Université de Strasbourg
Modèles de Calcul Feuille de TP

Automates cellulaires : le jeu de la vie

Malgré des règles très simples, le jeu de la vie permet le développement de motifs extrêmement complexes. Le jeu se déroule sur une grille à deux dimensions, théoriquement infinie (mais de longueur et de largeur finies et plus ou moins grandes dans la pratique), dont les cases — qu'on appelle des « cellules », par analogie avec les cellules vivantes — peuvent prendre deux états distincts : « vivantes » ou « mortes ». Chaque cellule vit, meurt, ou bien génère une nouvelle cellule (division cellulaire) en fonction du nombre de cellules qui l'environnent. À chaque étape, l'évolution d'une cellule est entièrement déterminée par l'état de ses huit voisines de la façon suivante :

- 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.

On représentera les cellules vivantes par 1 et les cellules mortes par 0. On représentera une grille par une liste de listes de 0 ou de 1.

1. Programmer une fonction `acces` qui prend en argument une grille et les coordonnées d'une cellule et renvoie la valeur de cette cellule ou zero si la cellule n'est pas dans la grille.
2. Programmer une fonction `nb_voisins` qui prend en argument une grille et les coordonnées d'une cellule et qui renvoie le nombre de voisins de cette cellule.
3. Programmer une fonction `afficher_jeu` qui affiche l'état du jeu à l'écran. On représentera les cellules vivante par `o` et les cellules mortes par `#`.

   Exemple :

   ```
   ##########
   #oooo#####
   #o##o#####
   #o##o#####
   #o##o#####
   #o##o#####
   #o##o#####
   #o##o#####
   #o##o#####
   #oooo#####
   ##########
   ```

4. Programmer une fonction `naissances_deces` qui prend en argument une grille et qui modifie cette grille afin de traiter les naissances et les décès.

   Aide : Attention les naissances et les décés doivent être calculés simultanement sur toute la grille prise en argument. On utilisera donc une `copie` de la grille.
5. Programmer une fonction `des_etapes` qui prend en argument une grille initiale et un entier n et qui exécute n étapes du jeu de la vie en affichant les différentes étapes. On fera en sorte que la fonction `des_etapes` ne modifie pas la grille initiale.
6. On souhaite pouvoir trouver des grilles initiales particulières :
   (a) Ecrire une fonction qui teste si une grille est stable (en anglais *still-life*). C'est à dire si les cellules ont stoppé toute évolution : elles sont dans un état stationnaire et n'évoluent plus.

(b) Ecrire une fonction qui donne la période des grilles oscillantes, c'est à dire si il existe une entier n telle que les cellules reviennent à leur configuration de départ après n étapes la fonction renvoie le plus petit n qui permet de revenir à la configuration de départ.

7. On souhaite pouvoir évaluer d'autres automates cellulaires que le jeu de la vie.

   Par exemple, vous pouvez généraliser vos fonctions afin que l'on puisse prendre en argument les règles sous la forme de deux tableaux (un pour les naissances et un pour les décès) qui indiquent pour chacun des nombres de voisins vivants possibles (de 0 à 8) si la cellule naît ou décède. Par exemple, les règles du jeu de la vie sont `[0,0,0,1,0,0,0,0,0]` et `[0,0,1,1,0,0,0,0,0]`.

8. Pour aller plus loin, on peut généraliser à d'autres grilles (toriques, en 1D, ...), on peut réaliser un affichage graphique, ...