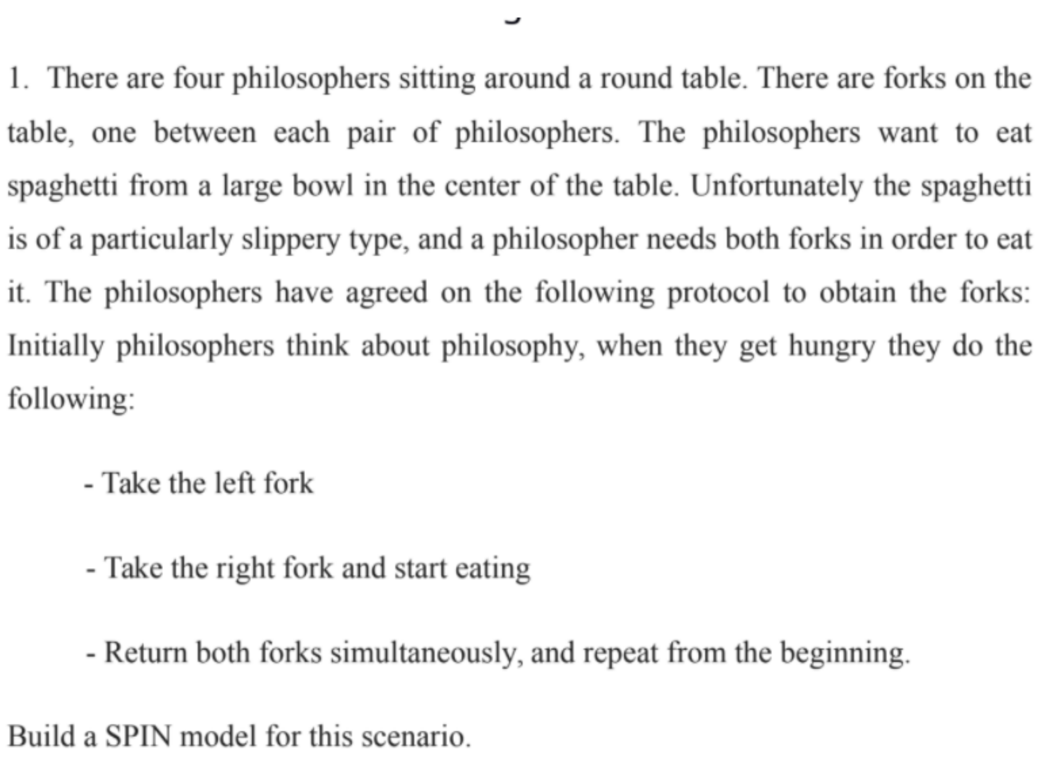
U18CO018

Shekhaliya Shubham

SE

Assignment 5



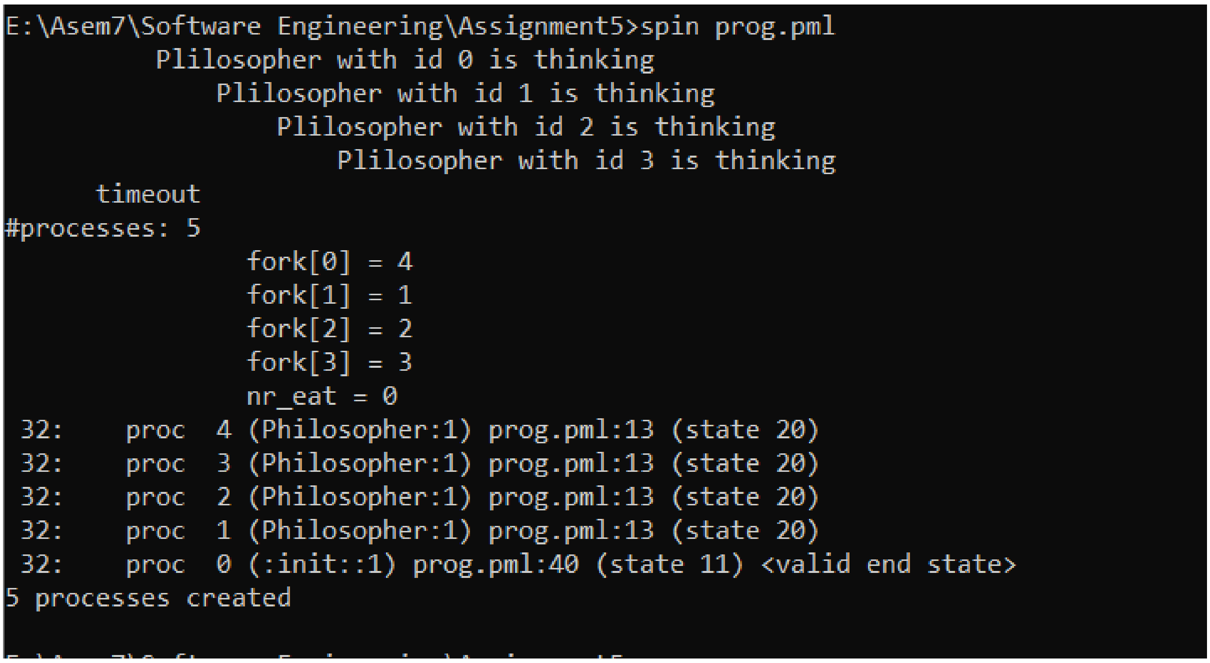
1. There are four philosophers sitting around a round table. There are forks on the table, one between each pair of philosophers. The philosophers want to eat spaghetti from a large bowl in the center of the table. Unfortunately the spaghetti is of a particularly slippery type, and a philosopher needs both forks in order to eat it. The philosophers have agreed on the following protocol to obtain the forks: Initially philosophers think about philosophy, when they get hungry they do the following:

- Take the left fork
- Take the right fork and start eating
- Return both forks simultaneously, and repeat from the beginning.

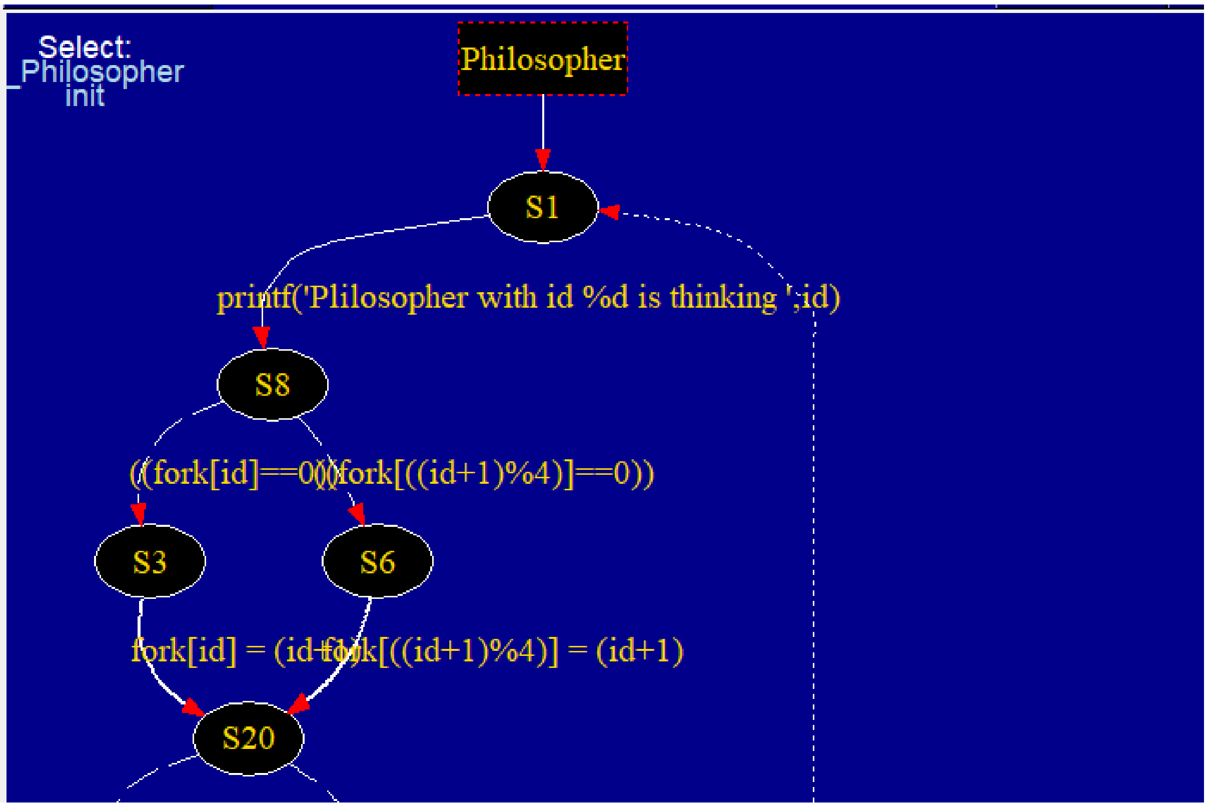
Build a SPIN model for this scenario.

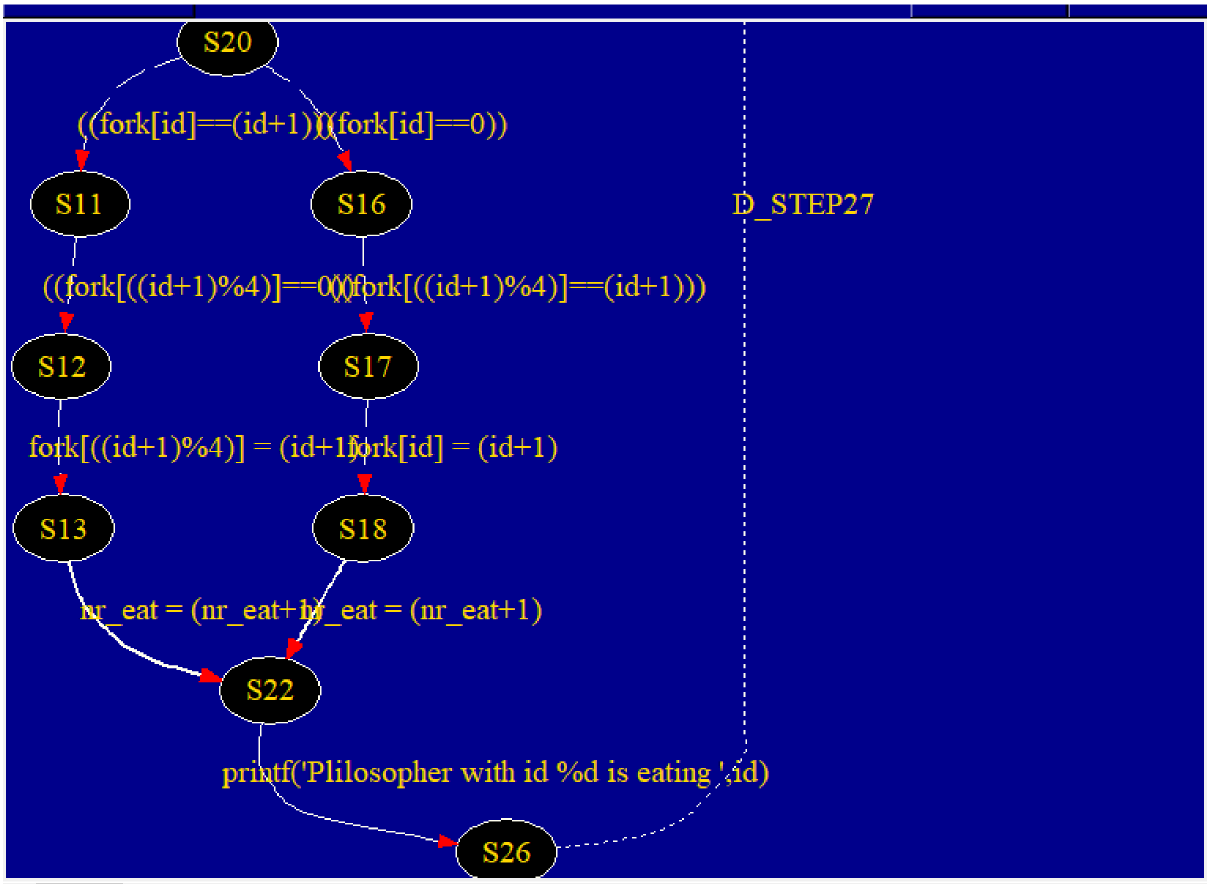
Code:

```
#define N 4
byte fork[N];
byte nr_eat;
proctype Philosopher(byte id)
{
   Think:
        printf("Plilosopher with id %d is thinking\n",id);
        if
        :: atomic { fork[id] == 0 -> fork[id] = id + 1; };
        :: atomic { fork[(id + 1)%N] == 0 -> fork[(id + 1)%N] = id + 1; };
        fi;
    One:
        if
        :: atomic
        {
            fork[id] == id + 1 -> fork[(id + 1)%N] == 0 -> fork[(id + 1)%N] = id + 1;
            nr_eat++;
        }
        :: atomic
        {
            fork[id] == 0 -> fork[(id + 1)%N] == id + 1 -> fork[id] = id + 1;
            nr_eat++;
        }
        fi;
    Eat:
        printf("Plilosopher with id %d is eating\n",id);
        d_step { nr_eat--; fork[(id + 1)%N] = 0; fork[id] = 0;}
    goto Think;
}

init {
    atomic
    {
        byte i = 0;
        do
        :: i < N -> run Philosopher(i); i++;

        :: else -> break;

        od;

    }

}
```



```
E:\Asem7\Software Engineering\Assignment5>spin prog.pml
          Plilosopher with id 0 is thinking
              Plilosopher with id 1 is thinking
                  Plilosopher with id 2 is thinking
                      Plilosopher with id 3 is thinking
      timeout
#processes: 5
                fork[0] = 4
                fork[1] = 1
                fork[2] = 2
                fork[3] = 3
                nr_eat = 0
 32:    proc  4 (Philosopher:1) prog.pml:13 (state 20)
 32:    proc  3 (Philosopher:1) prog.pml:13 (state 20)
 32:    proc  2 (Philosopher:1) prog.pml:13 (state 20)
 32:    proc  1 (Philosopher:1) prog.pml:13 (state 20)
 32:    proc  0 (:init::1) prog.pml:40 (state 11) <valid end state>
5 processes created

E:\Asem7\Software Engineering\Assignment5
```


Select:
_Philosopher
init
Philosopher
S1
printf('Plilosopher with id %d is thinking ',id)
S8
((fork[id]==0))
((fork[((id+1)%4)]==0))
S3
S6
fork[id] = (id+1)
fork[((id+1)%4)] = (id+1)
S20


S20
((fork[id]==(id+1)))
((fork[id]==0))
S11
S16
D_STEP27
((fork[((id+1)%4)]==0))
((fork[((id+1)%4)]==(id+1)))
S12
S17
fork[((id+1)%4)] = (id+1)
fork[id] = (id+1)
S13
S18
nr_eat = (nr_eat+1)
nr_eat = (nr_eat+1)
S22
printf('Plilosopher with id %d is eating ',id)
S26