```
%--- Normalize an LTL formula by pushing negations to the level of propositions
%--- and applying some absorption and idempotency laws.


normalize( X, Normalized ) :-
        (
            var( X )
        ->
            write( '*** The formula is a Prolog variable!' ),
            nl,
            Normalized = ?
        ;
            once( norm( X, Normalized, OK ) )
        ->
            var( OK )
        ).
```

```
% If an error is found, OK will cease to be a variable.

norm( ~( A v B ) , Normalized, OK ) :-
       once( norm( ~ A ^ ~ B , Normalized, OK ) ).

norm( ~( A ^ B ) , Normalized, OK ) :-
       once( norm( ~ A v ~ B , Normalized, OK ) ).

norm( ~ ~ A , Normalized, OK ) :-
       once( norm( A , Normalized, OK ) ).

norm( ~ x A , Normalized, OK ) :-
       once( norm( x ~ A , Normalized, OK ) ).

norm( ~ f A , Normalized, OK ) :-
       once( norm( g ~ A , Normalized, OK ) ).

norm( ~ g A , Normalized, OK ) :-
       once( norm( f ~ A , Normalized, OK ) ).

norm( ~( A u B ) , Normalized, OK ) :-
       once( norm( ~ A r ~ B , Normalized, OK ) ).

norm( ~( A r B ) , Normalized, OK ) :-
       once( norm( ~ A u ~ B , Normalized, OK ) ).

norm( A v B , NA v NB, OK ) :-
       once( norm( A, NA, OK ) ),
       once( norm( B, NB, OK ) ).

norm( A ^ B , NA ^ NB, OK ) :-
       once( norm( A, NA, OK ) ),
       once( norm( B, NB, OK ) ).

norm( ~ A , ~ NA, OK ) :-
       once( norm( A, NA, OK ) ).

norm( x A , x NA, OK ) :-
       once( norm( A, NA, OK ) ).

norm( f A , f NA, OK ) :-
       once( norm( A, NA, OK ) ).

norm( g A , g NA, OK ) :-
       once( norm( A, NA, OK ) ).

norm( A u B , NA u NB, OK ) :-
       once( norm( A, NA, OK ) ),
       once( norm( B, NB, OK ) ).

norm( A r B , NA r NB, OK ) :-
       once( norm( A, NA, OK ) ),
       once( norm( B, NB, OK ) ).

norm( f f A , Normalized, OK ) :-
       once( norm( f A , Normalized, OK ) ).

norm( g g A , Normalized, OK ) :-
       once( norm( g A , Normalized, OK ) ).

norm( A u (A u B) , Normalized, OK ) :-
       once( norm( A u B , Normalized, OK )  ).
```

```
norm( (A u B) u B , Normalized, OK ) :-
        once( norm( A u B , Normalized, OK )  ).

norm( f g f A , Normalized, OK ) :-
        once( norm( g f A , Normalized, OK ) ).

norm( g f g A , Normalized, OK ) :-
        once( norm( f g A , Normalized, OK ) ).

norm( x A ^ x B , Normalized, OK ) :-
        once( norm( x (A ^ B) , Normalized, OK ) ).

norm( g A ^ g B , Normalized, OK ) :-
        once( norm( g (A ^ B) , Normalized, OK ) ).

norm( f A v f B , Normalized, OK ) :-
        once( norm( f (A v B) , Normalized, OK ) ).

norm( ~ P, ~ P, _ ) :-
        proposition( P ).

norm( P, P, _ ) :-
        proposition( P ).

norm( X, ?, ? ) :-
        write( '*** This is not a well-formed (sub)formula: \"' ),
        write( X ),
        write( '\"' ),
        nl.
```