

Université de Rennes 1
IFSIC
Master 1 d'informatique
Programmation, logique et calcul

2eme session 2005-2006
16 juin 2006, 14-16h

Ce sujet comporte 2 pages. Tous les documents sont autorisés. Le barème est indicatif. *Toutes les réponses doivent être justifiées (de façon concise mais précise).*

Partie I - Manipulation de termes (8 points)

Question 1

Ecrire le prédicat `liste_des_plus_grands_que_E/3`, qu'on notera en abrégé `lpg/3`, de mode `lpg(+E, +A, -L)`, où `A` est un arbre binaire, `E` est un entier et `L` est une liste, et tel que `L` est la liste des éléments de `A` strictement supérieurs à `E`.

Un arbre binaire sera représenté par un terme Prolog au moyen d'un constructeur `a/2`. Les étiquettes des feuilles sont des entiers. Exemple : `a(a(4, 8), a(5, 1))` .

Ainsi le but `lpg(4, a(a(a(4, 8), 5), a(10, a(2, 8))), L)` produit le résultat `L = [8, 5, 10, 8]` .

Question 2

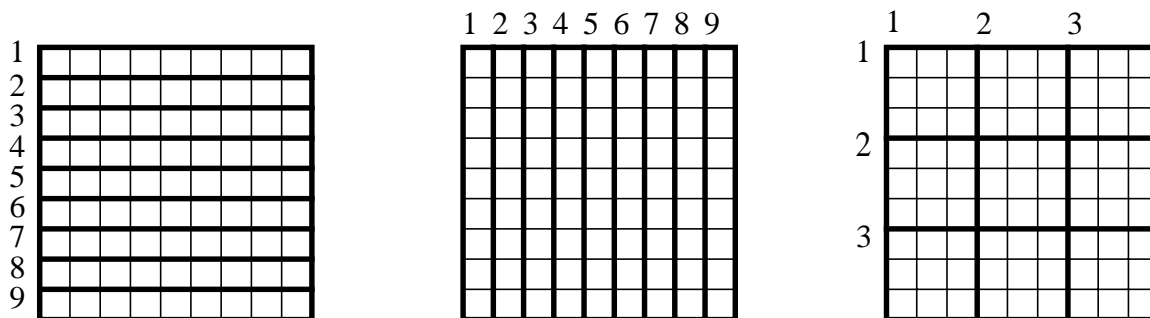
Ecrire une nouvelle version, `lpgBis/3`, de votre prédicat `lpg` qui sera rendu plus efficace du fait de l'utilisation des D-listes pour représenter les résultats intermédiaires.

Partie II - Programmation par contraintes (12 points)

Soit un problème de Sudoku générique.

Sa donnée est une grille $N \times N$ où N est un nombre carré M^2 et un remplissage partiel de la grille à l'aide de symboles choisis parmi un vocabulaire de N symboles. Par exemple, une grille habituelle est une grille $3^2 \times 3^2$.

Une telle grille peut être vue selon les besoins comme un liste de N lignes à N positions, ou bien une liste de N colonnes à N positions, ou une grille de $M \times M$ carrés à $M \times M$ positions. La figure suivante illustre ces trois points de vue dans le cas d'une grille $3^2 \times 3^2$ et propose une numérotation des lignes, colonnes et carrés :



Traditionnellement, les N symboles sont les N premiers entiers naturels.

L'objectif est de compléter le remplissage de la grille avec des symboles choisis parmi le même vocabulaire de N symboles en respectant les contraintes suivantes :

1. Tous les symboles d'une même ligne sont différents.
2. Tous les symboles d'une même colonne sont différents.
3. Tous les symboles d'un même carré sont différents.

Une autre façon d'exprimer ces contraintes et de dire que chacun des N symboles doit apparaître dans *toutes* les lignes, *toutes* les colonnes et *tous* les carrés.

Question 3

Pourquoi ces deux façons d'exprimer les contraintes sont-elles équivalentes ?

Question 4

Soit la représentation d'une grille comme une liste de N lignes de N positions.

Programmer à l'aide de la contraintes `alldifferent/1` les contraintes qui s'exercent sur une ligne I , une colonne J et un carré de coordonnées U et V . Présenter toutes les sections du programme.

Question 5

Soit la représentation d'une grille comme une liste de N carrés de N positions.

Programmer à l'aide de la contraintes `card/3` les contraintes qui s'exercent sur une ligne I et un carré de coordonnées U et V . Ne présenter que la section qui concerne l'imposition des contraintes.

Question 6

Quel pourrait être l'intérêt de combiner les deux solutions ?