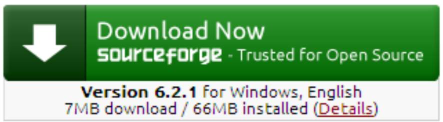
SWI Prolog Portable

http://portableapps.com/apps/development/swi-prolog_portable

```
1 ?- [p1].
% p1 compiled 0.00 sec, 22 clauses
true.
```

p1.pl



Download Now
sourceforge - Trusted for Open Source
Version 6.2.1 for Windows, English
7MB download / 66MB installed (Details)

```
2 ?- listing.
max([H],H).
max([H|T],H):-max(T,X),H>=X.
max([H|T],X):-max(T,X),H<X..
true.
```

```
3 ?- max([2,5,3,4],A).
A = 5 ;
```

```
4 ?- halt.
Ctrl-D
```

```
5 ?- trace.

6 ?- notrace.

7 ?- consult('animals.pl').

8 ?- write('hello, world').

9 ?- edit(queens).
```

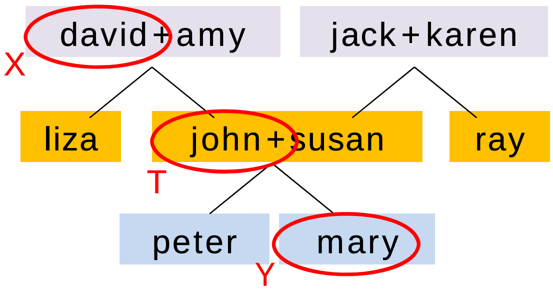
INS = copy
↑↓ = history

Facts事實

father

```
sunny.
father(john,    peter).
father(john,    mary).
father(david,   liza).
father(david,   john).
father(jack,    susan).
father(jack,    ray).

mother(susan,   peter).
mother(susan,   mary).
mother(amy,     liza).
mother(amy,     john).
mother(karen,   susan).
mother(karen,   ray).
```



Rules規則

if

and

```
yeye(X, Y) :-
        father(X, T), father(T, Y).

mama(X, Y) :-
        mother(X, T), father(T, Y).

gunggung(X, Y) :-
        father(X, T), mother(T, Y).

popo(X, Y) :-
        mother(X, T), mother(T, Y).
```

Goals目標、 Queries查詢

```
? yeye(david,peter).
```

```
? yeye(A,peter).
```

```
? yeye(david,A).
```

SWI Prolog Portable

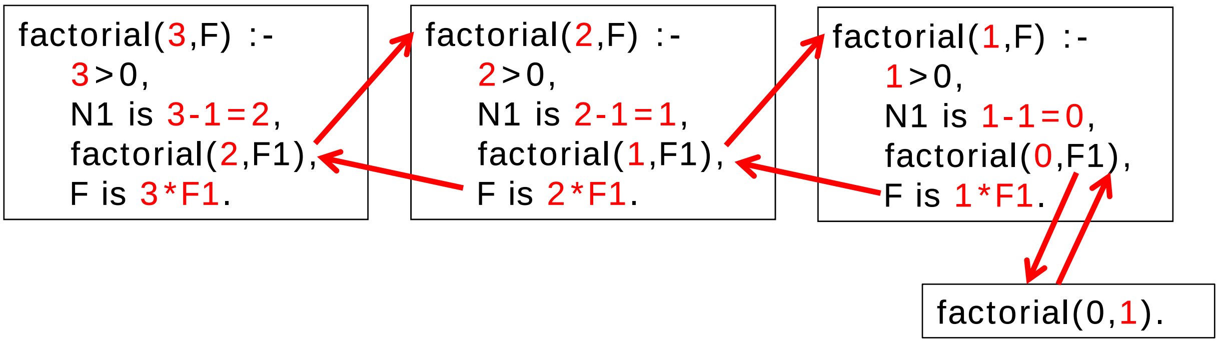
factorial(5) = 5×4×3×2×1

```
factorial(0,1).

factorial(N,F) :-
          N>0,
          N1 is N-1,
          factorial(N1,F1),
          F is N*F1.
```

```
   factorial(0) = 1
or
   factorial(N) = F
   if        N>0
   &&        N1 is N-1,
   &&        factorial(N1) = F1,
   &&        F is N*F1.
```

factorial(3,X).



```
factorial(3,F) :-
    3>0,
    N1 is 3-1=2,
    factorial(2,F1),
    F is 3*F1.
```

```
factorial(2,F) :-
    2>0,
    N1 is 2-1=1,
    factorial(1,F1),
    F is 2*F1.
```

```
factorial(1,F) :-
    1>0,
    N1 is 1-1=0,
    factorial(0,F1),
    F is 1*F1.
```

```
factorial(0,1).
```

SWI Prolog Portable

Rules規則

Don’t care

```
   first([H|_],H).

R1 last([H],H).

R2 last([_|T],X) :- last(T,X).
```

R1 or R2

Queries查詢

H H

first([1|2,3,4],X).

X=

T

last([1|2,3,4],X).

last([2|3,4],X).

last([3|4],X).

last([4],X).

R1

X=

```
R1  max([H],H).

R2  max([H|T],H):-
            max(T,X),
            H>=X.

R3  max([H|T],X):-
            max(T,X),
            H<X.
```

H is max, if Only 1 element left

H is max, if
X is max(T)
and H>=X

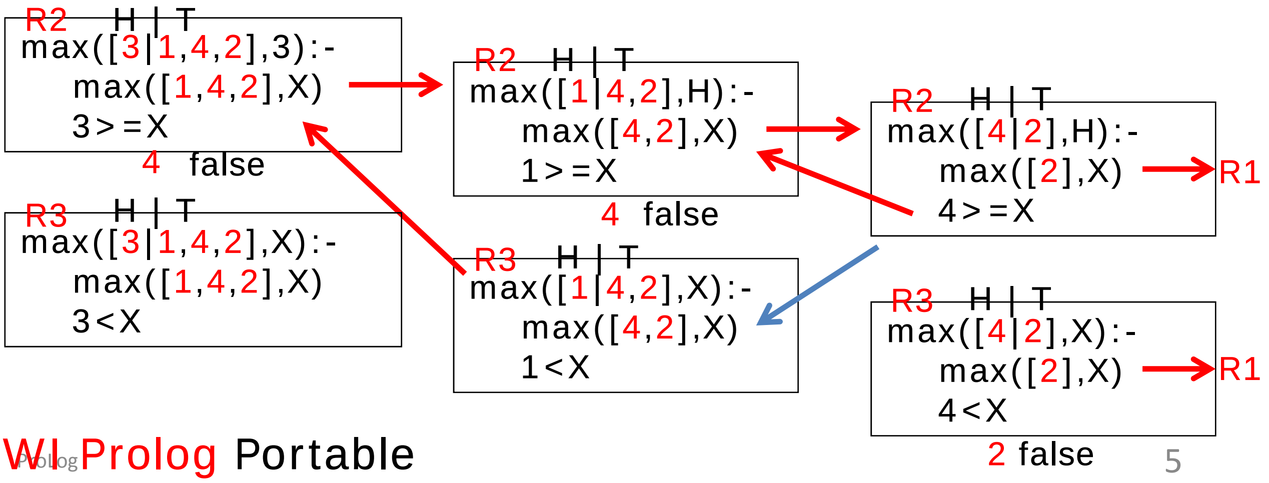
X is max, if
X is max(T)
and H<X

只有1個元素

X = max(Tail)
Head >= X

X = max(Tail)
X > Head

max([3,1,4,2],H).



R2 H | T
max([3|1,4,2],3):-
max([1,4,2],X)
3>=X
4 false

R3 H | T
max([3|1,4,2],X):-
max([1,4,2],X)
3<X

R2 H | T
max([1|4,2],H):-
max([4,2],X)
1>=X
4 false

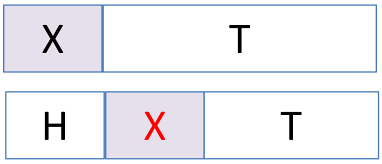
R3 H | T
max([1|4,2],X):-
max([4,2],X)
1<X

R2 H | T
max([4|2],H):-
max([2],X)
4>=X
R1

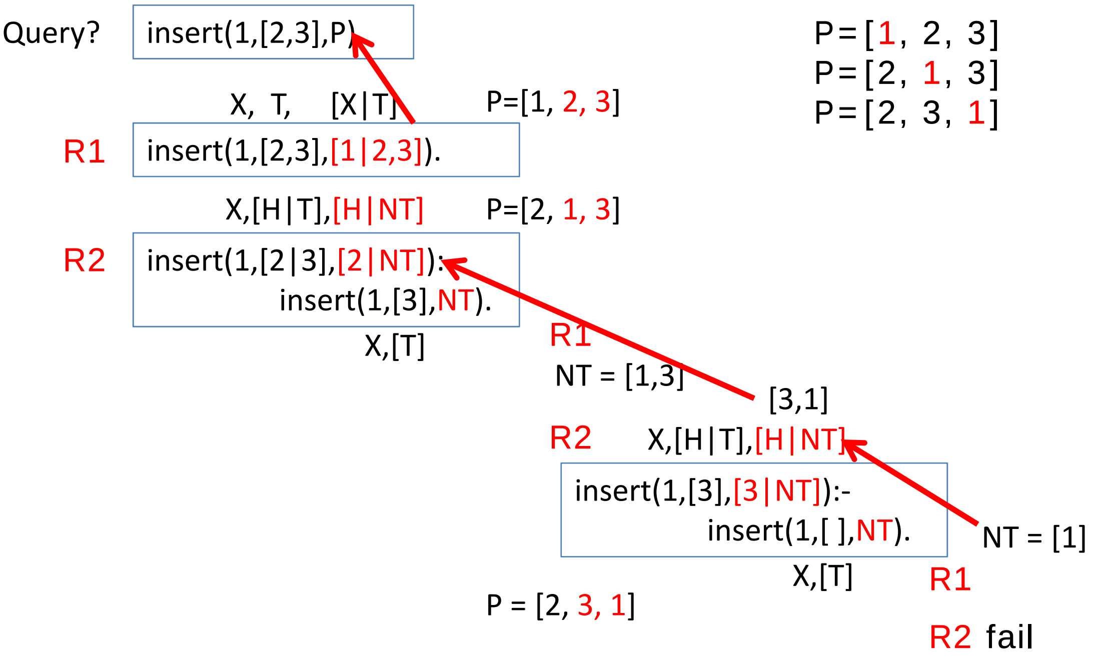
R3 H | T
max([4|2],X):-
max([2],X)
4<X
R1
2 false

SWI Prolog Portable

```
R1  insert(X,T,[X|T]).
R2  insert(X,[H|T],[H|NT]):-
              insert(X,T,NT).
```



P=[1, 2, 3]
P=[2, 1, 3]
P=[2, 3, 1]



Query? insert(1,[2,3],P)

X, T, [X|T]
P=[1, 2, 3]

R1 insert(1,[2,3],[1|2,3]).

X,[H|T],[H|NT]
P=[2, 1, 3]

R2 insert(1,[2|3],[2|NT]):-
insert(1,[3],NT).
X,[T]

R1
NT = [1,3]
[3,1]

R2 X,[H|T],[H|NT]
insert(1,[3],[3|NT]):-
insert(1,[ ],NT).
X,[T]
NT = [1]
R1
P = [2, 3, 1]
R2 fail

DSE2012 Q.4(c)

事實		程式子句
體育 (sports)學會	屬於學生會union	belongsto(sports, union),
音樂 (music)學會	屬於學生會	belongsto(music, union).
棋藝 (chess)學會	屬於學生會	belongsto(chess, union).
美術 (art)學會	屬於學生會	belongsto(art, union).
長笛 (flute)組	屬於音樂學會	belongsto(flute, music).
雙簧管(oboe)組	屬於音樂學會	belongsto(oboe, music).
籃球 (basketball)組	屬於體育學會	belongsto(basketball, sports).

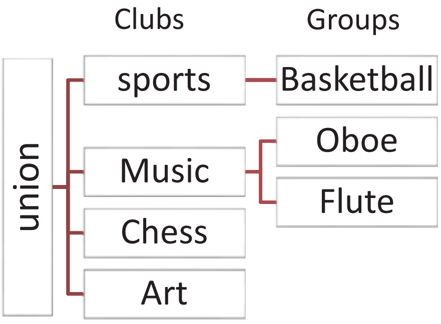
規則

若X屬於 學生會，它便是一個學會。

若X屬於一個學會，它便是一個組別。

程式子句

```
club(X):-  belongsto(X,union).
group(X):- belongsto(X,Y) & club(Y).
```



以下的例子是一些查詢的結果。

查詢

```
?- belongsto(chess, union).
?- belongsto(science, union).
?- club(A).
```

結果?

(i) 下列查詢的結果是什麼?

```
(1) ?- group(volleyball).
(2) ?- belongsto(B, music).
(3) ?- group(C).
```

(ii) 寫出尋找美術學會所屬組織的查詢。

(iii) 與過程語言(C)比較，使用邏輯語言(prolog,lisp)有什麼好處?

Prolog Tutorial

http://www.cpp.edu/~jrfisher/www/prolog_tutorial/contents.html

Animal Identification

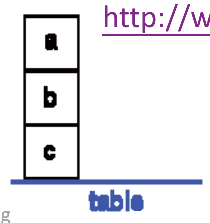
http://www.cpp.edu/~jrfisher/www/prolog_tutorial/2_17.html

Queens

http://www.cpp.edu/~jrfisher/www/prolog_tutorial/2_11.html

Blocks World

http://www.cpp.edu/~jrfisher/www/prolog_tutorial/2_19.html



```
on(a,b).
on(b,c).
on(c,table).
```