Topics in Timed Automata

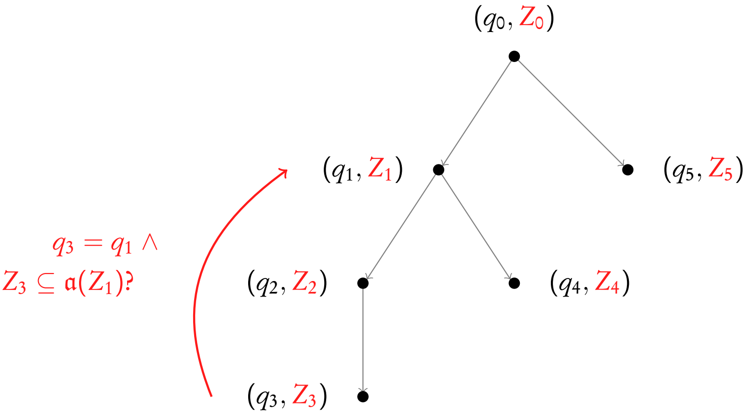
B. Srivathsan

RWTH-Aachen

Software modeling and Verification group

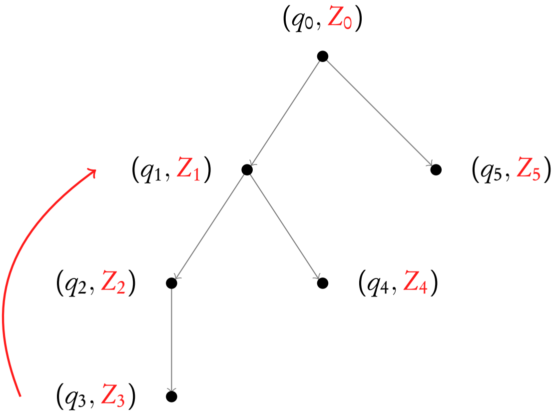
Reachability for timed automata

Key idea: Compute the **zone graph**, use **abstraction** for termination



Reachability for timed automata

Key idea: Compute the **zone graph**, use **abstraction** for termination

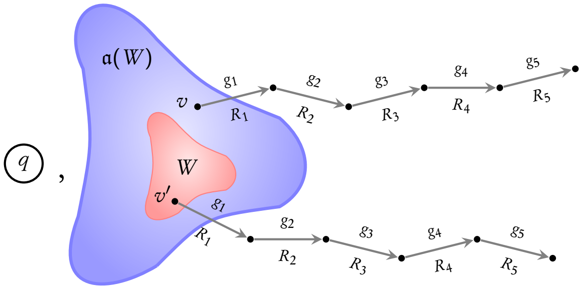


$q_3 = q_1 \wedge$
$Z_3 \subseteq \mathfrak{a}(Z_1)$?

Coarser the abstraction, **smaller** the zone graph

Condition 1: $\mathfrak{a}$ should have **finite range**

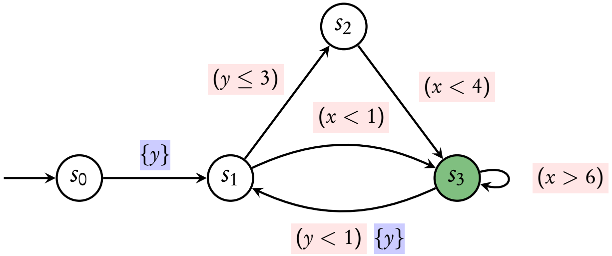
Condition 2: $\mathfrak{a}$ should be sound $\Rightarrow \mathfrak{a}(W)$ can contain only valuations **simulated** by W



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]

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

$$v \preccurlyeq_M v'$$

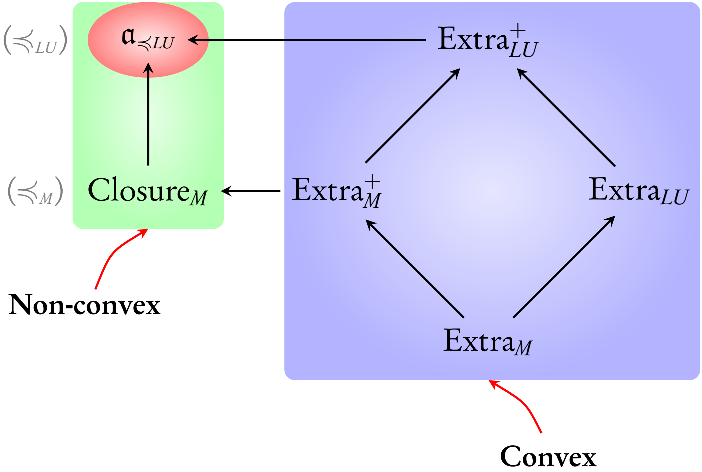
LU-bounds [BBLP04]

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

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

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

Abstractions in literature [BBLP04, Bou04]



Last lecture: **Efficiently** using the M-bounds based Closure_M abstraction

Lecture 7:

Lower-upper bounds for abstraction

LU-guards: guards consistent with given L and U

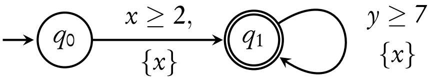
LU-guards for $L(x) = 3, U(x) = 5, L(y) = 8, U(y) = -\infty$

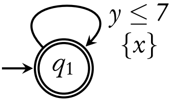
$$x \geq 0, x \geq 1, x \geq 2, x \geq 3$$
$$x \leq 0, x \leq 1, \ldots, x \leq 5$$
$$y \geq 0, y \geq 1, \ldots, y \geq 8$$

(same with $<$ and $>$)

LU-automata: automata with only LU-guards

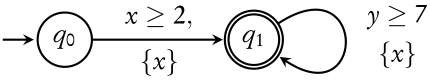
$$L(x) = 3, U(x) = 5, L(y) = 8, U(y) = -\infty$$

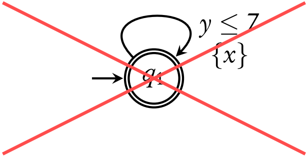




LU-automata: automata with only LU-guards

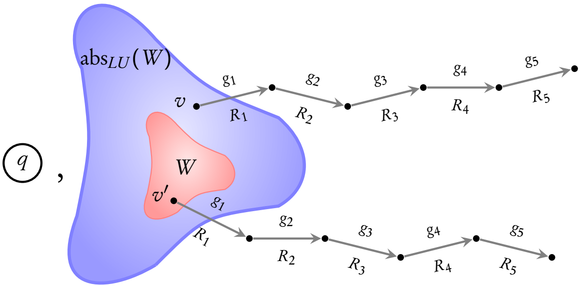
$$L(x) = 3, U(x) = 5, L(y) = 8, U(y) = -\infty$$





What do we need?

1. An abstraction abs_{LU} that is **sound** and **complete** for **all** LU-automata



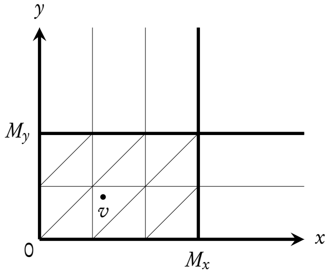
2. **Efficient** inclusion testing $Z \subseteq abs_{LU}(Z')$

Step 1:

LU-regions

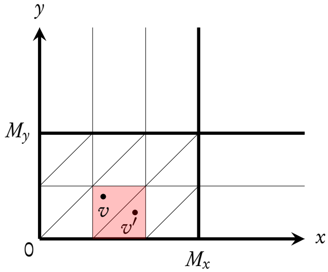
Classic regions [AD94] : v' belongs to $[v]^M$ if:

- **Invariance by guards:** v' satisfies the same guards as v,
- **Invariance by time-elapse:** for every time elapse $\delta \in \mathbb{R}_{\geq 0}$, there is a $\delta' \in \mathbb{R}_{\geq 0}$ such that $v' + \delta' \in [v + \delta]^M$.



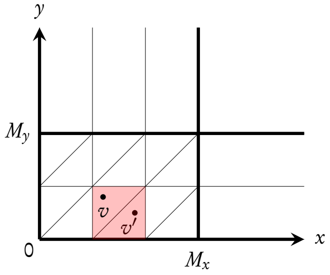
Classic regions [AD94] : v' belongs to $[v]^M$ if:

- **Invariance by guards:** v' satisfies the same guards as v, $\surd$
- **Invariance by time-elapse:** for every time elapse $\delta \in \mathbb{R}_{\geq 0}$, there is a $\delta' \in \mathbb{R}_{\geq 0}$ such that $v' + \delta' \in [v + \delta]^M$.



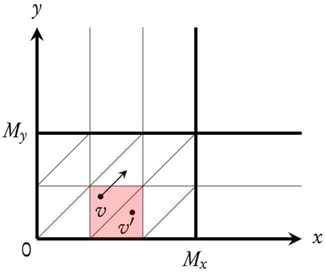
Classic regions [AD94] : v' belongs to $[v]^M$ if:

- **Invariance by guards:** v' satisfies the same guards as v, $\surd$
- **Invariance by time-elapse:** for every time elapse $\delta \in \mathbb{R}_{\geq 0}$, there is a $\delta' \in \mathbb{R}_{\geq 0}$ such that $v' + \delta' \in [v + \delta]^M$.



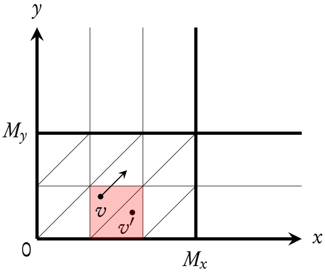
Classic regions [AD94] : v' belongs to $[v]^M$ if:

- **Invariance by guards:** v' satisfies the same guards as v, $\surd$
- **Invariance by time-elapse:** for every time elapse $\delta \in \mathbb{R}_{\geq 0}$, there is a $\delta' \in \mathbb{R}_{\geq 0}$ such that $v' + \delta' \in [v + \delta]^M$.



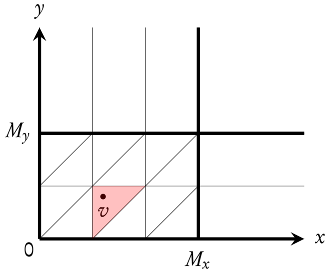
Classic regions [AD94] : v' belongs to $[v]^M$ if:

- **Invariance by guards:** v' satisfies the same guards as v, $\surd$
- **Invariance by time-elapse:** for every time elapse $\delta \in \mathbb{R}_{\geq 0}$, there is a $\delta' \in \mathbb{R}_{\geq 0}$ such that $v' + \delta' \in [v + \delta]^M$. $\times$



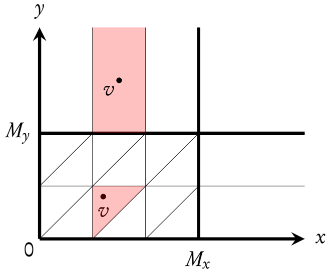
Classic regions [AD94] : v' belongs to $[v]^M$ if:

- **Invariance by guards:** v' satisfies the same guards as v, $\surd$
- **Invariance by time-elapse:** for every time elapse $\delta \in \mathbb{R}_{\geq 0}$, there is a $\delta' \in \mathbb{R}_{\geq 0}$ such that $v' + \delta' \in [v + \delta]^M$. $\surd$



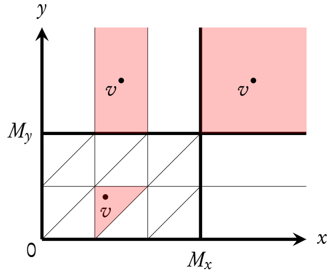
Classic regions [AD94] : v' belongs to $[v]^M$ if:

- **Invariance by guards:** v' satisfies the same guards as v, $\surd$
- **Invariance by time-elapse:** for every time elapse $\delta \in \mathbb{R}_{\geq 0}$, there is a $\delta' \in \mathbb{R}_{\geq 0}$ such that $v' + \delta' \in [v + \delta]^M$. $\surd$



Classic regions [AD94] **:** v' belongs to $[v]^M$ if:

- **Invariance by guards:** v' satisfies the same guards as v, $\surd$
- **Invariance by time-elapse:** for every time elapse $\delta \in \mathbb{R}_{\geq 0}$, there is a $\delta' \in \mathbb{R}_{\geq 0}$ such that $v' + \delta' \in [v + \delta]^M$. $\surd$



Classic regions [AD94] **:** Given M, v' belongs to $[v]^M$ if:

- **Invariance by guards:** v' satisfies the same guards as v,
- **Invariance by time-elapse:** for every pair of clocks x, y with:

$$v(x) \leq M_x, \quad v(y) \leq M_y$$
$$\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor \text{ and } \lfloor v(y) \rfloor = \lfloor v'(y) \rfloor$$

  we have:

  - if $0 < \{v(x)\} < \{v(y)\}$, then $0 < \{v'(x)\} < \{v'(y)\}$
  - if $0 < \{v(x)\} = \{v(y)\}$, then $0 < \{v'(x)\} = \{v'(y)\}$

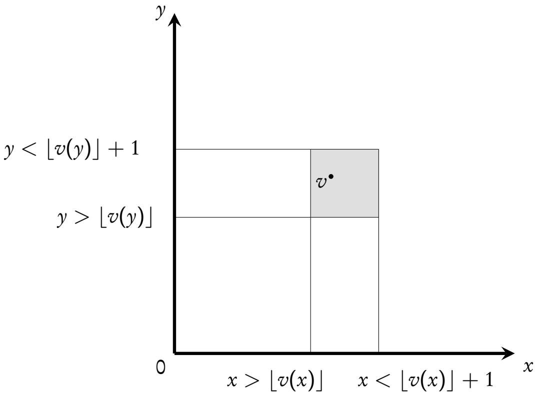
$\lfloor v(x) \rfloor$: integer part of $v(x)$

$\{v(x)\}$: fractional part of $v(x)$

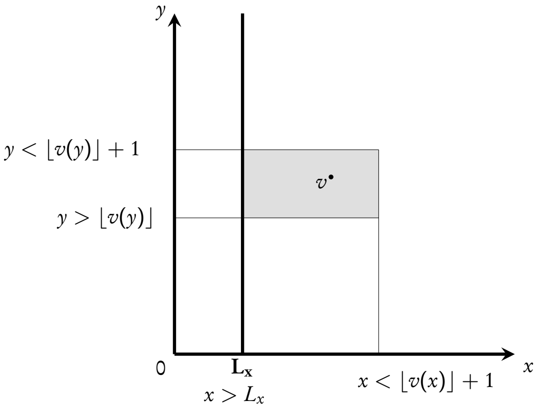
Coming next...

Regions for the **LU**-case

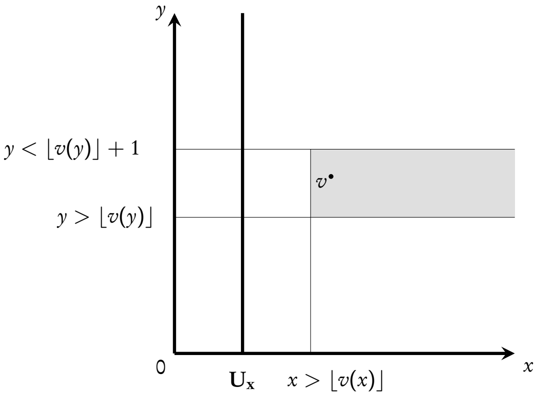
Invariance by (LU-) guards: $v(x)$ is less than both L_x, U_x



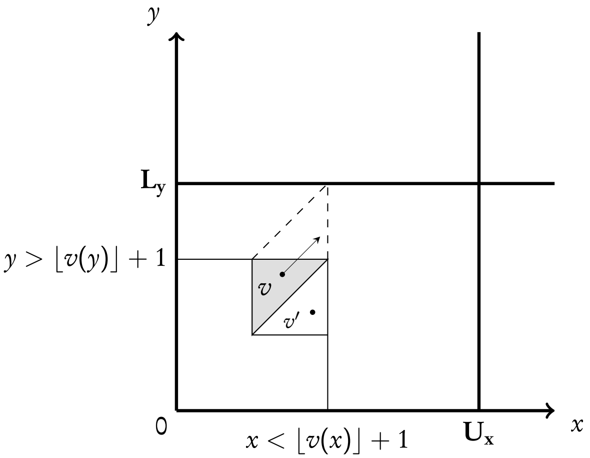
Invariance by (LU-) guards: $v(x) > L_x$



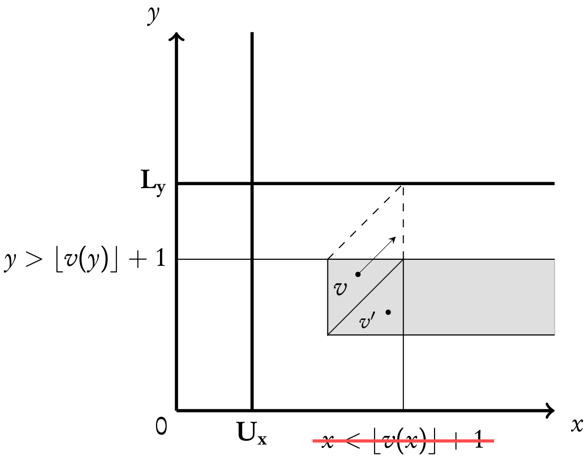
Invariance by (LU-) guards: $v(x) > U_x$



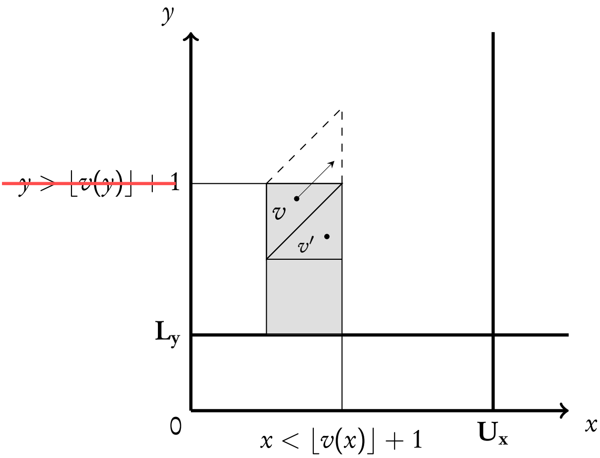
Invariance by time-elapse: $v(x) \leq U_x$, $v(y) \leq L_y$



Invariance by time-elapse: $v(x) > U_x, \quad v(y) \le L_y$



Invariance by time-elapse: $v(x) \leq U_x, \quad v(y) > L_y$



LU-regions

Definition: v' belongs to $\langle v \rangle^{LU}$ if:

- **Invariance by guards:** v' satisfies the same guards as v,
- **Invariance by time-elapse:** for every pair of clocks x, y with:

$$v(x) \leq U_x,\ v(y) \leq L_y$$
$$\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor \text{ and } \lfloor v(y) \rfloor = \lfloor v'(y) \rfloor,$$

  we have:

  - if $\mathtt{0} < \{v(x)\} < \{v(y)\}$, then $\mathtt{0} < \{v'(x)\} < \{v'(y)\}$
  - if $\mathtt{0} < \{v(x)\} = \{v(y)\}$, then $\mathtt{0} < \{v'(x)\} = \{v'(y)\}$

Step 2:

An abstraction abs_{LU}

$$v \sqsubseteq_{LU} v'$$

if

$$\exists \delta' \in \mathbb{R}_{\geq 0} \text{ s.t. } v' + \delta' \in \langle v \rangle^{LU}$$

$$v \sqsubseteq_{LU} v'$$

if

$$\exists \delta' \in \mathbb{R}_{\geq 0} \text{ s.t. } v' + \delta' \in \langle v \rangle^{LU}$$

Definition

$$\mathrm{abs}_{LU}(W) = \{v \mid \exists v' \in W \text{ s.t. } v \sqsubseteq_{LU} v'\}$$

$$v \sqsubseteq_{LU} v'$$

if

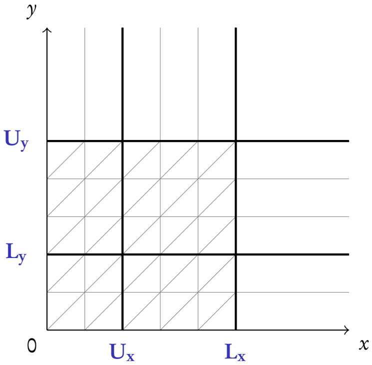
$$\exists \delta' \in \mathbb{R}_{\geq 0} \text{ s.t. } v' + \delta' \in \langle v \rangle^{LU}$$

Definition

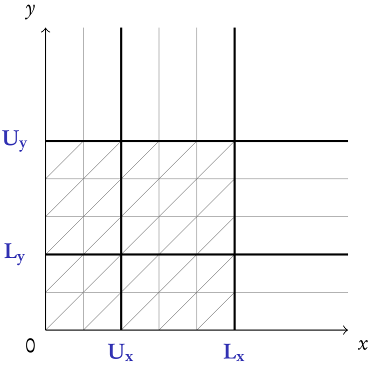
$$\mathrm{abs}_{LU}(W) = \{v \mid \exists v' \in W \text{ s.t. } v \sqsubseteq_{LU} v'\}$$

abs_{LU} is **sound** and **complete**

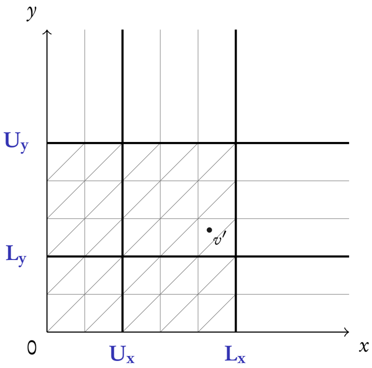
Example



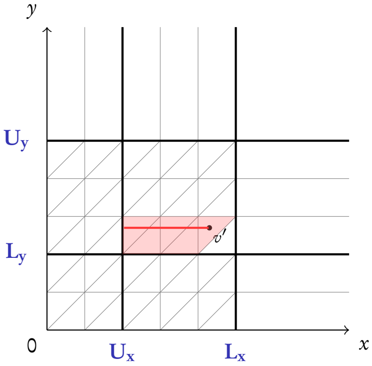
Example



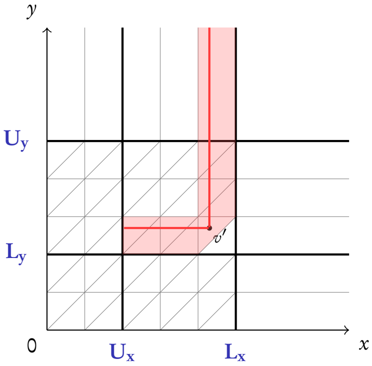
Example



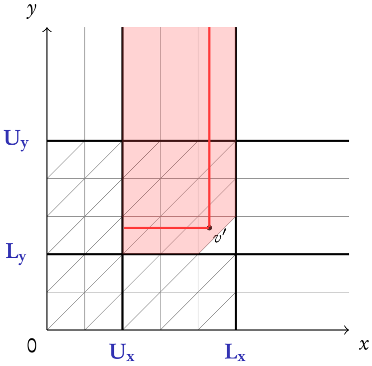
Example



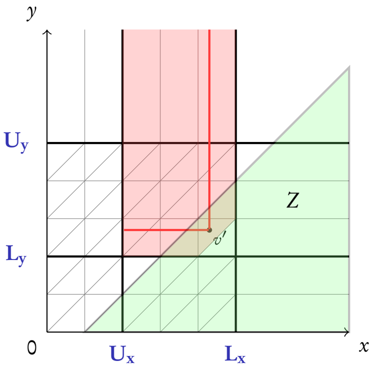
Example



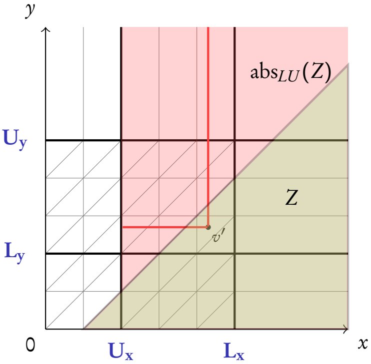
Example

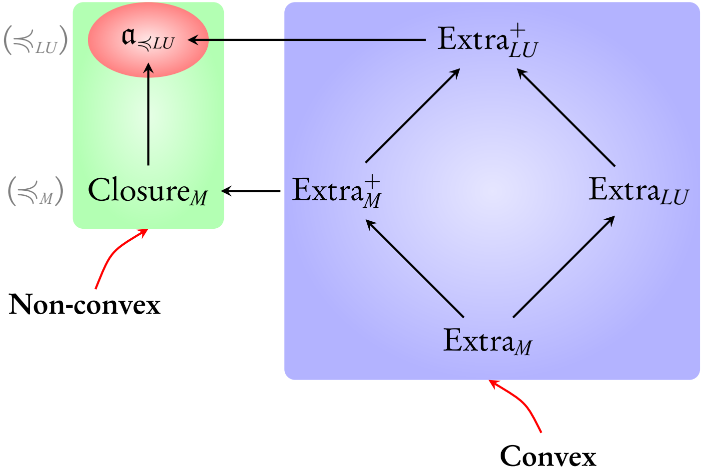


Example



Example





$(\preccurlyeq_{LU})$
$\mathfrak{a}_{\preccurlyeq LU}$
Extra^{+}_{LU}
$(\preccurlyeq_{M})$
$\mathrm{Closure}_M$
Extra^{+}_{M}
Extra_{LU}
Non-convex
Extra_M
Convex

Time-elapsed zone Z: if $v \in Z$, then $v + \delta \in Z$ for all $\delta \in \mathbb{R}_{\geq 0}$

$\mathfrak{a}_{\preccurlyeq LU}$ coincides with abs_{LU}

If Z is time-elapsed, then $\mathfrak{a}_{\preccurlyeq LU}(Z) = \mathrm{abs}_{LU}(Z)$

Better abstractions for timed automata

F. Herbreteau, B. Srivathsan, I. Walukiewicz. *LICS'12*

Time-elapsed zone Z: if $v \in Z$, then $v + \delta \in Z$ for all $\delta \in \mathbb{R}_{\geq 0}$

$\mathfrak{a}_{\preccurlyeq LU}$ coincides with abs_{LU}

If Z is time-elapsed, then $\mathfrak{a}_{\preccurlyeq LU}(Z) = \mathrm{abs}_{LU}(Z)$

Optimality

$\mathfrak{a}_{\preccurlyeq LU}(Z)$ is the **coarsest** abstraction that is **sound** and **complete** for all LU-automata

Better abstractions for timed automata

F. Herbreteau, B. Srivathsan, I. Walukiewicz. *LICS'12*

Step 3:

Efficient inclusion

$$v \sqsubseteq_{LU} v'$$

if

$$\exists \delta' \in \mathbb{R}_{\geq 0} \text{ s.t. } v' + \delta' \in \langle v \rangle^{LU}$$

Definition

$$\mathrm{abs}_{LU}(W) = \{v \mid \exists v' \in W \text{ s.t. } v \sqsubseteq_{LU} v'\}$$

$$v \sqsubseteq_{LU} v'$$

if

$$\exists \delta' \in \mathbb{R}_{\geq 0} \text{ s.t. } v' + \delta' \in \langle v \rangle^{LU}$$

Definition

$$\mathrm{abs}_{LU}(W) = \{v \mid \exists v' \in W \text{ s.t. } v \sqsubseteq_{LU} v'\}$$

Z, Z': time-elapsed zones

$Z \not\subseteq \mathrm{abs}_{LU}(Z')$ iff there **exists** $v \in Z$ s.t.
$\langle v \rangle^{LU}$ **does not intersect** Z'

Efficient inclusion testing

Reduction to two clocks

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

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

Better abstractions for timed automata

F. Herbreteau, B. Srivathsan, I. Walukiewicz. *LICS'12*

Efficient inclusion testing

Reduction to two clocks

$Z \not\subseteq \mathfrak{a}_{\preccurlyeq LU}(Z')$ 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

Better abstractions for timed automata

F. Herbreteau, B. Srivathsan, I. Walukiewicz. *LICS'12*

Efficient inclusion testing

Reduction to two clocks

$Z \not\subseteq \mathfrak{a}_{\preccurlyeq LU}(Z')$ 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 $Z \subseteq Z'$!

Better abstractions for timed automata

F. Herbreteau, B. Srivathsan, I. Walukiewicz. *LICS'12*

Efficient inclusion testing

Reduction to two clocks

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

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

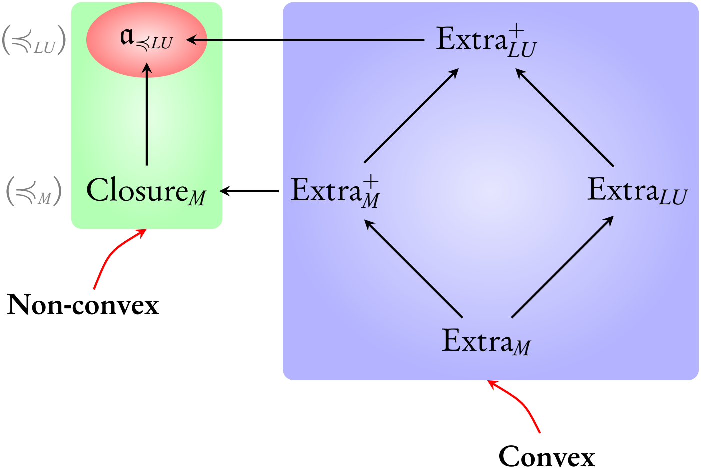
Complexity: $\mathcal{O}(|X|^2)$, where X is the set of clocks

Same complexity as $Z \subseteq Z'$!

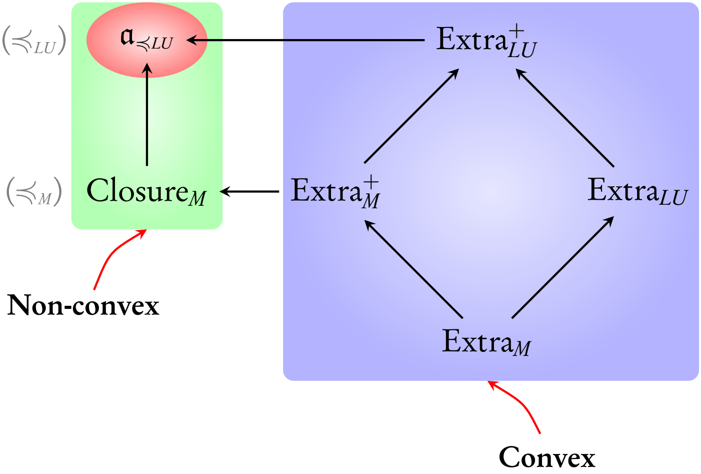
Slightly modified comparison works!

Better abstractions for timed automata

F. Herbreteau, B. Srivathsan, I. Walukiewicz. *LICS'12*



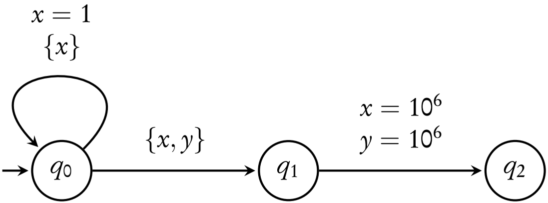
$(\preccurlyeq_{LU})$
$\mathfrak{a}_{\preccurlyeq LU}$
Extra^+_{LU}
$(\preccurlyeq_{M})$
$\mathrm{Closure}_M$
Extra^+_M
Extra_{LU}
Extra_M
Non-convex
Convex



Question: If $\mathfrak{a}_{\preccurlyeq LU}$ is best, can we do better?

Get better **LU-bounds!**

Global LU-bounds

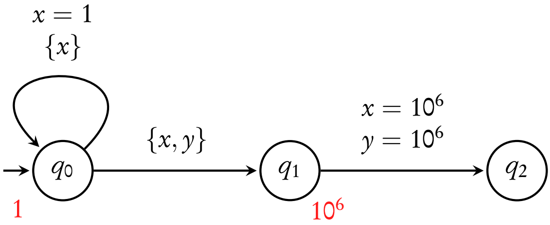


Naive: $L_x = U_x = 10^6$, $L_y = U_y = 10^6$

Size of graph $\sim 10^6$

Static analysis: bounds for every q

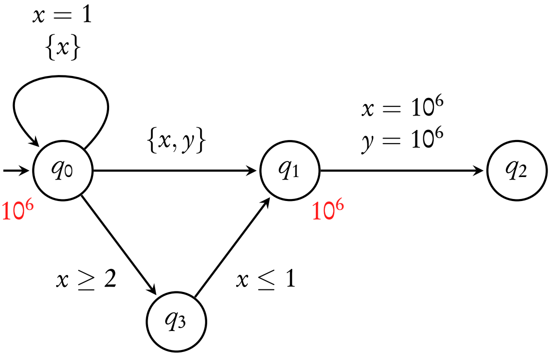
[BBFL03]



Size of graph < 10

Static analysis: bounds for every q

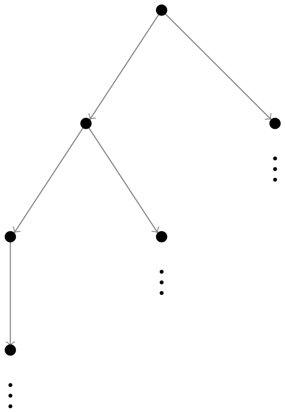
[BBFL03]



Size of graph $\sim 10^6$

Need to look at semantics...

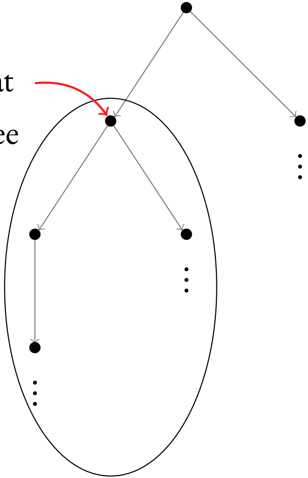
LU bounds for every (q, Z) in zone graph



LU bounds for every (q, Z) in zone graph

constants at

depend on subtree

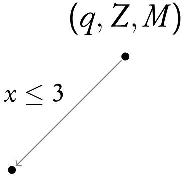


$M(x) = -\infty$

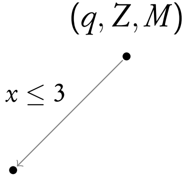
$$(q, Z, M)$$

$\bullet$

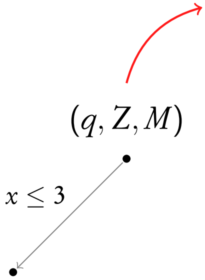
$M(x) = -\infty$



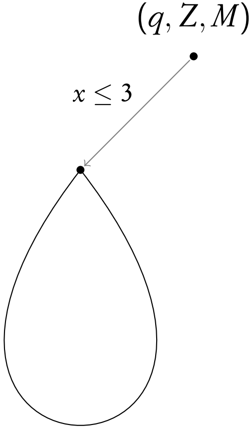
$M(x) = 3$



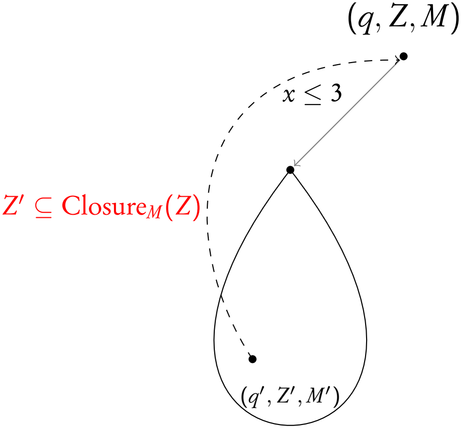
$M(x) = 3$



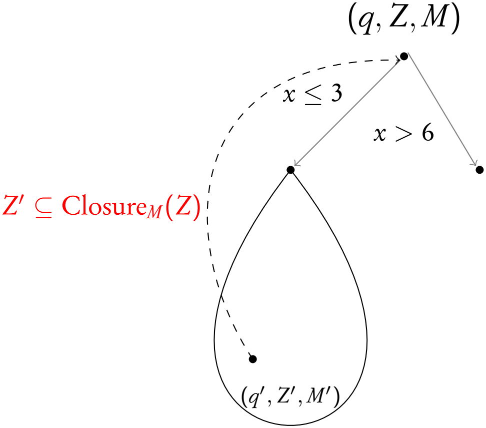
$M(x) = 5$



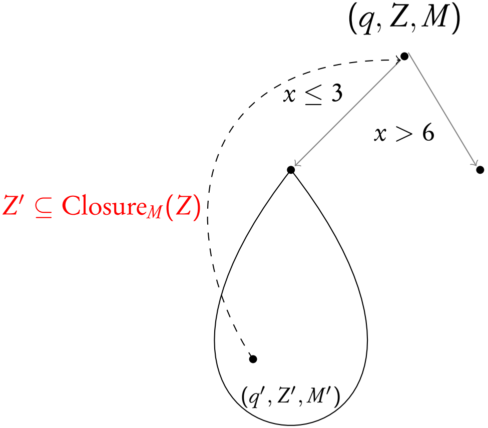
$M(x) = 5$

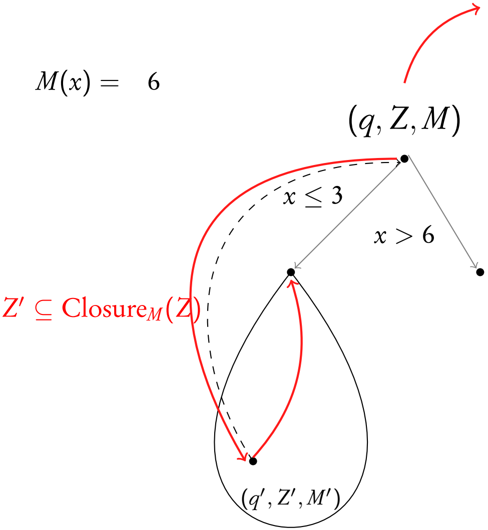


$M(x) = \ 5$

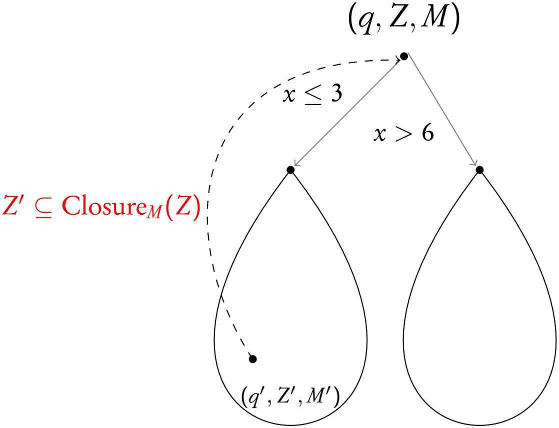


$M(x) = \quad 6$

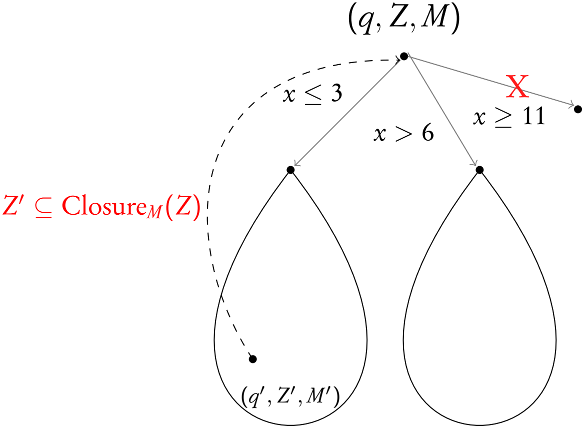



$M(x) = \quad 6$
(q, Z, M)
$x \leq 3$
$x > 6$
$Z' \subseteq \text{Closure}_M(Z)$
(q', Z', M')

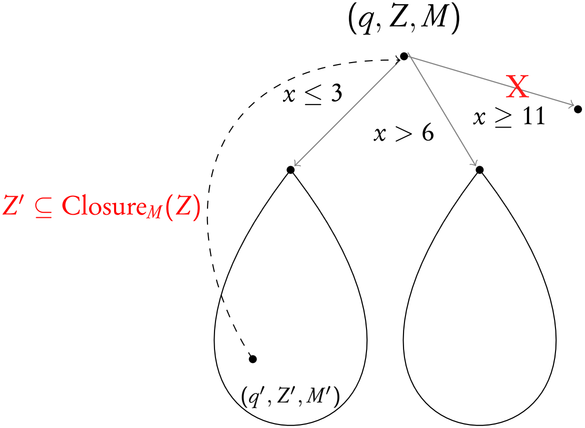
$M(x) = \quad 6$



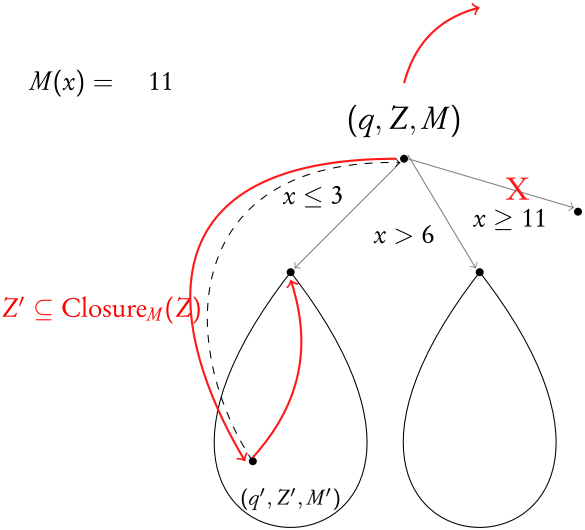
$M(x) = \quad 6$



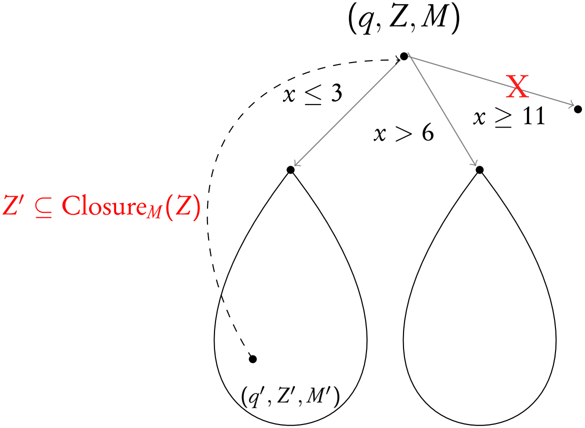
$M(x) = \quad 11$



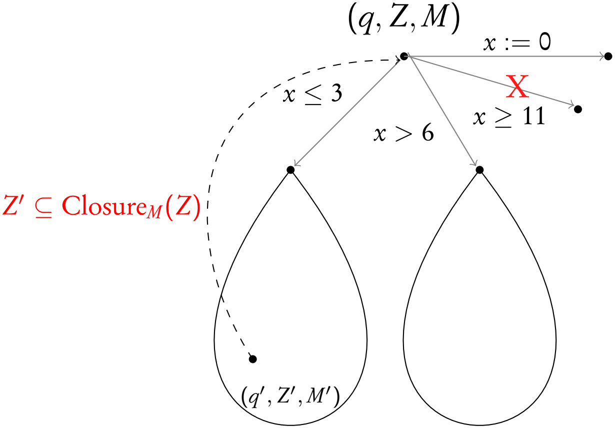
$M(x) = \quad 11$



$M(x) = \quad 11$

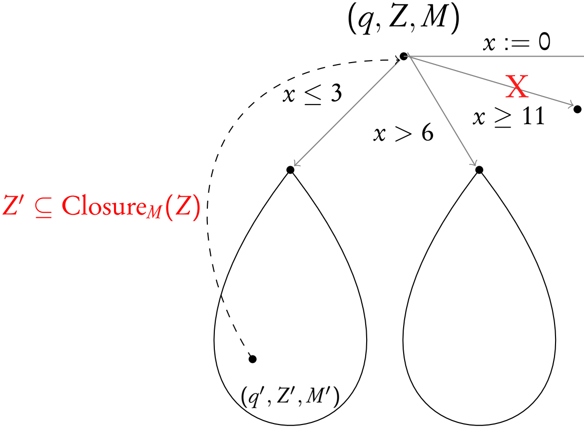


$M(x) = \quad 11$

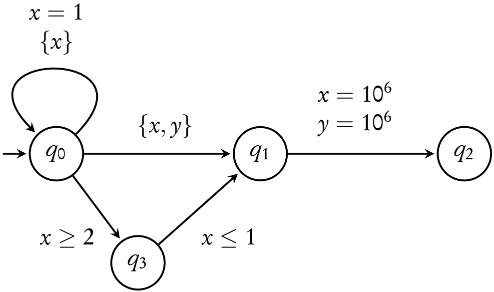


$M(x) = \quad 11$

All **tentative nodes** consistent
+ No more **exploration**
$\rightarrow$ Terminate!



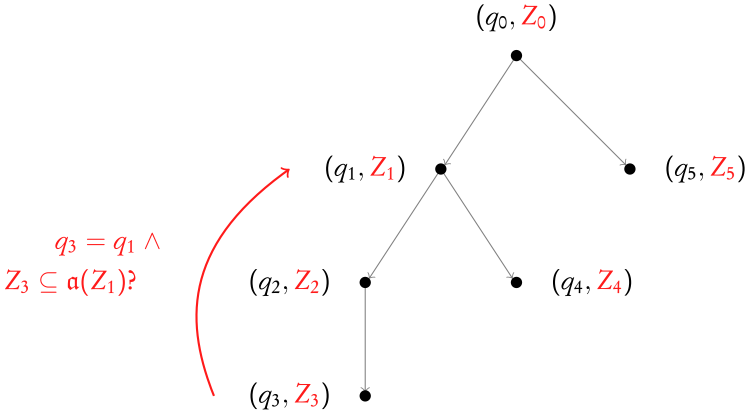
Constant propagation



Theorem (Correctness)

An accepting state is reachable in $\mathcal{A}$ iff the constant propagation algorithm reaches a node with accepting state and a non-empty zone.

Key idea: Compute the **zone graph**, use **abstraction** for termination



Developments are recent, a lot of (not-so-low) hanging fruit available

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.

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.