

## Contrôle terminal du 12 janvier 2010

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

**Exercice 1** On appelle motif dans une liste une suite non vide d'éléments consécutifs que l'on représente également par une liste. Un même motif peut avoir plusieurs occurrences dans une liste. On appelle position d'une occurrence le rang du premier caractère du motif dans la liste pour cette occurrence. Par exemple la liste  $[a, a]$  est un motif de la liste  $[a, a, a, b, a, a]$  ayant trois occurrences en positions 1, 2 et 5.

Etant données deux listes  $L$  et  $M$  ( $M$  non vide), écrire les prédicats suivants :

1. Un prédicat  $entete(M, L)$  pour tester si  $M$  est une entête de  $L$  ( $M$  a une occurrence en position 1 dans  $L$ ).
2. Un prédicat pour calculer le nombre d'occurrences de  $M$  dans  $L$  (pensez que deux occurrences peuvent se chevaucher).
3. Un prédicat pour calculer la position d'une occurrence de  $M$  en tant que motif de  $L$  (échec si aucune occurrence).
4. Un prédicat pour calculer une liste binaire  $[b_1, \dots, b_l]$ , avec  $l$  la longueur de  $L$ , telle que  $b_i = 1$  si  $i$  est la position d'une occurrence de  $M$  dans  $L$ , et  $b_i = 0$  sinon. Par exemple si  $M = [a, a]$  et  $L = [a, a, a, b, a, a]$ , alors la liste binaire est  $[1, 1, 0, 0, 1, 0]$ .
5. Utilisez le prédicat précédent pour calculer la liste des positions de  $M$  en tant que motif de  $L$ . Dans l'exemple précédent le résultat sera la liste  $[1, 2, 5]$ .
6. En déduire un prédicat pour tester si l'ensemble des occurrences du motif  $M$  recouvre  $L$  (c'est-à-dire si tout élément de  $L$  appartient à au moins une occurrence de  $M$ ).
7. Etant donné également un indice  $I$ , écrire un prédicat pour calculer la liste de position des occurrences de  $M$  qui couvre l'élément d'indice  $I$  dans  $L$ . Par exemple, si  $M = [a, a]$ ,  $L = [a, a, b, a, a, a]$  et  $I = 5$ , alors le résultat est la liste  $[4, 5]$ .

**Exercice 2** On veut placer les chiffres de 1 à 9 dans une grille de taille  $3 \times 3$ , de sorte qu'il n'y ait pas de ligne ni de colonne qui est composée de trois chiffres consécutifs. Pour résoudre ce problème, on utilisera une liste de 9 variables  $X_1, \dots, X_9$ , chacune a comme domaine l'intervalle  $[1, 9]$ .

X1	X2	X3
X4	X5	X6
X7	X8	X9

1. Quelle est la contrainte pour indiquer que la première ligne n'est pas composée de trois chiffres consécutifs ? et celle pour la première colonne ?
2. En utilisant cette modélisation, écrivez un programme en Gnu-Prolog avec contraintes pour résoudre ce problème.

**Exercice 3** Ecrivez un programme permettant d'évaluer une expression arithmétique sous forme d'un terme avec les symboles de fonction `add`, `sub`, `mult` et `div` et des nombres. Par exemple :

```
?- eval(5.5,N).  
N = 5.5  
?- eval(mult(5.5,add(1.2,3.3)),N).  
N = 24.75
```

Le prédicat `number(X)` permet de tester si `X` est un nombre.

**Exercice 4** On considère le programme suivant :

```
a(1).  
a(2).  
b(3).  
b(4).  
p(X,Y) :- X>0, !, a(Y).  
p(_,Y) :- !, b(Y).  
p(0,0).  
q(X,Y) :- X>0, a(Y), !.  
q(_,Y) :- b(Y).  
q(0,0).
```

Donnez toutes les réponses des buts suivants :

1. `p(1,Y)`.
2. `p(0,Y)`.
3. `q(1,Y)`.
4. `q(0,Y)`.