/* Syllables in LOC - an ndfa representation */

```
/* A syllable in LOC is made up of:
a) a single vowel
b) a single vowel followed by a 'u'
c) a  consonant followed by a single vowel
[rules a, b, and c define a core syllable]
d) a core syllable followed by a single consonant

The vowels in LOC are : a, e, i, o, u
The consonants in LOC are : b,c,d,f,g,k,l,m,n,p,r,s,t,v,z.  */

/* A list in Prolog is represented by [El1,El2,...,Eln]
   The | operator isolates a number of elements (one or more) at the head of
the list ; what follows the | operator is always a list.
 Consequently, if we apply the [X|Y] pattern to the list [a,b,c,d,e] using the
unification operator (=) i.e.
   [X|Y] = [a,b,c,d,e],
 we get the following unifications :
  X = a , Y=[b,c,d,e] */

% PROLOG PROGRAM
go :- nl,
    write('Syllable recognizer - in LOC, an imaginary language'), nl,
    write('---------------------------'),nl,nl,
    write('callable goal is: syloc(String_to_be_examined)'),nl,
    write('where String_to_be_examined is entered as a letter list'),nl,
    write('e.g. [a,t]'),nl,
    write('the output is an analysis in terms of syllable constituents'),
    nl,
    write('or the message : No'),nl,nl.

 syloc(String) :- ndfa(syloc,String,Analysis),
                  write(Analysis).

 ndfa(Graph_Name,Symbol_List,[Graph_Name,initial(Initial_State)|
Analysis]) :-
        initial(Graph_Name,Initial_State),
        traverse(Graph_Name,Initial_State,Symbol_List,Analysis).
```

```
traverse(Graph_Name,
         State,
         [Symbol|Symbol_List],
         [transition(from(State),
                     to(Next_State),
                     reading(Symbol),
                     of_category(Category))|Category_List]) :-
        transition(Graph_Name,State,Next_State,Category),
        link(Symbol,Category),
        traverse(Graph_Name,Next_State,Symbol_List,Category_List).

 traverse(Graph_Name,Final_State,[],[final(Final_State)]) :-
        final(Graph_Name,Final_State).

/* the transitions */
 % reading a vowel, moving from q0 (initial) to q1 (final)
   transition(syloc,q0,q1,vowel).
 % reading a consonant, moving from q0 to qc
   transition(syloc,q0,qc,cons).
 % reading a vowel, moving from qc to q1
   transition(syloc,qc,q1,vowel).
 % reading a vowel, moving from q0 to qv
   transition(syloc,q0,qv,vowel).
 % reading a u, moving from qv to q1
   transition(syloc,qv,q1,u).
 % reading a consonant, moving from q1 to qcf (final)
   transition(syloc,q1,qcf,cons).

/* linking symbols and categories */
  link(Symbol,Category) :- listof(Category,Category_List),
                           inlist(Symbol,Category_List).
/* initial and final states */
 initial(syloc,q0).
 final(syloc,q1).
 final(syloc,qcf).

 /* the category lists */
 listof(vowel,[a,e,i,o,u]).
 listof(u,[u]).
 listof(cons,[b,c,d,f,g,k,l,m,n,p,r,s,t,v,z]).

/* membership of a list  */
 inlist(First,[First|_]).
 inlist(Element,[_|Tail]):- inlist(Element,Tail).
```