S09 : Prolog (introduction)

Enseignant : Stéphane LE PEUTREC
Assistant : Jonathan LAUPER

Instructions

- Deadline : jeudi suivant à 11:00

1. Prédicats sur les listes

Développez les prédicats suivants (sans utilisez les prédicats prédéfinis équivalents).

- insert(+X, ?L1,?L2) : vrai si L1 et L2 sont des listes triées et L2 est la liste L1 augmentée de l'élément X
  exemple : ?- insert(5,[2,4,7,12],L).
  L=[2,4,5,7,12]
- mergeLists(+L1,+L2,?L3) vrai si L1, L2 et L3 sont triées et L3 est la fusion des listes L1 et L2
  exemple : ?- mergeLists([2,6,9],[1,8,13,23],L).
  L=[1,2,6,8,9,12,23]
- reverseRec(+L1,?L2) : vrai si L2 est la liste L1 renversée à tous les niveaux (y compris les listes internes). Indication : vous pouvez utiliser les prédicats prédéfinis atomic/1 et is_list/1.
  exemple : ?- reverseRec([a,[b,c],d,e,[f,g,h]],L).
  L=[[h,g,h],e,d,[c,b],a]
- myMin(+L, ?X) : vrai si X est le plus petit élément de L
  exemple : ?- myMin([6,4,8,2,5],X).
  X=2

2. Liste d'association

Développez les prédicats suivants :

- createLA(?Keys, ?Values, ?ListAssoc) où Keys, Values sont des listes et ListAssoc est une liste d'association. Une liste d'association est une liste dont les éléments sont des listes à deux éléments modélisant des couples clé-valeur. Le premier élément joue le rôle de clé et le second joue le rôle de valeur.
  Exemple de liste d'association à trois éléments : [[a,34], [r,56], [u,65]]. Le premier élément est composé de la clé a et de la valeur 34, le second élément est composé de la clé r et de la valeur 56, et le troisième élément est composé de la clé u et de la valeur 65.

  ListeAssoc est la liste d'association dont les éléments ont pour clés les éléments de la liste Keys et pour valeur les éléments de la liste Values. La liste d'association est constituée des couples [clés,valeurs] créés par les ième éléments des listes Keys et Values.

  Exemple : createLA([a,r,u,o], [34,56,65], LA) est vrai si LA = [[a,34], [r,56], [u,65]]

- valuesLA(+LA, ?K, ?LV) est vrai si la liste LV est la liste des valeurs associées à la clé K dans la liste d'association LA.
  Exemple : valuesLA([[a,34], [r,56], [u,65], [a,23]], a, LV) est vrai si LV = [34,23]

3. Accumulateurs

Développez une version des prédicats suivants qui appelle un prédicat avec accumulateurs

- sum_list(+ L, ?S) : vrai S est la somme des éléments de L

- reverseRec(+L1,?L2) : vrai si L2 est la liste L1 renversée à tous les niveaux (y compris les listes internes)
- myMin(+L, ?X) : vrai si X est le plus petit élément de L
- fibonacci(+N, ?FN) : vrai si FN est est la valeur de fibonacci de N

4. Tri rapide

- quicksort(L1,L2) : vrai si L2 est la liste L1 triée par ordre croissant. Utilisez le principe du tri rapide.
  Rappel : le tri rapide consiste à former deux sous listes à l'aide d'un pivot (le premier élément de la liste par exemple). La première sous liste contient tous les éléments du reste de la liste qui sont inférieurs au pivot, et la seconde sous liste contient tous ceux supérieurs ou égaux au pivot. La liste triée finale s'obtient en rassemblant la première sous liste triée, le pivot et la deuxième sous-liste triée.
  Ex : on doit trier la liste [7,13,4,2,7,34,1,9], le premier élément (ici 7) joue le rôle de pivot. On extrait du reste de la liste, les deux sous-listes [4,2,1] (contenant les éléments plus petits que 7) et [13,7,34,9] (les éléments plus grands ou égaux à 7). Puis on obtient la liste initiale triée en rassemblant la première sous-liste triée (ici [1,2,4]) le pivot (ici 7) et la seconde sous-liste triée (ici [7,9,13,34]) .