

Projet d'automate cellulaire en Ocaml

Document de présentation des choix d'implémentation

BEN AMMAR Ramzi

STASYSZYN Romain

Année 2015/2016

Parties traitées :

Les trois premières parties sont totalement traitées, la quatrième l'est aussi mais la formule obtenue n'est pas totalement correcte donc n'est qu'affichée et écrite dans un fichier au format Dimacs à titre d'exemple. Ce même fichier au format Dimacs contient lui aussi des erreurs telle la première ligne n'étant pas p cnf x x et une ligne sur deux contient seulement un 0.

Choix des types :

Pour ce projet de programmation fonctionnel dans lequel nous devons implémenter un automate cellulaire, nous avons choisis de gérer le voisinage de niveau 1 (voisinage de Moore) pour chaque cellules de la grille. Ainsi nous avons mis en place plusieurs types indispensables au déroulement de notre programme.

Le premier est le type state prenant les valeurs Dead ou Alive décrivant respectivement l'état de mort ou l'état de vie de la cellule considérée.

Le deuxième type est rule, une state list permettant dans notre cas d'avoir des ensembles cohérents de cinq éléments chacun décrivant l'état de la cellule Nord,Est,Sud,Ouest et celui de la cellule courante.

Le troisième type est automaton, une rule list permettant de décrire l'ensemble des règles rendant la cellule courante vivante à la prochaine génération.

Le quatrième type est generation, une matrice d'état représentant une grille à un instant donné.

Il y a un type state_N représentant l'état de la cellule courante via une valeur négative si morte et positive si vivante. La valeur numérique est liée à la position de la cellule au sein de la grille.

Le dernier type est formule permettant d'utiliser des littéraux de type state_N, la négation, la conjonction ainsi que la disjonction.

Choix du découpages en modules :

Le module `automate.ml` contient les fonctions gérant les opérations sur les automates, grilles, générations.

Le module `affichage.ml` contient les fonctions d'affichages de la grille.

Le module `programme.ml` contient les fonctions permettant la mise en place d'une interactivité avec l'utilisateur à travers un système de menu proposant à l'utilisateur de saisir l'entier correspondant à l'action qu'il souhaite réaliser (afficher la génération suivante, sortir du programme etc).

Il y a une interface `automate.mli` afin d'explicitier le contenu du module correspondant et de présenter les types choisis.

Description des fonctions :

Commençons par les fonctions du module `affichage.ml`, il y a la sous fonction `show_state` permettant affichant le caractère associé à l'état de la cellule et la fonction `show_generation` affichant la grille en paramètre, en parcourant linéairement la matrice carrée qui nous sert de grille.

Pour ce qui est des fonctions du module `automate.ml`, nous avons la sous fonction `egalite` renvoyant `true` si une des règles de l'automate peut être associé à une configuration (cellule courante/voisins) sinon renvoie `false`. La fonction `next_generation` quant à elle parcourt toute la matrice carrée faisant office de grille pour appliquer à chaque cellule la fonction `egalite` afin de savoir si elle sera vivante ou morte à la prochaine génération. La fonction renvoie la grille associée à la génération suivante. Pour ce qui est des cases « au bord » de la grille l'infinité (grille torique) de la grille est prise en compte.

Pour le module `programme.ml`, la fonction `main` permet de lancer une boucle d'interprétation des choix de l'utilisateur. On lui affiche les options possibles (afficher la génération courante, calculer et afficher la génération suivante ou bien sortir du programme).

La fonction `nbLignes` calcule le nombre de ligne du fichier d'initialisation passé en paramètre et la fonction `lire` initialise notre grille représentant notre génération initiale.

Enfin la fonction parse permet d'initialiser correctement nos données pour notre automate cellulaire à partir des informations données par le fichier. La fonction stables permet d'obtenir une formule non valide qui se rapproche de celle souhaitée et est présente à titre informatif sur l'avancée de notre travail.