FONAL

File Operations in NAtural Language

APPENDIX - 2

Project Report

Work of

Rajesh .B
Muruganand .K
Aju Kuriakose Ninan

Guided by

Ms. R. Pavai. B.E.,

1991-92

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING OF BHARATHIAR UNIVERSITY COIMBATORE

Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore-641 006

```
/* Global domain declarations */

global domains
   Key=cr;up;down;esc;left;right;pgup;pgdn;otherspec

/* Global predicate declarations */

global predicates
    gomenu(integer)-(o)
    nondeterm go1(string,string)-(o,o)
    semant(string,string)-(i,i)
    intro
    readkey(Key)-(o)
    nondeterm
```

```
/*  Global database declarations */

database
    noun(string)
    noun1(string)
    verb(string)
    verb1(string)
    aux(string)
    art(string)
    adj(string)
    prep(string)
```

```
/* To give the pressed key value */

predicates
    readkey1(Key,char,integer)
    readkey2(key,integer)
clauses
    readkey(Key):-
         readchar(T),
         char_int(T,Val),
         readkey1(Key,T,Val).
    readkey1(Key,_,0):-
         !,
         readchar(T),
         char_int(T,Val),
         readkey2(Key,Val).
    readkey1(cr,_,13):-!.
    readkey1(esc,_,27):-!.
    readkey1(otherspec,_,_).
    readkey2(up,72):-!.
    readkey2(down,80):-!.
    readkey2(left,75):-!.
    readkey2(right,77):-!.
    readkey2(pgup,73):-!.
    readkey2(pgdn,81):-!.
    readkey2(otherspec,_).
```

```
/* MAIN PROGRAM */

 nowarnings
 code=3500

 /* Include files */

 include "globals.pro"
 include "gdb.pro"
 include "keys.pro"
 include "kb.pro"
 include "nlp.pro"
 include "mainmenu.pro"
 include "intro.pro"

 predicates
   checkop(integer)
   go

 /* Goal section */

 goal
   makewindow(1,113,27,"",0,0,24,80),
   makewindow(2,49,78,"MAIN MENU",2,25,19,30),
   makewindow(3,127,27,"INTRODUCTION",0,0,24,80),
   makewindow(4,113,27,"FONAL",0,0,24,80),
   makewindow(5,63,27,"PARSING",5,5,15,30),
   makewindow(6,63,75,"COMMENT",8,38,10,40),
   makewindow(7,116,27,"KNOWLEDGE BASE",0,0,24,80),
   makewindow(8,113,78,"MENU",5,25,13,30),
   makewindow(9,49,23,"ADDITION",5,18,15,45),
   makewindow(10,49,23,"DELETION",5,18,15,45),
   makewindow(11,49,78,"INTRO-MENU",7,25,13,30),
   makewindow(12,67,75,"FONAL-INTRO",5,13,15,55),
   makewindow(13,67,75,"DATABASE-INTRO",5,13,15,55),
   makewindow(80,75,0,"",24,0,1,80),
   consult("word.lst"),
   go.
 clauses
   go:-
     cursorform(-1,-1),
     shiftwindow(2),
     gomenu(A),
     checkop(A),
     go.

   /* EXIT */

   checkop(5):-
      !,cursorform(1,1),
      fail.
```

```
/* OS SHELL */

checkop(4):-
     existfile("a:\\command.com"),!,
     system("a:\\command.com").

checkop(4):-
     shiftwindow(80),
     clearwindow,
     field_str(0,2,32,"Boot disk is missing in A drive."),
     field_str(0,35,30,"Press any key to continue..."),
     readkey(_),
     clearwindow,
     shiftwindow(2).

/*  FONAL  */

checkop(3):-
       shiftwindow(1),
       shiftwindow(4),
       cursorform(1,1),
       go1(N,V),
       semant(N,V),
       cursorform(-1,-1),
       !.

/* KNOWLEDGE BASE operations */

checkop(2):-
       done,
       !.

/* INTRODUCTION */

checkop(1):-
       shiftwindow(3),
       intro,
       !.

intro:-
       start,!.

intro.
```

```
/* Main menu */

predicates
    print1a(integer)
    check1a(Key,integer)
    print2a(integer)
    check2a(Key,integer)
    print3a(integer)
    check3a(Key,integer)
    print4a(integer)
    check4a(Key,integer)
    print5a(integer)
    check5a(Key,integer)

clauses
    gomenu(A):-
        shiftwindow(1),
        clearwindow,
        print1a(A).

    /* Highlighting INTRO option */

    print1a(A):-
        shiftwindow(2),
        clearwindow,
        field_str(2,7,16,"1.INTRODUCTION"),
        field_attr(2,7,16,151),
        field_str(5,7,16,"2.KNOWLEDGE BASE"),
        field_attr(5,7,16,49),
        field_str(8,7,16,"3.FONAL"),
        field_attr(8,7,16,49),
        field_str(11,7,16,"4.OS  SHELL"),
        field_attr(11,7,16,49),
        field_str(14,7,16,"5.EXIT "),
        field_attr(14,7,16,49),
        readkey(Key),
        check1a(Key,A).

    /* checking the key pressed */

    check1a(Key,A):-
        Key=up,!,
        print5a(A).

    check1a(Key,A):-
        Key=down,!,
        print2a(A).

    check1a(Key,1):-
        Key=cr,!,
        shiftwindow(3).

    check1a(_,A):-
        print1a(A).
```

```
/* Highlighting KNOWLEDGE BASE option */

  print2a(A):-
       shiftwindow(2),
       clearwindow,
       field_str(2,7,16,"1.INTRODUCTION"),
       field_attr(2,7,16,49),
       field_str(5,7,16,"2.KNOWLEDGE BASE"),
       field_attr(5,7,16,151),
       field_str(8,7,16,"3.FONAL"),
       field_attr(8,7,16,49),
       field_str(11,7,16,"4.OS  SHELL"),
       field_attr(11,7,16,49),
       field_str(14,7,16,"5.EXIT "),
       field_attr(14,7,16,49),
       readkey(Key),
       check2a(Key,A).

  /* checking the key pressed */

  check2a(Key,A):-
       Key=up,!,
       print1a(A).

  check2a(Key,A):-
       Key=down,!,
       print3a(A).

  check2a(Key,2):-
       Key=cr,!,
       shiftwindow(4).

  check2a(_,A):-
       print2a(A).

  /* Highlighting FONAL option */

  print3a(A):-
       shiftwindow(2),
       clearwindow,
       field_str(2,7,16,"1.INTRODUCTION"),
       field_attr(2,7,16,49),
       field_str(5,7,16,"2.KNOWLEDGE BASE"),
       field_attr(5,7,16,49),
       field_str(8,7,16,"3.FONAL"),
       field_attr(8,7,16,151),
       field_str(11,7,16,"4.OS  SHELL"),
       field_attr(11,7,16,49),
       field_str(14,7,16,"5.EXIT "),
       field_attr(14,7,16,49),
       readkey(Key),
       check3a(Key,A).
```

```
/* checking the key pressed */

check3a(Key,A):-
    Key=up,!,
    print2a(A).

check3a(Key,A):-
    Key=down,!,
    print4a(A).

check3a(Key,3):-
    Key=cr,!,
    shiftwindow(5).

/* Highlighting OS SHELL option */

print4a(A):-
    shiftwindow(2),
    clearwindow,
    field_str(2,7,16,"1.INTRODUCTION"),
    field_attr(2,7,16,49),
    field_str(5,7,16,"2.KNOWLEDGE BASE"),
    field_attr(5,7,16,49),
    field_str(8,7,16,"3.FONAL"),
    field_attr(8,7,16,49),
    field_str(11,7,16,"4.OS  SHELL"),
    field_attr(11,7,16,151),
    field_str(14,7,16,"5.EXIT "),
    field_attr(14,7,16,49),
    readkey(Key),
    check4a(Key,A).

/* checking the key pressed */

check4a(Key,A):-
    Key=up,!,
    print3a(A).

check4a(Key,A):-
    Key=down,!,
    print5a(A).

check4a(Key,4):-
    Key=cr,!.

check4a(_,A):-
    print4a(A).

/* Highlighting EXIT option */

print5a(A):-
    shiftwindow(2),
    clearwindow,
    field_str(2,7,16,"1.INTRODUCTION"),
          field_attr(2,7,16,49),
          field_str(5,7,16,"2.KNOWLEDGE BASE"),
          field_attr(5,7,16,49),
          field_str(8,7,16,"3.FONAL"),
          field_attr(8,7,16,49),
          field_str(11,7,16,"4.OS  SHELL"),
          field_attr(11,7,16,49),
          field_str(14,7,16,"5.EXIT "),
          field_attr(14,7,16,151),
          readkey(Key),
          check5a(Key,A).

/* checking the key pressed */

check5a(Key,A):-
          Key=up,!,
          print4a(A).

check5a(Key,A):-
          Key=down,!,
          print1a(A).

check5a(Key,5):-
          Key=cr,!.

check5a(_,A):-
          print5a(A).
```

```
/* NLP for file operations */

predicates
  check_vp(string,string,string)
  check_np(string,string)
  check_np1(string,string)
  go_do
  go_err
  checksint(string,string,string)
  check_npr(string,string,string)
  findout(string,string)
  findswap(string,string)
  gorenam
  goswap

clauses
  /* Getting the sentence */

  go1(N,V):-
     clearwindow,
     write(" Enter the sentence : "),
     readln(S),
     shiftwindow(5),
     clearwindow,
     checksint(S,N,V).

  /*  SYNTAX checking */

  checksint(S,N,V):-  /* Npr + Verb + Prep + VP */
     fronttoken(S,H,S1),
     upper_lower(A,H),
     noun1(A),!,
     write("     Noun____________ ",A),nl,
     fronttoken(S1,H1,S2),
     upper_lower(A1,H1),
     verb1(A1),
     write("     Verb____________ ",A1),nl,
     fronttoken(S2,H2,S3),
     upper_lower(A2,H2),
     prep(A2),
     write("     Preposition_____ ",A2),nl,
     check_vp(S3,N,V),
     go_do.

  checksint(S,N,V):-  /* Aux + Npr + VP */
     fronttoken(S,H,S1),
     upper_lower(A,H),
     aux(A),!,
     write("     Aux.verb________ ",A),nl,
     check_npr(S1,N,V),
     go_do.
```

```
checksint(S,N,V):-    /*  VP  */
    check_vp(S,N,V),!,
    go_do.

checksint(_,N,V):-
    N="ERR",
    V="ERR",
    go_err.

check_npr(S1,N,V):-
     fronttoken(S1,H1,S2),
     upper_lower(A1,H1),
     noun1(A1),
     write("     Noun____________  ",A1),nl,
     check_vp(S2,N,V),!.

check_vp(S2,N,A2):-     /* VP check */
     fronttoken(S2,H2,S3),
     upper_lower(A2,H2),
     verb(A2),
     write("    Verb______________  ",A2),nl,
     check_np(S3,N).

check_np(S3,A4):-      /*  NP check   */
     fronttoken(S3,H3,S4),
     upper_lower(A3,H3),
     art(A3),
     write("     Article_________  ",A3),nl,
     check_np1(S4,A4).

check_np1(S4,A4):-     /*  NP check   */
     fronttoken(S4,H4,S5),
     upper_lower(A4,H4),
     noun(A4),!,
     write("     Noun_____________  ",A4),nl,
     S5="".

check_np1(S4,A5):-     /*  NP check   */
     fronttoken(S4,H4,S5),
     upper_lower(A4,H4),
     adj(A4),!,
     write("    Adjective________  ",A4),nl,
     fronttoken(S5,H5,S6),
     upper_lower(A5,H5),
     noun(A5),
     write("    Noun______________  ",A5),nl,
     S6="".

go_err:-    /*  ERROR routine */
     write(""),nl,
     write(" I am not able to parse any more."),nl,
      shiftwindow(6),
      clearwindow,
      write(" SYNTAX of your sentence may be wrong (OR)"),
      write(" not in the required format."),nl,nl,
      shiftwindow(80),
      clearwindow,
      field_str(0,30,50," Press any key to continue..."),
      shiftwindow(6),
      cursorform(-1,-1),
      readkey(_).

  go_do:-
      shiftwindow(6),
      clearwindow,
      write(""),nl,
      write(" SYNTAX of your sentence is correct."),nl,
      write(" Wait for my proceedings..."),nl,nl,
      sound(100,2750),
      shiftwindow(80),
      clearwindow,
      field_str(0,30,50," Press any key to continue..."),
      cursorform(-1,-1),
      readkey(_),
      cursorform(1,1).

/***  SEMANTICS check ***/

/* File creation */

  semant(N,V):-
      N="FILE",
      V="CREATE",!,
      system(create).

/* File deletion */

  semant(N,V):-
      N="FILE",
      V="DELETE",!,
      system(delete).

/* Files listing */

  semant(N,V):-
      N="FILES",
      V="LISTOUT",!,
      system(direct).

  semant(N,V):-
      N="FILES",
      V="LIST",!,
      system(direct).
```

```
/* File display */

  semant(N,V):-
      N="FILE",
      V="DISPLAY",!,
      system(display).



/* File copy */

  semant(N,V):-
      N="FILE",
      V="COPY",!,
      system(cop).

/* File rename */

  semant(N,V):-
      V="RENAME",
      N="FILE",!,
      gorenam.

/* File swaping */

  semant(N,V):-
      V="SWAP",
      N="FILES",!,
      goswap.

  semant(N,V):-
      V="SWAP",
      N="FILE",!,
      goswap.

/* File comparison */

  semant(N,V):-
      V="COMPARE",
      N="FILES",!,
      system(comp).

  semant(N,V):-
      V="COMPARE",
      N="FILE",!,
      system(comp).

/* File merging */

  semant(N,V):-
      V="MERGE",
      N="FILES",!,
      system(merge).
```

```
semant(N,V):-
    V="MERGE",
    N="FILE",!,
    system(merge).

/* SEMANTIC error */

semant(N,V):-
    N="ERR",
    V="ERR",!.

semant(_,_):-
    shiftwindow(6),
    clearwindow,
    write(""),nl,
    write(" Your sentence is SEMANTICALLY  wrong  (OR)"),
    write(" not required for this application."),nl,
    write(""),nl,
    write(" Press any key to continue..."),
    readkey(_).

/* Rename routine */

gorenam:-
    shiftwindow(4),
    clearwindow,
    write(" Enter the file name to be renamed : "),
    readln(FN),nl,
    write(" Enter the new file name  : "),
    readln(FN1),nl,nl,
    findout(FN,FN1),
    cursor(21,40),
    write("Press any key to continue..."),
    readkey(_).

  findout(FN,FN1):-
    not(existfile(FN)),!,
    upper_lower(F1,FN),
    write(" The file ",F1," is not existing...").

  findout(FN,FN1):-
    renamefile(FN,FN1),!,
    upper_lower(F1,FN),
    upper_lower(F2,FN1),
    write("The file ",F1," has been renamed to ",F2," ."

  /* Swaping the files */

  goswap:-
    shiftwindow(4),
    clearwindow,
   write(" Enter the file name(1) : "),
   readln(FN),nl,
   write(" Enter the file name(2) : "),
   readln(FN1),nl,nl,
   findswap(FN,FN1),
   cursor(21,40),
   write("Press any key to continue..."),
   readkey(_).

findswap(FN,FN1):-
   existfile(FN),
   existfile(FN1),!,
   renamefile(FN,x1x),
   renamefile(FN1,FN),
   renamefile(x1x,FN1),
   upper_lower(F1,FN),
   upper_lower(F2,FN1),
   write(" The files  ",F1," and ",F2," are swaped .").

findswap(FN,_):-
   not(existfile(FN)),!,
   upper_lower(F1,FN),
   write(" The file ",F1," is not existing..."),nl.

findswap(_,FN1):-
   not(existfile(FN1)),!,
   upper_lower(F1,FN1),
   write(" The file ",F1," is not existing..."),nl.
```

```
    /* Introduction */

predicates
  start
  find1(Key)
  find2(Key)
  find3(Key)
  find4(Key)
  type1
  type2
  type3
  type4
  type11
  type12
  type13
  find11(Key)
  find12(Key)
  find13(Key)
  type21
  find21(Key)
  type22
  find22(Key)

clauses
          /*  clauses section */
start:-
 shiftwindow(3),
 clearwindow,
 shiftwindow(11),
 type11.

start.

type11:-
     clearwindow,
     field_str(2,4,18,"1.FONAL------INTRO"),
     field_attr(2,4,18,151),
     field_str(5,4,18,"2.DATABASE---INTRO"),
     field_attr(5,4,18,49),
     field_str(8,4,18,"3.EXIT TO MAINMENU"),
     field_attr(8,4,18,49),
     readkey(A),
     find11(A).

type12:-
     clearwindow,
     field_str(2,4,18,"1.FONAL------INTRO"),
     field_attr(2,4,18,49),
     field_str(5,4,18,"2.DATABASE---INTRO"),
     field_attr(5,4,18,151),
     field_str(8,4,18,"3.EXIT TO MAINMENU"),
     field_attr(8,4,18,49),
     readkey(A),
     find12(A).
```

```
type13:-
      clearwindow,
      field_str(2,4,18,"1.FONAL------INTRO"),
      field_attr(2,4,18,49),
      field_str(5,4,18,"2.DATABASE---INTRO"),
      field_attr(5,4,18,49),
      field_str(8,4,18,"3.EXIT TO MAINMENU"),
      field_attr(8,4,18,151),
      readkey(A),
      write(A),
      find13(A).

find11(A):-
      A=up,!,
      type13.

find11(A):-
      A=down,!,
      type12.

find11(A):-
      A=cr,!,
      shiftwindow(3),
      clearwindow,
      shiftwindow(12),
      type1.

find11(_):-
    !,type11.

find12(A):-
      A=up,!,
      type11.

find12(A):-
      A=down,!,
      type13.

find12(A):-
      A=cr,!,
      shiftwindow(3),
      clearwindow,
      shiftwindow(13),
      type21.

find12(_):-
      !, type12.

find13(A):-
      A=up,!,
      type12.
```

```
find13(A):-
    A=down,!,
    type11.

find13(A):-
    A=cr,!.

find13(_):-
    type13.

type1:-
clearwindow,
field_str(0,46,6,"Page 1"),
field_attr(0,46,6,74),
field_str(2,5,45,"FONAL has been coined to imply File"),
field_attr(2,5,5,74),
field_str(3,1,50,"Operations in NAtural Language.It  performs
          well"),
field_str(4,1,50,"both as an English sentence parser as  well
          as an"),
field_str(5,1,50,"understanding   system   of   considerable
          merit.The "),
field_str(6,1,50,"software   includes   an   effective    and
          extensive"),
field_str(7,1,50,"parsing  algorithm together with a  Natural
          Language"),
field_str(8,1,50,"Understanding   System.The    Understanding
          system"),
field_str(9,1,50,"makes  optimum  use of  the  dictionary  in
          it's"),
field_str(10,1,50,"particular application."),
shiftwindow(80),
clearwindow,
field_str(0,20,40,"Press    Esc->Intromenu    ,     Pgdn->Next
          page"),
field_attr(0,20,40,74),
readkey(W),
clearwindow,
shiftwindow(12),
find1(W).

type2:-
clearwindow,
field_str(0,46,6,"Page 2"),
field_attr(0,46,6,74),
field_str(2,5,45,"FONAL is a software designed to "),
field_attr(2,5,5,74),
field_str(3,1,50,"execute  various file operations  with  the
          commands"),
field_str(4,1,50,"given in the  natural  language. FONAL  has
          a "),
field_attr(4,34,5,74),
field_str(5,1,50,"restricted  grammar  for  this   particular
          application."),
field_str(6,1,52,"The  grammar is restricted because  of  the
          complexity"),
field_str(7,1,50,"of the English Language."),
shiftwindow(80),
clearwindow,
field_str(0,11,59,"Press    Esc->Intromenu,    Pgup->Previous
          Page, Pgdn->Next page"),
field_attr(0,11,59,74),
readkey(W),
clearwindow,
shiftwindow(12),
find2(W).

type3:-
clearwindow,
field_str(0,46,6,"Page 3"),
field_attr(0,46,6,74),
field_str(2,1,50,"The input sentence would be command to"),
field_str(3,1,50,"carry   out  operations  like  creating   a
          file,deleting"),
field_str(4,1,50,"a  file,displaying the directory  etc.After
          the"),
field_str(5,1,50,"sentence  has been parsed,depending on  the
          verb and"),
field_str(6,1,50,"the   noun  used,the  operation  would   be
          executed.This"),
field_str(7,1,50,"is done by the understanding system."),
shiftwindow(80),
clearwindow,
field_str(0,11,59,"Press    Esc->Intromenu,    Pgup->Previous
          Page, Pgdn->Next page"),
field_attr(0,11,59,74),
readkey(W),
clearwindow,
shiftwindow(12),
find3(W).

type4:-
clearwindow,
field_str(0,46,6,"Page 4"),
field_attr(0,46,6,74),
field_str(3,3,50,"In this software interfacing of prolog"),
field_str(4,3,50,"with   the  C  language  has  been    done.C
          language"),
field_str(5,3,50,"has  been  chosen becos  it's  the  highest
          level"),
field_str(6,3,50,"language that can be interfaced easily."),
shiftwindow(80),
clearwindow,
field_str(0,17,45,"Press   Esc->Intromenu ,    Pgup->Previous
          Page"),
field_attr(0,17,45,74),
readkey(W),
clearwindow,
shiftwindow(12),
find4(W).

find1(W):-
  W=pgdn,!,
  type2.

find1(W):-
  W=esc,!,
  clearwindow,
  shiftwindow(3),
  clearwindow,
  shiftwindow(11),
  type11.

find1(_):-
  type1.

find2(W):-
  W=pgdn,!,
  type3.

find2(W):-
  W=pgup,!,
  type1.

find2(W):-
  W=esc,!,
  clearwindow,
  shiftwindow(3),
  clearwindow,
  shiftwindow(11),
  type11.

find2(_):-
  type2.

find3(W):-
  W=pgdn,!,
  type4.

find3(W):-
  W=pgup,!,
  type2.

find3(W):-
  W=esc,!,
  clearwindow,
   shiftwindow(3),
   clearwindow,
   shiftwindow(11),
   type11.

find3(_):-
   type3.

find4(W):-
   W=pgup,!,
   type3.

find4(W):-
   W=esc,!,
   clearwindow,
   shiftwindow(3),
   clearwindow,
   shiftwindow(11),
   type11.

find4(_):-
   type4.

type21:-
    clearwindow,
    field_str(0,46,6,"Page 1"),
    field_attr(0,46,6,74),
    field_str(2,1,50,"The Knowledge Base represents the memory
                of the"),
    field_str(3,1,50,"FONAL.In FONAL,the knowledge base is a
                DYNAMIC "),
    field_attr(3,1,5,74),
    field_attr(3,10,5,74),
    field_attr(3,40,7,77),
    field_str(4,1,50,"database i.e.,the facts are included
                during the"),
    field_str(5,1,50,"execution of the program. The addition
                of the "),
    field_str(6,1,50,"facts is done using the assertz
                predicate.The "),
    field_attr(6,25,7,77),
    field_str(7,1,50,"assertz predicate adds the fact at the
                end of the"),
    field_attr(7,1,7,77),
    field_str(8,1,50,"same type of the fact.The deletion of a
                 fact is"),
    field_str(9,1,50,"done by the retract predicate.The
                addition of the"),
    field_attr(9,13,7,77),
    field_str(10,1,50,"same fact is not allowed in FONAL.All
                the facts"),
    field_attr(1Ø,29,5,74),
    field_str(11,1,5Ø,"are stored in the file 'word.lst'."),
    field_attr(11,25,8,77),
    shiftwindow(8Ø),
    clearwindow,
    field_str(Ø,22,35,"Esc->Intromenu, Pgdn->Next page"),
    field_attr(Ø,22,35,74),
    readkey(A),
    clearwindow,
    shiftwindow(13),
    find21(A).

find21(A):-
    A=esc,!,
    clearwindow,
    shiftwindow(3),
    clearwindow,
    shiftwindow(11),
    type12.

find21(A):-
    A=pgdn,!,
    type22.

find21(_):-
    type21.

type22:-
      clearwindow,
      field_str(Ø,46,6,"Page 2"),
      field_attr(Ø,46,6,74),
      field_str(1,1,3Ø,"Valid file operations in FONAL"),
      field_attr(1,26,5,78),
      field_str(2,1,3Ø,"------------------------------"),
      field_attr(2,1,3Ø,77),
      field_str(4,2,1Ø,"1.CREATE"),
      field_str(4,2Ø,1Ø,"5.DELETE"),
      field_str(6,2,1Ø,"2.COPY"),
      field_str(6,2Ø,1Ø,"6.DISPLAY"),
      field_str(8,2,1Ø,"3.LISTOUT"),
      field_str(8,2Ø,1Ø,"7.LIST"),
      field_str(1Ø,2,1Ø,"4.MERGE"),
      field_str(1Ø,2Ø,1Ø,"8.SWAP"),
      field_str(4,38,1Ø,"9.RENAME"),
      field_str(6,37,1Ø,"1Ø.COMPARE"),
      shiftwindow(8Ø),
      clearwindow,
      field_str(Ø,2Ø,42,"Press  Esc->Intromenu,  Pgup->Previous
                  page"),
      field_attr(Ø,2Ø,42,74),
   readkey(A),
   clearwindow,
   shiftwindow(13),
   find22(A).

find22(A):-
     A=pgup,!,
     type21.

find22(A):-
     A=esc,!,
     clearwindow,
     shiftwindow(3),
     clearwindow,
     shiftwindow(11),
     type12.
find22(_):-
     type22.
```

```
/* KNOWLEDGE BASE operations */

predicates
  checkfact1(string)
  checkfact2(string)
  print1
  print2
  print3
  check1(Key)
  check2(Key)
  check3(Key)
 go31
 go32
 go33
 go34
 go35
 go36
 go37
 go38
 go41
 go42
 go43
 go44
 go45
 go46
 go47
 go48
 check11(Key)
 check12(Key)
 check13(Key)
 check14(Key)
 check15(Key)
 check16(Key)
 check17(Key)
 check18(Key)
 check21(Key)
 check22(Key)
 check23(Key)
 check24(Key)
 check25(Key)
 check26(Key)
 check27(Key)
 check28(Key)
 checknoun(string)
 checkpro(string)
 checkverb(string)
 checkkverb(string)
 checkprep(string)
 checkadj(string)
 checkart(string)
 checkaux(string)
 check1noun(string)
 check1pro(string)
```

```
check1verb(string)
check1kverb(string)
check1prep(string)
check1adj(string)
check1art(string)
check1aux(string)

clauses
    done:-
        clearwindow,nl,
        shiftwindow(7),
        print1.

    print1:-
        clearwindow,
        shiftwindow(8),
        clearwindow,
        field_str(2,5,20,"1.KNOWLEDGE ADDITION"),
        field_attr(2,4,22,151),
        field_str(5,5,20,"2.KNOWLEDGE DELETION"),
        field_attr(5,4,22,127),
        field_str(8,5,20,"3.EXIT TO MAIN MENU"),
        field_attr(8,4,22,127),
        readkey(Key),
        check1(Key).

    check1(Key):-
        Key=up,!,
        print3.

    check1(Key):-
        Key=down,!,
        print2.

    check1(Key):-
        Key=cr,!,
        clearwindow,
        shiftwindow(9),
        go31,
        clearwindow,
        shiftwindow(7),
        print1.

    check1(_):-
        print1.

    print2:-
        clearwindow,
        shiftwindow(8),
        clearwindow,
        field_str(2,5,20,"1.KNOWLEDGE ADDITION"),
        field_attr(2,4,22,127),
      field_str(5,5,20,"2.KNOWLEDGE DELETION"),
      field_attr(5,4,22,151),
      field_str(8,5,20,"3.EXIT TO MAIN MENU"),
      field_attr(8,4,22,127),
      readkey(Key),
      check2(Key).

check2(Key):-
      Key=up,!,
      print1.

check2(Key):-
      Key=down,!,
      print3.

check2(Key):-
      Key=cr,!,
      clearwindow,
      shiftwindow(10),
      go41,
      clearwindow,
      shiftwindow(7),
      print2.

check2(_):-
      print2.

print3:-
      clearwindow,
      shiftwindow(8),
      clearwindow,
      field_str(2,5,20,"1.KNOWLEDGE ADDITION"),
      field_attr(2,4,22,127),
      field_str(5,5,20,"2.KNOWLEDGE DELETION"),
      field_attr(5,4,22,127),
      field_str(8,5,20,"3.EXIT TO MAIN MENU"),
      field_attr(8,4,22,151),
      readkey(Key),
      check3(Key).

check3(Key):-
      Key=up,!,
      print2.

check3(Key):-
      Key=down,!,
      print1.

check3(Key):-
      Key=cr,!,
      clearwindow.

check3(_):-
      print3.
```

```
go31:-
    clearwindow,
    shiftwindow(9),
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,151),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit addition"),
    field_attr(0,7,26,74),
    readkey(Key),
    check11(Key).

check11(Key):-
    Key=left,!,
    clearwindow,
    go38.

check11(Key):-
    Key=right,!,
    go32.

check11(Key):-
    Key=up,!,
    go37.

check11(Key):-
    Key=down,!,
    go33.

check11(Key):-
    Key=esc,!.

check11(Key):-
    Key=cr,!,
    clearwindow,
    checkfact1(noun1),
    go31.

check11(_):-
    go31.
```

```
go32:-
    clearwindow,
    shiftwindow(9),
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,151),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit addition"),
    field_attr(0,7,26,74),
    readkey(Key),
    check12(Key).

 check12(Key):-
     Key=right,!,
     go33.

 check12(Key):-
     Key=left,!,
     go31.

 check12(Key):-
     Key=up,!,
     go38.

 check12(Key):-
     Key=down,!,
     go34.

 check12(Key):-
     Key=esc,!.


 check12(Key):-
     Key=cr,!,
     clearwindow,
     checkfact1(noun),
     go32.

 check12(_):-
     go32.
 go33:-
```

```
    shiftwindow(9),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,151),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit addition"),
    field_attr(0,7,26,74),
    readkey(Key),
    check13(Key).

check13(Key):-
    Key=left,!,
    clearwindow,
    go32.

check13(Key):-
    Key=right,!,
    clearwindow,
    go34.

check13(Key):-
    Key=up,!,
    clearwindow,
    go31.

check13(Key):-
    Key=down,!,
    clearwindow,
    go35.

check13(Key):-
    Key=esc,!,
    clearwindow.

check13(Key):-
    Key=cr,!,
    clearwindow,
    checkfact1(verb),
    go33.
```

```
check13(_):-
    clearwindow,
    go33.

go34:-
    shiftwindow(9),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,151),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit addition"),
    field_attr(0,7,26,74),
    readkey(Key),
    check14(Key).

check14(Key):-
    Key=right,!,
    clearwindow,
    go35.

check14(Key):-
    Key=left,!,
    clearwindow,
    go33.

check14(Key):-
    Key=up,!,
    clearwindow,
    go32.

check14(Key):-
    Key=down,!,
    clearwindow,
    go36.

check14(Key):-
    Key=esc,!,
    clearwindow.

check14(Key):-
```

```
    Key=cr,!,
    clearwindow,
    checkfact1(verb1),
    go34.

check14(_):-
    clearwindow,
    go34.

go35:-
    shiftwindow(9),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,151),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit addition"),
    field_attr(0,7,26,74),
    readkey(Key),
    check15(Key).

check15(Key):-
    Key=left,!,
    clearwindow,
    go34.

check15(Key):-
    Key=right,!,
    clearwindow,
    go36.

check15(Key):-
    Key=up,!,
    clearwindow,
    go33.

check15(Key):-
    Key=down,!,
    clearwindow,
    go37.
```

```
check15(Key):-
    Key=esc,!,
    clearwindow.

check15(Key):-
    Key=cr,!,
    clearwindow,
    checkfact1(adj),
    go35.

check15(_):-
    clearwindow,
    go35.

go36:-
    shiftwindow(9),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,151),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit addition"),
    field_attr(0,7,26,74),
    readkey(Key),
    check16(Key).

check16(Key):-
    Key=right,!,
    clearwindow,
    go37.

check16(Key):-
    Key=left,!,
    clearwindow,
    go35.

check16(Key):-
    Key=up,!,
    clearwindow,
    go34.
```

```
check16(Key):-
    Key=down,!,
    clearwindow,
    go38.

check16(Key):-
    Key=esc,!,
    clearwindow.

check16(Key):-
    Key=cr,!,
    clearwindow,
    checkfact1(art),
    go36.

check16(_):-
    clearwindow,
    go36.

go37:-
    shiftwindow(9),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,151),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit addition"),
    field_attr(0,7,26,74),
    readkey(Key),
    check17(Key).

check17(Key):-
    Key=left,!,
    clearwindow,
    go36.

check17(Key):-
    Key=right,!,
    clearwindow,
    go38.
```

```
check17(Key):-
    Key=up,!,
    clearwindow,
    go35.

check17(Key):-
    Key=down,!,
    clearwindow,
    go31.

check17(Key):-
    Key=esc,!,
    clearwindow.

check17(Key):-
    Key=cr,!,
    clearwindow,
    checkfact1(prep),
    go37.

check17(_):-
    clearwindow,
    go37.

go38:-
    shiftwindow(9),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,151),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit addition"),
    field_attr(0,7,26,74),
    readkey(Key),
    check18(Key).

check18(Key):-
    Key=right,!,
    clearwindow,
    go31.
```

```
check18(Key):-
    Key=left,!,
    clearwindow,
    go37.

check18(Key):-
    Key=up,!,
    clearwindow,
    go36.

check18(Key):-
    Key=down,!,
    clearwindow,
    go32.

check18(Key):-
    Key=esc,!,
    clearwindow.

check18(Key):-
    Key=cr,!,
    clearwindow,
    checkfact1(aux),
    go38.

check18(_):-
    clearwindow,
    go38.

go41:-
    shiftwindow(10),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,151),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press Esc to quit deletion"),
    field_attr(0,7,26,74),
    readkey(Key),
    check21(Key).
```

```
check21(Key):-
    Key=left,!,
    clearwindow,
    go48.

check21(Key):-
    Key=right,!,
    clearwindow,
    go42.

check21(Key):-
    Key=up,!,
    clearwindow,
    go47.

check21(Key):-
    Key=down,!,
    clearwindow,
    go43.

check21(Key):-
    Key=esc,!,
    clearwindow.

check21(Key):-
    Key=cr,!,
    clearwindow,
    checkfact2(noun1),
    go41.

check21(_):-
    clearwindow,
    go41.

go42:-
    shiftwindow(10),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,151),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),

    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit deletion"),
    field_attr(0,7,26,74),
    readkey(Key),
    check22(Key).

check22(Key):-
    Key=right,!,
    clearwindow,
    go43.

check22(Key):-
    Key=left,!,
    clearwindow,
    go41.

check22(Key):-
    Key=up,!,
    clearwindow,
    go48.

check22(Key):-
    Key=down,!,
    clearwindow,
    go44.

check22(Key):-
    Key=esc,!,
    clearwindow.

check22(Key):-
    Key=cr,!,
    clearwindow,
    checkfact2(noun),
    go42.

check22(_):-
    clearwindow,
    go42.

go43:-
    shiftwindow(10),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,151),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),

    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit deletion"),
    field_attr(0,7,26,74),
    readkey(Key),
    check23(Key).

check23(Key):-
    Key=left,!,
    clearwindow,
    go42.

check23(Key):-
    Key=right,!,
    clearwindow,
    go44.

check23(Key):-
    Key=up,!,
    clearwindow,
    go41.

check23(Key):-
    Key=down,!,
    clearwindow,
    go45.

check23(Key):-
    Key=esc,!,
    clearwindow.

check23(Key):-
    Key=cr,!,
    clearwindow,
    checkfact2(verb),
    go43.

check23(_):-
    clearwindow,
    go43.

go44:-
    shiftwindow(10),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),

    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,151),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit deletion"),
    field_attr(0,7,26,74),
    readkey(Key),
    check24(Key).

check24(Key):-
    Key=right,!,
    clearwindow,
    go45.

check24(Key):-
    Key=left,!,
    clearwindow,
    go43.

check24(Key):-
    Key=up,!,
    clearwindow,
    go42.

check24(Key):-
    Key=down,!,
    clearwindow,
    go46.

check24(Key):-
    Key=esc,!,
    clearwindow.

check24(Key):-
    Key=cr,!,
    clearwindow,
    checkfact2(verb1),
    go44.

check24(_):-
    clearwindow,
    go44.


go45:-
```

```
    shiftwindow(10),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,151),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit deletion"),
    field_attr(0,7,26,74),
    readkey(Key),
    check25(Key).

check25(Key):-
    Key=left,!,
    clearwindow,
    go44.

check25(Key):-
    Key=right,!,
    clearwindow,
    go46.

check25(Key):-
    Key=up,!,
    clearwindow,
    go43.

check25(Key):-
    Key=down,!,
    clearwindow,
    go47.

check25(Key):-
    Key=esc,!,
    clearwindow.

check25(Key):-
    Key=cr,!,
    clearwindow,
    checkfact2(adj),
    go45.
```

```
check25(_):-
    clearwindow,
    go45.

go46:-
    shiftwindow(10),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,151),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit deletion"),
    field_attr(0,7,26,74),
    readkey(Key),
    check26(Key).

check26(Key):-
    Key=right,!,
    clearwindow,
    go47.

check26(Key):-
    Key=left,!,
    clearwindow,
    go45.

check26(Key):-
    Key=up,!,
    clearwindow,
    go44.

check26(Key):-
    Key=down,!,
    clearwindow,
    go48.

check26(Key):-
    Key=esc,!,
    clearwindow.

check26(Key):-
```

```
    Key=cr,!,
    clearwindow,
    checkfact2(art),
    go46.

check26(_):-
    clearwindow,
    go46.

go47:-
    shiftwindow(10),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,151),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,49),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit deletion"),
    field_attr(0,7,26,74),
    readkey(Key),
    check27(Key).

check27(Key):-
    Key=right,!,
    clearwindow,
    go48.

check27(Key):-
    Key=left,!,
    clearwindow,
    go46.

check27(Key):-
    Key=up,!,
    clearwindow,
    go45.

check27(Key):-
    Key=down,!,
    clearwindow,
    go41.
```

```
check27(Key):-
    Key=esc,!,
    clearwindow.

check27(Key):-
    Key=cr,!,
    clearwindow,
    checkfact2(prep),
    go47.

check27(_):-
    clearwindow,
    go47.

go48:-
    shiftwindow(10),
    clearwindow,
    field_str(1,2,13,"1.PRONOUN"),
    field_attr(1,1,14,49),
    field_str(1,22,10,"2.NOUN"),
    field_attr(1,21,11,49),
    field_str(4,2,13,"3.KEYVERB"),
    field_attr(4,1,14,49),
    field_str(4,22,10,"4.VERB"),
    field_attr(4,21,11,49),
    field_str(7,2,13,"5.ADJECTIVE"),
    field_attr(7,1,14,49),
    field_str(7,22,10,"6.ARTICLE"),
    field_attr(7,21,11,49),
    field_str(10,2,13,"7.PREPOSITION"),
    field_attr(10,1,14,49),
    field_str(10,22,10,"8.AUX.VERB"),
    field_attr(10,21,11,151),
    shiftwindow(80),
    clearwindow,
    field_str(0,7,26,"Press ESC to quit deletion"),
    field_attr(0,7,26,74),
    readkey(Key),
    check28(Key).

check28(Key):-
    Key=right,!,
    clearwindow,
    go41.

check28(Key):-
    Key=left,!,
    clearwindow,
    go47.

check28(Key):-
    Key=up,!,
    clearwindow,
    go46.
```

```
check28(Key):-
    Key=down,!,
    clearwindow,
    go42.

check28(Key):-
    Key=esc,!,
    clearwindow.

check28(Key):-
    Key=cr,!,
    clearwindow,
    checkfact2(aux),
    go48.

check28(_):-
    clearwindow,
    go48.

/* fact addition */

checkfact1(F1):-
    F1="noun",!,
    shiftwindow(9),
    clearwindow,
    write("Enter the noun to be added"),nl,
    readln(R1),
    upper_lower(R,R1),
    checknoun(R).

checkfact1(F1):-
    F1="noun1",!,
    shiftwindow(9),
    clearwindow,
    write("Enter the pronoun to be added"),nl,
    readln(R1),
    upper_lower(R,R1),
    checkpro(R).

checkfact1(F1):-
    F1="verb",!,
    shiftwindow(9),
    clearwindow,
    write("Enter the keyverb to be added"),nl,
    readln(R1),
    upper_lower(R,R1),
    checkkverb(R).

checkfact1(F1):-
    F1="verb1",!,
    shiftwindow(9),
    clearwindow,


    write("Enter the verb to be added"),nl,
    readln(R1),
    upper_lower(R,R1),
    checkverb(R).

checkfact1(F1):-
    F1="adj",!,
    shiftwindow(9),
    clearwindow,
    write("Enter the adjective to be added"),nl,
    readln(R1),
    upper_lower(R,R1),
    checkadj(R).

checkfact1(F1):-
    F1="prep",!,
    shiftwindow(9),
    clearwindow,
    write("Enter the preposition to be added"),nl,
    readln(R1),
    upper_lower(R,R1),
    checkprep(R).

checkfact1(F1):-
    F1="art",!,
    shiftwindow(9),
    clearwindow,
    write("Enter the article to be added"),nl,
    readln(R1),
    upper_lower(R,R1),
    checkart(R).

checkfact1(F1):-
    F1="aux",!,
    shiftwindow(9),
    clearwindow,
    write("Enter the auxillary to be added"),nl,
    readln(R1),
    upper_lower(R,R1),
    checkaux(R).

checknoun(R):-
    not(noun(R)),!,
    assertz(noun(R)),
    cursor(5,4),
    write("The fact NOUN(",R,") "),
    cursor(7,7),
    write("has been added."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").


checknoun(R):-
```

```
     cursor(5,4),
     write("The fact NOUN(",R,") "),
     cursor(7,7),
     write("is already present."),
     cursor(12,8),
     write("Press any key to continue"),
     readkey(_),
     clearwindow.

checkpro(R):-
     not(noun1(R)),!,
     assertz(noun1(R)),
     cursor(5,4),
     write("The fact PRONOUN(",R,") "),
     cursor(7,7),
     write("has been added."),
     cursor(12,8),
     write("Press any key to continue"),
     readkey(_),
     clearwindow,
     save("word.lst").

checkpro(R):-
     cursor(5,4),
     write("The fact PRONOUN(",R,") "),
     cursor(7,7),
     write("is already present."),
     cursor(12,8),
     write("Press any key to continue"),
     readkey(_),
     clearwindow.

checkverb(R):-
     not(verb1(R)),!,
     assertz(verb1(R)),
     cursor(5,4),
     write("The fact VERB(",R,") "),
     cursor(7,7),
     write("has been added."),
     cursor(12,8),
     write("Press any key to continue"),
     readkey(_),
     clearwindow,
     save("word.lst").

checkverb(R):-
     cursor(5,4),
     write("The fact VERB(",R,") "),
     cursor(7,7),
     write("is already present."),
     cursor(12,8),
     write("Press any key to continue"),
     readkey(_),
     clearwindow.
```

```
checkkverb(R):-
    not(verb(R)),!,
    assertz(verb(R)),
    cursor(5,4),
    write("The fact KEYVERB(",R,") "),
    cursor(7,7),
    write("has been added."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").

checkkverb(R):-
    cursor(5,4),
    write("The fact KEYVERB(",R,") "),
    cursor(7,7),
    write("is already present."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

checkart(R):-
    not(art(R)),!,
    assertz(art(R)),
    cursor(5,4),
    write("The fact ARTICLE(",R,") "),
    cursor(7,7),
    write("has been added."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").

checkart(R):-
    cursor(5,4),
    write("The fact ARTICLE(",R,") "),
    cursor(7,7),
    write("is already present."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

checkadj(R):-
    not(adj(R)),!,
    assertz(adj(R)),
    cursor(5,4),
    write("The fact ADJECTIVE(",R,") "),
    cursor(7,7),
    write("has been added."),

    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").

checkadj(R):-
    cursor(5,4),
    write("The fact ADJECTIVE(",R,") "),
    cursor(7,7),
    write("is already present."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

checkprep(R):-
    not(prep(R)),!,
    assertz(prep(R)),
    cursor(5,4),
    write("The fact PREPOSITION(",R,") "),
    cursor(7,7),
    write("has been added."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").

checkprep(R):-
    cursor(5,4),
    write("The fact PREPOSITION(",R,") "),
    cursor(7,7),
    write("is already present."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

checkaux(R):-
    not(aux(R)),!,
    assertz(aux(R)),
    cursor(5,4),
    write("The fact AUXVERB(",R,") "),
    cursor(7,7),
    write("has been added."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").



checkaux(R):-
```

```
    cursor(5,4),
    write("The fact AUXVERB(",R,") "),
    cursor(7,7),
    write("is already present."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

/* fact deletion */

checkfact2(F2):-
    F2="noun",!,
    shiftwindow(10),
    clearwindow,
    write("Enter the noun to be deleted"),nl,
    readln(R1),
    upper_lower(R,R1),
    check1noun(R).

checkfact2(F2):-
    F2="noun1",!,
    shiftwindow(10),
    clearwindow,
    write("Enter the pronoun to be deleted"),nl,
    readln(R1),
    upper_lower(R,R1),
    check1pro(R).

checkfact2(F2):-
    F2="verb",!,
    shiftwindow(10),
    clearwindow,
    write(" Enter the keyverb  to be deleted"),nl,
    readln(R1),
    upper_lower(R,R1),
    check1kverb(R).

checkfact2(F2):-
    F2="verb1",!,
    shiftwindow(10),
    clearwindow,
    write("Enter the verb to be deleted"),nl,
    readln(R1),
    upper_lower(R,R1),
    check1verb(R).

checkfact2(F2):-
    F2="adj",!,
    shiftwindow(10),
    clearwindow,


    write("Enter the adjective to be deleted"),nl,
        readln(R1),
        upper_lower(R,R1),
        check1adj(R).

    checkfact2(F2):-
        F2="prep",!,
        shiftwindow(10),
        clearwindow,
        write("Enter the preposition to be deleted"),nl,
        readln(R1),
        upper_lower(R,R1),
        check1prep(R).

    checkfact2(F2):-
        F2="art",!,
        shiftwindow(10),
        clearwindow,
        write("Enter the article to be deleted"),nl,
        readln(R1),
        upper_lower(R,R1),
        check1art(R).

    checkfact2(F2):-
        F2="aux",!,
        shiftwindow(10),
        clearwindow,
        write("Enter the auxillary verb to be deleted"),nl,
        readln(R1),
        upper_lower(R,R1),
        check1aux(R).

    check1noun(R):-
        noun(R),!,
        retract(noun(R)),
        cursor(5,4),
        write("The fact NOUN(",R,") "),
        cursor(7,7),
        write("has been deleted."),
        cursor(12,8),
        write("Press any key to continue"),
        readkey(_),
        clearwindow,
        save("word.lst").

    check1noun(R):-
        cursor(5,4),
        write("The fact NOUN(",R,") "),
        cursor(7,7),
        write("is not available."),
        cursor(12,8),
        write("Press any key to continue"),
        readkey(_),
        clearwindow.

    check1pro(R):-
    noun1(R),!,
    retract(noun1(R)),
    cursor(5,4),
    write("The fact PRONOUN(",R,") "),
    cursor(7,7),
    write("has been deleted."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").

check1pro(R):-
    cursor(5,4),
    write("The fact PRONOUN(",R,") "),
    cursor(7,7),
    write("is not available."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

check1kverb(R):-
    verb(R),!,
    retract(verb(R)),
    cursor(5,4),
    write("The fact KEYVERB(",R,") "),
    cursor(7,7),
    write("has been deleted."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").

check1kverb(R):-
    cursor(5,4),
    write("The fact KEYVERB(",R,") "),
    cursor(7,7),
    write("is not available."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

check1verb(R):-
    verb1(R),!,
    retract(verb1(R)),
    cursor(5,4),
    write("The fact VERB(",R,") "),
    cursor(7,7),
    write("has been deleted."),


    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").

checklverb(R):-
    cursor(5,4),
    write("The fact VERB(",R,") "),
    cursor(7,7),
    write("is not available."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

checklprep(R):-
    prep(R),!,
    retract(prep(R)),
    cursor(5,4),
    write("The fact PREPOSITION(",R,") "),
    cursor(7,7),
    write("has been deleted."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").

checklprep(R):-
    cursor(5,4),
    write("The fact PREPOSITION(",R,") "),
    cursor(7,7),
    write("is not available."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

checklart(R):-
    art(R),!,
    retract(art(R)),
    cursor(5,4),
    write("The fact ARTICLE(",R,") "),
    cursor(7,7),
    write("has been deleted."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").



checklart(R):-
```

```
    cursor(5,4),
    write("The fact ARTICLE(",R,") "),
    cursor(7,7),
    write("is not available."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

checkladj(R):-
    adj(R),!,
    retract(adj(R)),
    cursor(5,4),
    write("The fact ADJECTIVE(",R,") "),
    cursor(7,7),
    write("has been deleted."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").

checkladj(R):-
    cursor(5,4),
    write("The fact ADJECTIVE(",R,") "),
    cursor(7,7),
    write("is not available."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.

checklaux(R):-
    aux(R),!,
    retract(aux(R)),
    cursor(5,4),
    write("The fact AUXVERB(",R,") "),
    cursor(7,7),
    write("has been deleted."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow,
    save("word.lst").

checklaux(R):-
    cursor(5,4),
    write("The fact AUXVERB(",R,") " ),
    cursor(7,7),
    write("is not available."),
    cursor(12,8),
    write("Press any key to continue"),
    readkey(_),
    clearwindow.
```

```
/* To do graphics */

 # include <stdio.h>
 # include <string.h>
 # include <graphics.h>

 void g();
    int driver,x,w,y,z,i,mode;

 void g(val)
    char val[15];
       {
  start: driver=CGA;
     mode=CGAC1;
     initgraph(&driver,&mode,"");
     x=1;
     w=1;
     y=319;
     z=199;
     for(i=1;i<=30;i++)
      {
    rectangle(w,x,y,z);
    x=x+3;
    y=y-3;
    z=z-3;
    w=w+3;
    delay(100);
      }
     gotoxy(13,13);
     printf("%s",val);
     delay(1000);
     if (getch())
     {
     textmode();
     }
 }
 /*   To merge any two given files   */

 # include "grap.c"
 # include <stdio.h>

main()
  {
    FILE *fp1,*fp2,*fp3;
    char fn1[14],fn2[14],fn3[14],str[80];
    clrscr();
    g("    FILE MERGE  ");
    clrscr();
    printf("\n Enter the first filename : ");
    scanf("%s",fn1);
    fp1=fopen(fn1,"r");
    if (fp1=='\0')
     {
       printf("\n The file is not existing...");
       goto end;
     }
    printf("\n Enter the second filename : ");
    scanf("%s",fn2);
    fp2=fopen(fn2,"r");
    if (fp2=='\0')
     {
       printf("\n The file is not existing...");
       goto end;
     }
    printf("\n Enter the merged filename : ");
    scanf("%s",fn3);
    fp3=fopen(fn3,"w");
    if (fp3=='\0')
     {
       printf("\n Error in opening the file.");
       goto end;
     }
    while(!feof(fp1))
       {
             fgets(str,80,fp1);
             fprintf(fp3,"%s",str);
       }

   while(!feof(fp2))
       {
             fgets(str,80,fp2);
             fprintf(fp3,"%s",str);
       }
   printf("\n Merging has been done successfully...\n");

end:;
 fclose(fp1);
 fclose(fp2);
 fclose(fp3);
 }
```

```
/* program to display the file contents */

# include <stdio.h>
# include "grap.c"

main()
{

  FILE *fp1;
  char str[80],fn[14];
  clrscr();
  g("  FILE DISPLAY ");
  clrscr();
  printf(" Enter the file to be displayed : ");
  scanf("%s",fn);
  fp1=fopen(fn,"r");
  while(!feof(fp1))
     {
       fgets(str,80,fp1);
       printf("%s",str);
     }
printf("\n The file has been displayed ");
fclose(fp1);
}
```

```
/* program to copy a file  */

# define NULL '\0'
# include <stdio.h>
# include "b:\c\grap.c"

main()
{
  FILE *fp1,*fp2;
  char str[80],fn1[25],fn2[25];

  clrscr();
  g(" FILE  COPY ");
  clrscr();

  printf("\n Enter the source file name : ");
  scanf("%s",fn1);
  printf("\n Enter the destination file name : ");
  scanf("%s",fn2);

  fp1=fopen(fn1,"r");
  if(fp1==NULL)
    {
      printf("\n The source file %s is not existing.",fn1);
    }
  else
  {
      fp2=fopen(fn2,"w");
      if(fp2==NULL)
      {
 printf("\nError in creating the destination file %s .",fp2)
      }
      else
      {
    while(!feof(fp1))
     {
       fgets(str,80,fp1);
       fputs(str,fp2);
     }
    fclose(fp2);
    gotoxy(50,23);
    printf("\n The file has been copied. ");
       }
   fclose(fp1);
 }
 }
```

```
    /*  Routine to delete a file */

    # define MAX 25
    # define MAXLEN 255
    # define NULL  '\0'
    # include<stdio.h>
    # include<ctype.h>
    # include<string.h>
    # include<math.h>
    # include "grap.c"

char fn[8];
    int n;
    struct
     {
   char fld[MAX][10];
     } rec[MAX];
     char field[MAX][10];
     int fld_no;
     int rec_no;

main()
     {
     int i,j,no;
     FILE *f1;

clrscr();
     g(" FILE DELETION ");
clrscr();

printf("\n Enter the filename to be deleted(with path) : ")
     scanf("%s",fn);
     f1=fopen(fn,"r");
     if (f1==NULL)
     {printf(" The file %s is not existing. \n",fn);
     }
  else
   {
   unlink(fn);
   printf(" The file %s has been deleted . ",fn);
   }
}
```

```
/* To list the files in the directory */

# include <dos.h>
# include <dir.h>
# include <string.h>
# include "grap.c"

main()
{

struct ffblk *ff;
char *p,*q,*r;
int i;

   clrscr();
   g("DIRECTORY LISTING");
   clrscr();

printf("\n Do you know how to enter the path (Y/N) ? ");
if (toupper(getch())=='Y')
  {
   printf("\n Enter the path : ");
   scanf("%s",p);
  }
else
  {
    printf(" Enter the directory name(A,B,C,D,E or F) : ");
    scanf("%s",p);
    strcat(p,":");
rep:printf("\n Now you can see %s\\*.* ",p);
printf("Do you want enter the subdirectory of %s (Y/N)?",p);
    if (toupper(getch())=='Y')
       {
     printf(" \n Enter the sub directory name : ");
     scanf("%s",q);
     strcat(p,"\\");
     strcat(p,q);
     goto rep;
     }
   printf("\n  Do  you  want the  files  with  any  particular
              extension (Y/N) ?");
      if (toupper(getch())=='Y')
        {
      printf(" \n Enter the extension : ");
      scanf("%s",r);
      strcat(p,"\\*.");
      strcat(p,r);
        }
      else
        {
      strcat(p,"\\*.*");
        }
   }
  clrscr();
  printf("\n LISTING OF %s\n",p);
  if((findfirst(p,ff,Ø))==Ø)
   {
       printf("\n");
       i=Ø;
     do{
     printf("%s\n",ff->ff_name);
     i=i+1;
       }while((findnext(ff))!=(-1));
      printf("\n\n (%d) files in the directory \n\n ",i);
   }
   else
   printf("\n  No  files in the directory (or) drive  is  not
                existing ");
}
```

```
   else
  {
    gotoxy(45,24);
    printf(" The file is being created... ");
    delay(3000);
    clrscr();
    printf(" \n Do you want to enter fields (Y/N) ? ");
     if (toupper(getch())==´Y´)
     {
      clrscr();
      printf("\n Enter the number of fields: ");
      scanf("%d",&no_fld);
      for(i=1;i<=no_fld;i++)
       {
        printf("\n Enter the %d field name: ", i);
        scanf("%s",field[i]);
        fprintf(f1," %s",field[i]);
        fprintf(f1,"%s","\0");
        fprintf(f1,"\t");
       }
     }
     printf(" \n Do you want to enter Records (Y/N) ? ");
     if (toupper(getch())==´Y´)
     {
      clrscr();
      printf("\n Enter the number of records: ");
      scanf("%d",&rec_no);
      for(j=1;j<=rec_no;j++)
       {
      for(i=1;i<=no_fld;i++)
       {
        printf("\n Enter the data for - %d - record´s %s :",
                                         j,field[i]);
        scanf("%s",rec[j].fld[i]);
       }}
         for(j=1;j<=rec_no;j++)
      {
      fprintf(f1,"\n");
      for(i=1;i<=no_fld;i++)
       {
            fprintf(f1," %s\t",rec[j].fld[i]);
       }}

      }
      fprintf(f1,"\0");
      fprintf(f1,"\n");
      gotoxy(50,23);
      printf(" The file has been created "   );
      fclose(f1);
      delay(3000);
      }
}
```