1. Let us examine the cabbage-goat-wolf system, presented in 2nd tutorial. Fold the net in such a way that there is only one place for each passenger. That is, the places cabbage1 and cabbage2 should be folded into one place etc.. Fold also the places modeling the location of the boat.

   The net can be folded further. The places modeling the passengers can be folded, that is places cabbage, goat and wolf. We can also fold the transitions moving the passengers.

   Finally, the places modeling the boat and passengers can be folded.

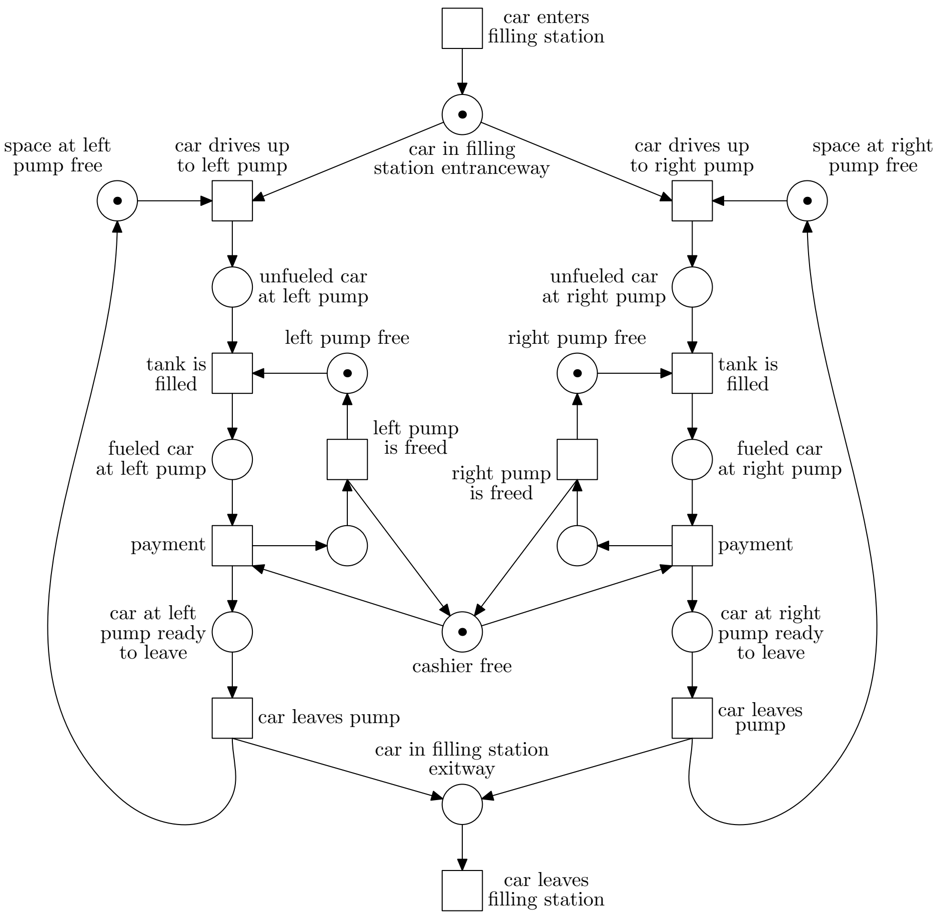
2. There are n writer processes and m reader processes in a system. Reader processes can read concurrently, but if a writer process wants to write, no other process can neither read nor write. Model the system with an algebraic net.

3. Model the following Peterson's algorithm for mutual exclusion using algebraic nets. The processes P_0 and P_1 are concurrent processes which are both trying to enter the critical section CS. The variables `in0`, `in1` and `turn` are shared between teh processes. Assume that all assignments and tests are atomic. Provide a fact transition or a Maria `reject` statement to test the mutual exclusion.

```
P0 :
WHILE true
  in0 := 1;
  turn := 1;
  WHILE in1 = 1 AND turn = 1
    skip;
  ENDWHILE
  CS();
  in0 := 0;
  NCS0();
ENDWHILE

P1 :
WHILE true
  in1 := 1;
  turn := 0;
  WHILE in0 = 1 AND turn = 0
    skip;
  ENDWHILE
  CS();
  in1 := 0;
  NCS1();
ENDWHILE
```

4. Examine the following place/transition system modeling a self-service fueling station:



Assume that we are only interested in the quantitative aspects of the fueling station:

(a) Simplify the net so that it is no longer possible to observe if the left or right pump is being used.

(b) Add another cashier to the net.

(c) Assume that the system models a fueling station of a garage for n cars. Modify the system so that the reachability graph will be finite.