Roberto E. Vargas
Diego Garcia-Olano
Rajagopal Hariharan

***************FSP Models**

Deadline: 10/22

1) Exercise 2.3 from problems list.

```
BISTABLE = BISTABLE[0],
BISTABLE[0] = (trigger -> one -> BISTABLE[1]),
BISTABLE[1] = (trigger -> zero -> BISTABLE[0]).
```

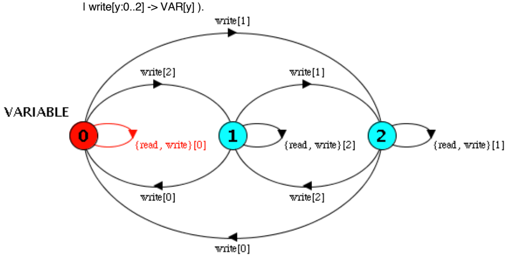
2) Exercise 2.6 from problem list.

```
const Top = 108
const Bottom = 88
range Spectrum = Bottom..Top


RADIO = OFF,
OFF =                    ( on -> STATION[Top] ),

STATION[x:Spectrum] =    ( off -> OFF
                 |  when(x > Bottom ) scan -> STATION[x-1]
                 |  when(x == Bottom ) {scan,end} -> STATION[Bottom]

                 |  lock -> STATION[x]
                 | reset -> STATION[Top] ).
```

for each of the exercises 2.2 to 2.6
draw the state machine diagram that corresponds to your FSP specification and check that it can perform the required actions.

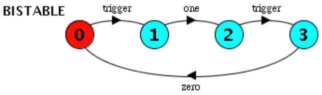
2.2

```
VARIABLE = VAR[0],
VAR[x:0..2] = ( read[x] -> VAR[x]
              | write[y:0..2] -> VAR[y] ).
```



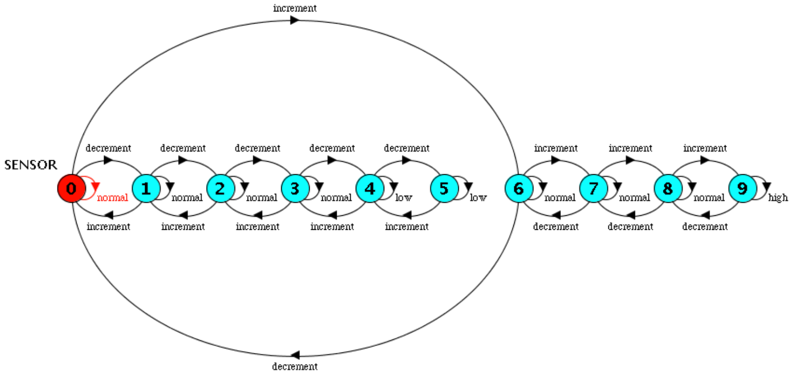
2.3

```
BISTABLE = BISTABLE[0],
BISTABLE[0] = (trigger -> one -> BISTABLE[1]),
BISTABLE[1] = (trigger -> zero -> BISTABLE[0]).
```



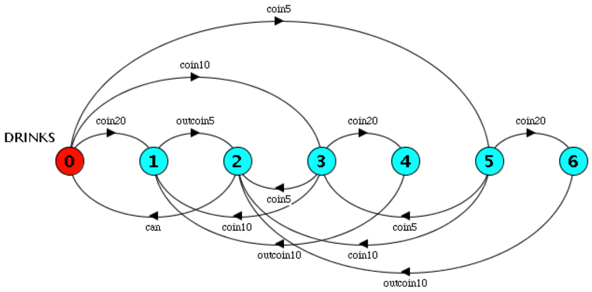
2.4

```
SENSOR = Level[5],
Level[i:0..9] =  ( when(i > 0) decrement -> Level[i-1]
                 | when(i < 9) increment -> Level[i+1]
                 | when( i < 2 ) low -> Level[i]
                 | when( i > 8 ) high -> Level[i]
                 | when( i >= 2 && i <= 8) normal -> Level[i] ).
```

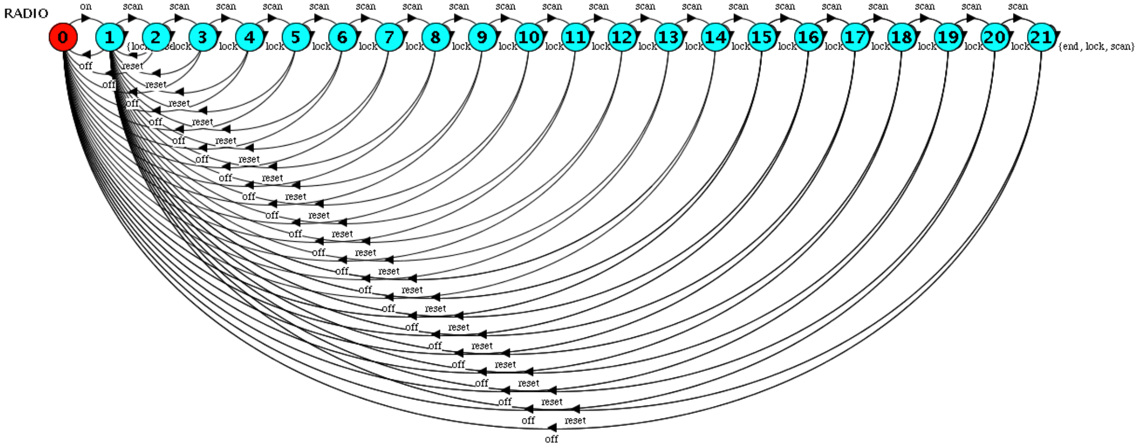


2.5

```
DRINKS = CREDIT[0],
CREDIT[0]  = ( coin5 -> CREDIT[5] | coin10 -> CREDIT[10] | coin20 -> CHANGE[5] ),
CREDIT[5]  = ( coin5 -> CREDIT[10]| coin10 -> CHANGE[0] | coin20 -> CHANGE[10] ),
CREDIT[10] = ( coin5 -> CHANGE[0] | coin10 -> CHANGE[5] | coin20 -> CHANGE[15] ),
CHANGE[0] = ( can -> DRINKS ),
CHANGE[5] = ( outcoin5 -> CHANGE[0] ),
CHANGE[10] = ( outcoin10 -> CHANGE[0] ),
CHANGE[15] = ( outcoin10 -> CHANGE[5] ).
```

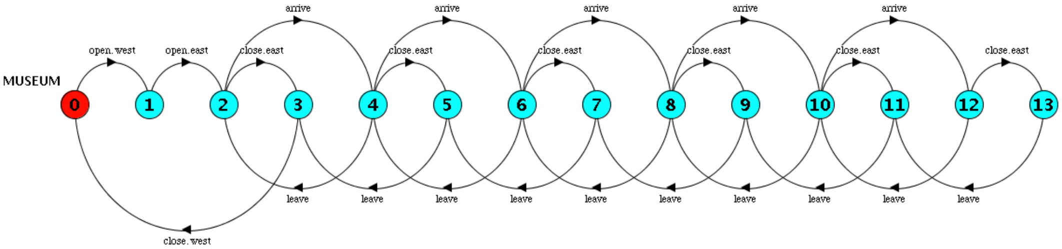


2.6



3) Complete the following solution for Exercise 3.6 from the problem list.

```
const N = 5

EAST = (open.east -> EASTOPEN),
EASTOPEN = (arrive -> EASTOPEN | close.east -> EAST).

WEST = (open.west  -> WESTOPEN),
WESTOPEN = ( leave -> WESTOPEN | close.west -> WEST).

DIRECTOR = (open.west -> open.east -> close.east -> close.west -> DIRECTOR).

CONTROL        = CLOSED,
CONTROL[i:0..N] = ( when(i < 1) close.west -> CLOSED
                  | when( i < N ) arrive -> CONTROL[i+1]
                  | when (i >0) leave -> CONTROL[i-1] ),
      CLOSED = (open.west -> open.east -> CONTROL[0]).


|| MUSEUM = (EAST || WEST || DIRECTOR || CONTROL ).
```


MUSEUM
arrive
arrive
arrive
arrive
arrive
open.west
open.east
close.east
close.east
close.east
close.east
close.east
close.east
0
1
2
3
4
5
6
7
8
9
10
11
12
13
leave
leave
leave
leave
leave
leave
leave
leave
leave
leave
close.west