Université d'Orléans **M1 Informatique**
Programmation en logique et par contraintes **Année 2009-2010**

Contrôle continu du 6 novembre 2009

Durée : 2h. Documents autorisés : notes de cours et de TD. Expliquez et commentez bien vos programmes. Le sujet est sur 2 pages.

Exercice 1 Pour spécifier la relation *dernier(L,X)* qui indique que X est le dernier élément de la liste L, on propose le programme Prolog suivant :

```
dernier([X],X).
dernier([_|L],X) :- dernier(L,X).
```

1. Donner l'arbre de dérivation pour chacun des buts suivants :
   ```
   dernier([a,b,c],X).
   dernier(L,a).
   ```
2. Même question si l'on définit la relation *dernier* par le programme suivant :
   ```
   dernier([X],X) :- !.
   dernier([_|L],X) :- dernier(L,X).
   ```
3. Ecrire un prédicat *ad(L,X)* qui définit la relation où X est l'avant dernier élément de la liste L.
4. Ecrire un prédicat *kd(K,L,X)* qui définit la relation où X est le k-ième à partir de la fin de la liste L. Conseil : vous pouvez utiliser le prédicat pré-définit `length(L,N)`.
   ```
   ?- kd(4,[a,b,c,d,e,f,f],X).
   X = d
   ```

Exercice 2 On suppose modéliser une hiérarchie dans une entreprise à l'aide des prédicats suivants :
- employe(X) : est vrai si X est un(e) employé(e) de l'entreprise.
- grandpatron(X) : est vrai uniquement si X est le grand patron (supposé unique) de l'entreprise.
- superieurdirect(X,Y) : est vrai si X est un supérieur hiérarchique direct de Y.

On suppose en outre que, dans cette entreprise, chaque employé a un 'rang hiérarchique" unique : le grand patron est de rang 1, et si X est supérieur hiérachique direct de Y, cela implique que le rang de Y vaut le rang de X plus 1. Dans toutes les questions suivantes (sauf la question 4), faire en sorte qu'on puisse toujours interroger les prédicats à écrire en mettant une variable en dernier argument, pour construire le résultat et pas seulement vérifier une valeur fournie.

1. Définir le prédicat rang(X,N) qui est vrai si N est le rang de l'employé X.
2. Pour que la hiérarchie soit un arbre, il faut s'assurer en outre que chaque employé a un unique supérieur hiérarchique direct. Quelle requête permet de tester cela ?
3. Définir le prédicat unsuperieur(E,S) qui est vrai si S est un supérieur (pas forcément direct) de E.
4. Définir le prédicat unehierarchie(L) qui est vrai si la liste L est une suite d'employés en ordre de hiérarchie directe.
5. En supposant que la hiérarchie est un arbre, définir un prédicat superieur(E,L) qui est vrai si la liste L est la liste de tous les supérieurs, directs ou indirects, de E (note : on peut très bien se passer de métaprédicats pour le faire).
6. Définir le prédicat listesuperieurs(L1,L2) qui est vrai si la liste L2 contient l'ensemble des supérieurs hiérarchiques communs à tous les employés de la liste L1 (indication : définir d'abord un prédicat pour construire l'intersection entre deux listes).

Exercice 3 Matrices

Dans cet exercice, on présente les matrices par des listes de listes, c'est-à-dire une matrice de taille $n \times m$ est une liste de n éléments dont chacun est une liste de m éléments.

1. Définir un prédicat *matrice(L,N,M)* : L est une matrice de taille $N \times M$.
2. La somme de deux matrices $(a_{ij})_{n\times m}$ et $(b_{ij})_{n\times m}$ est la matrice $(a_{ij} + b_{ij})_{n\times m}$. Définir un prédicat *somMat(A,B,S)* qui représente la relation où S est la matrice somme de A et B. Indication : définir d'abord un prédicat *somL(La,Lb,Ls)* pour la relation somme de deux lignes.
3. Le produit d'une matrice $(a_{ij})_{n\times m}$ avec une constante c est la matrice $(ca_{ij})_{n\times m}$. Définir un prédicat *prodConst(X,A,R)* définissant la relation où R est le résultat du produit de la matrice A avec la constante X.