

# Examen de Prolog

3 juin 2015

La clarté et la précision de la rédaction seront prises en compte dans l'évaluation.

Le barème est indicatif.

Aucun document autorisé.

Machines (ordinateurs, calculatrices, téléphones...) interdites.

Durée : 2h

Un cryptarithme est une opération arithmétique dans laquelle chaque chiffre a été remplacé par une lettre. Une même lettre représente toujours le même chiffre, deux lettres différentes représentent deux chiffres différents. Le but du jeu est de trouver une correspondance entre lettres et chiffres qui donne un résultat exact. Par exemple, le cryptarithme SEND+MORE=MONEY a pour solution S = 9, E = 5, N = 6, D = 7, M = 1, O = 0, R = 8, Y = 2. Dans la suite, nous allons écrire 5 prédicats nous permettant de résoudre les cryptarithmes.

## 1 Listes (4)

Écrire un prédicat qui retourne vrai si la liste en sortie (second paramètre) est la concaténation des listes contenues dans la liste en entrée (premier paramètre).

```
?- append([[a,b],[c,d],[e]],L).  
L = [a, b, c, d, e].
```

## 2 Doublons (4)

Écrire un prédicat qui retourne vrai si la liste en sortie (second paramètre) contient les éléments distincts de la liste en entrée (premier paramètre), ce qui revient à supprimer les doublons.

```
?- list_to_set([a,b,a,b,c],S).  
S = [a, b, c].
```

### 3 Exposants (4)

Écrire un prédicat qui retourne vrai si la liste en sortie (second paramètre) contient la liste décroissante des puissances  $10^i$  pour  $i$  de  $N-1$  à  $0$ ,  $N$  étant un entier en entrée (premier paramètre).

```
?- create_exp(4,L).  
L = [1000, 100, 10, 1].
```

### 4 Ordre supérieur (4)

Écrire un prédicat qui retourne vrai si l'expression en sortie (second paramètre) correspond au nombre représenté par les chiffres de la liste en entrée (premier paramètre).

```
?- create_sum([1,2,3],L).  
L = 123.
```

### 5 Contraintes (4)

Écrire un prédicat qui résout un cryptarithme somme. Vous pourrez utiliser la programmation par contrainte via la librairie `clpfd` vue en cours, quitte à modifier la fonction `create_sum` de l'exercice précédent.

```
?- cryptarithme([O,A,S,I,S]+[S,O,L,E,I,L]=[M,I,R,A,G,E]).  
O = 7,  
A = 3,  
S = 8,  
I = 5,  
L = 6,  
E = 4,  
M = 9,  
R = 0,  
G = 1 ;  
false.
```