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) What are the reachable states $\varphi_{\text{reach}}$ of $T_2$?

Hint: Solve this exercise intuitively, i.e., do not apply a formal algorithm.

(b) Translate $T_2$ to a program $P_2$.

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

---

Figure 1: A timed automaton.

\[ x = 1; x := 0 \]

\[
\begin{array}{c}
\text{l}_0 \\
x \leq 1 \\
{} \\
\text{l}_1 \\
x > 1 \\
\text{err}
\end{array}
\]
(d) Find a suitable set of predicates $Preds$ such that the predicate abstraction (iteration of $post^*$) can prove safety (specified by the TCTL formula $A\neg err$) 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 ($RTS$) construction.