T. P. Programmation par contraintes.
POLYTECH. 4ième année. Alternance
UNIVERSITÉ D'AIX-MARSEILLE
Année universitaire 2018/2019

Enseignant : Odile Papini

Feuille de T. P. 2 : Listes en PROLOG avec Gnu Prolog

Utilisation de Gnu Prolog :

- Le site web de Gnu Prolog : *http* : *//www.gprolog.org/*
- La documentation de Gnu Prolog : *http* : *//www.gprolog.org/*♯*manual*

Pour utiliser Gnu Prolog :

- création (avec un éditeur quelconque) d'un fichier d'extension .pl qui contient le programme.
- compilation (2 manières possibles) :
  - gplc nom_prog.pl
  - gplc -o nom_prog_exe nom_prog.pl

  Dans le premier cas, l'exécutable a le même nom que le programme source, dans le deuxième cas, il peut avoir un nom différent.
- execution
  - ./ nom_prog_exe

  affichage d'une nouvelle invite |?-
  il est alors possible d'interroger le programme
- quelques commandes : **a** : abort, **c** : continue, **d** : debug, **h** : help, **b** : break, **e** : exit, **t** : trace.

Exercice 1 **:**

Manipultation de Gnu Prolog. Créer un fichier .pl pour chacune des manipulations des listes vues en cours :

- parcours : qui permet d'afficher les éléments d'une liste l.
- meme_longueur : qui permet de tester si deux listes ont le même nombre d'éléments.
- rang_pair : qui construit la liste des éléments de rang pair d'une liste contenant un nombre pair d'éléments.
- renserver : qui construit la liste des éléments dans l'ordre inverse.

Exercice 2 **:** Une liste l qui contient un nombre pair d'éléments est donnée.

Ecrire un programme logique qui construit la liste m des éléments de rang impair de l.

1) dans l'ordre d'apparition dans l.
2) en inversant cet ordre.
3) reprendre cet exercice dans le cas où l contient un nombre impair d'éléments.

Exercice 3 **:**

- Ecrire les règles qui définissent element-de(x,l) exprimant qu'un élément x appartient à la liste l.
- Ecrire les règles qui définissent hors-de(x,l) exprimant qu'un élément x n'appartient pas à la liste l.

Exercice 4 **:** Ecrire un programme logique longueur qui permet de calculer le nombre déléments d'une liste.

Exercice 5 **:** Ecrire un programme logique facto qui calcule factorielle n, $n \geq 0$.

(facultatif) Exercice 6 **:**

Deux listes u et v ont le même nombre d'éléments. Ecrire un programme fusionner(u,v,l) qui prend successivement un élément de u et de v pour construire l. En déduire un calcul direct des listes des éléments de rang pair et de rang impair d'une liste l donnée.

(facultatif) Exercice 7 **:**

Ecrire les règles qui définissent differents(l) exprimant que la liste l ne contient pas de répétition.

(facultatif) Exercice 8 **:**

Ecrire les règles qui définissent debut(m,l) exprimant qu'une liste m débute une autre liste l. Que se passe-t-il si m est une variable lors du lancement ?

(facultatif) Exercice 9 **:**

Ecrire les règles qui définissent contenue(m,l) exprimant que tous les éléments de m sont dans l (dans un ordre quelconque).

(facultatif) Exercice 10 **:**

Ecrire les règles qui permettent de retirer un élément x d'une liste l.

(facultatif) Exercice 11 **:**

Dans cet exercice les listes représentent des ensembles. Il n'y a donc pas d'élement répété.

- Ecrire les règles qui définissent intersection(u,v,l) où l est l'intersection de u et de v.
- Ecrire les règles qui définissent reunion(u,v,l) où l est la reunion de u et de v.
- écrire les règles intersection et réunion en utilisant la coupure.
- écrire les règles intersection et réunion sans utiliser la coupure.