The goal of this homework is to understand *discrete-timed automata*. In class, we assumed that timestamps of events are non-negative reals. In the discrete time case, we impose a (rather natural) restriction that all timestamps are natural numbers.

Notation: $\mathbb{N}$ denotes the set of natural numbers; Σ is a finite alphabet; X is a finite set of clocks; for a set S, we will write 2^X to denote its power set. The set of *guards* ϕ over clocks X is given by the grammar:

$$\phi := x \leq c \mid x \geq c \mid \phi \wedge \phi$$

where $c \in \mathbb{N}$ and $x \in X$. Note that we do not need comparisons of the form $x < c$ or $x > c$ in the discrete-time case, as they can be replaced with $x \leq c-1$ or $x \geq c+1$ respectively. We will write $\Phi(X)$ for the set of all guards over X.

Definition 1 (Discrete-timed words, discrete-timed languages) A *discrete-timed word* over Σ is a finite sequence of the form $(a_1, t_1)(a_2, t_2)\cdots(a_k, t_k)$ where each $a_i \in \Sigma$, each $t_i \in \mathbb{N}$ and $t_i \leq t_j$ whenever $i < j$. Let $\mathbb{N}\Sigma^*$ denote the set of all discrete-timed words. A *discrete-timed language* is a subset of $\mathbb{N}\Sigma^*$.

Definition 2 (Discrete-timed automata) A *discrete-timed automaton* $\mathcal{A}$ is a tuple $(Q, \Sigma, X, Q_0, T, F)$ where Q is a finite set of states, $Q_0 \subseteq Q$ is the set of initial states, $F \subseteq Q$ is a set of final states and $T \subseteq Q \times \Phi(X) \times \Sigma \times 2^X \times Q$ is a set of transitions: each transition is of the form (q, g, a, R, q') where g is a guard and R is a reset.

Definition 3 (Runs) Let $\mathcal{A} = (Q, \Sigma, X, Q_0, T, F)$ be a discrete-timed automaton. A configuration of $\mathcal{A}$ is a pair (q, v) where $q \in Q$ is a state and $v \in \mathbb{N}^{|X|}$ is a $|X|$-tuple of natural numbers, called a *valuation*. A run of the automaton on a word $(a_1, t_1)(a_2, t_2)\cdots(a_k, t_k)$ is a sequence:

$$(q_0, v_0) \xrightarrow{\Delta_1} (q_1, v_1) \xrightarrow{\Delta_2} (q_2, v_2) \xrightarrow{\Delta_3} \cdots \xrightarrow{\Delta_k} (q_k, v_k)$$

where (q_i, v_i) are configurations, $q_0 \in Q_0$ is an initial state, v_0 assigns 0 to every clock, each $\Delta_i := (q_{i-1}, g_i, a_i, R_i, q_i)$ is a transition in T such that:

$$v_{i-1} + (t_i - t_{i-1}) \text{ satisfies guard } g_i \quad \text{and}$$
$$\text{for all } x \in X : v_{i+1}(x) = 0 \text{ if } x \in R \text{ and } v_{i+1}(x) = v_i(x) \text{ otherwise}$$

A run is *accepting* if $q_k \in F$. A discrete-timed word w is accepted by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on w. The *language* of $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$, is the set of words accepted by $\mathcal{A}$.

Definition 4 (Determinism) A discrete-timed automaton $\mathcal{A} = (Q, \Sigma, X, Q_0, T, F)$ is said to be *deterministic* if $|Q_0| = 1$ and for every pair of transitions (q, g_1, a, R_1, q_1) and (q, g_2, a, R_2, q_2) emerging from a state q on letter a, we have $g_1 \wedge g_2$ to be unsatisfiable (that is, there is no common solution to g_1 and g_2).

Questions:

1. Let $\Sigma = \{a\}$, and let $L_1 = \{(a, t_1)(a, t_2) \cdots (a, t_k) \in \mathbb{N}\Sigma^* \mid k \geq 2$ and there exist i, j s. t. $t_j - t_i = 1\}$. Give a deterministic discrete timed automaton for L_1.

2. Give a *one clock* discrete-timed automaton for the language $L_2 = \{(a, t_1)(b, t_2)(c, t_3)(d, t_4) \mid t_3 - t_1 \leq 3$ and $t_4 - t_2 \geq 3\}$.

3. Define $\text{Untime}((a_1, t_1)(a_2, t_2) \cdots (a_k, t_k)) = a_1 a_2 \cdots a_k$ and for a set of discrete-timed words L, define $\text{Untime}(L) = \{\text{Untime}(w) \mid w \in L\}$.

   Show that if L is accepted by a discrete timed automaton, the language $\text{Untime}(L)$ is regular.

4. Show that every discrete timed automaton can be converted into a language equivalent one-clock discrete timed automaton (in other words, one clock is sufficient to capture all languages accepted by a discrete-timed automaton).