```
Script started on Tue Mar  6 21:01:00 2012
bash-3.2$ cat -n fox\007chicken.pl
     1  % $Id: foxchicken.pl,v 1.3 2011-05-19 19:53:59-07 - - $ */
     2
     3  %
     4  % A farmer has with him a fox, a chicken, and a sack of grain.
     5  % He comes to a river and sees a small boat.  He needs to bring
     6  % all three of his things across the river, but the boat is so
     7  % small that only one thing will fit in it with him.  He can not
     8  % leave the fox and the chicken together or the fox will eat the
     9  % chicken.  He can not leave the chicken and the grain together
    10  % or the chicken will eat the grain.  The fox, however, does not
    11  % eat grain.  How should the farmer proceed?
    12  %
    13
    14  not( X ) :- X, !, fail.
    15  not( _ ).
    16
    17  %
    18  % FACTS AND CULLINARY HABITS:
    19  %
    20
    21  eats( fox, chicken ).
    22  eats( chicken, grain ).
    23  property( [ fox, chicken, grain ] ).
    24  goal( other ).
    25  start( first ).
    26
    27  chow_time( List ) :-
    28     member_of( Diner, List ),
    29     member_of( Dinner, List ),
    30     eats( Diner, Dinner ).
    31
    32  %
    33  % SET RELATIONS:
    34  %
    35
    36  member_of( H, [ H | _ ] ).
    37  member_of( H, [ _ | T ] ) :- member_of( H, T ).
    38
    39  matches( [], [] ).
    40  matches( [ H | T1 ], [ H | T2 ] ) :- matches( T1, T2 ).
    41
    42  removex( H, [ H | T ], T ).
    43  removex( X, [ H | T ], [ H | U ] ) :- removex( X, T, U ).
    44
    45  insert( H, T, [ H | T ] ).
    46
```

```
%
% TRAVEL PLANS AND FREQUENT PADDLER MILES.
%

travel :-
   start( From ),
   goal( To ),
   property( Property ),
   print_start( From, To, Property, [] ),
   move( From, To, Property, [], nothing ),
   print_done.

move( From, _, _, [], _ ) :-
   goal( From ).

move( From, To, This, That, _ ) :-
   start( To ),
   goal( From ),
   not( chow_time( This )),
   print_alone( From, To, This, That ),
   move( To, From, That, This, nothing ).

move( From, To, This, That, Just_took ) :-
   removex( What, This, This_later ),
   not( Just_took = What ),
   not( chow_time( This_later )),
   insert( What, That, That_later ),
   print_takes( What, From, To, This_later, That_later ),
   move( To, From, That_later, This_later, What ).

```

```
%
% TRAVEL INSTRUCTIONS.
%

print_start( From, To, This, That ) :-
   nl,
   write( ’The farmer is by a river and ’ ),
   write( ’wants to take his cargo across.’ ),
   nl,
   print_status( From, This ),
   print_status( To, That ).

print_done :-
   nl,
   write( ’Finally, the farmer is done!’ ),
   nl.

print_takes( What, From, To, This, That ) :-
   nl,
   write( ’The farmer takes the ’ ),
   write( What ),
   write( ’ from the ’ ),
   write( From ),
   write( ’ side to the ’ ),
   write( To ),
   write( ’ side.’ ),
   nl,
   print_status( From, This ),
   print_status( To, That ).

print_alone( From, To, This, That ) :-
   nl,
   write( ’The farmer travels alone from the ’ ),
   write( From ),
   write( ’ side to the ’ ),
   write( To ),
   write( ’ side.’ ),
   nl,
   print_status( From, This ),
   print_status( To, That ).

print_status( Where, What ) :-
   tab( 10 ),
   write( ’On the ’ ),
   write( Where ),
   write( ’ side is ’ ),
   print_list( nothing, ’’, What ).

print_list( Nothing, _, [] ) :-
   write( Nothing ),
   write( ’.’ ),
   nl.

print_list( _, Comma, [ H | T ] ) :-
   write( Comma ),
   write( ’the ’ ),
   write( H ),
   print_list( ’’, ’, ’, T ).

% TEST: travel.

bash-3.2$ gprolog
GNU Prolog 1.3.1
```

```
By Daniel Diaz
Copyright (C) 1999-2009 Daniel Diaz
| ?- [foxchicken].
compiling /afs/cats.ucsc.edu/courses/cmps112-wm/Languages/prolog/Examples/foxchicken.pl for byte code...
/afs/cats.ucsc.edu/courses/cmps112-wm/Languages/prolog/Examples/foxchicken.pl compiled, 137 lines read - 9261 bytes written, 14 ms

(1 ms) yes
| ?- travel.

The farmer is by a river and wants to take his cargo across.
          On the first side is the fox, the chicken, the grain.
          On the other side is nothing.

The farmer takes the chicken from the first side to the other side.
          On the first side is the fox, the grain.
          On the other side is the chicken.

The farmer travels alone from the other side to the first side.
          On the other side is the chicken.
          On the first side is the fox, the grain.

The farmer takes the fox from the first side to the other side.
          On the first side is the grain.
          On the other side is the fox, the chicken.

The farmer takes the chicken from the other side to the first side.
          On the other side is the fox.
          On the first side is the chicken, the grain.

The farmer takes the grain from the first side to the other side.
          On the first side is the chicken.
          On the other side is the grain, the fox.

The farmer travels alone from the other side to the first side.
          On the other side is the grain, the fox.
          On the first side is the chicken.

The farmer takes the chicken from the first side to the other side.
          On the first side is nothing.
          On the other side is the chicken, the grain, the fox.

Finally, the farmer is done!

true ?

(1 ms) yes
| ?-

bash-3.2$ exit

Script done on Tue Mar  6 21:01:59 2012
```