Master ILI et SIA – Université d'Artois **2009/2010**
Programmation par contraintes

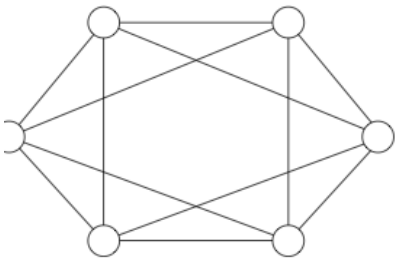
TD : programmation logique par contraintes – domaines finis

Exercice 1 : (Zebra problem)
Cinq personnes de nationalités différentes habitent 5 maisons différentes (qui sont sur le même côté de la rue) de couleur différente. Chaque personne a une profession différente, un animal préféré différent, une boisson préférée différente.

- L'anglais habite la maison rouge.
- L'espagnol a un chien.
- Le japonais est peintre.
- L'italien boit du thé.
- Le norvégien habite la première maison.
- L'habitant de la maison verte boit du café.
- La maison verte vient après la maison blanche.
- Le sculpteur élève des escargots.
- Le diplomate habite la version jaune.
- Dans la troisième maison on boit du lait.
- La maison du norvégien est à côté de la maison bleue.
- Le violoniste boit du jus d'orange.
- Il y un renard dans la maison à côté de celle du docteur.
- Il y a un cheval dans la maison à côté de celle du diplomate.
- Il y a un zèbre dans la maison blanche.
- Il y a une personne qui boit de l'eau.

Écrire un prédicat zebra(X) qui résout ce puzzle, et qui unifie X avec le numéro de la maison dans laquelle est le zèbre. On attend de vous d'écrire un programme en Sictus-Prolog, on n'attend pas que vous trouvez le numéro de la maison du zèbre vous-même !
Indication : Les maisons ont les numéros 1 à 5. Vous utiliserez plusieurs variables à domaine fini; chacune des variables dénote un numéro de maison.

Exercice 2 : (coloriage de graphes)
Ecrire un programme prolog permettant de calculer le nombre minimum de couleur permettant de colorier correctement le graphe suivant :



Un graphe est correctement colorié si deux nœuds adjacents dans le graphe ne reçoivent pas la même couleur.

Exercice 3 : (investissements)
Un investisseur dispose de 14000 $ de réserve d'argent. Il souhaite investir son capital de manière à maximiser les gains ou NPV (Net Present Value). Le tableau suivant indique pour chaque investissement (investment), le budget nécessaire (cash required) et le gain espéré (NPV).

investment	1	2	3	4	5	6
cash required	$5k	$7k	$4k	$3k	$4k	$6k
NPV	$16k	$22k	$12k	$8k	$11k	$19k

cash = $14k

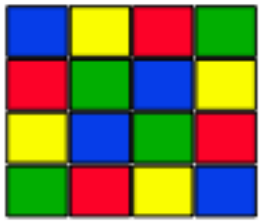
NPV = $42k

Il s'agit d'écrire un programme permettant de trouver la liste des investissements à faire pour maximiser ses gains.

Exercice 4 : (Latin squares)

Étant donnée n couleurs, un carré latin (ou quasigroup) d'ordre n est un carré n x n colorié tel que :

- toute cellule est coloriée
- chaque couleur apparaît exactement une fois sur chaque ligne
- chaque couleur apparaît exactement une fois sur chaque colonne

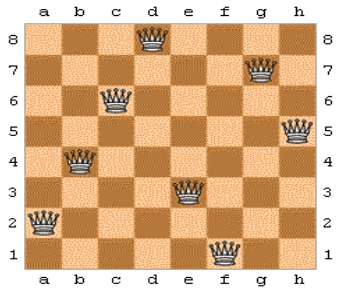


Latin squares d'ordre 4

Écrire un programme permettant de résoudre ce problème.

Exercice 4 : (n-reines)

Étant donné un échiquier de taille n x n, le problème de n-reines consiste à placer n reines sur un tel échiquier sans que deux reines en soient en prises : deux reines ne peuvent pas être sur la même ligne, colonne ou diagonale. Écrire le prédicat reines(N, L) qui résout ce problème où N est la taille de l'échiquier et L la liste des variables de taille N. Une variable Li de L indique que la reine de la ligne numéro i est placée sur la colonne Li.



8-reines

Exercice 5 (sudoku)

Le problème du Sudoku consiste à remplir la grille de sorte que chaque ligne, chaque colonne et chaque carré contiennent les chiffres 1 à 9. Exemple :

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

sudoku 9 x 9

Ecrire un programme en sicstus Prolog pour résoudre des sudoku. Vous écrirez un prédicat sudoku(L) qui réussit quand le sudoku avec les valeurs initiales données par L a une solution.

Deux solutions peuvent être envisagées :

- La liste L est une liste de triplets (ligne, colonne, valeur).
- La liste L est une liste de lignes