T. P. Logique.
Polytech'Marseille. 3ième année
UNIVERSITÉ D'AIX-MARSEILLE
Année universitaire 2017/2018

Enseignants : Motasem Nawaf
& Odile Papini

Feuille de T. P. 3 : Les listes en Prolog

Utilisation de SWI-Prolog:

- SWI-Prolog est installé sur votre compte sous Linux. Pour une utilisation sur votre machine personnelle, URL : **http://www.swi-prolog.org** version 6.6.6.
- Documentation, URL : **http://www.swi-prolog.org/pldoc**
- Pour lancer SWI-Prolog:
  - se connecter sous Linux
  - ouvrir un terminal, pour démarrer l'interpréteur taper :

    ```
    prolog
    ```

  - avec un éditeur créer un programme avec une extension .pl (par exemple : nomfic.pl)
  - pour charger le fichier dans l'interpréteur (ne pas oublier le point final):

    ```
    ['nomfic'].
    ```

- saisir la liste de buts séparés par des virgules et terminée par un point.

Exercice 1 **:**
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.
- tester les programmes avec SWI Prolog.

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.