

# **AOA: Arquitectures Sistèmiques**

---

## **Comparació de cadenes usant la distància de Levenshtein.**

Alumnes

Francisco J. Aguilar Celdrán  
Sergio Blanco Cuaresma

# AOA: Arquitectures Sistèmiques

Per realitzar el càlcul de la distància de Levenshtein, es construeix una taula a partir de dues paraules:

		A	B	C	D
	0	1	2	3	4
A	1				
B	2				
C	3				
D	4				

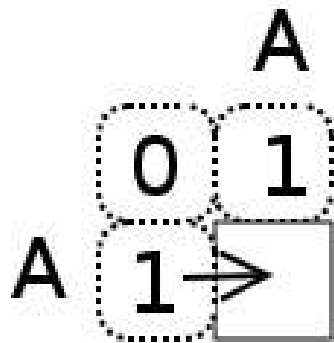
# AOA: Architectures Sistèmiques

A la construcció de la taula hi ha dependències que ens limiten:

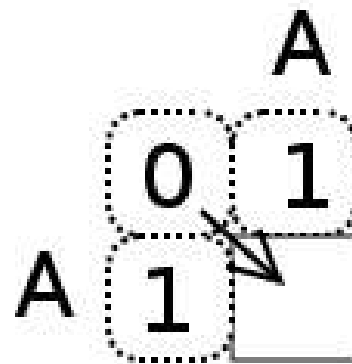
		A	B	C	D
	0	1	2	3	4
A	1				
B	2				
C	3				
D	4				

# AOA: Arquitectures Sistèmiques

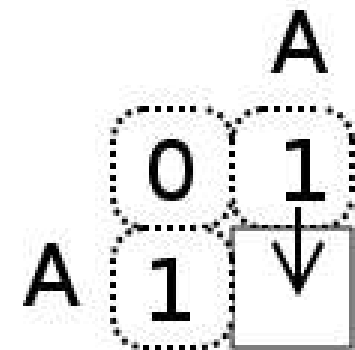
Dependències: Necessitem tenir ja calculat el valor abans de poder continuar



Esborrat



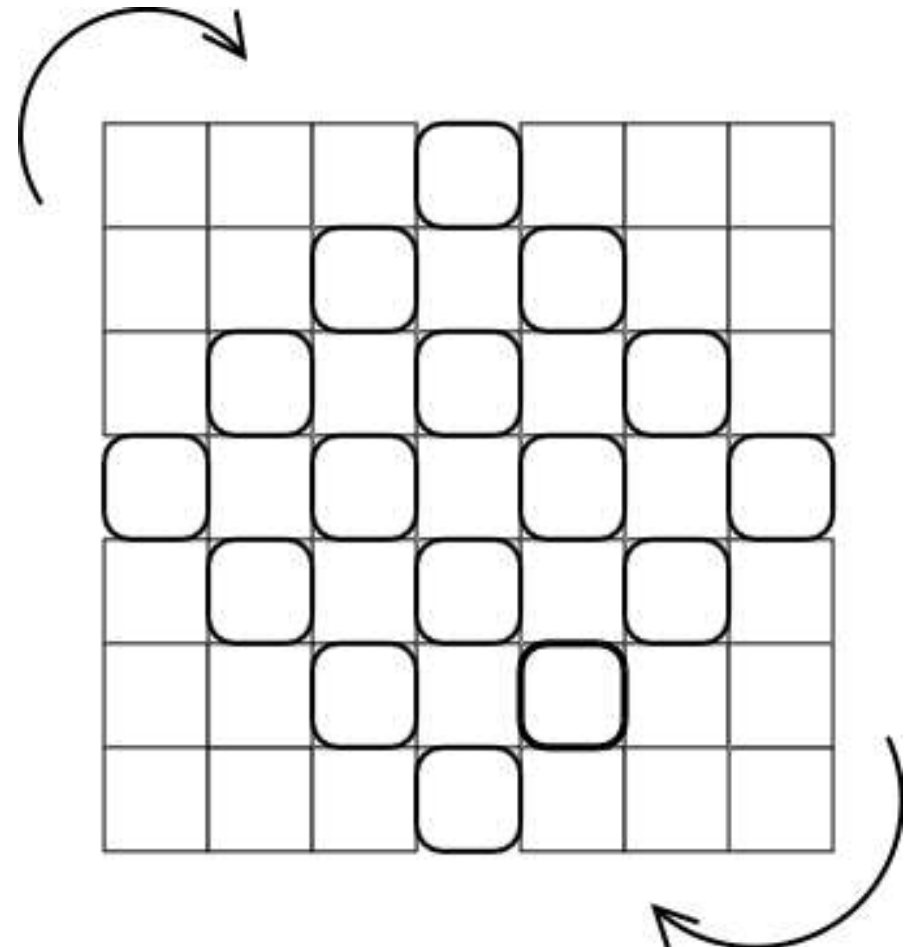
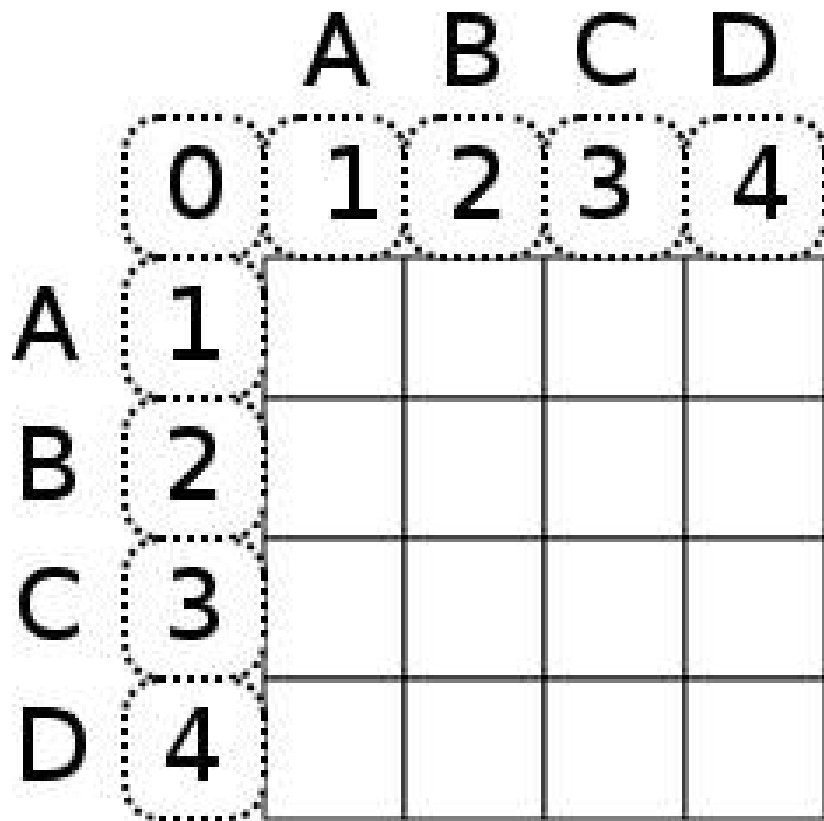
Substitució



Inserció

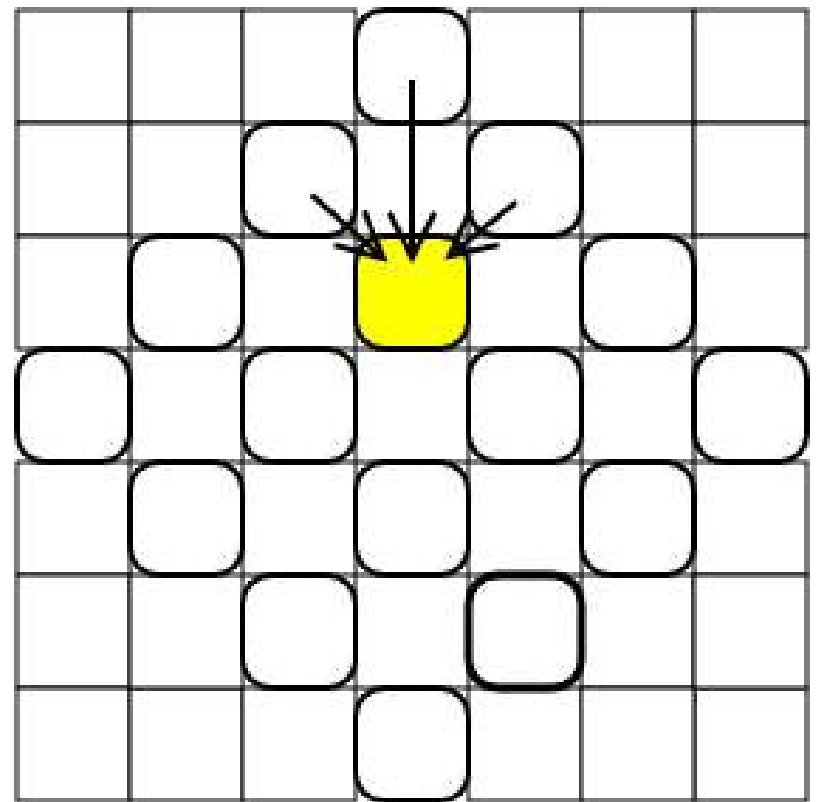
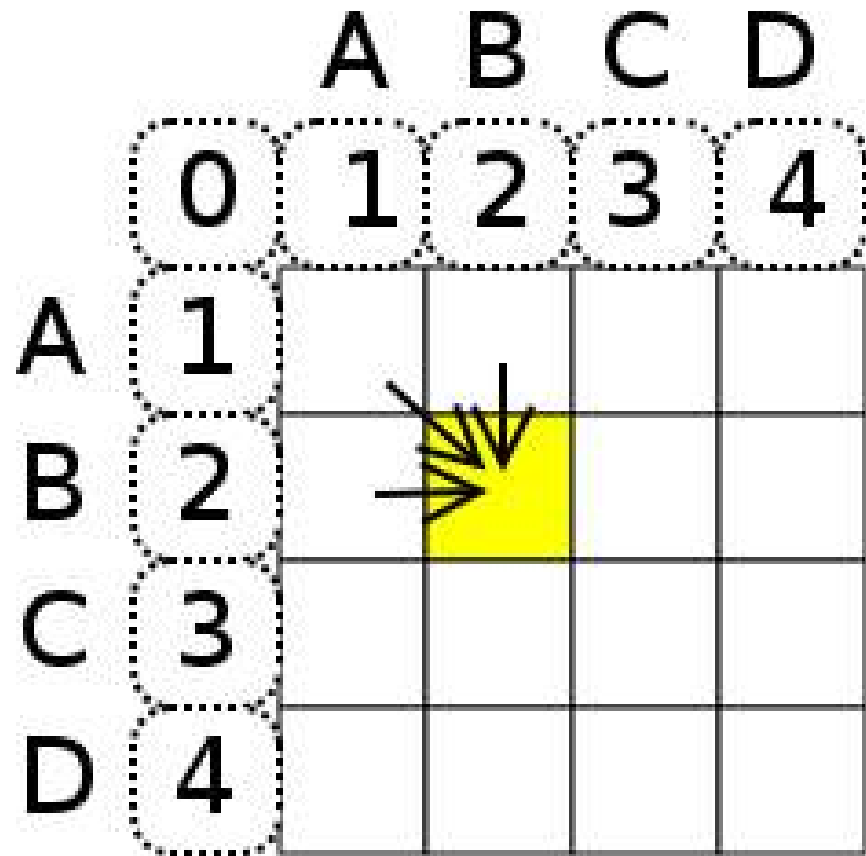
# AOA: Architectures Sistòliques

Solució a les dependències:



# AOA: Architectures Sistèmiques

## Solució a les dependències:



# AOA: Architectures Sistòliques

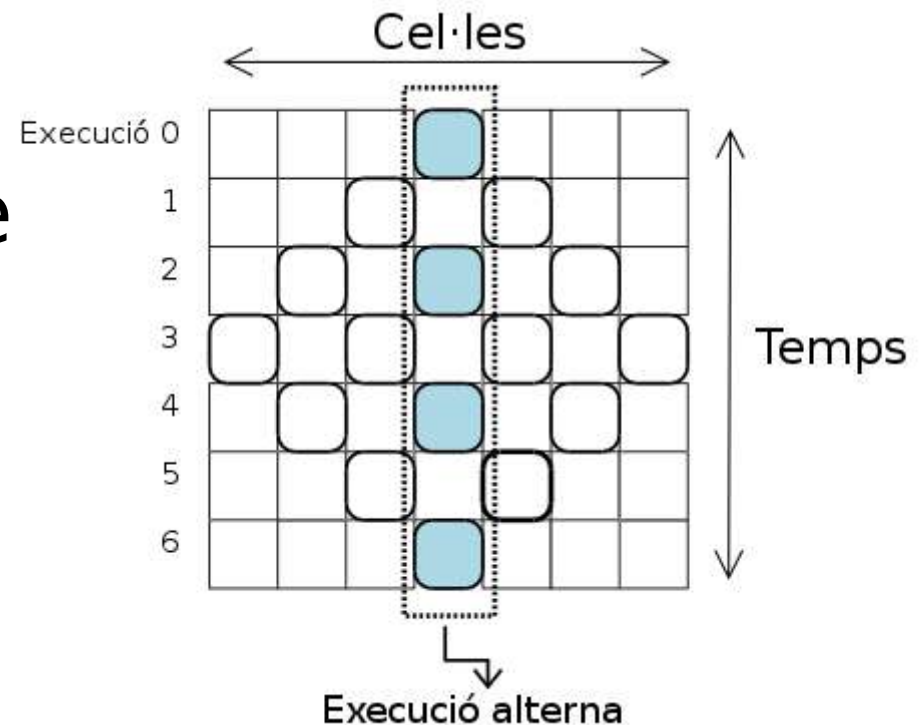
1 columna = 1 cel·la sistòlica

1 fila = 1 execució (1 iteració de bucle)

- Cada cel·la s'executa de forma **alternativa**.

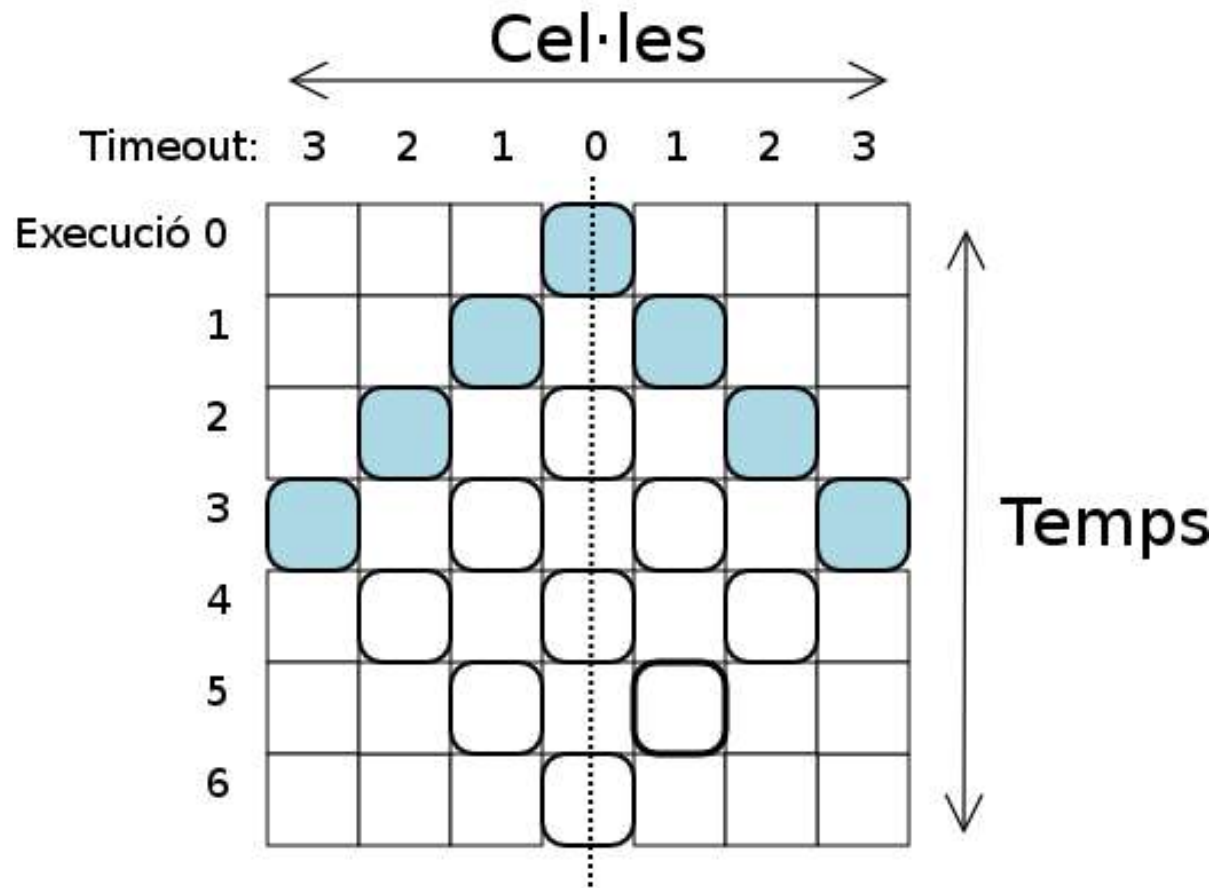
- Considerem només paraules de tamany fixe i idèntic.

- Nombre de cel·les = Nombre d'execucions totals = (tamany paraula x 2) - 1



# AOA: Arquitectures Sistèmiques

No totes les cel·les inicien l'execució a la vegada. Dividint la taula per la meitat observem que el moment d'inici de l'execució segueix una regla (**timeout**):





# AOA: Architectures Sistòliques

---

Condicions de càlcul per una cel·la:

- $\text{torn} == 1$

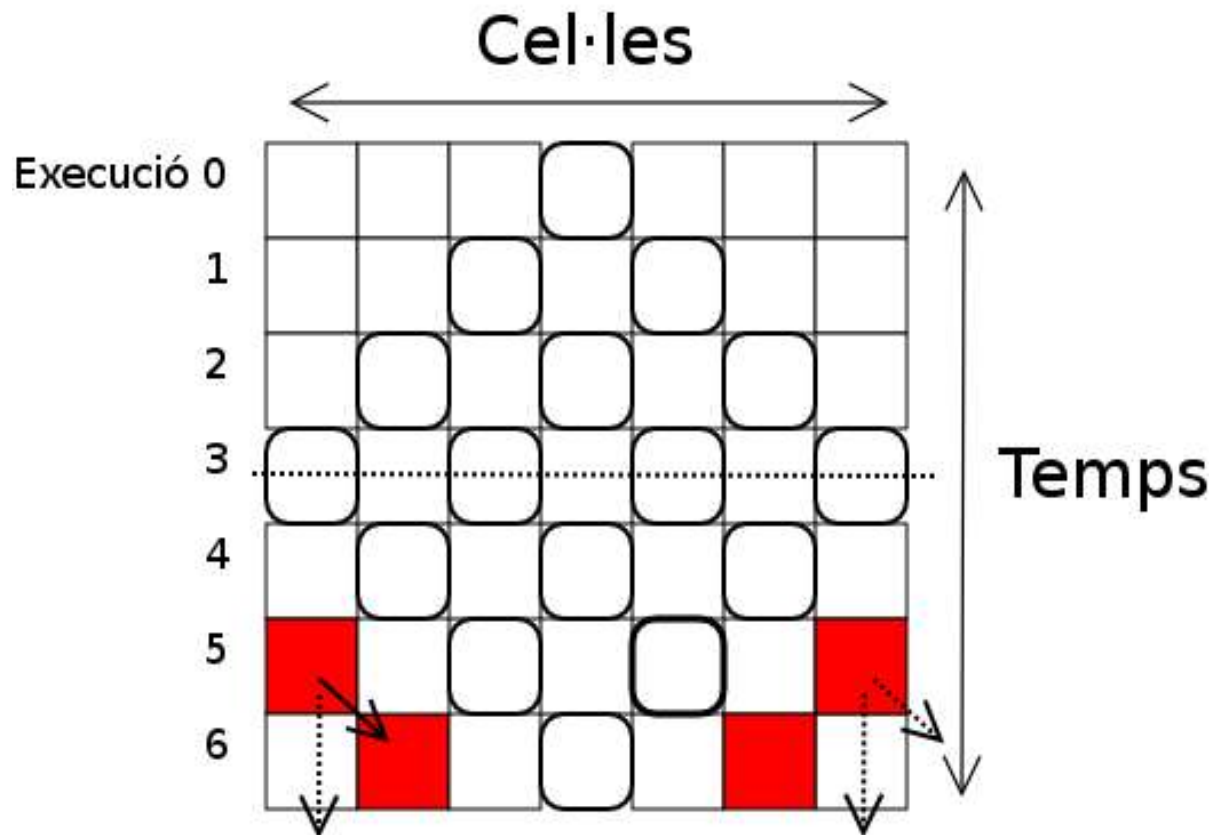
El torn s'inicialitza de forma alterna a cada cel·la i es canvia el seu valor de 0 a 1 i de 1 a 0 a cada execució.

- $\text{timeout} \leq 0$

El timeout s'inicialitza de forma individual a cada cel·la i es decrementa en 1 a cada execució.

# AOA: Arquitectures Sistèmiques

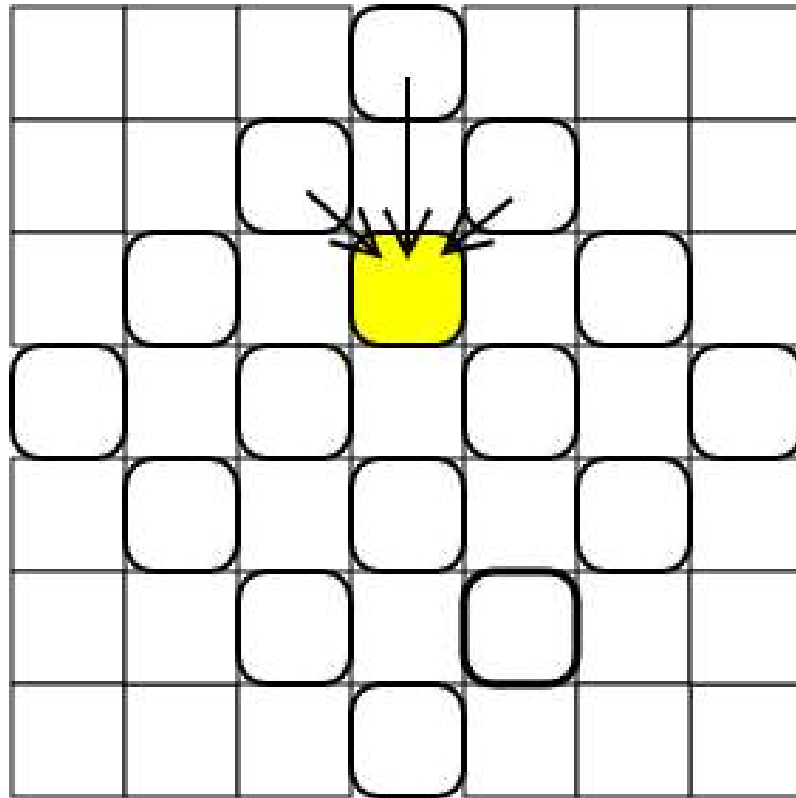
Un cop iniciada l'execució, totes les cel·les poden continuar calculant sense conflictes. Les cel·les que fan càlculs no útils no afecten a la resta:



# AOA: Arquitectures Sistèmiques

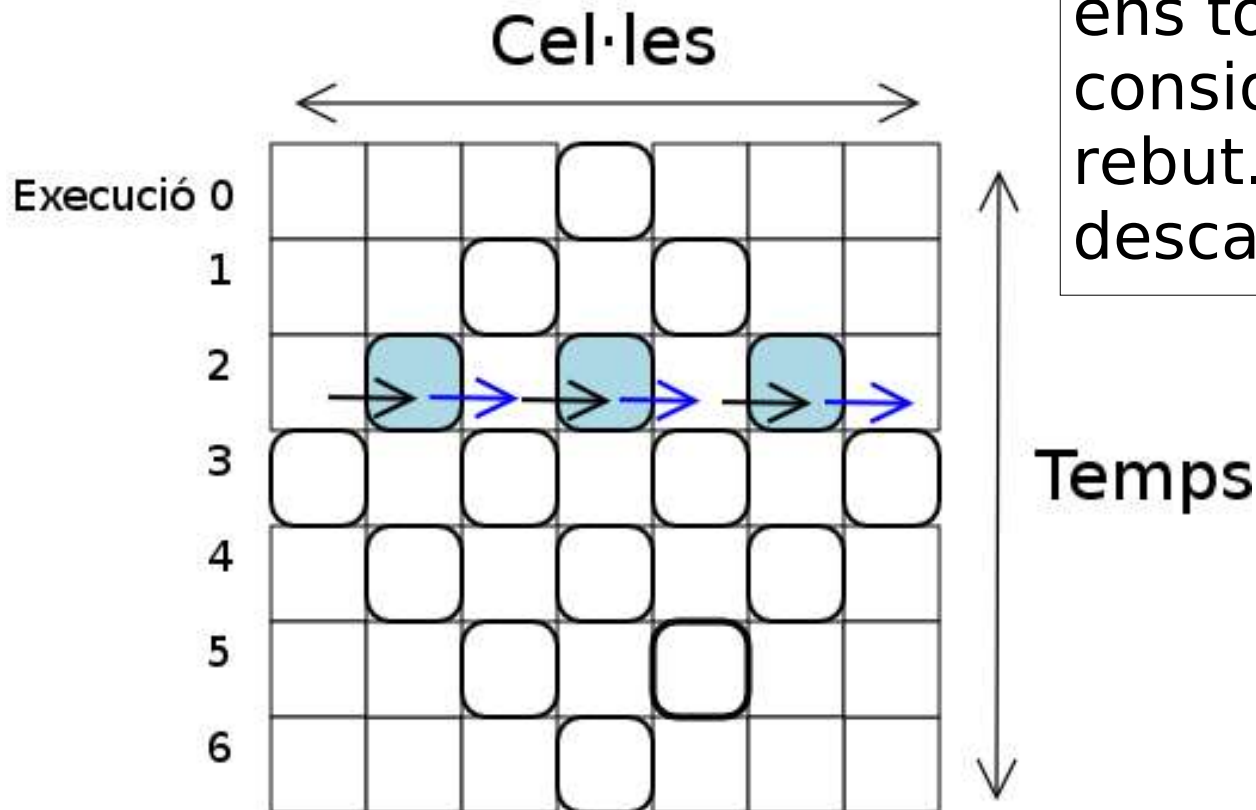
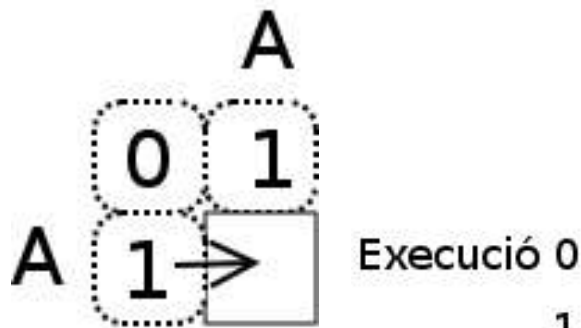
---

Per cobrir les nostres dependències s'han de realitzar comunicacions i guardar historials abans de la següent execució.



# AOA: Arquitectures Sistèmiques

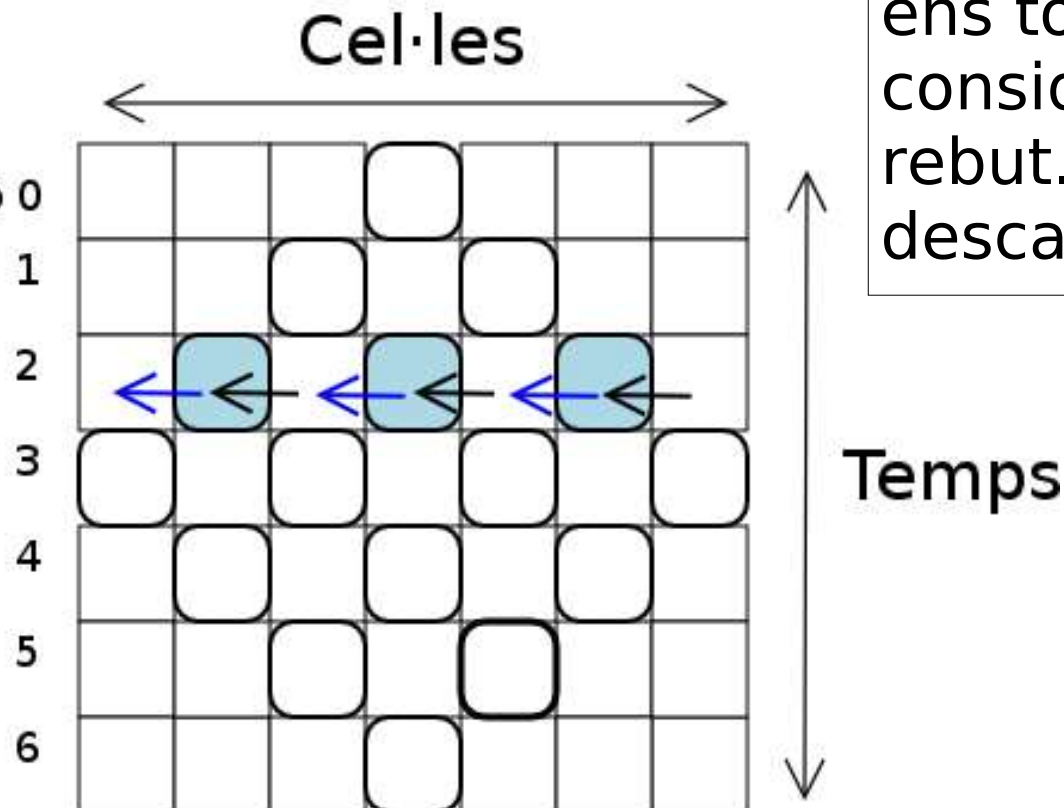
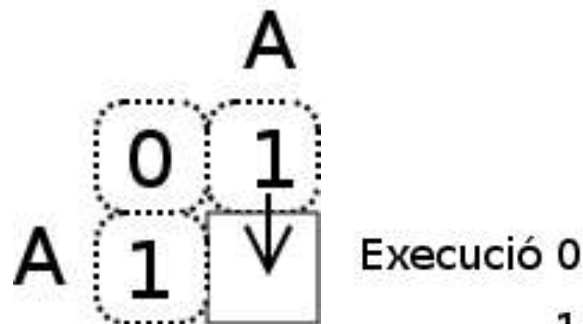
Dependència d'esborrat: Comuniquem la dada a la veïna de la dreta/Rebem dada per l'esquerra.



Cel·la X: Si a la propera execució ens toca calcular, considerem valor rebut. Sinó el descartem.

# AOA: Arquitectures Sistèmiques

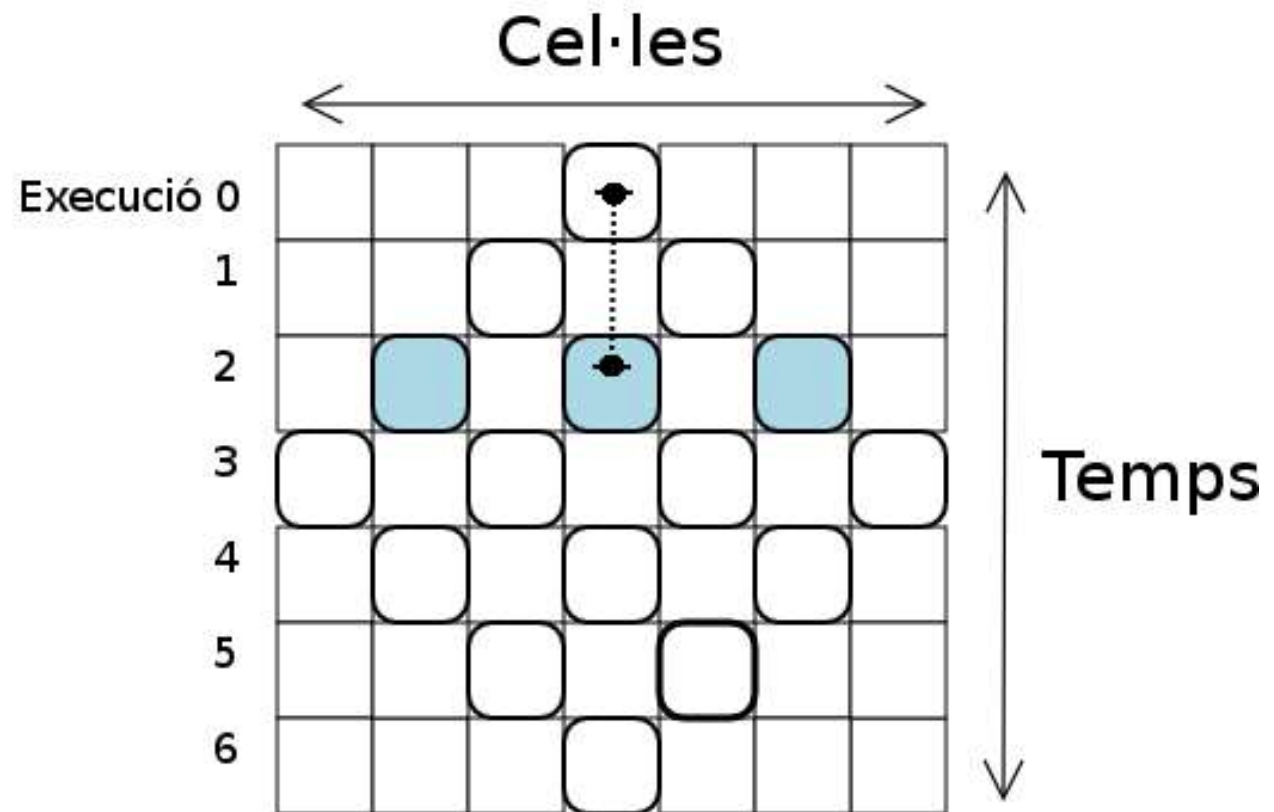
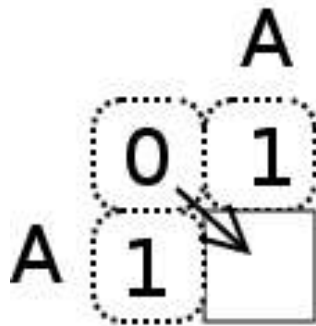
Dependència d'inserció: Comunicuem la dada a la veïna de l'esquerra/Rebem dada per la dreta.



Cel·la X: Si a la propera execució ens toca calcular, considerem valor rebut. Sinó el descartem.

# AOA: Arquitectures Sistèmiques

Dependència de substitució: Guardem l'últim valor que vam calcular (calcula de forma alterna).



# AOA: Architectures Systèmiques

---

Cada cel·la conté:

- Valor rebut per la dreta/esquerra temporal
- Valor rebut per la dreta/esquerra útil
- Valor anterior (últim càlcul)
- Valor actual
- Torn
- Inici (timeout)
- Identificador de cel·la
- Paraules senceres: Per calcular el cost de substitució es pot accedir a les lletres concretes a partir de les dades que tenim i d'un parell de regles

# AOA: Arquitectures Sistèmiques

## Inicialització:

- Valor rebut per la dreta temporal = útil  
0,0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10
- Valor rebut per la esquerra temporal = útil  
10,9,8,7,6,5,4,3,2,1,0,0,0,0,0,0,0,0,0,0,0
- Valor anterior (últim càlcul) = 0
- Valor actual = 0
- Torn = 0,1,0,1,0,1,0,1,0,1...
- Inici (timeout) = 9,8,7,6,5,4,3,2,1,0,1,2,3,4,5,6,7,8,9
- Identificador de cel·la = 0,1,2,3,4,5,6,7,8,9,...

		A	B	C	D
	0	1	2	3	4
A	1				
B	2				
C	3				
D	4				



# AOA: Arquitectures Sistèmiques

## Accés a les lletres:

- Rangs 0..N per:  
paraula1, paraula2  
- Valor inicial 0 per:  
identificador, càlculs útils

- Per poder calcular el cost de substitució s'ha d'accedir a les lletres de la paraula concretes segons en quin moment del temps i en quina cel·la estem. Amb dos conjunts de regles es pot solucionar el problema:
- Si la cel·la esta situada de la meitat cap a l'esquerra:
  - $\text{paraula1}[(\text{Tamany\_maxim\_paraula} - \text{identificador cel·la}) + \text{nombre\_càlculs\_utils\_d'aquesta\_cel·la} - 1]$
  - $\text{paraula2}[\text{nombre\_càlculs\_utils\_d'aquesta\_cel·la}]$
- Si la cel·la esta situada de la meitat cap a la dreta:
  - $\text{paraula1}[\text{nombre\_càlculs\_utils\_d'aquesta\_cel·la}]$
  - $\text{paraula2}[(\text{identificador cel·la} \% \text{tamany\_maxim\_paraula}) + \text{nombre\_càlculs\_utils\_d'aquesta\_cel·la}]$

# **AOA: Architectures Systèmiques**

---

Implementació...

# AOA: Arquitectures Sistòliques

---

Test I utilitzant simulador:

- Cercar les 5 paraules més properes a 'hola' a un fitxer que conté 11 paraules: *“roma hola cola munga chungu coca tarragona colegi paraula resta veure”*

- Temps no sistòlic (versió 2)

0.043, 0.039, 0.047, 0.038, 0.039

Promig 0.041

- Temps sistòlic

0.194, 0.559, 0.646, 0.224, 0.243

Promig 0.373

# AOA: Arquitectures Sistòliques

---

Test II utilitzant simulador:

- Cercar les 40 paraules més properes a 'parla' a un fitxer que conté 23740 paraules.

- Temps no sistòlic (versió 2)

1.731 segons

- Temps sistòlic

5 min. 1.388 segons