

# PRÀCTICA 3

## CREACIÓ D'UN AUTÒMATA CEL·LULAR

ADRIÀ MOYA MORERA  
JAVIER PUERTA DEL VALLE

UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Facultat d'Informàtica de Barcelona

**FIB**



# ÍNDIX

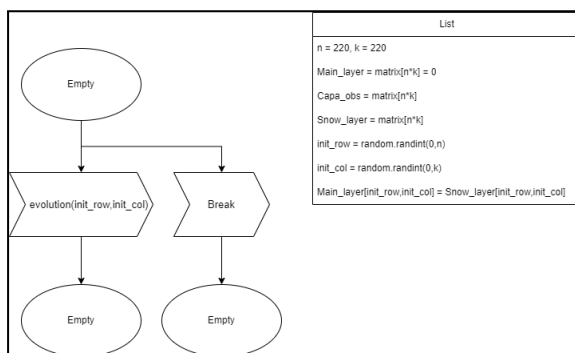
<b>1. Hipòtesis emprades en el model.....</b>	<b>1</b>
<b>2. Diagrama de la funció d'evolució de l'autòmata.....</b>	<b>1</b>
<b>3. Descripció resumida de la implementació realitzada.....</b>	<b>3</b>
3.1. La classe OP_Problema.....	3
<b>4. Descripció del funcionament de l'autòmat.....</b>	<b>4</b>
<b>4.1 Possibilitat d'experimentació amb l'autòmat.....</b>	<b>4</b>
<b>5. Optimització del sistema a partir del model proposat.....</b>	<b>6</b>

## 1. Hipòtesis emprades en el model

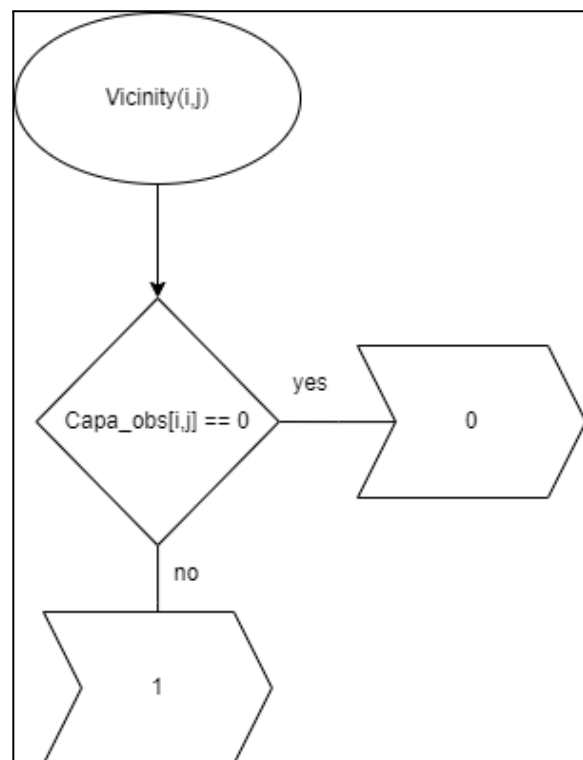
- L'espai bidimensional on es troba l'autòmata no té pendent. (**sistemàtica**)
- L'allau de neu comença a caure a partir d'un punt de l'espai determinat, no hi haurà diverses allaus, només 1 amb la seva evolució. (**estructural**)
- Quan l'allau arriba a un punt frontera, aquest punt frontera queda determinat com un espai "blanc". (**estructural**)
- Hi haurà més quantitat de neu en la part superior. (**sistemàtica**)
- Els obstacles tenen "vida infinita". (**simplificadora**)
- Si es genera una allau en un obstacle, l'allau evolucionarà. (**sistemàtica**)

## 2. Diagrama de la funció d'evolució de l'autòmata

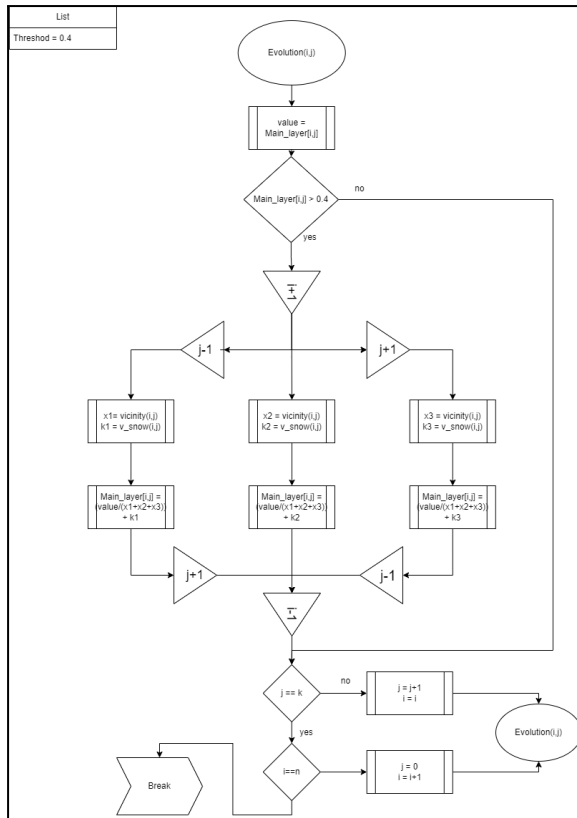
El diagrama en SDL és el següent:



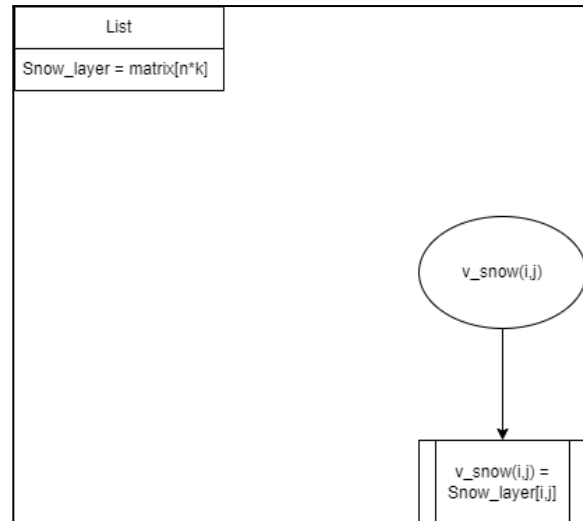
2.A) Diagrama d'Empty\_State



2.B) Diagrama de Vicinity



2.C) Diagrama d'evolució



2.D) Diagrama d'Snow\_layer

Nosaltres hem realitzat dues implementacions d'aquest escenari, però ens centrarem sobretot en la segona:

-En la primera només afegim una capa secundària estàtica d'obstacles que poden parar l'allau sense tenir en compte la velocitat ni quantitat de neu de l'allau. En el codi es pot cridar a través del mètode .evolve()

-La segona implementació és una extensió de la 1a, en aquesta afegim una capa secundària de neu estàtica formada per valors "float" que indiquen la quantitat de neu per cada cel·la. Implementació realitzada a través del mètode .evolve\_prop(nrow,ncol), on [init\_row,init\_col] especifiquen la casella on comença l'allau.

### 3. Descripció resumida de la implementació realitzada

La implementació d'aquest autòmat consta de 2 classes principals:

- La Classe OP\_Problema:
  - Es la classe que s'encarrega d'articular tot l'autòmata. Té totes les capes afegides al sistema i la funció de veïnatge i evolució.
- La Classe Capa
  - S'encarrega de crear o inicialitzar les capes a afegir a l'autòmat.

#### 3.1. La classe OP\_Problema

Si observem l'init, trobem:

- El número de files i columnes de la capa principal.
- La seed del generador de números aleatoris.
- La Capa principal
- La Capa d'obstacles
- La Capa de neu.

Per tal de generar la capa de neu, cal crear primer la seva matriu per tal de donar-la com a argument a la classe Capa i, per tant, generar la nova capa.

- Utilitzem el mètode GEN\_SNOW\_LAYER
  - En aquest mètode trobem la generació de la quantitat de neu que hi ha a la capa principal.
  - Hem fixat el criteri de que sempre hi haurà més neu en la zona superior de la matriu.
  - Per tal de fer això es fa un doble bucle per tal de donar un valor a cada cel·la, on hi ha un algorisme que indica el seu valor depenent de la seva latitud.

Per tal de trobar el veïnatge per ambdues capes, trobem els dos mètodes:

- VICINITY => Troba els veïns de la capa principal. Si troba un punt on hi ha un obstacle, es desactiva la possibilitat d'expansió en aquella cel·la.
- GET\_SNOW\_NEIGHBOURS => Troba els veïns de la capa de neu. Està creada pel fet de que l'autòmat arribi a un límit lateral i sàpigui que fer amb els seus veïns.

Per tal de fer evolucionar l'allau, trobem 2 funcions d'evolució:

- EVOLVE => Funció d'evolució sencilla on té en compte si en el seu veïnatge es troba algun obstacle.

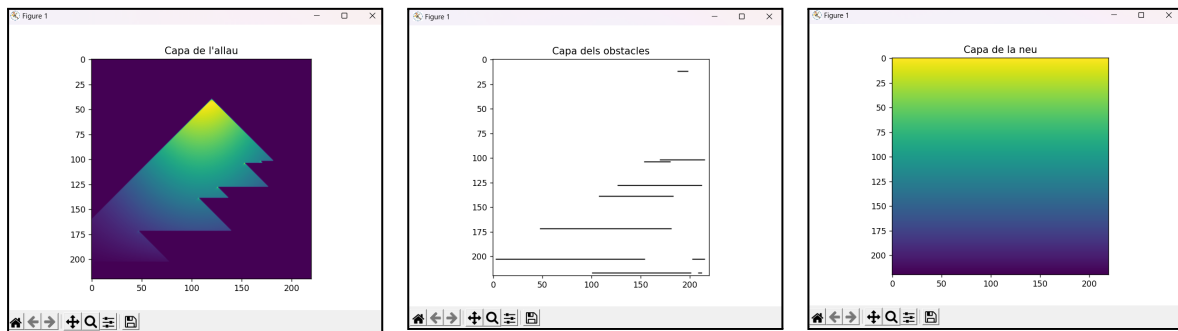
- EVOLVE\_PROP => És la mateixa funció d'evolució però aquí també té en compte la quantitat de neu que porta l'allau. Si la quantitat de neu arriba a un *threshold* determinat, es deixarà de propagar-se.

## 4. Descripció del funcionament de l'autòmat

Per tal de posar en marxa l'autòmat cal escriure desde la consola. Aquest autòmat té la seed posada en 2224:

- `python main.py`

A continuació, sortirà una finestra gràfica on es veurà l'evolució de l'allau de neu. Si tanquem aquesta finestra, sortirà una altre amb la capa de "obstacles". Si tanquem la finestra, sortirà la imatge de la capa de quantitat de neu.



4.A) Capa de l'allau

4.B) Capa d'obstacles

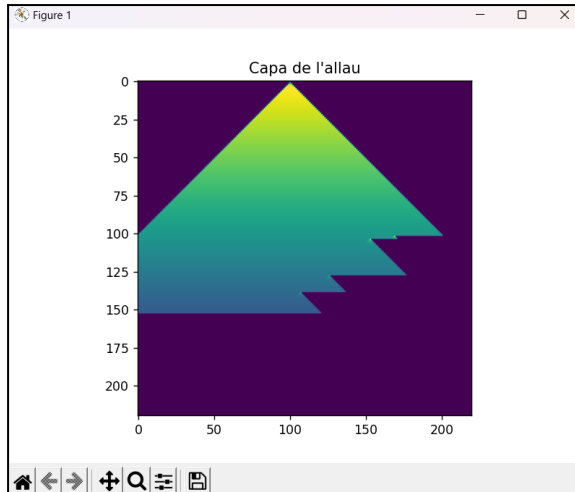
4.C) Capa de neu

### 4.1 Possibilitat d'experimentació amb l'autòmat

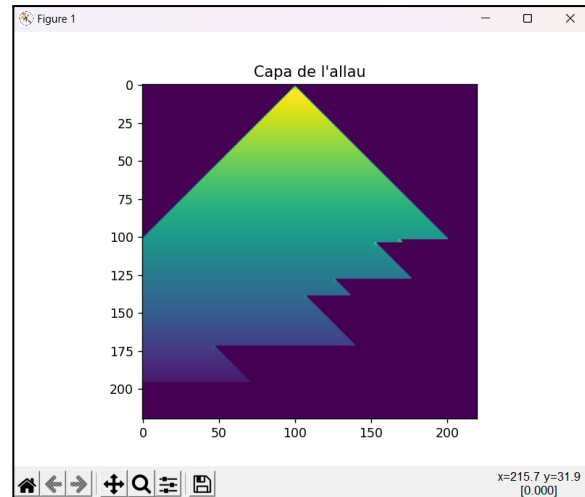
Gràcies a la creació de la capa de quantitat de neu, podem experimentar amb el model, variant la quantitat de neu que hi ha a la muntanya, veient com evoluciona el model segons la posició d'inici de l'allau i canviant el valor pel qual considerem que no hi ha més neu per a que l'allau es segueixi expandint. Aquest últim valor el podem veure a la línia 63, considerem que és el *threshold* pel qual l'allau ja no té suficient neu com per evolucionar.

A continuació podem veure diferents casos canviant aquest valor:

Iniciem l'experiment a la posició [1,100] i en el cas de l'esquerra iniciem l'experiment amb un *threshold* de 0.4 i en el de la dreta el baixem a 0.1:



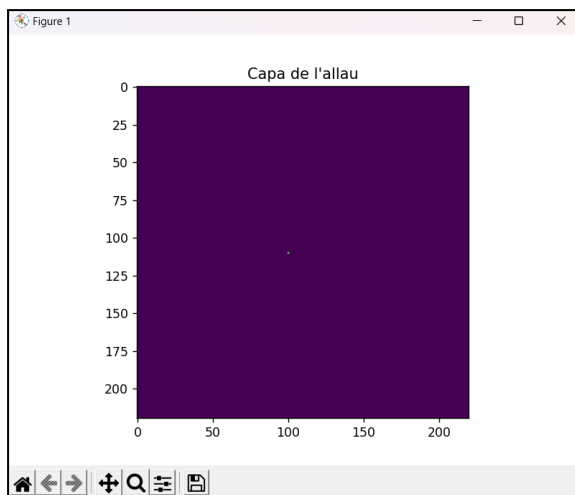
4.1.A) Experimentació amb threshold de 0.4



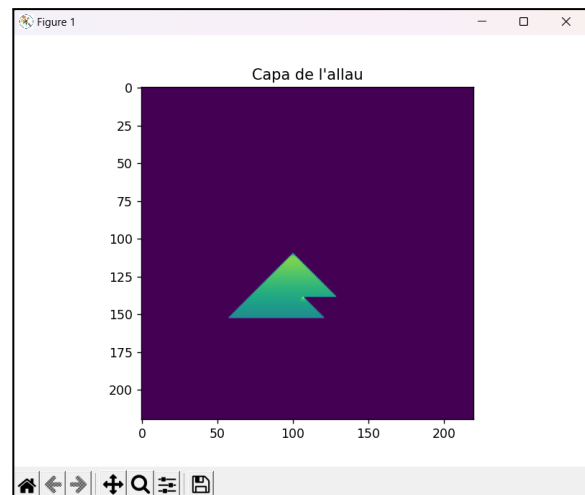
4.1.B) Experimentació amb threshold de 0.1

Podem veure clarament com al baixar el threshold l'allau evoluciona unes capes més fins a quedar-se sense neu, per tant, aquest threshold té un paper fonamental en l'evolució de l'allau, veiem un altre experiment.

En aquest cas iniciarem en [110,100] i en primer lloc l'iniciarem amb un threshold de 0.5 i posteriorment, en el cas de la dreta, hem iniciat l'experiment amb un threshold de 0.4.



4.1.C) Experimentació amb threshold de 0.5



4.1.D) Experimentació amb threshold de 0.4

Aquí podem observar com la precisió d'aquest threshold és molt acurada, en canviar-lo només un 0.1, podem veure un gran canvi de paradigma. De la mateixa forma, veiem la importància del lloc de creació de l'allau, ja que al crear-se molt més a baix de la muntanya, s'ha propagat de manera significativament inferior i amb menys neu.

Aquest fet ens destaca la rellevància de la perillositat dels allaus que es situen a posicions de la muntanya amb menor quantitat de neu, en comparació amb aquells que comencen en llocs amb una densitat inferior de neu. Tot i així s'ha de tenir en compte que com no estem modelant l'allau tenint en compte la densitat de neu, pendent la muntanya, etc, es tracta només d'un entorn de simulació bàsica.

En aquest cas només hem usat una llavor per generar els experiments, però també podem jugar amb on posar els obstacles, canviar la quantitat de neu en la capa corresponent o canviant la forma de propagació de la neu, amb el que es poden extreure conclusions més significatives.

## 5. Optimització del sistema a partir del model proposat

Per tal d'optimitzar el nostre model, seria bastant eficaç afegir algunes capes secundàries més que tinguin efecte sobre la funció d'evolució. Nosaltres hem pensat en aquestes propostes:

- Capa de Força : Aquesta capa tindria com a objectiu obtenir la força que té la allau quan està baixant. D'aquesta forma si xoca amb algun objecte, podem veure si aquest es trencaria o no. A més a més, també ens serviria com a mètode per parar la propagació, ja que quan arribi a un umbral límit de força, podem dir que ja no té la suficient força per ser un allau i per tant, parar la propagació.
- Capa de Relleu : Aquesta capa és la que indica quina pendent té el relleu. D'aquesta forma, es pot augmentar la força de la allau quan hi ha una pendent alta o reduir la força quan aquesta és petita o propera a 0. A més a més, és la capa que més s'apropa a un terreny real.
- Capa de Qualitat de neu : Com es evident, no és el mateix tenir una neu tova o una neu dura i compacta quan existeix un allau. És per això que aquesta capa seria una de les més simples però a la vegada la que donaria més informació, ja que seria capaç de veure els efectes que té una allau quan es troba amb diferents tipus de neu. Cada cel·la d'aquesta capa tindria un valor entre 0 y 1, on 1 és neu forta i 0 neu molt tova, que indicaria la qualitat d'aquesta.
- Capa de neu soterrània : Aquesta capa indica si sota la capa general de neu, trobem gel o si trobem neu tova. Això pot provocar un augment en la força de l'allau o una disminució de la seva propagació. Per això l'estructura de la capa seria una matriu amb les mateixes dimensions que la capa principal, on els valors de cada cel·la tindran valors entre 0 i 1, sent 1 gel i 0 neu molt tova. No seria la mateixa capa que la de Qualitat de neu degut a que aquesta es centraria en la capa que hi ha sota la "superfície" de la neu.