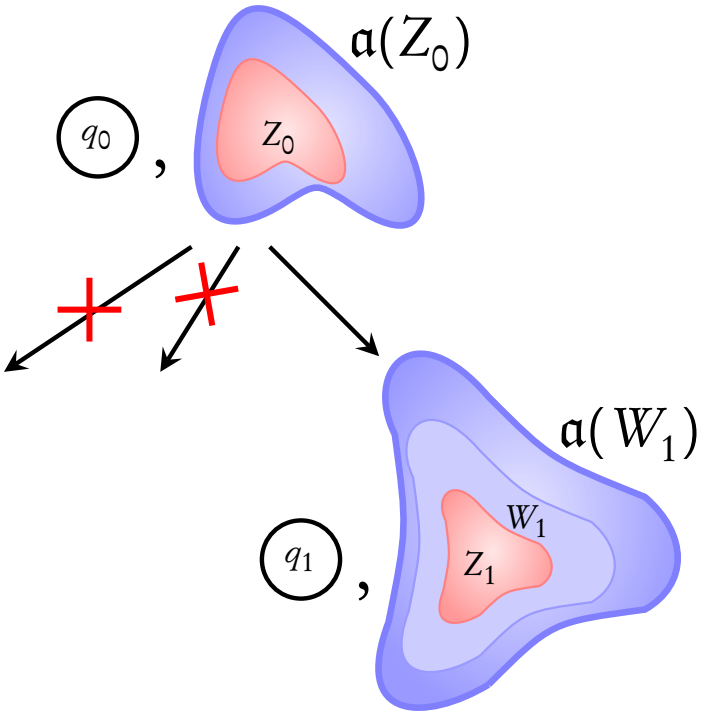
potentially infinite...



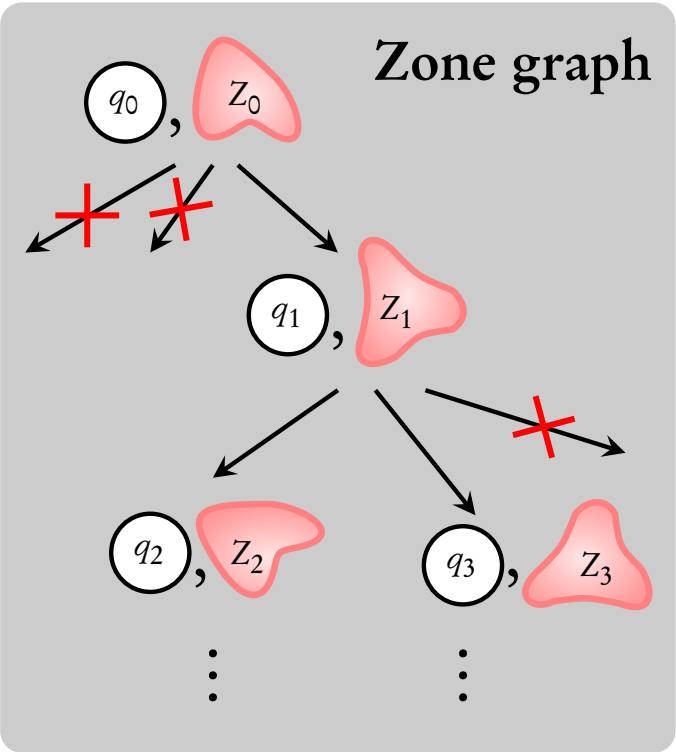
Abstractions



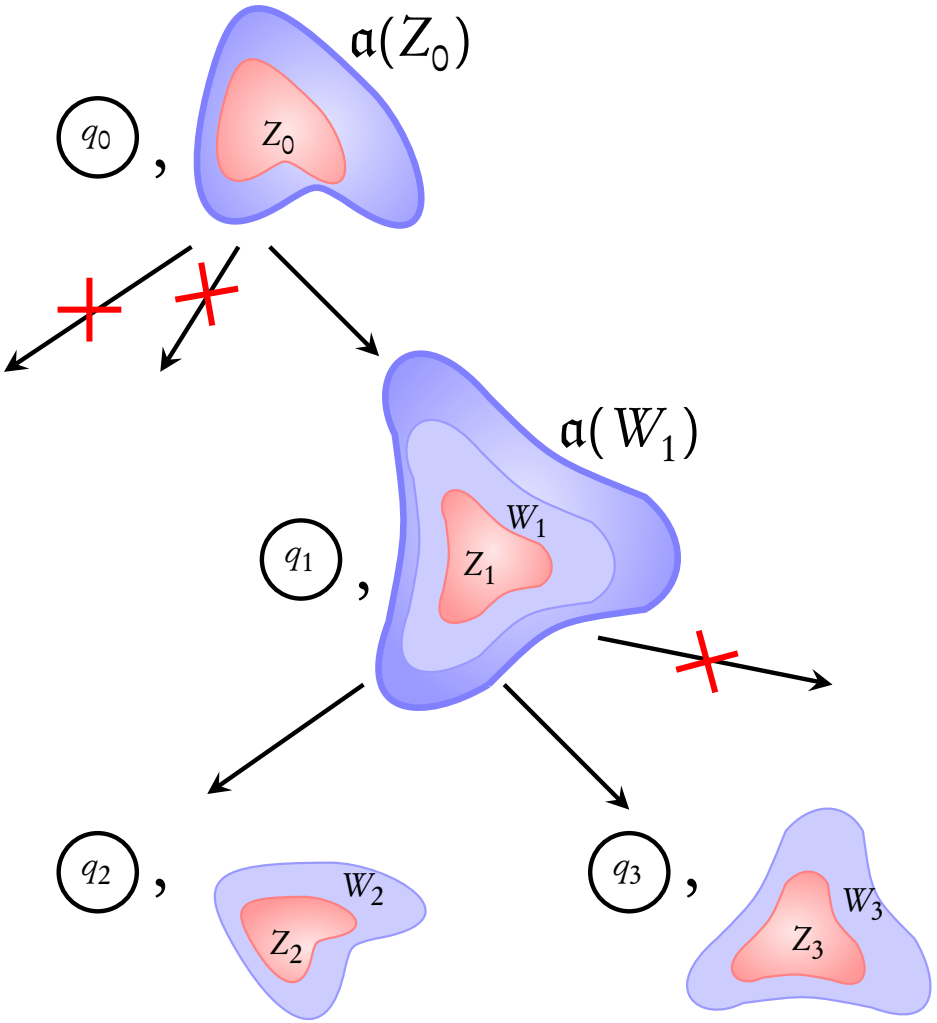
potentially infinite...



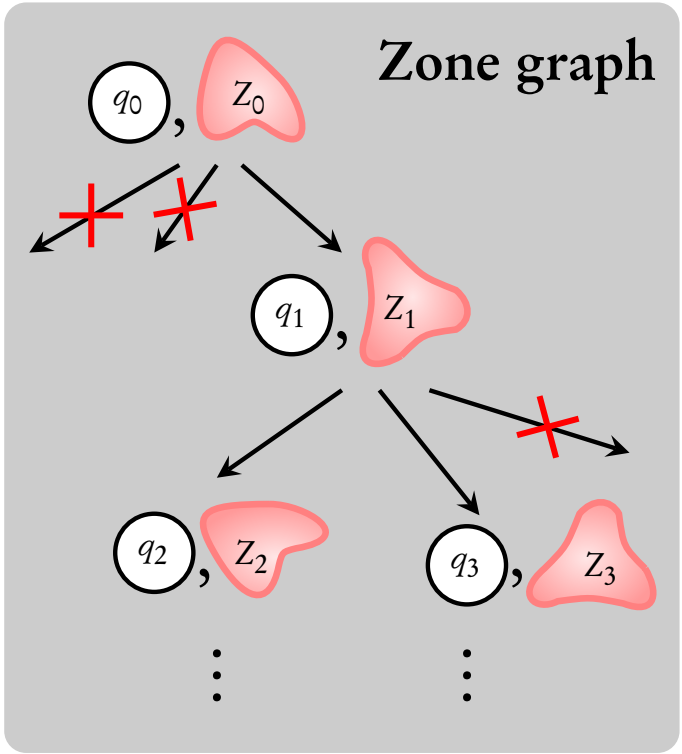
Abstractions



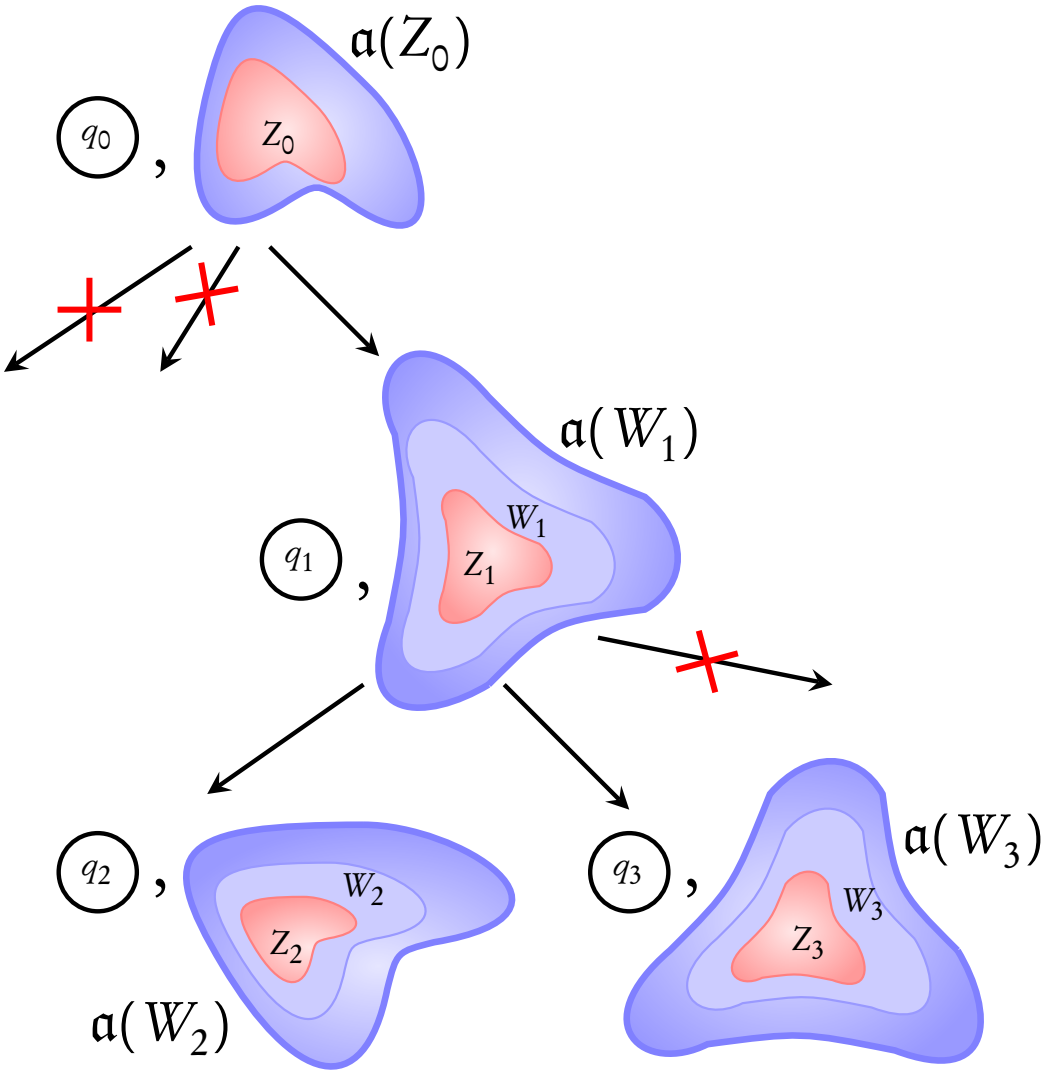
potentially infinite...



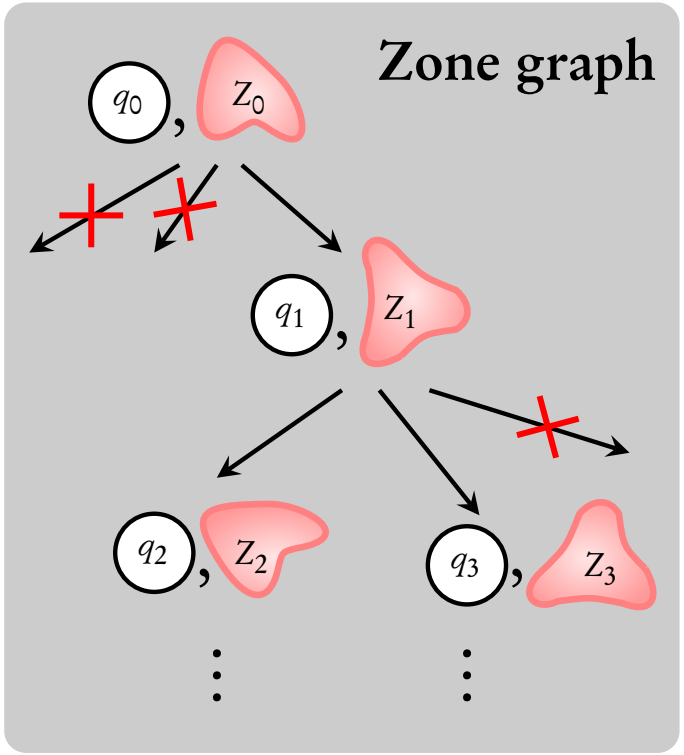
Abstractions



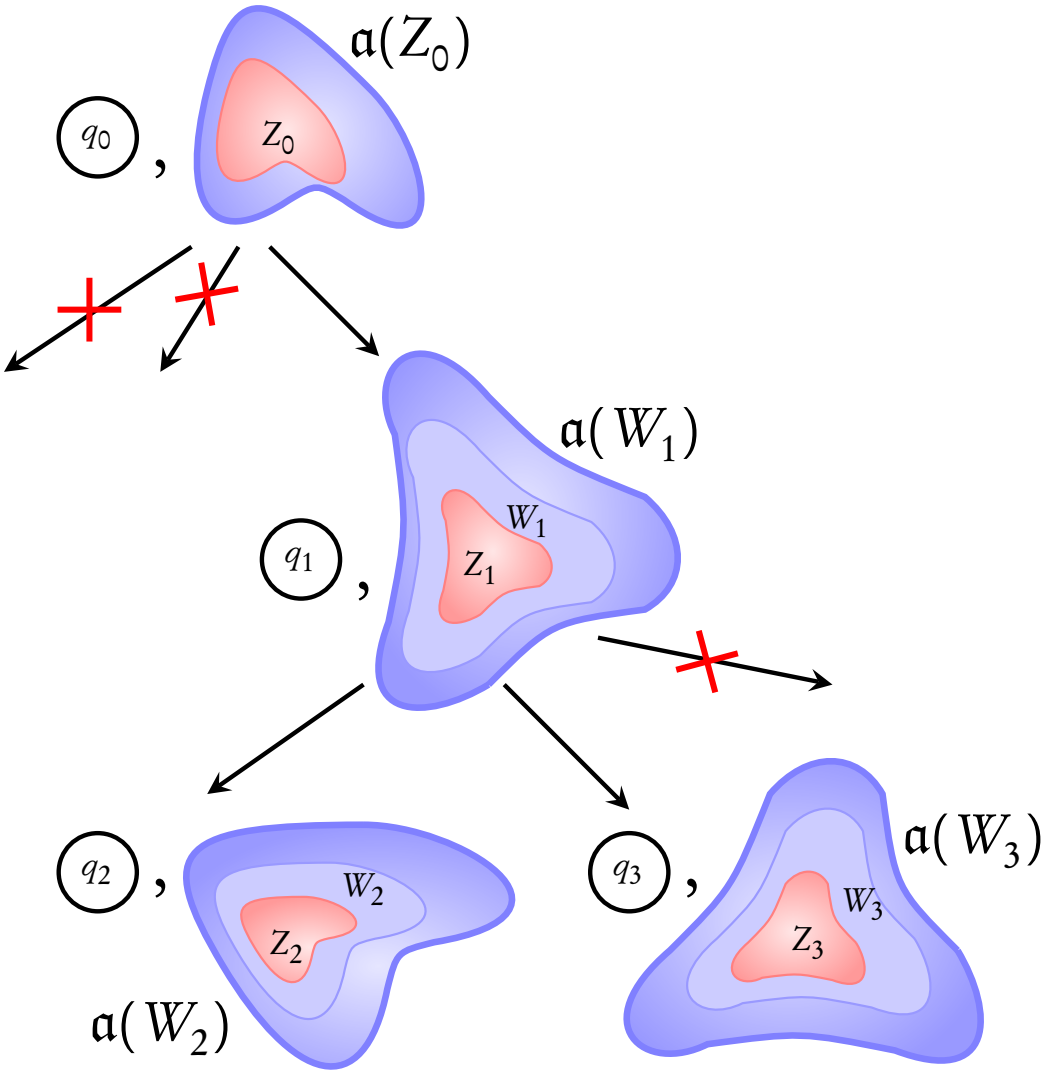
potentially infinite...



Abstractions



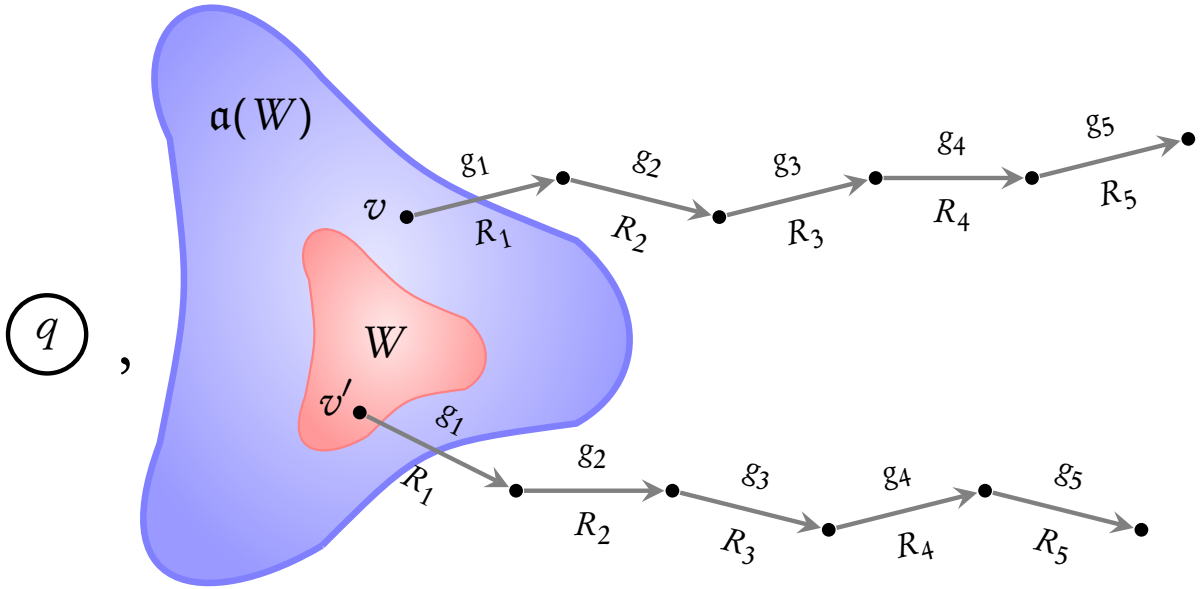
potentially infinite...



Question: How do we choose these abstraction functions?

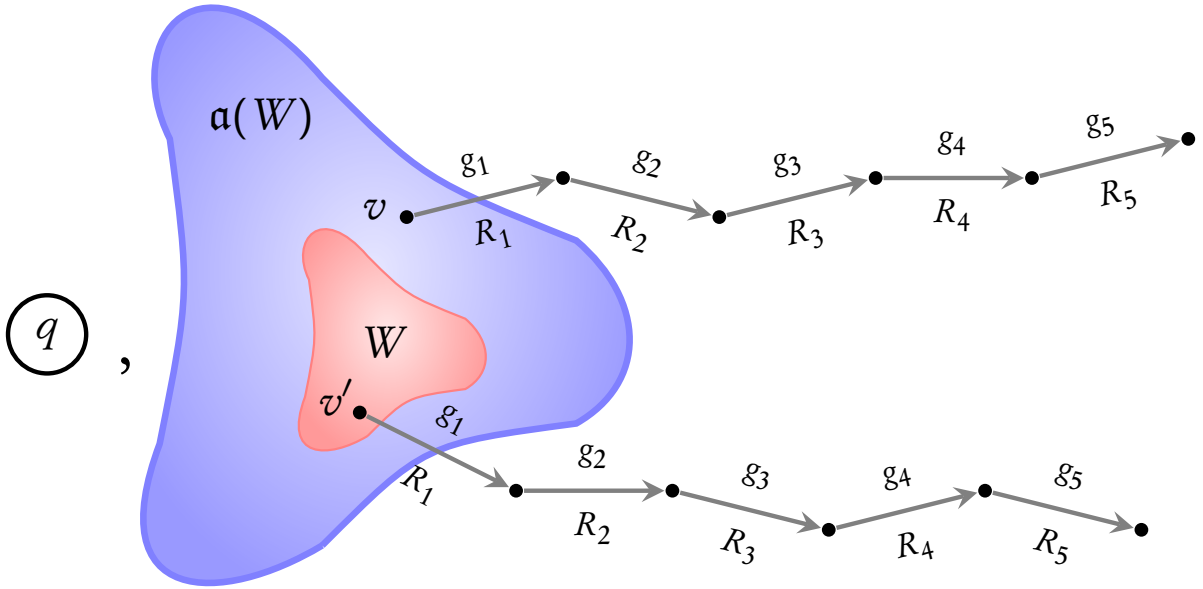
Condition 1: Abstractions should have **finite range**

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



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

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



Coming next: Simulation relations

(q, v) (q, v')

is simulated by

$$(q, v) \preccurlyeq (q, v')$$

is simulated by

$$\begin{tikzcd}
(q, v) \arrow[d, "g"', "R"] & \precsim & (q, v') \\
(q_1, v_1) & &
\end{tikzcd}$$

is simulated by

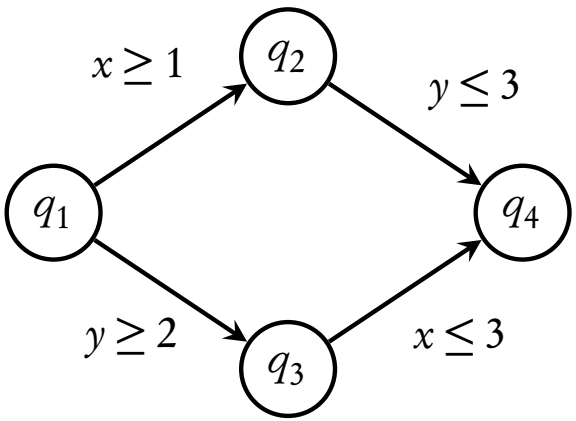
$$
\begin{tikzcd}
(q, v) \arrow[d, "g"', "R"] & \preccurlyeq & (q, v') \arrow[d, "g"', "R"] \\
(q_1, v_1) & & (q_1, v'_1)
\end{tikzcd}
$$

is simulated by

$$
\begin{tikzcd}
(q, v) \arrow[d, "g"', "R"] & \precsim & (q, v') \arrow[d, "g"', "R"] \\
(q_1, v_1) & \precsim & (q_1, v'_1)
\end{tikzcd}
$$

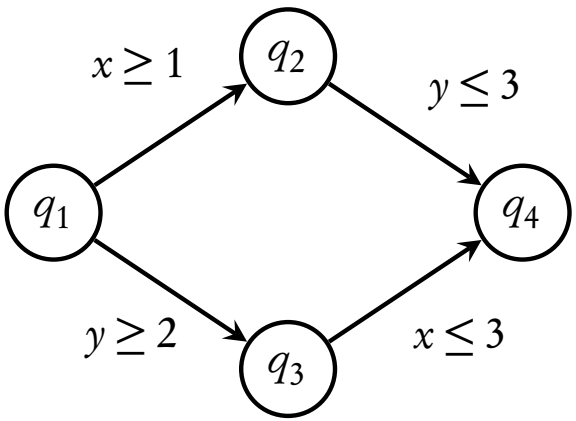
is simulated by

$$
\begin{tikzcd}
(q, v) \arrow[d, "g"', "R"] & \preccurlyeq & (q, v') \arrow[d, "g"', "R"] \\
(q_1, v_1) & \preccurlyeq & (q_1, v'_1)
\end{tikzcd}
$$



is simulated by

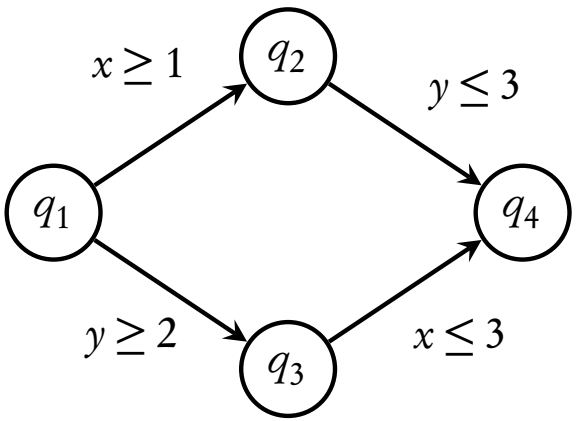
$$
\begin{tikzcd}
(q, v) \arrow[d, "g"', "R"] & \preceq & (q, v') \arrow[d, "g"', "R"] \\
(q_1, v_1) & \preceq & (q_1, v'_1)
\end{tikzcd}
$$



$(q_1, \langle x = 0.5, y = 2.1 \rangle)$

is simulated by

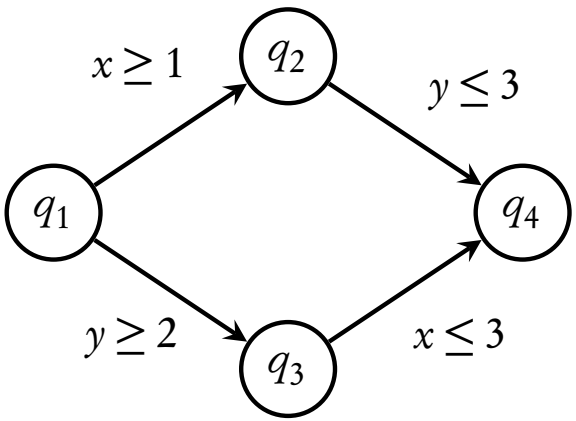
$$
\begin{tikzcd}
(q, v) \arrow[d, "g"', "R"] & \preceq & (q, v') \arrow[d, "g"', "R"] \\
(q_1, v_1) & \preceq & (q_1, v'_1)
\end{tikzcd}
$$



$(q_1, \langle x = 0.5, y = 2.1 \rangle)$ $\quad$ $(q_1, \langle x = 0.5, y = 2.9 \rangle)$

is simulated by

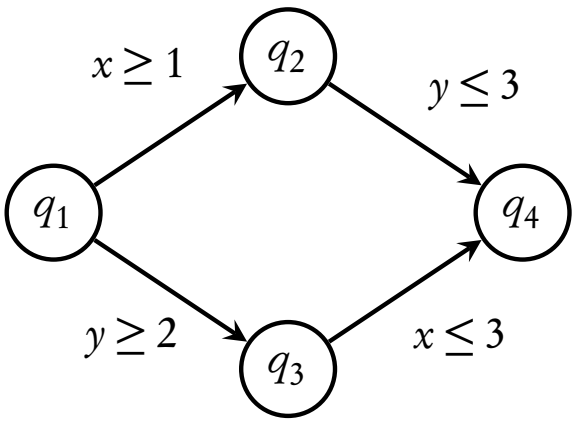
$$
\begin{tikzcd}
(q, v) \arrow[d, "g"', "R"] & \not\preceq & (q, v') \arrow[d, "g"', "R"] \\
(q_1, v_1) & \not\preceq & (q_1, v'_1)
\end{tikzcd}
$$



$(q_1, \langle x = 0.5, y = 2.1 \rangle) \not\preceq (q_1, \langle x = 0.5, y = 2.9 \rangle)$

is simulated by

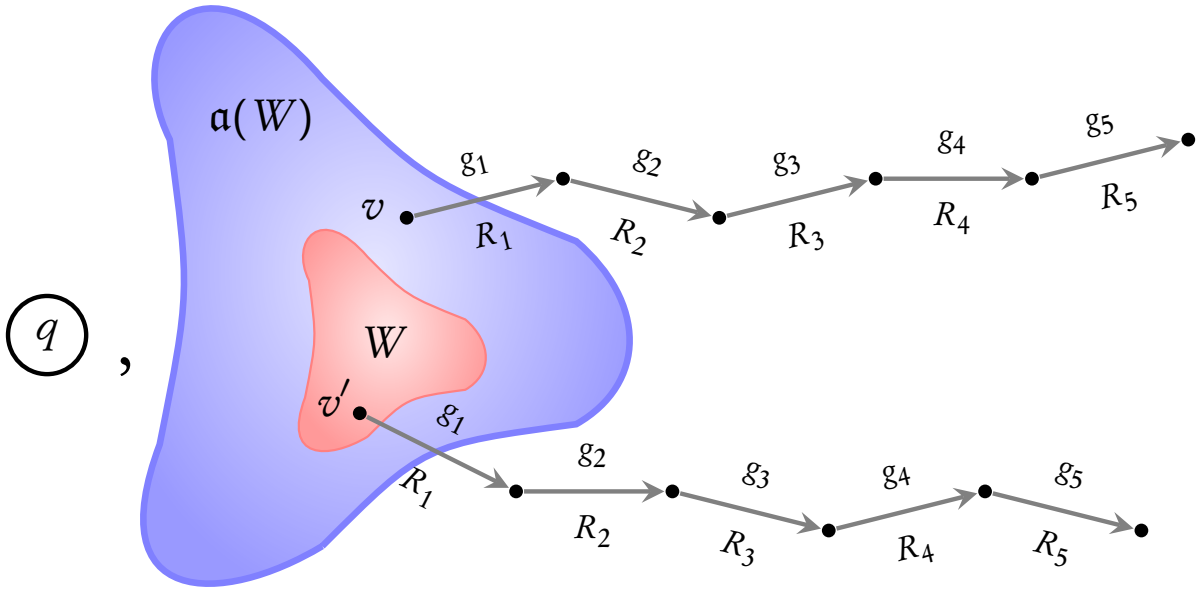
$$
\begin{tikzcd}
(q, v) \arrow[d, "g"', "R"] & \precsim & (q, v') \arrow[d, "g"', "R"] \\
(q_1, v_1) & \precsim & (q_1, v'_1)
\end{tikzcd}
$$



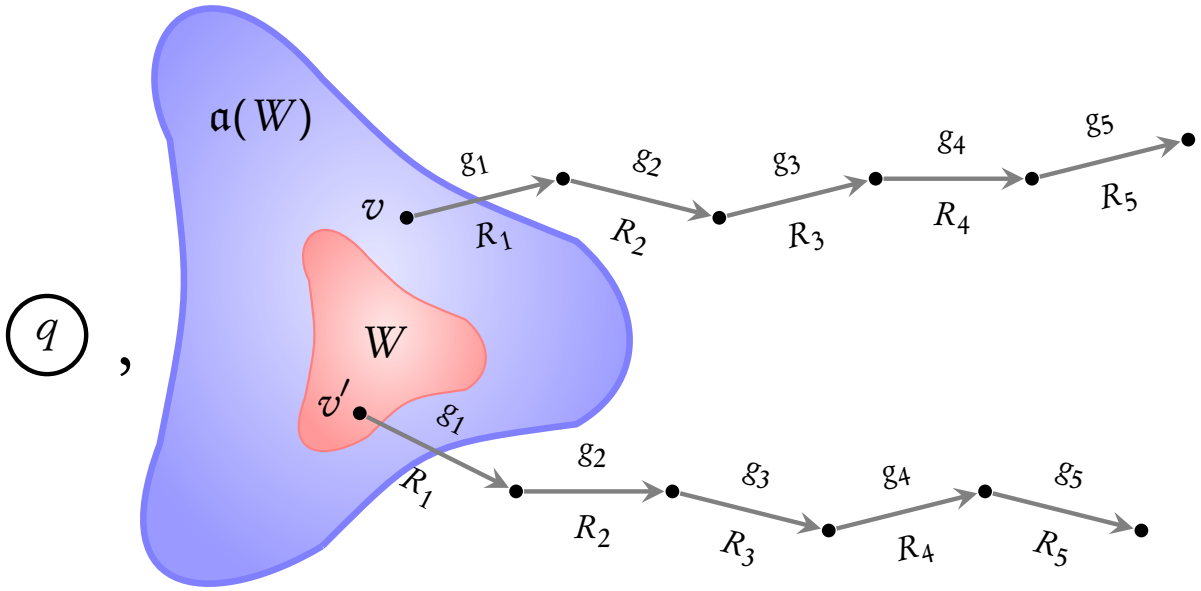
$(q_1, \langle x = 0.5, y = 2.1 \rangle) \not\precsim (q_1, \langle x = 0.5, y = 2.9 \rangle)$

$(q_1, \langle x = 0.5, y = 2.1 \rangle) \precsim (q_1, \langle x = 0.5, y = 1 \rangle)$

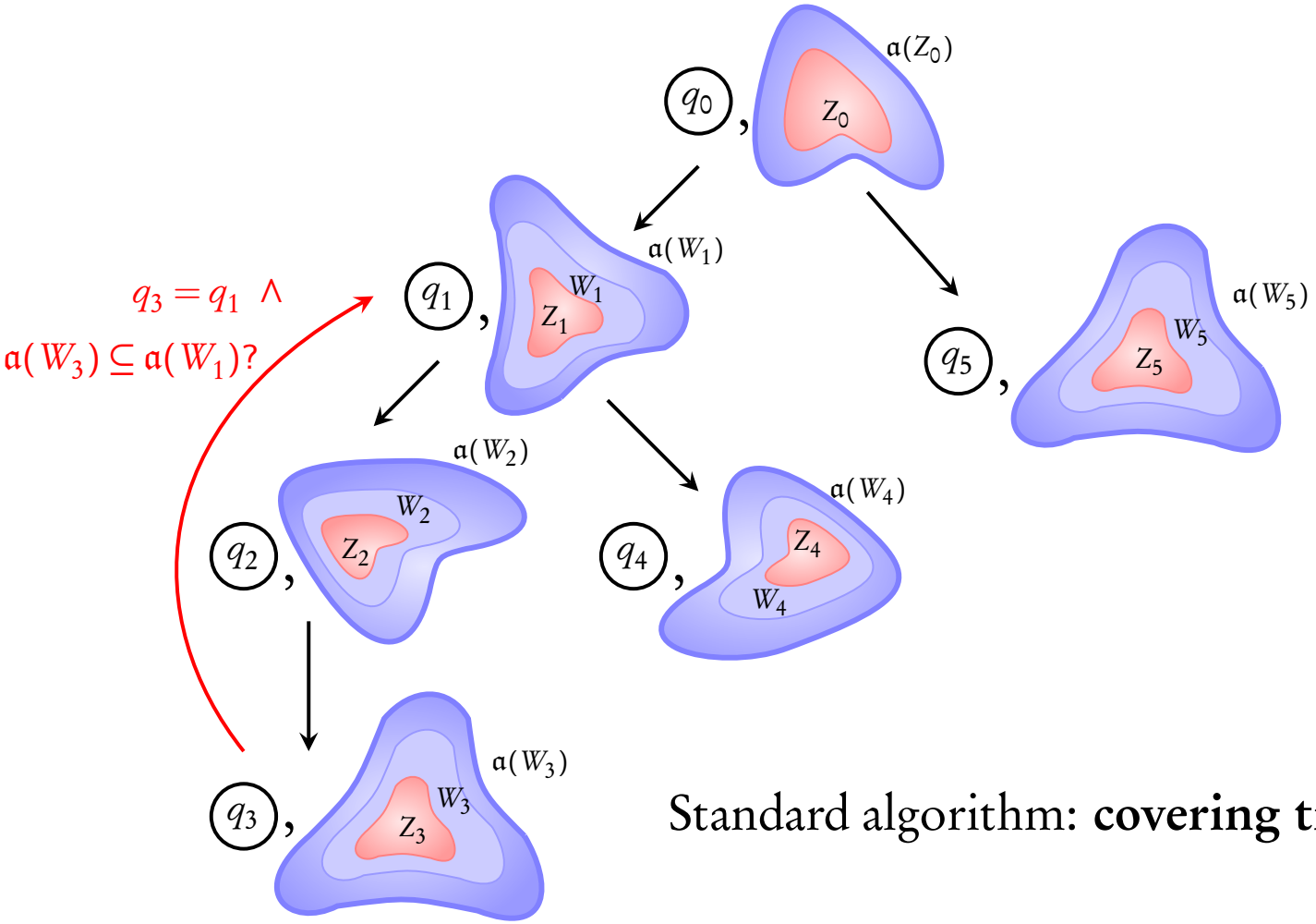
Abstractions from simulations



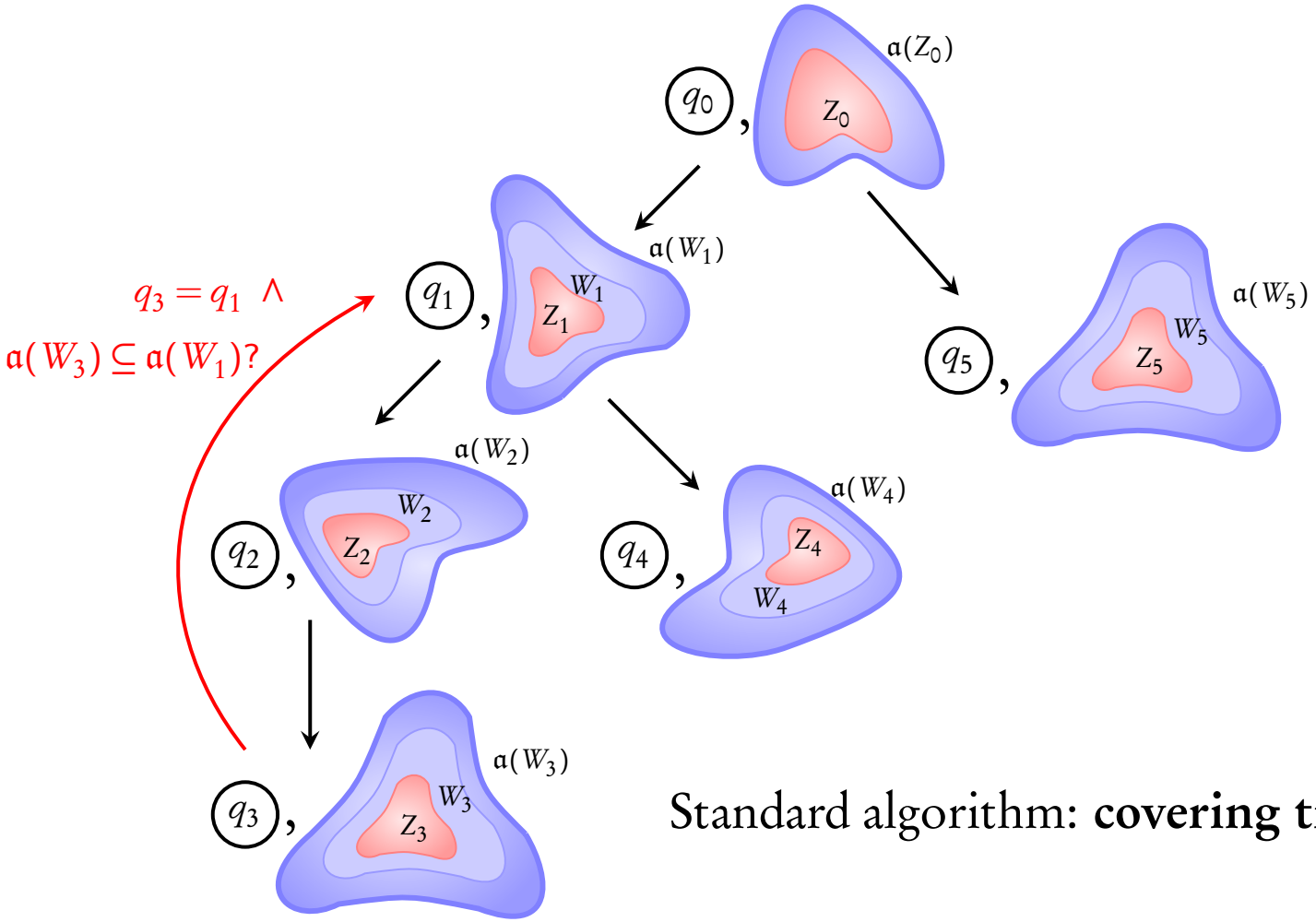
Abstractions from simulations



- $\mathfrak{a}(W) = \mathfrak{a}_q(W) : \{ v \mid \textbf{exists } v' \in W \text{ s.t. } (q,v) \preceq (q,v') \}$



Standard algorithm: **covering tree**

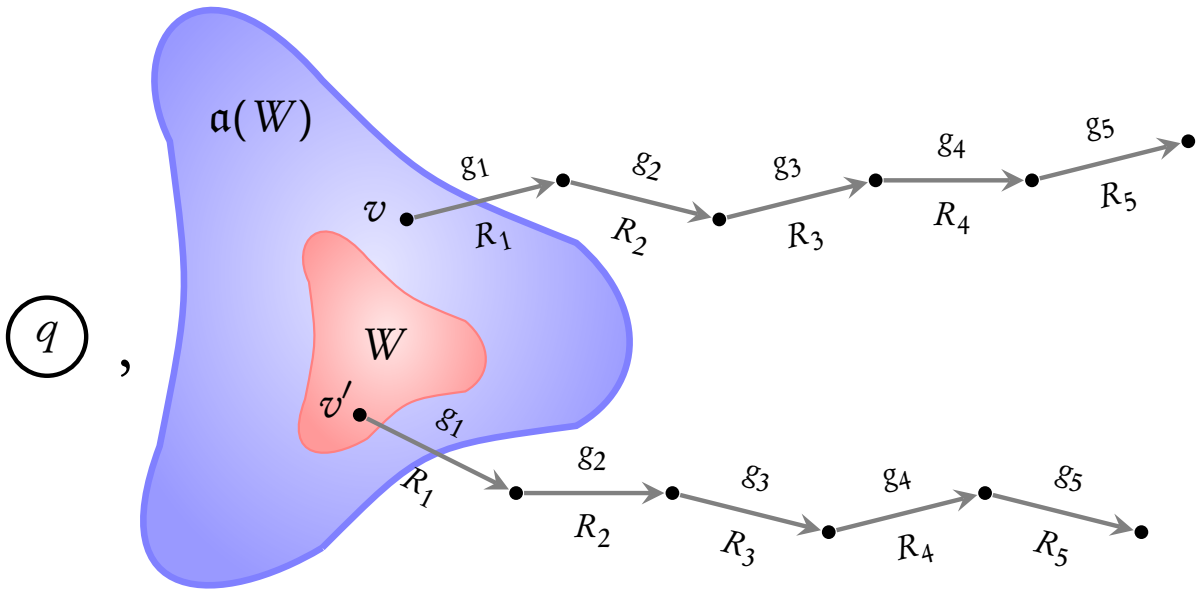


Standard algorithm: **covering tree**

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

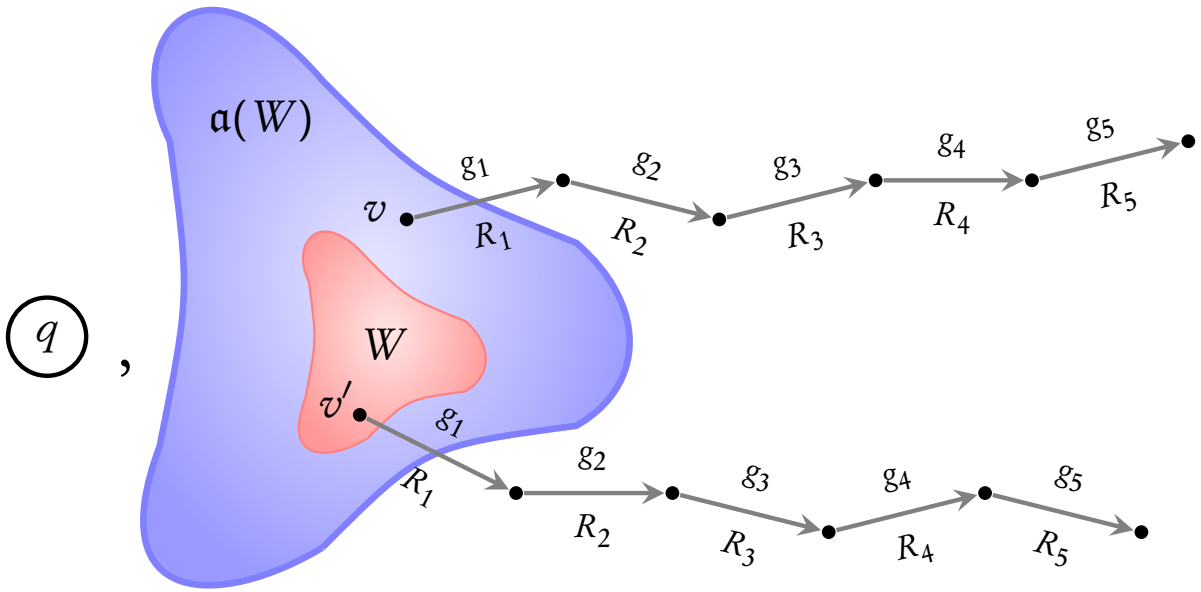
Condition 1: Abstractions should have **finite range**

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



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

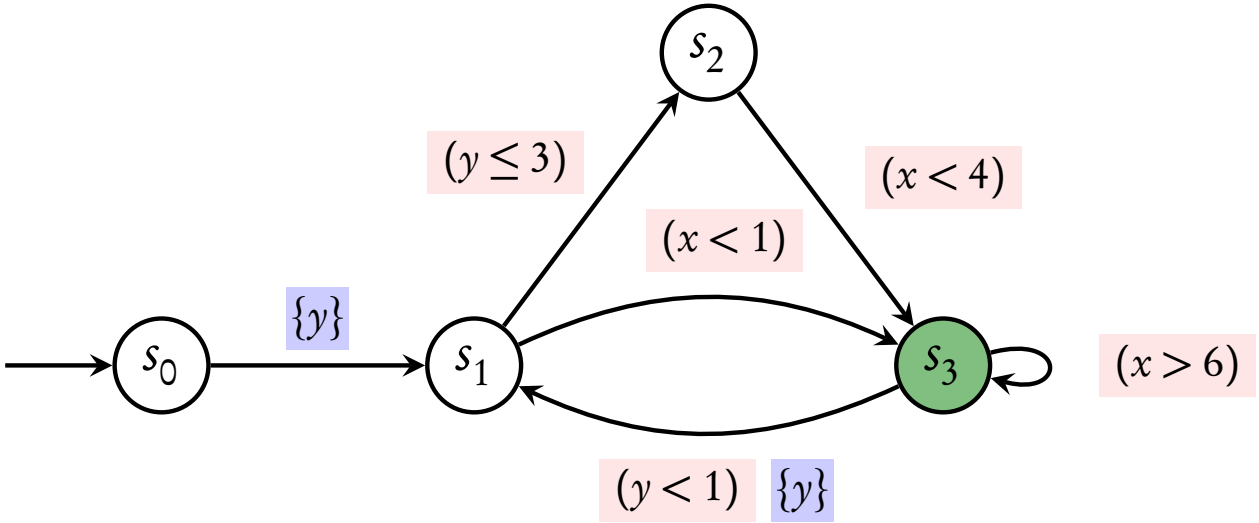
Condition 2: Abstractions should be sound $\Rightarrow$ $\mathfrak{a}(W)$ can contain only valuations **simulated** by W



Why not add **all valuations simulated** by W?

Theorem [Laroussinie, Schnoebelen'00]

Deciding **coarsest** simulation relation for a given automaton is **EXPTIME-hard**



Theorem [Laroussinie, Schnoebelen'00]

Deciding **coarsest** simulation relation for a given automaton is **EXPTIME-hard**

$(y \leq 3)$

$(x < 4)$

$(x < 1)$

$(x > 6)$

$(y < 1)$

Theorem [Laroussinie, Schnoebelen'00]

Deciding **coarsest** simulation relation for a given automaton is **EXPTIME-hard**

$(y \leq 3)$

$(x < 4)$

$(x < 1)$

$(x > 6)$

$(y < 1)$

M-bounds [Alur, Dill'90]

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

$$v \preceq_M v'$$

Theorem [Laroussinie, Schnoebelen'00]

Deciding **coarsest** simulation relation for a given automaton is **EXPTIME-hard**

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

$(x < 1)$

$(x > 6)$

$(y < 1)$

M-bounds [Alur, Dill'90]

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

$$v \preceq_M v'$$

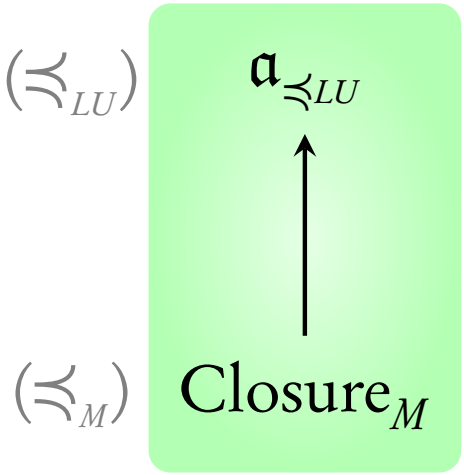
LU-bounds Behrmann et al'06

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

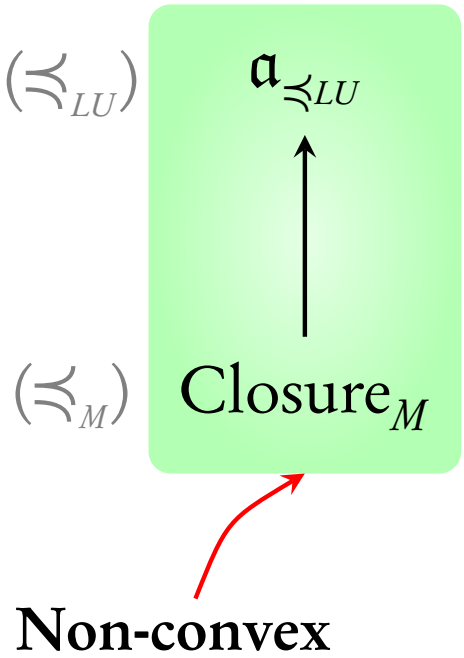
$$U(x) = 4,\ U(y) = 3$$

$$v \preceq_{LU} v'$$

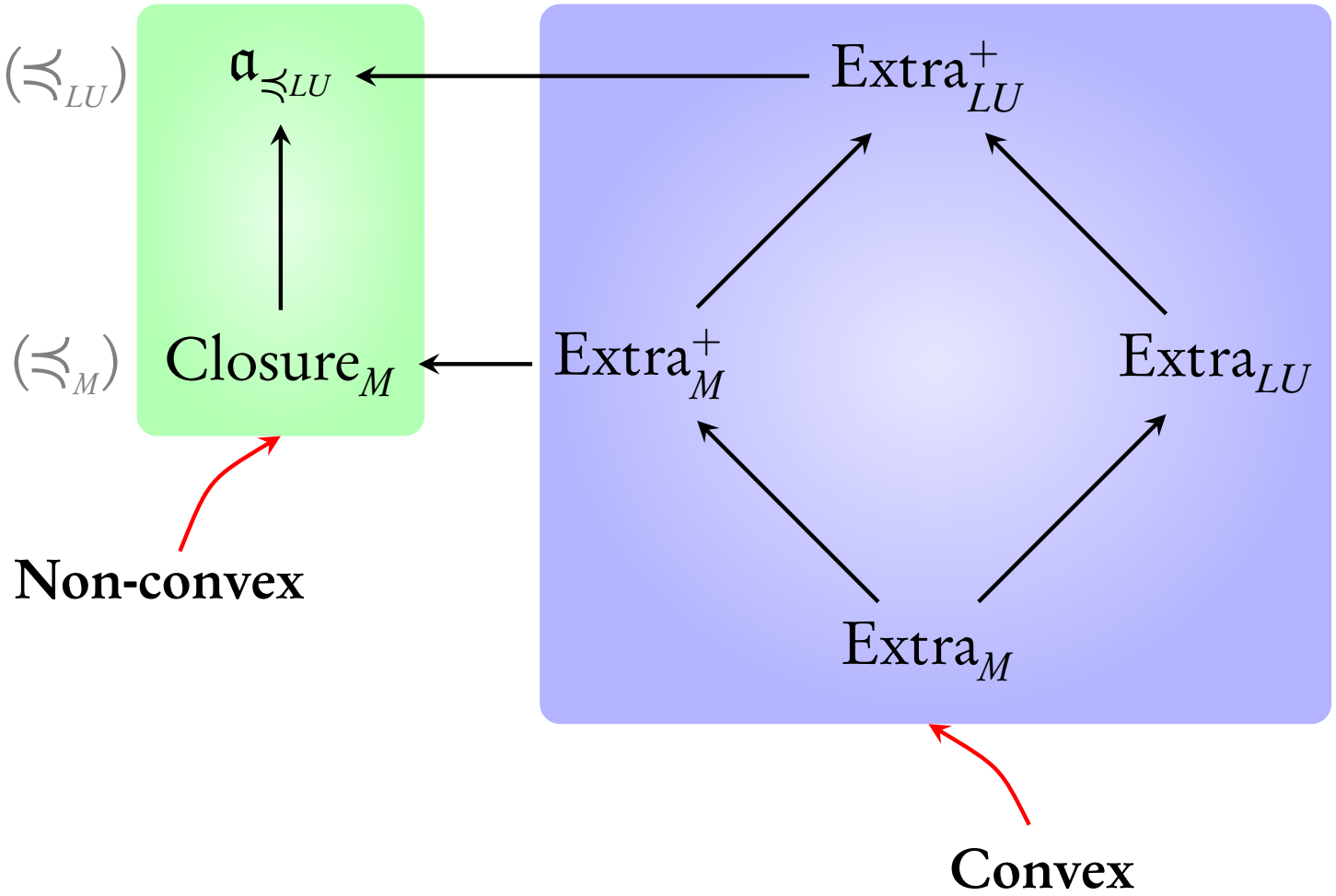
Abstractions in literature [Behrmann, Bouyer, Larsen, Pelanek'06]



Abstractions in literature [Behrmann, Bouyer, Larsen, Pelanek'06]



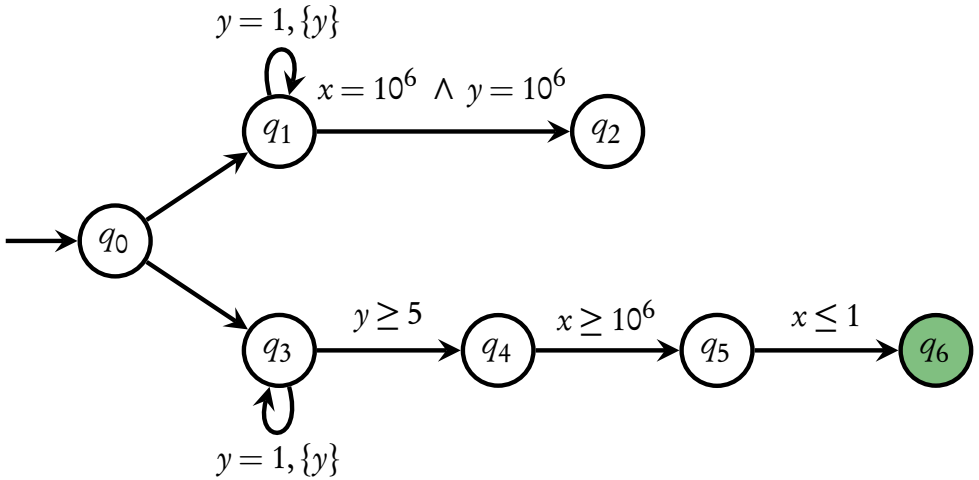
Abstractions in literature [Behrmann, Bouyer, Larsen, Pelanek'06]

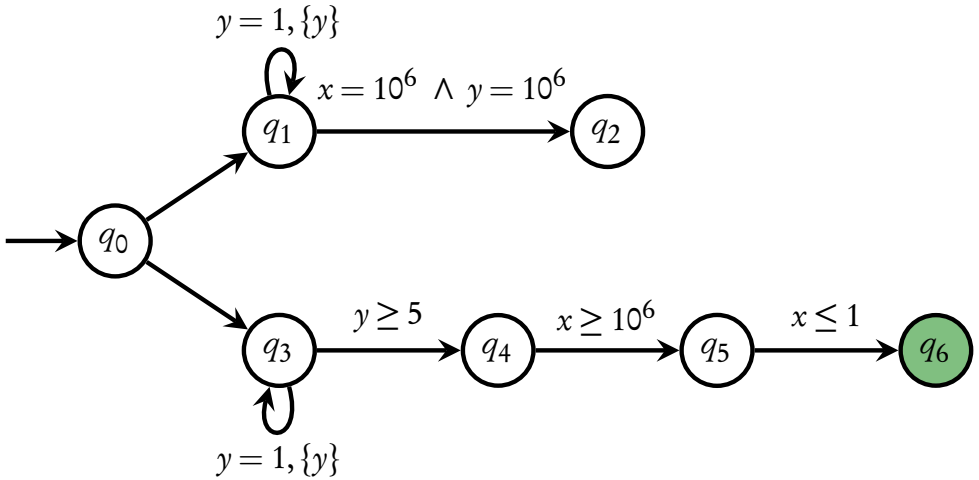


Only convex abstractions used in **implementations!**

Getting LU-bounds

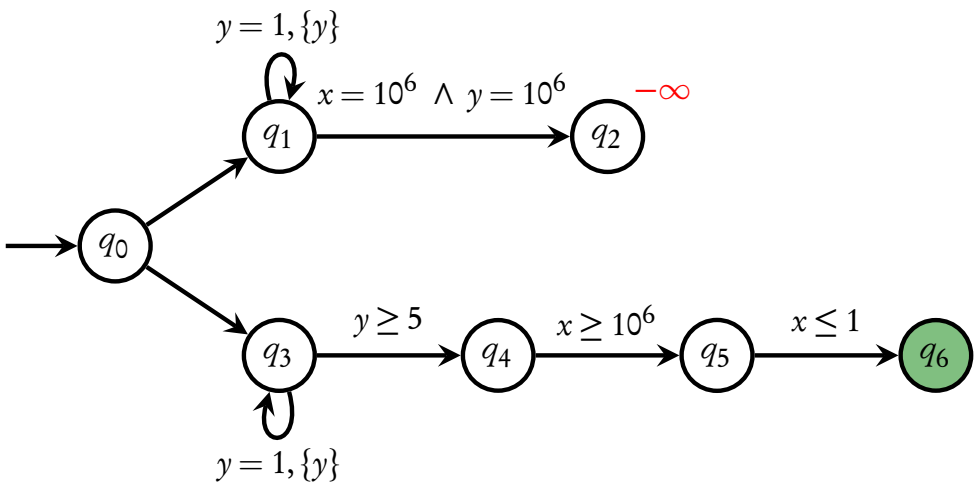
Smaller the LU bounds, **bigger** is the abstraction


$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$



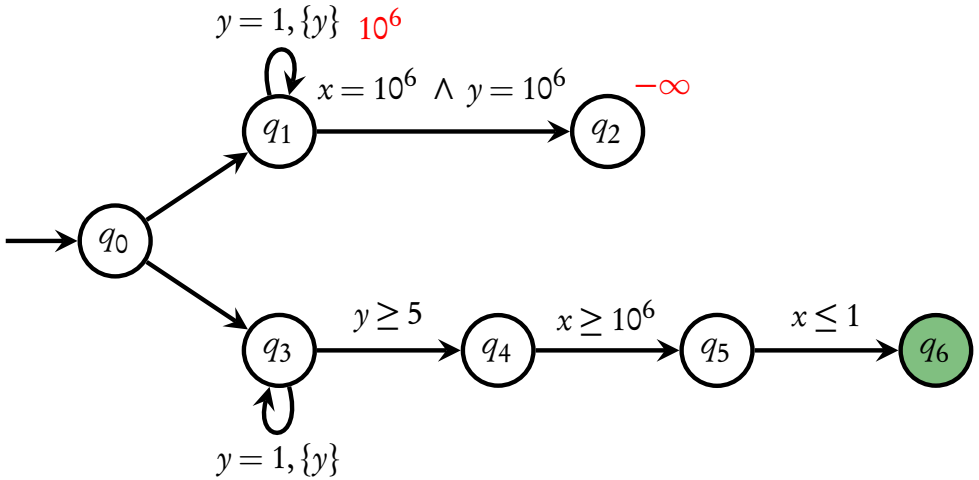
bounds by Static analysis

[Behrmann, Bouyer, Fleury, Larsen'03]



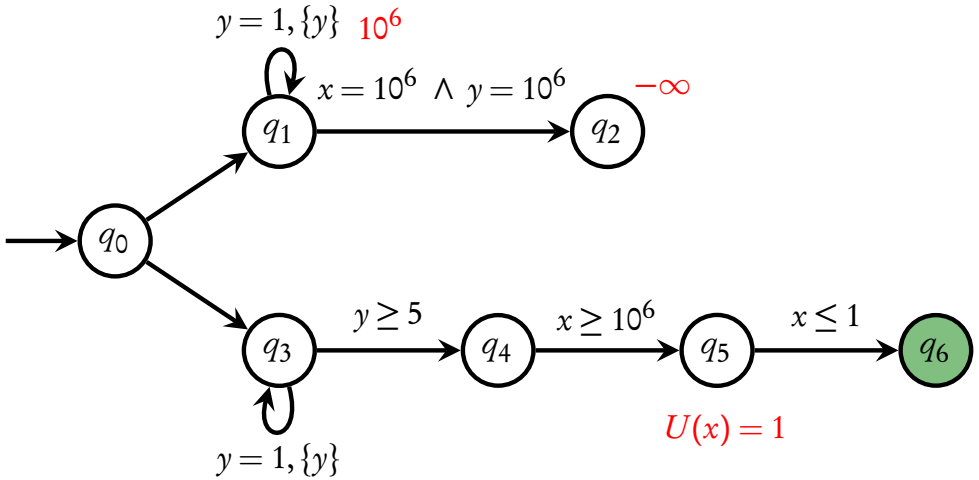
bounds by Static analysis

[Behrmann, Bouyer, Fleury, Larsen'03]



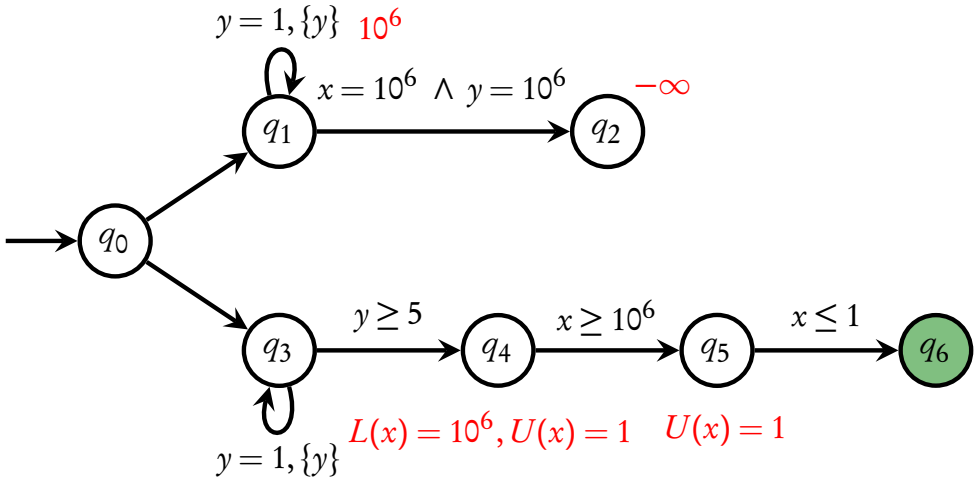
bounds by Static analysis

[Behrmann, Bouyer, Fleury, Larsen'03]



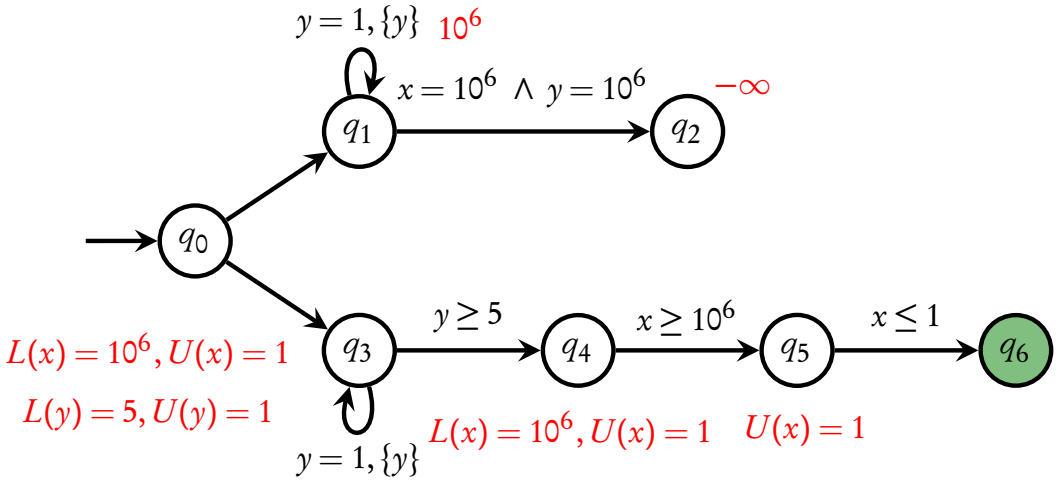
bounds by Static analysis

[Behrmann, Bouyer, Fleury, Larsen'03]



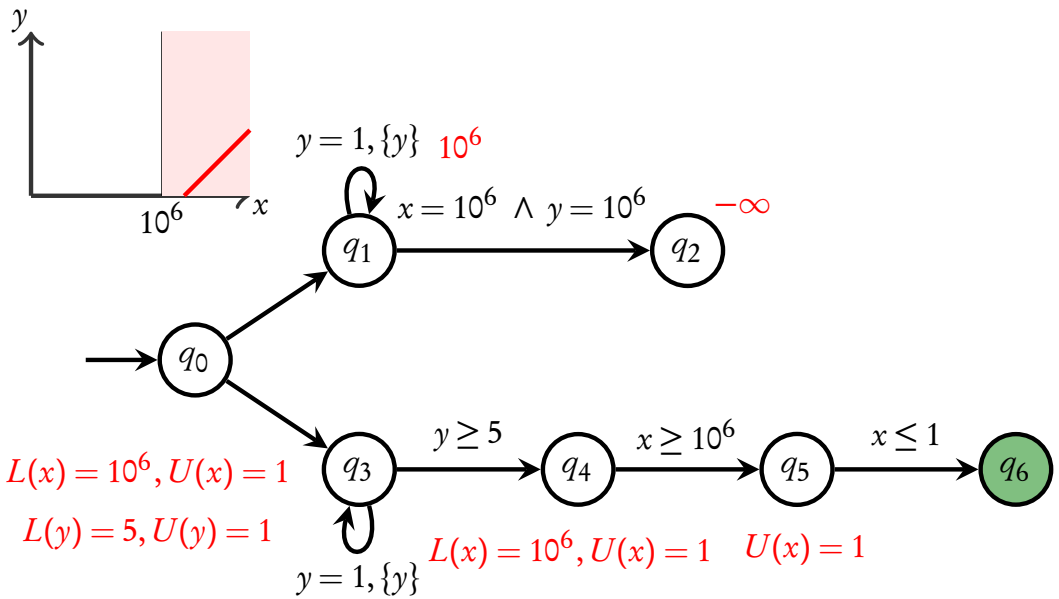
bounds by Static analysis

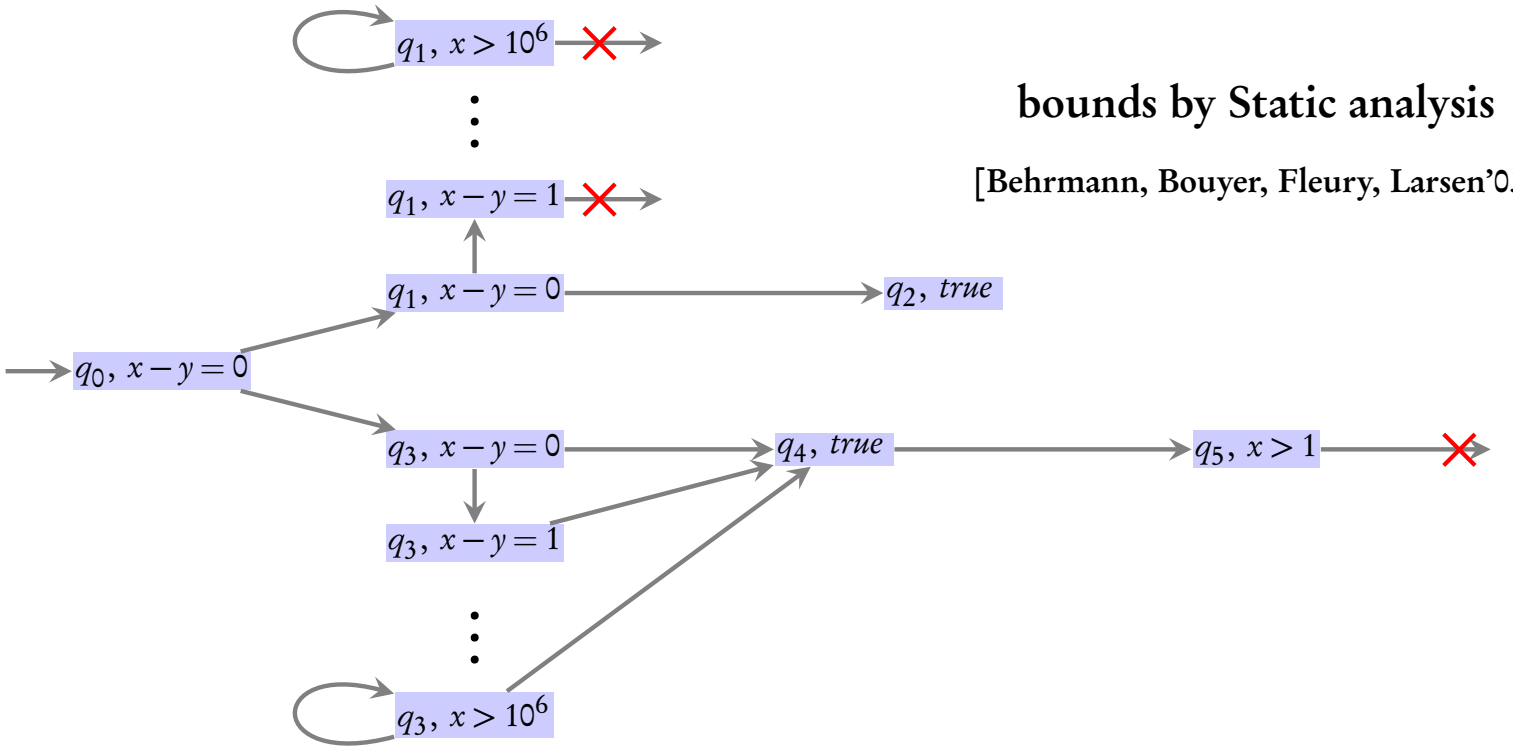
[Behrmann, Bouyer, Fleury, Larsen'03]



bounds by Static analysis

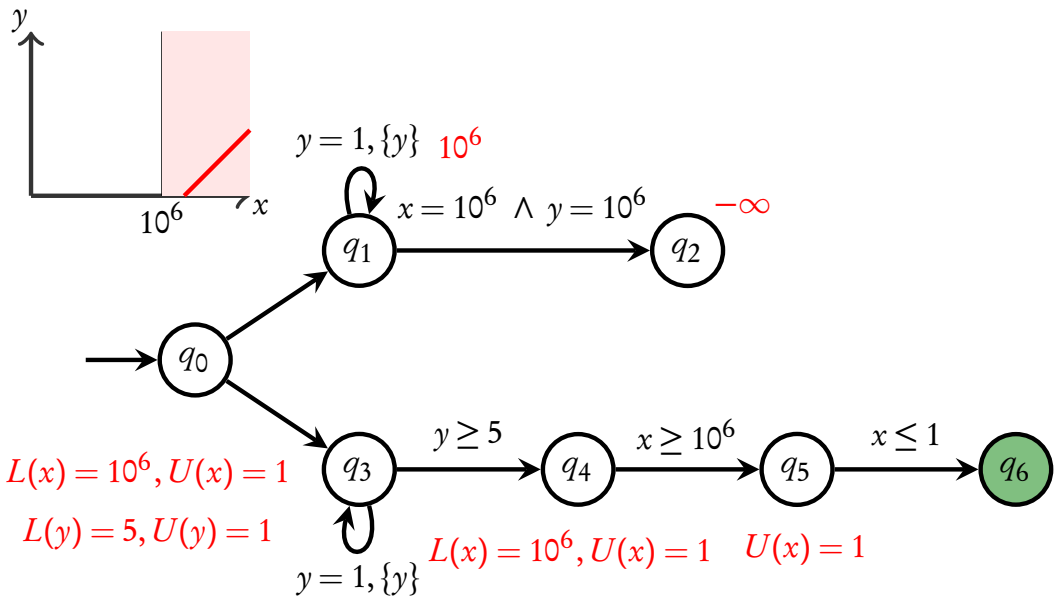
[Behrmann, Bouyer, Fleury, Larsen'03]

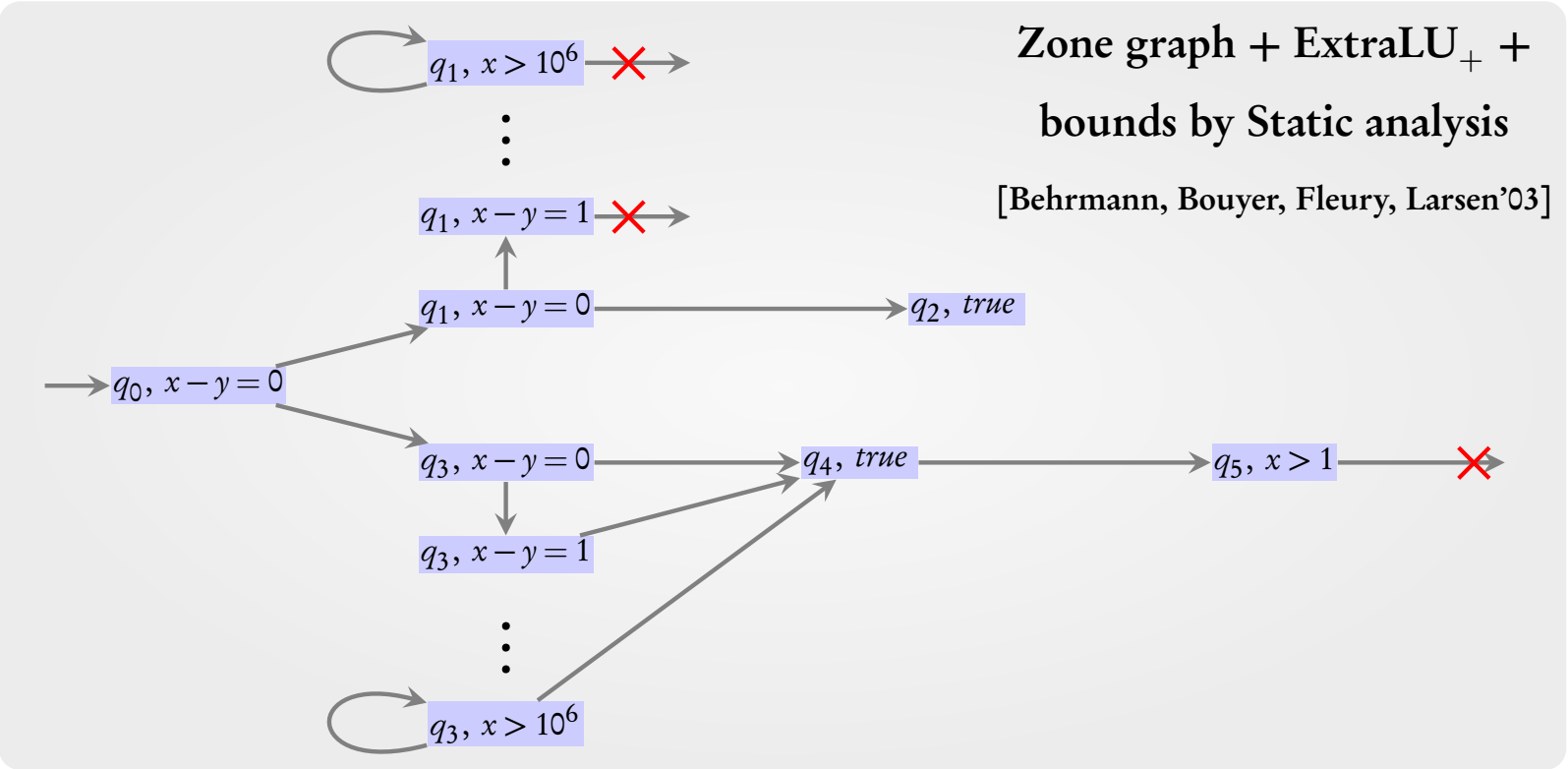




bounds by Static analysis

[Behrmann, Bouyer, Fleury, Larsen'03]





Zone graph + ExtraLU$_+$ + bounds by Static analysis

[Behrmann, Bouyer, Fleury, Larsen'03]

Reachability for timed automata

Standard algorithm: **covering** tree

Convex abstractions

Bounds by **static analysis**

Observation 1

Observation 2

Reachability for timed automata

Standard algorithm: **covering** tree

Convex abstractions

Bounds by **static analysis**

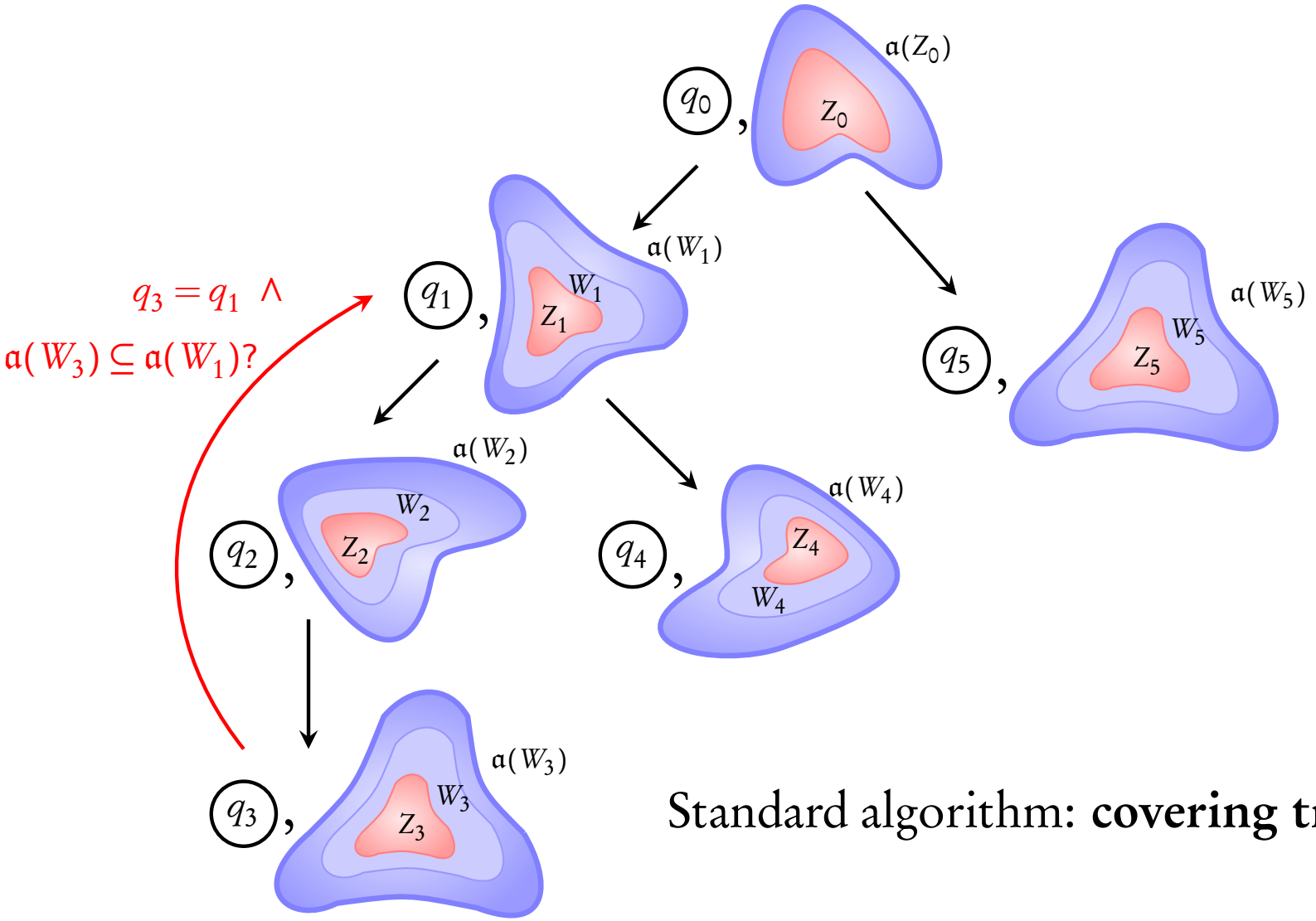
Observation 1

Non-convex abstractions

Observation 2

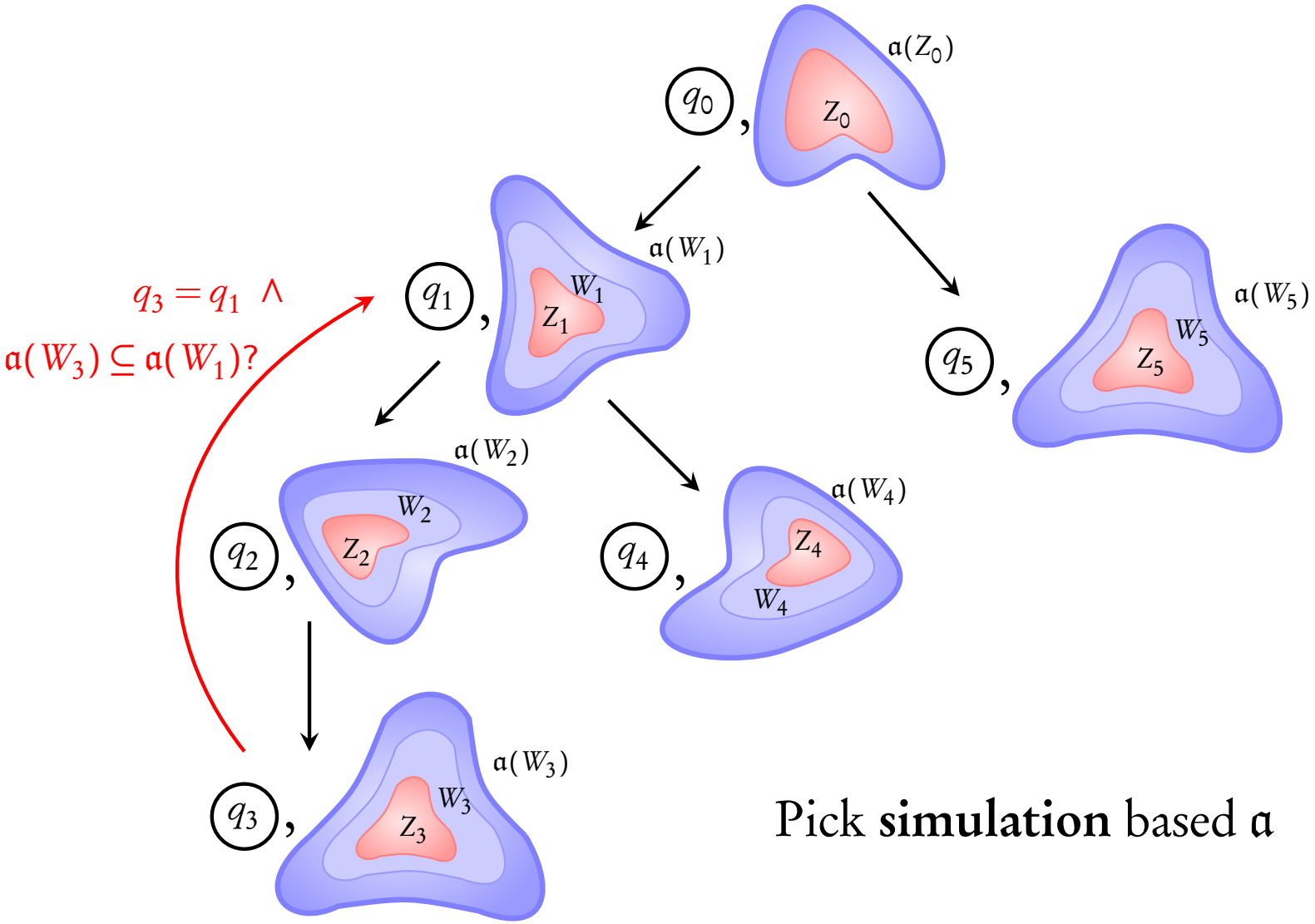
Step 1: We can use abstractions **without storing** them

Using non-convex abstractions



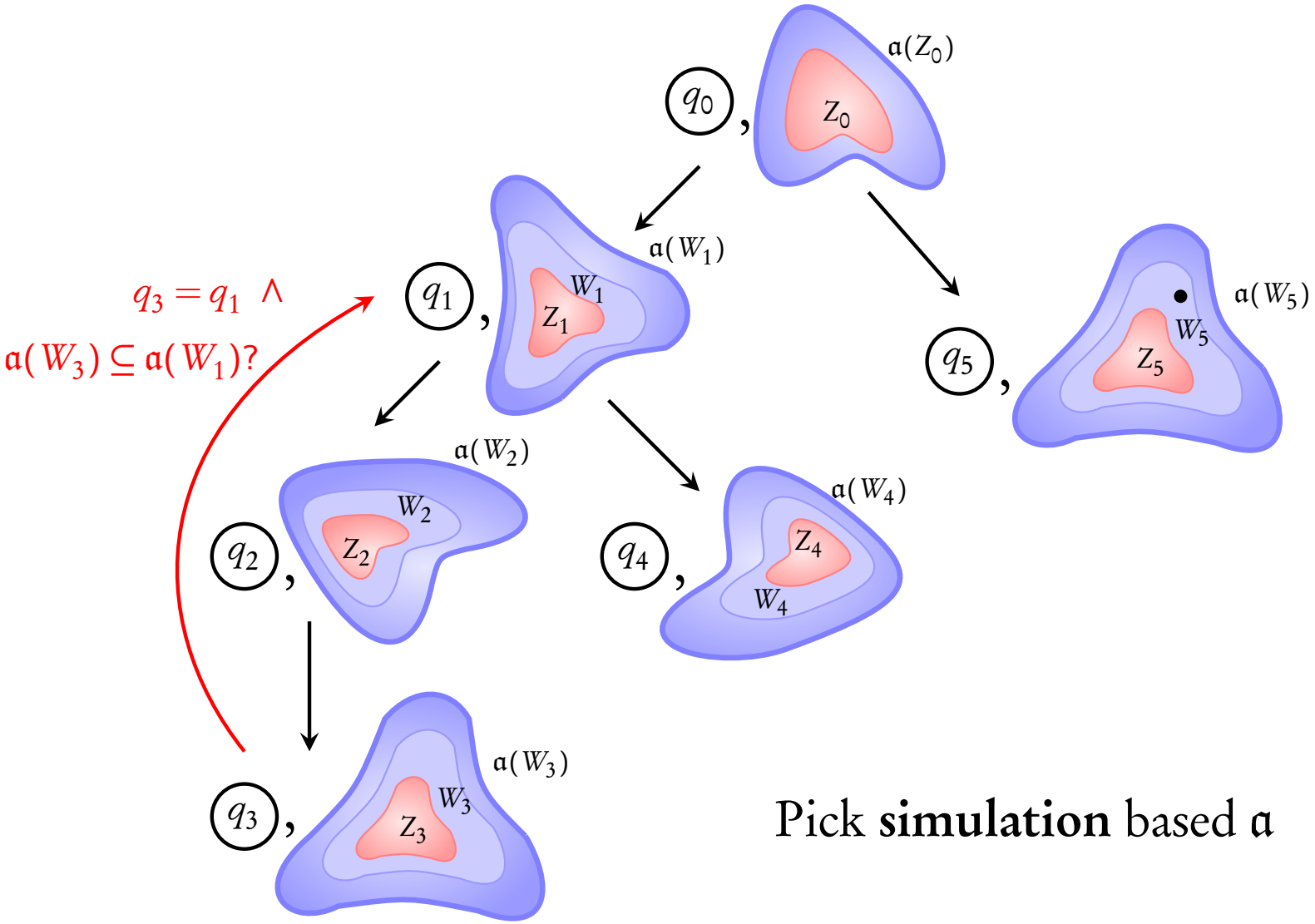
Standard algorithm: **covering tree**

Using non-convex abstractions



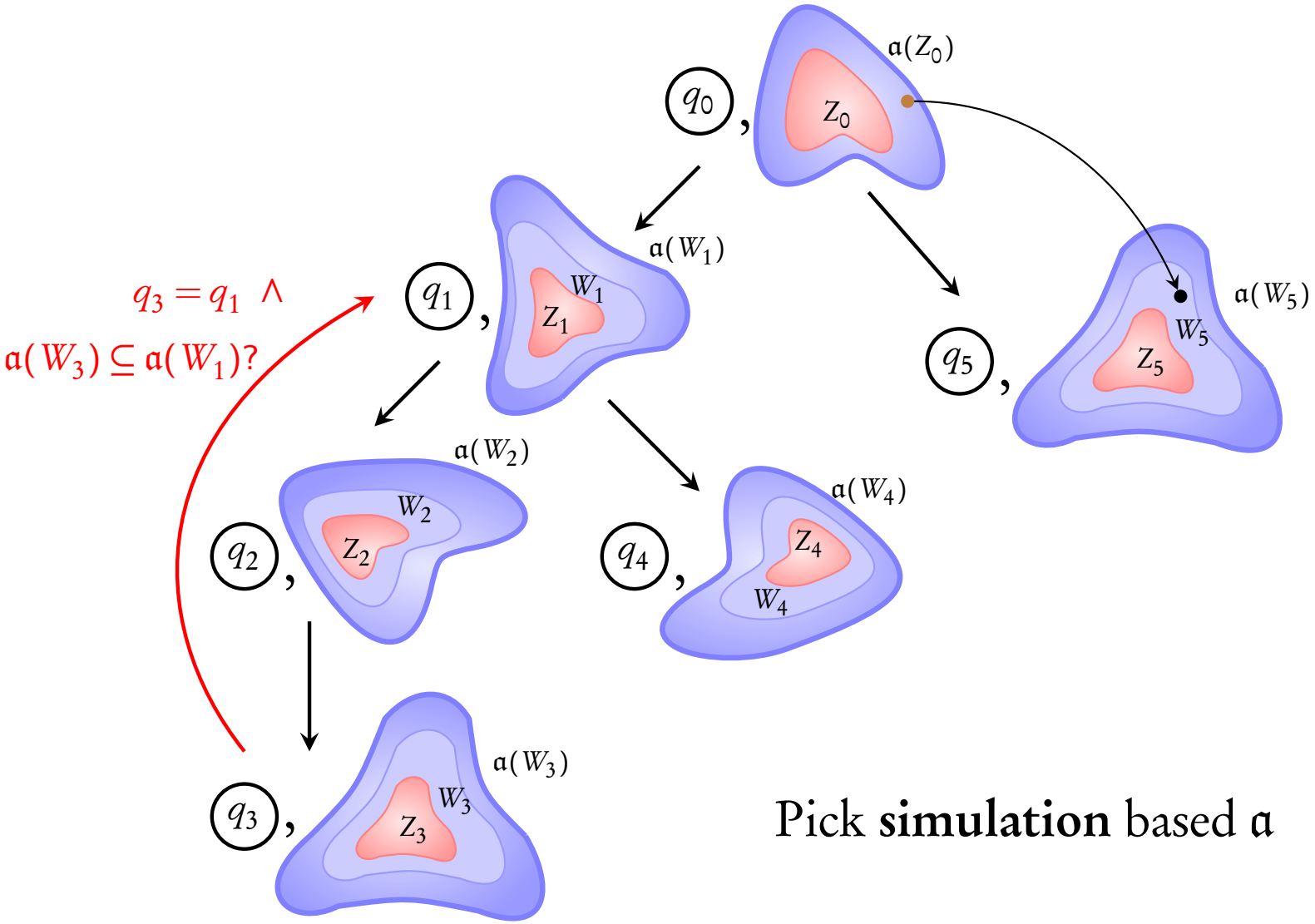
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



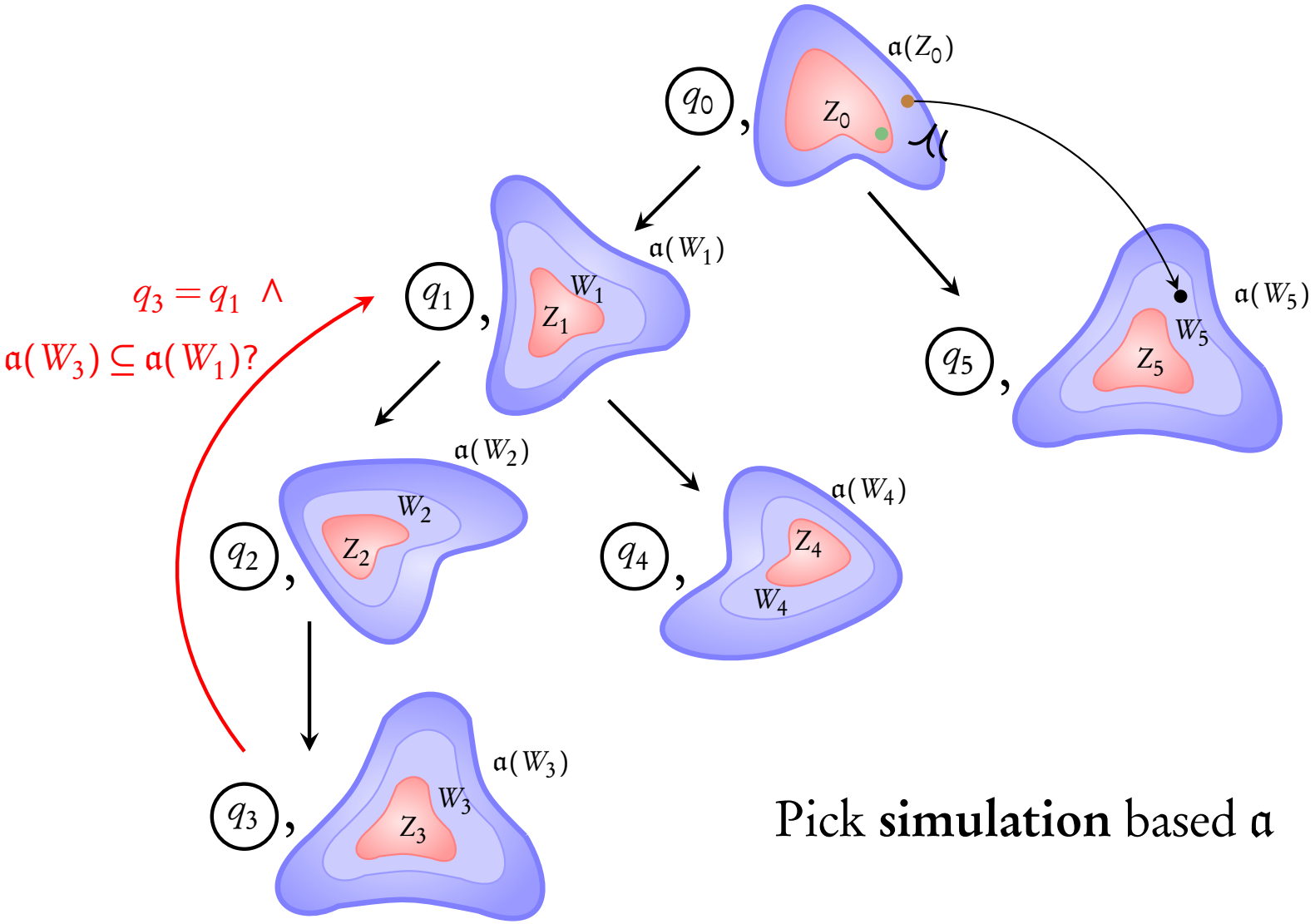
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



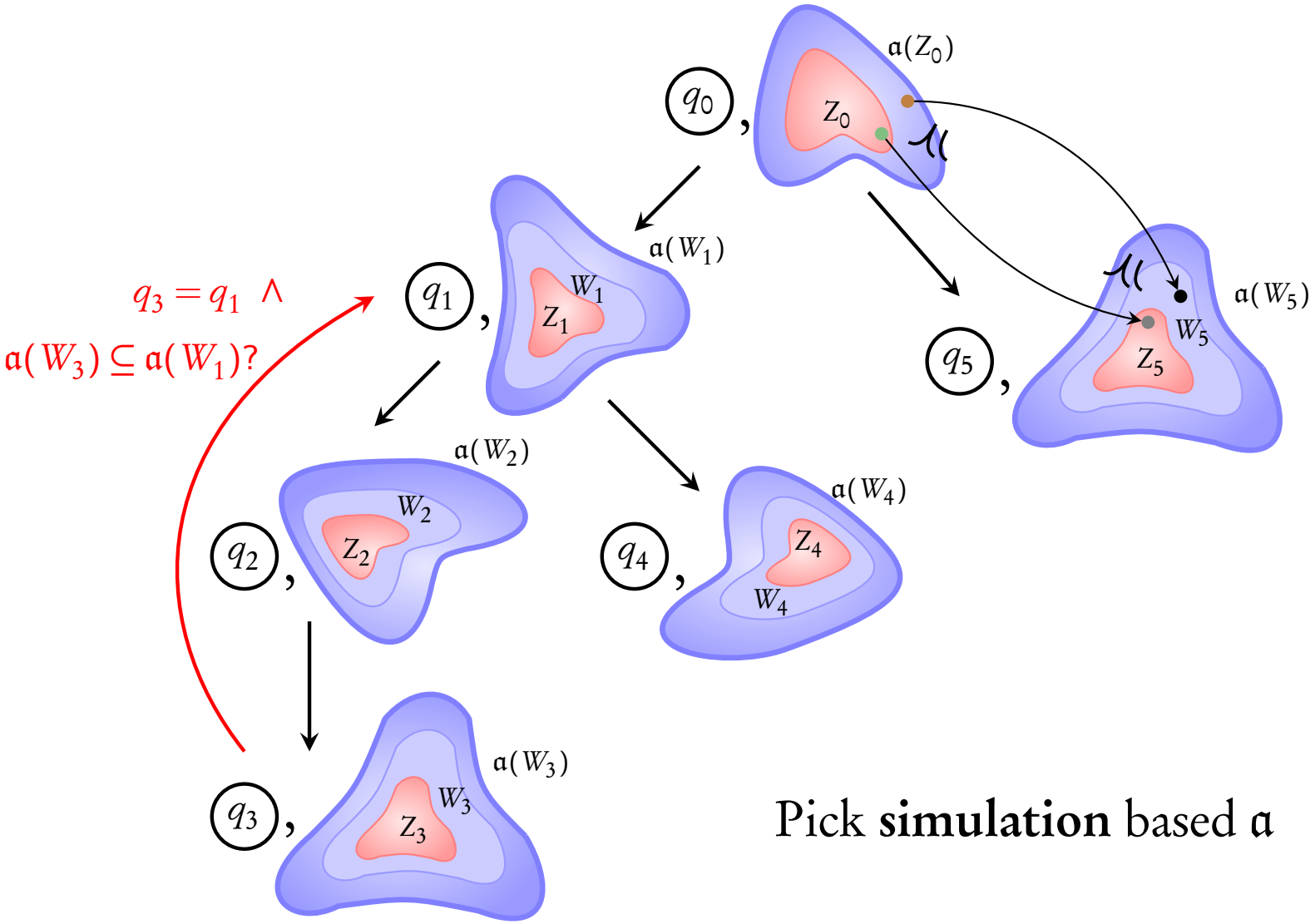
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



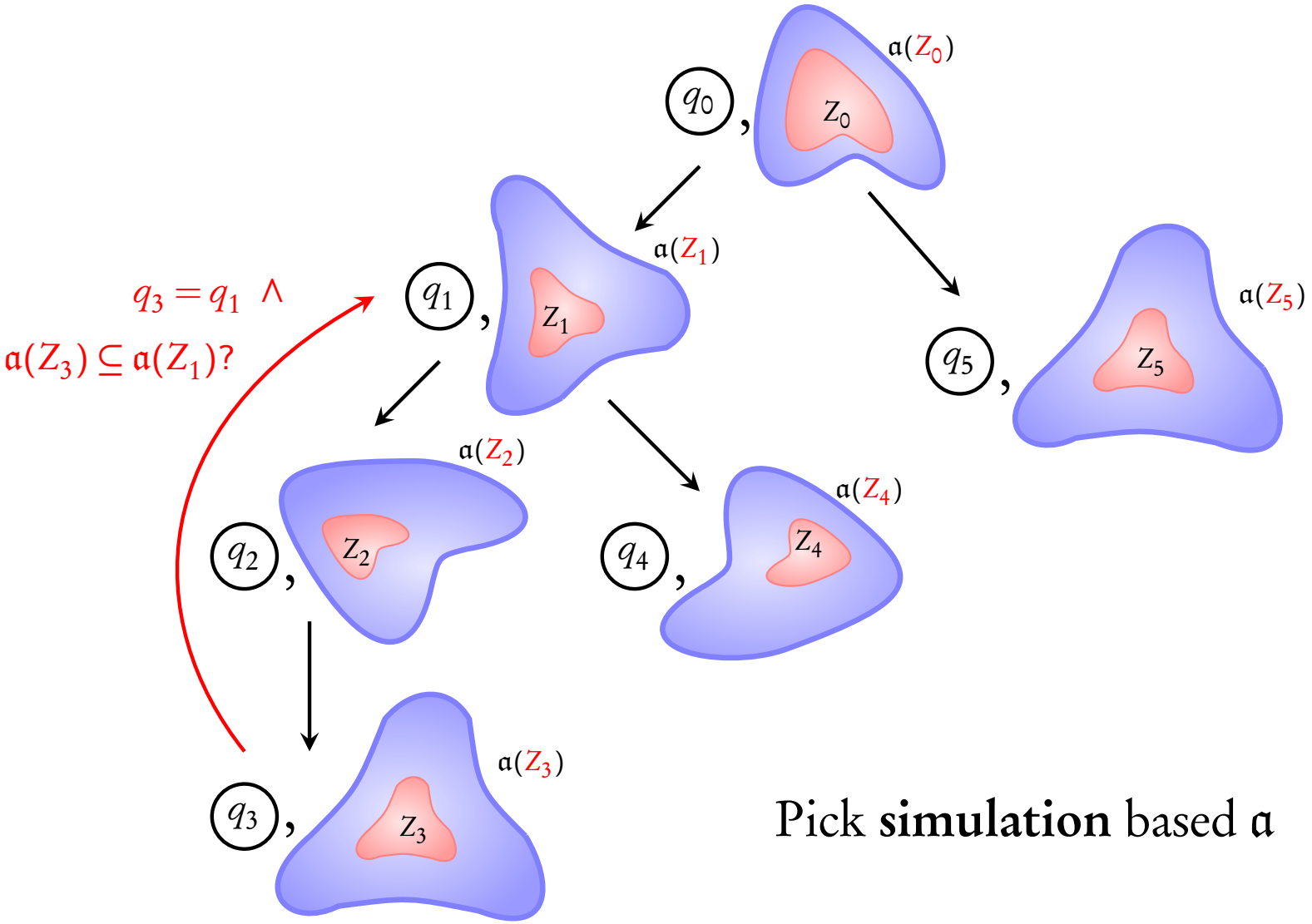
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



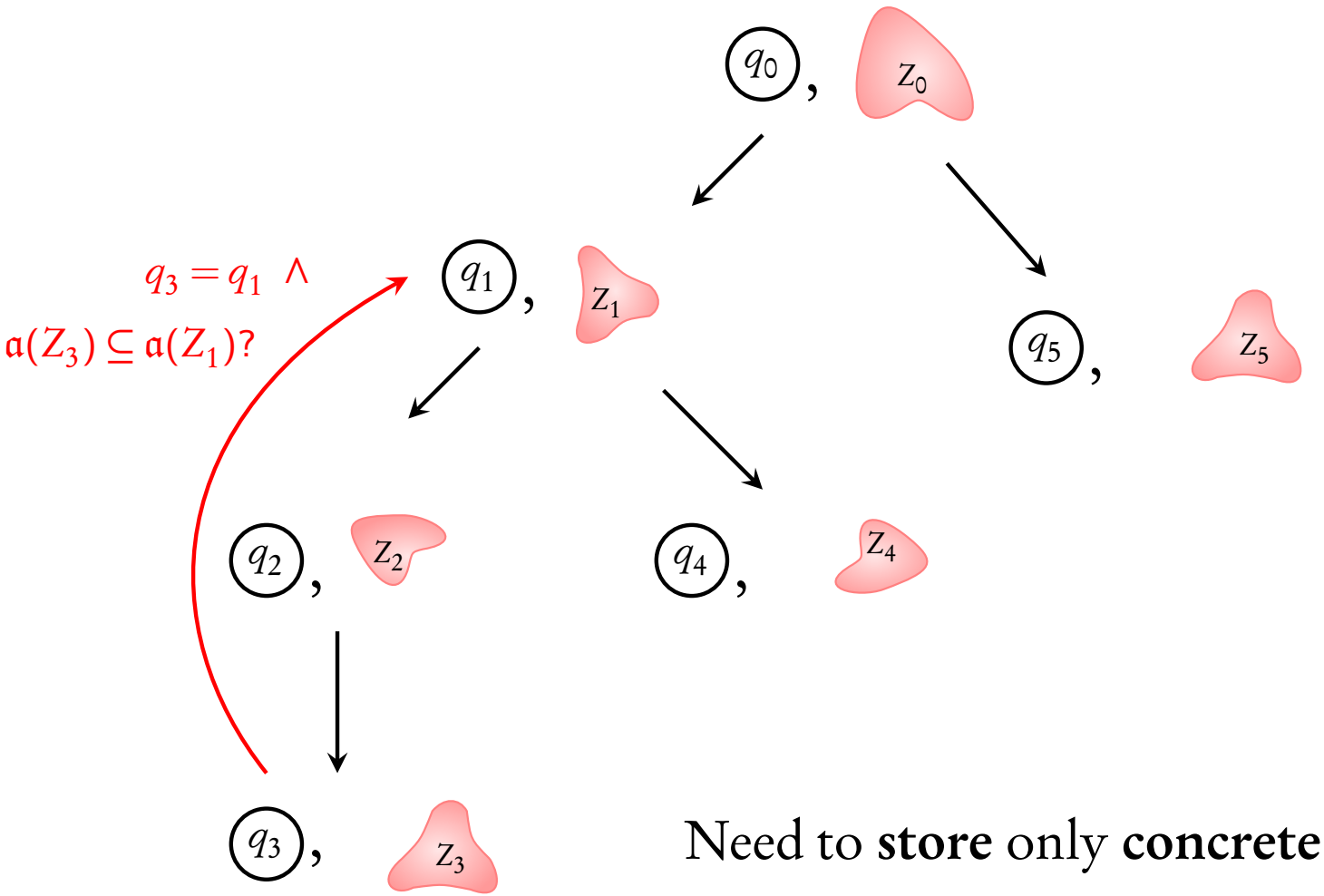
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



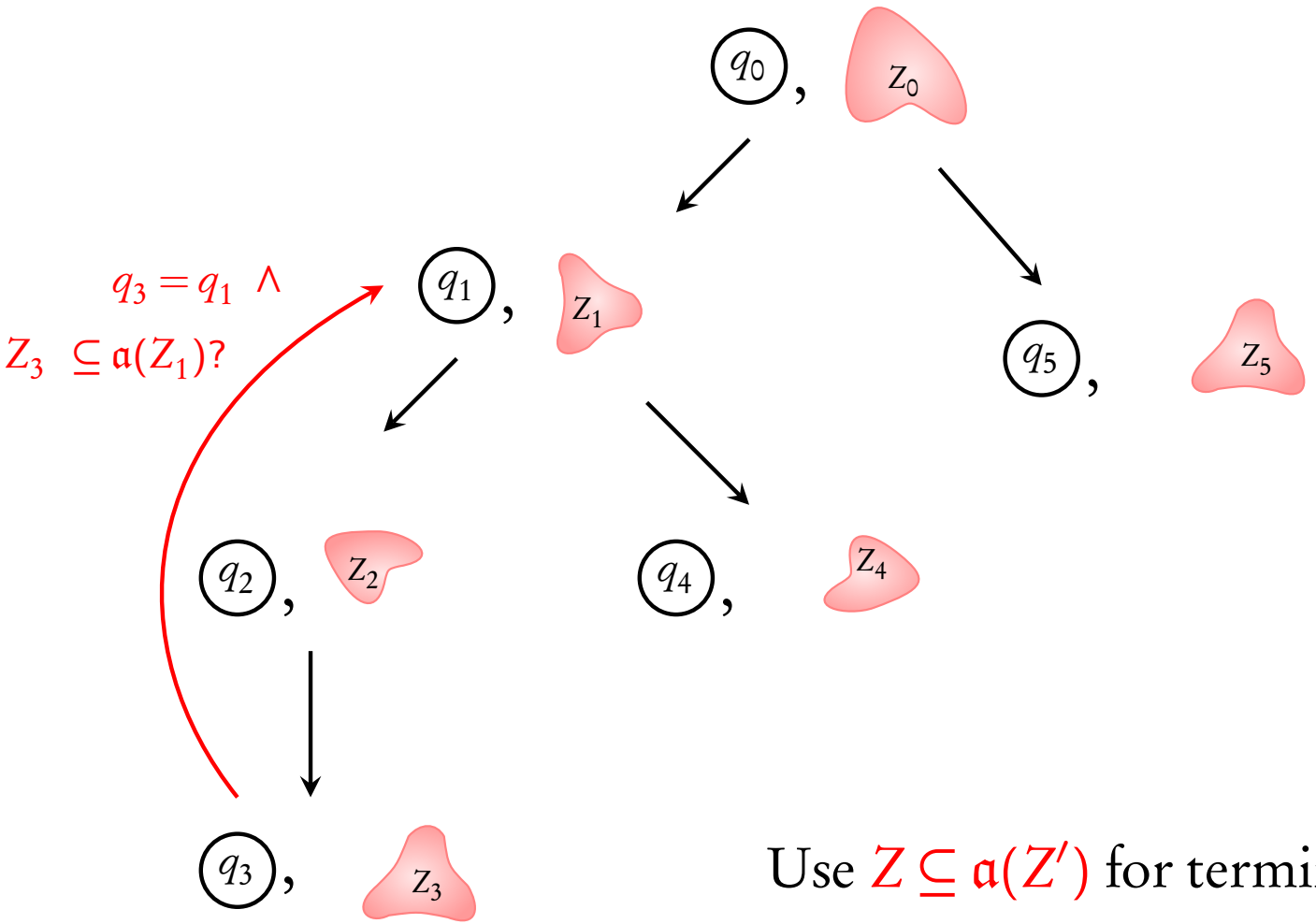
Pick **simulation** based $\mathfrak{a}$

Using non-convex abstractions



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

Using non-convex abstractions



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

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

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

Efficient inclusion testing

Main result

$Z \not\subseteq \mathfrak{a}_{\preccurlyeq_{LU}}(Z')$ if and only if there **exist 2 clocks** x, y s.t.

$$\mathbf{Proj}_{xy}(Z) \not\subseteq \mathfrak{a}_{\preccurlyeq_{LU}}(\mathbf{Proj}_{xy}(Z'))$$

Efficient inclusion testing

Main result

$Z \not\subseteq \mathfrak{a}_{\preccurlyeq_{LU}}(Z')$ if and only if there **exist 2 clocks** x, y s.t.

$$\mathbf{Proj}_{xy}(Z) \not\subseteq \mathfrak{a}_{\preccurlyeq_{LU}}(\mathbf{Proj}_{xy}(Z'))$$

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

Efficient inclusion testing

Main result

$Z \not\subseteq \mathfrak{a}_{\preccurlyeq_{LU}}(Z')$ if and only if there **exist 2 clocks** x, y s.t.

$$\mathbf{Proj}_{xy}(Z) \not\subseteq \mathfrak{a}_{\preccurlyeq_{LU}}(\mathbf{Proj}_{xy}(Z'))$$

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

Same complexity as $Z \subseteq Z'$!

Efficient inclusion testing

Main result

$Z \not\subseteq \mathfrak{a}_{\preccurlyeq_{LU}}(Z')$ if and only if there **exist 2 clocks** x, y s.t.

$$\mathbf{Proj}_{xy}(Z) \not\subseteq \mathfrak{a}_{\preccurlyeq_{LU}}(\mathbf{Proj}_{xy}(Z'))$$

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

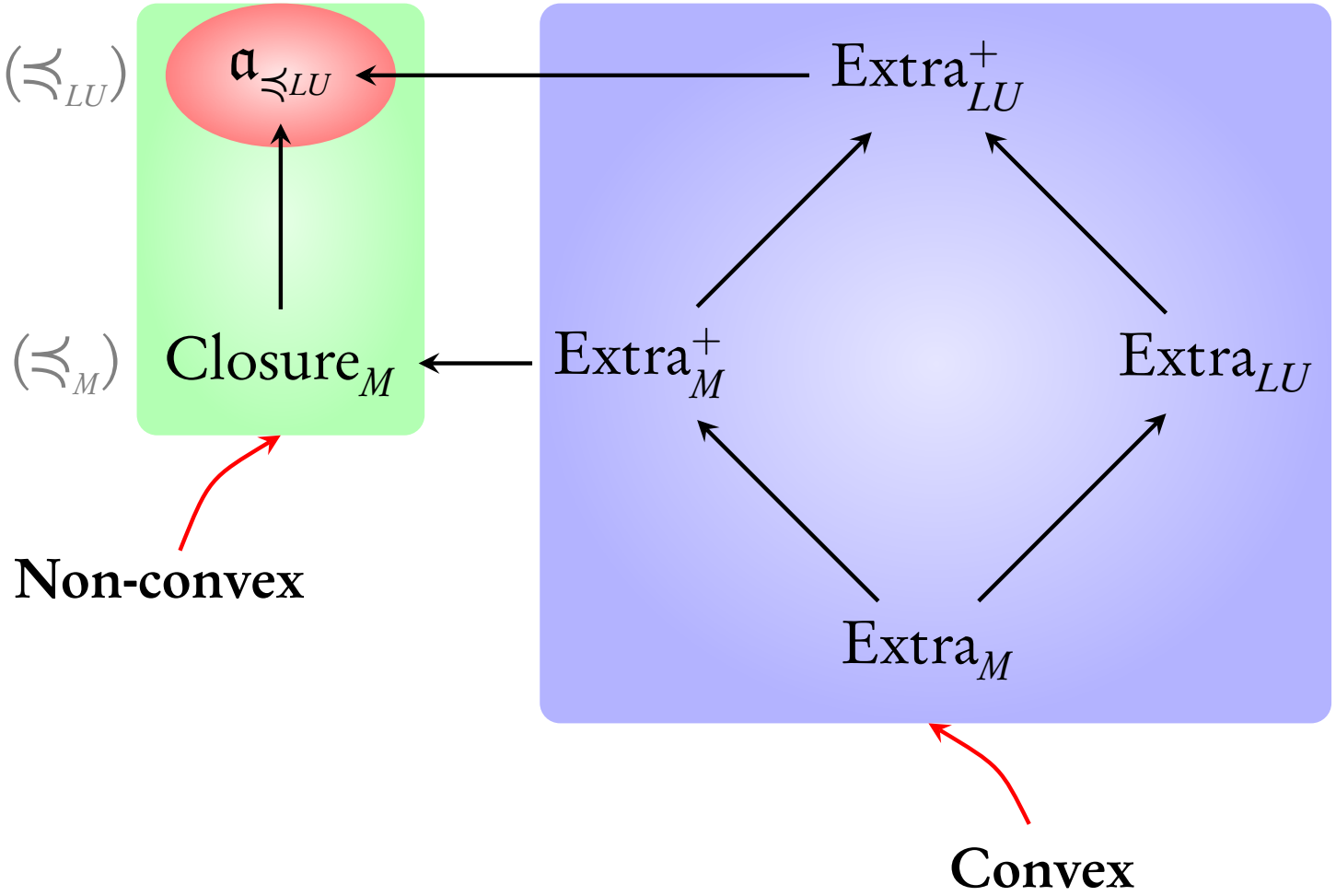
Same complexity as $Z \subseteq Z'$!

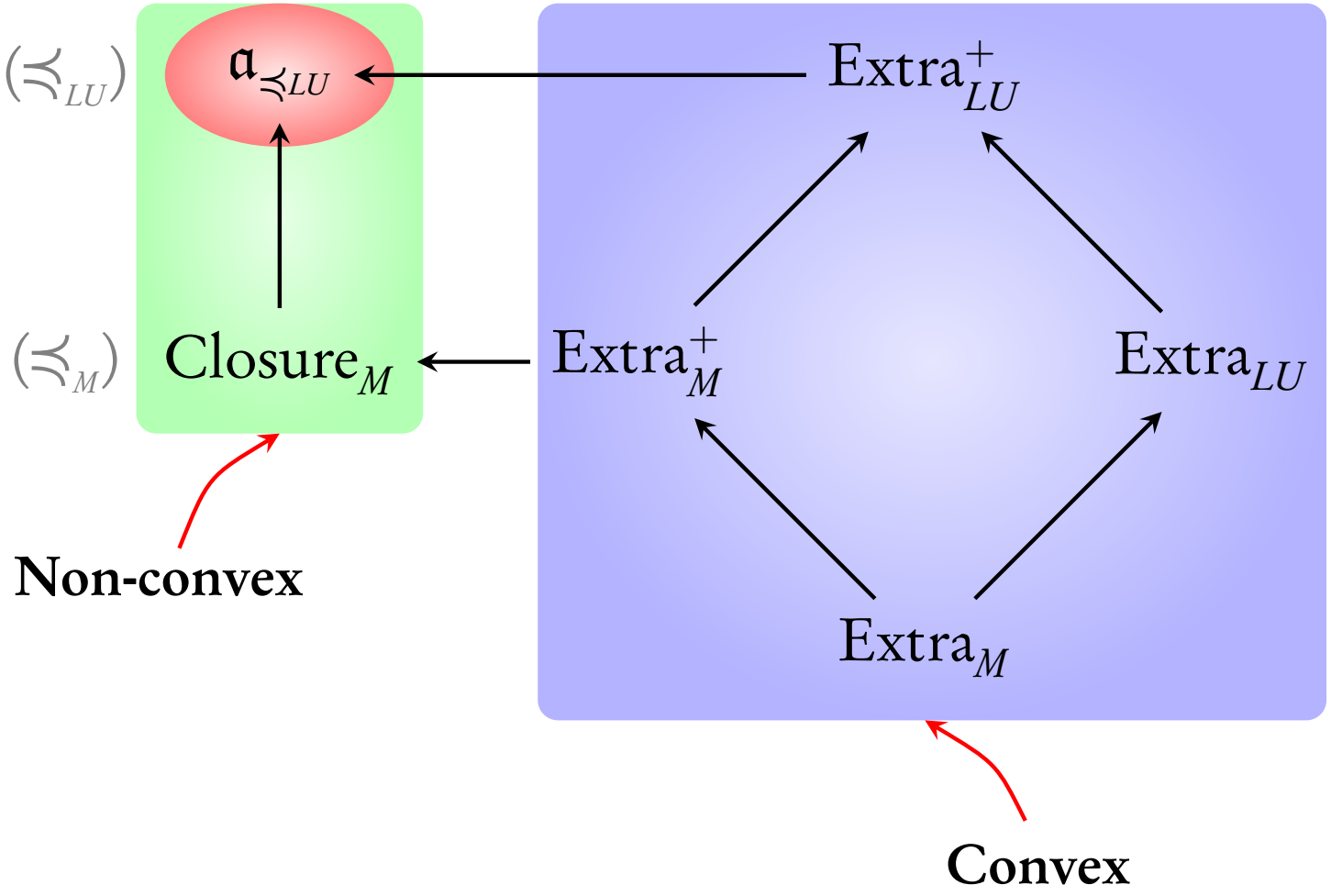
Slightly modified comparison works!

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

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

$\Rightarrow$ **new algorithm** for reachability


$(\precsim_{LU})$
$\mathfrak{a}_{\precsim LU}$
Extra^{+}_{LU}
$(\precsim_{M})$
$\mathrm{Closure}_{M}$
Extra^{+}_{M}
Extra_{LU}
Extra_{M}
Non-convex
Convex



Question: Can we do better than $\mathfrak{a}_{\precsim_{LU}}$?

Optimality

LU-automata: automata with guards **determined by** L and U

Theorem

The $\alpha_{\precsim_{LU}}$ abstraction is the **coarsest abstraction** that is **sound** and **complete** for all LU-automata.

Reachability for timed automata

Standard algorithm: **covering** tree

Convex abstractions

Bounds by **static analysis**

Observation 1

Non-convex abstractions

Optimality

Observation 2

Reachability for timed automata

Standard algorithm: **covering** tree

Convex abstractions

Bounds by **static analysis**

Observation 1

Non-convex abstractions

Optimality

Observation 2

Better LU-bounds

Reachability for timed automata

Standard algorithm: **covering** tree

Convex abstractions

Bounds by **static analysis**

Observation 1

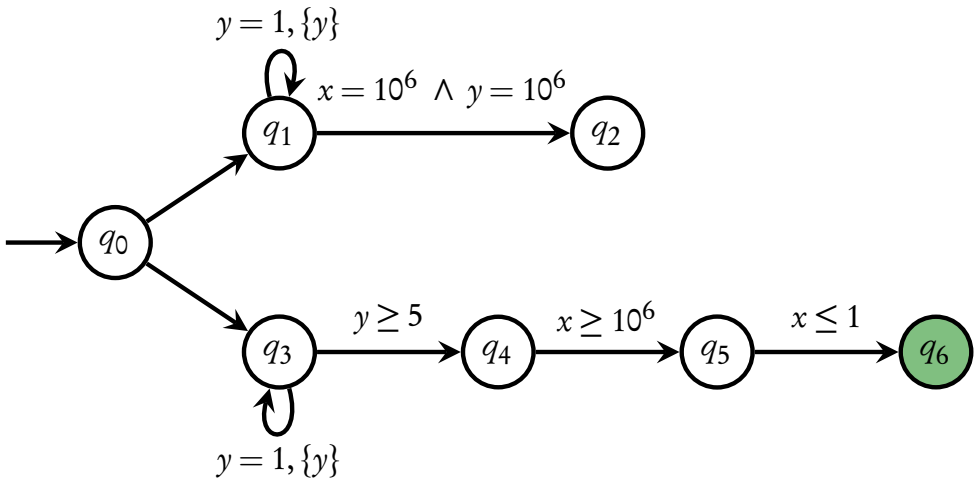
Non-convex abstractions

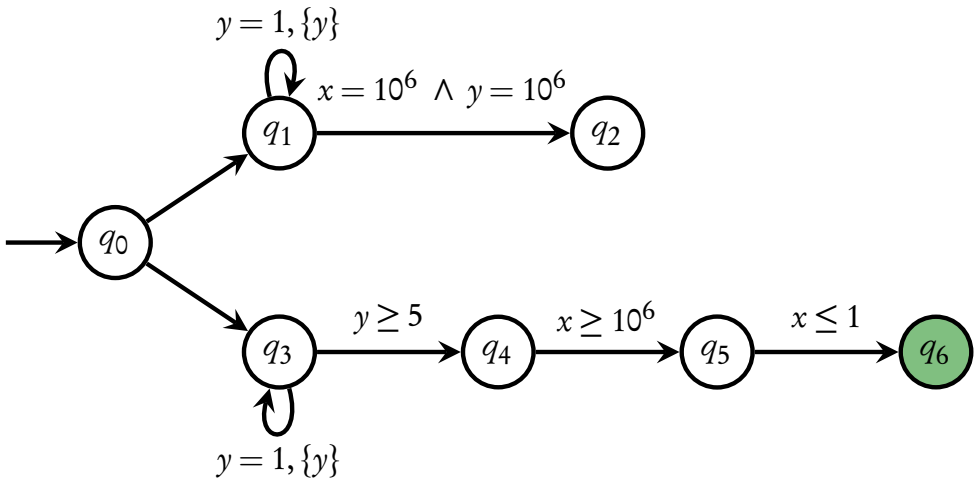
Optimality

Observation 2

Better LU-bounds

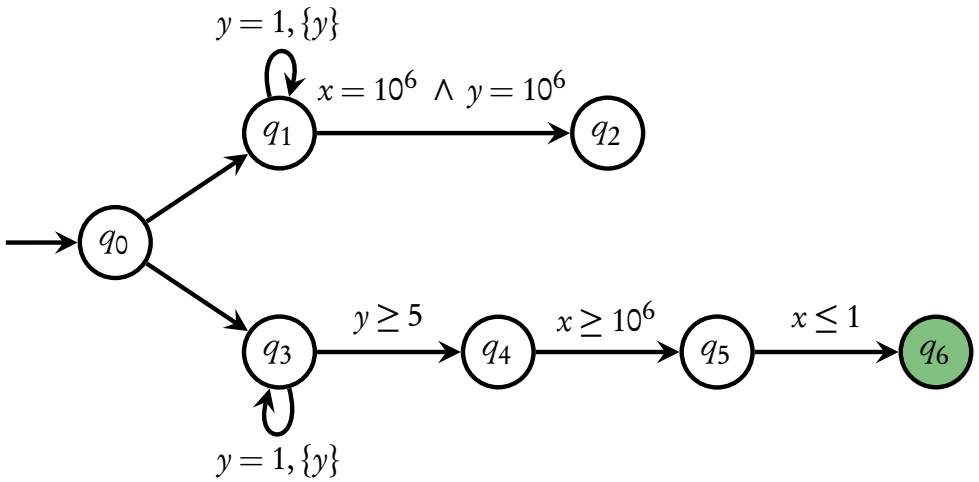
Smaller the LU-bounds, **bigger** is the $\mathfrak{a}_{\preccurlyeq_{LU}}$ abstraction

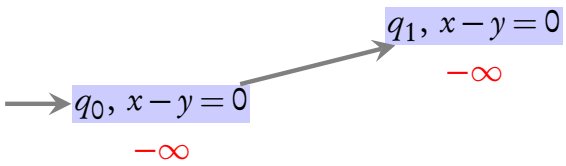

$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

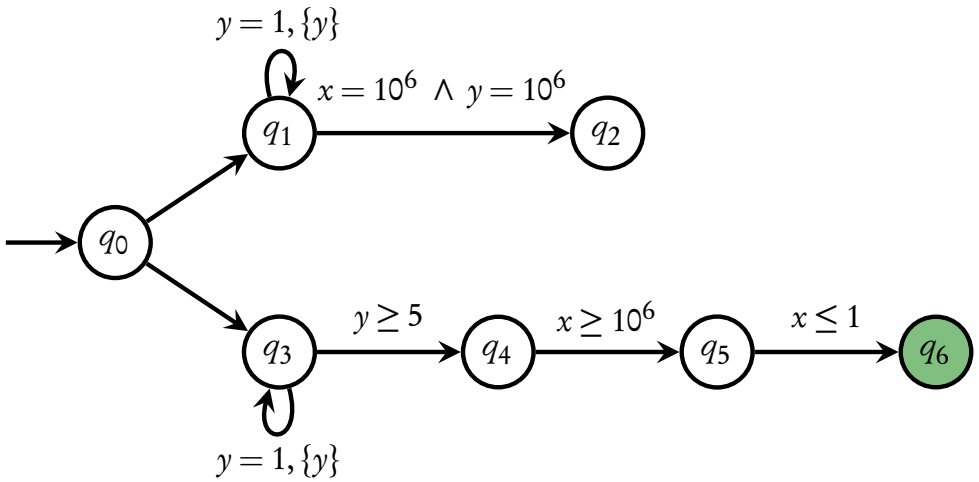


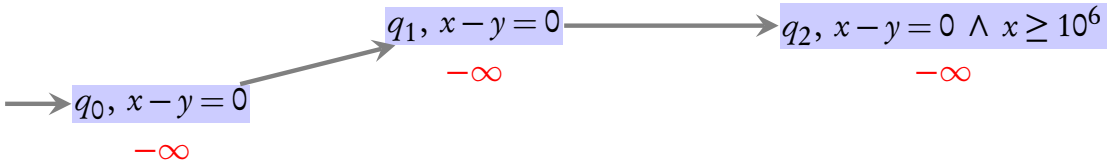
$\longrightarrow q_0,\ x - y = 0$

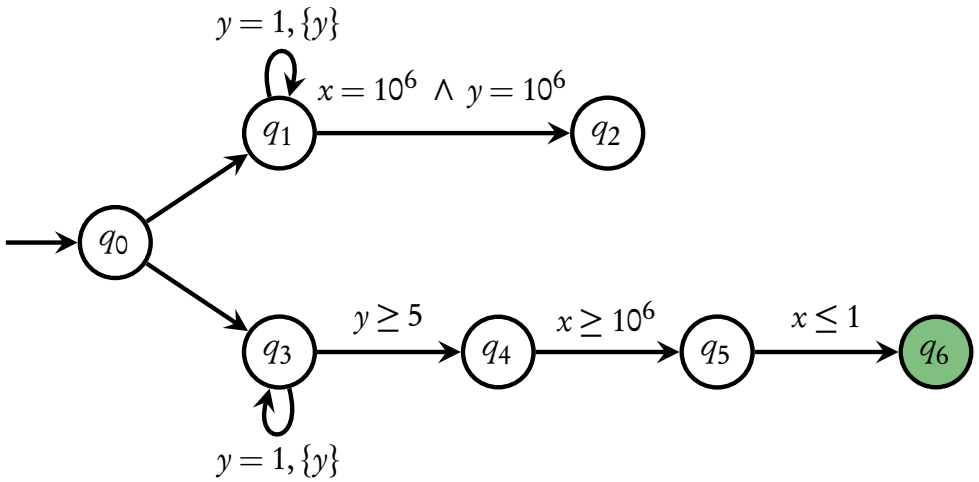
$-\infty$

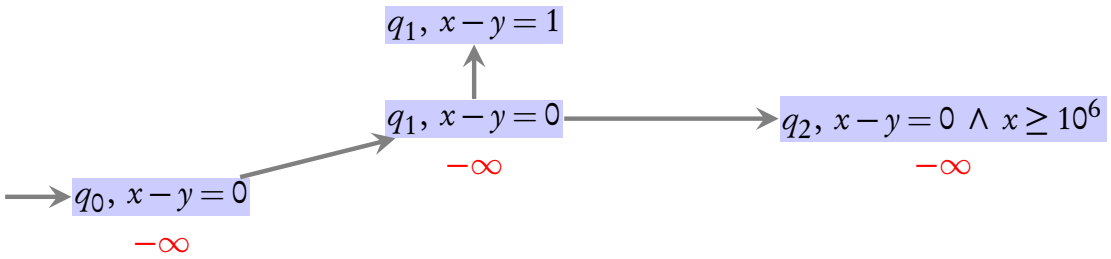

$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

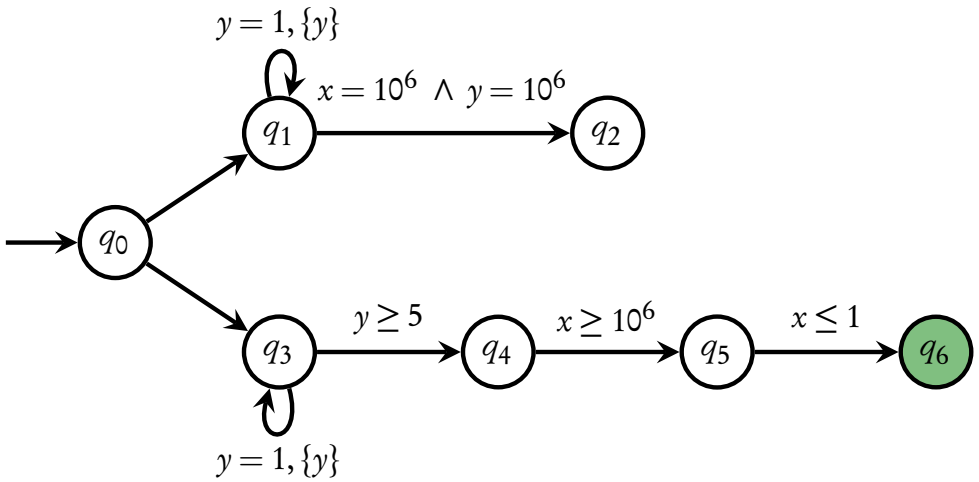

$q_1, \ x - y = 0$
$-\infty$
$q_0, \ x - y = 0$
$-\infty$

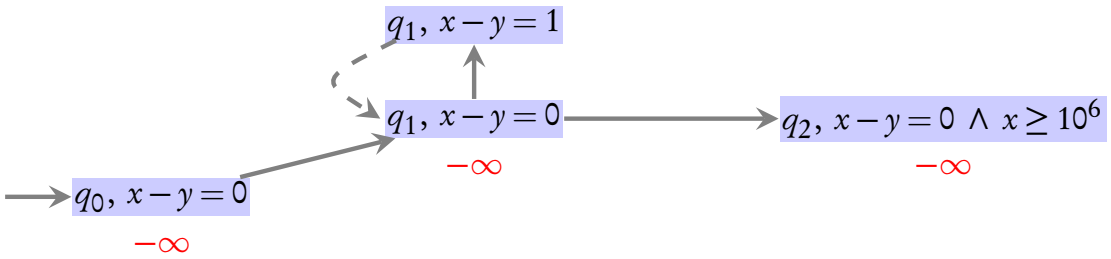

$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

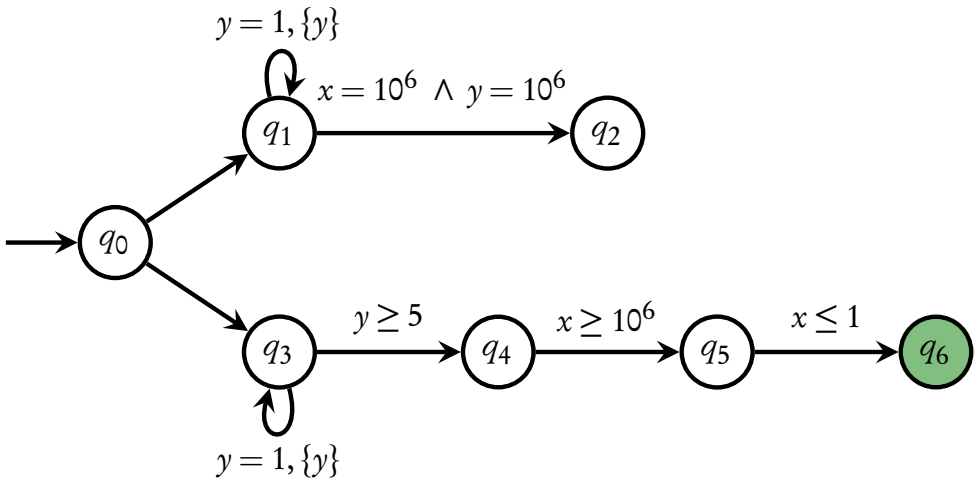

$q_0, \ x - y = 0$
$-\infty$
$q_1, \ x - y = 0$
$-\infty$
$q_2, \ x - y = 0 \ \wedge \ x \geq 10^6$
$-\infty$


$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

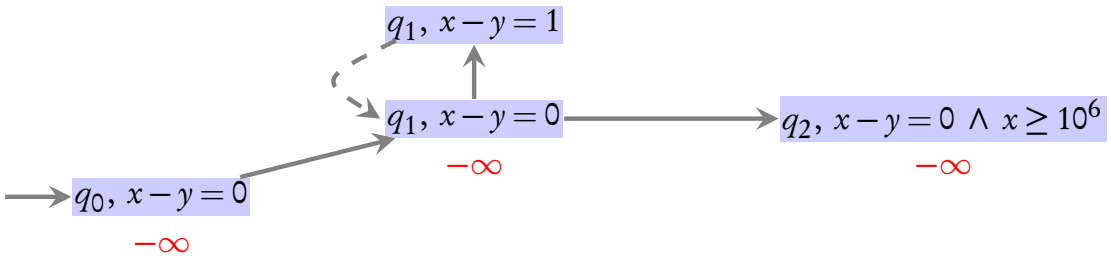

$q_1, \ x - y = 1$
$q_1, \ x - y = 0$
$-\infty$
$q_2, \ x - y = 0 \ \wedge \ x \geq 10^6$
$-\infty$
$q_0, \ x - y = 0$
$-\infty$

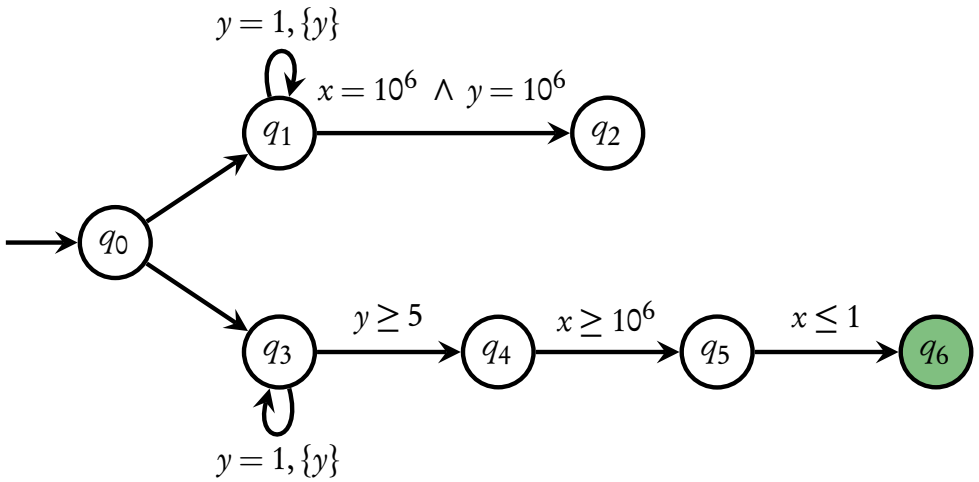

$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$


$q_1, \ x - y = 1$
$q_1, \ x - y = 0$
$-\infty$
$q_2, \ x - y = 0 \ \wedge \ x \geq 10^6$
$-\infty$
$q_0, \ x - y = 0$
$-\infty$

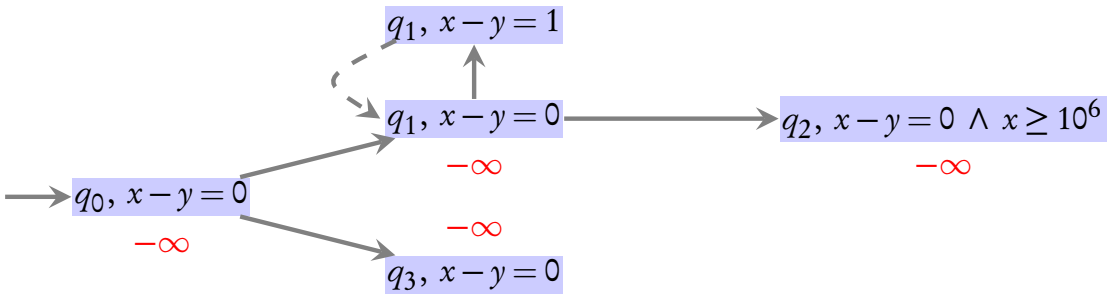


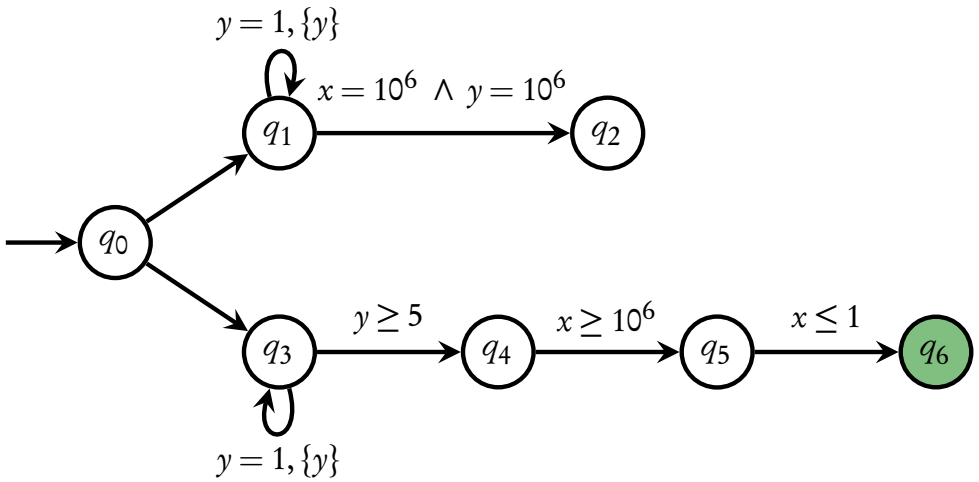
No disabled edge ⇒ all states seen ⇒ no need for bounds!



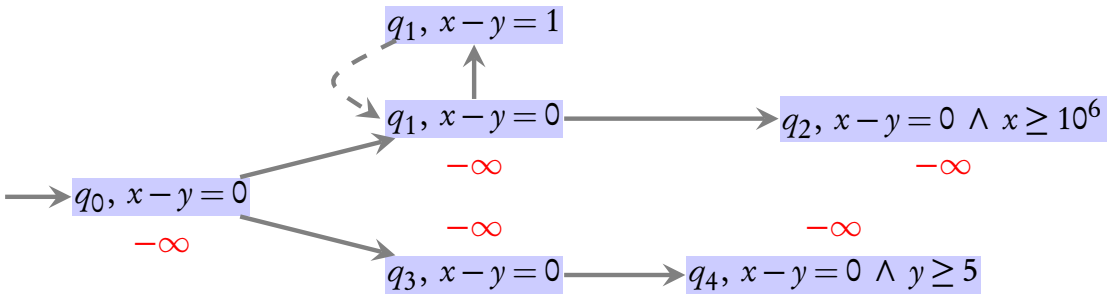


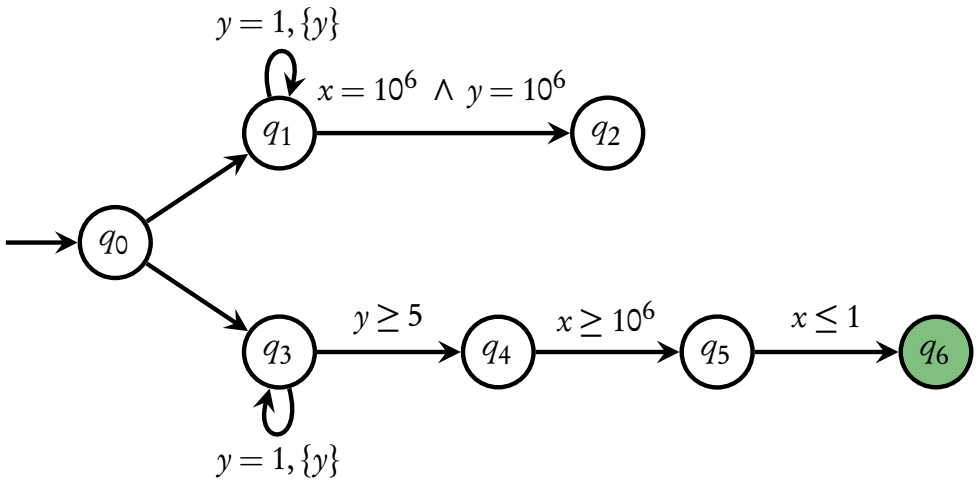
No disabled edge ⇒ all states seen ⇒ no need for bounds!



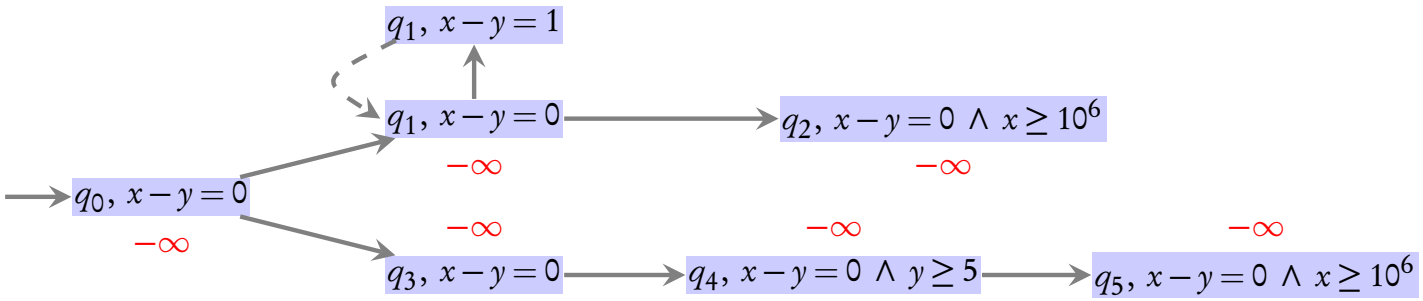


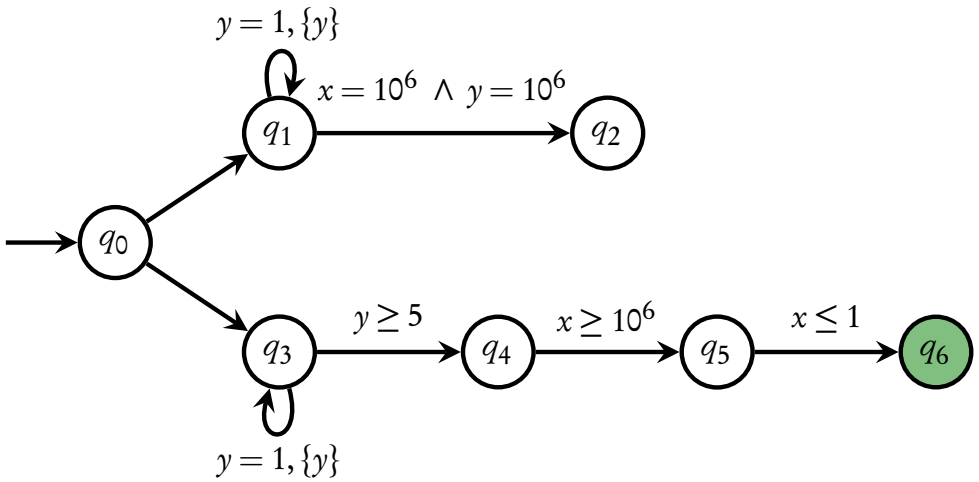
No disabled edge ⇒ all states seen ⇒ no need for bounds!



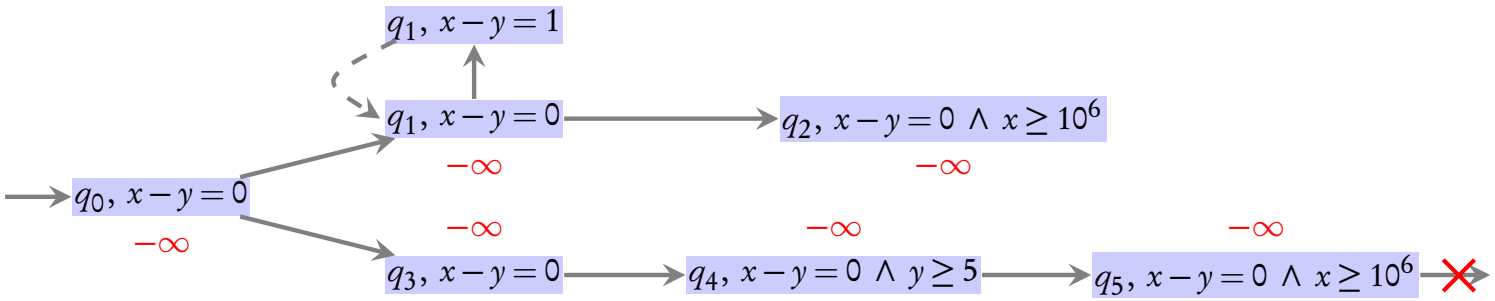


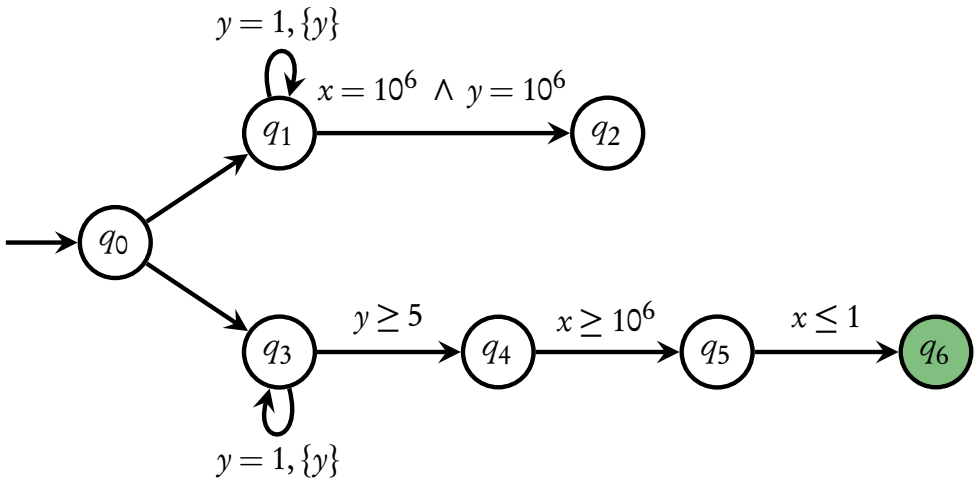
No disabled edge ⇒ all states seen ⇒ no need for bounds!



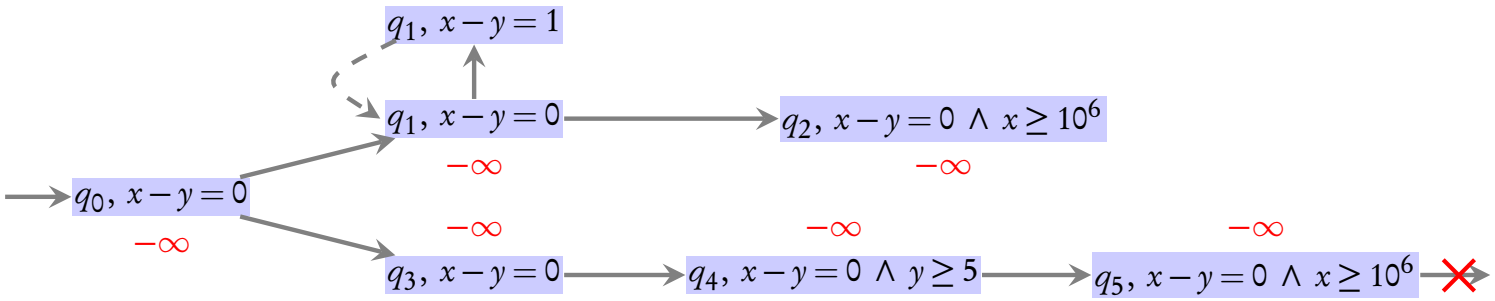


No disabled edge ⇒ all states seen ⇒ no need for bounds!

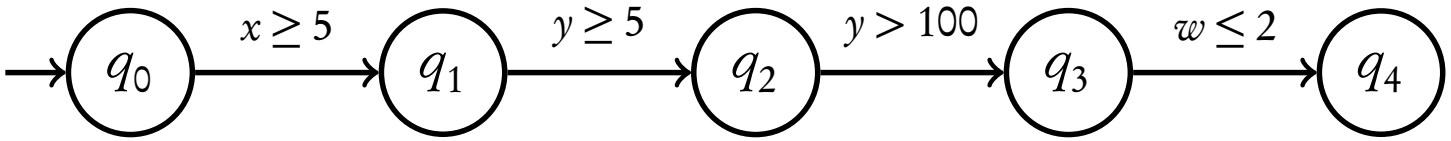


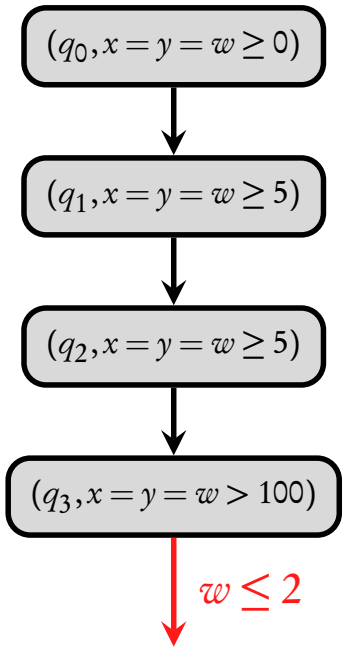


No disabled edge ⇒ all states seen ⇒ no need for bounds!

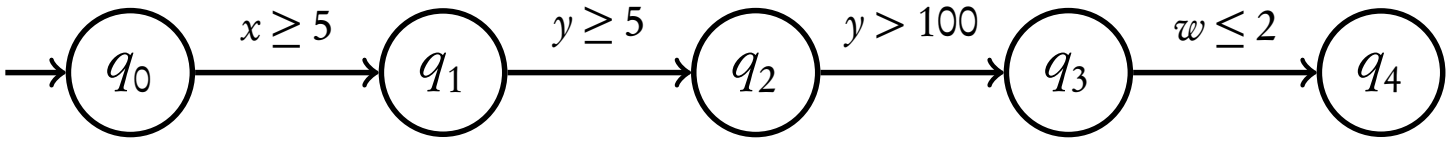


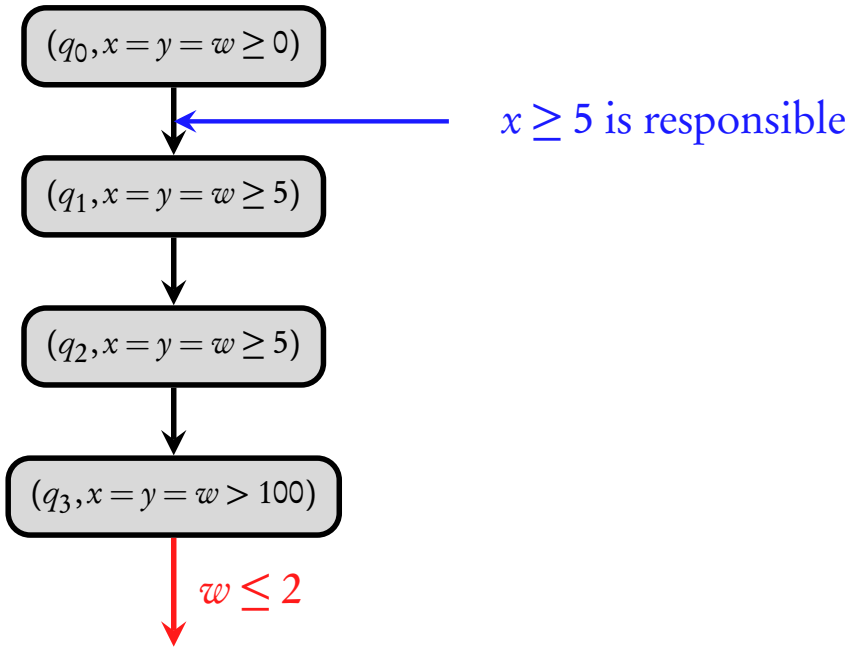
What to do when a disabled edge is seen?


q_0
$x \geq 5$
q_1
$y \geq 5$
q_2
$y > 100$
q_3
$w \leq 2$
q_4

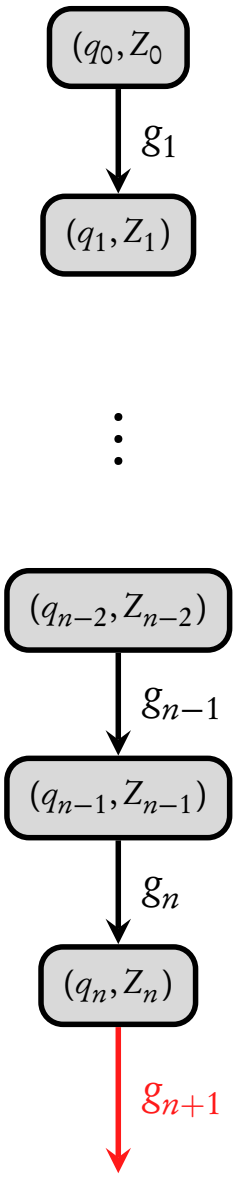

$(q_0, x = y = w \geq 0)$
$(q_1, x = y = w \geq 5)$
$(q_2, x = y = w \geq 5)$
$(q_3, x = y = w > 100)$
$w \leq 2$

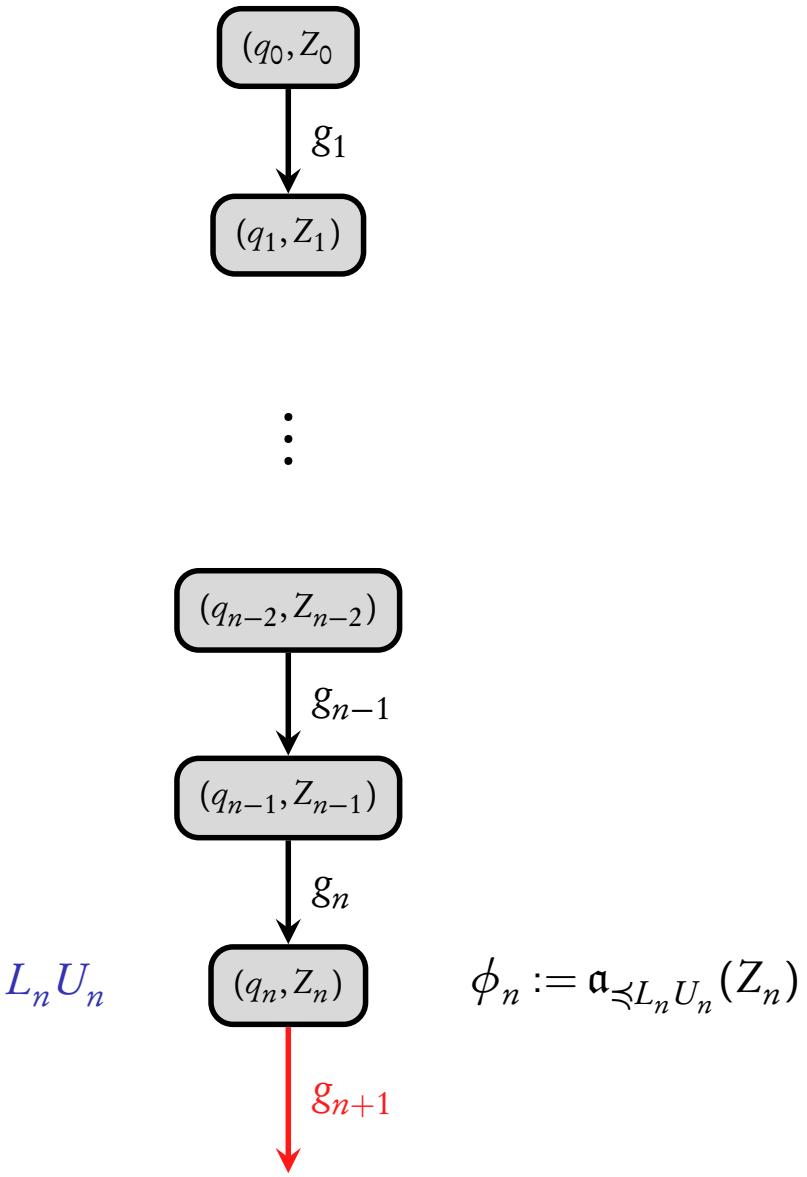
$$L(x) = 5,\ U(w) = 2$$


q_0
$x \geq 5$
q_1
$y \geq 5$
q_2
$y > 100$
q_3
$w \leq 2$
q_4


$(q_0, x = y = w \geq 0)$
$x \geq 5$ is responsible
$(q_1, x = y = w \geq 5)$
$(q_2, x = y = w \geq 5)$
$(q_3, x = y = w > 100)$
$w \leq 2$

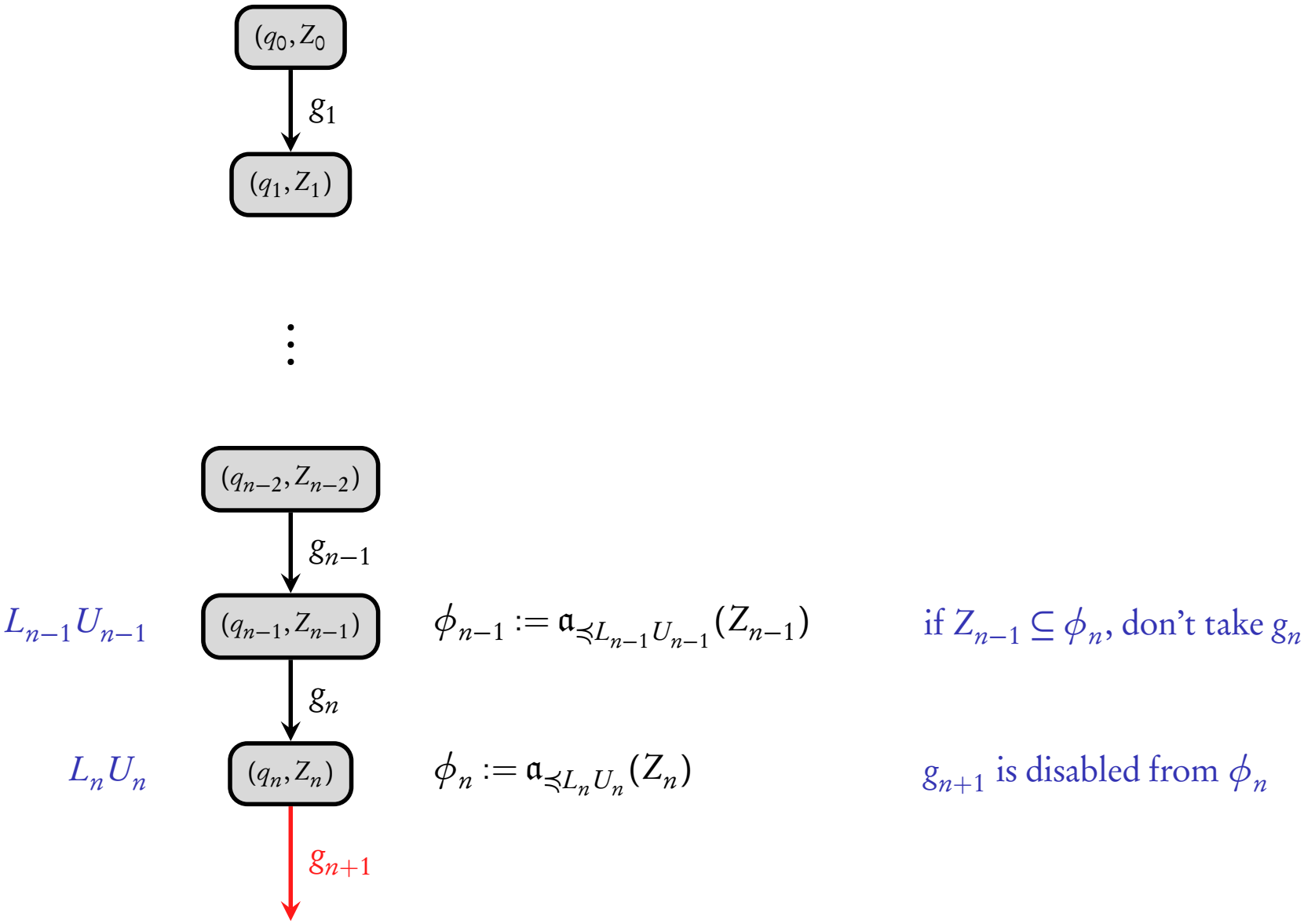
$$L(x) = 5,\ U(w) = 2$$


$(q_0, Z_0$
g_1
(q_1, Z_1)
(q_{n-2}, Z_{n-2})
g_{n-1}
(q_{n-1}, Z_{n-1})
g_n
(q_n, Z_n)
g_{n+1}

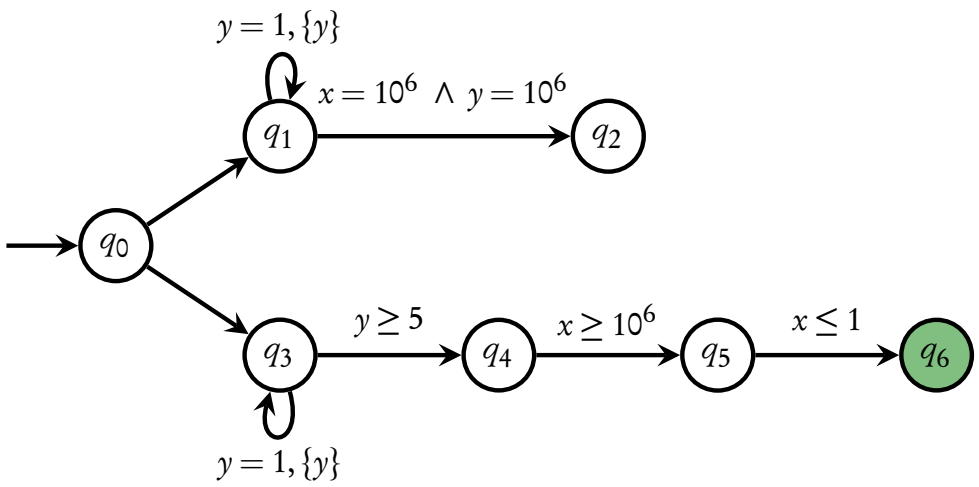


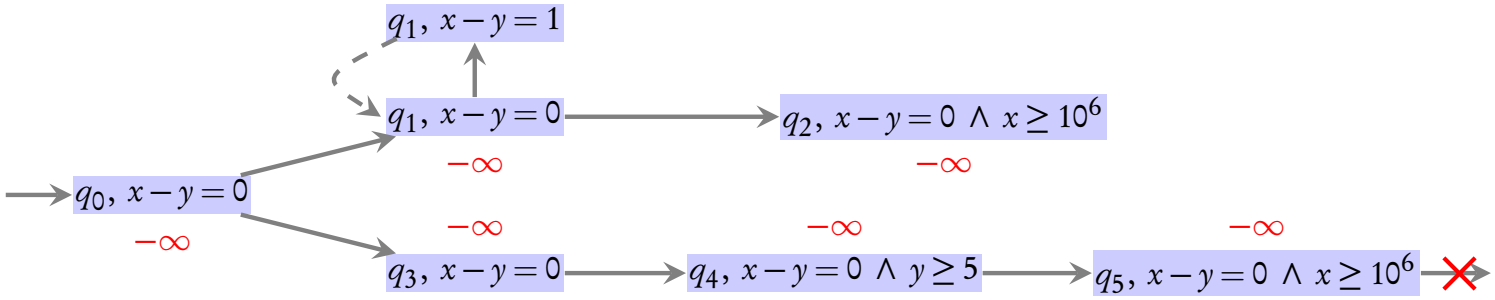
$\phi_n := \mathfrak{a}_{\preceq L_n U_n}(Z_n)$

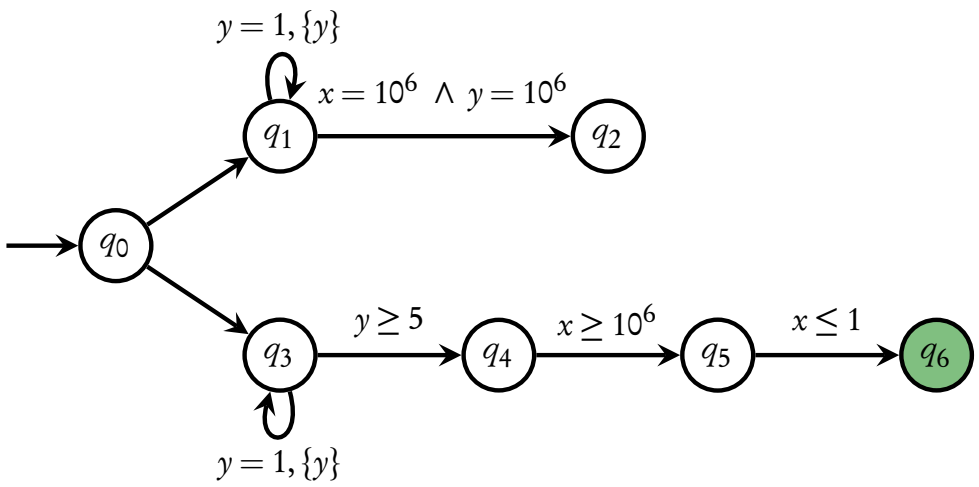
g_{n+1} is disabled from ϕ_n

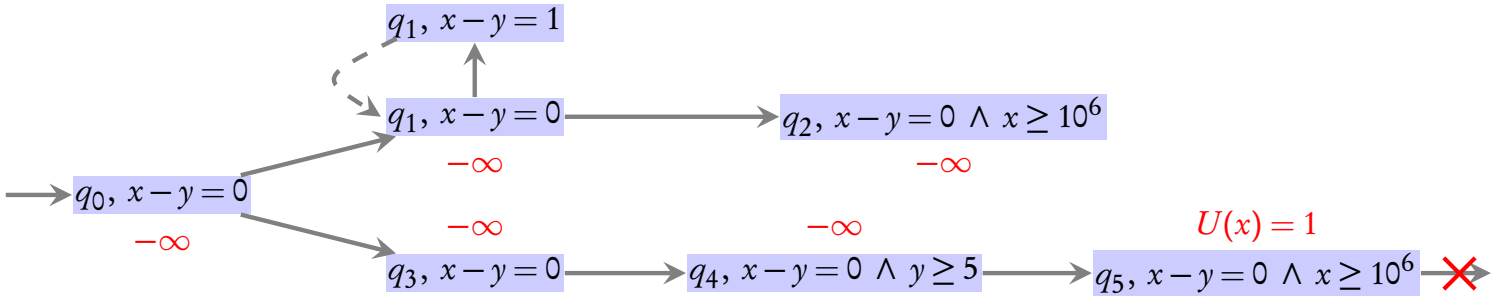

(q_0, Z_0)
g_1
(q_1, Z_1)
(q_{n-2}, Z_{n-2})
g_{n-1}
$L_{n-1}U_{n-1}$
(q_{n-1}, Z_{n-1})
$\phi_{n-1} := \mathfrak{a}_{\preccurlyeq L_{n-1}U_{n-1}}(Z_{n-1})$
if $Z_{n-1} \subseteq \phi_n$, don't take g_n
g_n
L_nU_n
(q_n, Z_n)
$\phi_n := \mathfrak{a}_{\preccurlyeq L_nU_n}(Z_n)$
g_{n+1} is disabled from ϕ_n
g_{n+1}

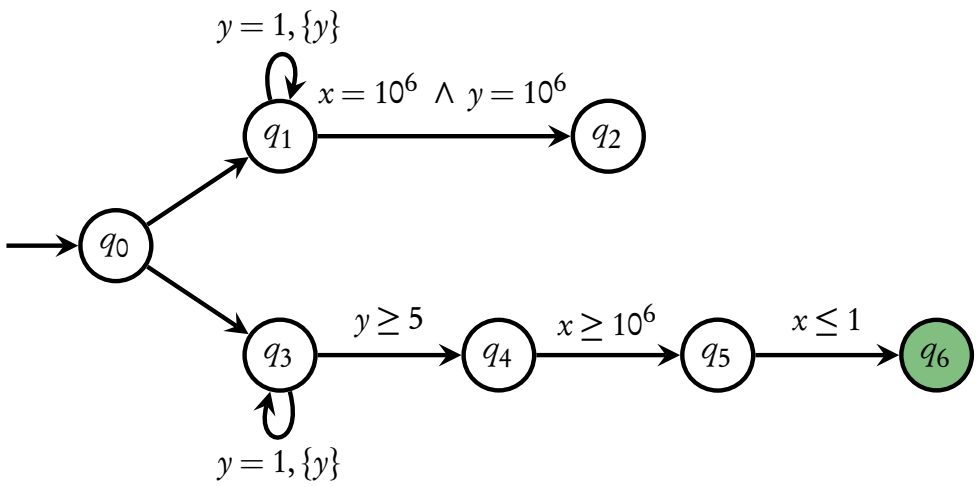
$L_0 U_0$	(q_0, Z_0)	$\phi_1 := \mathfrak{a}_{\preccurlyeq L_0 U_0}(Z_0)$	if $Z_0 \subseteq \phi_1$, don't take g_1
	$\downarrow g_1$		
$L_1 U_1$	(q_1, Z_1)	$\phi_1 := \mathfrak{a}_{\preccurlyeq L_1 U_1}(Z_1)$	if $Z_1 \subseteq \phi_2$, don't take g_2
	$\vdots$		
$L_{n-2} U_{n-2}$	(q_{n-2}, Z_{n-2})	$\phi_{n-1} := \mathfrak{a}_{\preccurlyeq L_{n-2} U_{n-2}}(Z_{n-2})$	if $Z_{n-2} \subseteq \phi_{n-1}$, don't take g_{n-1}
	$\downarrow g_{n-1}$		
$L_{n-1} U_{n-1}$	(q_{n-1}, Z_{n-1})	$\phi_{n-1} := \mathfrak{a}_{\preccurlyeq L_{n-1} U_{n-1}}(Z_{n-1})$	if $Z_{n-1} \subseteq \phi_n$, don't take g_n
	$\downarrow g_n$		
$L_n U_n$	(q_n, Z_n)	$\phi_n := \mathfrak{a}_{\preccurlyeq L_n U_n}(Z_n)$	g_{n+1} is disabled from ϕ_n
	$\downarrow g_{n+1}$		

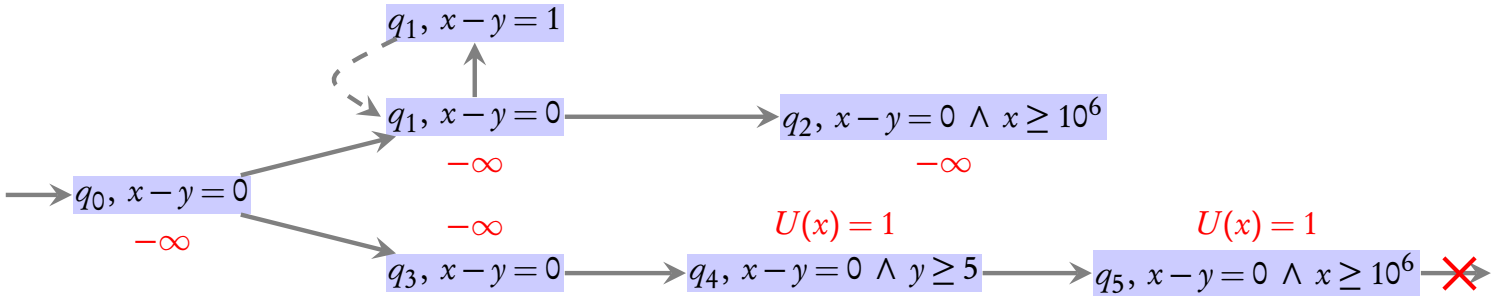

$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

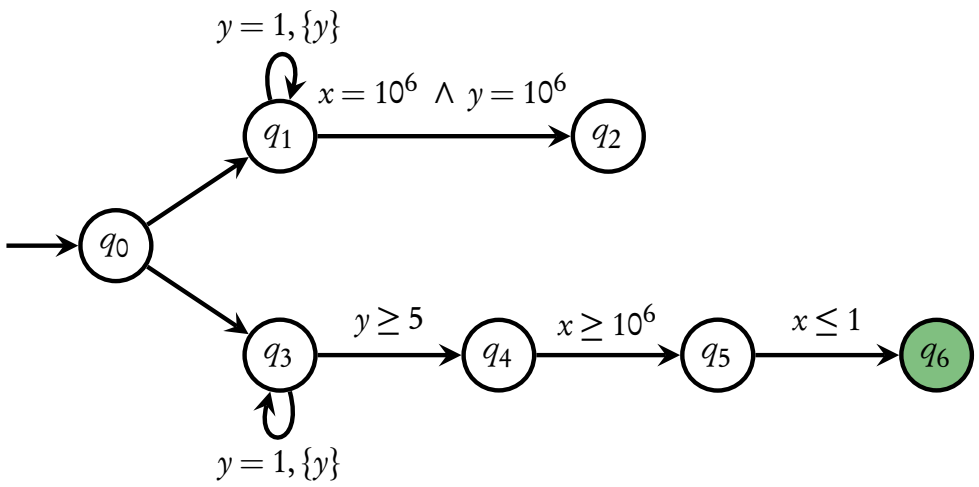

$q_1, \ x - y = 1$
$q_1, \ x - y = 0$
$q_2, \ x - y = 0 \ \wedge \ x \geq 10^6$
$-\infty$
$-\infty$
$q_0, \ x - y = 0$
$-\infty$
$-\infty$
$-\infty$
$-\infty$
$q_3, \ x - y = 0$
$q_4, \ x - y = 0 \ \wedge \ y \geq 5$
$q_5, \ x - y = 0 \ \wedge \ x \geq 10^6$

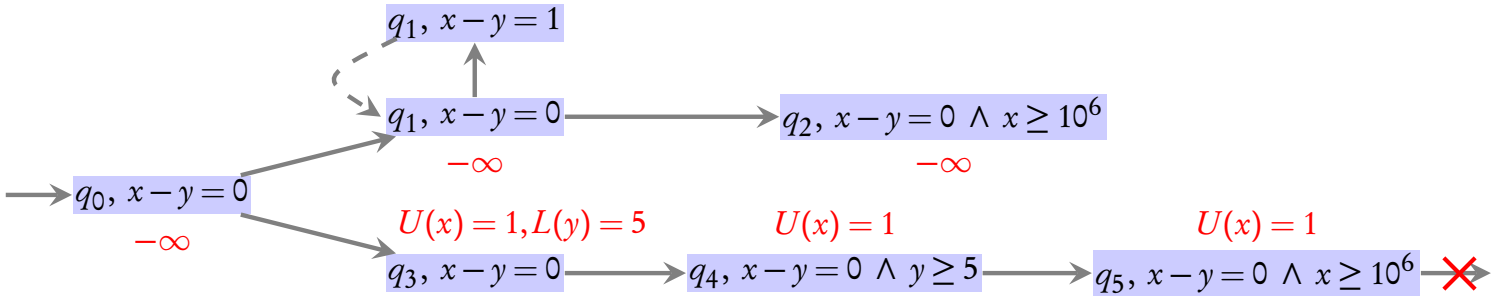

$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

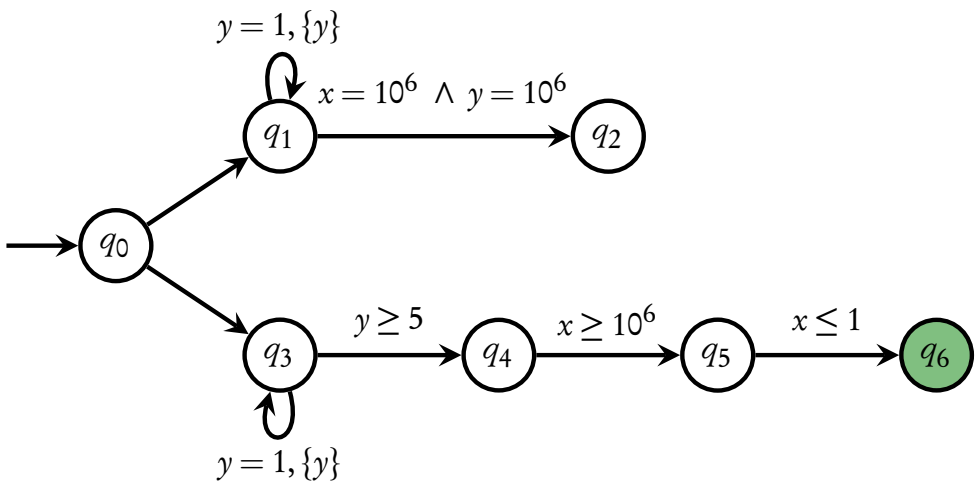

$q_1,\ x - y = 1$
$q_1,\ x - y = 0$
$q_2,\ x - y = 0 \ \wedge \ x \geq 10^6$
$-\infty$
$-\infty$
$q_0,\ x - y = 0$
$-\infty$
$-\infty$
$-\infty$
$U(x) = 1$
$q_3,\ x - y = 0$
$q_4,\ x - y = 0 \ \wedge \ y \geq 5$
$q_5,\ x - y = 0 \ \wedge \ x \geq 10^6$

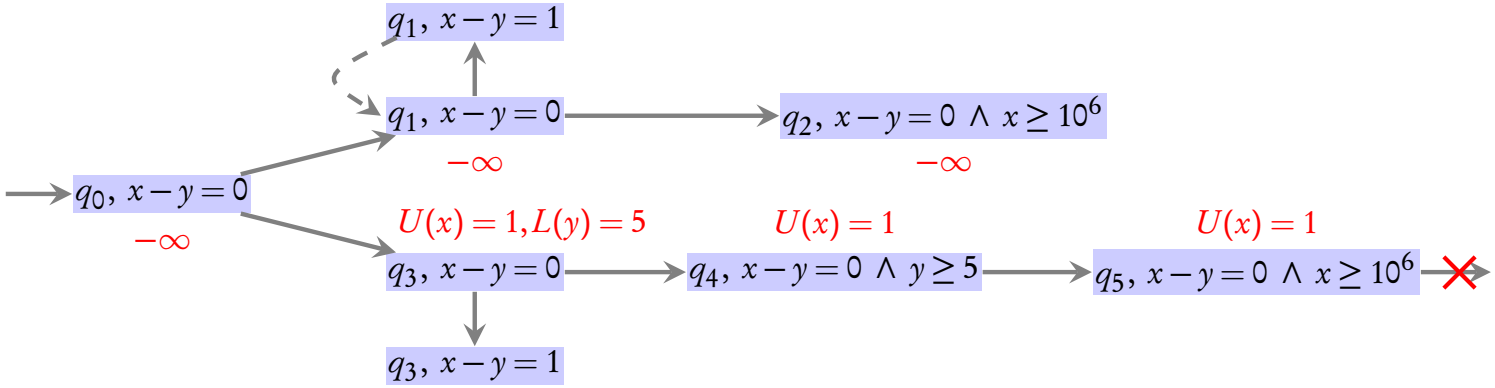

$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

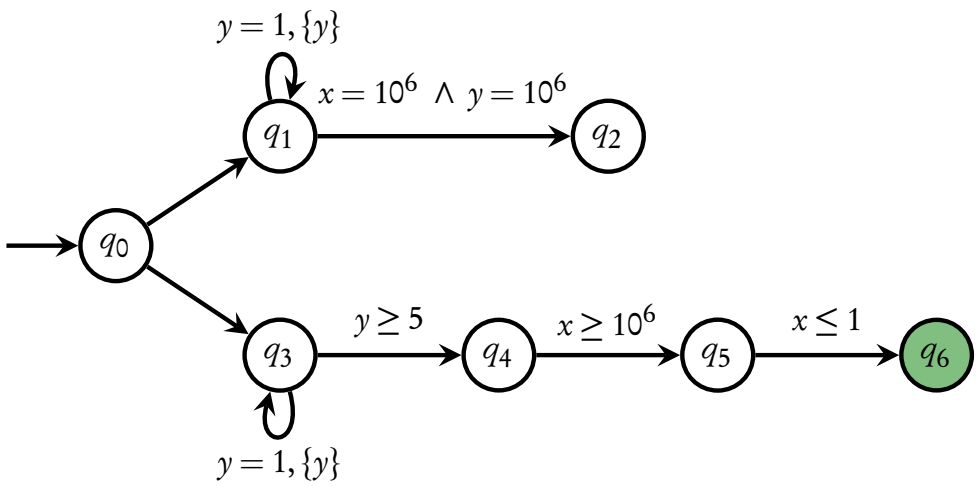

$q_1,\ x - y = 1$
$q_1,\ x - y = 0$
$-\infty$
$q_2,\ x - y = 0 \ \wedge \ x \geq 10^6$
$-\infty$
$q_0,\ x - y = 0$
$-\infty$
$-\infty$
$U(x) = 1$
$U(x) = 1$
$q_3,\ x - y = 0$
$q_4,\ x - y = 0 \ \wedge \ y \geq 5$
$q_5,\ x - y = 0 \ \wedge \ x \geq 10^6$

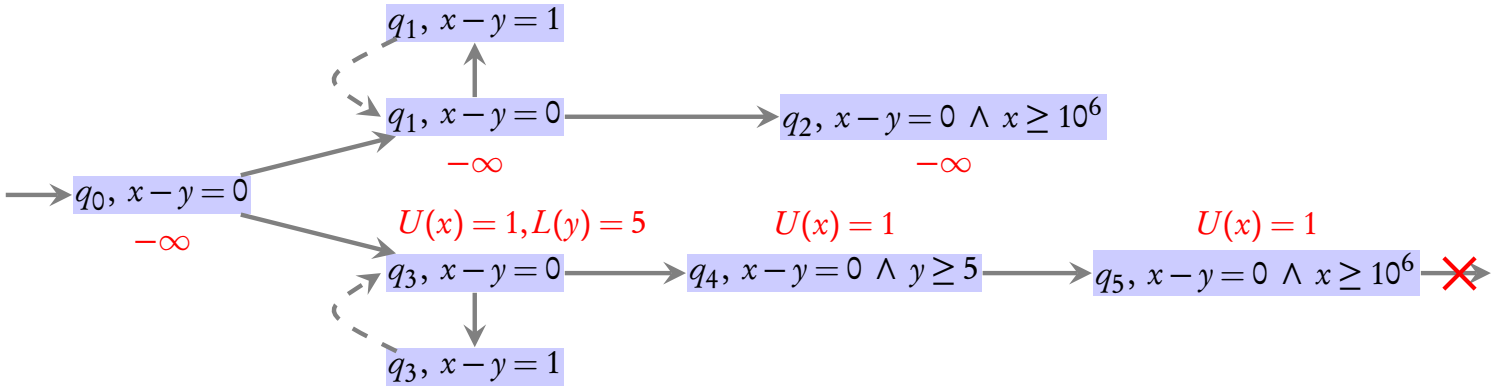

$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

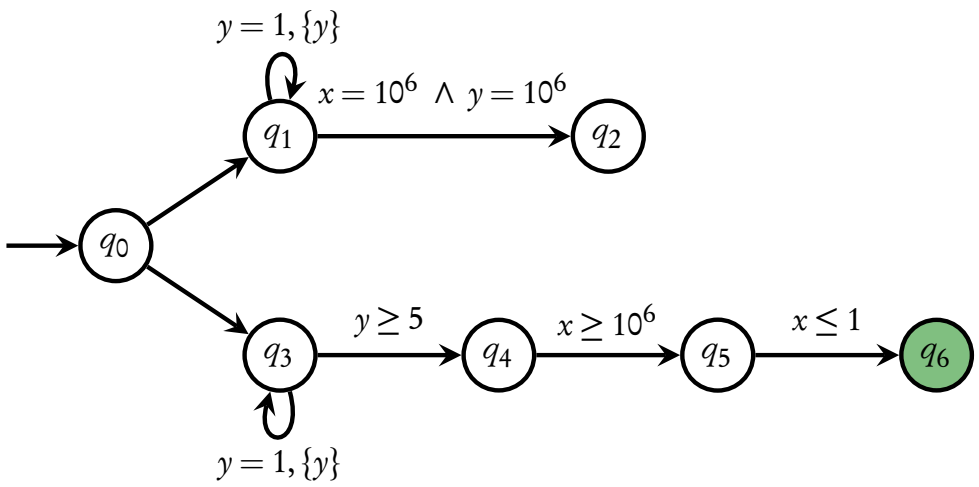

$q_1, \ x - y = 1$
$q_1, \ x - y = 0$
$-\infty$
$q_2, \ x - y = 0 \ \wedge \ x \geq 10^6$
$-\infty$
$q_0, \ x - y = 0$
$-\infty$
$U(x) = 1, L(y) = 5$
$q_3, \ x - y = 0$
$U(x) = 1$
$q_4, \ x - y = 0 \ \wedge \ y \geq 5$
$U(x) = 1$
$q_5, \ x - y = 0 \ \wedge \ x \geq 10^6$


$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$


$q_1, \ x - y = 1$
$q_1, \ x - y = 0$
$-\infty$
$q_2, \ x - y = 0 \ \wedge \ x \geq 10^6$
$-\infty$
$q_0, \ x - y = 0$
$-\infty$
$U(x) = 1, L(y) = 5$
$q_3, \ x - y = 0$
$U(x) = 1$
$q_4, \ x - y = 0 \ \wedge \ y \geq 5$
$U(x) = 1$
$q_5, \ x - y = 0 \ \wedge \ x \geq 10^6$
$q_3, \ x - y = 1$

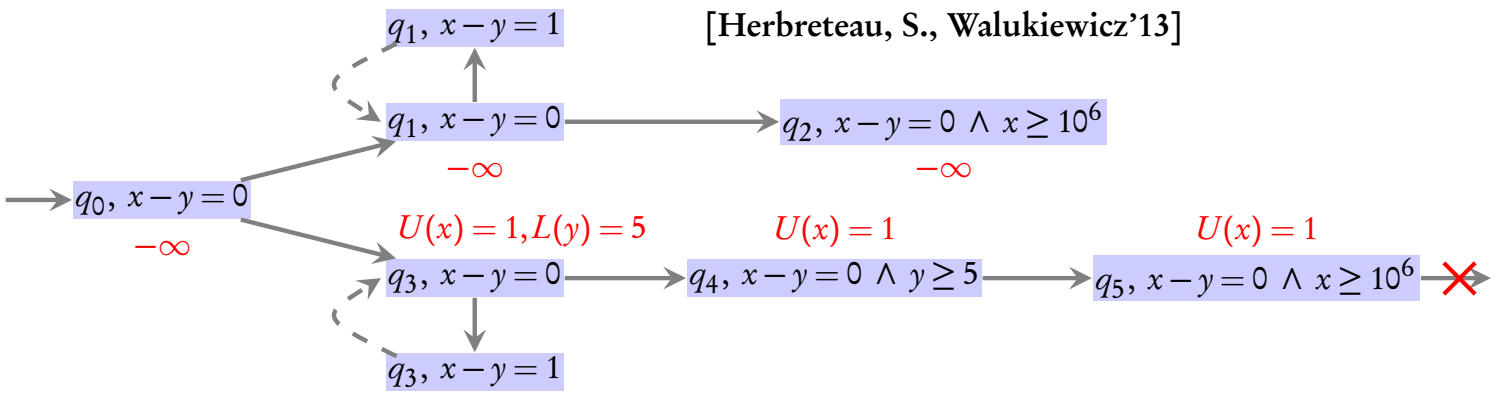

$y = 1, \{y\}$
$x = 10^6 \ \wedge \ y = 10^6$
q_1
q_2
q_0
q_3
$y \geq 5$
q_4
$x \geq 10^6$
q_5
$x \leq 1$
q_6
$y = 1, \{y\}$

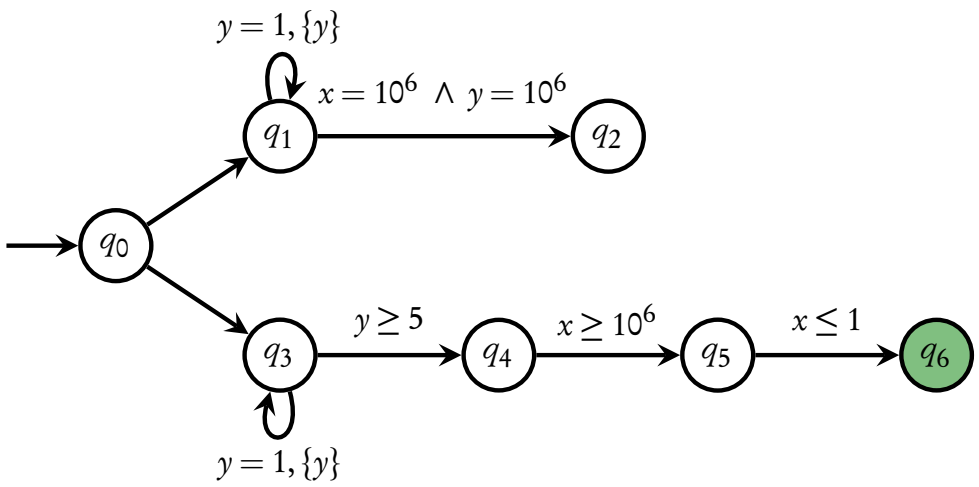

$q_1, \ x - y = 1$
$q_1, \ x - y = 0$
$-\infty$
$q_2, \ x - y = 0 \ \wedge \ x \geq 10^6$
$-\infty$
$q_0, \ x - y = 0$
$-\infty$
$U(x) = 1, L(y) = 5$
$q_3, \ x - y = 0$
$U(x) = 1$
$q_4, \ x - y = 0 \ \wedge \ y \geq 5$
$U(x) = 1$
$q_5, \ x - y = 0 \ \wedge \ x \geq 10^6$
$q_3, \ x - y = 1$



Lazy bounds propagation

[Herbreteau, S., Walukiewicz'13]

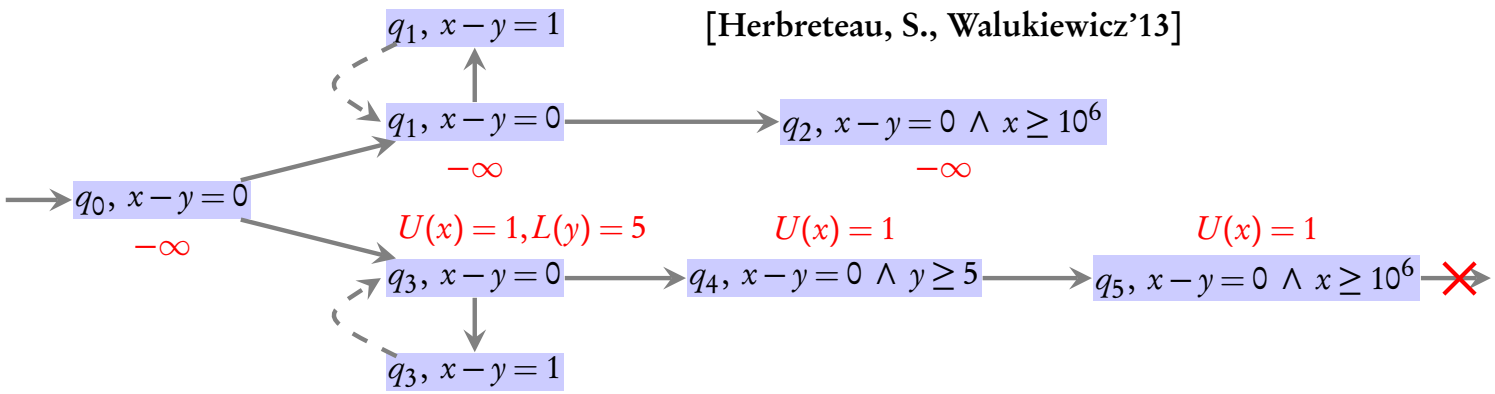


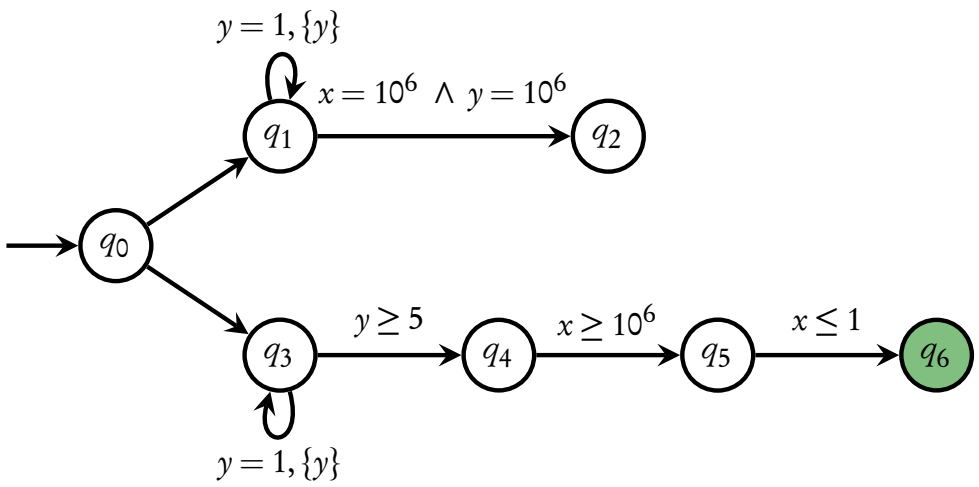


Zone graph + $\alpha_{\preccurlyeq LU}$

Lazy bounds propagation

[Herbreteau, S., Walukiewicz'13]

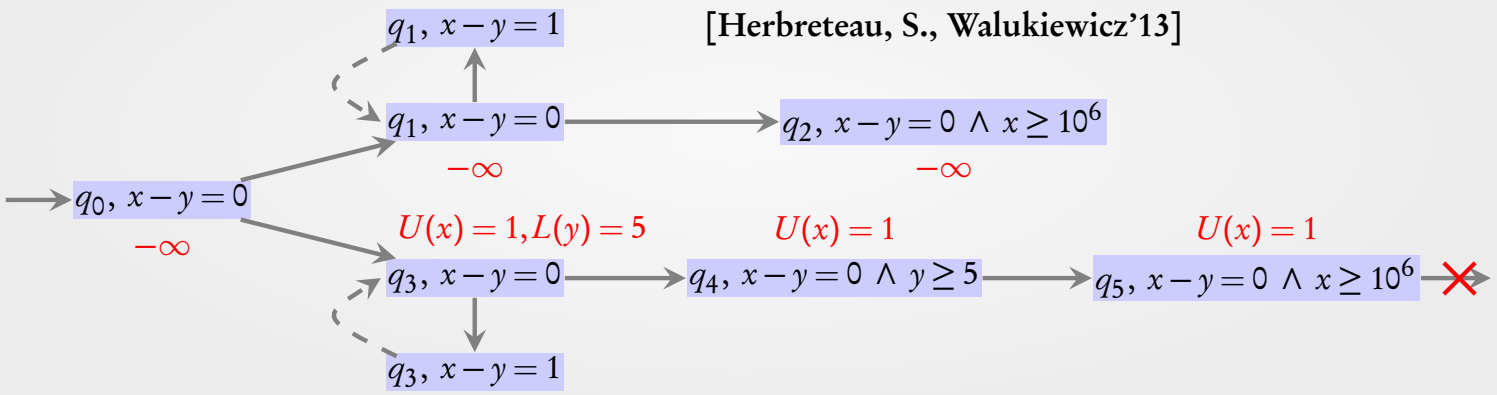




Zone graph + $\alpha_{\preccurlyeq LU}$

Lazy bounds propagation

[Herbreteau, S., Walukiewicz'13]



Reachability for timed automata

Standard algorithm: **covering** tree

Convex abstractions

Bounds by **static analysis**

Observation 1

Non-convex abstractions

Optimality

Observation 2

Dynamic LU-bounds

Experiments

Model	nb. of clocks	UPPAAL (-C)		static		lazy	
		nodes	sec.	nodes	sec.	nodes	sec.
CSMA/CD 10	11	120845	1.12				
CSMA/CD 11	12	311310	3.23				
CSMA/CD 12	13	786447	14.8				
C-CSMA/CD 6	6	8153	0.19				
C-CSMA/CD 7							
C-CSMA/CD 8							
FDDI 50	151						
FDDI 70	211						
FDDI 140	421						
Fischer 9	9	135485	1.17				
Fischer 10	10	447598	5.04				
Fischer 11	11	1464971	20.5				

- **UPPAAL (-C)** shows results from UPPAAL tool which uses static bounds
- **static** is our implementation of UPPAAL's algo
- Time out (180s)

Experiments

Model	nb. of clocks	UPPAAL (-C)		static		lazy	
		nodes	sec.	nodes	sec.	nodes	sec.
CSMA/CD 10	11	120845	1.12	78604	1.89		
CSMA/CD 11	12	311310	3.23	198669	5.07		
CSMA/CD 12	13	786447	14.8	493582	13.58		
C-CSMA/CD 6	6	8153	0.19				
C-CSMA/CD 7							
C-CSMA/CD 8							
FDDI 50	151			10299	13.61		
FDDI 70	211						
FDDI 140	421						
Fischer 9	9	135485	1.17	135485	3.23		
Fischer 10	10	447598	5.04	447598	12.73		
Fischer 11	11	1464971	20.5	1464971	46.97		

- **UPPAAL (-C)** shows results from UPPAAL tool which uses static bounds
- **static** is our implementation of UPPAAL's algo
- Time out (180s)

Experiments

Model	nb. of clocks	UPPAAL (-C)		static		lazy	
		nodes	sec.	nodes	sec.	nodes	sec.
CSMA/CD 10	11	120845	1.12	78604	1.89	78604	2.10
CSMA/CD 11	12	311310	3.23	198669	5.07	198669	5.64
CSMA/CD 12	13	786447	14.8	493582	13.58	493582	14.71
C-CSMA/CD 6	6	8153	0.19			1876	0.09
C-CSMA/CD 7						18414	0.97
C-CSMA/CD 8						172040	10.36
FDDI 50	151			10299	13.61	401	0.4
FDDI 70	211					561	1.36
FDDI 140	421					1121	18.25
Fischer 9	9	135485	1.17	135485	3.23	135485	4.38
Fischer 10	10	447598	5.04	447598	12.73	447598	17.27
Fischer 11	11	1464971	20.5	1464971	46.97	1464971	67.61

- **UPPAAL (-C)** shows results from UPPAAL tool which uses static bounds
- **static** is our implementation of UPPAAL's algo
- Time out (180s)

Conclusion

- Computing shorter proofs for un-reachability **dynamically**
- Main technical ingredient: efficient **inclusion** test
- Extended to **Priced** Timed Automata [Bouyer, Colange, Markey'16]

Conclusion

- Computing shorter proofs for un-reachability **dynamically**
- Main technical ingredient: efficient **inclusion** test
- Extended to **Priced** Timed Automata [Bouyer, Colange, Markey'16]

- **Future work:**
  - Improve lazy propagation
  - partial-order reduction
  - timed games