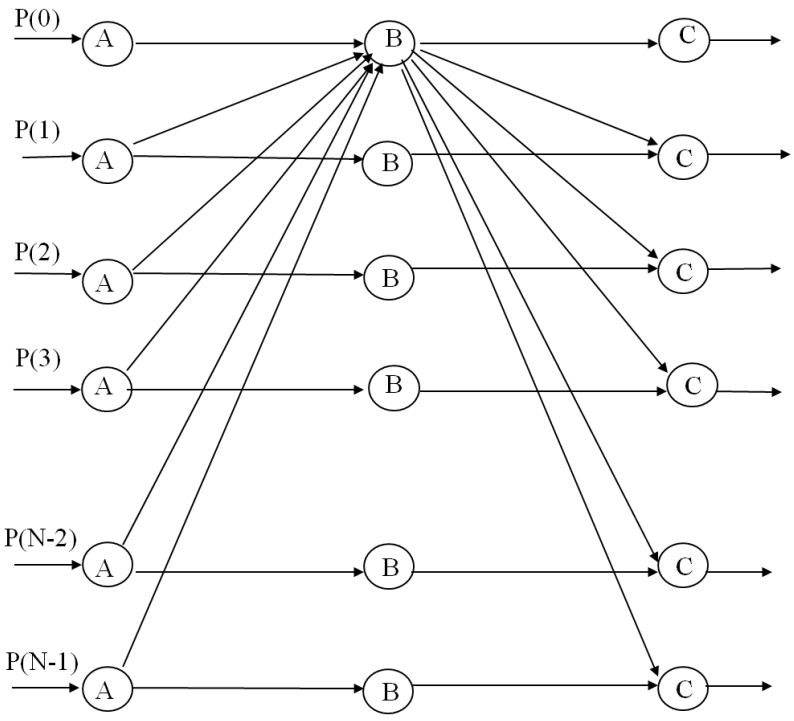
Ejercicios 4. SEMÁFOROS (I)

Ejercicio 1. El paquete *pc.ejercicios4i.simple* del proyecto "PC Ejercicios Enunciado 4. Semaforos (I)" de la plataforma muestra un programa concurrente formado por N procesos concurrentes cuyos ciclos de vida son los siguientes:

```
PROCESS Proceso(int idProceso) {
   A;
   B;
   C;
}
```

donde A, B y C son bloques de instrucciones que ya están codificados y no se permite su modificación.

Se desea resolver un problema de sincronización entre los procesos anteriores mediante un objeto de la clase *Sincronizacion*, intercalando entre los bloques de instrucciones anteriores llamadas a métodos de esa clase para asegurar que se cumpla el siguiente diagrama de precedencia en la ejecución de los bloques de instrucciones A, B y C por parte de los procesos.



Del anterior diagrama de precedencia, se deducen las siguientes restricciones:

a) El proceso P(0) no puede comenzar a ejecutar B hasta que todos los procesos hayan terminado de ejecutar A.

b) Ningún proceso P(i) con i > 0 puede comenzar a ejecutar C hasta que el proceso P(0) haya terminado de ejecutar B.

SE PIDE añadir todo lo necesario en la clase *Sincronizacion* para cumplir las anteriores especificaciones usando únicamente semáforos, y con el mínimo número posible de ellos.

A modo de ejemplo, ésta sería una traza de una ejecución correcta del programa:

```
A2 A1 B1 A0 A3 B2 A4 B3 B0 C0 B4 C1 C2 C4 C3
```

A modo de ejemplo, ésta sería una traza de una ejecución ERRÓNEA del programa:

```
A0 A4 B4 C4 B0 C0 A2 B2 A3 A1 C2 B3 B1 C3 C1
```

Ejercicio 2. El paquete *pc.ejercicios4i.ciclico* del proyecto "PC Ejercicios Enunciado 4. Semaforos (I)" de la plataforma muestra un programa concurrente formado por los mismos N procesos concurrentes del apartado anterior, pero que ejecutan repetidamente un bucle, y en cada paso del mismo ejecutan los bloques de instrucciones A, B y C.

Se desea resolver el mismo problema de sincronización del apartado anterior, pero de forma que los procesos se sincronicen de la manera anteriormente descrita **en cada paso del bucle**.

SE PIDE añadir todo lo necesario en la clase *Sincronizacion* para cumplir las anteriores especificaciones usando únicamente semáforos, y con el mínimo número posible de ellos.

Ejercicio 3. Busque en la bibliografía un algoritmo para resolver el problema de la sincronización de barrera cíclica para N procesos usando únicamente semáforos y variables compartidas.

Cite la fuente de donde obtenga el algoritmo, escríbalo en cualquier lenguaje de programación y comente su funcionamiento.