

Programmation Logique

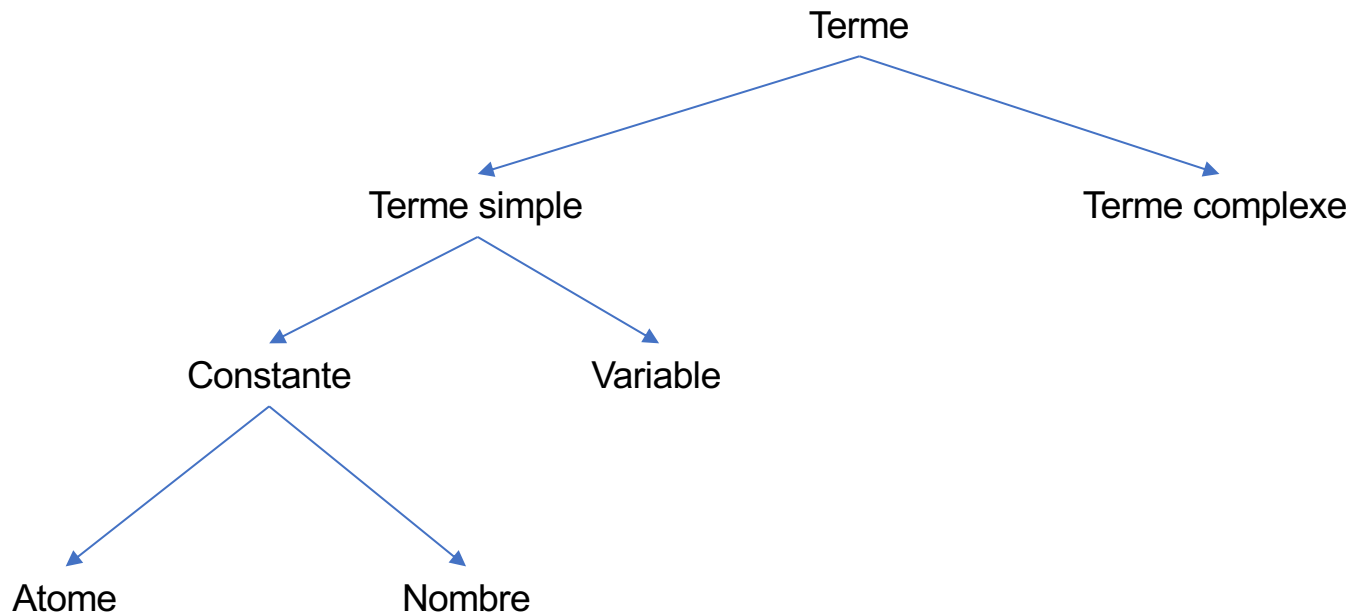
Recherche de preuves par l'unification

Enseignant: NGUYEN Thi Minh Tuyen

Rappel: Unification

- Deux termes t_1 et t_2 sont unifiables s'il existe une substitution σ des variables de t_1 et t_2 telle que $\sigma t_1 = \sigma t_2$
- `pere(X,Y)` s'unifie avec `pere(trung, thang)` si
 - `trung` \vdash `X` et
 - `thang` \vdash `Y`

Rappel: Termes en Prolog



Unification

- Deux termes s'unifient si
 - Ils sont égaux ou
 - Ils contiennent des variables qui peuvent être instanciées avec des termes pour que les termes résultants soient égaux.

- C'est-à-dire:
 - `trung` s'unifie avec `trung`
 - `thang` s'unifie avec `thang`
 - `pere(trung, thang)` s'unifie avec `pere(trung, thang)`
 - ➔ Ils sont identiques.
 - `trung` ne s'unifie pas avec `thang`
 - `pere(trung, thang)` ne s'unifie pas avec `pere(thang, trung)`

- `trung` et `X` ?
- `feminin(X)` et `feminin(an)` ?
- `pere(trung, Y)` et `pere(X, thang)` ?

Instanciación

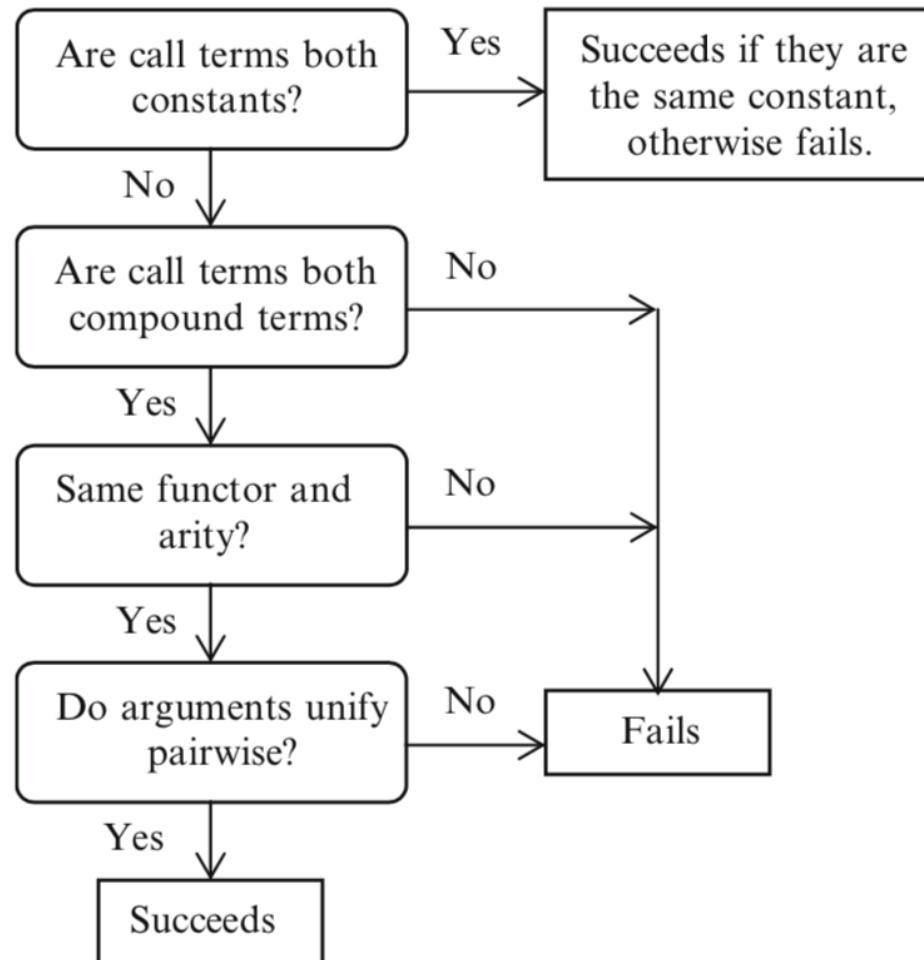
- Quand Prolog unifie deux termes:
 - Il génère toutes les instanciaciones nécessaires, puis il cherche deux termes égaux.

Rappel: Définition

- T1 et T2 sont deux constantes: T1 s'unifie avec T2 s'ils sont le même atome ou le même nombre (constante)
- T1 est une variable, T2 est un terme: T1 s'unifie avec T2 si T1 est une instanciation de T2 et l'inverse.
- Si T1 et T2 sont deux termes complexes: T1 s'unifie avec T2 si:
 - Ils ont le même nom du prédicat et même arité et
 - Tous les paramètres sont unifiables et
 - Tous les instanciations des variables sont compatibles.

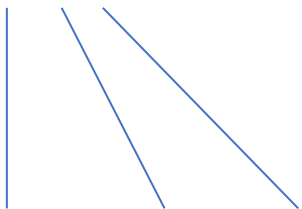
Unification de deux termes

9



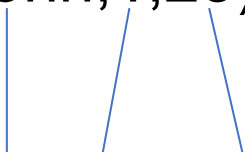
Example [1]

person(X,Y,Z)



person(john,smith,27)

person(john,Y,23)



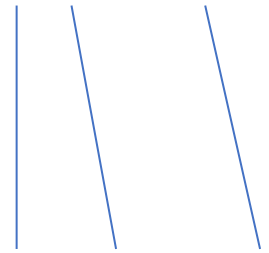
person(X,smith,27)

Exemple [2]

11

pred1(X,Y,[a,b,c])

pred1(A,prolog,B)



The diagram consists of three blue lines connecting the arguments of the two predicate calls. The first line connects 'X' in the top call to 'A' in the bottom call. The second line connects 'Y' in the top call to 'prolog' in the bottom call. The third line connects '[a,b,c]' in the top call to 'B' in the bottom call.

Variables répétées

- Une variable apparaît plus d'une fois dans un terme composé

pred2(X,X,man)

pred2(london,dog,A)



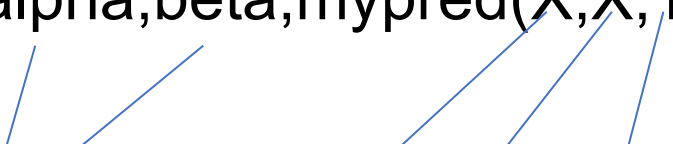
pred3(X,X,man)

pred3(london,london,A)

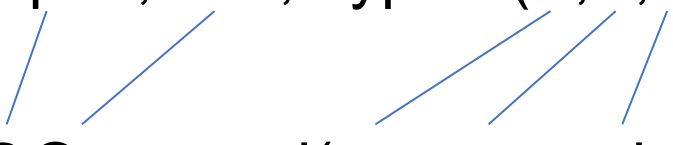


Example

pred(alpha,beta,mypred(X,X,Y))
pred(P,Q,mypred(no,yes,maybe))



pred(alpha,beta,mypred(X,X,Y))
pred(P,Q,mypred(no,no,maybe))



Évaluer un but

14

pred(london,A).

pred(X,'european capital'):-capital(X,Y),european(Y),write(X),nl.

?-pred(london,A).

pred(london,'european capital'):-
capital(london,england),european(england),write(london),nl.

capital(london,england).

european(england):-write('God Save the Queen!'),nl.

Évaluer un but

15

?-pred(london,A).

God Save the Queen!

london

A = 'european capital'

Prolog unification: =/2

16

?- thang=thang.

true.

?- trung=thang.

false.

?- 20=20.

true.

?- trung=X.

X = trung.

?- X=trung,X=thang.

false.

?- pere(trung,X)=pere(trung,thang).

X = thang.

?- X=Y.

X = Y.

?- pere(X,X)=pere(trung,thang).

false.

Prolog et unification

17

- Prolog n'utilise pas l'algorithme d'unification standard.
- On considère la requête suivante:
 - $\text{male}(X)=X$.
 - Est-ce que ces deux termes sont unifiables?

Terme infini

18

?- male(X)=X.

X = male(male(male(male...))).

"Occurs Check"

- Avec l'algorithme d'unification standard: on prend en compte un "occurs check".
- En Prolog:
?- unify_with_occurs_check(male(X), X).
false.

Programmer à l'aide de l'unification

20

- lines.pl

vertical(line(point(X,Y),point(X,Z))).

horizontal(line(point(X,Y),point(Z,Y))).

Requêtes:

?- vertical(line(point(1,1),point(1,3))).

true.

?- vertical(line(point(1,1),point(3,2))).

false.

?-horizontal(line(point(1,1),point(1,Y))).

Y = 1.

?- horizontal(line(point(2,3),Point)).

Point = point(_G273, 3).

Stratégie de recherche [1]

21

$f(a).$

$f(b).$

$g(a).$

$g(b).$

$h(b).$

$k(X):- f(X), g(X), h(X).$

Requête:

$?-k(Y).$

Stratégie de recherche [2]

22

loves(vincent,mia).

loves(marsellus,mia).

jealous(A,B):- loves(A,C), loves(B,C).

Requête:

?- jealous(X,Y).

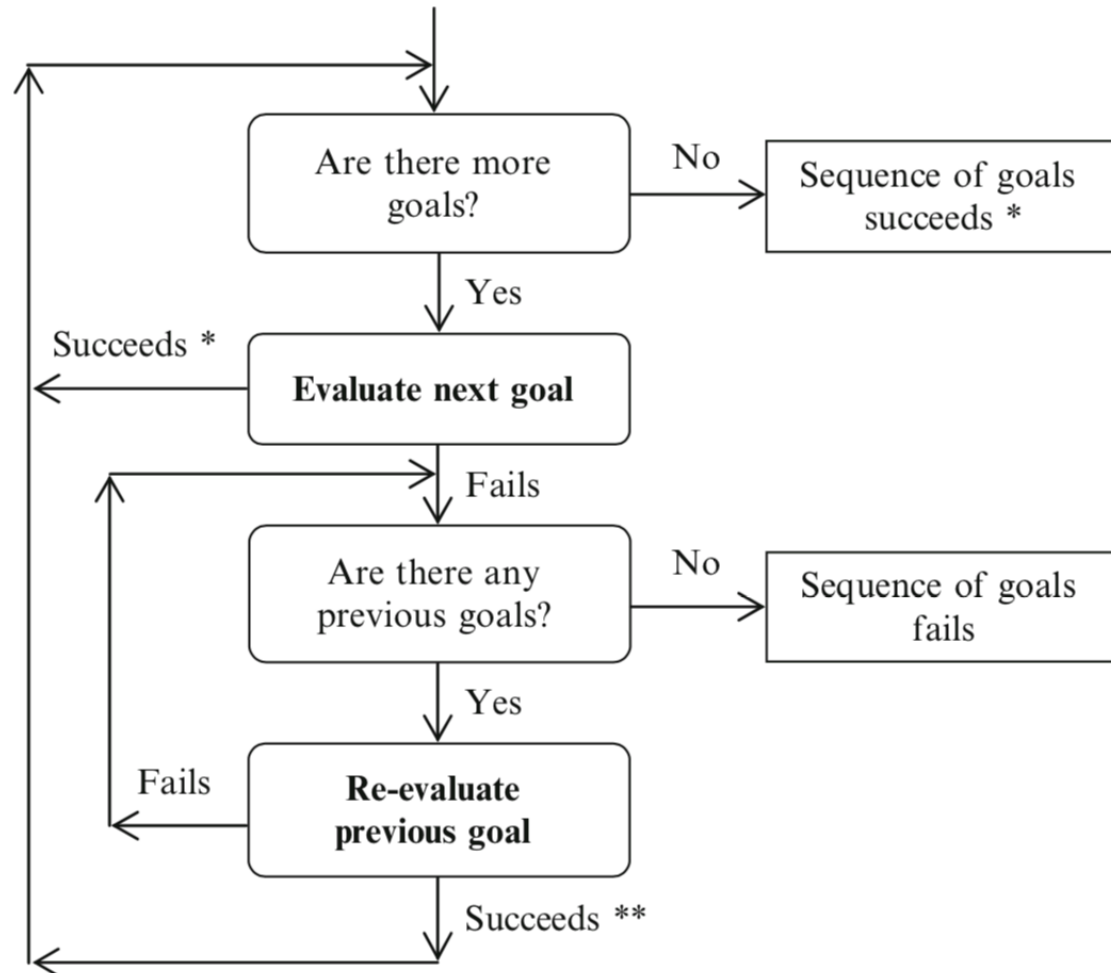
Exercices:

23

- Lesquelles de ces paires de termes s'unifient ? Le cas échéant, donnez les instanciations de variables qui permettent l'unification.

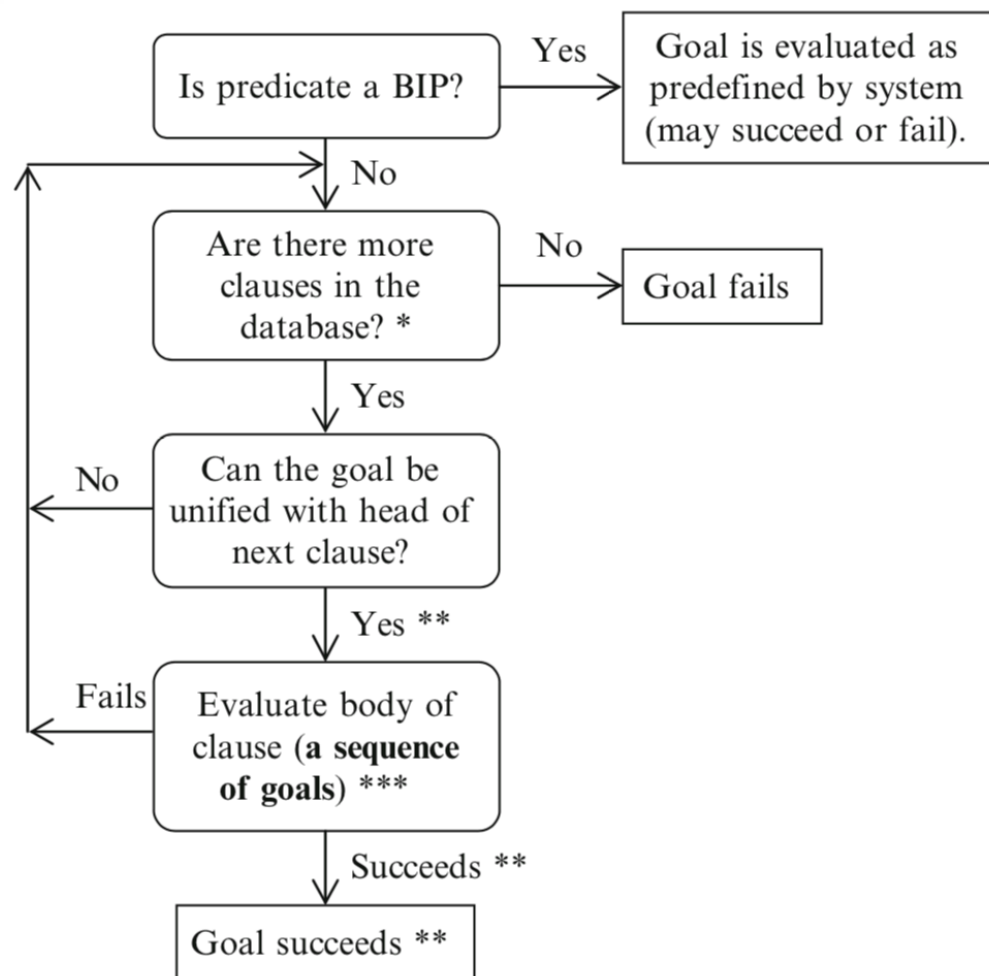
Évaluation d'une séquence des buts

24



Évaluation d'un but

25



Backtracking

26

- Un processus consistant à revenir à un but et à essayer de le satisfaire. C'est-à-dire, de trouver un autre moyen de le satisfaire.

Example

27

[M1] mother(ann,henry).
[M2] mother(ann,mary).
[M3] mother(jane,mark).
[M4] mother(jane,francis).
[M5] mother(annette,jonathan).
[M6] mother(mary,bill).
[M7] mother(janice,louise).
[M8] mother(lucy,janet).
[M9] mother(louise,caroline).
[M10] mother(louise,martin).
[F1] father(henry,jonathan).
[F2] father(john,mary).
[F3] father(francis,william).
[F4] father(francis,louise).
[F5] father(john,mark).
[F6] father(gavin,lucy).
[F7] father(john,francis).
[F8] father(martin,david).

[F9] father(martin,janet).

[P1] parent(victoria,george).
[P2] parent(victoria,edward).
[P3] parent(X,Y):-
write('mother?'),nl,mother(X,Y),
write('mother!'),nl.
[P4] parent(A,B):-
write('father?'),nl,father(A,B),
write('father!'),nl.
[P5] parent(elizabeth,charles).
[P6] parent(elizabeth,andrew).

**?-parent(john,Child),write('The child
is '),write(Child),nl.**

Requête

28

?-parent(john,Child),write('The child is '),write(Child),nl.

[P3] parent(john,Y):-write('mother?'),nl,mother(john,Y),write('mother!'),nl.

[P4] parent(john,B):-write('father?'),nl,father(john,B),write('father!'),nl.

[F2] father(john,mary).

?- parent(john,Child),write('The child is '),write(Child),nl.

mother?

father?

father!

The child is mary Child = mary

Forcer le système à revenir en arrière pour trouver d'autres solutions

29

?-parent(john,Child),write('The child is '),write(Child),nl.

[P4] parent(john,mary):-write('father?'),nl,father(john,mary),write('father!'),nl.

[F2] father(john,mary).

[P4] parent(john,mark):-write('father!'),nl,father(john,mark) write('father!'),nl.

[F5] father(john,mark).

?-parent(john,Child)),write('The child is '),write(Child),nl.

[P4] parent(john,francis):-write('father!'),nl,father(john,francis).

[F7] father(john,francis).

Question?

30