```
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
```

```
49:     47  %
50:     48  % TRAVEL PLANS AND FREQUENT PADDLER MILES.
51:     49  %
52:     50
53:     51  travel :-
54:     52     start( From ),
55:     53     goal( To ),
56:     54     property( Property ),
57:     55     print_start( From, To, Property, [] ),
58:     56     move( From, To, Property, [], nothing ),
59:     57     print_done.
60:     58
61:     59  move( From, _, _, [], _ ) :-
62:     60     goal( From ).
63:     61
64:     62  move( From, To, This, That, _ ) :-
65:     63     start( To ),
66:     64     goal( From ),
67:     65     not( chow_time( This )),
68:     66     print_alone( From, To, This, That ),
69:     67     move( To, From, That, This, nothing ).
70:     68
71:     69  move( From, To, This, That, Just_took ) :-
72:     70     removex( What, This, This_later ),
73:     71     not( Just_took = What ),
74:     72     not( chow_time( This_later )),
75:     73     insert( What, That, That_later ),
76:     74     print_takes( What, From, To, This_later, That_later ),
77:     75     move( To, From, That_later, This_later, What ).
78:     76
```

```
 79:     77  %
 80:     78  % TRAVEL INSTRUCTIONS.
 81:     79  %
 82:     80
 83:     81  print_start( From, To, This, That ) :-
 84:     82     nl,
 85:     83     write( 'The farmer is by a river and ' ),
 86:     84     write( 'wants to take his cargo across.' ),
 87:     85     nl,
 88:     86     print_status( From, This ),
 89:     87     print_status( To, That ).
 90:     88
 91:     89  print_done :-
 92:     90     nl,
 93:     91     write( 'Finally, the farmer is done!' ),
 94:     92     nl.
 95:     93
 96:     94  print_takes( What, From, To, This, That ) :-
 97:     95     nl,
 98:     96     write( 'The farmer takes the ' ),
 99:     97     write( What ),
100:     98     write( ' from the ' ),
101:     99     write( From ),
102:    100     write( ' side to the ' ),
103:    101     write( To ),
104:    102     write( ' side.' ),
105:    103     nl,
106:    104     print_status( From, This ),
107:    105     print_status( To, That ).
108:    106
109:    107  print_alone( From, To, This, That ) :-
110:    108     nl,
111:    109     write( 'The farmer travels alone from the ' ),
112:    110     write( From ),
113:    111     write( ' side to the ' ),
114:    112     write( To ),
115:    113     write( ' side.' ),
116:    114     nl,
117:    115     print_status( From, This ),
118:    116     print_status( To, That ).
119:    117
120:    118  print_status( Where, What ) :-
121:    119     tab( 10 ),
122:    120     write( 'On the ' ),
123:    121     write( Where ),
124:    122     write( ' side is ' ),
125:    123     print_list( nothing, '', What ).
126:    124
127:    125  print_list( Nothing, _, [] ) :-
128:    126     write( Nothing ),
129:    127     write( '.' ),
130:    128     nl.
131:    129
132:    130  print_list( _, Comma, [ H | T ] ) :-
133:    131     write( Comma ),
134:    132     write( 'the ' ),
135:    133     write( H ),
136:    134     print_list( '', ', ', T ).
137:    135
138:    136  % TEST: travel.
139:    137
140: bash-3.2$ gprolog
141: GNU Prolog 1.3.1
```

```
  142: By Daniel Diaz
  143: Copyright (C) 1999-2009 Daniel Diaz
  144: | ?- [foxchicken].
  145: compiling /afs/cats.ucsc.edu/courses/cmps112-wm/Languages/prolog/Examples/foxchi
cken.pl for byte code...
  146: /afs/cats.ucsc.edu/courses/cmps112-wm/Languages/prolog/Examples/foxchicken.pl co
mpiled, 137 lines read - 9261 bytes written, 14 ms
  147:
  148: (1 ms) yes
  149: | ?- travel.
  150:
  151: The farmer is by a river and wants to take his cargo across.
  152:           On the first side is the fox, the chicken, the grain.
  153:           On the other side is nothing.
  154:
  155: The farmer takes the chicken from the first side to the other side.
  156:           On the first side is the fox, the grain.
  157:           On the other side is the chicken.
  158:
  159: The farmer travels alone from the other side to the first side.
  160:           On the other side is the chicken.
  161:           On the first side is the fox, the grain.
  162:
  163: The farmer takes the fox from the first side to the other side.
  164:           On the first side is the grain.
  165:           On the other side is the fox, the chicken.
  166:
  167: The farmer takes the chicken from the other side to the first side.
  168:           On the other side is the fox.
  169:           On the first side is the chicken, the grain.
  170:
  171: The farmer takes the grain from the first side to the other side.
  172:           On the first side is the chicken.
  173:           On the other side is the grain, the fox.
  174:
  175: The farmer travels alone from the other side to the first side.
  176:           On the other side is the grain, the fox.
  177:           On the first side is the chicken.
  178:
  179: The farmer takes the chicken from the first side to the other side.
  180:           On the first side is nothing.
  181:           On the other side is the chicken, the grain, the fox.
  182:
  183: Finally, the farmer is done!
  184:
  185: true ?
  186:
  187: (1 ms) yes
  188: | ?-
  189:
  190: bash-3.2$ exit
  191:
  192: Script done on Tue Mar  6 21:01:59 2012
```