

Exclusión Mutua

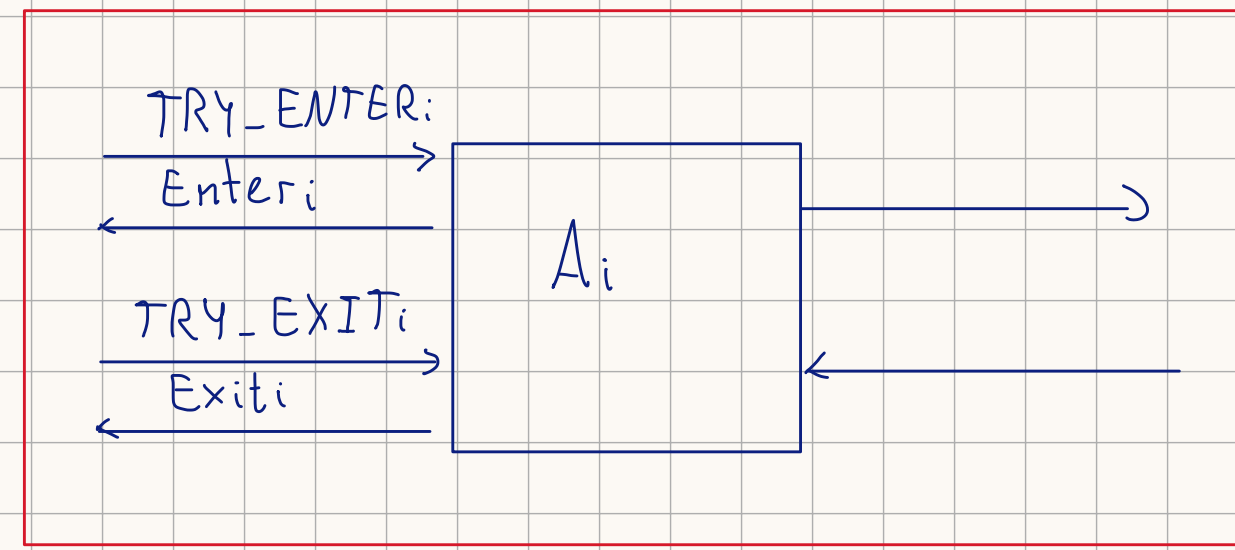
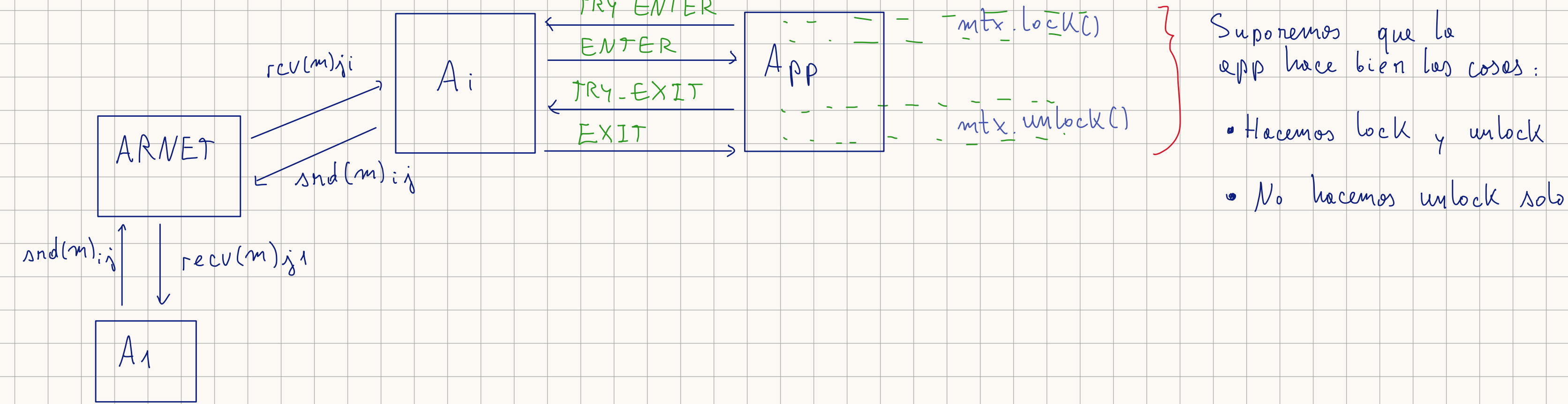
1. Todos los nodos acabarán entrando a la sección crítica **Viveza**
2. No puede haber más de un nodo a la vez en la sección crítica **Seguridad**
3. Si alguien entró es porque pidió entrar **Viveza**

Problema:

Informal:

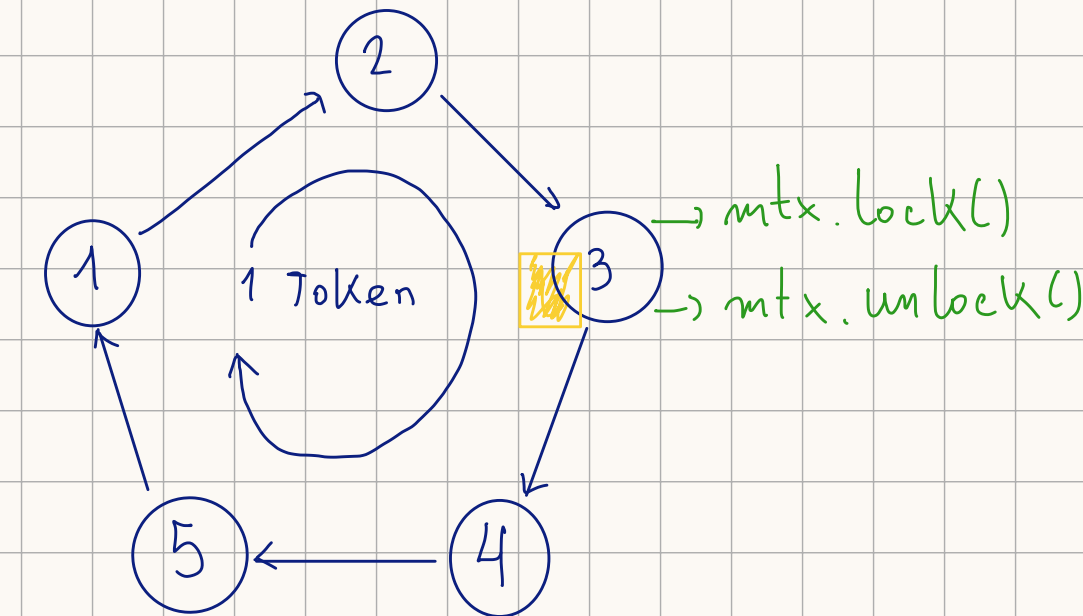
- * Varios hilos/nodos compiten por entrar a su sección crítica
- * Garantizando que como máximo 1 ejecutará la sección crítica en todo instante
- * Sin inanición: todo nodo logrará entrar antes o después.

Formal

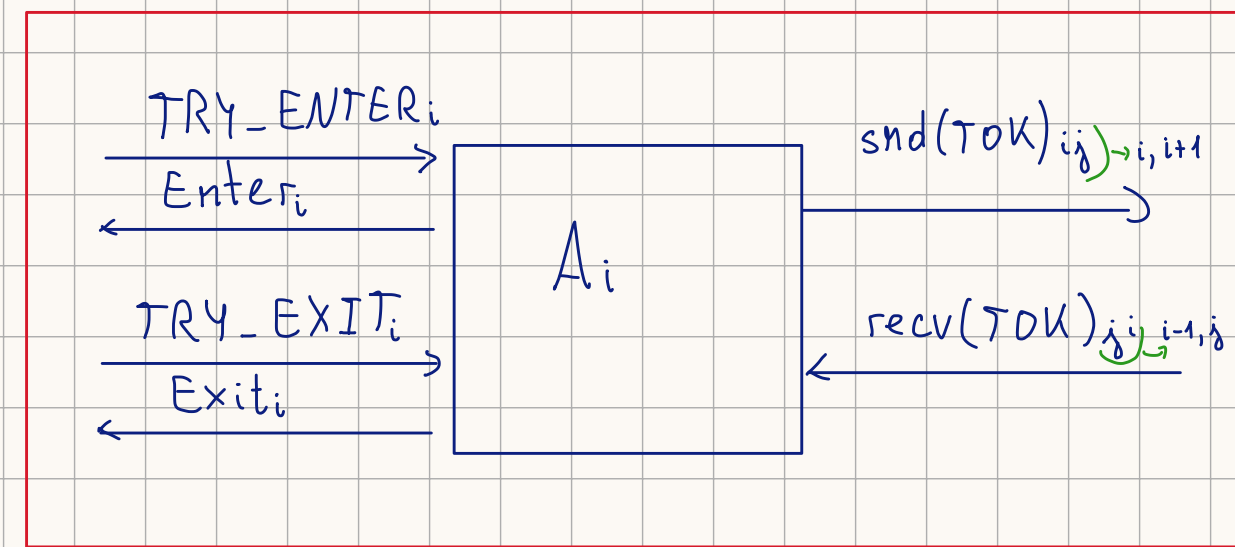


Token Circulante

3 Algoritmo Informal



4 Algoritmo Formal



Propiedades

Seguridad

Siempre ejecuciones finitas

$\forall a \in \text{finjecs}(EM): a = a_1 \cdot \text{ENTER}_i \cdot a_2 \cdot \text{ENTER}_s \cdot a_3 \Rightarrow$

$a_2 = a_{21} \cdot \text{EXIT}_i \cdot a_{22}$ Máximo 1

$\forall a \in \text{finjecs}(EM): a = a_1 \cdot \text{ENTER}_i \cdot a_2 \Rightarrow$

$a_1 = a_{11} \cdot \text{TRY-ENTER}_i \cdot a_{12}$ Si sale o entra es pq lo pide
 TRY-EXIT_i

Viveza

$\forall d \in \text{ejecjecs}(EM): d = d_1 \cdot \text{TRY-ENTER}_i \cdot d_2 \Rightarrow$

$d_2 = d_{21} \cdot \text{ENTER}_i \cdot d_{22}$ Si pide entrar o salir, lo consigue
 EXIT_i

5 Acciones

● TRY-ENTER_i
Prec: X
Efecto: $st \leftarrow \text{tryenter}$

ENTER_i
Prec: $st = \text{tryenter} \wedge tok = \text{true}$
Efecto: $st \leftarrow \text{sc}$

$\text{snd}(TOK)_{i,i+1}$
Prec: $q_{i,i+1} = TOK \cdot q'_{i,i+1}$
Efecto: $q_{i,i+1} \leftarrow q'_{i,i+1}$

6 Estados

$st \in \{\text{idle}, \text{tryenter}, \text{entering}, \text{sc}, \text{exiting}\}$, init: idle

$q_e: \{\text{Ids de mensajes}\}$, init: \emptyset
 $q_s:$

Q desde, hasta quien envia recibe

● TRY-EXIT_i
Prec: $st = \text{sc} \wedge tok = \text{true}$ No hace falta, asumimos que no pide nada hasta recibir una respuesta
Efecto: $st \leftarrow \text{exiting}$

EXIT
Prec: $st = \text{exiting} \wedge tok = \text{true}$
Efecto: $tok \leftarrow \text{false}$
 $st \leftarrow \text{idle}$
 $q_{i,i+1} \leftarrow q_{i,i+1} \cdot TOK$
 q_s

● $\text{rcv}(TOK)_{i+1,i}$
Efecto: $q_e \leftarrow q_{i+1,i} \leftarrow q_{i+1,i} \cdot TOK$

● Entradas, no tienen precondiciones

7 Tareas

- * Se ponen las funciones que tienen PRECONDICIÓN
- * Internas y salida

$\{\text{snd}, \text{rcv-tok}, \text{rcv-tok-sc}, \text{enter}, \text{exit}\}$

$\{\text{snd}, \{\text{rcv-tok}, \text{rcv-tok-sc}\}, \text{enter}, \text{exit}\}$

↓
Esta es otra forma de expresar estas dos son una partición (misma precondition) Son mutuamente excluyentes.

* INTERNAL

rcv-tok-sc
Prec: $q_{i+1,i} = TOK \cdot q'_{i+1,i} \wedge st = \text{tryenter}$
Efecto: $q_{i+1,i} \leftarrow q'_{i+1,i}$
 $tok = \text{true}$

rcv-tok
Prec: $q_{i+1,i} = TOK \cdot q'_{i+1,i} \wedge st \neq \text{tryenter}$
Efecto: $q_s \leftarrow q'_s \cdot TOK$ } Esto es un snd
 $q_e \leftarrow q'_{(i+1,i)}$ $q'_{(i+1,i)}$