Conception et implémentation

IFT-17586 Intelligence artificielle I - A2002

Séance 5

- Les listes et la récursivité

Les prédicats pour gérer les listes

- membre/2 : « est-ce qu'un élément E appartient à une liste L ? »
- E appartient à L
  - si E=premier élément de L ou
  - si E appartient au reste de L

```
membre(E, [E|L]).
membre(E, [Y|Reste]) :- membre(E, Reste).
```


Les prédicats pour gérer les listes

- longueur/2 : « calculer la longueur d'une liste (0 à n éléments) »
- La longueur est égale à :
  - 0 si L=[] ou
  - 1 si L=[E1] ou ⟷ longueur([]) + 1
  - 2 si L=[E2, E1] ou ⟷ longueur([E1]) + 1
  - 3 si L=[E3, E2, E1] ou ⟷ longueur([E2,E1]) + 1
  - N si L=[En|Reste] ⟷ longueur(Reste) + 1

```
longueur([], 0).
longueur([E|Reste],N) :- longueur(Reste,V), N is V+1.
```


Autre prédicat récursif : concaténer/3

- Construire une liste L3 à partir de L1 et L2
- L3 est égale à :
  - L2 si L1=[] ou
  - [E1|L2] si L1=[E1] ou
  - [E2|[E1|L2]] si L1=[E2, E1] ou
  - [E3|[E2,E1|L2]] si L1=[E3, E2, E1] ou
  - [En|[En-1, ..., E1|L2]] si L1=[En, ..., E1|Reste]

```
concaténer([], L2, L2).
concaténer([E|R1], L2, [E|Reste]) :-
                        concaténer(R1, L2, Reste).
```


Autre prédicat récursif : enlever/3

- Enlever un élément X d'une liste L
- NL est égale à :
  - Reste si L=[X|Reste] ou
  - [E1|Reste] si L=[E1, X|Reste] ou
  - [E2|[E1|Reste]] si L=[E2, E1, X|Reste] ou
  - [En|[En-1, ..., E1|Reste]] si L=[En, ..., E1, X|Reste]

```
enlever(X, [X|Reste], Reste).
enlever(X, [E|Reste], [E|Résultat]) :-
                    X\==E, enlever(X, Reste, Résultat).
```


Exercices : programmer les prédicats suivants en Prolog

- enlever_tous(X,L,R) : enlever toutes les occurrences de X dans L (résultat = R)
- intersection(L1,L2,I) : faire l'intersection des listes L1 et L2 (résultat = I)
- union(L1,L2,U) : faire l'intersection des listes L1 et L2 (résultat = U)
- position(L,X,N) : donner la première position de X dans la liste L (N=0 si X $\notin$ L)
- dernier(L,X) : X est le dernier élément de la liste L
- milieu(L,X) : X est l'élément en position médiane dans la liste L

Exercices : programmer les prédicats suivants en Prolog

- palindrome(P) : P est-il un palindrome ?
- remplacer(L,X,Y,R) : remplacer la première occurrence de X par Y dans L (résultat = R)
- remplacer_tous(L,X,Y,R) : remplacer toutes les occurrences de X par Y dans L (résultat = R)
- séparer(L,X,LAvant,LApres) : séparer la liste L en deux sous-listes (LAvant = sous-liste de L précédant la première occurrence de X et LApres = sous-liste qui suit cette occurrence)
- somme(L,S) : calculer la somme des éléments de la liste L (résultat = S)

