USTL Licence mention Informatique

UE ELFE - Programmation Logique

Devoir Surveillé

durée 1h30 – documents autorisés

Précisez votre numéro de groupe sur votre copie.

Exercice 1: *Arbre de résolution*

Soit le prédicat

```
insere(?X,?Liste,?ListeResultat)
%% ListeResultat est obtenue par insertion de X à l'une des positions de L
```

codé ainsi :

```
insere(X,Liste,[X|Liste]).
insere(X,[Tete|Liste],[Tete|Reste]) :- insere(X,Liste,Reste).
```

1.1 Avec quelle tête de clause le but `insere(X,[],Y)` s'unifie t-il ?
Donnez l'unificateur et expliquez alors la réponse de PROLOG à ce but.

1.2 Avec quelle tête de clause le but `insere(a,Liste,Resultat)` s'unifie t-il ?.
Décrivez et expliquez la réponse de PROLOG à ce but.

1.3 Dessinez (lisiblement !) l'arbre de résolution du but : `insere(a,[b,c],Res)`.

Exercice 2: *Conjecture de Syracuse*

On considère la fonction suivante :

$$f(x) = \begin{cases} \frac{x}{2} & \text{si x est pair} \\ 3x+1 & \text{si x est impair} \end{cases}$$

La conjecture de Syracuse affirme que si l'on itère suffisamment de fois cette fonction on finit nécessairement par atteindre 1. Cette conjecture est supposée vraie.

Par exemple, $f(20) = 10$, $f(10) = 5$, $f(5) = 16$, $f(16) = 8$, $f(8) = 4$, $f(4) = 2$ et $f(2) = 1$. Il faut donc itérer 7 fois cette fonction à partir de 20 pour atteindre 1.

$$\underbrace{f(\cdots(f(}_{7 \text{ fois}}20)\cdots) = 1$$

et la suite des nombres empruntés est : [10,5,16,8,4,2,1].

2.1 Ecrivez un prédicat `f(+N,-V)` qui unifie `V` avec la valeur de $f(N)$. Vous utiliserez l'opérateur de coupure pour éviter des calculs inutiles si cela est possible.

2.2 Donnez le code d'un prédicat

```
syracuse(+N,-ListeValeurs)
%% N est un entier > 0
%% ListeValeurs est la liste des valeurs obtenues par itérations successives de
%% la fonction f sur N jusqu'à l'obtention de 1
% Exemple d'utilisation
% ?- syracuse(20,L).
% L=[10,5,16,8,4,2,1] ;
% No
```

2.3 Donnez un code pour le prédicat

```
duree_syracuse(+N,-NbPas)
%% N est un entier > 0
%% NbPas est le nombre d'itérations nécessaires pour atteindre 1
%%     par itérations de f sur N.
% Exemple d'utilisation
?- duree_syracuse(20,NbPas).
NbPas = 7;
No
```

Exercice 3: *Complétion automatique*

Le but de cet exercice est de réaliser en PROLOG des outils utiles pour mettre en œuvre un système de complétion automatique.

Dans cet exercice les mots sont représentés par la liste des lettres qui les composent.

3.1 Réalisez en PROLOG le prédicat

```
est_prefixe(?Mot1,?Mot2)
%% Teste si le Mot1 est un prefixe du Mot2
```

3.2 Quelle est la réponse de PROLOG si on lui demande de résoudre le but suivant :

```
est_prefixe(X,[e,l,f,e]).
```

3.3 Le prédicat que l'on va réaliser maintenant admet la spécification suivante

```
complete(+Dictionnaire,+PrefixeAcompleter,-MotComplet)
%% Dictionnaire est une liste de mots, un momt est représenté
%%      par une liste de lettres
%% PrefixeAcompleter est une liste de lettres
%% MotComplet est un mot du dictionnaire s'il en existe qui commence
%%     par PrefixeAcompleter
%% exemple d'utilisation
?- complete([ [a,a,b],[a,a,c],[a,b,c] ] , [a,a] , X).
X=[a,a,b];
X=[a,a,c];
No.
```

Codez le prédicat `complete`

3.4 On souhaite maintenant obtenir seulement le premier mot du dictionnaire qui convient. Comment faire ? Donnez le code correspondant à votre proposition.

3.5 Realisez un prédicat nommé `listeCompletion(+D,+M,-L)` à trois paramètres, le premier est un dictionnaire `D`, le second un mot `M` à compléter, et le troisième est la liste `L` des mots du dictionnaire `D` qui commencent par `M`

Exercice 4: *Longueur des chemins d'un graphe*

Un graphe orienté est représenté par un ensemble d'arcs pondérés par leur longueur. Ces arcs sont donnés par des faits `arc(+X,+Y,+C).` où `X` désigne le sommet de départ de l'arc, `Y` son sommet d'arrivée, et `C` la longueur de l'arc :

```
arc(a,b,5).
arc(b,c,2).
arc(c,d,1).
arc(a,c,3).
```

On souhaite calculer l'ensemble des chemins ainsi que leurs longueurs respectives entre un sommet X et un sommet Y. On utilise por cela le prédicat à quatre paramètres :

```
%% chemin(+X,+Y,-L,-C)
%% X est le sommet de départ
%% Y est le sommet à atteindre
%% L est la liste des sommets à parcourir pour aller de X à Y
%% C est  la longueur du chemin.
% Exemple d'utilisation
?- chemin(a,d,L,C).
L = [a, b, c, d]
C = 8 ;
L = [a, c, d]
C = 4 ;
No
```

4.1 En supposant que le prédicat `chemin` est défini, réalisez un prédicat `chemindemoinsde(+X,+Y,+N,-L,-C)` qui donne à chaque réponse (" ;") un nouveau chemin entre X et Y de longueur inférieure ou égale à N .

4.2 Réalisez le prédicat `chemin(+X,+Y,-L,-C)`