

Insertion dans une liste

?- insert(a, L, [b, a, d, a, f]).

L = [b, d, a, f] ;

L = [b, a, d, f] ;

no

L'insertion et le retrait sont 2 concepts complémentaires!

Retrait dans une liste

notre-delete(R,[R|L],L).

notre-delete(R,[X|LL], [X|L]) :- notre-delete(R,LL,L).

Retrait dans une liste

```
deleteall(X,[],[]).  
deleteall(X,[X|T],Result) :-  
    deleteall(X,T,Result),!.  
deleteall(X,[H|T],[H|Result]) :-  
    deleteall(X,T,Result).
```

```
?- deleteall(2,[1,2,4,3,2,6,2,2],L).  
L = [1, 4, 3, 6].
```

Qu'arrive-t-il si on retire la coupe?

Intersection entre listes

```
intersection( [], Ys, [] ).
```

```
intersection( [ X | Xs ], Ys, Zs ) :-  
    not member( X, Ys ),  
    intersection( Xs, Ys, Zs ).
```

```
intersection( [ X | Xs ], Ys, [ X | Zs ] ) :-  
    member( X, Ys ),  
    intersection( Xs, Ys, Zs ).
```

Tri d'une liste

```
tri([],[]).
tri([P|Q],T) :- partition(P,Q,G,D),
                tri(G,GG), tri(D,DD),
                append(GG,[P|DD],T).
partition(P,[X|L],[X|PG],PD) :- X < P,
                                partition(P,L,PG,PD).
partition(P,[X|L],PG,[X|PD]) :- X >= P,
                                partition(P,L,PG,PD).
partition(P,[],[],[]).
```

Opérations répétitives

- ▶ Effectuer un traitement sur les éléments de listes

```
traite-liste([],[]).
```

```
traite-liste([X|L],[Y|T]) :- traite(X,Y), traite-liste(L,T).
```

```
somme(L,S) :- somme(L,0,S).
```

```
somme([X|L],T,S) :- TT is T+X, somme(L,TT,S).
```

```
somme([],S,S).
```

Inversion d'une liste (double récursion)

```
mirror([ ], [ ]).  
  mirror([X|L1], L2) :-  
mirror(L1,L3),  
  append(L3, [X], L2). % append will dig into the list a second time
```

Inversion d'une liste (avec accumulateur)

```
mirror2(Left, Right) :-
```

```
    invert(Left, [ ], Right).
```

```
invert([X|L1], L2, L3) :-      % the list is 'poured'
```

```
    invert(L1, [X|L2], L3). % into the second argument
```

```
invert([ ], L, L). % at the deepest level, the result L is merely copied
```


Représentation des Listes

- ▶ Les listes peuvent être représentée avec le symbole fonctionnel binaire « . »
 - suite $\{e1, e2, \dots\} \Rightarrow$ liste $(e1.(e2.(...)))$
- ▶ La liste vide est notée « nil ». Elle sert souvent à marquer la fin de liste.
- ▶ Exemples :
 - suite des variables X et Y $\Rightarrow (X.Y)$
 - suite {gateau, fruit, glace}
 $\Rightarrow (\text{gateau} . (\text{fruit} . (\text{glace} . \text{nil})))$