

Université de Rennes 1
IFSIC
Master 1 d'informatique
Programmation, logique et calcul

2ème session 2004-2005
22 juin 2005, 14h-16h

Ce sujet comporte 2 pages.

Tous documents autorisés.

Le barème est indicatif.

Toutes les réponses doivent être justifiées (de façon concise mais précise).

Partie I - Arbres de recherche (8 points)

Soit le programme

$m(E, [E_L])$.

$m(E, [_X|L]) :- m(E, L)$.

$o(P) :- P, !$.

$t(in-N, N)$.

$t(in-N, R) :- R \text{ is } N+N*10$.

$affiche(X, RX) :- format(' valeur : ~p resultat : ~p\n', [X, RX])$.

$go1 :-$

$m(X, [in-1, in-2]),$

$t(X, RX),$

$affiche(X, RX),$

$fail.$

$go1 :- format(' FIN ! ', []).$

$go2 :-$

$m(X, [in-1, in-2]),$

$o(t(X, RX)),$

$affiche(X, RX),$

$fail.$

$go2 :- format(' FIN ! ', []).$

Rappel :

Le prédicat `format` produit un affichage sur la sortie standard et réussit (c-à-d. produit un succès).

Le prédicat `fail` échoue (c-à-d. produit un échec).

Question 1

Dessiner l'arbre de recherche du but suivant : `go1`.

Indiquer clairement pour chaque coupure éventuelle les branches qu'elle coupe.

Faire le bilan de la preuve : indiquer les témoins des preuves si il y en a ; le résultat des affichages sur la sortie standard s'il y en a.

Question 2

Dessiner l'arbre de recherche du but suivant : go2.

Indiquer clairement pour chaque coupure éventuelle les branches qu'elle coupe.

Faire le bilan de la preuve : indiquer les témoins des preuves si il y en a et le résultat des affichages sur la sortie standard s'il y en a.

Partie II - Programmation logique (12 points)

Soit la grammaire suivante :

```
phrase → gr-sujet gr-verbe
gr-sujet → gr-nom gr-prep | gr-nom
gr-nom → article nom gr-prep | article nom
gr-prep → prep gr-nom
article → 'le' | 'la' | 'les' | 'un' | 'une' | 'des'
nom → 'pomme' | 'pommes' | 'pie' | 'pies' | 'queue' | 'queues'
prep → 'de'
gr-verbe → 'tombe' | 'tombent' | 'bouge' | 'bougent'
```

On considère que les symboles entre simples quotes (ex. 'pie') sont des symboles terminaux du langage.

Question 3

Coder cette grammaire en DCG.

On souhaite vérifier l'accord en nombre du sujet et du verbe et l'accord en genre et en nombre de l'article et du nom.

Question 4

Quelle information faut-il ajouter aux règles de la grammaire pour y parvenir ?

Augmenter la grammaire DCG, afin de vérifier ces accords.

On veut représenter la structure syntaxique des phrases reconnues par la grammaire. Pour cela, on découpe une phrase en groupes consécutifs, eux-mêmes découpés en sous-groupes consécutifs, etc, en utilisant des constructeurs pour typer les groupes et sous-groupes. On trouve aux feuilles de cette structure tous les mots de la phrases analysée. Par exemple, la structure syntaxique de «La queue de la pie bouge» est représentée par

```
ph([gs([ gn(['la', 'queue']), gp(['de', gn(['la', 'pie']) ]) ], gv(['bouge']) ])
```

Question 5

Augmenter la grammaire DCG produite en réponse à la question 3, afin que l'analyse d'une phrase retourne un terme PROLOG qui représente la structure syntaxique de la phrase lue.

Question 6

Écrire un prédicat `aplatir/2`, de type `aplatir(+structure, ?phrase)`, qui associe à une représentation de structure syntaxique la liste de ses feuilles.

Quelle relation ce prédicat et le prédicat `phrase/2` de l'environnement de PROLOG entretiennent-ils ?