Laboratoire 3 (partie 2/2) - Programmation par contraintes

Avant propos

Ces exercices ont pour objectif de vous familiariser avec la programmation par contraintes ainsi qu'au langage de modélisation Minizinc qui est utilisé pour le Devoir 2. Il est donc bien important de bien comprendre comment utiliser ce langage. A cette fin, plusieurs ressources (tutoriel, documentation, cours complémentaire, etc.) vous sont rendues disponibles sur la page Moodle. Il y a volontairement beaucoup d'exercices pour vous permettre de pratiquer, et il n'est pas attendu à ce que vous les réalisiez tous lors de la séance de laboratoire.

⚠**Savoir réaliser un modèle MiniZinc ne fait pas partie de la matière d'examen. Par contre, vous devez être capable de fournir un modèle sur papier d'un CSP ou d'un COP.**

Exercice 1 : Carré magique

Un carré magique de taille n est composé de n^2 entiers *différents*, consécutifs à partir de 1, et strictement positifs, écrits sous la forme d'une matrice carré de dimension $n \times n$. Les nombres sont disposés de sorte que leurs sommes sur chaque rangée, sur chaque colonne et sur chaque diagonale principale soient toutes égales.

1. Modélisez ce problème sur papier et sur MiniZinc en programmation par contraintes. La contrainte `all_different` vous sera utile pour ce problème. Regardez dans la documentation de MiniZinc comment l'utiliser. Un fichier .mzn exemple vous est donné, remarquez la ligne `include "globals.mzn";` qui indique que l'on souhaite importer des contraintes globales.
2. Résolvez le problème pour un carré de taille 3. Utilisez le solveur `Gecode` ou `Chuffed`, qui sont tous les deux des solveurs de programmation par contraintes. Affichez toutes les solutions obtenues. Pour cela, changez la configuration du solveur. Combien de solutions obtenez vous ?
3. Des solutions *symétriques* sont des solutions différentes du point de vue d'un modèle mais que l'on peut considérer comme similaires. On souhaite généralement les enlever afin de réduire l'espace de recherche et le nombre de solutions. Dans notre cas, on peut considérer que les carrés obtenus par rotation d'un autre sont identiques. Comment faire pour éliminer cette symétrie dans votre modèle ? Combien de solutions faisables obtenez vous après avoir éliminé les symétries ?

Exercice 2 : Série magique

Une série magique de taille n est une série qui prend ses valeurs dans l'intervalle $[0, n-1]$ et qui vérifie la propriété suivante : le i^e élément correspond au nombre d'occurrences de l'élément i dans la série.

1. Modélisez ce problème sur MiniZinc en programmation par contraintes.
2. Résolvez le problème pour des séries de taille 4, 50, 100 et 400. Utilisez le solveur `Gecode` ou `Chuffed`.
3. Des *scontraintes redondantes* sont des contraintes qui ne sont pas nécessaires pour que le modèle soit correct (i.e., elles sont redondantes par rapport à d'autres) mais qui peuvent être utilisées pour réduire l'espace de recherche. Exprimez deux contraintes redondantes pour ce problème et ajoutez les à votre modèle.

4. Comparez votre nouveau modèle avec le précédent. Qu'observez vous ? Note, vous pouvez afficher dans MiniZinc les statistiques de résolution, comme le nombre de noeuds exploré.

Exercice 3 : The Mother Chord

En composition musicale, il existe un accord appelé le *Mother chord*. Cet accord consiste à former une séquence des 12 notes fondamentales (do, $do\sharp$, re, $re\sharp$, mi, fa, $fa\sharp$, sol, $sol\sharp$, la, $la\sharp$, si) de sorte que chaque note n'apparaisse qu'une seule fois dans la séquence. Il en va de même pour la valeur absolue de la différence de demi-tons entre deux notes consécutives dans la séquence qui ne peut apparaître qu'une seule fois. Pour simplifier l'écriture, nous utiliserons une dénomination numérique de 1 à 12 des notes ($do = 1$, $do\sharp = 2$, etc.). Par exemple, la séquence $[1, 12, 9, 8, 5, 3, 6, 2, 1, 10, 11, 4]$ n'est pas valide, car la note 1 apparait deux fois. Par ailleurs, une différence de demi-ton de 3 apparait également au moins deux fois ($12 - 9 = 3$ et $8 - 5 = 3$), ce qui n'est pas valide.

1. Résolvez ce problème et trouvez un accord respectant ces contraintes. Pour cela, modélisez ce problème sur papier et sur MiniZinc.
2. Afin de dépasser le cadre musical, proposez un modèle pouvant gérer une séquence de taille n de sorte que les deux conditions soient toujours respectées.

Exercice 4 : Drones de surveillance

Vous êtes en charge de la surveillance d'une zone carrée de dimension $n \times n$. Pour cela, vous disposez de n drones qui ont la faculté de détecter une intrusion sur les cases se trouvant sur les positions situées à la verticale, l'horizontale et aux diagonales de sa position. Ces lignes définissent le champ de vision du drone. Lorsqu'un drone se situe dans le champ de vision d'un autre, il y a une interférence : ils vont détecter une intrusion alors qu'il ne s'agit que d'un autre drone. Votre objectif est de placer n drones sur la zone de sorte qu'aucun champ de vision ne se superpose.

1. Implémentez un modèle générique pour résoudre ce problème. Un fichier .mzn exemple vous est donné. Utilisez le solveur `Gecode` et résolvez une situation d'un carré 10×10.
2. Augmentez progressivement la taille du problème, et regardez jusqu'à quelle taille le solveur est capable de trouver une solution en moins de 5 secondes.

Exercice 5 : SEND + MORE = MONEY

Le problème *SEND + MORE = MONEY* consiste à assigner chacune des variables { S, E, N, D, M, O, R, Y} une valeur *différente* dans { 0, ..., 9} de sorte à ce que l'équation suivante soit vérifiée :

$$
\begin{array}{r}
S\ E\ N\ D \\
+\, M\ O\ R\ E \\
\hline
M\ O\ N\ E\ Y
\end{array}
$$

1. Supposons que vous souhaitiez résoudre ce problème par recherche exhaustive. Quelle est la taille de l'espace de solutions à explorer ?

2. Modélisez ce problème sur papier et sur MiniZinc en programmation par contraintes. Reportez le nombre de noeuds explorés dans votre arbre de recherche.
3. Existe-il une solution unique à ce problème ? Posons maintenant que S et M soient tout deux différents de zéros, que pouvez-vous conclure ?

Exercice 6 : L'énigme d'Einstein

"L'énigme d'Einstein, aussi appelée l'énigme des cinq maisons, est un intégramme qui aurait été inventé par le physicien et mathématicien Albert Einstein, bien que ça n'ait jamais été prouvé. Elle est apparue pour la première fois dans le magazine Life le 17 décembre 1962, soit 7 ans après la mort d'Einstein en 1955. Selon ceux qui publient maintenant cette énigme, Einstein aurait dit que seulement 2 % de la population est capable de la résoudre de tête." Wikipédia.

Le problème est le suivant : il y a cinq maisons de cinq couleurs différentes, alignées le long d'une route. Dans chacune de ces maisons, vit une personne de nationalité différente. Chacune de ces personnes boit une boisson différente, pratique un sport différent et a un animal domestique différent. On connait les informations suivantes :

1. L'Anglais vit dans la maison rouge.
2. Le Suédois a des chiens.
3. Le Danois boit du thé.
4. La maison verte est à gauche de la maison blanche.
5. Le propriétaire de la maison verte boit du café.
6. La personne qui joue au football élève des oiseaux.
7. Le propriétaire de la maison jaune joue au baseball.
8. La personne qui vit dans la maison du centre boit du lait.
9. Le Norvégien habite dans la première maison.
10. L'homme qui pratique le volley vit à côté de celui qui a des chats.
11. L'homme qui a un cheval est le voisin de celui qui pratique le baseball.
12. Celui qui pratique le tennis boit de la bière.
13. L'Allemand joue au hockey.
14. Le Norvégien vit juste à côté de la maison bleue.
15. L'homme qui joue au volley a un voisin qui boit de l'eau.

La question est de savoir qui possède le poisson. Résolvez ce problème sur MiniZinc.