Université Paris 7 - Master 1 Informatique - Prolog et Programmation par Contraintes

Examen du 20 juin 2011 - Durée : 2 heures

Informations : Tous les documents reliés sont autorisés. Le barème est donné à titre indicatif et peut être modifié. Les deux parties sont à rendre séparément.

Partie 1

Exercice 1 (**5 points**) Une liste d'entiers est *biordonnée* si ses éléments en position impair (le premier, troisième, ...) sont rangés en ordre croissant, et ses éléments en position pair (le deuxième, quatrième, ...) sont rangés en ordre décroissant. Par exemple `[3,12,5,7,9]` est biordonnée, et `[3,12,2,7,9]` ne l'est pas.

- Définir un prédicat `biordonne(+liste_argument)` qui réussit si la liste argument est biordonnée, et échoue sinon.
- Définir un prédicat `inverser(+liste_argument,-liste_resultat)`, qui renvoie dans `liste_resultat` la liste des éléments en position pair de `liste_argument`, inversée ; par exemple

```
 ?- inverser([3,5,4,9,1,3,8,13,15],L).
L = [13,3,9,5] ?
```

- Définir un algorithme efficace (de complexité linéaire, en termes de comparaisons entre entiers effectuées lors de l'exécution) pour trier dans l'ordre croissant une liste biordonnée, et l'implémenter en Prolog.

Exercice 2 (**5 points**)

On considère la variante suivante du jeu de 16 allumettes : une position du jeu consiste en deux tas d'objets, le tas gauche G et le tas droit D.

Les coups jouables sont :

Transfert : Un objet est transféré de G à D (à condition que G ne soit pas vide).

Élimination : Deux objets sont retirés de D (à condition que D contienne au moins deux objets).

Deux joueurs jouent à tour de rôle, jusqu'à ce que l'un d'entre eux ne puisse plus jouer. Le joueur qui ne peut plus jouer est le perdant.

1. Choisir une représentation des positions du jeu.
2. Définir , en fonction de la représentation choisie, un prédicat `move(+position_courante, -position_suivante)` qui implémente la règle du jeu.
3. Définir un prédicat `gagne(+position)` qui réussit si la position donnée est gagnante.
4. Considérons la position P dans laquelle les tas G et D contiennent deux objets chacun. Dessiner l'arbre de jeu dont la position initiale est P, et le joueur MAX joue. Évaluer (à la main) cet arbre en utilisant l'algorithme minimax à arbre de jeu (on dira que la valeur d'une feuille est 1 si MAX gagne, 0 si MIN gagne).
5. Considérons la variante de ce jeu où un seul objet est retiré lors d'un coup "élimination". Soient n et m les tailles respectives de G et D au début du jeu. Donner une condition nécessaire et suffisante , portant sur n et m, pour que le joueur qui commence ait une stratégie gagnante.

   Facultatif : même question pour le jeu non modifié.

Partie 2

Exercice 3 (**4 points**) Considérez le problème suivant :

Minimiser $-X - 3 * Z + 2$ par rapport à

$2 * Z + X \leq 6 \wedge$

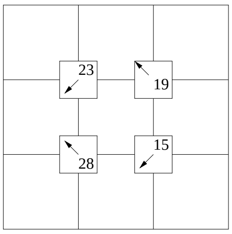
$2 * Z - X \leq 4 \wedge$

$Z + 2 * X \leq 7 \wedge$

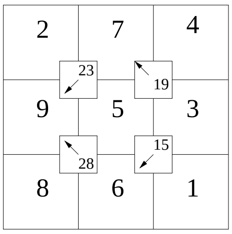
$X \geq 0 \wedge Z \geq 0$

- Visualiser le problème en dessinant l'espace des solutions possibles (un polygone) déterminé par les contraintes.
- Donner les détails de la résolution de ce problème par la méthode simplex. **Indication :** Pour cela, il faut entre autres mettre le problème en forme simplex, trouver une solution de base, etc.
- Pour quelles valeurs de X et Z le minimum est-il atteint ?
- Si on exige que X et Z soient entières, quel est le minimum ? Pour quelles valeurs de X et Z est-il atteint ? Répondre à ces deux questions sans justification.

Exercice 4 (**3 points**) On considère la grille suivante :



Il faut remplir les neuf grandes cases de la grille avec des chiffres de 1 à 9. Chaque chiffre ne peut être utilisé qu'une seule fois. Les nombres dans les quatre petites cases correspondent à la sommes des chiffres des cases adjacentes. Les flèches indiquent le plus grand des quatre chiffres adjacents. Par exemple la solution de la grille ci-dessus est donnée par :



- Modéliser le problème de l'exemple comme un problème de satisfaction de contraintes (Quelles sont les variables, leur domaines et les contraintes ?)
- Donner un programme en GPROLOG qui résout le problème.

Exercice 5 (**3 points**) Rendre bornes-consistantes la contrainte
$2 * X + 3 * Y \leq 2 * Z \ \wedge \ 3 * X + Y \geq 15 + Z$
avec les domaines $D(X) = [3..11]$ et $D(Y) = [2..14]$ et $D(Z) = [5..12]$.