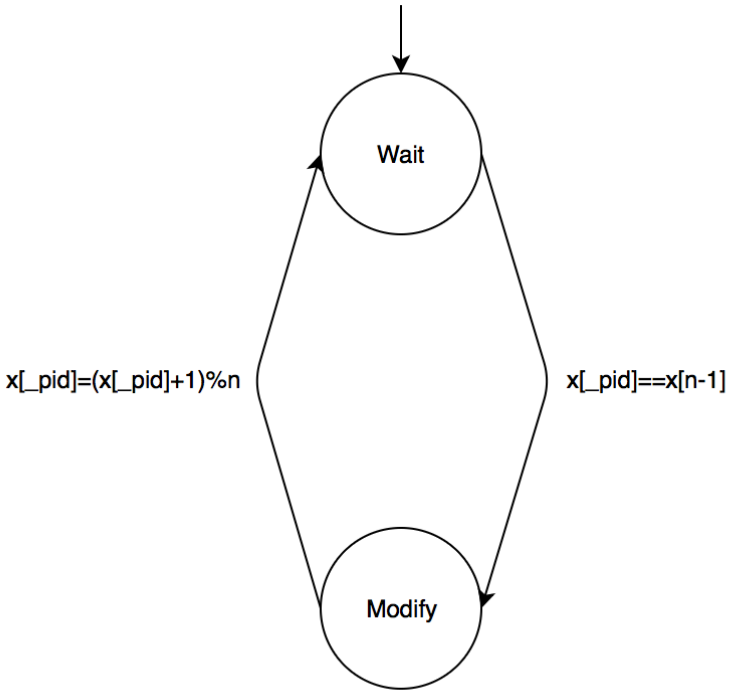
ASSIGNMENT 3

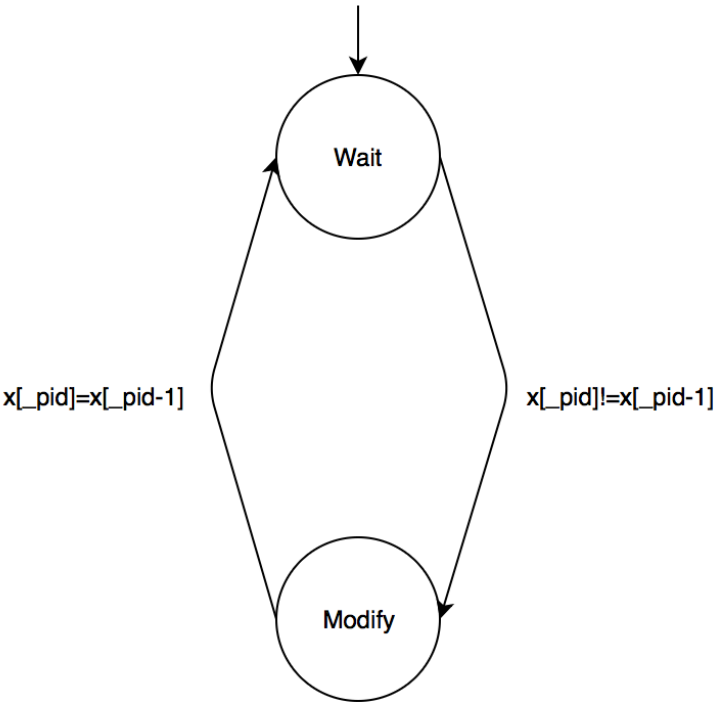
AISHWARIYA TALATHI

CSULB ID: 016131932

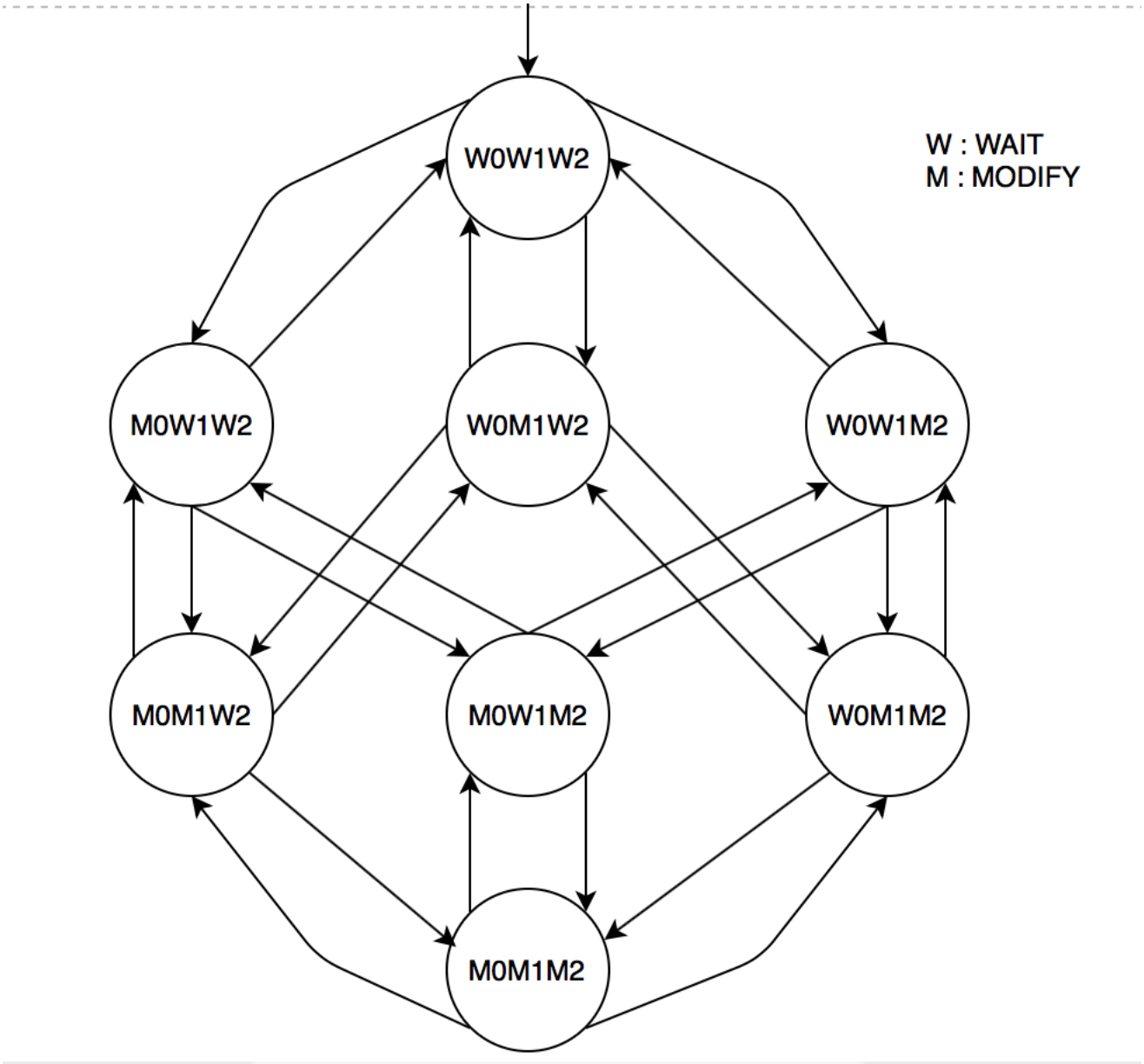
PROGRAM GRAPHS: 1)specProcess



2)process



PARALLEL COMPOSITION FOR ALL 3 PROCESSES:



MODEL IMPLEMENTATION IN SPIN FOR Dijkstra.cpp:

```
#define n 3
int x[n];

active proctype specProcess()
{
x[_pid]= 0;
label1: if :: x[_pid]==x[n-1] -> x[_pid]=(x[_pid]+1)%n;
printf("Process ID := %d\n", _pid);
fi;
goto label1;
}

active[2] proctype process()
{
x[_pid] = 0;
label2: if :: x[_pid]!=x[_pid-1] -> x[_pid]=x[_pid-1];
printf("Process ID := %d\n", _pid);
fi;
goto label2;
}
```

TERMINAL OUTPUT FOR 100 STEPS:

```
Aishwariyas-MacBook-Pro:Exercises aishwariyatalathi$ spin -u30 -p -l ass3.pml
  0:  proc  - (:root:) creates proc  0 (specProcess)
  0:  proc  - (:root:) creates proc  1 (process)
  0:  proc  - (:root:) creates proc  2 (process)
  1:  proc  1 (process:1) ass3.pml:15 (state 1) [x[_pid] = 0]
  2:  proc  2 (process:1) ass3.pml:15 (state 1) [x[_pid] = 0]
  3:  proc  0 (specProcess:1) ass3.pml:6 (state 1)    [x[_pid] = 0]
  4:  proc  0 (specProcess:1) ass3.pml:7 (state 2)    [((x[_pid]==x[(3-1)]))]
  5:  proc  0 (specProcess:1) ass3.pml:7 (state 3)    [x[_pid] = ((x[_pid]+1)%3)]
      Process ID := 0
  6:  proc  0 (specProcess:1) ass3.pml:8 (state 4)    [printf('Process ID :=
%d\\n',_pid)]
  7:  proc  1 (process:1) ass3.pml:16 (state 2) [((x[_pid]!=x[(_pid-1)]))]
  8:  proc  1 (process:1) ass3.pml:16 (state 3) [x[_pid] = x[(_pid-1)]]
  9:  proc  2 (process:1) ass3.pml:16 (state 2) [((x[_pid]!=x[(_pid-1)]))]
          Process ID := 1
 10:  proc  1 (process:1) ass3.pml:17 (state 4) [printf('Process ID := %d\\n',_pid)]
 11:  proc  0 (specProcess:1) ass3.pml:10 (state 6)   [.(goto)]
 12:  proc  2 (process:1) ass3.pml:16 (state 3) [x[_pid] = x[(_pid-1)]]
 13:  proc  1 (process:1) ass3.pml:19 (state 6) [.(goto)]
 14:  proc  1 (process:1) ass3.pml:19 (state 7) [goto label2]
 15:  proc  0 (specProcess:1) ass3.pml:10 (state 7)   [goto label1]
 16:  proc  0 (specProcess:1) ass3.pml:7 (state 2)    [((x[_pid]==x[(3-1)]))]
 17:  proc  0 (specProcess:1) ass3.pml:7 (state 3)    [x[_pid] = ((x[_pid]+1)%3)]
 18:  proc  1 (process:1) ass3.pml:16 (state 2) [((x[_pid]!=x[(_pid-1)]))]
 19:  proc  1 (process:1) ass3.pml:16 (state 3) [x[_pid] = x[(_pid-1)]]
          Process ID := 1
 20:  proc  1 (process:1) ass3.pml:17 (state 4) [printf('Process ID := %d\\n',_pid)]
               Process ID := 2
 21:  proc  2 (process:1) ass3.pml:17 (state 4) [printf('Process ID := %d\\n',_pid)]
 22:  proc  1 (process:1) ass3.pml:19 (state 6) [.(goto)]
      Process ID := 0
 23:  proc  0 (specProcess:1) ass3.pml:8 (state 4)    [printf('Process ID :=
%d\\n',_pid)]
 24:  proc  1 (process:1) ass3.pml:19 (state 7) [goto label2]
 25:  proc  2 (process:1) ass3.pml:19 (state 6) [.(goto)]
 26:  proc  2 (process:1) ass3.pml:19 (state 7) [goto label2]
 27:  proc  0 (specProcess:1) ass3.pml:10 (state 6)   [.(goto)]
```

```
 28:  proc  0 (specProcess:1) ass3.pml:10 (state 7)   [goto label1]
 29:  proc  2 (process:1) ass3.pml:16 (state 2) [((x[_pid]!=x[(_pid-1)]))]
 30:  proc  2 (process:1) ass3.pml:16 (state 3) [x[_pid] = x[(_pid-1)]]
-------------
depth-limit (-u30 steps) reached
#processes: 3
            x[0] = 2
            x[1] = 2
            x[2] = 2
 30:  proc  2 (process:1) ass3.pml:17 (state 4)
 30:  proc  1 (process:1) ass3.pml:16 (state 5)
 30:  proc  0 (specProcess:1) ass3.pml:7 (state 5)
3 processes created
```

HENCE, AS SEEN FROM ABOVE, AFTER 30 STEPS, THE VALUE OF X[0] = 2, X[1] = 2 AND X[2] = 2 FOR n=3;