Automata for Real-time Systems

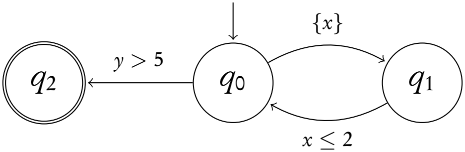
B. Srivathsan

Chennai Mathematical Institute

Lecture 15:

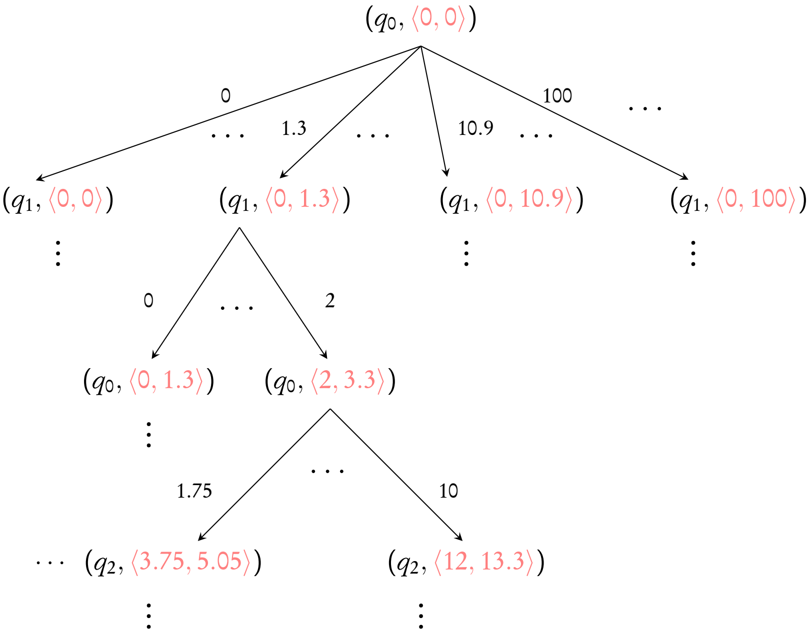
Better abstractions through better constants

Reachability problem



Given a TA, does there **exist** a run to a **final state**?

Main challenge: **infinite** behaviour of timed automata


(q0, ⟨0, 0⟩)
0
1.3
10.9
100
(q1, ⟨0, 0⟩)
(q1, ⟨0, 1.3⟩)
(q1, ⟨0, 10.9⟩)
(q1, ⟨0, 100⟩)
0
2
(q0, ⟨0, 1.3⟩)
(q0, ⟨2, 3.3⟩)
1.75
10
(q2, ⟨3.75, 5.05⟩)
(q2, ⟨12, 13.3⟩)

Abstraction

- **Forget** unnecessary information
- **Retain** essential information

Aim: Get a **finite abstraction**, as **small** as possible

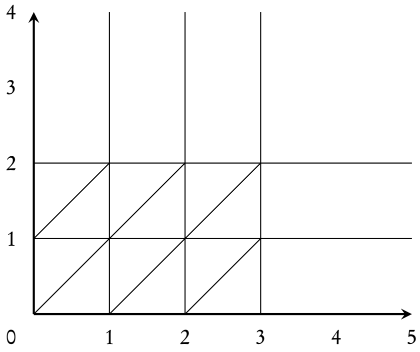
Abstraction

- **Forget** unnecessary information
- **Retain** essential information

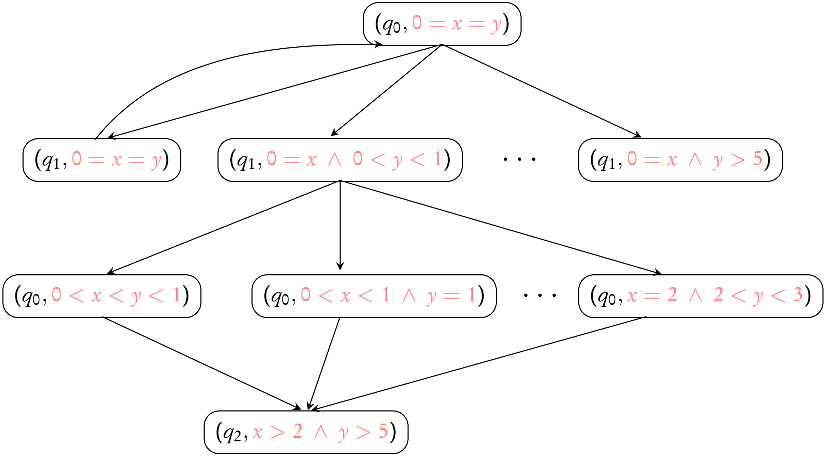
Aim: Get a **finite abstraction**, as **small** as possible

Regions
[AD94]

Maximal bounds: $M : X \mapsto \mathbb{N} \cup \{-\infty\}$



- **Forget:** Exact clock values
- **Retain:**
  1. Integral values upto max
  2. Relative ordering of fractional values for clocks less than max



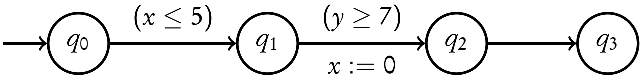
If X is set of clocks, $\mathcal{O}(|X|!\, M^{|X|})$ many regions!

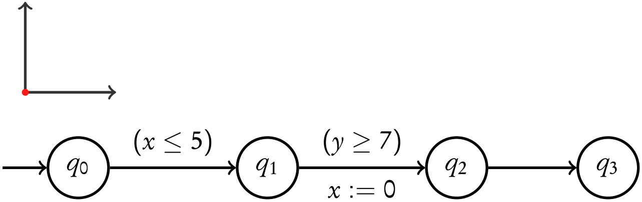
Abstraction

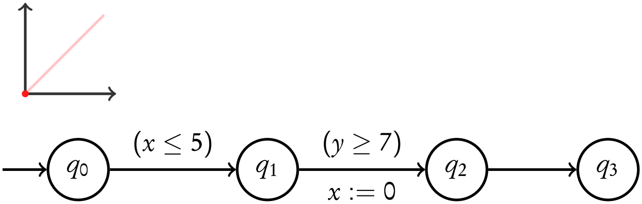
- **Forget** unnecessary information
- **Retain** essential information

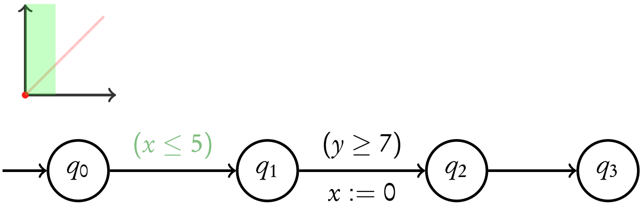
Aim: Get a **finite abstraction**, as **small** as possible

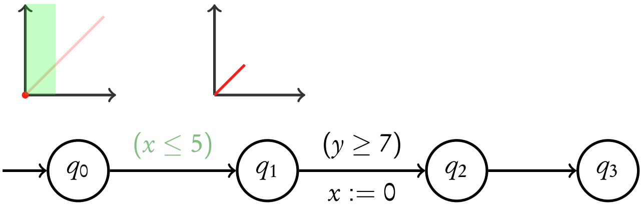
Regions	Zones
[AD94]	[DT98]


$(x \leq 5)$
$(y \geq 7)$
$x := 0$
q_0
q_1
q_2
q_3

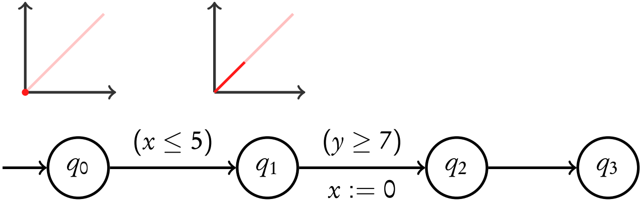

q_0
$(x \leq 5)$
q_1
$(y \geq 7)$
$x := 0$
q_2
q_3

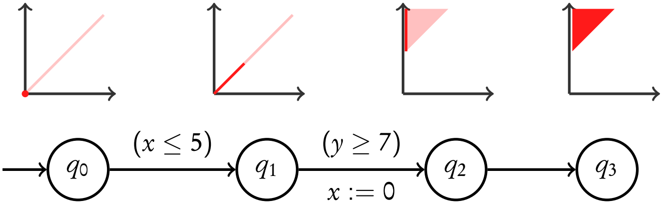

q_0
$(x \leq 5)$
q_1
$(y \geq 7)$
$x := 0$
q_2
q_3


q_0
$(x \leq 5)$
q_1
$(y \geq 7)$
$x := 0$
q_2
q_3

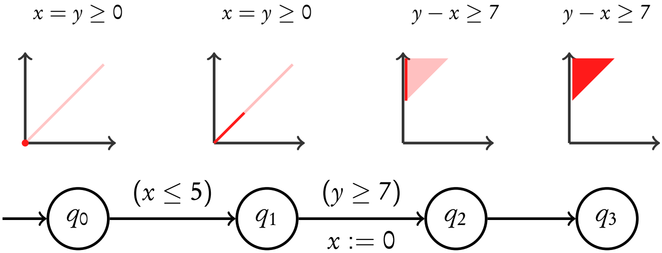


$(x \leq 5)$
$(y \geq 7)$
$x := 0$
q_0
q_1
q_2
q_3

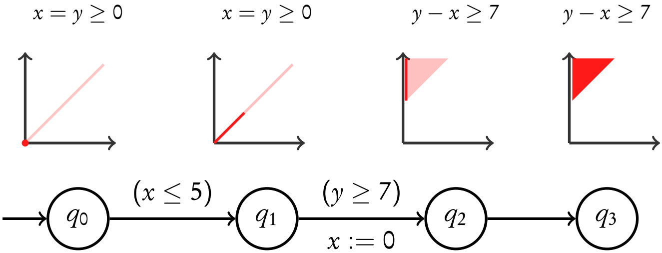

q_0
$(x \leq 5)$
q_1
$(y \geq 7)$
$x := 0$
q_2
q_3



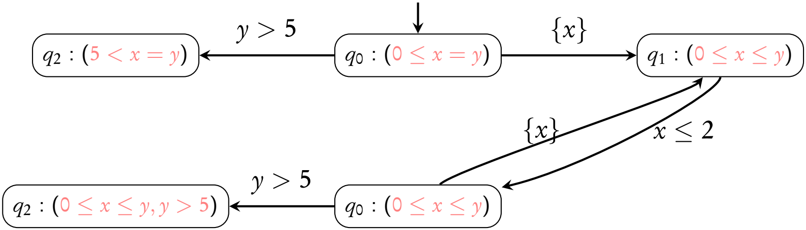
(x ≤ 5)
(y ≥ 7)
x := 0
q0
q1
q2
q3



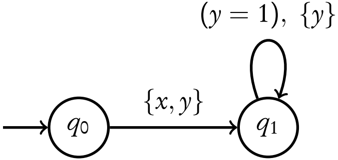
x = y ≥ 0
x = y ≥ 0
y − x ≥ 7
y − x ≥ 7
(x ≤ 5)
(y ≥ 7)
x := 0
q0
q1
q2
q3

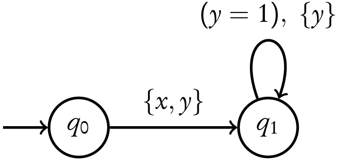


- **Forget:** Exact times taken along a run
- **Retain:** Sequence of discrete transitions

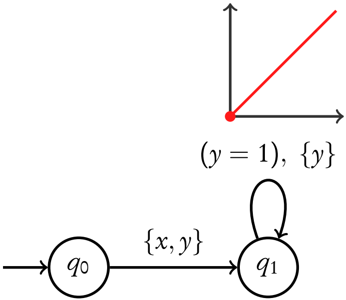


But the zone graph could be **infinite**

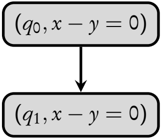

(y = 1), {y}
{x, y}
q0
q1



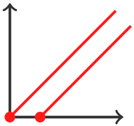
$(q_0, x - y = 0)$

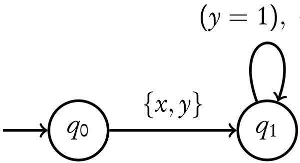


$(y = 1),\ \{y\}$
$\{x, y\}$
q_0
q_1

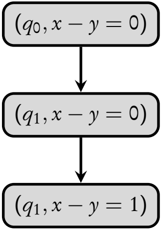


$(q_0, x - y = 0)$
$(q_1, x - y = 0)$

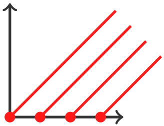


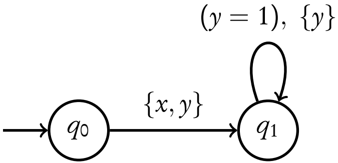


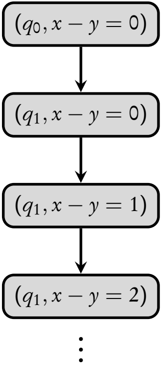
$(y = 1),\ \{y\}$
$\{x, y\}$
q_0
q_1



$(q_0, x - y = 0)$
$(q_1, x - y = 0)$
$(q_1, x - y = 1)$




$(y = 1), \{y\}$
$\{x, y\}$
q_0
q_1


$(q_0, x - y = 0)$
$(q_1, x - y = 0)$
$(q_1, x - y = 1)$
$(q_1, x - y = 2)$
⋮

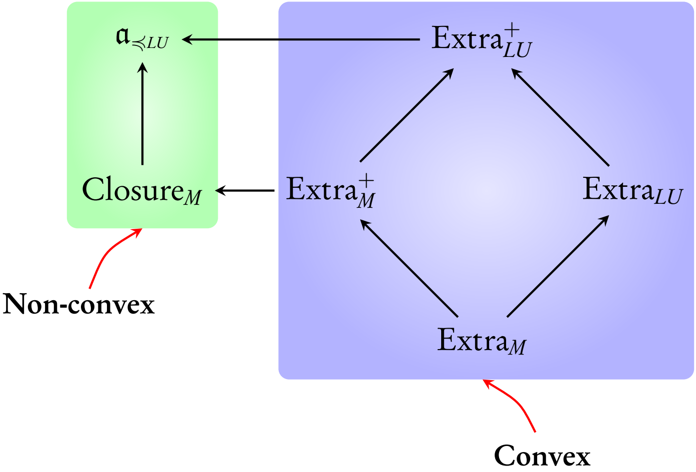
Abstraction

- **Forget** unnecessary information
- **Retain** essential information

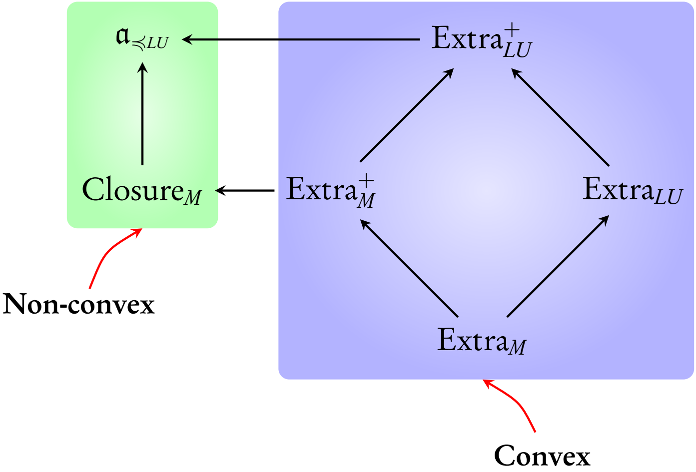
Aim: Get a **finite abstraction**, as **small** as possible

Regions	Zones	Zones + abstraction function
[AD94]	[DT98]	[DT98]
		[BBLP06]
		[HSW12]

Abstraction functions



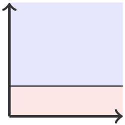
Abstraction functions

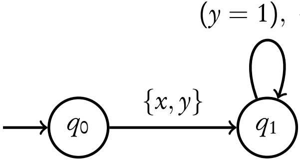


In our course: Closure_M

$M(x) = -\infty$

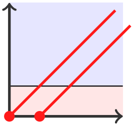
$M(y) = 1$

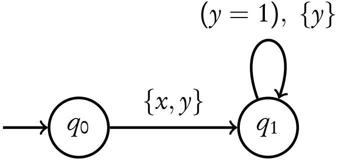




$M(x) = -\infty$

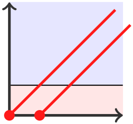
$M(y) = 1$

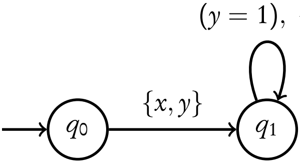


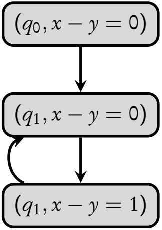


$M(x) = -\infty$

$M(y) = 1$



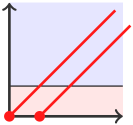


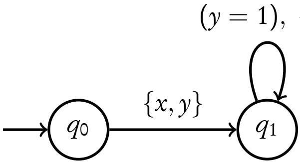


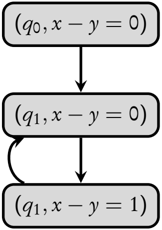
$$x - y = 1 \subseteq \text{Closure}_M(x - y = 0)$$

$$M(x) = -\infty$$

$$M(y) = 1$$







$$x - y = 1 \subseteq \text{Closure}_M(x - y = 0)$$

Using Closure

1. $Z \subseteq \text{Closure}_M(Z')$ can be done **efficiently** [HKSW11] (seen last class)
2. Given M, Closure_M is **optimal** [HSW12] (proof not needed)

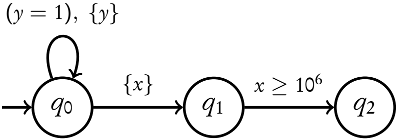
Reachability algorithm:

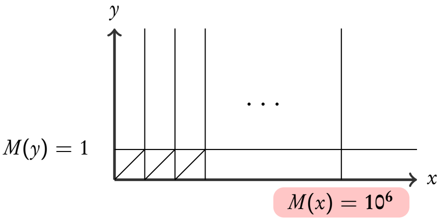
- Compute zones
- Use $Z \subseteq \text{Closure}_M(Z')$ for termination
- Given M, Closure_M is optimal

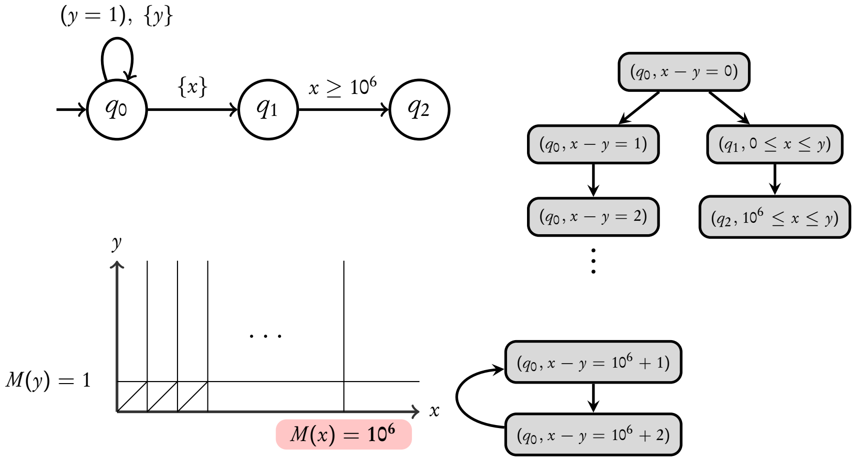
Reachability algorithm:

- Compute zones
- Use $Z \subseteq \text{Closure}_M(Z')$ for termination
- Given M, Closure_M is optimal

Coming next: get **better** M bounds!


$(y = 1),\ \{y\}$
$\{x\}$
$x \geq 10^6$
q_0
q_1
q_2


y
$\cdots$
$M(y) = 1$
x
$M(x) = 10^6$



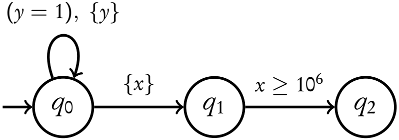
More than 10^6 nodes unnecessary

$$q \rightarrow q_1 \rightarrow \ldots q_i \xrightarrow[\{x\}]{} q_{i+1} \rightarrow \ldots \rightarrow q_n \xrightarrow{x \geq c} q'$$

Constant c is **not relevant** for x at q

Static guard analysis [BBFL03], [UPPAAL]

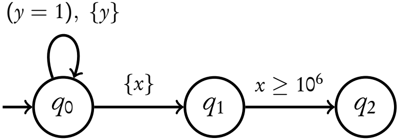
Key idea: Bounds for every q of the automaton



$M_0(x) = -\infty$	$M_1(x) = 10^6$
$M_0(y) = 1$	$M_1(y) = -\infty$

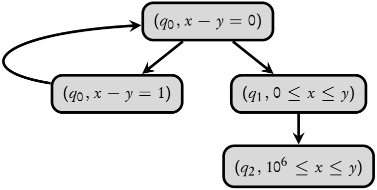
Static guard analysis [BBFL03], [UPPAAL]

Key idea: Bounds for every q of the automaton



$M_0(x) = -\infty \qquad M_1(x) = 10^6$

$M_0(y) = 1 \qquad M_1(y) = -\infty$



More details about static guard analysis on the board

Abstraction

- **Forget** unnecessary information
- **Retain** essential information

Aim: Get a **finite abstraction**, as **small** as possible

Regions	Zones	Zones + abstraction function
[AD94]	[DT98]	[DT98]
		[BBLP06]
		[HSW12]

+ better abstraction parameters [BBFL03, HSW13]

Experiments

Model	nb. of clocks	UPPAAL (-C)		Better abst.	
		nodes	sec.	nodes	sec.
CSMA/CD 10	11	120845	1.9	51210	4.0
CSMA/CD 11	12	311310	5.4	123915	10.2
CSMA/CD 12	13	786447	14.8	294924	25.2
FDDI 50	151	12605	52.9	401	0.8
FDDI 70	211			561	2.7
FDDI 140	421			1121	40.6
Fischer 9	9	135485	2.4	135485	14.8
Fischer 10	10	447598	10.1	447598	56.8
Fischer 11	11	1464971	40.4		
Stari 2	7	7870	0.1	4305	0.4
Stari 3	10	136632	1.7	43269	4.5
Stari 4	13	1323193	26.2	296982	41.5

- **UPPAAL (-C)** shows results from UPPAAL tool which uses static analysis bounds and convex abstraction Extra^{+}_{LU}
- **Better abst.** shows results from the paper [HSW13] that uses non convex abstraction $\mathfrak{a}_{\preccurlyeq LU}$ and a generalization of static guard analysis
- Time out (150s), Memory out (1Gb)

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. G. Larsen, and R. Pelanek.
Lower and upper bounds in zone-based abstractions of timed automata.
Int. Journal on Software Tools for Technology Transfer, 8(3):204–215, 2006.

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.

F. Herbreteau, D. Kini, B. Srivathsan, and I. Walukiewicz.
Using non-convex approximations for efficient analysis of timed automata.
In *Proceedings of FSTTCS*, volume 13 of *LIPIcs*, pages 78–89. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

F. Herbreteau, B. Srivathsan, and I. Walukiewicz.
Better abstractions for timed automata.
In *LICS*, 2012.

F. Herbreteau, B. Srivathsan, and I. Walukiewicz.
Computer aided verification - 25th international conference, cav 2013, saint petersburg, russia, july 13-19, 2013. proceedings.
In *CAV*, volume 8044 of *Lecture Notes in Computer Science*. Springer, 2013.