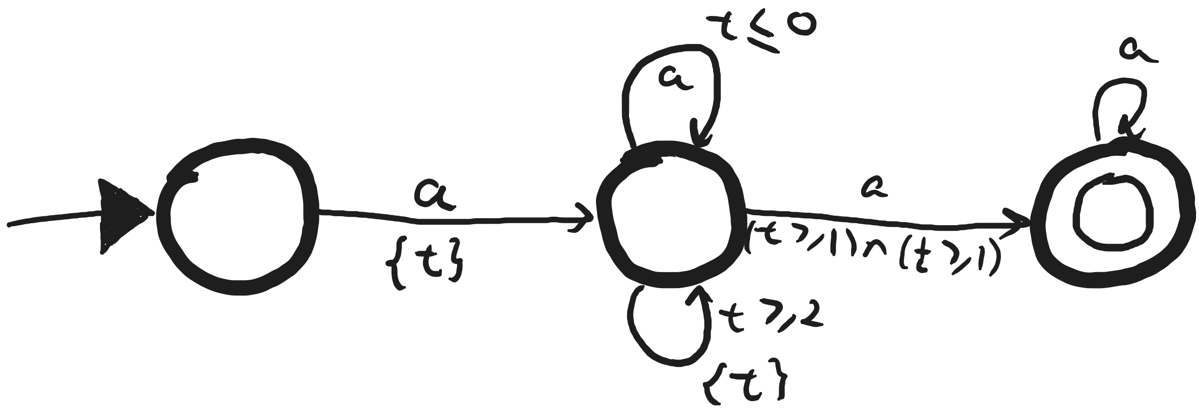
Timed Automata Homework 1

Name: Shubh Sharma
Roll Number: BMC202170

1.



2.

Let

- $Q = S \cup A \cup B \cup C \cup D$ where
  - $S = \{s\}$
  - $A = \{a_0\}$
  - $B = \{b_0, \ldots, b_3\}$
  - $C = \{c_{(i,j)}\}$ where $i, j \in \{0..3\}$
  - $D = \{d_3, \ldots d_6\}$
- $\Sigma = \{a, b, c, d\}$
- $X = \{*\}$
- $Q_0 = S$
- $F = D$

And for the transitions, we have a transition from s to a_0 accepting a and resetting the clock.

We have transitions a_0 to b_i for $i \in \{0..3\}$ accepting b if clock reads i. We reset the clock

We have transitions $b_i \in B$ to $c_{(i,i+t)}$ if $c_{(i,i+t)} \in C$ which accept c if clock reads $t \in \{0..3\}$. We reset the clock. $i + t$ is when c is accepted after a so first condition checks

We have transitions $c_{(i,j)} \in C$ to d_{j+t} if $d_{j+t} \in D$ and $j + t - i \geq 3$ which accept d if the clock reads $t \in \mathbb{N}$. $j + t$ is when d is accepted and i is when b was accepted, so second conditions also checks

3.

Consider a language L, we can construct a timed automaton $\mathcal{A}$ that accepts L, and because of question 4, without loss of generality, assume $\mathcal{A}$ has just 1 clock and let l be one more than the largest number in any guard. And that the **clock is reset on every transition**.

Since the clock resets on every transition. If there is a transition, from a state whose guard is satisfiable. Then we keep the transition otherwise we remove it. As, given a state and such a transition, we can always wait for a suitable amount of time and take the transition.

Construct a new timed automaton $\mathcal{A}'$ where we remove everything related to clocks, it becomes a standard finite state automaton. This automata accepts $\mathrm{Untime(L)}$ as for any run in $\mathcal{A}$, there is a corresponding run in $\mathcal{A}'$ accepting the same word. And for any run in $\mathcal{A}'$, we can look at the corresponding transitions in $\mathcal{A}$, which can always be taken because the clock always resets.

4.

Suppose we are given a Discrete Timed Autotmaton $\mathcal{A}$ such that $|X| = n$. let l be one more than the largest number occurring in any guard.

We construct a new automata $\mathcal{A}'$ such that

- $Q' = Q \times \{0, 1, 2 \ldots l\}^n$
- $\Sigma' = \Sigma$
- $X' = \{*\}$

- $Q'_0 = \{(q, \underbrace{0, \ldots 0}_{n \text{ 0s}}) \ : \ q \in Q_0\}$
- $F'_0 = \{q' : \ \text{ such that } q \text{ is the first component of } q' \text{ and } q \in F_0\}$
- T' = all tuples (q', g', a, R', q'_1) for all $t \in \{0..\,l\}$ where
    - $q' = (q, a_1, a_2, \ldots a_n)$
    - $R' = \{*\}$
    - $q''_1 = (q_1, b_1, b_2 \ldots b_n)$ where $b_i = \min\{l, a_i + t\}$
    - We get g' by replacing clock i with b_i in g
    - if replacing clock i with b_i in g satisfies it
    - $q' = (q_1, c_1, \ldots c_n)$ where $c_i = 0$ if clock i is in R and b' otherwise.

Here we have constructed a discrete timed automaton with just 1 clock, that is $*$. Since none of the guards can differentiate between number $\geq l$ we represent all of them by l. This gives a finite amount of configurations represented by clocks.