

Informatique - Master 1

Module IA

TP 1 - Résolution de grilles de Sudoku

2013/2014

Lors de ce TP, vous allez vous familiariser avec les recherches de solutions avec retour arrière. Vous allez appliquer ces méthodes à la résolution de grilles de Sudoku.

Pour rappel, le but du jeu du Sudoku est de compléter une grille à partir de chiffres déjà placés initialement. Nous nous intéresserons ici aux grilles de taille 9×9 . Les contraintes sur le remplissage de la grille sont alors les suivantes :

- chaque ligne doit contenir une seule fois chaque chiffre de 1 à 9 ;
- chaque colonne doit contenir une seule fois chaque chiffre de 1 à 9 ;
- chaque région (de taille 3×3) doit contenir une seule fois chaque chiffre de 1 à 9.

Le jeu est représenté par une grille contenant une matrice de cases.

1 Recherche de solution par retour-arrière

Complétez la fonction *resoudreSudoku(Grille G)* qui effectue la résolution d'une grille, en utilisant un algorithme de recherche par retour-arrière (l'algorithme est donné en annexe).

Le choix d'une case vide à traiter s'effectue ici en choisissant la première case non-vide, en considérant les cases selon leurs indices croissants de ligne et de colonne, dans la grille.

2 Ajout d'une heuristique sur le choix des variables

Modifier l'algorithme précédent pour pouvoir choisir, comme case vide à traiter, la case la plus contrainte, c'est-à-dire celle pour laquelle le nombre de choix de valeurs possibles est le plus petit.

3 Filtrage des valeurs de variables (*forward-checking*)

Modifier à nouveau l'algorithme afin de propager les contraintes sur les valeurs possibles des cases appartenant à la même ligne ou à la même colonne ou à la même région que la case à laquelle une valeur vient d'être affectée.

Algorithme de retour arrière appliqué au Sudoku

booléen backtrack(Grille G)

```
    si G n'a pas de case vide
        renvoyer vrai
    sinon
        pour tout C : case vide de G
            pour tout V : valeur possible pour C dans G
                G' <- copie de G
                Affecter V à la case C de G'
                si G' respecte toutes les contraintes du Sudoku
                    si backtrack(G')
                        renvoyer vrai
                fin si
            fin si
        fin pour
    fin pour
    renvoyer faux
fin
```