

```
% [version 1.2]

% rp( +Expr, ?List ) - List is expression Expr
%                       in postfix (reverse Polish) notation

rp( A, [A] ) :- constant( A ).
rp( +(A,B), AABBP ) :- rp( A, AA ), rp( B, BB ),
    append( BB, [+], BBP ),
    append( AA, BBP, AABBP ).
rp( *(A,B), AABBP ) :- rp( A, AA ), rp( B, BB ),
    append( BB, [*], BBP ),
    append( AA, BBP, AABBP ).

% constant( +C ) - C is a constant (i.e. has no arguments)
constant( C ) :- functor( C, _, 0 ).

% rp( A,...) loops for A uninstantiated
% We change the control without changing the logic:

% rpl( ?Expr, +List )

rpl( A, [A] ) :- constant( A ).
rpl( +(A,B), AABBP ) :-
    append( AA, BBP, AABBP ),
    append( BB, [+], BBP ),
    rpl( A, AA ), rpl( B, BB ).
rpl( *(A,B), AABBP ) :-
    append( AA, BBP, AABBP ),
    append( BB, [*], BBP ),
    rpl( A, AA ), rpl( B, BB ).

% Notation, used in the documentation, for specifying requirements
% on predicate arguments (rather obscure)

% + - input
% ? - output or input
% - - unbound
% @ - will not be further instantiated
```