

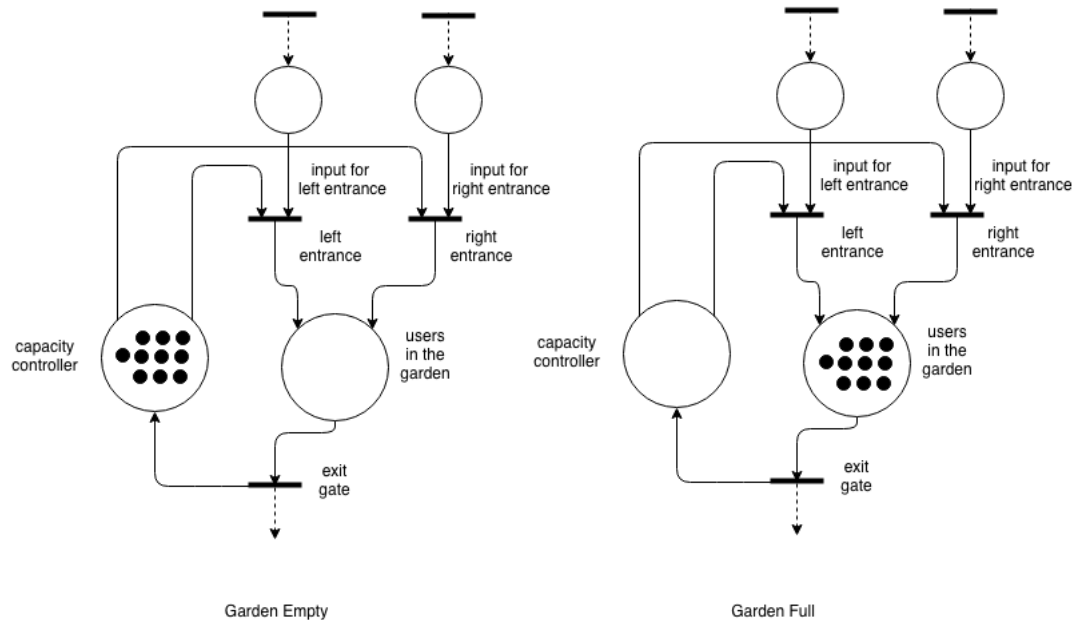
Petri Nets 1

Dario A Lencina-Talarico

Homework #13

Due: 26 November 2018

1. Consider a small garden, similar to the one described by Kramer and Magee, that has two entrances, an East entrance and a West entrance. Users may enter or leave either entrance through a small gate that permits only one user to enter or leave at a time. The park can hold up to 10 people. Model this system using a Petri Net.



The left entrance and right entrance transitions are not synchronized, users can enter through either entrance. When the users enter, a token is moved from “capacity controller” to “users in the garden”.

Users waiting to join the gardening wait in the places below the transitions with no inputs (top).

When the garden becomes full, meaning, 10 users are presents and the capacity controller has no token left, the left and right transitions can not be triggered until a user gate transition is activated moving at least one of the tokens back to the capacity controller.

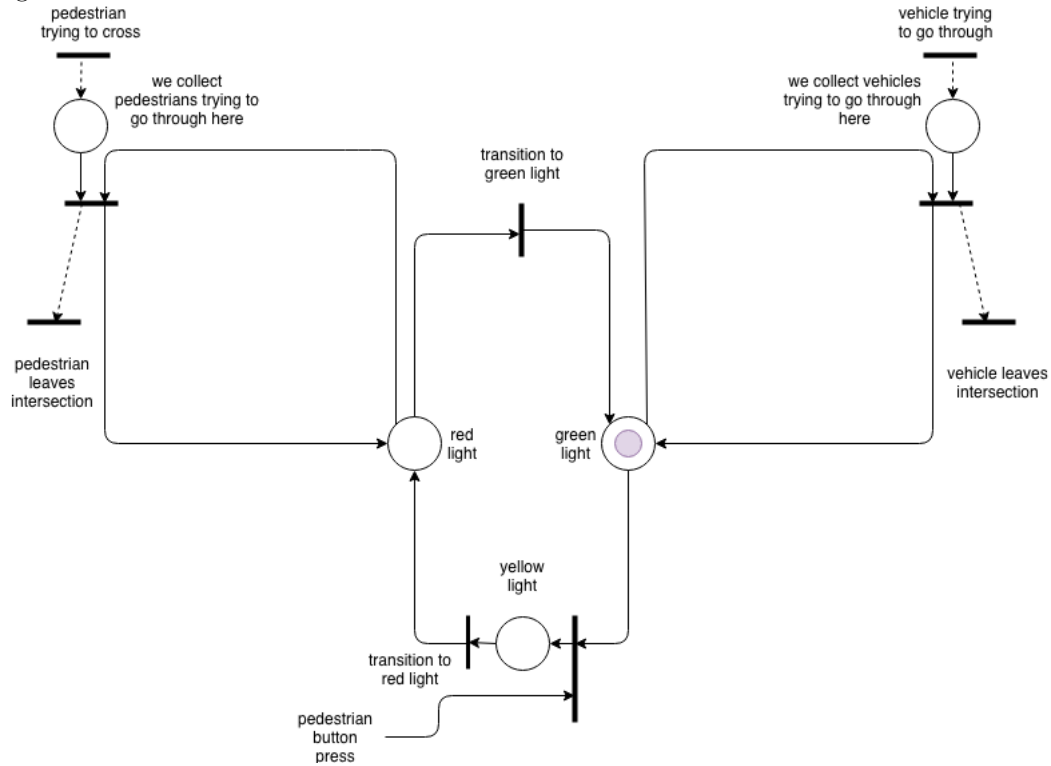
This is an example of using the “Limited capacity” design pattern from Petri Nets part 2 page 12.

2. Consider a traffic light that controls traffic on a two-lane road. The light is normally green, allowing cars to pass through. But when a user presses a button, the traffic light should turn yellow, and then red, allowing some number of users to cross the street. It then turns green again, and users are prohibited from crossing.

(a) Describe this system using a Petri Net.

I modified my implementation so that pedestrians and vehicles share the green and red

lights and synchronize when they should start and stop moving accordingly.
 I also added the “user button press” input that models when the pedestrians request for the traffic light to turn red, this implies that if no user presses the button, then the traffic light will never transition to red.



- (b) Does your net guarantee that the light cannot turn red without first turning yellow? Justify briefly.

Yes, I introduced a transition that forces the traffic light to switch to yellow and another one that makes the light transition to red.

- (c) Does your net guarantee that an arriving car will eventually be allowed to pass through the light?

Yes, on the top right of the diagram it can be seen that there's a transition with two inputs: the infinite supply of vehicles and the green light place that trigger allowing vehicles through.

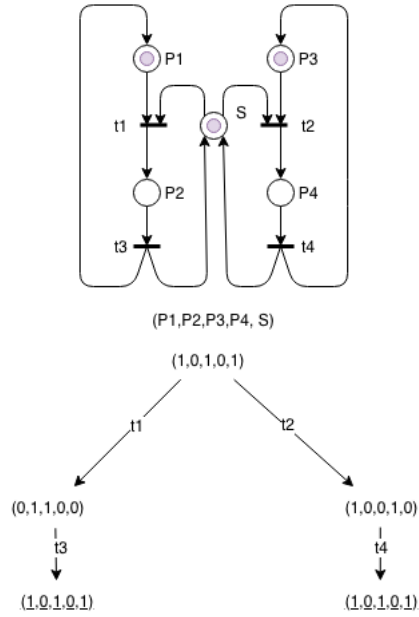
- (d) List three aspects of the real system that are abstracted away by your model.

In my diagram there's no mention of how many vehicles can drive through concurrently, I did not discuss the capacity of the pedestrian crossing or the duration of the traffic light.

3. For the P/V Petri Net shown in Figure 15 of Peterson's article (Pet77.pdf, page 234):

Ran out of time but will do my best to submit a decent redo with this section.

- (a) Construct the reachability tree, using the tree condensation algorithm shown in class. Assume that the ordering of places in the state vector of the Petri Net shown in Figure 15 is:



- (b) Use this to argue that the design is correct, i.e., that it is not possible for the system to be simultaneously in the critical sections of both processes.
 As the reachability tree shows, we were not able to find a state where there is a token in both critical sections (P2, P4).

We also learned that after the system executes the critical section, it returns the token to S so that the other branch has a chance to acquire the token.
 Imo this is a correct implementation of a mutex petri net.