Exercise 1: (Coffee machine**)**

Consider a coffee machine specified as follows:

– Initially the machine waits for a ***coin***,
– As soon as the coin is provided, the user can choose if she wants a ***coffee***, a ***tee*** or getting her coin ***back***,
– **After 20 seconds**, if she has not chosen yet, she can only get her coin ***back***,
– If she chooses a ***coffee*** or a ***tea*** in **less than 20 seconds** she ***gets*** it.

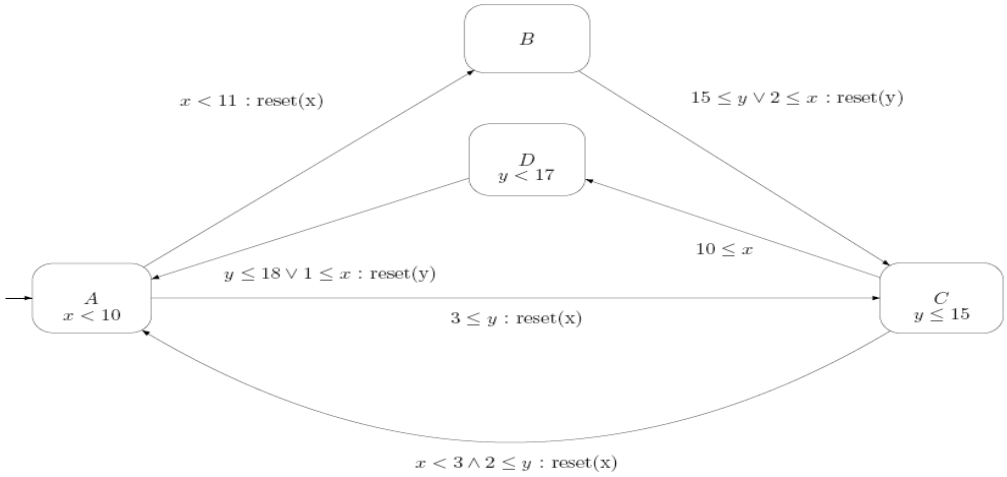
1. Describe this machine as a timed automaton.

We add the following requirements for our coffee machine:

– The coin is automatically ***returned*** to the user if she has not chosen her beverage ***in less than 20 seconds***,
– It takes ***35 seconds*** to prepare a tea and ***40 seconds*** to prepare a coffee.

2. Describe this new version of coffee machine as a timed automaton with ***urgent*** locations.

Exercise 2:



1) Use your knowledge on the Uppaal to edit and simulate this automaton (try to use the function concept in Uppaal)
2) Verify: reachability, safety, and deadlock-free.

Exercise 3: (autonomous elevator)

Consider an **autonomous elevator** which operates **between two floors**. The requested behaviour of the elevator is as follows:

- The elevator can stop either at the **ground floor** or the **first floor**.
- When the elevator arrives at a **certain floor**, its door automatically ***opens***. It takes **at least 2 seconds** from its arrival before the door opens but the door **must definitely open within 5 seconds**.
- Whenever the elevator's door is open, ***passengers can enter***. They enter one by one and we (optimistically) assume that the elevator has a sufficient capacity to accommodate any number of passengers waiting outside.
- The door can ***close*** only **4 seconds after the last passenger entered**.
- After the door closes, the elevator ***waits*** **at least 2 seconds** and then travels ***up*** or ***down*** to the other floor.

Questions:

1) Using one timed automaton, specify the above behaviour: actions={up, down, open, close, enter}
2) Provide two different timed traces of the system starting at the ground floor with the door open.
3) Using two automatons: one for the **elevator** and one for the passenger, give a second modelling.
4) Propose a solution with three automatons: **elevator**, **door**, and **passenger**.

Exercise 4: (the CSMA/CD protocol)

The protocol is composed of two entities: the **sender** and the **bus**. Two key parameters must be known: the "emission delay" **em_delay** of a tram and the "maximal propagation delay between two stations" **pro_delay**. The behaviour of the system is as follows:

1. The bus can be either: **free**, **active** or in **collision**. From the free state, the bus will be active after sending a ***begin*** to a ready sender. From the active state, the bus can return to the free state by receiving ***end*** of transmission from the current sender. The senders sense (***busy?***) that the bus is active after a period of time (**pro_delay**). The bus can transit to the collision state if it allows (by sending a ***begin***) to a sender before the end of the current transmission (this is possible if the second sender request the bus before it senses that the bus is active, so before the **pro_delay**). When the collision appears, the bus informs (***CD!***) the senders.

2. A sender can be either: **waiting** for transmission, **retrying** a transmission after a collision was detected, **transmitting**, or **finishing** its transmission. In case of collision detection (**CD**), the sender will stays forever in the waiting state. The sender leaves the waiting state in two cases: (1) when it receives a ***begin*** from the bus, hence it transit to the transmitting state, or when it senses that the bus is active (***busy***), hence, it transits to the retrying state. In the transmitting state, the sender will stays while the bus is active (***busy***). The sender leaves the transmitting state in two cases: when it has finished transmission (**em_delay** is elapsed), hence it sends (***end*** of transmission the bus) and transits to the finishing state. The second case is when a collision is detected (***CD***) before ending its transmission, hence it transits to the retrying state, where it will retry a new transmission. In the retry state, the sender can stay a period of time less than (2* **pro_delay**) while there is a collision detection or while the bus is active. The sender can transit to the transmitting state if it receives a ***begin*** from the bus which must be before the elapsing of (2*pro_delay) too.

Questions:

1) Using the above description, propose two automatons for the bus and the sender. Justify your choices. Make simulation.
2) Is this system deadlock free? Is this system equitable?