TIMED AUTOMATA

LECTURE 11

So far in the course:

- Timed automata basics

  i) Timed languages, closure Properties
  ii) Region construction
  iii) Deterministic T.A. Vs. Non-deterministic T.A.
  iv) Undecidability of inclusion
  v) Event-clock automata

Next module:

- Verification

We will study the core algorithm implemented in T.A. tools.

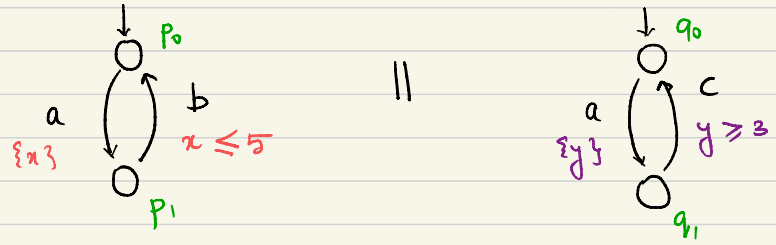
Tools: UPPAAL, TCHECKER

TODAY'S LECTURE

- 1. Network of timed automata
- 2. Introduction to "zone - based" algorithm

Networks of timed automata:

Example 1:

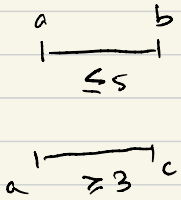


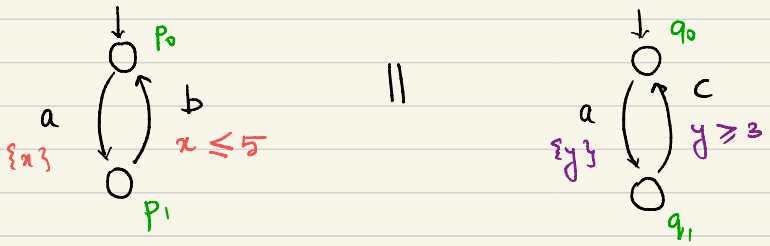
	a	b	c	a	c	b
	0	4	5	10	13	19
p_0	p_1	p_0	p_0	p_1	p_1	p_0
q_0	q_1	q_1	q_0	q_1	q_0	q_0
x	0	4	5	0	3	4
y	0	4	5	0	3	4

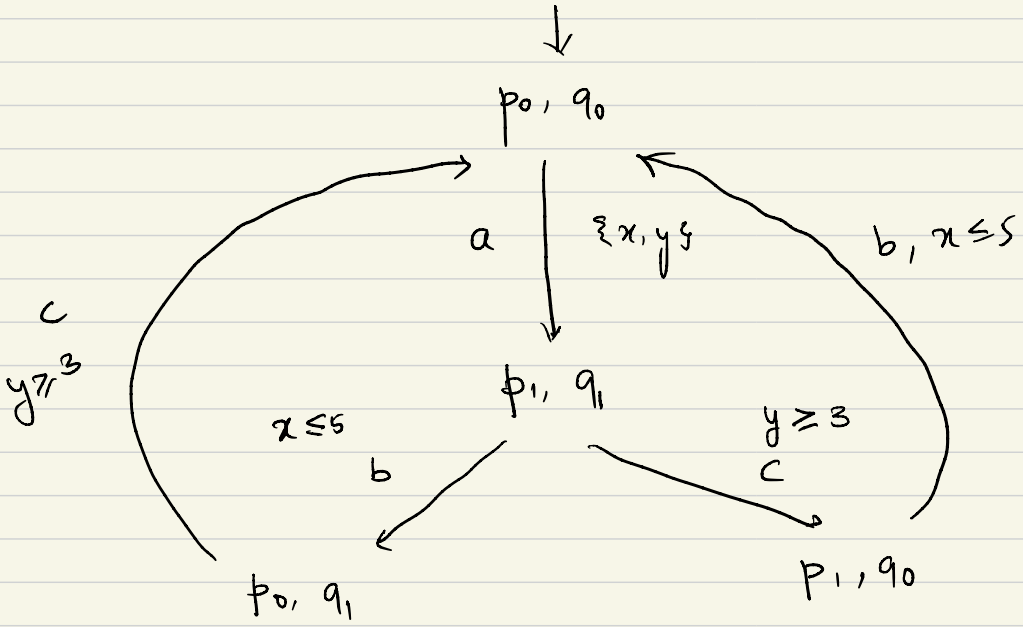
Set of executions of this network are as follows:

$$a \quad bc + cb \quad a \quad bc + cb \quad \cdots \qquad [a\ (bc + cb)]^*$$

Timing constraints:



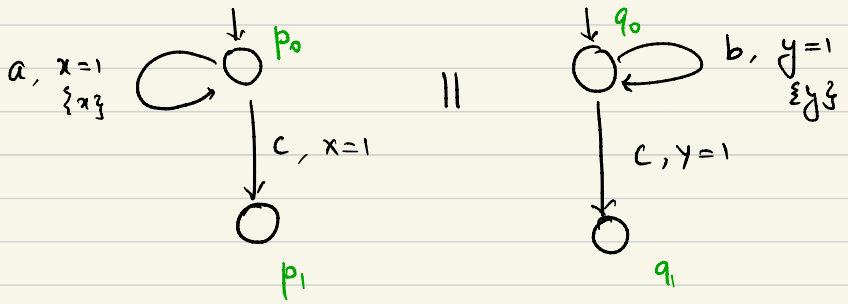




↳ T.A. that represents the semantics of the network

→ Synchronous product.

Example 2:



a a b c
t_1 t_2 t_3 t_4
1 2 $y=2$, not possible to do b.

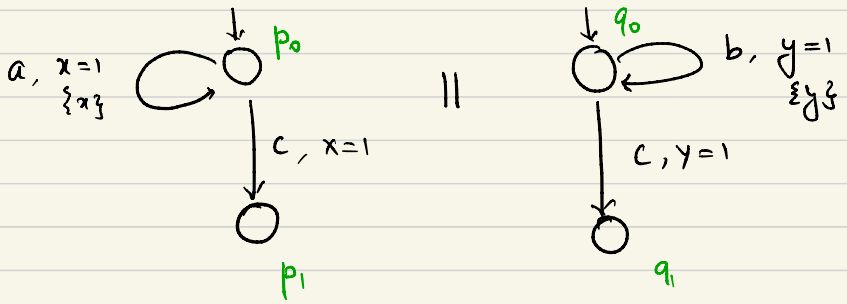
a b a b c
t_1 t_2 t_3 t_4 t_5
1 1 2 2 3

a b b a a b a b b a c
1 1 2 2 3 3 4 4 5 5 6

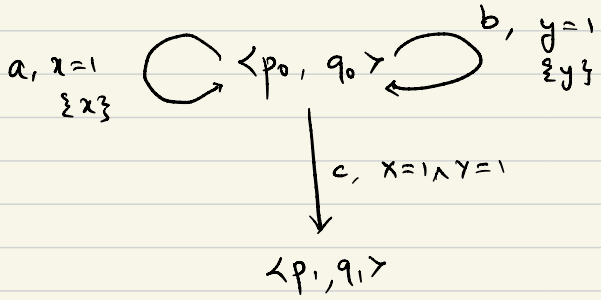
Language of all timed words that reach $\langle p_1, q_1 \rangle$:

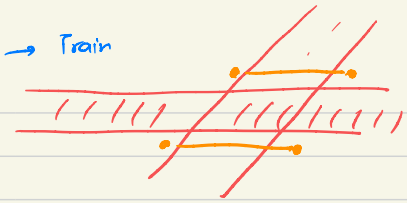
$$(ab + ba)^* c$$

s.t. - the pair 'ab' or 'ba' occurs at the same 't'
- difference between consecutive 'ab' / 'ba' is 1
- difference between 'ab' \ 'ba' and c is 1

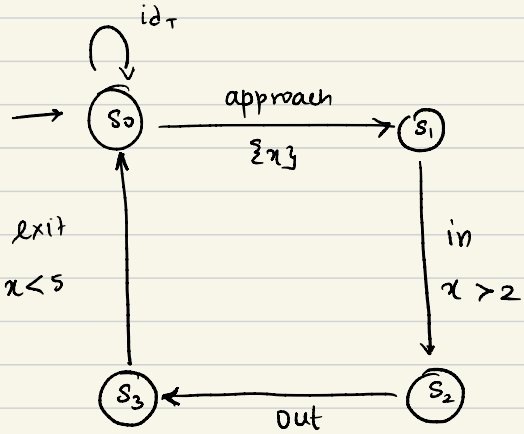


Synchronous product:

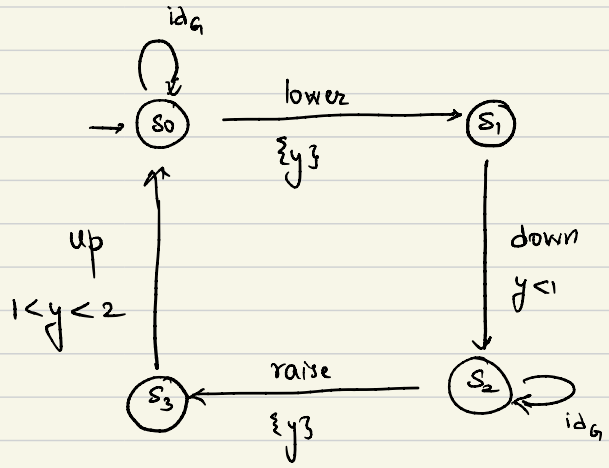




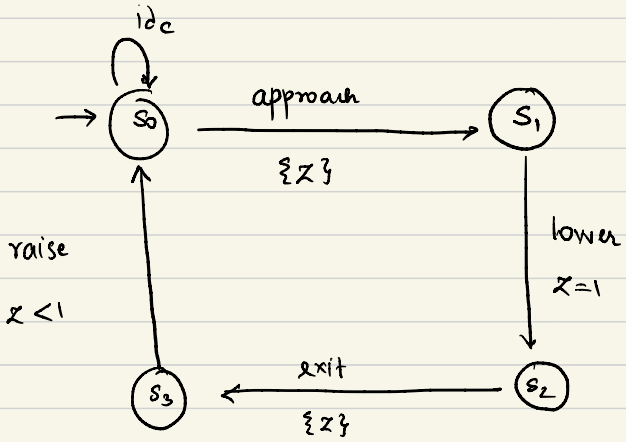
Example 3: From [Alur-Dill '94]



||



||



Specification (property to be checked): Whenever the train is inside the crossing, the gate should be down.

Error state: $\langle s_2, s_0, - \rangle$ $\langle s_2, s_1, - \rangle$, $\langle s_2, s_3, - \rangle$

$\langle s_3, s_0, - \rangle$ $\langle s_3, s_1, - \rangle$ $\langle s_3, s_3, - \rangle$

Property is violated $\Leftrightarrow$ there is a run of the network leading to one of these state

Network of timed automata: syntax:

$$\langle A_1, A_2, \ldots, A_k \rangle$$

each A_i is a timed automaton given by:

$$A_i = (Q_i, \Sigma_i, X_i, Q_i^0, \Delta_i)$$

Assumptions: $X_i \cap X_j = \emptyset$ No shared clocks

$Q_i \cap Q_j = \emptyset$

$A_1 \parallel A_2 \parallel \ldots \parallel A_k$ is a timed automaton:

- States: $Q_1 \times Q_2 \times \ldots \times Q_k$
- Alphabet: $\Sigma = \Sigma_1 \cup \Sigma_2 \ldots \cup \Sigma_k$

  for $a \in \Sigma$, $dom(a) = \{ i \mid a \in \Sigma_i \}$
- Clocks: $X_1 \cup X_2 \cup \ldots \cup X_k$
- Transitions:

$$\langle q_1, q_2, \ldots, q_k \rangle \xrightarrow[R]{a,\ g} \langle q_1', q_2', \ldots, q_k' \rangle$$

if there exist transitions:

$$\{ (p_i, a, g_i, R_i, p_i') \}_{i \in dom(a)}$$

s.t.

- i) $q_i = p_i \quad \forall i \in dom(a)$
- ii) $g = \bigwedge_i g_i \quad i \in dom(a)$
- iii) $R = \bigcup_i R_i \quad i \in dom(a)$
- iv) $q_i' = p_i' \quad \forall i \in dom(a) \qquad q_i' = q_i \quad \forall i \notin dom(a)$

Reachability Problem:

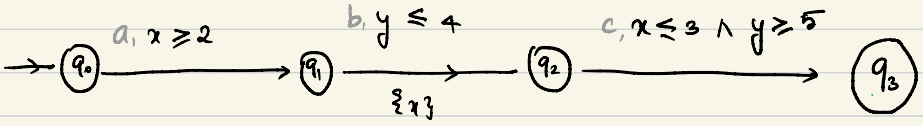
Given a network of T.A., and a state 'q' of the network, does there exist a run that leads to q.

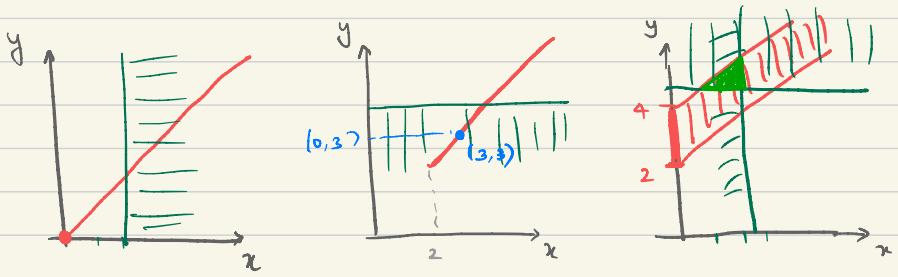
Complexity: PSPACE-complete [Alur-Dill '94], proof via region automata

What we will see next:

- Idea behind the algorithm used in tools.
- We will restrict to single automata instead of networks. The same method can be extended to networks in a straightforward manner by considering the synchronized product.

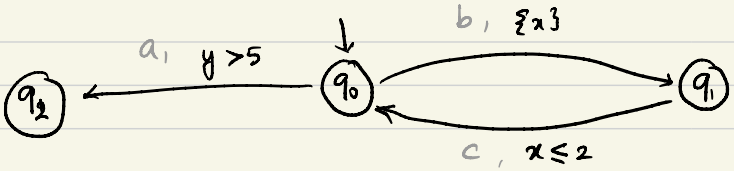
Example:

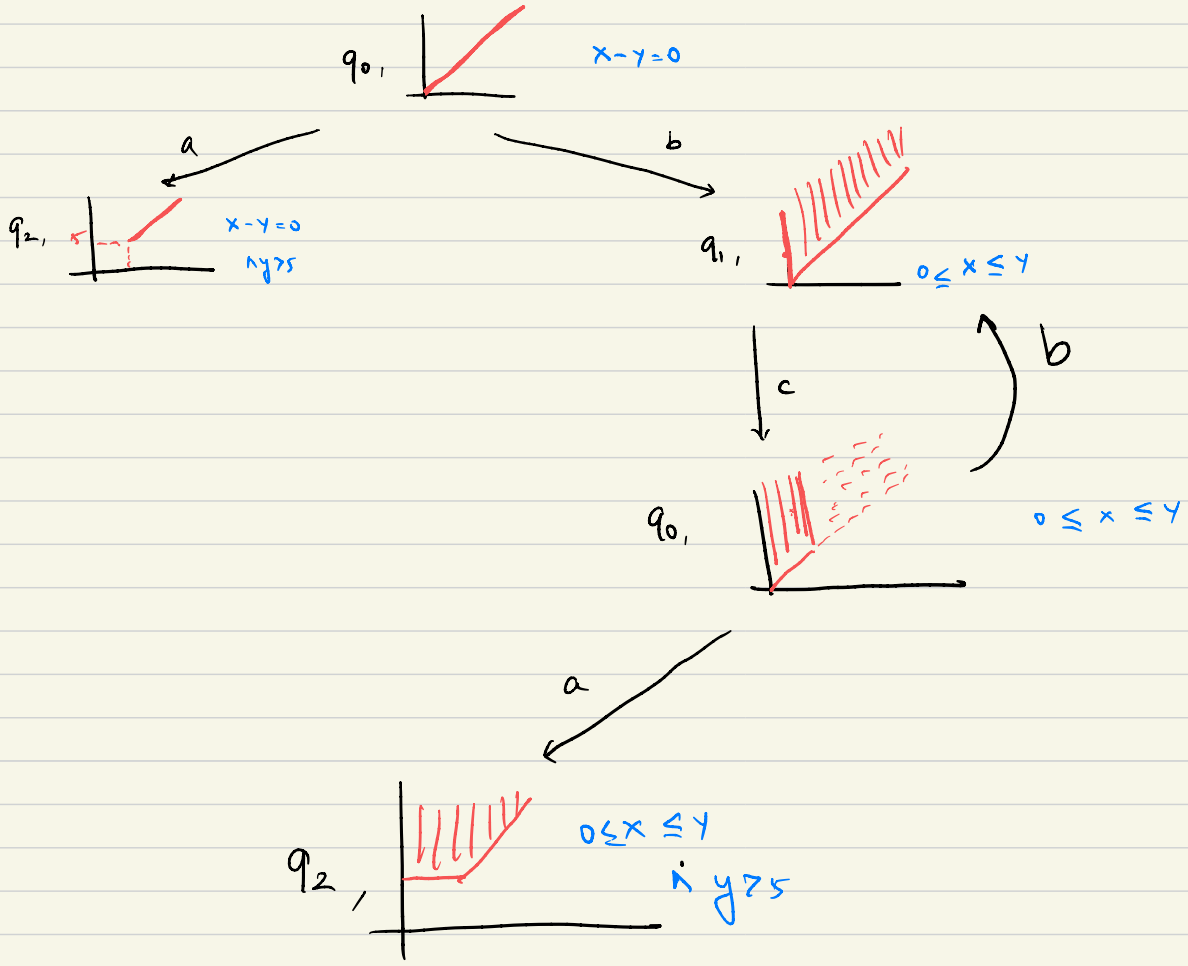




$\langle q_0, x - y = 0 \rangle \rightarrow \langle q_1, x - y = 0 \wedge x \geq 2 \rangle \rightarrow \langle q_2, 2 \leq y - x \leq 4 \rangle$

Example:





Zone graph.

Notice that:

the constraints are special. They involve only difference between clocks or comparison to a constant.

— No $x+y$, $3x-4y$, etc.

Zones:

X

A valuation $v : X \longrightarrow \mathbb{R}_{\geq 0}$

A zone is a set of valuations given by $\wedge$ (conjunctions of) constraints of the following form:

$$x - y \sim c \qquad c \in \mathbb{N}$$

$$x \sim c \qquad \sim \in \{<, \leq, =, >, \geq\}$$

- Graph computed in the previous example will be called the zone graph.

We will see more examples and definitions in the next lecture.

Summary:

1. Network of timed automata
2. An intuition behind zone graph computation