

Programmation Fonctionnelle

Projet

Michele Pagani

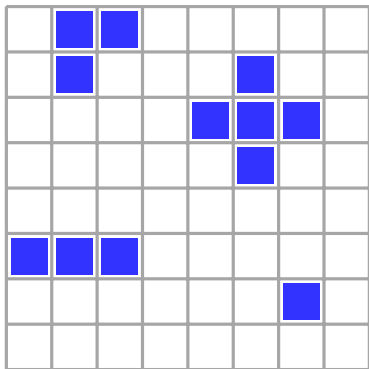


Université Paris Diderot
UFR Informatique
Laboratoire Preuves, Programmes et Systèmes

`michele.pagani@pps.univ-paris-diderot.fr`

19 octobre 2015

Automates cellulaires bidimensionnels

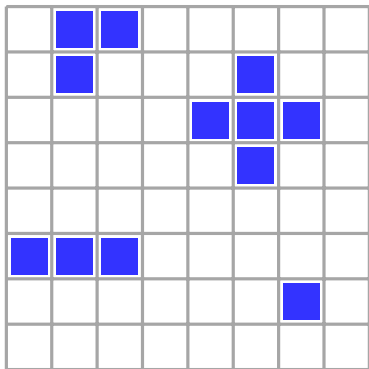


- le **plateau** du jeu est une grille
- une **cellule** est une case de la grille
 - deux états: **vivante** ou **morte**
- une **génération** est un ensemble finis de cellules vivantes
- un **automate** est un ensemble des règles faisant évoluer les générations
 - une règle dit si une cellule mort ou vit en fonction de son **voisinage**

Exemple (Jeu de la vie)

- une cellule morte possédant exactement trois voisines vivantes devinet vivante (elle naît);
- une cellule vivante possédant deux ou trois voisines vivantes le reste, sinon elle morts.

Automates cellulaires bidimensionnels

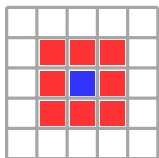


- le **plateau** du jeu est une grille
- une **cellule** est une case de la grille
 - deux états: **vivante** ou **morte**
- une **génération** est un ensemble finis de cellules vivantes
- un **automate** est un ensemble des règles faisant évoluer les générations
 - une règle dit si une cellule mort ou vit en fonction de son **voisinage**

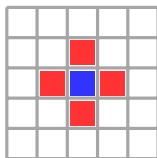
Exemple (Jeu de la vie)

- une cellule morte possédant exactement trois voisines vivantes devient vivante (elle naît);
- une cellule vivante possédant deux ou trois voisines vivantes le reste, sinon elle meurt.

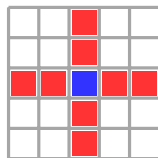
Voisinage



voisinage de Moore
(jeu de la vie)



voisinage de Von Neumann
rayon 1



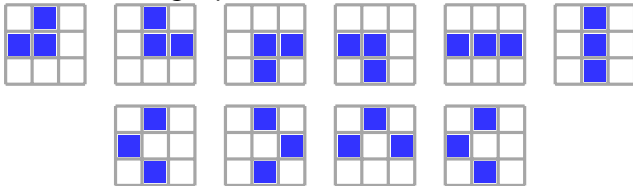
voisinage de Von Neumann
rayon 2

Il y a plusieurs types différent de voisinage:

- le projet minimal: **voisinage de Von Neumann de rayon 1**
- états possibles des voisinages : $2^5 = 32$
- un automate est une partition de ces états entre
 - ceux qui rend la cellule au milieu vivante;
 - ceux qui rend la cellule au milieu morte.

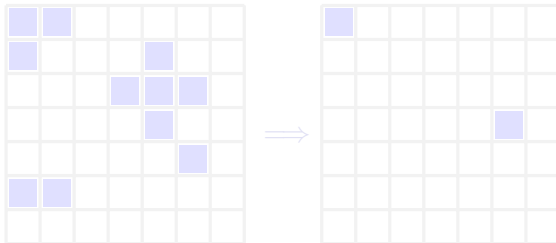
Un exemple d'automate

- les états du voisinage qui rendent la cellule au milieu vivante:



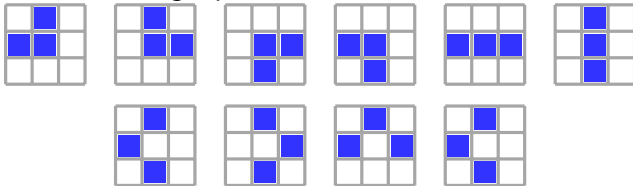
- tout autre état du voisinage rend la cellule au milieu morte.

Cet automate donne l'évolution suivante:



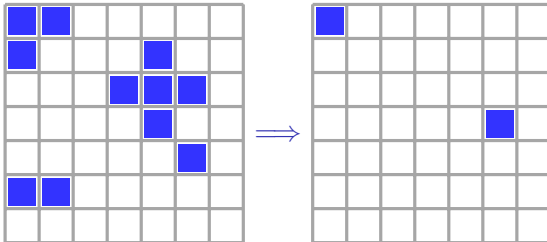
Un exemple d'automate

- les états du voisinage qui rendent la cellule au milieu vivante:



- tout autre état du voisinage rend la cellule au milieu morte.

Cet automate donne l'évolution suivante:



Intérêt automates cellulaires

- exemples de système dynamique discret
 - évolutions chaotiques, dépeuplement, réversibilité, ...
 - générations périodiques, **stables**, attractives, ...
- modèle de calcul parallèle (Turing-complet)
 - les règles sont appliquées **simultanément** à toutes les cellules de la grille
 - **Attention à votre implémentation !**

Projet minimal

Il est **obligatoire** de :

- ① mettre en oeuvre un simulateur des automates cellulaires à la Von Neumann de rayon 1
- ② mettre en oeuvre une fonction qui trouve toute génération stable d'un automate
 - 1 étant donné une grille trouver ensemble des variables telle que:
 $\text{assignation} \Leftrightarrow \text{génération}$
 - 2 traduire automate dans formule propositionnelle P telle que:
 $\text{assignation satisfaisable} \Leftrightarrow \text{génération stable}$
 - 3 donner P à minisat et lire la solution

Definition

Une génération stable d'un automate est une génération qui évolue dans elle même en une étape.

Génération stable

Auch, un problème !

Trouver les générations stables est un problème très difficile:

- état chaque cellule interfère avec état cellules voisines
- problème NP-complet (pour grilles de dimension quelconque)
 - possible vérifier une solution “efficacement” (temps polynomial)
 - mais on ne sait pas trouver une solution “efficacement”

La solution !

Traduire notre problème dans un autre problème équivalent mais avec résolveurs (solvers en anglais) déjà implémentés avec de bonnes heuristiques

SAT : étant donnée une formule propositionnelle, trouver une assignation des variables rendant la formule vrai.

Plein de SAT-solvers, nous utiliserons:

<http://minisat.se>

Génération stable

Auch, un problème !

Trouver les générations stables est un problème très difficile:

- état chaque cellule interfère avec état cellules voisines
- problème **NP-complet** (pour grilles de dimension quelconque)
 - **possible vérifier** une solution “efficacement” (temps polynomial)
 - mais on ne sait **pas trouver** une solution “efficacement”

La solution !

Traduire notre problème dans un autre problème équivalent mais avec **résolveurs** (**solvers** en anglais) déjà implémentés avec de bonnes heuristiques

SAT : étant donnée une formule propositionnelle, trouver une assignation des variables rendant la formule vrai.

Plein de SAT-solvers, nous utiliserons:

<http://minisat.se>

Génération stable

Auch, un problème !

Trouver les générations stables est un problème très difficile:

- état chaque cellule interfère avec état cellules voisines
- problème **NP-complet** (pour grilles de dimension quelconque)
 - **possible vérifier** une solution “efficacement” (temps polynomial)
 - mais on ne sait **pas trouver** une solution “efficacement”

La solution !

Traduire notre problème dans un autre problème équivalent mais avec **résolveurs** (**solvers** en anglais) déjà implémentés avec de bonnes heuristiques

SAT : étant donnée une formule propositionnelle, trouver une assignation des variables rendant la formule vrai.

Plein de SAT-solvers, nous utiliserons:

<http://minisat.se>

Génération stable

Auch, un problème !

Trouver les générations stables est un problème très difficile:

- état chaque cellule interfère avec état cellules voisines
- problème **NP-complet** (pour grilles de dimension quelconque)
 - **possible vérifier** une solution “efficacement” (temps polynomial)
 - mais on ne sait **pas trouver** une solution “efficacement”

La solution !

Traduire notre problème dans un autre problème équivalent mais avec **résolveurs** (**solvers** en anglais) déjà implémentés avec de bonnes heuristiques

SAT : étant donnée une formule propositionnelle, trouver une assignation des variables rendant la formule vrai.

Plein de SAT-solvers, nous utiliserons:

<http://minisat.se>

Projet minimal

Il est **obligatoire** de :

- ① mettre en oeuvre un simulateur des automates cellulaires à la Von Neumann de rayon 1
- ② mettre en oeuvre une fonction qui trouve toute génération stable d'un automate
 - 1 étant donné une grille trouver ensemble des variables telle que:
 $\text{assignation} \Leftrightarrow \text{génération}$
 - 2 traduire automate dans formule propositionnelle P telle que:
 $\text{assignation satisfaisable} \Leftrightarrow \text{génération stable}$
 - 3 donner P à minisat et lire la solution

Definition

Une génération stable d'un automate est une génération qui évolue dans elle même en une étape.

Projet minimal

Il est **obligatoire** de :

- ① mettre en oeuvre un simulateur des automates cellulaires à la Von Neumann de rayon 1
- ② mettre en oeuvre une fonction qui trouve toute génération stable d'un automate
 - 1 étant donné une grille trouver ensemble des variables telle que:
$$\text{assignation} \Leftrightarrow \text{génération}$$
 - 2 traduire automate dans formule propositionnelle P telle que:
$$\text{assignation satisfaisable} \Leftrightarrow \text{génération stable}$$
 - 3 donner P à minisat et lire la solution

Definition

Une génération stable d'un automate est une génération qui évolue dans elle même en une étape.

Attention !

- plagiats = 0
(en plus: on se rappelle bien de vos noms. . .)
- à rendre sur Didel **au plus tard le**
mercredi 30 décembre à 23h59
(pensez bien à rendre une première version en avance. . .)
- à faire en **binôme** = 2, deux, two, due, 兩 . . .