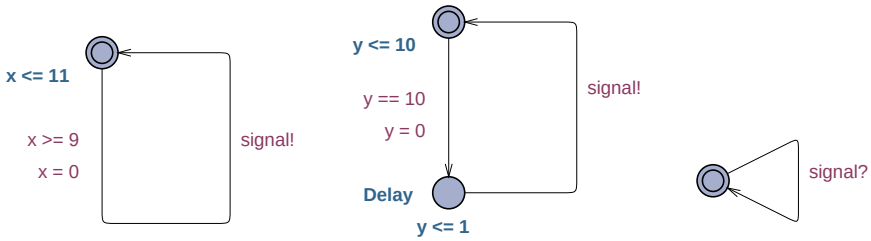
Solution to Exercise 1 Skew and Jitter

Models exhibiting skew and jitter respectively can be expressed by:



(a) Skew (b) Jitter (c) Observer

In this solution the observer is always ready, and hence the signal event is under the control of the processes. These may control the time of the signal using a combination of invariants and guards as shown.

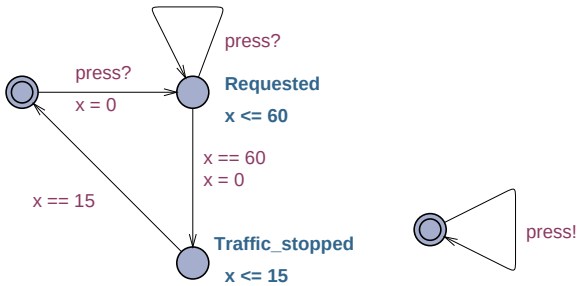
For the skewed process, the distance betweeen the first and last of five events is found by hand calculations to be within the interval [36..44]. For the jittering process, the interval is [39..41].

[Using verification, a clock can be added in the observer and these intervals can be verified by global invariants.]

The full UPPAAL model is available as `SkewJitter.xml`.

Solution to Exercise 2 Pelican Light

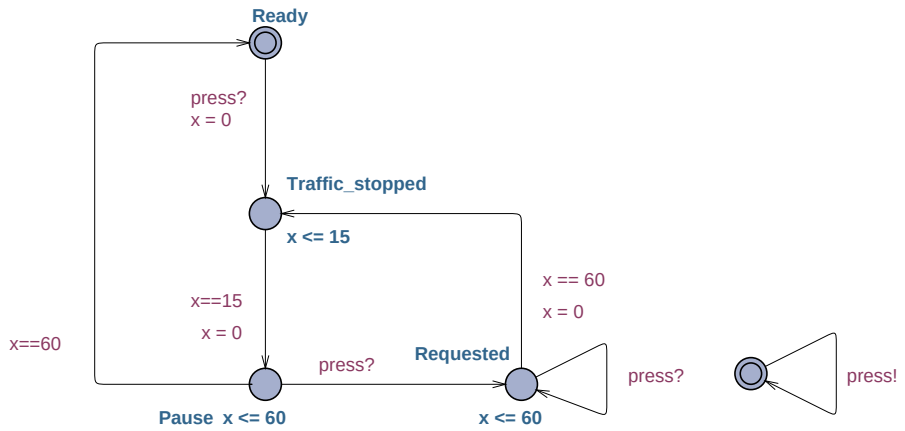
If we are only interested in the light control and not the actual crossing by the pedestrians, and if we would like to allow the pedistrian to request a crossing at any time, a simple solution can be given by:



(d) Pelican (e) Pedestrian

In this solution, the maximum blocking of the traffic is ensured by always starting with a delay of 60 seconds before stopping the traffic. However, for the occational pedestrians, this is quite annoying.

A more pedestrian-oriented solution only delays the crossing if the traffic was recently stopped:



(f) Pelican (g) Pedestrian

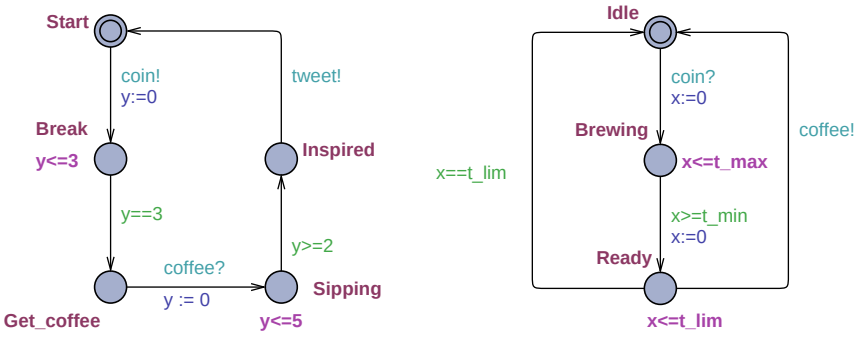
This solutions is available as the UPPAAL model `Pelican1.xml`.

In both solutions, the press channel should not be declared urgent, as we should not force the pedestrians to press the button.

Solution to Exercise 3 Coffee Machine

Task 1

A solution for the coffee machine is:

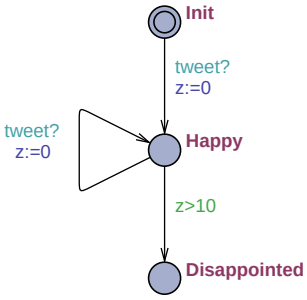


(h) Person (i) Machine

In this model, all channels should be urgent reflecting the eagerness of the person.

Task 2

The follower simply resets his/her clock whenever a tweet is received. If the 10 minutes are exceeded, the follower may become disappointed:



(j) Follower

Note that the follower may choose to ignore the lateness of the tweet. We could make a more strict observer by adding the invariant $z \leq 10$ to the Happy state and changing the guard $z > 10$ to $z = 10$. But then the tweet has to arrived strictly before 10 minutes have elapsed.

The UPPAAL model can be found as `Coffee.xml`.

Task 3

Obviously we must require $t_{min} \leq t_{max}$. Otherwise a time deadlock will occur.

The system will also enter a deadlock if the Machine times out before the coffee is taken. This may happen if the coffee is brewed quickly and the time out occurs before the break has ended. So we should avoid

$$t_{min} + t_{lim} \leq 3$$

and hence require

$$t_{min} + t_{lim} > 3$$

Finally, the observer may become disappointed if a slow brewing followed by slow sipping delays the tweet by more than 10 minutes. To avoid this:

$$t_{max} + 5 \leq 10 \quad \Leftrightarrow \quad t_{max} \leq 5$$