

8INF846 – Intelligence Artificielle – Hiver 2022

Travail Pratique #2

Création CSP pour le jeu du Sudoku

Le contexte

Son nouveau projet au sein de l'entreprise d'intelligence artificielle est de construire un robot capable de résoudre des jeux Sudokus. Vous n'avez pas besoin d'utiliser le paradigme de modélisation d'agent vu dans le travail pratique #1, mais le Sudoku doit être représenté sous la forme d'un Problème de Satisfaction de Contraintes (CSP).

Des contraintes à respecter

- La dimension du tableau doit être de 9x9, divisée en 9 sections de 3x3
- Au démarrage, le système doit présenter le tableau Sudoku initial avec quelques cellules pré-remplies
- Le tableau de départ pré-rempli doit être générée aléatoirement
- Le système doit permettre d'ajouter un nouveau Sudoku, c'est-à-dire que l'utilisateur peut créer sa propre table initiale
- Après avoir terminé le jeu, le système devrait présenter le tableau de Sudoku final avec toutes les cellules remplies
- Vous devez respecter la modélisation d'un CSP avec *backtracking*, même s'il existe d'autres moyens plus simples de mettre en œuvre la solution du problème
- Vous devez utiliser au moins une de ces combinaisons d'algorithmes :
 - AC-3 + *backtracking*
 - *backtracking* + MRV avec degré heuristique
 - *backtracking* + LCV avec degré heuristique
- Vous devez modéliser votre CSP à l'aide de contraintes binaires

Entrée et sortie

Pour la saisie des informations dans le système, il n'est pas nécessaire d'avoir une interface graphique. Lors de l'exécution du système, il faut demander à l'utilisateur :

- Générer un tableau Sudoku automatiquement (génération aléatoire) OU
- Générer un tableau à partir d'un fichier .txt
 - La disposition du fichier .txt devrait être comme ci-dessous, où zéro sont des cellules vides :

```

sudoku.txt
1  000500008
2  400030020
3  030700000
4  006000040
5  080091002
6  000000000
7  000070000
8  090023001
9  800000900

```

Il n'y a également pas besoin d'une interface graphique pour la sortie du système. Voici des exemples de sortie via console :

```

Sudoku de départ :

| | | 5 | | | 8 |
4 | | | 3 | | 2 | |
| 3 | 7 | | | | |

| | 6 | | | | 4 | |
| 8 | | 9 | 1 | | 2 | |
| | | | | | | |

| | | 7 | | | | |
| 9 | | 2 | 3 | | 1 | |
8 | | | | | 9 | |

-----

Sudoku de fin :

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

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

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

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

-----

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

```

Autres informations importantes

- La modélisation et la mise en œuvre doivent être cohérentes avec les concepts vus en classe. Vous perdrez des points si les modèles sont inadaptés ou si l'algorithme est copié d'internet, même s'il fonctionne parfaitement.
- Le langage de programmation le plus recommandé est **Python**. Si vous souhaitez utiliser une autre langue, vous devez demander l'approbation du professeur.
- Vous devez former un groupe de **3 ou 4 personnes**.

Soumission du travail

- Vous devez soumettre une **copie électronique de votre code source**.
- Un **rapport** (en format .pdf) contenant les sections suivantes :
 - Page de titre avec **les noms, les code permanents et le titre du cours** ;
 - Section expliquant **comment exécuter votre programme** : quelle version de langue (Python, Java, etc.) doit être installée, quelles bibliothèques supplémentaires doivent être installées, paramètres supplémentaires, etc. Indiquez quel fichier .py exécuter. Attention : si le professeur a besoin de modifier une ligne de votre code pour exécuter votre programme, vous perdrez des points.
 - Section expliquant **comment vous avez modélisé la solution**, quelles sont les classes AC-3, classes *backtracking*, etc. Vous pouvez utiliser des schémas si vous préférez. Essayez d'utiliser le standard (par exemple *Unified Modeling Language* - UML) pour faire vos diagrammes.
 - Section avec une **capture d'écran du code source** de chacun des éléments ci-dessous et **une brève explication** de chacun :
 - L'algorithme de *backtracking*
 - L'algorithme AC-3 et/ou MRV et/ou LCV
 - Heuristique lorsque MRV ou LCV
 - Collection de contraintes binaires (liste, array, matrix, etc.)
- La **soumission** doit être **sur Moodle** uniquement.
- Une **seule soumission par groupe**.

Date limite de remise : 07 mars 2022 à 23h59 (heure de Chicoutimi, QC, Canada)

Bon travail !