

Réalisation d'un solveur du SUDOKU
Par propagation et satisfaction de contraintes

Equipes de 4 étudiants.

Langage C ou C++.

Doit compiler dans un environnement MS Windows.

SUDOKU

Une grille de Sudoku est composée de 9 « petits » carrés de 3 cases sur 3, placés côte-à-côte afin de former un « grand » carré de 9 cases sur 9. Sur la grille, il faut placer des chiffres de 1 à 9 en respectant les règles suivantes :

- les 9 chiffres doivent être présents dans chacun des 9 « petits » carrés ;
- chaque ligne du « grand » carré doit contenir les 9 chiffres (donc chacun une et une seule fois) ;
- chaque colonne du « grand » carré doit contenir les 9 chiffres (donc chacun une et une seule fois).

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Objectif du mini projet

Mise en œuvre de mécanismes de résolution de problème par propagation et satisfaction de contraintes.
Application du mécanisme au SUDOKU.

Ceci implique la réalisation d'un système disposant entre autres des éléments suivants :

- Représentation du problème sous la forme d'un PSC, à savoir :
 - o Représentation des variables qui le compose,
 - o Représentation des domaines de valeurs,
 - o Représentation des affectations effectuées.
- Mise en place d'algorithmes de résolution, à des degrés d'efficacité diverses : voir plus loin.
- Choix de stratégie de résolution : voir plus loin.
- Mise en place d'algorithmes de vérification / propagation de contraintes.
- Production de traces (choix effectués) et indicateurs de mesure de comparaison (nombre d'affectations, nombre de retours en arrière, profondeur de l'arbre analysé, ...)

Algorithmes de résolution

Construction complète de l'arbre d'affectations et vérification aux feuilles

Construction de l'arbre d'affectation complet, en parcours en profondeur d'abord, avec validation des affectations aux feuilles de l'arbre.

Algorithme peu efficace, c'est le moins que l'on puisse dire, mais peut être suffisamment rapide à mettre en œuvre pour vous permettre de valider vos choix de SdD (représentation du problème) et d'algorithme de vérification de contraintes.

Construction élaguée de l'arbre d'affectations par propagation des contraintes

Sur la base de l'algorithme précédent, mais chaque affectation réduit les domaines de valeurs possibles pour les variables n'ayant pas encore de valeur. La taille de l'arbre est ainsi grandement réduite !

Anticipation des culs-de-sac par cohérence d'arêtes

Anticipation des problèmes futurs par mise en œuvre d'un mécanisme AC3.

Stratégie de résolution : choix de la variable à assigner

Quelque soit l'algorithme de résolution choisi, il est possible de l'optimiser (ou non...) en fonction du choix des variables à chaque affectation effectuée (cf. cours) : variable la plus contrainte, la plus contraignante, ...

Travail à réaliser

Etape 1 – Installation de l'environnement

Mise en place des SdD permettant de représenter le problème.

Attention notamment à la place mémoire nécessaire pour représenter toutes les informations nécessaires, car cela aura certainement un impact sur l'exécution de votre programme...

Mise en place de l'algorithme de résolution par construction complète de l'arbre et vérification aux feuilles.

Note : étant donné la complexité du problème, en temps comme en espace, il peut être opportun de réaliser cette étape 1 sur une version simplifiée du SUDOKU, avec une grille de 2X2 « petits carrée » de 2x2 « cases », les chiffres utilisés étant de 1 à 4.

Il est vraisemblable que les choix faits (SdD et algorithmes) seront ensuite exploitables pour le 'vrai' SUDOKU. Assurez-vous en pour ne pas avoir à tout refaire par la suite !!!

Livrables :

Dossier d'architecture technique de votre programme : choix des SdD (préférez des schémas !) et algorithmique de résolution (pseudo-code). Maximum 6 pages.

Etape 2 – Amélioration de la résolution par la propagation de contrainte.

Outre la résolution par propagation de contrainte, votre programme doit fournir des indicateurs de complexité en mémoire (taille de l'arbre) et en temps (nombre d'affectations effectuées, nombre de retour en arrière, ...).

Livrables :

Dossier d'architecture technique de votre programme. Cette seconde version doit impérativement indiquer ce qui a été modifié par rapport à l'étape 1.

Code source.

Etape 3 – Amélioration de votre système

Allez le plus loin possible :

- Mise en place de AC3,
- Utilisation de stratégies de choix de variables différentes et étude comparative des résultats (cf. indicateurs de complexité),
- ...

Livrables :
Dossier d'architecture final.
Code source.

Planning

Etapes 1 et 2 pour les séances de TP. Résultat à fournir en fin de séance.

Etape 3 pour la soutenance. Le résultat de cette étape devra être envoyé à votre enseignant de TD/TP au plus tard 3 jours (week-end non compris) avant votre jour de soutenance.

Organisation des soutenances

30 minutes par équipe.

10 à 20 minutes de présentation et Q&R : chaque membre de l'équipe devra s'exprimer sur un sujet.

10 à 20 minutes de démonstration.

Les soutenances auront lieu en salle TD. Il est donc de votre responsabilité de venir avec un ordinateur portable pour effectuer votre démonstration !

Éléments de notation

Note de groupe (50%) pour le travail effectué. Cette note sera déterminée sur la base : des documents fournis, du code source, du résultat final. La qualité sera primordiale (clarté des documents, lisibilité du code, ...).

Notes individuelles (50%) selon notamment la prestation orale.

Une absence le jour de la soutenance rapportera 0 en note individuelle.

Toute détection de triche avérée (partage de code évidente entre plusieurs équipes, ...) sera sanctionnée de façon globale (0 à tous les membres des équipes concernées), définitive et non contestable.

Bon travail !