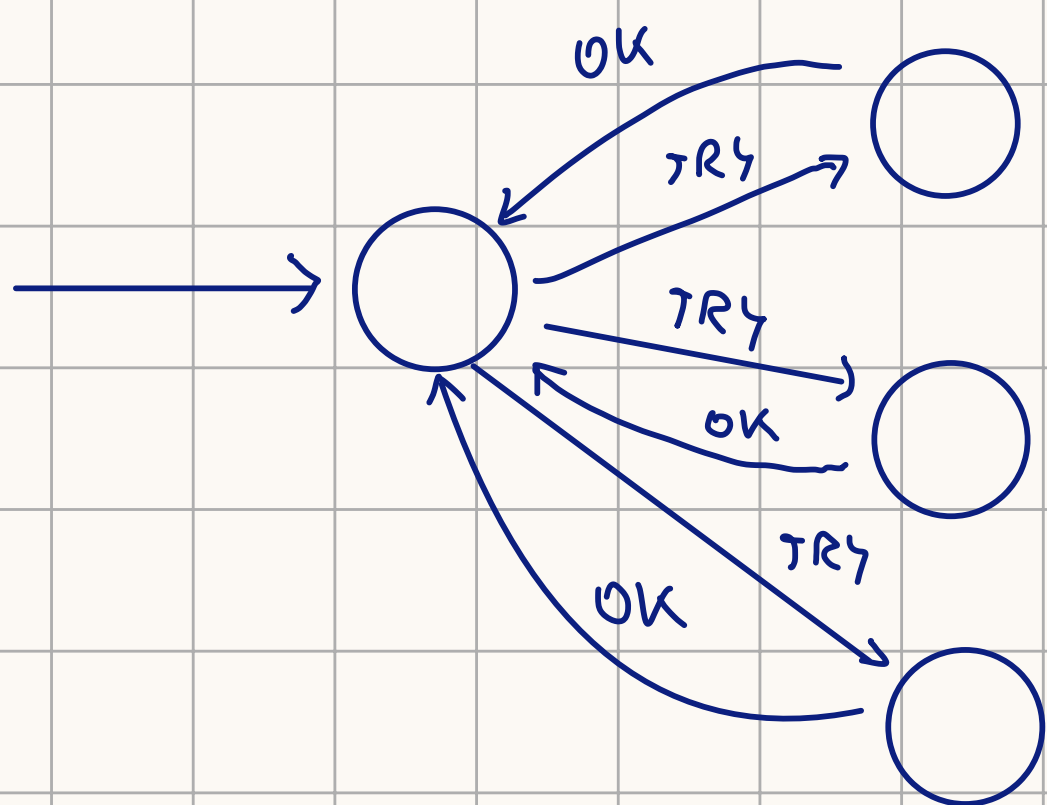
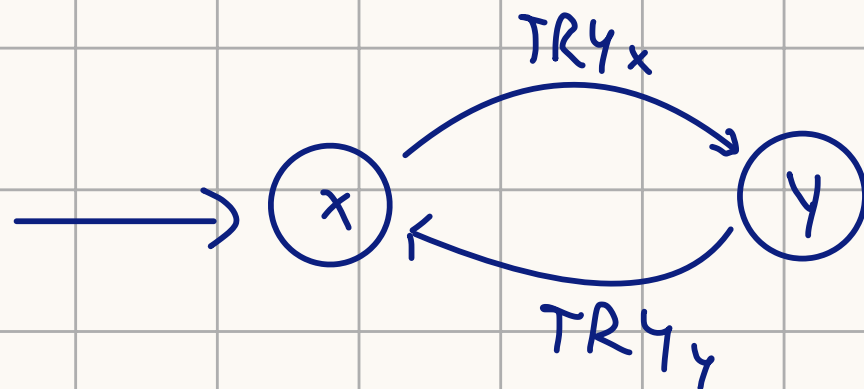


Algoritmo Exclusión Mutua: Basado en Try + Lamport

3 Algoritmo Informal

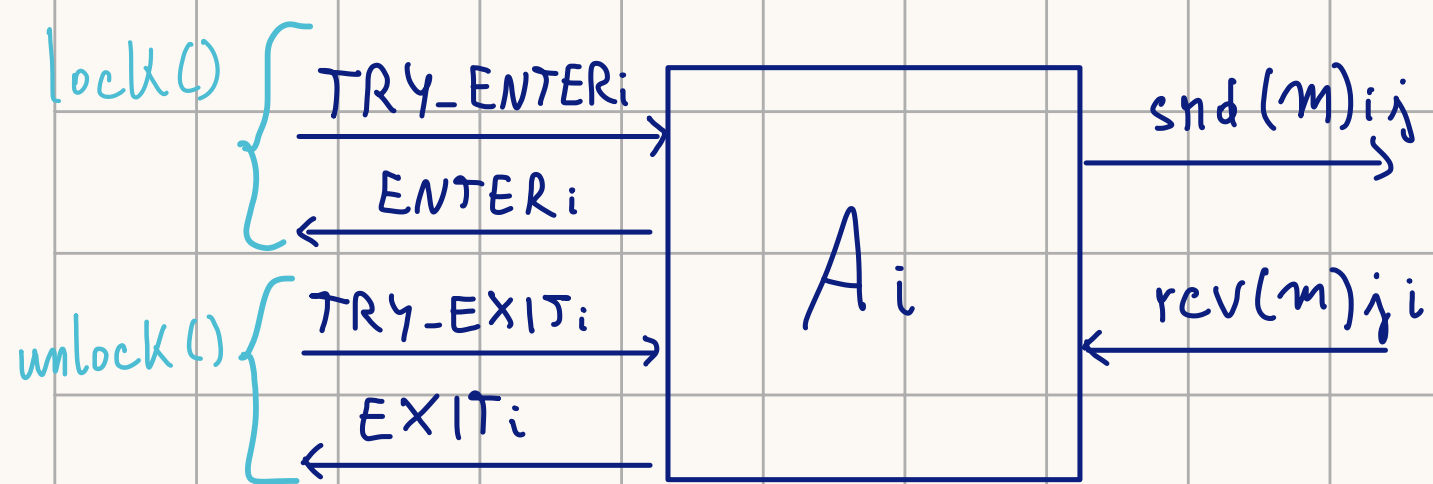


$$\text{Lamp}(\text{TRY}_x) < \text{Lamp}(\text{TRY}_y)$$



4 Algoritmo Formal

4.1 Signatura



4.3 Estado

$st \in \{\text{idle}, \text{trying}, \text{sc}, \text{exiting}\}$ init idle
 $qe_{ij} \forall j$, cola de mensajes, init \emptyset
 $qs_{ji} \forall i$, cola de mensajes, init \emptyset
 qn , cola de nodos pendientes, init \emptyset
 Lamp , entero, init 0
 $nOKs$, entero, init 0

4.2 Pasos

TRY_ENTER

efecto:

$st \leftarrow \text{trying}$

$\forall j \quad qs_{ij} \leftarrow qs_{ij} \cdot \text{TRY}[\text{Lamp}]$

$\text{Lamp} \leftarrow \text{Lamp}(\text{TRY_ENTER})$

rcv_try_sc

prec:

$qe_{ji} = \text{TRY}[\text{Lamp}] \cdot qe_{ji}' \wedge st = \text{sc}$

efecto:

$qe_{ji} \leftarrow qe_{ji}'$

$qn \leftarrow qn \cdot j$

rcv_try_idle

prec:

$qe_{ji} = \text{TRY}[\text{Lamp}] \cdot qe_{ji}' \wedge st = \text{idle}$

efecto:

$qe_{ji} \leftarrow qe_{ji}'$

$qs_{ij} \leftarrow qs_{ij} \cdot \text{OK}$

rcv_try_trying_l

prec:

$qe_{ji} = \text{TRY}[\text{Lamp}] \cdot qe_{ji}' \wedge st = \text{trying}$

$\wedge \text{Lamp} < \text{Lamp}(\text{TRY}_j)$

efecto:

$qe_{ji} \leftarrow qe_{ji}'$

$qn \leftarrow qn \cdot j$

recv_try_trying_loser

prec:

$qe_{ji} = TRY \cdot qe_{ji}'$

$\wedge st = trying$

$\wedge lamp > lamp(TRY)$

efecto:

$qe_{ji} \leftarrow qe_{ji}'$

$qs_{ij} \leftarrow qs_{ij} \cdot OK$

recv_OK

prec:

$qe_{ji} = OK \cdot qe_{ji}'$

efecto:

$nOKs \leftarrow nOKs + 1$

$qe_{ji} \leftarrow qe_{ji}'$

ENTER

prec: // suponemos que no llegan
OKs sin trying

$nOKs = p - 1$

efecto:

$st \leftarrow sc$

$nOKs \leftarrow 0$

TRY_EXIT

prec:

$st = sc$

efecto

$st \leftarrow exiting$

$qs_{ij} \leftarrow qs_{ij} \cdot OK$

$q_n \leftarrow \emptyset$

EXIT

prec:

$st = exiting$

efecto:

$st \leftarrow idle$

recv_try_exiting

prec:

$st = exiting \wedge qe_{ji} = TRY \cdot qe_{ji}'$

efecto:

$qs_{ij} \leftarrow qs_{ij} \cdot OK$

$qe_{ji} \leftarrow qe_{ji}'$

5. Tareas

* Se ponen todas las que tienen Prec.

$\{ \{snd(m)_i\}_{i \in p}, ENTER, EXIT, \{recv_try_*, INTERNAS\}, recv_OK \}$

* Las particiones se juntan

Propiedades

Seguridad \rightarrow Siempre ejecuciones finitas

$\forall \alpha \in \text{finejecs}(EM) : \alpha = \alpha_1 \cdot \text{ENTER}_i \cdot \alpha_2 \cdot \text{ENTER}_j \cdot \alpha_3 \Rightarrow$

$\alpha_2 = \alpha_{21} \cdot \text{EXIT}_i \cdot \alpha_{22}$ Máximo 1

$\forall \alpha \in \text{finejecs}(EM) : \alpha = \alpha_1 \cdot \text{ENTER}_i \cdot \alpha_2 \Rightarrow$

$\alpha_1 = \alpha_{11} \cdot \text{TRY_ENTER}_i \cdot \alpha_{12}$ Si sale o entra es pq lo pide
 TRY_EXIT_i

Viveza

$\forall \alpha \in \text{ejecs}(EM) : \alpha = \alpha_1 \cdot \text{TRY_EXIT}_i \cdot \alpha_2 \Rightarrow$

$\alpha_2 = \alpha_{21} \cdot \text{ENTER}_i \cdot \alpha_{22}$
 EXIT_i

Si pide entrar o salir, lo consigue

6. Demostración

Hay que hacer el trazado hacia atrás de lo que hay puesto en las propiedades

* Hay que argumentar de palabra el porqué de la razón por la que hay un exit entre dos TRY_ENTER

