

Examen de rattrapage de programmation logique

7 juillet 2017

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

Le barème est indicatif.

Aucun document autorisé.

Machines (ordinateurs, téléphones, montres connectées et calculatrices) interdites.

Aucune question ne pourra être posée durant l'examen. En cas de doute concernant le sujet, vous expliquerez vos hypothèses.

Durée : 2h

1 Mise en jambes /4 pt

Écrire les 2 prédicats suivants :

1. `duplicate` qui retourne vrai si la liste en sortie (deuxième paramètre) correspond à la liste en entrée (premier paramètre) dans laquelle tous les éléments ont été dupliqués.

```
?- duplicate([a,b,c,c,d],X).  
X = [a,a,b,b,c,c,c,c,d,d]
```

2. `duplicate_n` qui retourne vrai si la liste en sortie (troisième paramètre) correspond à la liste en entrée (premier paramètre) dans laquelle tous les éléments ont été dupliqués un nombre entier N de fois (deuxième paramètre).

```
?- duplicate_n([a,b,c],3,X).  
X = [a,a,a,b,b,b,c,c,c]
```

2 Arbres binaires /5 pt

Un arbre binaire est soit vide soit composé d'une racine et de deux sous-arbres qui sont eux-mêmes des arbres binaires. Nous utiliserons l'atome `nil` pour un arbre vide et le terme composé `t(X,L,R)` où X est un noeud et L et R sont les sous-arbres gauche et droit.

1. Écrire un prédicat `istree(T)` qui est vrai si T est un arbre binaire selon la définition ci-dessus.

```
?- istree(t(a,t(b,nil,nil),nil)).  
Yes  
?- istree(t(a,t(b,nil,nil))).  
No
```

2. Un arbre binaire équilibré possède la propriété suivante : le nombre de noeuds dans son sous-arbre gauche et dans son sous-arbre droit sont quasi égaux, c'est-à-dire que leur différence est d'au plus un. Écrire un prédicat `cbal_tree(N,T)` qui est vrai si T retourne par backtracking, tous les arbres binaires équilibrés de taille N. Vous utiliserez l'atome `x` pour modéliser les noeuds.

```
?- cbal_tree(4,T).  
T = t(x, t(x, nil, nil), t(x, nil, t(x, nil, nil))) ;  
T = t(x, t(x, nil, nil), t(x, t(x, nil, nil), nil)) ;  
etc.....No
```

3 Carré magique /5 pt

Un carré magique est une matrice $n * n$ qui contient les nombres de 1 à n^2 , disposés tels que la somme des éléments de chaque ligne, de chaque colonne et des deux diagonales principales donnent le même résultat. Écrire un prédicat qui vérifie qu'une matrice carrée sous forme d'une liste de listes est un carré magique.

```
?- carre_magique([[8,3,4],
                  [1,5,9],
                  [6,7,2]]).
```

true

4 Taxicab number /6 pt

Le n ième nombre taxicab, $taxicab(n)$ est défini comme le plus petit nombre pouvant être exprimé comme une somme de deux cubes consécutifs non nuls de n manières différentes. Écrire un prédicat `taxicab(N,R)` qui retourne vrai si R est le nombre taxicab de N . Il existe un nombre taxicab pour tout entier d'après le théorème de Hardy et Wright (1938).

Par exemple :

- $taxicab(1) = 2$ car $1^3 + 1^3 = 2$

- $taxicab(2) = 1729$ car $1^3 + 12^3 = 9^3 + 10^3 = 1729$

```
?- taxicab(1,R).
```

R=2

```
?- taxicab(2,R).
```

R=1729

Bonus

Quel est le nom de l'inventeur de Prolog disparu le 12 mai dernier ?

