

## Intelligence Artificielle TP2

### 1 Tests de l'unification

Indiquez les réponses aux requêtes suivantes en expliquant les réponses ou les réactions de prolog.

- ?- f(X,a) = f(b,Y).
- ?- f(X,a) = f(b,\_).
- ?- f(X,a) = f(Y,Z).
- ?- f(X,a) = f(X).
- ?- f(X,Y,Z) = f(g(Y),h(Z),i(U)).
- ?- X = a, X = b.
- ?- X = f(Y,Z), Y = g(U, a), U = f(b).
- ?- X = f(Y,Z), Y = g(U, a), U = h(X).
- ?- a \= a.
- ?- 'a' \= a.
- ?- A \= a.
- ?- f(a) \= a.
- ?- f(a) \= A.
- ?- f(A) \= f(a).
- ?- g(a,B,c) \= g(A,b,C).
- ?- g(a,b,c) \= g(A,C).
- ?- f(X) \= X.
- ?- 2 + 2 = 4.
- ?- 2 + 2 = +(2,2).
- ?- 2 + 2 = +(X,Y).
- ?- +(X,Y)='+'(U,V).
- ?- X = 2+2,Y is X.
- ?- X = 2+2,X is X.
- ?- [a, [b, c], d] = [X, Y, Z].
- ?- [a, [b, c], d] = [a, b | L].
- ?- [a, b, c, d | L1] = [a, b | L2].

### 1 Structure des listes

Une liste est une structure de données importante. Elle correspond à une suite finie d'objets, de taille quelconque, et elle est représentée entre crochets, les éléments étant séparés par des virgules : [1, 2, 3, 4] pour une liste d'entiers, [a, b, c] pour une liste de caractères, [] pour la liste vide.

Remarque : un élément d'une liste peut être un terme quelconque, par exemple une liste (voir les exemples ci-dessous).

La structure de liste peut être définie récursivement par

- la liste vide, représentée par [].
- si T est un terme et L une liste, [T | L] représente la liste de premier élément T (tête de liste), suivi de L (queue de liste). L'opérateur | permet de construire les listes.

#### 1.1 Représentations de listes

Représenter par un arbre la liste Prolog [a,b,c].

#### 1.2 prédicat prédéfinis

- ?-append( [a, b], [c, d],L).
- ?-reverse( [a, b, c, d],L).
- ?-member( b,[a, b, c, d]).

### 2. Construction

a) Écrire le prédicat premier/2 : premier(X,L) : X est le premier élément de la liste L.

- ?- premier(a,[a,b,c]).

true.

?- premier(X,[a,b,c]).

X = a.

Que donne les requêtes : premier(a,L) et premier(X,L) ?

b) Écrire le prédicat dernier/2 : dernier(X,L) : X est le dernier élément de la liste L.

?-dernier(X,[a,b,c]).

X = c ;

false.

Que donne la requête : dernier(X,L).

c) Ecrire le prédicat unique/2 : unique(X,L) : X n'apparaît qu'une seule fois dans la liste L.

?- unique(b,[a,b,c,a]).

true ;

false.

?- unique(X,[a,b,c,a]).

X = b ;

X = c ;

false.

**Donner l'arbre d'exécution de unique (X,[a,b,a,a]).**