

7. Especificación de un recurso compartido: mezclador 2-a-1

En este ejercicio os pedimos que especifiquéis formalmente una pequeña variación del almacén de un dato. En este caso habrá dos productores – llamados *izquierdo* y *derecho* – y un único consumidor. La idea es que ambos productores generan *secuencias ordenadas* de números enteros y el consumidor extrae del almacén *una única secuencia ordenada* resultante de mezclar las secuencias izquierda y derecha.

La interacción entre productores y consumidores viene dada por un recurso Mezclador2a1 con capacidad para almacenar un dato de cada uno de los productores. El consumidor invoca la acción `get` que extrae el menor de ambos datos cuando los dos están presentes en el mezclador. Debe bloquear si alguno de los datos falta.

Por su parte, los productores invocan la acción `put(lado,dato)`, donde $lado \in \{Izda, Dcha\}$. La llamada debe bloquear si el anterior dato de dicho productor aún no ha sido extraído.

Para hacer más útil este recurso vamos a permitir *múltiples secuencias* de números. Para ello, tendremos una acción `fin(lado)` que permite a cada productor indicar el fin de una secuencia. Así, si el productor izquierdo indica fin de secuencia el consumidor podrá extraer números de la secuencia derecha hasta que el productor derecho indique fin de secuencia, y viceversa. Cuando ambos productores han marcado fin de secuencia se vuelve al estado inicial y se mezclan las nuevas secuencias izquierda y derecha. Añadiremos un segundo parámetro booleano de salida a la acción `get` para notificar al consumidor del fin de un par de secuencias.

Por ejemplo, si el productor izquierdo invoca las llamadas

```
put(Izda,3); put(Izda,11); put(Izda,19); fin(Izda); put(Izda,2); put(Izda,13)
```

y el derecho invoca

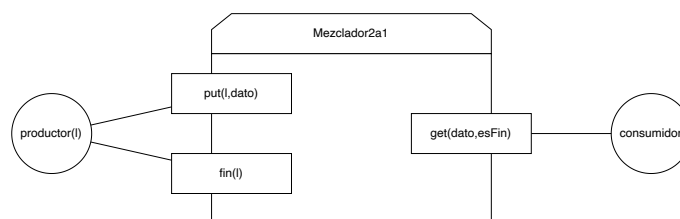
```
put(Dcha,5), put(Dcha,7), fin(Dcha), put(Dcha,3)
```

entonces el consumidor podrá ejecutar

```
get(3,false); get(5,false); get(7,false); get(11,false); get(19,false); get(2,true); get(13,false); get(3,false)
```

Consejo: Especifica primero el recurso *sin* el tratamiento de múltiples secuencias y, cuando ya lo tengas, añade dicha funcionalidad.

A continuación mostramos un diagrama de procesos/recursos y un esqueleto de la especificación del recurso compartido que debéis completar:



C-TAD Mezclador2a1**OPERACIONES**ACCIÓN put: $Lado[e] \times \mathbb{Z}[e]$ ACCIÓN fin: $Lado[e]$ ACCIÓN get: $\mathbb{Z}[s] \times \mathbb{B}[s]$ **SEMÁNTICA****DOMINIO:**TIPO: $Lado = Izda \mid Dcha$ TIPO: $Mezclador2a1 = \mathbb{Z} \times \mathbb{B} \times \mathbb{B} \times \mathbb{Z} \times \mathbb{B} \times \mathbb{B}$ INICIAL: INVARIANTE: CPRE:

put(l,dato)

POST: CPRE:

fin(l)

POST: CPRE:

get(dato,esFin)

POST: