Travaux Pratiques en Logique : Prolog (2)

1 Automate

Le but est d'écrire un automate capable de reconnaître un mot d'un langage donné. L'automate constituera la base de fait du programme

ex :
entr(et1).
sort(et3).
existe(et1, a, et2).
existe(et2, b, et3).
existe(et3, a, et3).

Complèter ce programme et définir le prédicat reconnaître(l) où l est une liste de lettres correspondant au mot à reconnaître.

2 Listes

Une liste en Prolog est construite récursivement à l'aide d'un foncteur de la forme : `[_ | _]` la première place (premier underscore) indiquant le premier élément de la liste et la deuxième place la liste des éléments restants : c'est une liste. La liste vide est notée `[]`. En typant ces opérations, on aurait :

```
[] : -> liste
[_ | _] : element liste ->liste
```

Cette écriture peut être allégée en prolog. Ainsi la liste `[1 | [2 | [ 3 | []]]]` peut aussi s'écrire : `[1,2,3]`.
a) Définissez un prédicat donnant la longueur d'une liste.
b) Définissez un prédicat de la forme `estdans(Elem, Liste)` réussissant si l'élément est dans la liste. Donnez des clauses définissant un prédicat permettant de trouver le nème élément d'une liste (et qui échoue si la liste est trop petite).
c) Écrivez un programme prolog permettant de concaténer deux listes.
d) Définir le prédicat Prolog Somele(P,n) qui pour toute liste d'entiers l, renvoie dans n la somme des éléments de l.
ex : Somele([1,2,3,4,5], N).
renvoie N=15
e) Définir le prédicat Prolog compte-aa(l,n) qui pour toute liste d'identificateurs l renvoie dans N le nombre d'occurences de aa dans l.
ex : compte-aa([bb,aa,bb,aa,cc,aa], N).
renvoie N=3