Dispense15with Change, of the given figure is a modification of the previous dispenser, but this version also outputs change. For example, when in state 10, and an dime is input, an item is output and change (of 5 cents) is output also, and the next state is 0.

Tables of transitions also describe such sequential systems, as shown below to the right. For each of the 3 states there are 2 possible coins input thus yielding 6 possible combinations. Such representations help check that all possibilities are considered.

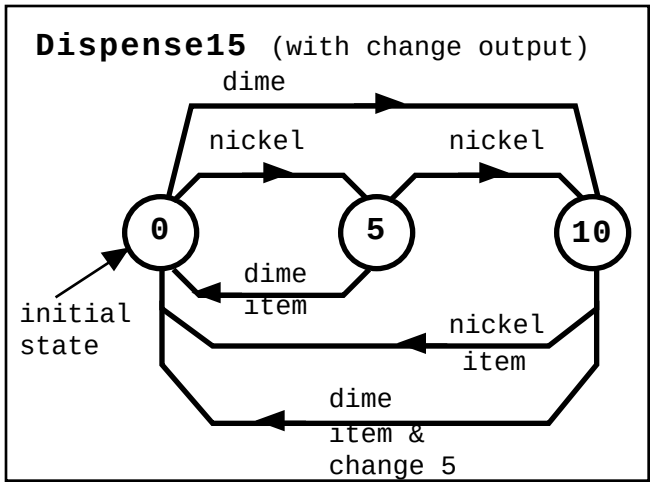

Table of transitions

state now	coin input	state next	actions output
0	nickel	5	none
0	dime	10	none
5	nickel	10	none
5	dime	0	item
10	nickel	0	item
10	dime	0	item & 5

Programs corresponding to such state diagrams are easy to do, as follows. They consist of a large choice, corresponding to the states, and then for each state is a sub choice depending on the input. Notice that the repeat on false causes the loop to continue forever.

```
 Boxes state, coin ofType int

Set state = 0

Repeat
ExitOn (false) -- loop forever!

   Output "Enter coin " -- prompt
   Input coin           -- enter
   Outputln coin        -- echo

   If (state == 0) then

      If (coin == 5) then
         Set state = 5
      Else -- coin is 10
         Set state = 10
      EndIf

ElseIf (state == 5) then

   If (coin == 5) then
      Set state = 10
   Else -- coin is 10
      Outputln "Give item "
      Set state = 0
   EndIf -- state is 5

Else -- (state is 10)

   If (coin == 5) then
      Outputln "Give item "
      Set state = 0
   Else -- coin is 10
      Outputln "Give item "
      Outputln "Give change "
      Set state = 0
   EndIf -- state is 10

EndIf

EndRepeat
```