Backtracking et graphes

Toutes vos questions et remarques sont bienvenues ! Mon adresse est `antonin.delpeuch@ens.fr`. Vous pouvez retrouver les sujets et leurs corrigés sur ma page personnelle :
`http://antonin.delpeuch.eu/maple`.

1 Résolution de Sudoku

Une grille de Sudoku est représentée par une matrice 9×9, contenant 0 dans les cases vides et le chiffre inscrit dans la case sinon.

✽ **Question 1** Écrire une procédure `casevide(grille)` qui renvoie sous forme d'un couple la position d'une case vide dans la grille, ou `[-1,-1]` s'il n'y en a pas

✽ **Question 2** Écrire une procédure `admissible(grille,i,j,x)` qui détermine s'il est possible d'insérer le chiffre `x` à la position déterminée par `i` et `j` dans la `grille` sans violer les règles du Sudoku.

Voilà notre stratégie pour résoudre une grille incomplète.
On cherche une case vide. S'il n'y en a pas, le Sudoku est résolu et on renvoie Vrai.
Sinon, on en choisit une, et pour chaque chiffre c possible :
- On met c dans la case
- On tente de résoudre le Sudoku en appelant récursivement notre algorithme. Si la résolution a réussi, on renvoie Vrai.
- Sinon, on efface le chiffre inséré.

Si aucun chiffre ne convient, on renvoie Faux.

✽ **Question 3** Implémenter cet algorithme et le tester sur des grilles.

2 Jeu de Marienbad

On pose des allumettes sur une table, réparties en rangées. Par exemple, une rangée de 7 allumettes, une rangée de 5, une de 3 et une de 1. C'est un jeu à deux joueurs. Tour à tour, chaque joueur enlève des allumettes. Il doit en enlever au moins une et toutes doivent venir de la même rangée. Celui qui enlève la dernière allumette gagne.

✽ **Question 4** Jouer une partie avec son voisin (une seule, je vous ai à l'œil !)

Une configuration du jeu est une liste d'entiers naturels qui représentent les nombres d'allumettes dans chaque rangée.

Une configuration b est dite accessible à partir d'une configuration a quand on peut passer de a à b en un coup (c'est à dire en réduisant strictement un des éléments de a).

On dit qu'un joueur a une stratégie gagnante à partir d'une configuration a quand
- la configuration a n'est pas `[ 1 ]`
- une configuration b où il n'existe pas de stratégie gagnante est accessible depuis a

✽ **Question 5** Écrire une procédure `estgagnante(config)` qui détermine s'il existe une stratégie gagnante depuis `config`.

✽ **Question 6** Écrire une procédure qui détermine un bon coup à jouer à partir d'une configuration gagnante

Vous avez programmé une intelligence artificielle !

3 Coloriage de graphes

Un graphe non orienté est un couple (S, A) où $A \subset S \times S$: S est l'ensemble des sommets et A est l'ensemble des arêtes.

Un k-coloriage d'un graphe $G = (S, A)$ est une fonction $f : S \to \{1, \ldots, k\}$ telle que si $(u, v) \in A$ alors $f(u) \neq f(v)$.

Autrement dit, on colorie les sommets avec k couleurs et on exige que tout couple de sommets reliés entre eux soit bicolore.

✤ **Question 7** En s'inspirant de l'algorithme de résolution de Sudoku, écrire une procédure qui k-colorie un graphe (si c'est possible)

✤ **Question 8** Comment utiliser votre algorithme pour résoudre un Sudoku ?

4 Le mot de la fin

Il existe une méthode beaucoup plus rapide et assez élégante pour déterminer si une configuration est gagnante dans le jeu de Marienbad. Wikipédia est votre ami.