Université Paris 7 - Master 1 Informatique - Programmation Logique et Programmation logique avec Contraintes

Examen du 5 Septembre 2005 - Durée : 2 heures

Informations : Tous les documents reliés sont autorisés. Le barème est donné à titre indicatif et peut être modifié.

Exercice 1 (**4 points**)

Ecrire la définition en Prolog d'un prédicat `perm/1` tel que le but `perm(L).` réussit si et seulement si `L` est une liste d'entiers, qui contient (une et une seule fois) tous les entiers dans l'interval $[0, longuer(\mathtt{L}) - 1]$.

Ainsi par exemple:

```
  ?- perm([3,0,2,1]).
yes
   ?- perm([]).
yes
   ?- perm([1,2]).
no
   ?- perm([0,toto]).
no
```

Tous les éventuels prédicats auxiliaires utilisés pour définir `perm` seront définis explicitement.

Exercice 2 (**3 points**)

Considérons les prédicats unaires `neg1`, `neg2` et `neg3` définis par:

```
neg1(G):-!,G,fail.
neg1(G).

neg2(G):-G,!,fail.
neg2(G).

neg3(G):-G,fail,!.
neg3(G).
```

Pour chacun de ces prédicats, dir ce qu'il "calcule" (c'est à dir, indiquer quel est le résultat de l'évaluation du but `neg`j`(G)`, j =1,2,3, en fonction du résultat de l'évaluation du but `G`). Motiver briévement vos réponses.

Exercice 3 (**3 points**)

Définir un prédicat binaire `while` dont la spécification est la suivante: l'évaluation du but `while(C,G)` commence par l'évaluation de `C`. Si `C` echoue, `while(C,G)` réussit. Si `C` réussit, on évalue le but `G` et (si l'évaluation de `G` réussit) on évalue de nouveau `while(C,G)`. (rappel: le prédicat ternaire `(P->Q;R)` implemente la version Prolog du `si P alors Q sinon R`).

Pour lire une séquence d'entiers de l'entréé standard et imprimer en sortie leurs carrés, en s'arretant dès que l'entier donné est nul, on pourrait écrire:

```
carre:-while((read(X),X=\=0),(Y is X*X, print(Y))).
```

Est-ce-que l'évaluation du but `carre` a l'effet escompté, en général?

Simuler l'évaluation du but `carre` sur la séquence d'entrées `3`, `4`.