

Automates cellulaires: feu de forêt

1 Description de l'automate

Nous allons programmer un automate cellulaire qui simule la propagation d'un feu de forêt. La forêt sera représentée par une grille avec dans les cases rien, des arbres, du feu ou des cendres. La propagation du feu de forêt se fera de la manière suivante:

- Une case avec rien reste comme elle est.
- Une case avec un arbre devient en feu si une case adjacente est en feu.
- Une case en feu devient une case cendre.
- Une case cendre devient une case vide.

On construira la forêt initiale aléatoirement. Avec, pour chaque case, une probabilité p d'être un arbre et une probabilité $1 - p$ d'être vide.

Le but de ce TD est de programmer cet automate et d'avoir une animation affichant la propagation du feu de forêt.

2 Programmation de l'automate

Les étapes proposées dans la suite sont données à titre indicatif. Vous n'êtes pas obligé de les faire dans l'ordre ni même de les suivre si vous avez de meilleures idées.

2.1 Moteur du jeu

- **Stockage des données:** Comment va-t-on stocker l'état de la forêt?
- **Trame principale:** Écrire le programme principal `feuForet` prenant en argument p la probabilité d'être une case arbre initialement et affichant l'animation du feu de forêt. Ce programme devra juste décrire la trame et fera appel à d'autres fonctions que nous écrirons ensuite.

2.2 Forêt initiale

- **Forêt initiale:** Écrire un programme `foretIni` prenant en argument p et créant la forêt initiale.
- **Mettre le feu:** Écrire un programme `FeuIni` mettant le feu à une case.

2.3 Propagation du feu

- Écrire un programme `propaFeu` prenant en argument `feu` l'état de la forêt et renvoyant l'état de la forêt à l'étape suivante.

2.4 Affichage et animation

- **Afficher une cellule:** Écrire une procédure `dessCel` prenant en argument la position de la cellule l , c , la taille de la forêt n (on pourra supposer que la forêt est carré) et la couleur de la cellule et renvoyant cette cellule.

On rappelle que pour dessiner un polygone, on utilise `polygon` (il faut avoir chargé les bibliothèques `plots`, `geometry` et `plottools`).

- **Dessiner un forêt:** Écrire une procédure `dessForet` prenant en argument l'état de la forêt et renvoyant l'image représentant la forêt.

On rappelle que pour mettre plusieurs éléments sur une même image, on les stocks dans un ensemble puis on utilise `display` (on pourra utiliser les options suivantes pour avoir un meilleur rendu `scaling=constrained, axes=None`).

- **Animer le feu:** Écrire une procédure `anime` prenant en argument la liste des états de la forêts et renvoyant l'animation du feu de forêt.

3 Modifier l'automate

Voici quelques idées pour modifier votre automate.

- Rendre la forêt "périodique". C'est à dire que les cases de droites sont adjacentes à celles de gauches et que celle d'en haut sont adjacentes à celle du bas.
- Faire en sorte que le feu puisse se propager non seulement sur les cases adjacentes mais aussi sur celles aux angles.
- Modifier les règles de l'automate pour simuler le jeu de la vie.

Le jeu de la vie est un automate cellulaire avec deux états: mort ou vivant. Voici les règles de son évolution:

- 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.