INTERACT ( )  

DENDRAL PROGRAM FOR SPECIFYING DECOMPOSITIONS OF ORGANIC

GRAPHS. YOU ARE TO REPPLY TO QUESTIONS ASKED BY THE

COMPUTER, WHICH ARE INDICATED BY A ?. TYPE IN THE ANSWER,

THEN HIT THE RETURN KEY. IF YOU DO NOT UNDERSTAND A

QUESTION, TYPE A 0 AS THE ANSWER

READY?

TYPE Y FOR YES, N FOR NO

READY?

GRAPHNAME?

TYPE NAME BY WHICH GRAPH IS TO BE IDENTIFIED

GRAPHNAME?alcohol

GRAPHNAME IS ALCOHOL

GRAPHLIST?

TYPE IN THE GRAPH STRUCTURE IN VALID FORM

GRAPHLIST?????S=... O C O O O

GRAPH C=... O C O O O K.

DO YOU WISH TO SPECIFY AN ACTION NAME?

TYPE IN Y IF YES, N IF NO.

DO YOU WISH TO SPECIFY AN ACTION NAME?

ACTION NAME?

TYPE NAME YOU WISH TO GIVE FOLLOWING ACTION

ACTION NAME?action1

ACTION NAME IS ACTION1

A NEW OPERAND?

TYPE Y FOR YES, OR N FOR NO

A NEW OPERAND?

OPERAND?alcohol1c

ERROR! INVALID OPERAND NAME

A NEW OPERAND?

OPERAND?migrate from 2 to ;

ERROR! INVALID OR INSUFFICIENT ARGUMENTS

A NEW OPERAND?

OPERAND?migrate from 2 to 4

COLLECTED 455 CELLS AND STACK HAS 1215 UNITS LEFT. (USING 10 PAGES) %

VALID OPERAND AND ARGUMENTS ACCEPTED

END OF ACTION?

TYPE Y IF YES OR N IF NO

END OF ACTION?

A NEW OPERAND?

OPERAND?breakbond between 5 and 3

VALID OPERAND AND ARGUMENTS ACCEPTED

END OF ACTION?

A NEW OPERAND?

OPERAND?making by joining 3 to 1 with bond of 1

VALID OPERAND AND ARGUMENTS ACCEPTED

END OF ACTION?

RELATIVE FREQUENCY?

TYPE IN RELATIVE FREQUENCY OF ACTION DEFINED ABOVE

AS A NUMBER BETWEEN 0 AND 100

RELATIVE FREQUENCY?23

DO YOU WISH THE ABOVE ACTION IN RULES LIST?

TYPE IN Y IF YES, OR N FOR NO

DO YOU WISH THE ABOVE ACTION IN RULES LIST?

COLLECTED 490 CELLS AND STACK HAS 1223 UNITS LEFT. (USING 10 PAGES) %

NEW ACTION?

END OF ACTION DEFINITIONS FOR ALCOHOL ?

TYPE Y FOR YES, OR N FOR NO

END OF ACTION DEFINITIONS FOR ALCOHOL ?

NEW ACTION?

TYPE Y FOR YES, N FOR NO

NEW ACTION?
END OF ACTION DEFINITIONS FOR ALCOHOL

TYPE Y FOR YES, OR N FOR NO

SIGN OFF?

NEW GRAPH?

TYPE Y FOR YES, N FOR NO

SIGN OFF?

NEW GRAPH?

SIGN OFF?

*** RULE LIST CONTENTS ***

NIL

ACTION PROGRAM TERMINATED

NOW EXIT FROM LISP AND LOGOFF, PLEASE

NIL

? exit

:EXITING STANFORD/LISP

? logoff
Program listing
1. DEFINE ((INTERACT (LAMBDA ()
2. (PROG (IN GNAME GLIST ANAME RELP ACTLIST ACTDEF OPD OPD1 RULES RULES1)
3. (PRINT (QUOTE $$$SENDRAL PROGRAM FOR SPECIFYING DECOMPOSITIONS OF ORGANIC))
4. (PRINT (QUOTE $$$GRAPH, WHICH ARE INDICATED BY A ?. TYPE IN THE ANSWER,)))
5. (PRINT (QUOTE $$$COMPUTER, WHICH ARE INDICATED BY A ?. TYPE IN THE ANSWER,)))
6. (PRINT (QUOTE $$$THEN HIT THE RETURN KEY. IF YOU DO NOT UNDERSTAND A $))
7. (PRINT (QUOTE $$$QUESTION, TYPE A Q AS THE ANSWER)))
8. STEPA1
9. (TREAD 0)
10. (PRINI (QUOTE $$$READY? $))
11. (SETQ IN (TREAD T))
12. (COND ((EQ IN (QUOTE Y)) (GO STEPA2)))
13. (COND ((EQ IN (QUOTE N)) (GO STEPA1)))
14. (PRINT (QUOTE $$$TYPE Y FOR YES, N FOR NO))
15. (GO STEPA1)
16. STEPA2
17. (TREAD 0)
18. (PRINI (QUOTE $$$GRAPHNAME? $))
19. (SETQ GNAME (TREAD T))
20. (COND ((NOT (EQ GNAME (QUOTE Q))) (GO STEPB1)))
21. (PRINT (QUOTE $$$TYPE NAME BY WHICH GRAPH IS TO BE IDENTIFIED))
22. (GO STEPA2)
23. STEPB1
24. (PRINI (QUOTE $$$GRAPHNAME IS $))
25. (PRINT GNAME)
26. STEPB2
27. (TREAD 0)
28. (PRINI (QUOTE $$$GRAPHLIST? $))
29. (SETQ GLIST (TREAD T))
30. (COND ((NOT (EQ GLIST (QUOTE Q))) (GO STEPC1)))
31. (PRINT (QUOTE $$$TYPE IN THE GRAPH STRUCTURE IN VALID FORMS))
32. (GO STEPB2)
33. STEPC1
34. (PRINI (QUOTE $$$GRAPH $))(PRINT GLIST)(PRINT (QUOTE $$$ O.K.$))
35. (SETQ ANAME (QUOTE WO))
36. (SETQ ACTDEF (LIST (QUOTE PROG) NIL))
37. STEPD1
38. (TREAD 0)
39. (PRINI (QUOTE $$$DO YOU WISH TO SPECIFY AN ACTION NAME? $))
40. (SETQ IN (TREAD T))
41. (COND ((EQ IN (QUOTE N)) (GO STEPD4)))
42. (COND ((EQ IN (QUOTE Y)) (GO STEPD2)))
43. (PRINT (QUOTE $$$TYPE IN Y IF YES, N IF NO))
44. (GO STEPD1)
45. STEPD2
46. (TREAD 0)
47. (PRINI (QUOTE $$$ACTION NAME? $))
48. (SETQ ANAME (TREAD T))
49. (COND ((NOT (EQ ANAME (QUOTE Q))) (GO STEPD3)))
50. (PRINT (QUOTE $$$TYPE NAME YOU WISH TO GIVE FOLLOWING ACTION$))
51. (GO STEPD2)
52. STEPD3
53. (PRINI (QUOTE $$$ACTION NAME IS $))
54. (SETQ IN (APPEND ACTLIST (LIST ANAME)))
55. (SETQ ACTLIST IN)
56. (PRINT ANAME)
57. STEPD4
58. (TREAD 0)
(PRIN1 (QUOTE $$$$A NEW OPERAND? $))
(SETQ IN (TREAD T))
(COND ((NOT (EQ IN (QUOTE Q))) (GO STEPF1)))
(PRINT (QUOTE $$$$TYPE Y FOR YES, OR N FOR NO))
(GO STEPD4)

STEP1

(COND ((NOT (EQ IN (QUOTE Y))) (GO STEPE4)))

STEP2

(TREAD O)
(PRIN1 (QUOTE $$$$OPERAND? $))
(SETQ IN (TREAD T))
(COND ((NOT (EQ IN (QUOTE Q))) (GO STEPE3)))
(PRINT (QUOTE $$$$TYPE A VALID OPERAND-NAME AND CORRESPONDING ARGUMENTS))
(QUERY $$$$TYPE A VALID OPERAND-NAME AND CORRESPONDING ARGUMENTS)

STEP3

(setq opd (actionfm in actlist))
(cond ((atom opd) (go step3b))
((atom (car opd)) (go step3c)))
(setq opd1 (append actdef opd))
(setq actdef opd1)
(go step3d)

STEP3A

(cond ((eq opd (quote err)) (print (quote $$$$ERROR INVALID OR INSUFFICIENT ARGUMENTS)))
(t (print (quote $$$$ERROR INVALID OPERAND NAME))))

(go stepd4)

STEP3C

(setq opd1 (append actdef (list opd)))
(setq actdef opd1)

STEP3D

(print (quote $$$$VALID OPERAND AND ARGUMENTS ACCEPTED))

STEP4

(tread o)
(prin1 (quote $$$$END OF ACTION? $))
(SETQ IN (TREAD T))
(cond ((not (eq in (QUOTE Q))) (GO STEP5)))
(print (QUOTE $$$$TYPE Y IF YES OR N IF NO))
(QUERY $$$$TYPE Y IF YES OR N IF NO)

STEP5

(cond ((eq in (QUOTE N)) (GO STEPD4)))

STEP1

(TREAD O)
(prin1 (quote $$$$RELATIVE FREQUENCY? $))
(setq relp (tread t))
(cond ((not (eq relp (QUOTE Q))) (go stepf2)))
(print (quote $$$$TYPE IN RELATIVE FREQUENCY OF ACTION DEFINED ABOVE))
(print (quote $$$$AS A NUMBER BETWEEN 0 AND 100))

STEPF1

(go stepf1)

STEPF2

(TREAD O)
(prin1 (quote $$$$DO YOU WISH THE ABOVE ACTION IN RULES LIST? $))
(setq in (tread t))
(cond ((not (eq in (QUOTE Q))) (go stepg1)))
(print (quote $$$$TYPE IN Y FOR YES, OR N FOR NO))
(QUERY $$$$TYPE IN Y FOR YES, OR N FOR NO)

STEPG1

(cond ((eq in (QUOTE Y)) (go stepg1b)))
(setq opd1 (replace (quote qqqq) aname actlist))
(SETQ ANAME OPD1)
(go stepg1c)

STEPG1B
(SETQ RULESI (APPEND RULES (LIST (LIST GNAME ANAME REL?))))

(SETQ RULES RULESI)

(SETQ OPDI1 (APPEND (LIST (QUOTE LAMBDA) NIL) (LIST ACTDEF)))

(DEFLIST ((ANAME OPDI1) (QUOTE FXX)))

(STEP1C

(TREAD 0)

(PRINT (QUOTE $$$NEW ACTION? $))

(SETQ IN (TREAD T))

(COND ((EQ IN (QUOTE Y)) (GO STEPD1)))

(PRINT (QUOTE $$$TYPE Y FOR YES, N FOR NO$))

(GO STEP1G)

(STEP2

(TREAD 0)

(PRINT (QUOTE $$$SEND OF ACTION DEFINITIONS FOR $))

(PRINT (QUOTE ($NAME) (PRINT (QUOTE $$$ ? $))

(SETQ IN (TREAD T))

(COND ((EQ IN (QUOTE Y)) (GO STEPD1)))

(PRINT (QUOTE $$$TYPE Y FOR YES, OR N FOR NO$))

(GO STEPD2)

(STEP3

(TREAD 0)

(PRINT (QUOTE $$$NEW GRAPH? $))

(PRINT (QUOTE ($NAME) (PRINT (QUOTE $$$ ? $))

(SETQ IN (TREAD T))

(COND ((EQ IN (QUOTE Y)) (GO STEPD2)))

(PRINT (QUOTE $$$TYPE Y FOR YES, OR N FOR NO$))

(GO STEPD3)

(STEPH1

(TREAD 0)

(PRINT (QUOTE $$$SIGN OFF? $))

(SETQ IN (TREAD T))

(COND ((EQ IN (QUOTE Y)) (GO STEPD2)))

(PRINT (QUOTE $$$TYPE Y FOR YES, OR N FOR NO$))

(GO STEPD3)

(STEPH2

(PRINT (QUOTE $$$ $))

(PRINT (QUOTE $$$ $))

(PRINT (QUOTE $$$ RULE LIST CONTENTS $$))

(PRINT RULES)

(STEPH3

(PRINT (QUOTE $$$ $))

(COND ((null RULESI) (GO STEPH4)))

(PRINT (GET (CADAR RULESI) (QUOTE FXX)))

(SETQ OPDI1 (CDR RULESI))

(GO STEPH3)

(STEPH4

(PRINT (QUOTE $$$PROGRAM TERMINATED$))

(PRINT (QUOTE $$$NOW EXIT FROM LISP AND LOGOFF, PLEASE$))

(RETURN NIL))))

(DEFLIST ((OPLIST (FROM TO))) MIGRATE)

(DEFLIST ((OPLIST ((BY JOINING) TO) (WITH BOND OF))) MAKING)

(DEFLIST ((OPLIST (BETWEEN AND))) BREAK)

.DEFINE ((REPLACE (LAMBDA (A B X))

(COND ((ATOM X) (COND ((EQUAL B X) A) (T X)))))

(T (CONS (REPLACE A B) (CAR X)))

(QUOTE (REPLACE A B) (CDR X)))))

.DEFINE ((ELEMENT (LAMBDA (A B)
(COND ((NULL B) NIL)
  ((EQUAL A (CAR B)) T)
  ((T (ELEMENT A (CDR B)))))))
(DEFINITE ((ACTIONEM (LAMBDA (X ACLIST)
  (PROG (X2 CONN EC M1 M2 INTERM XSTOR N1 N2 PAIRS PAIRINTER)
    (SETQ XSTOR (LIST X))
    (SETQ CONN EC (GET (QUOTE OPLIST) X))
    (COND ((NULL CONN EC) (GO STEPS))
    
    (STEP 1)
    (SETQ X2 (TREAD T))
    (COND ((NUMBERP X2) (GO STEP4))
    ((ATOM (CAR CONN EC)) (GO STEP3))
    
    (STEP 1A)
    (COND ((NULL (CAR CONN EC)) (RETURN (QUOTE ERR1))))
    ((EQ X2 (CAAR CONN EC)) (GO STEP2A))
    (SETQ INTERM (APPEND (LIST (CDAR CONN EC)) (CDR CONN EC)))
    (SETQ CONN EC INTERM)
    
    (STEP 2A)
    (SETQ INTERM (APPEND (LIST (CDAR CONN EC)) (CDR CONN EC)))
    (SETQ CONN EC INTERM)
    (GO STEP1))
    
    (STEP 3)
    (COND ((NOT (EQ X2 (CAR CONN EC))) (RETURN (QUOTE ERR1))))
    (SETQ INTERM (CDR CONN EC))
    (GO STEP1)
    
    (STEP 4)
    (APPEND XSTOR (LIST X2))
    (COND ((NULL CONN EC) (RETURN XSTOR)))
    (GO STEP1)
    
    (STEP 5)
    (COND ((NOT (ELEMENT X ACLIST)) (RETURN (QUOTE ERR2))))
    (SETQ INTERM (COND (CADR (GET X (QUOTE FXY))))))
    
    (STEP 5A)
    (TREAD 0)
    (PRINT (QUOTE $$$REPLACE NODE NUMBER ? $))
    (SETQ N1 (TREAD T))
    (COND ((NUMBERP N1) (GO STEP5B))
    ((EQ N1 (QUOTE N)) (GO STEP5C))
    (PR INI (QUOTE $$$TYPE NUMBER IN ORIGINAL EXPRESSION TO BE REPLACED$))
    (PRINT (QUOTE $$$OR N IF NO REPLACEMENT$))
    (GO STEP5A)
    
    (STEP 5B)
    (TREAD 0)
    (PRINT (QUOTE $$$REPLACE BY ? $))
    (SETQ N2 (TREAD T))
    (COND ((NUMBERP N1) (GO STEP5C))
    (PRINT (QUOTE $$$TYPE NUMBER TO REPLACE ORIGINAL NUMBER$))
    (GO STEP5B)
    
    (STEP 5C)
    (APPEND PAIRS (LIST (CONS N1 N2)))
    (GO STEP5A)
    
    (STEP 6)
    (COND ((NULL PAIRS) (GO STEP8)))
    
    (STEP 6A)
(SETQ M1 (CAAR PAIRS))
(SETQ M2 (PLUS M1 1000))
(SETQ PAINTER (APPEND PAIRS1 (LIST (CONS M2 (CDAR PAIRS))))))
(SETQ PAIRS1 PAINTER)
(SETQ PAINTER (REPLACE M2 M1 INTERM))
(SETQ INTERM PAINTER)
(SETQ PAINTER (COR PAIRS))
(COND ((NULL PAINTER) (GO STEP7)))
(SETQ PAIRS PAINTER)
(GO STEP6A)

STEP7
(SETQ M1 (CAAR PAIRS1))
(SETQ M2 (CDAR PAIRS1))
(SETQ PAINTER (REPLACE M2 M1 INTERM))
(SETQ INTERM PAINTER)

STEP8
(SETQ PAINTER (COR PAIRS1))
(COND ((NULL PAINTER) (GO STEP8)))
(SETQ PAIRS1 PAINTER)
(GO STEP7)

STEP8
(RETURN INTERM) )))})