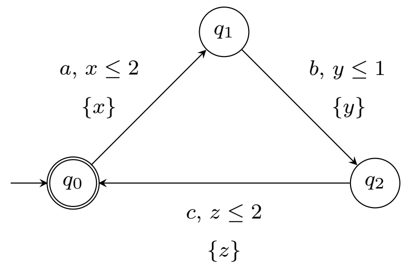
1. Let $\mathcal{B}$ be the following timed automaton:



Consider the timed word $s = (abcabc, 0.5, 1, 1.5, 1.8, 1.9, 3)$.

a) Does $\mathcal{B}$ accept s? If so, write down the accepting run of $\mathcal{B}$ on s.

b) For a timed word (w, τ) we define the *time span* of (w, τ) to be the time at which the last letter occurs, i.e., if $|w| = n$, then time span of (w, τ) is τ_n.
For every $k \in \mathbb{N}$, give a timed word in $\mathcal{L}(\mathcal{B})$ that has length greater than k and whose time span is lesser than 1.

2. Are the following timed languages timed regular? Justify.

i. $L_1 \ = \ \{\ (a^k, \tau) \mid \tau_{i+2} - \tau_i \leq 1 \text{ for all } i \leq k-2\}$

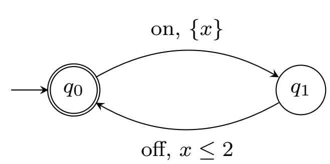
ii. $L_2 \ = \ \{\ (w, \tau) \mid w \in (a+b)^*,\ \exists i \text{ s.t. } w_i = a \text{ and } \forall j \text{ we have } \tau_j \neq \tau_i + 1\}$

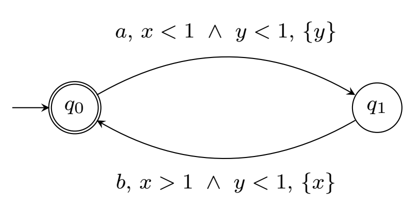
iii. $L_3 \ = \ \{\ (w, \tau) \mid w \in (a+b)^*,\ \exists i \text{ s.t. } w_i = a \text{ and } \forall j \text{ we have } \tau_j \neq \tau_i - 1\}$

3. Consider an automaton with 2 clocks $\{x, y\}$. Let the maximum bounds function M for the automaton be given by: $M(x) = 3, M(y) = 4$. Draw the division of the xy-plane into regions.

4. Let R be a region over clock set X and bound function M. Give an algorithm to compute the time-successors of a region R.

5. Draw the region graph for the following automata:





6. Suppose R is a region over clock set X and bound function M. Let x, y be two arbitrary clocks in X. Is the projection of R on to the xy-plane a region over $\{x, y\}$ with the bounds function M restricted to x and y?

7. Determinize the event recording automaton over the alphabet $\{a, b\}$ illustrated in Figure 1:

8. Give an example of a timed automaton whose underlying graph is connected, but no accepting state is reachable.

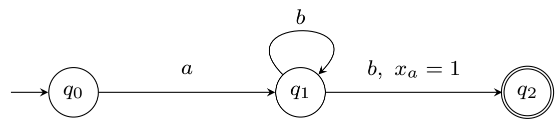


Figure 1: An event-recording automaton

9. Can you give an example of a timed automaton whose underlying graph is connected, but no accepting state is reachable, and whose only guards are upper bound guards: that is guards of the form $x < c$ or $x \leq c$ where $c \in \mathbb{N}$? Note that there could be multiple upper bound guards with different constants.

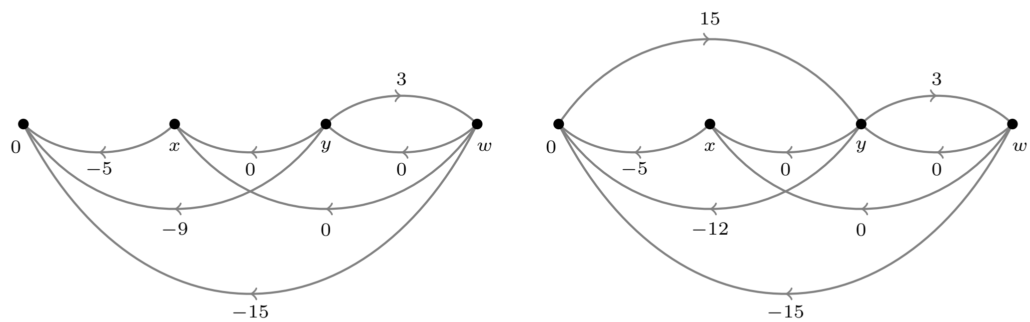
10. The same question as above but now assume that the automaton has only lower bound guards: that is, guards of the form $x \geq c$ or $x > c$.

11. Let Z be the zone defined by $-3 \leq y - x \leq 4$. Can you construct an automaton whose zone graph contains a node (q, Z)?

12. Provide an example of an automaton with a single state, $n \geq 2$ clocks and with $M_x = 1$ for each clock x, such that the zone graph computed by Algorithm 1.3 (in notes of Lecture 6) gives at least 2^n nodes.

13. Let $\mathcal{A}$ be an automaton with a single clock. Are simulation relations (from Notes of Lecture 6) necessary? Is there an example of such an automaton for which Algorithm 1.1 (from Notes of Lecture 6) does not terminate?

14. Which of the following distance graphs are in canonical form? If not, canonicalize them.



15. Is the set of solutions represented by the above graphs non-empty?

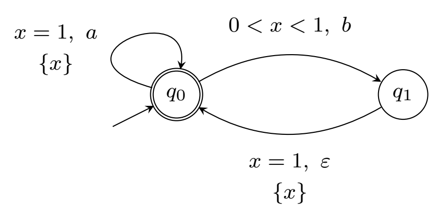
16. Recall that for a set of clocks X, we denote the set of all clock valuations by $\mathbb{R}^X_{\geq 0}$. Also, for a valuation v and a clock x, we denote by $[x]v$ the valuation obtained by resetting x from v. Example: let $X = \{x_1, x_2, x_3\}$ and let $v = \langle 0.5, 23, 47 \rangle$. Then $[x_2]v = \langle 0.5, 0, 47 \rangle$.

    For a zone Z and a clock x, define the $\text{Unreset}(Z, x)$ operation as follows:

    $$\text{Unreset}(Z, x) := \{v \in \mathbb{R}^X_{\geq 0} \mid [x]v \in Z\}$$

    Give an algorithm that takes as input the canonical distance graph G_Z of Z and the clock x, and outputs the canonical distance graph of $\text{Unreset}(Z, x)$.

17. (a) What is the language accepted by the following automaton?



(b) Draw the automaton (if needed with ε-transitions) for the language over $\Sigma = \{a, b\}$ given by:

$$\{\ (w,\tau) \mid w \in \Sigma^*,\ \forall i \le |w| :\ w_i = a \text{ implies } \tau_i \text{ is an integer and } w_i = b \text{ implies } \tau_i \text{ is not an integer}\}$$

where w_i denotes the i^{th} letter in the word w and τ_i denotes the corresponding time-stamp.

18. We know that the universality problem for one-clock timed automata is decidable. Suppose we consider timed automata with multiple clocks, but restrict guards to contain only the constant 0; that is, for a set of clocks X, the guards come from the set $\Phi_0(X)$ defined inductively as

$$\Phi_0(X) \ := \ x = 0 \mid x > 0 \mid \Phi_0(X) \ \wedge \ \Phi_0(X)$$

where $x \in X$. For instance, if $y, z \in X$, then $y = 0 \wedge z > 0$ is a guard in $\Phi_0(X)$.

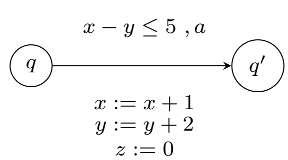
Show that the universality problem is decidable for timed automata with multiple clocks, but whose guards come only from the set $\Phi_0(X)$.

19. Let us add an extra feature to the timed automaton model. Suppose in addition to resets that set a clock to 0, we also allow resetting a clock to 1: that is, each transition is of the form (q, a, g, R_0, R_1, q') where g is the guard, R_0 is the set of clocks that have to be reset to 0 and R_1 is the set of clocks that need to be reset to 1 (assume that $R_0 \cap R_1 = \emptyset$ in every transition).

Let TA_{+1} denote the set of timed automata that have these special resets to either 0 or 1.

Show that this extra feature does not add expressive power to the model. In other words, prove that for every automaton $\mathcal{A}$ in TA_{+1} there exists a normal timed automaton $\mathcal{B}$ that has resets only to 0, such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$.

20. Consider timed automata *with diagonal constraints*. Suppose we extend this model to allow more complicated resets of the form: $x := x + c$ where x is a clock and c is a natural number, in addition to the traditional resets that assign a subset of clocks to 0. For example, the following figure shows a transition in such an automaton:



The transition from q to q' has the diagonal guard $x - y \le 5$. Once the transition is taken, the value of x is increased by 1 from its current value, the value of y is increased by 2 and the value of z is set to 0 (the normal reset). More formally, each transition is of the form (q, a, g, R, q') where R is a function that maps each clock x to either 0 or $x + c$, where $c \in \mathbb{N}$.

Let $\text{TA}^d_{x:=x+c}$ denote the set of timed automata that can have diagonal guards and the special resets described above. Show that the following language can be recognized by a timed automaton in $\text{TA}^d_{x:=x+c}$:

$$\{\ (w,\tau) \mid w \in (a+b)^*, \tau \text{ is some time sequence, and } w \text{ has the same number of } a\text{'s and } b\text{'s}\}$$

Can the above language be recognized by a normal timed automaton which has resets only to 0?

21. Prove that the language emptiness problem for the class of timed automata $\mathrm{TA}^{d}_{x:=x+c}$ described in the above question is undecidable.

You may use the following undecidable problem.

A Minsky machine (a version of 2-counter machine) consists of a finite set of labeled instructions $I_1, \ldots, I_n$ and two counters c_1, c_2. There is a specified initial instruction I_0 and a special instruction labeled HALT. The instructions are of two types:

- an *incrementation* instruction of counter $c \in \{c_1, c_2\}$

$$p: \; c := c + 1; \; \text{goto } q \qquad \text{(where } p, q \text{ are instruction labels)}$$

- or a *decrementation (or zero-testing)* instruction of counter $c \in \{c_1, c_2\}$

$$p: \; \text{if } c > 0 \begin{cases} \text{then } c := c - 1; \text{ goto } q \\ \text{else goto } r \end{cases} \qquad \text{(where } p, q, r \text{ are instruction labels)}$$

The machine starts at instruction I_0 with counters $c_1 = c_2 = 0$, executes the instructions successively, and stops only when it reaches the instruction HALT. The halting problem for Minsky machine is to decide if there is an execution of the machine that reaches the instruction HALT.

It is known that the halting problem for Minsky machines is undecidable.