

# ESIR SPP – TD6

## Exercise 1: Petri Nets

Consider the following code from the session on semaphores.

```
queue q
sem q_sem = new sem(1)
sem empty = new sem(N)
sem full = new sem(0)

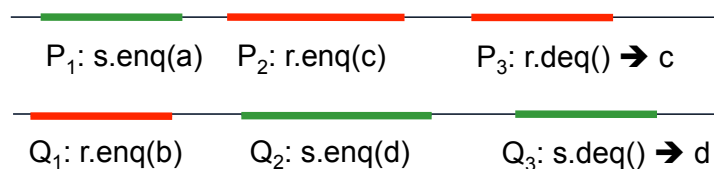
method produce(x) is
  empty.down()
  q_sem.down()
  q.add(x)
  q_sem.up()
  full.up()
end

method int consume() is
  full.down()
  q_sem.down()
  result = q.get()
  q_sem.up()
  empty.up()
  return result
end
```

- Model this code as a Petri net.
- Extend your model with 2 threads: one that continuously invokes 'produce', and another one that continuously invokes 'consume'.
- Compute the reachability graph of your whole Petri net for N=2.
- Is your Petri net safe? Is it lively?

## Exercise 2: Atomicity, Sequential Consistency, and Locality

Consider the following execution H (or history) of two threads P and Q. These two threads use two shared FIFO queues s and r, which supports two operations, enq() for enqueue and deq() for dequeue.



- What is the partial order induced by this execution on both thread's operations?
- Do s and r behave atomically in this execution? (i.e. are H|s and H|r atomic?)
- Do s and r behave in a sequentially consistent manner in this execution?
- If we now consider the composition of s and r as one single object (i.e. offering 2 methods enq(x,a) and deq(x) where x is the name of queue being manipulated, "s" or "r"), is H sequentially consistent?
- From the above, what can you conclude on the locality of sequential consistency?