Librairie des domaines finis

Listes en Prolog : la liste formée des éléments a,b,c,d s'écrit [a,b,c,d], la liste formée d'un élément e suivi par les éléments de la liste LP s'écrit [e|LP]. (Les éléments peuvent être des variables).

Exercice 1
Le problème suivant correspond au choix de casernes de pompiers dans les quartiers d'une grande ville (la modélisation sera vue en TD). Cela revient à minimiser $z = \sum_{i=1}^{11} x_i$ sous les contraintes suivantes :

$$
\begin{array}{l}
1 \leq x_1 + x_2 + x_3 + x_4 \\
1 \leq x_1 + x_2 + x_3 + x_5 \\
1 \leq x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
1 \leq x_1 + x_3 + x_4 + x_6 + x_7 \\
1 \leq x_2 + x_3 + x_5 + x_6 + x_8 + x_9 \\
1 \leq x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \\
1 \leq x_4 + x_6 + x_7 + x_8 \\
1 \leq x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \\
1 \leq x_5 + x_8 + x_9 + x_{10} + x_{11} \\
1 \leq x_8 + x_9 + x_{10} + x_{11} \\
1 \leq x_9 + x_{10} + x_{11} \\
x_i \in \{0, 1\}
\end{array}
$$

1. Supprimez les contraintes redondantes puis lancez `Eclipse` avec la librairie `clpq` pour calculer le nombre z minimal dans les rationnels (après avoir relaché la contrainte d'appartenance en $0 \leq x_i \leq 1$).
2. On souhaite maintenant obtenir toutes les solutions $(x_1, x_2, ..., x_{11})$ minimales. Recopier puis ouvrir le fichier `~lugiez/TPMI/fd1.pl` :

```
L=[X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11],
L:: 0..1,
X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11#=3,
1 #=< X1 + X2 + X3 + X4,
1 #=< X1 + X2 + X3 + X5,
1 #=< X1 + X2 + X3 + X4 + X5 + X6,
1 #=< X1 + X3 + X4 + X6 + X7,
1 #=< X2 + X3 + X5 + X6 + X8 + X9,
1 #=< X3 + X4 + X5 + X6 + X7 + X8,
1 #=< X4 + X6 + X7 + X8,
1 #=< X5 + X6 + X7 + X8 + X9 + X10,
1 #=< X5 + X8 + X9 + X10 + X11,
1 #=< X8 + X9 + X10 + X11,
1 #=< X9 + X10 + X11,
labeling(L).
```

   Utiliser le manuel sur la librairie *domaine fini* (fd) pour comprendre le programme.

   Lancer `Eclipse` avec la librairie `fd` (en la chargeant par `lib(fd).`) puis compiler `fd1.pl`. Qu'observe-t-on ? Combien y a-t-il de solutions.

Exercice 2
Reprendre le problème de sac à dos de la dernière séance et le résoudre en utilisant la librairie *fd*.

Rappel : la charge utile est $14kg$, le premier objet pèse $5kg$ et se revend avec un bénéfice de 8 euros, le second pèse $7kg$ et rapporte 11 euros, le troisième pèse 4kg et rapporte 6 euros et le dernier pèse 3kg et rapporte 4 euros.

Exercice 3
Le problème suivant provient d'une question de répartition de sucursales pour une banque (il sera vu en TD) et revient à minimiser

$$z = 50.(y_1 + y_2 + y_3 + y_4) + \sum_{i=1}^{4} \sum_{j=1}^{4} a_{i,j}.x_{i,j}$$

où les coefficients $a_{i,j}$ sont déterminés par la matrice $A = (a_{i,j})$ telle que

$$A = \begin{pmatrix} 28 & 84 & 112 & 112 \\ 60 & 20 & 50 & 50 \\ 96 & 60 & 24 & 60 \\ 64 & 40 & 40 & 16 \end{pmatrix}$$

D'autre part, les contraintes sur les 20 inconnues $x_{i,j}$ et y_j dans $\{0, 1\}$ sont les suivantes :

$$\begin{array}{lcll} \sum_{j=1}^{4} x_{i,j} & = & 1 & \text{pour chaque } i \in [1, 4] \\ \sum_{i=1}^{4} x_{i,j} & \leq & 100.y_j & \text{pour chaque } j \in [1, 4] \end{array}$$

Il s'agit d'utiliser `Eclipse` pour résoudre ce problème.

1. Quelle est la valeur z optimale si l'on néglige la contrainte $x_{i,j}, y_j \in \{0, 1\}$?
2. Pourquoi peut-on remplacer les quatre dernières contraintes par les seize contraintes $x_{i,j} \leq y_j$?
3. Utilisez le nouveau modèle pour trouver la valeur optimale entière.
4. La valeur optimale est-elle atteinte par plusieurs solutions ? Si oui, trouver une solution optimale particulière.

Exercice 4 *(Résolution de puzzles sur domaine finis)*

1. Reprendre l'exemple du puzzle SEND+MORE=MONEY du manuel et trouver la(es) solution(s) avec Eclipse.
2. Trouver tous les sous-ensembles de trois nombres distincts X,Y,Z inclus dans [1..10] tels que $X + Y * Z = 10$
3. Trouver tous les entiers positifs X et Y tels que leur somme vaut 15 et leur produit 50.
4. Résoudre l'équation :

   $$\begin{array}{ccccc} & A & B & A & C \\ - & F & E & D & B \\ \hline & G & A & B & F \end{array}$$

5. On veut peindre une rangée de 5 maisons avec les couleurs *bleu, rouge,jaune* de manière à ce que chaque couleur est utilisée au moins une fois et qu'au plus plus deux maisons soient en *bleu*. Trouver les dispositions possibles.

   Le prédicat *atleast(element,liste)* qui est vrai si *liste* contient au moins un exemplaire de l'element peut se définir par les clauses :

   ```
   atleast(X,[X|L]):-true.
   atleast(X,[Y|L]):-atleast(X,L).
   ```

6. Même question si on demande en plus que si une maison est en *rouge* alors aucune de ses voisines ne soit en *bleu*.