

## Esquema monitor túnel de Kiyotaki

Se utilizará un monitor con las siguientes variables, que clasificaremos como:

- Variable compartida (tipo)
- Método
- Variable no compartida (tipo)

### Lista de variables

- direction Var (Int):  $\begin{cases} 2 & \emptyset \\ 0 & \text{SOUTH} \\ 1 & \text{NORTH} \end{cases} \rightarrow \text{Dirección permitida al tránsito}$
- dir (Int):  $\begin{cases} 0 & \text{SOUTH} \\ 1 & \text{NORTH} \end{cases} \rightarrow \text{Dirección del coche actual}$
- crossing Var (Int): número de coches en el túnel
- waiting Array (Int): de longitud 2, guarda los que esperan en función de la dirección
- mutex : semáforo
- cross-allowed : condition (mutex)
- next hurry Var (Int):  $\begin{cases} 0 & \text{Nada} \\ 1 & \text{Mucha cola, hay que cambiar la dirección de túnel} \end{cases}$
- const-too-much-queue (Int): cuando lo está se considera "grande"

## Métodos definidos en la práctica

- `Wants_enter(self, direction):`

- Esta función asigna a dir la dirección del coche.
- Aumentamos el número de coches esperando en la dirección dada.
- Si hay mucha cola, cerramos el semáforo, cuando los coches salgan del túnel abrimos el semáforo en la dirección apropiada.
- Por último, tiene que esperar a que se pueda cruzar. Anoto el número de coches cruzando y reducimos el de coches esperando
- Liberar el semáforo

- `crossing_allowed(self):`

- Comprueba si la dirección permitida es la del coche en cuestión

- `leaves_tunnel(self, direction):`

- Reducimos el número de coches cruzando el túnel y vemos si se puede cambiar la dirección

- `ch_exit(self)`:

→ Si no hay coches cruzando cambia la dirección  
Si hay coches esperando o activa la variable  
que permite a los coches cambiar la dirección  
al intentar entrar al túnel

- `change(self, direction)`:

→ Cambia la dirección si se puede.