Exercise 1: Timed automata and programs 1
Consider the timed automaton $T_1$ from Figure 1 with clock variable $x$, i.e., $C = \{x\}$.

(a) Translate $T_1$ to an equivalent program $P_1$, i.e., with the same executions/paths.

(b) Compute the reachable states $\varphi_{\text{reach}}$ of $P_1$ by iteration of $\text{post}$. 
Exercise 2: Timed automata and programs 2
Consider the timed automaton $T_2$ which is obtained from $T_1$ (see Figure [1]) by adding another clock variable $y$, i.e., $C = \{x, y\}$. Note that $y$ is never read in any guard or invariant. Still, the state space changes (recall that a state is a pair $(\ell, \nu)$ with $\nu : C \rightarrow \mathbb{R}$).

You may wonder why adding an unused clock should affect the reachability of a state. In fact, it does not (in some sense). However, the algorithms behave differently.

(a) Translate $T_2$ to an equivalent program $P_2$.

(b) What are the reachable states $\varphi_{\text{reach}}$ of $P_2$?
   Solve this exercise intuitively, i.e., do not apply a formal algorithm. Alternatively, you may reason about the reachable states $\text{Reach}$ of $T_2$.

(c) What happens when you try to compute the reachable states $\varphi_{\text{reach}}$ of $P_2$ by iteration of $\text{post}$?

(d) Find a suitable set of predicates $\text{Preds}$ such that the predicate abstraction (iteration of $\text{post}^\#$) can prove safety (specified by the TCTL formula $\mathbf{AG} \neg \text{err}$) of $P_2$.
   Provide the abstract reachability graph that you obtain.

   Hint: Consider some of the predicates which are used to define the regions for the region transition system ($\text{RTS}$) construction.