

TD noté de
Prolog
Durée : 2 h
Tous documents autorisés

Envoyer par m^êl à `pbrisset` votre fichier `.pl` solution où vous incluez des traces d'exécution. Les commentaires sont bienvenus.

1. On considère la grammaire suivante pour décrire des phrases en langage naturel :

```

racine ::= sujet verbe [complement]
sujet ::= groupe_nominal
complement ::= groupe_nominal
groupe_nominal ::= article nom [adjectif]
article ::= un|le
adjectif ::= gros|...
nom ::= chat|...
verbe ::= mange|...

```

- (a) Implémenter cette grammaire à l'aide de DCG en utilisant le fichier `/usr/local/serveur/ECLIPSE/mots` qui définit *adjectif*, *nom* et *verbe*.
 (b) Écrire un prédicat d'affichage d'une liste de mots et utiliser la grammaire en génération.
2. On considère un langage de programmation \mathcal{L} dont la grammaire est la suivante :

```

variable ::= {x, y, z ...}
terme ::= 0
           ::= variable
           ::= terme + 1
programme ::= variable := terme
              ::= programme ; programme
              ::= do variable times programme end

```

où chaque variable peut contenir un entier, `:=` est l'affectation, `;` désigne la séquence, `do x times p end` est évalué en calculant la valeur a de x puis en exécutant a fois le programme p . Les variables du programmes ne sont pas initialisées et on pourra fixer leur valeur au début de l'interprétation.

- (a) Choisir une représentation pour les programmes de ce langage et écrire un interpréteur (on ne s'intéresse pas à l'analyse syntaxique dans cette question). Le résultat de l'interprétation est la valeur de toutes les variables intervenant dans le programme. On pourra interpréter le programme suivant avec x et y initialisés à 5.

```

z := y;
do x times z := z + 1 end;
t := 0;
do x times do y times t := t + 1 end end
pour obtenir quelque chose du genre
?- interpreteur(<le programme>, [{"x", 5}, {"y", 5}], R).
R = [{"t", 25}, {"z", 10}, {"x", 5}, {"y", 5}]

```