



Figura 1: Tableros de ejemplo

Los ejercicios que corresponden a los **contenidos mínimos** recomendados se encuentran marcados con el símbolo \otimes

1. Valores, expresiones y tipos

Ejercicio 1

\otimes Indicar cuáles de las siguientes tiras de símbolos son expresiones de Gobstones (es decir, denotan un valor). Además, para aquellas que sean expresiones, indicar si son literales o no y cuál es su tipo (i.e., si denota un número, una dirección o un color). **Justifique.**

1. 7
2. 1+6
3. +
4. -1+6
5. 4 div 5
6. 4 mod 5
7. Verde
8. Rojo+Azul
9. Siguiente(Rojo)
10. siguiente(Rojo)
11. siguiente(Este)
12. maxColor(Rojo)
13. siguiente(maxColor())
14. Mover(Este)
15. opuesto(previo(minDir()))
16. x+5, con x parámetro
17. MoverN(2 * n), con n parámetro
18. Poner(c), con c parámetro
19. siguiente(c), con c parámetro
20. siguiente(Poner(Rojo))

Ejercicio 2

\otimes Agrupar las siguientes expresiones de forma tal que las expresiones que denotan el mismo valor pertenezcan al mismo grupo, suponiendo que se evalúan en el tablero de la Figura 1 (a). ¿Cómo habría que agruparlas si el tablero fuera el de la Figura 1 (b) y (c)?

<code>minColor()</code>	Azul	<code>nroBolitas(Azul)</code>	--3
<code>nroBolitas(Azul) mod 4</code>	Este	<code>siguiente(minDir())</code>	3
<code>nroBolitas(Azul) div 4</code>	<code>puedeMover(Este)</code>	<code>nroBolitas(Verde)</code>	Verde

Observar que expresiones diferentes pueden denotar el mismo valor, y que una expresión puede denotar distintos valores en distintos tableros.

Ejercicio 3

\otimes Detectar el error que tiene el siguiente programa. Justificar.

```
procedure A(c){Poner(c)}
program{A(9)}
```

Discutir por qué es importante documentar la naturaleza (tipo) de un parámetro junto con el propósito y la precondition.

Ejercicio 4

⊗ Escribir expresiones que denoten

1. la cantidad de bolitas total (de los 4 colores) en la celda actual;
2. el doble de la cantidad de bolitas rojas en la celda actual;

Ejercicio 5

Indicar cuáles de las siguientes invocaciones están bien escritas en GOBSTONES, teniendo en cuenta que `Pepe(c,d,n)` es un procedimiento que recibe un color `c`, una dirección `d` y un número `n`. **Justificar**.

- `Pepe(Azul, Este, 5)`
- `Poner(Poner)`
- `Poner(Mover(Este), Azul)`
- `Poner(Poner(Azul))`
- `Pepe(Poner(Azul), Mover(Este), 4)`
- `Pepe(minColor(), minDir(), 7 div 5)`
- `Pepe(siguiete(siguiete(minColor())), Este, nroBolitas(Azul))`

Ejercicio 6

⊗ Indicar cuáles de los siguientes procedimientos están bien escritos en GOBSTONES, donde `Quique(n, c)` es un procedimiento que recibe un numero `n` y un color `c`. **Justificar** todas sus afirmaciones.

```
procedure Pepe(n, c) {
  // n: n\ '{u}mero, c: color
  Poner(c); Mover(Este)
  Quique(n, c)
}
```

```
procedure Francisco(c) {
  // c: color
  Sacar(nroBolitas(c))
  nroBolitas(c) + 4
}
```

```
procedure Papa() {
  Poner(Azul)
  Quique(nroBolitas(Azul), minColor())
}
```

```
procedure Benedicto() {
  Poner(Poner(Poner(Azul)))
  opuesto(Este)
}
```

2. Repetición

Ejercicio 7

⊗ Corresponder las siguientes oraciones que describen propósitos y preconditiones con sus respectivos procedimientos.

- Mueve el cabezal tantas posiciones al Este como indica el parámetro n
- Hay tantas celdas al este como bolitas rojas hay en la celda actual.
- Pone una bolita negra y una roja
- Si $n > 0$, entonces pone n bolitas de color c ; caso contrario, saca $-n$ bolitas de color c .
- Mueve el cabezal 100 posiciones al Este.
- No tiene precondition.
- Hay al menos n celdas al Este de la actual.
- Dibuja un cuadrado de color c de lado 2 hacia el Noreste.
- Hay al menos $-n$ bolitas de color c .
- El cabezal no se encuentra en el extremo Este ni en el extremo Norte.
- Mueve el cabezal al este en tantas posiciones como bolitas rojas hay en la celda actual.
- Dibuja un cuadrado de lado 2 en las que cada celda tiene una bolita de cada color.
- Hay al menos 100 celdas al este de la actual.

```

procedure A() {
  repeat(100) {
    Mover(Este)
  }
}

procedure B(n) {
  repeat(n) {
    Mover(Este)
  }
}

procedure C() {
  repeat(nroBolitas(Rojo)) {
    Mover(Este)
  }
}

procedure D() {
  foreach c in [Negro..Rojo] {
    Poner(c)
  }
}

procedure E(c) {
  foreach d in [minDir()..maxDir()] {
    Poner(c); Mover(d)
  }
}

procedure F(c, n) {
  repeat(n) {
    Poner(c)
  }

  repeat(-n) {
    Sacar(c)
  }
}

procedure G(n, c) {
  foreach c in [minColor()..maxColor()] {
    E(c)
  }
}

```

Ejercicio 8

Escribir un procedimiento `PonerCincuenta` usando `repeat` que ponga 50 bolitas azules en la celda actual.

Ejercicio 9

Escribir el procedimiento `PonerTantasVerdesComoRojas` que deposita, en la celda actual, tantas bolitas verdes como bolitas rojas hay en la misma.

Ejercicio 10

Usando como base el procedimiento anterior, escribir un procedimiento `PonerTantasXComoY(x,y)` que, dados dos colores `x` e `y`, ponga en la celda actual tantas bolitas de color `x` como bolitas de color `y` haya en la celda actual. ¿Cómo se debería invocar `PonerTantasXComoY` para resolver el ejercicio anterior?

Ejercicio 11

⊛ Escribir un procedimiento `PonerN(n, c)` que tenga como parámetros una cantidad `n` y un color `c` cuyo propósito sea colocar `n` bolitas de color `c` en la celda actual. ¿Qué ocurre cuando se invoca `PonerN(0, Azul)`? ¿Y si invocamos `PonerN(-4, Verde)`? ¿Cuál sería, entonces, la precondition de `PonerN`? Resolver el ejercicio anterior utilizando `PonerN` en conjunto con la operación `numeroBolitas`. ¿Cuál es la ventaja de utilizar `PonerN`?

Ejercicio 12

⊛ Escribir el procedimiento `SacarN(n, c)` que, dado un número `n` y un color `c`, saque `n` bolitas de color `c` de la celda actual. Suponiendo que no hay bolitas en la celda actual ¿Qué ocurre cuando se invoca `SacarN(0, Azul)`? ¿Y si invocamos `SacarN(-4, Verde)`? ¿Cuál sería, entonces, la precondition de `SacarN`?

Ejercicio 13

⊛ Utilizando `SacarN`, escribir el procedimiento `SacarTodas(c)` que, dado un color `c`, saque **todas** las bolitas de color `c` de la celda actual. ¿Este procedimiento funciona siempre?

Ejercicio 14

⊛ Escribir un procedimiento `PonerEnLindantes(c)` usando `foreach`, `minDir` y `maxDir`, que reciba un color `c` y deposite una bolita de color `c` en cada una de las celdas lindantes (al N, E, S, O) de la celda actual. Se recomienda utilizar un procedimiento auxiliar que reciba un color `c` y una dirección `d` y ponga una bolita de color `c` en la celda lindante en dirección `d` sin mover el cabezal.

Ejercicio 15

⊛ Escribir un procedimiento `PonerTodosLosColores` que deposite una bolita de cada color en la celda actual. Haga uso de `foreach`, `minColor` y `maxColor`. Este procedimiento es equivalente al procedimiento `PonerUnaDeCada` de la Práctica 2 ¿Qué procedimiento le parece mejor escrito? ¿Y cuál preferiría si GOBSTONES tuviera 32000 colores?

Ejercicio 16

⊛ Utilizando `SacarTodas`, escribir el procedimiento `LimpiarCelda` que saque todas las bolitas de la celda actual. ¿Este procedimiento funciona siempre? ¿Cómo se comporta su procedimiento si los `adornosdeGOBSTONES` agregan los colores `Amarillo`, `Celeste`, y `Violeta` al lenguaje (notar

Ejercicio 17

Escribir el procedimiento `EscaleraSeisVerdeAlEste` que ponga 1 bolita verde en la celda actual, 2 bolitas verdes en la celda lindante al este, 3 bolitas verdes en la celda que le sigue al este, y así siguiendo hasta poner 6 bolitas en la celda que se encuentra a 5 celdas de distancia de la celda actual.

Ejercicio 18

⊗ Usando como base el ejercicio anterior, escribir el procedimiento `Escalera(n,c,d)` que, dado un número `n`, un color `c` y una dirección `d`, recorra las primeras `n` celdas en dirección `d`, poniendo 1 bolita de color `c` en la celda actual, 2 bolitas de color `c` en la siguiente celda, 3 bolitas de color `c` en la siguiente, etc. Ejecutar `Escalera(3,Verde,Este)` en un tablero que tenga 3 columnas en el que el cabezal se encuentre en el extremo oeste. Si el procedimiento hace BOOM, modificarlo para que funcione en este caso.

Ejercicio 19

Escribir un procedimiento `CopiarCeldaAlNorte` que copie todas las bolitas de la celda actual a la celda lindante al Norte.

Ejercicio 20

⊗ Escribir un procedimiento `CopiarCeldaAl(d)` que reciba por parámetro una dirección `d` y copie las bolitas a la celda lindante en dirección `d`. Para ello, se recomienda escribir el procedimiento `PonerNA1(c,d, n)` que reciba un color `c`, una dirección `d` y un número `n` y ponga `n` bolitas de color `c` en la celda lindante en dirección `d`, dejando el cabezal en la celda inicial.

Ejercicio 21

⊗ Utilizando el procedimiento `CopiarCeldaAl(d)`, escribir `CopiarCeldaALindantes` que es un procedimiento que copia las bolitas de la celda actual a cada una de sus celdas lindantes (al Norte, Este, Sur y Oeste).

Ejercicio 22

⊗ Utilizando nuevamente el procedimiento del ejercicio 20, escribir un procedimiento que copie las bolitas de la celda actual a sus celdas lindantes incluyendo las diagonales (al Norte, Noreste, Este, Sureste, Sur, Suroeste, Oeste y Noroeste). El nuevo procedimiento debe llamarse `CopiarCeldaAAyacentes`.