

```

1 Asmad Murga Victor Manuel - LU: 760/19
2
3 Tablero. Notas:
4 Tengo un tablero de NxM dimensiones, a priori, este tablero inicia en 1x1. Entonces,
5 ese NxM serán parte de los resultados. No es ingreso del usuario
6
7
8 Metodos:
9 -> CrearTablero: -> tablero
10 No recibo nada, solo genero el tablero con una celda vacia y el puntero sobre ella
11
12 -> Puntero: tablero -> posicion
13 Me devuelve la posicion del puntero en el tablero. Al iniciar el tablero, esta es (0, 0)
14 Si me muevo a una posicion inexistente, es decir, a (1, 0) debo, crear una nueva celda y actualizar todas
15 las demas posiciones
16
17 -> ActualizarPosiciones:
18 Para toda posicion, si me muevo a derecha o arriba, sumo 1 a cada posicion. Caso contrario,
19 A derecha tomo la primera generada y la coloco en 0,0, abajo lo mismo, la primera a derecha abajo
20 y actualizo asi todas. Sumando 1 a cada posicion y a la nueva fila y nueva columna la seteo como 0 en fila
21 y 0 en columna
22 Esto es porque todas las posiciones son no negativas
23
24 IgObservacional Coloquial
25 Dame dos tableros. Estos tienen que tener las mismas dimensiones NxM
26 Por cada posicion, la cantidad de bolitas deben coincidir (sin importar color) y,
27 si la cantidad de bolitas que hay en esa posicion, coincide, entonces deben haber la misma
28 cantidad de bolitas por color en ambos tableros
29 Por otro lado, el puntero, tiene que estar apuntando a la misma posicion
30
31 Lamento bastante si hay cosas que no se entienden y la excesiva cantidad de funciones auxiliares
32 Hubieron algunas cosas que pueden no ser entendibles pues al final del recuperatorio me di cuenta de algunos
33 Errores que tenia entonces tuve que cambiar ciertas cosas, y casi que se hay cosas que pueden no entenderse
34 Por otro lado, el tema de las tuplas de posiciones, consideré en posicion del Gobstones o tablero, el primer nat
35 como Y y el segundo como X mientras que en el TAD de celda, fue al reves, esto se debe a que habia algo inconcluso
36 pues el tablero tomaba filas como X y columnas como Y. Y eso me generaba confusion. Puede no entenderse cuando tomo
37 alguna que otra conclusion. Desde ya, muchas gracias
38
39 TAD Color es {Azul, Rojo, Verde, Negro}
40 TAD Direccion es {Este, Oeste, Sur, Norte}
41 TAD Posicion es Tupla(Nat, Nat)
42
43 TAD Gobstones
44 Igualdad Observacional:(
45   (Vt1, t2: tablero)(s1=obs s2 ==>
46     dimension(s1) = dimension(s2) /\
47     celdas(s1) = celdas(s2) /\
48     cabezal(s1) = cabezal(s2) /\
49     celdasConBolitas(s1) = celdasConBolitas(s2) /\
50     (V c: celda)(c ∈ celdasConBolitas(s1) =>L
51       bolitas(t1, c) = bolitas(t2, c)))));
52 Géneros: tablero
53 Exporta: nombreTad, observadores, generadores,
54 Usa: Bool, Nat, Celda
55
56 Observadores Básicos:
57 dimension: tablero -> (nat, nat)
58 celdas: tablero -> conj(Celda)
59 cabezal: tablero -> Celda
60 celdasConBolitas: tablero t -> conj(Celda) { (Vp in celdasConBolitas(t))(t in posiciones(t))}
61 bolitasXCelda: tablero t x celda -> tupla(nat, nat, nat, nat) {celda ∈ celdas(tablero)}
62 Generadores:
63 IniciarTablero: -> tablero
64 Mover: tablero x direccion -> tablero {direccion ∈ {Este, Oeste, Sur, Norte}}
65 AgregarBolitas: tablero t x color c x nat cantidad x celda s -> tablero {c ∈ celdas(t) /\
66   c ∈ {Azul, Rojo, Verde, Negro} /\
67   cantidad ≥ 0
68   }
69
70 Otras Operaciones:
71 moverPosicion: tablero x Posicion -> Posicion
72 celdaXPosicion: tablero x Posicion -> celda
73 resultadoNuevaCelda: tablero x color x nat cantidad x celda x -> nat
74 agregarNBolitas: tablero x celda x nat cantidad x color -> celda
75 mayorCeldaDeBolitas: tablero -> celda
76 mayorCeldaDeBolitasAux: tablero x conj(celda) x celda -> celda
77 #bolitasEnCelda: tablero x conj(colores) x celda -> nat
78 crearYAgregar: tablero x direccion -> conj(celda)
79 crearYAgregarAux: tablero x posicion x direccion -> conj(celda)
80 agregarAOeste: tablero -> tablero
81 moverPosicionesOeste: tablero x conj(celda) -> tablero
82 crearNuevaEnOeste: tablero x posicion -> conj(celda)
83 agregarASur: tablero -> tablero
84 moverPosicionesSur: tablero x conj(celda) -> tablero
85 crearNuevaEnSur: tablero x celda -> tablero

```

```

1  Axiomas:
2
3  dimension(IniciarTablero()) = (1, 1)
4  // Me fijo las celdas. Si al mover la posicion entonces es 1 o 0, aumento en uno la dimension
5  // Esto se hace, si la posicion es igual al ancho o algo, o es igual a -1, aumento el tamaño respectivamente
6  dimension(Mover(t, d)) =
7      if n2(celda.posicion(celda.mover(cabezal(t), d))) = n2(dimension(t)) v
8          n2(celda.posicion(celda.mover(cabezal(t), d))) = -1 then
9          (n1(dimension(t)), n2(dimension(t) + 1))
10     else
11         if n1(celda.posicion(celda.mover(cabezal(t), d))) = n1(dimension(t)) v
12             n1(celda.posicion(celda.mover(cabezal(t), d))) = -1 then
13             (n1(dimension(t)) + 1, n2(dimension(t)))
14         else
15             dimension(t)
16         fi
17     fi
18 dimension(AgregarBolitas(t, c, cant, p)) = dimension(t)
19
20 celdas(IniciarTablero()) = Ag(celda.vacia((0, 0)) 0)
21 celdas(Mover(t, d)) =
22     if n1(celda.posicion(celda.mover(cabezal(t), d))) = -1 then
23         agregarAOeste(t)
24     else
25         if n2(celda.posicion(celda.mover(cabezal(t), d))) = -1 then
26             agregarASur(t)
27         else
28             if n1(celda.posicion(celda.mover(cabezal(t), d))) = n1(dimension(t)) then
29                 crearYAgregar(t, d)
30             else
31                 if n2(celda.posicion(celda.mover(cabezal(t), d))) = n2(dimension(t)) then
32                     crearYAgregar(t, d)
33                 else
34                     posiciones(t)
35                 fi
36             fi
37         fi
38     fi
39
40 celdas(AgregarBolitas(t, c, cant, p)) = posiciones(t)
41
42 celdas(AgregarBolitas(t, c, cant, p)) = posiciones(t)
43
44 // Si la posicion actual al mover es -1 en Y, entonces muevo hacia izquierda manteniendo el X pero dejando iniciado Y en 0
45 // pues creo una nueva columna a izquierda
46 if n1(celda.posicion(celda.mover(cabezal(t), d))) = -1 then
47     celda.vacia((n1(celda.posicion(0, n2(celda.posicion(cabezal(t))))))
48 else
49     // Si la posicion actual al mover es -1 en X, entonces muevo hacia abajo manteniendo el Y pero dejando iniciado X en 0
50     // pues creo una nueva fila abajo
51     if n2(celda.posicion(celda.mover(cabezal(t), d))) = -1 then
52         celda.vacia((n1(celda.posicion(n1(celda.posicion(cabezal(t))), 0))
53     else
54         // Si no, dejo aumentada la posicion en 1, pues esto es automatico y en otra funcion se crea la fila
55         celda.mover(cabezal, d)
56     fi
57 fi
58
59 cabezal(AgregarBolitas(t, c, cant, p)) = cabezal(t)
60
61 celdasConBolitas(IniciarTablero()) = 0
62 // Esto devuelve las celdas porque la operacion solo permite mover, no agregar nuevas bolitas
63 celdasConBolitas(Mover(t, d)) = celdasConBolitas(t, d)
64 celdasConBolitas(AgregarBolitas(tablero, color, cantidad, celda)) =
65     if cantidad = 0 then
66         celdasConBolitas(t)
67     else
68         Ag(celdasConBolitas(t), celda)
69     fi
70
71 bolitasXCelda(Mover(t, d), celda) = celdasConBolitas(t, celda)
72 bolitasXCelda(AgregarBolitas(t, col, cant, celda'), celda) =
73     if celda' = celda then
74         resultadoNuevaCelda(t, col, cant, agregarNBolitas(celda))
75     else
76         bolitasXCelda(t, celda)
77     fi
78
79 resultadoNuevaCelda(t, col, cant, celda) =
80     (celda.nroBolitas(celda, Azul),
81     celda.nroBolitas(celda, Rojo),
82     celda.nroBolitas(celda, Verde),
83     celda.nroBolitas(celda, Negro))
84
85 agregarNBolitas(tablero, celda, cant, color) =
86     if cant = 0 then
87         celda
88     else
89         agregarNBolitas(tablero, celda.poner(celda, color), cant-1, color)
90     fi
91

```

```

1 mayorCeldaDeBolitas(tablero) =
2 if 0?(celdasConBolitas(t)) then
3   celda.vacia((0, 0))
4 else
5   if #celdasConBolitas(t) = 1 then
6     dameUno(celdasConBolitas(t))
7   else
8     mayorCeldaDeBolitasAux(tablero, sinUno(celdasConBolitas(tablero)), dameUno(celdas))
9
10 mayorCeldaDeBolitasAux(tablero, cc, c) =
11   if 0?(cc) then
12     c
13   else
14     if #bolitasEnCelda(tablero, {Azul, Rojo, Verde, Negro} , c) < #bolitasEnCelda(tablero, {Azul, Rojo, Verde, Negro} , dameUno(cc)) then
15       mayorCeldaDeBolitasAux(tablero, sinUno(cc), dameUno(cc))
16     else
17       mayorCeldaDeBolitasAux(tablero, sinUno(cc), c)
18   fi
19
20 #bolitasEnCelda(tablero, colores, celda) =
21   if 0?(colores) then
22     0
23   else
24     nroBolitas(celda, dameUno(colores)) + #bolitasEnCelda(tablero, sinUno(colores), celda)
25   fi
26
27 crearYAgregar(tablero, direccion) =
28   if direccion = Norte then
29     crearYAgregarAux(tablero, (n1(dimension(t)), 0), d)
30   else
31     crearYAgregarAux(tablero, (0, n2(dimension(t))))
32   fi
33
34 //Si la direccion es norte, entonces la posicion tiene que ser el ( n1(dimension(t)), 0)
35 crearYAgregarAux(tablero, posicion, direccion) =
36   if direccion = Norte then
37     if n2(posicion) < n2(dimension(t)) then
38       Ag(celda.vacia(( n1(posicion), n2(posicion) + 1 )), crearYAgregarAux(tablero, ( n1(posicion), n2(posicion) + 1 ), direccion ))
39     else
40       0
41     fi
42   else
43     if n1(posicion) < n1(dimension(t)) then
44       Ag(celda.vacia(( n1(posicion) + 1, n2(posicion))), crearYAgregarAux(tablero, ( n1(posicion) + 1, n2(posicion)), direccion ))
45     else
46       0
47     fi
48   fi
49
50 agregarAOeste(tablero) = crearNuevaEnOeste(t, (0, 0)) u moverPosiciones(t, posiciones(t))
51 moverPosicionesOeste: tablero, conj(Posicion) -> conj(Posicion)
52
53 // Asumo que mover una posicion, la mueve junto con sus bolitas internas
54 moverPosicionesOeste(t, c) =
55   if 0?(c) then
56     0
57   else
58     Ag( ( n1(dameUno(c)) + 1 , n2(dameUno(c))), moverPosiciones(t, sinUno(c)) )
59   fi
60
61 //recibe el tablero y la posicion <0, 0>, y aumenta el n2 conforme sea menor al tamaño (n2 es el X, es decir, el vertical)
62 crearNuevaEnOeste(tablero, pos) =
63   if n2(pos) < n2(dimension(t)) then
64     Ag(celda.vacia((0, n2(pos))), crearNuevaEnOeste(t, (0, n2(pos) + 1)))
65   else
66     0
67   fi
68
69 // Asumo que mover una posicion, la mueve junto con sus bolitas internas
70 agregarASur(tablero) = crearNuevaEnSur(t, (0, 0)) u moverPosiciones
71 moverPosicionesSur: tablero, conj(Posicion) -> conj(Posicion)
72 moverPosicionesSur(t, c) =
73   if 0?(c) then
74     0
75   else
76     Ag(( n1(dameUno(c)) , n2(dameUno(c)) + 1), moverPosiciones(t, sinUno(c)) )
77   fi
78
79 //recibe el tablero y la posicion <0, 0>, y aumenta el n1 conforme sea menor al tamaño (n1 es el Y, es decir, horizontal)
80 crearNuevaEnOeste(tablero, pos) =
81   if n1(pos) < n1(dimension(t)) then
82     Ag(celda.vacia((n1(pos), 0)), crearNuevaEnOeste(t, ( n1(pos) + 1, 0)))
83   else
84     0
85   fi
86
87 Fin TAD

```