

TIMED AUTOMATA

LECTURE 26

TODAY's GOALS

- Expressiveness of Updatable T.A.

(Recall): updatable Timed automaton (UTA) :

Resets generalized to updates

Updates:

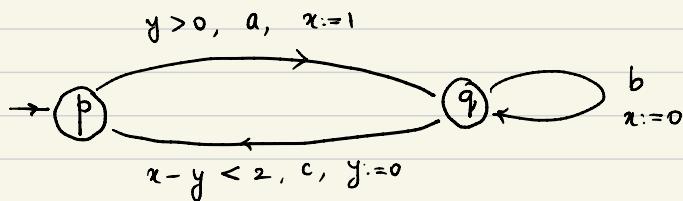
X : a set of clocks

For each clock $x \in X$, an update on x takes the following form:

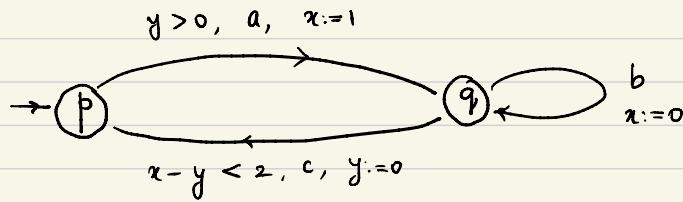
$$x := c \quad | \quad x := y + d \quad c \in \mathbb{N}, d \in \mathbb{Z}, y \in X$$

Examples: $x := 5, x := x - 1, x := y + 2, x := y, x := 2 - 1$

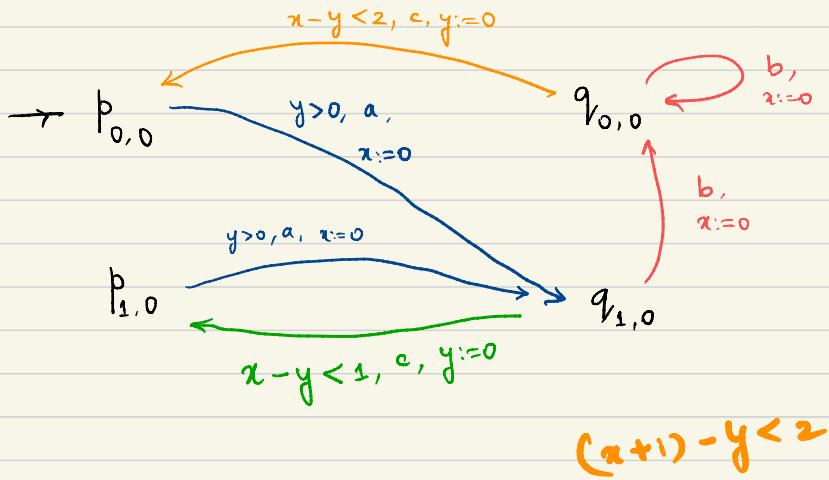
Question 1: Convert the following UTA into a TA:



Solution:



Idea: Remember the last update constant of each clock in the state.
Modify the guards depending on the appropriate constant in the state.



Question 2: Consider VTA restricted to updates of the form

$$x := c \quad (c \in \mathbb{N})$$

Construct an equivalent timed automaton with only resets for this class of VTA.

Solution: Generalizing the previous construction:

Convert a UTA with $x := c$ updates to a TA.

UTA $A = (Q, Q_0, X, \Sigma, \Delta, F)$

We will construct a TA B .

States: Let $S_x := \{c \mid x := c \text{ appears in some transition}\}$

$$S = \prod_{x \in X} S_x$$

$$\begin{aligned} x &= \{3, 5\} \cup \{0\} \\ y &= \{4, 7, 8\} \cup \{0\} \\ S &= (3, 4) \times (5, 7) \times (0, 8) \end{aligned}$$

States of B are of the form: $Q \times S$

Transitions:

$$\text{For every } q \xrightarrow[\text{up}]{a, g} q' \text{ in } A$$

we have a transition from every (q, σ)

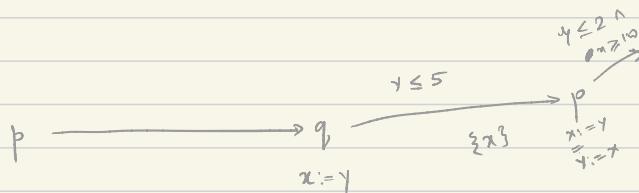
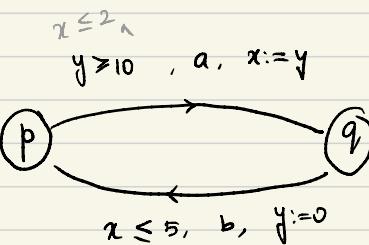
$$(q, \sigma) \xrightarrow[R_{\text{up}}]{a, g'} (q', \sigma')$$

where $g' := g [x \mapsto x + \sigma(x)]$ (replace x with $x + \sigma(x)$)

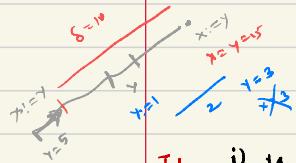
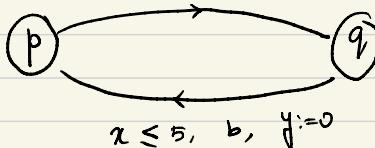
$R_{\text{up}} := \text{Set of clocks updated in up}$

$$\sigma'(x) = c \quad \text{if } x := c \in \text{up}, \text{ else } \sigma'(x) = \sigma(x).$$

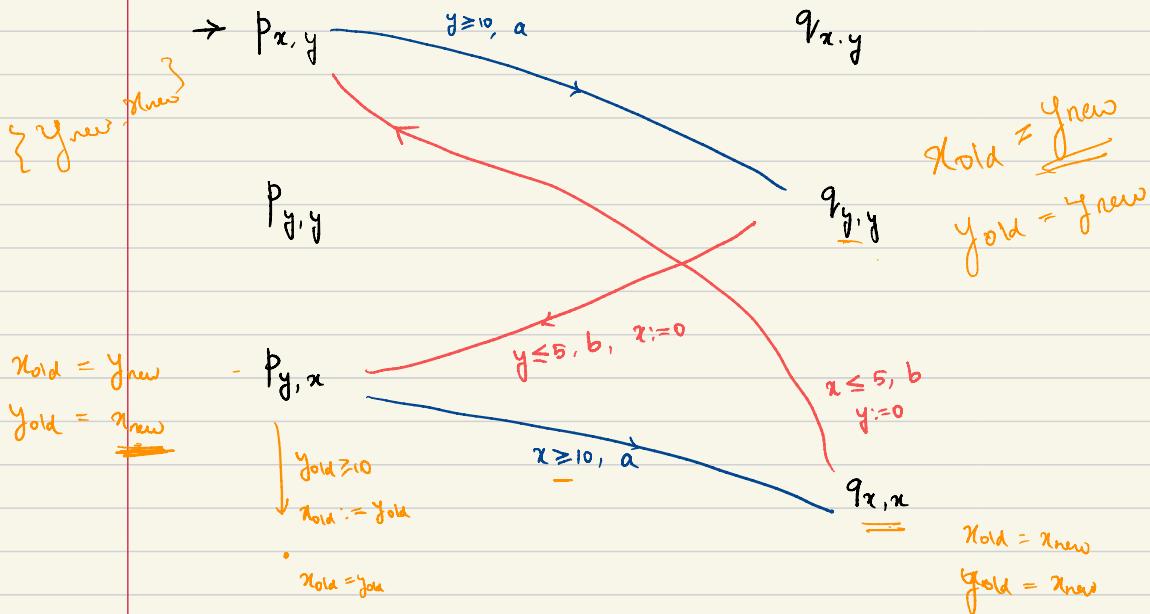
Question 3: Convert the following VTA into a TA:



$$y \geq 10, a, x := y$$

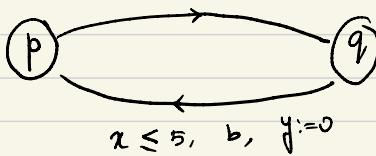


- Idea:
- Maintain the last clock to which each clock was reset to.
 - If value of some clock is stored in some other clock, then that clock should not be updated to a constant.



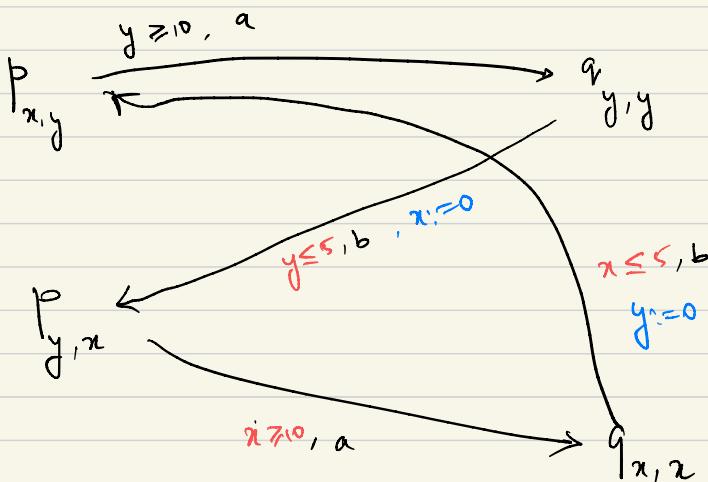
$y \geq 10, a, x := y$

UTA:

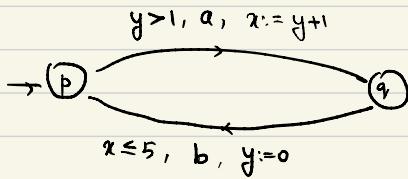


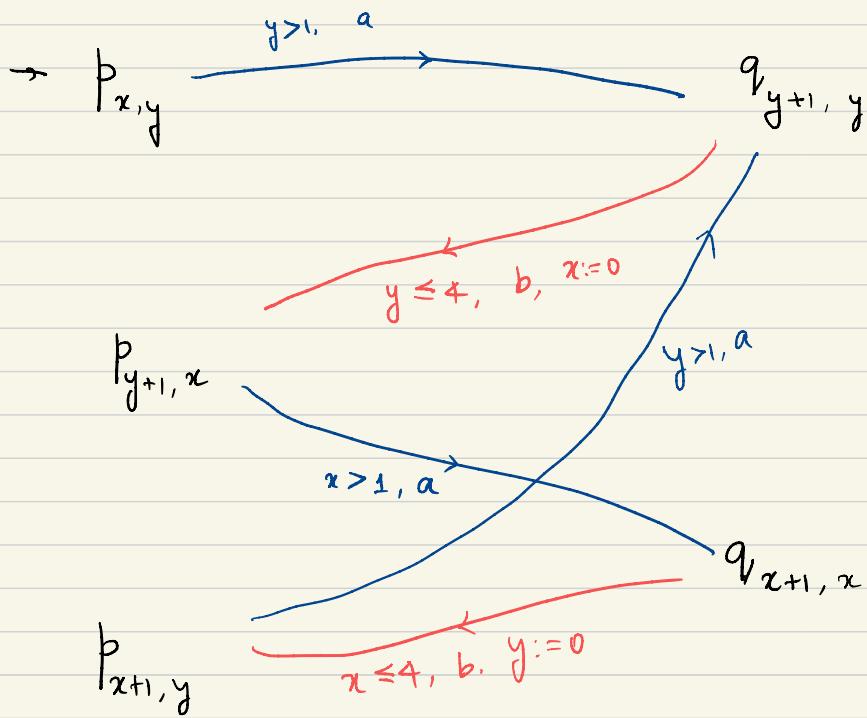
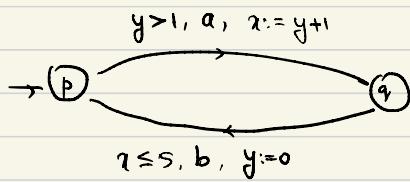
TA:

x, y

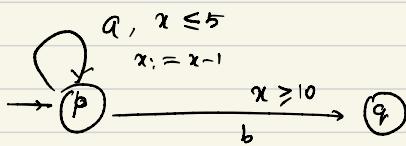


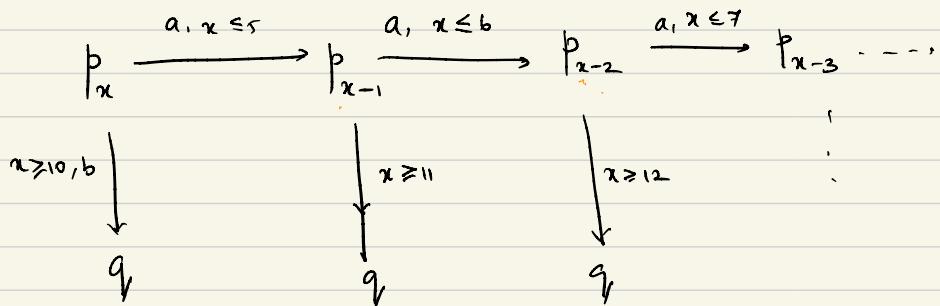
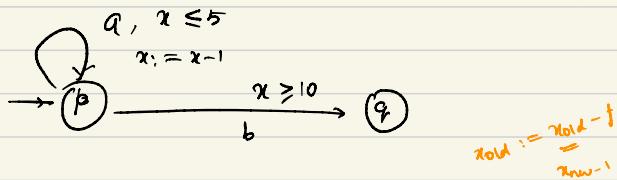
Question 4: Convert the following UTA into a TA:





Question 5: Apply the construction of previous question to the following URA:





Remark 1: This construction does not work always.

The paper on Updatable Timed Automata by Bouyer et al. gives a termination criterion: it generates a system of inequalities; if the system has a solution, then the above construction gives finitely many states.

Remark 2: In our first lecture on UTA we have given an example of a timed language that is accepted by a UTA, but not by TA.

Summary:

Diagonal-free	Diagonals
$x := c, \quad x := y$	TA
$x := y + c$	TA
$x := x - 1$	More expressive

$x := x + 1$
 $y = x$
 $y = (x+1)$
 $y = x^{-1}$