

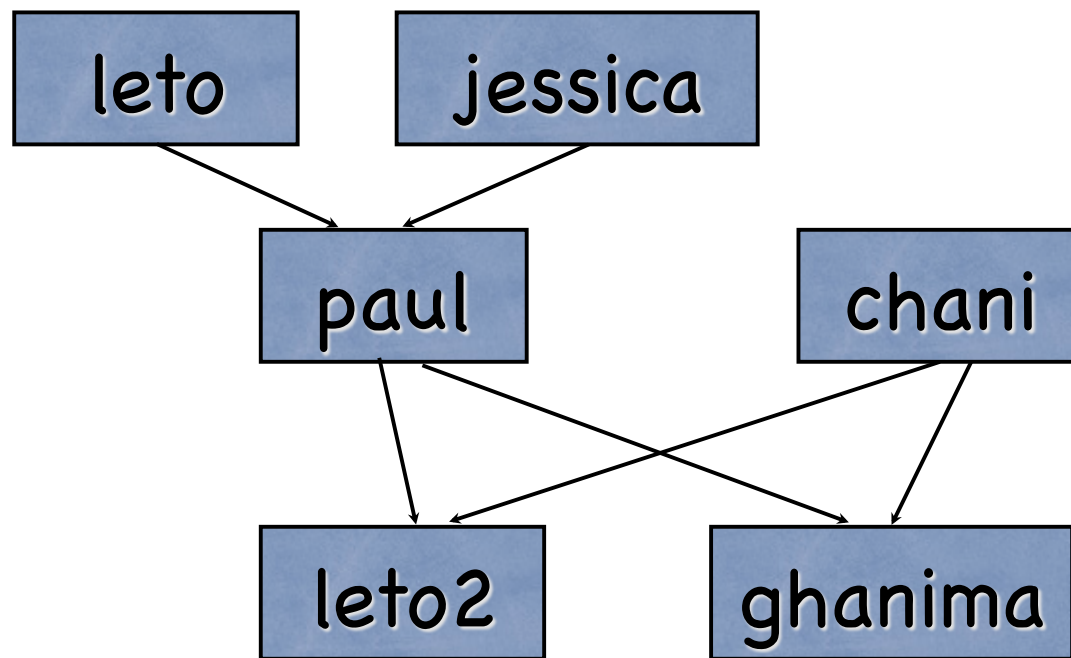
BD avancées ENSEIRB
PROLOG
Construction de base (suite)

- <http://www.labri.fr/perso/moot/SE/>
- <http://www.labri.fr/perso/retore/SE/>

Familles

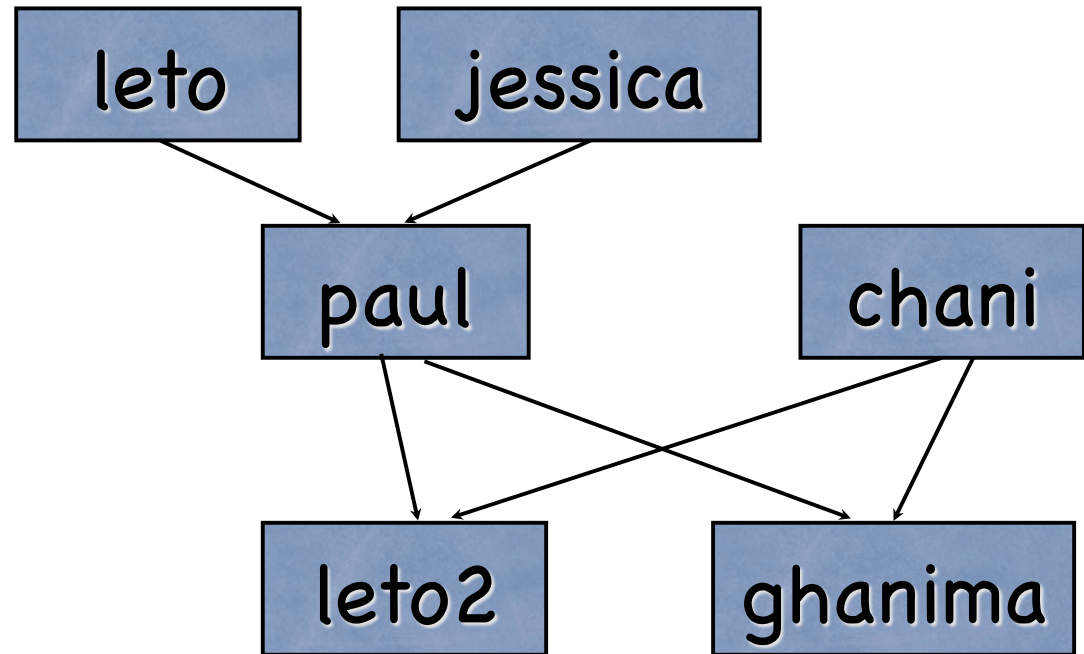
- 🗣️ Il est presque obligatoire de parler des familles dans un premier cours de Prolog.
- 🗣️ Donc, parlons-en un peu....

Familles



Familles

```
parent(leto, paul).  
parent(jessica, paul).  
parent(paul, leto2).  
parent(paul, ghanima).  
parent(chani, leto2).  
parent(chani, ghanima).
```



Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

pere(Parent, Enfant) :-
 parent(Parent, Enfant),
 homme(Parent).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

mere(Parent, Enfant) :-
 parent(Parent, Enfant),
 femme(Parent).

Familles

pere(leto, paul).
pere(paul, leto2).
pere(paul, ghanima).

mere(jessica, paul).
mere(chani, leto2).
mere(chani, ghanima).

On peut choisir les prédicats de base qui nous conviennent. Alors, on pourrait prendre pere/2 et mere/2 comme prédicats de base et donner des règles pour parent/2.

parent(Parent, Enfant) :-
 pere(Parent, Enfant).

parent(Parent, Enfant) :-
 mere(Parent, Enfant).

Familles

pere(leto, paul).
pere(paul, leto2).
pere(paul, ghanima).

mere(jessica, paul).
mere(chani, leto2).
mere(chani, ghanima).

Remarque: on a perdu de l'information ! homme/1 et femme/1 ne sont pas totalement définissable en termes de pere/2 et mere/2.
Pourquoi?

parent(Parent, Enfant) :-
 pere(Parent, Enfant).

parent(Parent, Enfant) :-
 mere(Parent, Enfant).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

GrandParent = leto,
PetitEnfant = Enfant

?- grandpere(leto, PetitEnfant).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(leto, PetitEnfant) :-
 parent(leto, Parent),
 parent(Parent, PetitEnfant),
 homme(leto).

GrandParent = leto,
PetitEnfant = Enfant

?- parent(leto, Parent), parent(Parent, PetitEnfant), homme(leto).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(leto, PetitEnfant) :-
 parent(leto, Parent),
 parent(Parent, PetitEnfant),
 homme(leto).

GrandParent = leto,
PetitEnfant = Enfant

?- parent(leto, Parent), parent(Parent, PetitEnfant), homme(leto).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-

parent(GrandParent, Parent), Parent = paul
parent(Parent, Enfant),
homme(GrandParent).

?- parent(leto, paul), parent(paul, PetitEnfant), homme(leto).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

PetitEnfant = leto2

?- parent(paul, PetitEnfant), homme(leto).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

PetitEnfant = leto2

?- parent(paul, leto2), homme(leto).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

Solution: PetitEnfant = leto2

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

“;” demande à Prolog
de trouver d’autres
solutions

Solution: PetitEnfant = leto2

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

PetitEnfant = ghanima

?- parent(paul, PetitEnfant), homme(leto).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

PetitEnfant = ghanima

?- parent(paul, ghanima), homme(leto).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

PetitEnfant = ghanima

?- homme(leto).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

PetitEnfant = ghanima

Solution: PetitEnfant = ghanima

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

“;” une deuxième fois
ne donne pas d’autres
solutions.

Solution: PetitEnfant = ghanima

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

“;” une deuxième fois
ne donne pas d’autres
solutions.

?- parent(paul, PetitEnfant), homme(leto).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

grandpere(GrandParent, Enfant) :-
 parent(GrandParent, Parent),
 parent(Parent, Enfant),
 homme(GrandParent).

“;” une deuxième fois
ne donne pas d’autres
solutions.

?- parent(leto, Parent), parent(Parent, PetitEnfant), homme
(leto).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

ancetre(Ancetre, Descendant) :-
 parent(Ancetre, Descendant).
ancetre(Ancetre, Descendant) :-
 parent(Parent, Descendant),
 ancetre(Ancetre, Parent).

Familles

parent(leto, paul).
parent(jessica, paul).
parent(paul, leto2).
parent(paul, ghanima).
parent(chani, leto2).
parent(chani, ghanima).

homme(leto).
homme(paul).
homme(leto2).

femme(jessica).
femme(chani).
femme(ghanima).

ancetre(Ancetre, Descendant) :-
 parent(Ancetre, Descendant).
ancetre(Ancetre, Descendant) :-
 ancetre(Ancetre1, Descendant),
 ancetre(Ancetre, Ancetre1).

Familles

?- ancetre(A, jean).

ancetre(Ancetre, Descendant) :-
 parent(Ancetre, Descendant).

ancetre(Ancetre, Descendant) :-
 ancetre(Ancetre1, Descendant),
 ancetre(Ancetre, Ancetre1).

Familles

?– parent(A, jean).

échec (car on ne sait rien de jean).

ancetre(Ancetre, Descendant) :-

parent(Ancetre, Descendant).

ancetre(Ancetre, Descendant) :-

ancetre(Ancetre1, Descendant),

ancetre(Ancetre, Ancetre1).

Familles

?– ancetre(A1,A), ancetre(A, jean).

on revient sur ancêtre et on essaie la deuxième (et dernière) clause

ancetre(Ancetre, Descendant) :-
 parent(Ancetre, Descendant).

ancetre(Ancetre, Descendant) :-
 ancetre(Ancetre1, Descendant),
 ancetre(Ancetre, Ancetre1).

Familles

?– ancetre(A2,A1), ancetre(A1,A),
ancetre(A, jean).

... et on peut continuer comme ça

ancetre(Ancetre, Descendant) :-
parent(Ancetre, Descendant).

ancetre(Ancetre, Descendant) :-
ancetre(Ancetre1, Descendant),
ancetre(Ancetre, Ancetre1).

Familles

?– ancetre(A3,A2), ancetre(A2,A1),
ancetre(A1,A), ancetre(A, jean).

... et on peut continuer comme ça

Conclusion: une bonne définition en logique ne
donne pas nécessairement un bon programme

ancetre(Ancetre, Descendant) :-
parent(Ancetre, Descendant).

ancetre(Ancetre, Descendant) :-
ancetre(Ancetre1, Descendant),
ancetre(Ancetre, Ancetre1).

Arithmétique

- ④ On peut faire l'arithmétique avec des termes en Prolog de façon purement logique. Néanmoins, il est utile de pouvoir faire du calcul arithmétique directement.
- ④ Prolog fournit le prédicat “is” à cet effet.

Arithmétique

🌀 Par exemple, on peut utiliser le prédicat:

$$X \text{ is } X0 + 1$$

🌀 Erreur si on ne connaît pas la valeur de $X0$

🌀 X est une variable.

Arithmétique

- 🌀 Quel est le sens de l'expression:
$$X \text{ is } X + 1$$
- 🌀 On connaît la valeur pour toute variable à droite de "is".
- 🌀 Supposons par exemple que la valeur de X dans le programme soit 2.
L'expression $X+1$ s'évalue alors comme 3.
- 🌀 On finit par $2 \text{ is } 3$ ce qui correspond à faux !

Arithmétique

- ④ $X \text{ is } Y + Z$
- ④ X est une variable libre, dont on sait pas encore la valeur.
- ④ Y et Z sont des variable déjà liées, dont on sait les valeurs. Prolog donne une erreur si cette condition n'est pas respecté pendant la résolution du programme.

Arithmétique

☞ $X \text{ is } Y + Z$

☞ rappel que ceci est une façon d'écrire
 $\text{is}(X, +(Y, Z))$

☞ Etant donnée que des expressions avec “is” n’ont un sens que quand on connaît les valeurs des variables à droite de “is”, cette opération n’est pas purement logique.

Arithmétique

🕒 Opérations de comparaison entre expressions arithmétique. Ces opérations ont un sens que quand on connaît X et Y.

🕒 $X < Y, X \leq Y,$

🕒 $X > Y, X \geq Y,$

🕒 $X ::= Y, X \neq Y$

Arithmétique

$\% = \max(X, Y, Z)$

$\%$ vrai si Z est le maximum de X et Y .

$\max(X, Y, X) :-$

$X \geq Y.$

$\max(X, Y, Y) :-$

$Y > X.$

Arithmétique

% = factoriel(X, F)

% vrai si F est le factoriel de X.

factoriel(0, 1).

factoriel(N0, F) :-

 N0 > 0,

 N is N0 - 1,

 factoriel(N, F0),

 F is F0 * N0.

Listes


- Comme en programmation fonctionnelle la liste est la principale structure de données.
 - [] désigne la liste vide:
 - [H|T] désigne une liste non vide dont le premier élément est H et dont le reste est la **LISTE** T.

Listes


 Liste

 []

 [1 | []]

 [1 | [2 | []]]


 [1 | [2 | [3 | []]]]


 [1 | [2 | [3 | [4 | []]]]]


 Version simple

 []

 [1]

 [1,2]

 [1,2,3]

 [1,2,3,4]

Listes

% = est_liste(Terme)	[2,3]
% vrai si Terme est un liste	[2 []]
	[1 2]
est_liste([]).	[1,2 3]
est_liste([_ L]) :-	[1,2 [3]]
est_liste(L).	

Listes

% = est_liste(Terme)
% vrai si Terme est un liste

est_liste([]).
est_liste([_|L]) :-
 est_liste(L).

$[2,3] = [2|[3|[]]]$

$[2|[]] = [2]$

~~$[1|2]$~~

~~$[1,2|3]$~~

$[1,2|[3]] =$

$[1,2,3] =$

$[1|[2|[3|[]]]]$

Listes

% = membre(Element, Liste)

%

% vrai si Liste contient Element.

membre(X, [X|_]).

membre(X, [_|Ys]) :-

 membre(X, Ys).

Listes

% = append(Liste1, Liste2, Liste3)

%

% vrai si Liste3 contient les éléments de Liste1

% suivi par les éléments de Liste2, c'est-à-dire

% Liste3 est la concatenation de Liste1 et

% Liste2

append([], Ys, Ys).

append([X|Xs], Ys, [X|Zs]) :-

append(Xs, Ys, Zs).

Listes

% = maximum(Liste, Max)

% vrai si Max est la valeur du plus grand

% element de Liste.

maximum([Max], Max).

maximum([H|T], Max) :-

 maximum(T, MO),

 max(MO, H, Max).

Listes

% = maximum(Liste, Max)

% vrai si Max est la valeur du plus grand

% element de Liste.

maximum([Max], Max).

maximum([H|T], Max) :-

 maximum(T, MO),

 max(MO, H, Max).

?- maximum([1,2,3],M).

Listes

% = maximum(Liste, Max)

% vrai si Max est la valeur du plus grand

% element de Liste.

maximum([Max], Max).

H = 1

maximum([H|T], Max) :-

T = [2,3]

maximum(T, MO),

M = Max

max(MO, H, Max).

?- maximum(T, MO), max(MO, H, Max).

Listes

% = maximum(Liste, Max)

% vrai si Max est la valeur du plus grand

% element de Liste.

maximum([Max], Max).

H = 1

maximum([H|T], Max) :-

T = [2,3]

maximum(T, MO),

M = Max

max(MO, H, Max).

?- maximum([2,3], MO), max(MO, 1, M).

Listes

% = maximum(Liste, Max)

% vrai si Max est la valeur du plus grand

% element de Liste.

maximum([Max], Max).

H = 2

maximum([H|T], Max) :-

T = [3]

maximum(T, MO),

MO = Max

max(MO, H, Max).

?- maximum(T, MO), max(MO, H, Max),

max(MO, 1, M).

Listes

% = maximum(Liste, Max)

% vrai si Max est la valeur du plus grand

% element de Liste.

maximum([Max], Max).

H = 2

maximum([H|T], Max) :-

T = [3]

maximum(T, MO),

MO = Max

max(MO, H, Max).

?- maximum([3], M1), max(M1, 2, MO),

max(MO, 1, M).

Listes

% = maximum(Liste, Max)

% vrai si Max est la valeur du plus grand

% element de Liste.

maximum([Max], Max).

maximum([H|T], Max) :-

 maximum(T, MO),

 max(MO, H, Max).

?- maximum([3], 3), max(3, 2, MO),

max(MO, 1, M).

Listes

% = maximum(Liste, Max)

% vrai si Max est la valeur du plus grand

% element de Liste.

maximum([Max], Max).

maximum([H|T], Max) :-

 maximum(T, MO),

 max(MO, H, Max).

?- max(3, 2, MO), max(MO, 1, M).

Listes

% = maximum(Liste, Max)

% vrai si Max est la valeur du plus grand

% element de Liste.

maximum([Max], Max).

maximum([H|T], Max) :-

 maximum(T, M0),

 max(M0, H, Max).

M = 3

?- max(3, 1, M).

Listes

% = maximum(Liste, Max)

% vrai si Max est la valeur du plus grand

% element de Liste.

maximum([Max], Max).

maximum([H|T], Max) :-

maximum(T, MO),

max(MO, H, Max).

M = 3

Listes

% = maximum, version deux

```
maximum([H|T], Max) :-  
    maximum(T, H, Max).
```

```
maximum([], Max, Max).  
maximum([H|T], Max0, Max) :-  
    max(H, Max0, Max1),  
    maximum(T, Max1, Max).
```


Listes

maximum([H|T], Max) :-
 maximum(T, H, Max).

maximum([], Max, Max).
maximum([H|T], Max0, Max) :-
 max(H, Max0, Max1),
 maximum(T, Max1, Max).

H = 1
T = [2,3]
M = Max

?- maximum(T, H, Max).

Listes

```
maximum([H|T], Max) :-  
    maximum(T, H, Max).
```

```
maximum([], Max, Max).  
maximum([H|T], Max0, Max) :-  
    max(H, Max0, Max1),  
    maximum(T, Max1, Max).
```

H = 1
T = [2,3]
M = Max

```
?- maximum([2,3], 1, M).
```

Listes

maximum([H|T], Max) :-
 maximum(T, H, Max).

maximum([], Max, Max).

maximum([H|T], Max0, Max) :-
 max(H, Max0, Max1),
 maximum(T, Max1, Max).

H = 2

T = [3]

Max0 = 1

M = Max

?- max(H, Max0, Max1), maximum(T, Max1, Max).

Listes

maximum([H|T], Max) :-
 maximum(T, H, Max).

maximum([], Max, Max).

maximum([H|T], Max0, Max) :-
 max(H, Max0, Max1),
 maximum(T, Max1, Max).

H = 2

T = [3]

Max0 = 1

M = Max

?- max(2, 1, Max1), maximum([3], Max1, M).

Listes

maximum([H|T], Max) :-
 maximum(T, H, Max).

maximum([], Max, Max).

maximum([H|T], Max0, Max) :-
 max(H, Max0, Max1),
 maximum(T, Max1, Max).

H = 2

T = [3]

Max0 = 1

M = Max

?- max(2, 1, 2), maximum([3], 2, M).

Listes

maximum([H|T], Max) :-
 maximum(T, H, Max).

maximum([], Max, Max).
maximum([H|T], Max0, Max) :-
 max(H, Max0, Max1),
 maximum(T, Max1, Max).

?- maximum([3], 2, M).

Listes

maximum([H|T], Max) :-
 maximum(T, H, Max).

maximum([], Max, Max).

maximum([H|T], Max0, Max) :-
 max(H, Max0, Max1),
 maximum(T, Max1, Max).

H = 3

T = []

Max0 = 2

Max = M

?- max(H, Max0, Max1), maximum(T, Max1, Max).

Listes

maximum([H|T], Max) :-
 maximum(T, H, Max).

maximum([], Max, Max).

maximum([H|T], Max0, Max) :-
 max(H, Max0, Max1),
 maximum(T, Max1, Max).

H = 3

T = []

Max0 = 2

Max = M

?- max(3, 2, Max1), maximum([], Max1, M).

Listes

maximum([H|T], Max) :-
 maximum(T, H, Max).

maximum([], Max, Max).

maximum([H|T], Max0, Max) :-
 max(H, Max0, Max1),
 maximum(T, Max1, Max).

H = 3

T = []

Max0 = 2

Max = M

?- max(3, 2, 3), maximum([], 3, M).

Listes

maximum([H|T], Max) :-
 maximum(T, H, Max).

maximum([], Max, Max).

Max = M = 3

maximum([H|T], Max0, Max) :-
 max(H, Max0, Max1),
 maximum(T, Max1, Max).

?- maximum([], 3, 3).

Listes

maximum([H|T], Max) :-
 maximum(T, H, Max).

maximum([], Max, Max).

Max = M = 3

maximum([H|T], Max0, Max) :-
 max(H, Max0, Max1),
 maximum(T, Max1, Max).