

Tutorial 10

Week of November 21, 2005

Overview

Dear Students,

1. We will continue with some of the Tutorial 9 list predicates that we didn't get to last week.
2. Question 1 builds on what you've done in A4. The predicate will be used in A5.
3. Question 2 reviews negation as failure.

Sheila

1 A list predicate for A5

Define a predicate `allmax(+List, -Index, -Maximum)` that given a list `List` holds if and only if `Maximum` is a maximal value in the list and `Index` is its index on the list.

*Your predicate should be able to find **all** maxima! That means if your list has more than one maximum then pressing ';' after the first answer should provide all other possible answers (i.e. Index/Maximum combinations, see example).*

Solution 1: Without ! or ->.

```
allmax(+List, -Index, -Max)
(use setof to find all ;-))
*/
```

```
allmax([X], 0, X).
allmax([H|L], N, X) :-
    L \= [],
    allmax(L, SN, SX),
    (
        H > SX,
        X = H,
        N = 0
    ;
        (
            H = SX,
            X = H,
            N = 0
        ;
            H =< SX,
            X = SX,
            N is SN+1
        )
    ).
```

Solution 2: With ! and ->.

```
/** find a maximum in a list allmax( +List, -Index, -Max)
(use setof to find all ;-))
```

You may use ! and ->

```
*/
```

```
allmax( [X], 0, X) :- !.
allmax( [H|L], N, X) :-
    allmax( L, SN, SX),
    (
        H > SX % IF H > SX
    ->
        % THEN
        X = H,
        N = 0
    ;
        % ELSE
        (
            H = SX,
            X = H,
            N = 0
        ;
            X = SX,
            N is SN+1
        )
    ).
```

Test these predicates out online. The code is on the tutorial web page together with some test cases.

2 Negation by failure

Example.

```
p(a).  
p(b).  
q(c).
```

```
?- \+p(X), q(X).  
No
```

```
?- q(X), \+p(X).  
X = c ;  
No
```

Another example:

```
bachelor(P) :- male(P), not(married(P)).  
male(henry).  
male(tom).  
married(tom).
```

```
?- bachelor(henry).  
Yes  
?- bachelor(tom).  
No  
?- bachelor(Who).  
Who= henry ;  
No  
?- not(married(Who)).  
No.
```

The first three responses are correct and as expected. The answer to the fourth query might have been unexpected at first. But consider that the goal `?-not(married(Who))` fails because for the variable binding `Who=tom`, `married(Who)` succeeds, and so the negative goal fails. Thus, negative goals `?-not(g)` with variables cannot be expected to produce bindings of the variables for which the goal `g` fails.