

Représentation de la Connaissance

Répétition 8

14 novembre 2006

Astuce du jour

Lorsqu'on a pas de connaissance experte sur un jeu et qu'on doit construire une fonction d'évaluation, la technique la plus classique consiste à faire jouer par la machine un certain nombre des parties aléatoires pour chacun des coups candidats et d'utiliser le nombre de parties gagnantes comme indicateur de la valeur du coup. Il s'agit en fait d'une application directe des méthodes de monte-carlo.

Exercices d'examen

1. plateau

Définir un prédicat `plateau(A,B)` vrai si et seulement si `B` est la plus longue sous-liste d'éléments identiques de la liste de nombres `A`. On demande que, dans le cas où `B` n'est pas complètement instancié mais que `A` l'est, les différentes instances de `B` telles que `plateau(A,B)` est vrai soient générées.

2. sub

Définir un prédicat `sub(Ls,A,B,Xs)`, vrai si `Xs` est une sous-liste de `Ls` de taille `A` au minimum et `B` au maximum (avec $A \leq B$).

3. appears-n

Ecrire un prédicat `appears-n(N,Es,Ls)` prenant comme arguments un entier strictement positif `N`, une liste d'ensembles d'entiers `Es` et une liste d'entiers `Ls`. Le prédicat est vrai si et seulement si la liste `Ls` contient les éléments qui apparaissent dans exactement `N` ensembles de la liste d'ensembles `Es`. On demande que dans le cas où `Ls` n'est pas complètement instancié mais que les autres arguments le sont, les différentes instances telles que le prédicat est vrai soient renvoyées.

4. Problème du portefeuille

On dispose d'un ensemble de coupures (pièces d'un euro, de 2, billets de 5, 10, 20, 50, 100, 200, 500) et on demande la liste de toutes les manières différentes de payer une certaine somme.

5. partition

Définir un prédicat `partition(E,P)`, où `E` est un ensemble d'éléments, représenté par une liste, et `P` est une liste, vrai si et seulement si `P` est une partition de l'ensemble `E`. L'ordre dans lequel apparaissent les éléments d'une classe d'une partition est celui dans lequel apparaissent ces éléments dans `E`. On demande que, dans le cas où `P` n'est pas complètement instancié mais que `E` l'est, les différentes instances de `P` telles que `partition(E,P)` est vrai soient générées.

6. nextperm

Définir un prédicat `nextperm(P1,P2)`, vrai si `P2` est la permutation qui suit `P1` dans la liste croissante des permutations de `P1`.

`nextperm([1,2,3,4],X) ==> X = [1,2,4,3]`.

`nextperm([3,1,4,2],X) ==> X = [3,2,1,4]`.

7. list2tree

Définir un prédicat `list2tree(Ls,Ts)`, vrai si `Ls` est une liste sans répétitions et `Ts` la liste de tous les arbres qu'on peut obtenir avec des feuilles étiquetées par les éléments de `Ls`. Les arbres de `Ts` n'auront pas d'étiquettes aux noeuds internes, et chaque noeud interne possédera au moins deux fils. L'ordre dans lequel les feuilles apparaissent dans l'arbre (parcours de gauche à droite) doit être le même que l'ordre d'apparition des feuilles dans la liste `Ls`.