

Feuille de TD 1 — Prolog

Michel Rueher

Les supports de cours, les sujets et les corrigés des TDs, les sujets des examens des années précédentes se trouvent à l'adresse: <http://www.i3s.unice.fr/rueher/Cours/CLP/>

Environnement & Mise en oeuvre

- Documentation en ligne:
<Http://www.gprolog.org/>
- Mise en oeuvre (cf. fichiers test.pl et using_test)

```
cerise:TD1 rueher$ gprolog
GNU Prolog 1.4.4 (64 bits)
Compiled Apr 23 2013, 17:26:17 with /opt/local/bin/gcc-apple-4.2
By Daniel Diaz
Copyright (C) 1999-2013 Daniel Diaz
| ?- [test].   Chargement fichier test.pl
compiling /Users/rueher/Enseignement/1_Cours/PROLOG/EnCours/TD/TD1/test.pl for byte code...
/Users/.../TD1/test.pl compiled, 31 lines read -557 bytes written, 4 ms
yes
| ?- my_append([a,b],[c,d],L).   Question 1
L = [a,b,c,d]
yes

| ?- my_append(L1,L2,[a,b]).

L1 = []
L2 = [a,b] ? ;   Demande d'autres réponses

L1 = [a]
L2 = [b] ? ;

L1 = [a,b]
L2 = [] ? ;

no
| ?- halt.   Quitte Prolog
micmacs-8:TD1 rueher$
```

Manipulation de listes:

Ecrire les prédicats suivants:

- élément/2
1^{er} argument: élément de la liste, 2^{eme} argument: liste.
On pourra réaliser deux versions (avec utilisation du prédicat append/3, sans utilisation du prédicat append/3)
- hors_de/2
1^{er} argument: élément recherché, 2^{eme} argument: liste.
Vrai si l'élément n'appartient pas à la liste

- **appartient/2**
1^{er} argument: liste, 2^{eme} argument: liste.
Vrai si le premier argument est sous liste du second.
- **retrait_elt/3**
1^{er} argument: élément à retirer, 2^{eme} argument: liste initiale, 3^{eme} argument: liste résultat.
Vrai si *liste résultat = liste initiale - élément à retirer*
On pourra réaliser plusieurs versions (avec utilisation du prédicat `append/3`, sans utilisation du prédicat `append/3`, retrait de la première instance de l'élément à retirer, retrait de toutes les instances)
- **permuter/2**
1^{er} argument: liste , 2^{eme} argument: liste
La deuxième liste est une permutation quelconque de la première.
- **renverser/2**
1^{er} argument: liste 2^{eme} argument: liste .
La deuxième liste doit correspondre à la première "renversée". On pourra réaliser deux versions (avec utilisation du prédicat `append/3`, en utilisant une liste auxiliaire).
- **palindrome/1**
Utiliser le prédicat `renverser/2` pour vérifier si une liste de lettres correspond à un palindrome.

La classique famille

Reprenre le programme de la `famille` vu en cours (transparent n° 16) et définir les prédicats

- `ascendant(Personne,Ascendants)`,
- `descendant(Personne,Descendants)`.

Le Zèbre

Ecrire un prédicat `zebre(ProprietaireZebre,BuveurEau)` qui utilise les assertions ci-dessous pour identifier le propriétaire du zebre et le buveur d'eau.

Enigme :

Cinq maisons consécutives, de couleurs différentes, sont habitées par des hommes de différentes nationalités. Ils possèdent tous un animal différent, ont chacun une boisson préférée différente et fument des cigarettes différentes.

On sait que :

1. *Le norvégien habite la première maison,*
2. *La maison à coté de celle du norvégien est bleue,*
3. *L'habitant de la troisième maison boit du lait,*
4. *L'anglais habite la maison rouge,*
5. *L'habitant de la maison verte boit du café,*
6. *L'habitant de la maison jaune fume des kool*
7. *La maison blanche se trouve juste après la verte,*
8. *L'espagnol a un chien,*
9. *L'ukrainien boit du thé,*
10. *Le japonais fume des craven*
11. *Le fumeur de old gold a un escargot,*
12. *Le fumeur de gitane boit du vin,*
13. *Un voisin du fumeur de chesterfield a un renard,*
14. *Un voisin du fumeur de kool a un cheval.*