```
BEGIN
RECORD NODE(INTEGER T,C;
                   REFERENCE(NODE) G;
                   REFERENCE(NODE,ATOM) D);
RECORD ATOM(STRING (6) NAME;
                   REFERENCE(TYP) TYPE;
                   LOGICAL BVARE);
RECORD TTN(LOGICAL SUCCESS);
RECORD TYP (STRING TRES;
                   REFERENCE(LTYP) TARG);
RECORD TERM( INTEGER HDGN,ARGN;
           REFERENCE(NODE) HD,NINI; REFERENCE(TYP)TY);
RECORD LTYP (REFERENCE(TYP) FST;
               REFERENCE(LTYP) RST );
RECORD UNITREE(REFERENCE(LISTPT)DISAGREE;
         REFERENCE(LISTNODE)FROZEN;
         REFERENCE(UNITREE)PRED;
         REFERENCE(LISTSUC)SUCC);
COMMENT NOEUD DE L ARBRE D UNIFICATION;
RECORD LISTPT(REFERENCE(TERM)TLEFT,TRIGHT;
         REFERENCE(LISTPT)PTREST);
RECORD LISTNODE(REFERENCE(NODE)NODEINI; REFERENCE(LISTNODE)NODEREST);
COMMENT LIST OF PAIRS OF TERMS;
RECORD LISTSUC(REFERENCE(UNITREE)UNITRINI;
         REFERENCE(LISTSUC)SUCREST);
COMMENT LINKS NODES IN THE MATCHING TREE;
STRING (6) EXP,SNOM;
INTEGER I1,I2,I3, I4,I5,I6,IST,NEXTV,IFVAR;
REFERENCE (NODE) N1,N2,ND,NG,NDG,NGG;
REFERENCE (NODE) ARRAY PILE,STACK,NFVAR(1::20);
STRING(6) ARRAY FVAR(1::20);
LOGICAL VARIABL;
REFERENCE(LISTNODE)F;

COMMENT *********FIN DES DECLARATIONS**************;
COMMENT *******
*****THIS PROCEDURE CREATES A NEW VARIABLE *****
****;
STRING(6) PROCEDURE NEWVAR(STRING (1) VALUE V);
    BEGIN
    STRING(6)STR;
    STRING(12)STEMP;
    NEXTV:=NEXTV+1;
    STEMP:=INTBASE10(NEXTV);
    STR(0|1):=V;
    STR(1|5):=STEMP(7|5);
    STR
    END;
COMMENT ****** FIN DE NEWVAR ****;
PROCEDURE REDUCTION (REFERENCE (NODE) VALUE N1);
BEGIN
    ND:=D(N1);
COMMENT *** NOEUD ARGUMENT DU NOEUD LAMBDA ***;
    NG:=G(N1);
    NDG:=D(NG);
```

```
    COMMENT   *** VARIABLE LIEE ***;
    NGG:=G(NG);
    IF C(NDG)=1 THEN
        BEGIN
    IF C(ND)=1 THEN COLLECT(ND)
                    ELSE C(ND):=C(ND)-1;
        END
                     ELSE
        BEGIN
        T(NDG):=3;
        G(NDG):=ND;
        D(NDG):=NULL;
        C(NDG):=C(NDG)-1;
        END;
    T(N1):=3;
    G(N1):=NGG;
    D(N1):=NULL;
    IF C(NG)¬=1 THEN
        BEGIN
        C(NG):=C(NG)-1;
        C(NGG):=C(NGG)+1;
        END;
END;
COMMENT  ****** FIN REDUCTION  ******;
 PROCEDURE COLLECT(REFERENCE(NODE) VALUE N1);
BEGIN
    IF C(N1)>1 THEN C(N1):=C(N1)-1
                    ELSE
        IF T(N1)¬=2 THEN
                   BEGIN
                   COLLECT(G(N1));
                   IF T(N1)¬=3 THEN COLLECT(D(N1));
                   END;
END;
COMMENT *** FIN DE COLLECT *******;
PROCEDURE SUIVANTG( REFERENCE(NODE) VALUE N1);
BEGIN
    REFERENCE (NODE) N2;
    N2:=G(N1);
        WHILE T(N2)=3 DO
        BEGIN
        C(N2):=C(N2)-1;
        IF C(N2)>0 THEN C(G(N2)):=C(G(N2))+1;
        G(N1):=G(N2);
        N2:=G(N2);
        END;
END;
COMMENT    *****************;
PROCEDURE SUIVANTD( REFERENCE(NODE) VALUE N1);
    BEGIN
    REFERENCE(NODE) N2;
    N2:=D(N1);
        WHILE T(N2)=3 DO
        BEGIN
        C(N2):=C(N2)-1;
```

```
        IF C(N2)>0 THEN C(G(N2)):=C(G(N2))+1;
        D(N1):=G(N2);
        N2:=G(N2);
        END;
    END;
    COMMENT *** FIN DE SUIVANTD *****;
COMMENT ****************************;
COMMENT ****  COPIE  ***************;
REFERENCE(NODE) PROCEDURE COPIE (REFERENCE(NODE) VALUE N1);
        BEGIN
        REFERENCE(NODE) N2;
        IF T(N1)=2 THEN
                BEGIN
                N2:=G(N1);
                C(N2):=C(N2)+1;
                END
                     ELSE
                BEGIN
                N2:=DUPLIQUE(N1);
                IF T(N1)=1 THEN D(N2):=DUPLIQUE(D(N1))
                           ELSE
                           BEGIN
                           SUIVANTD(N1);
                           D(N2):=COPIE(D(N1));
                           END;
                SUIVANTG(N1);
                G(N2):=COPIE(G(N1));
                END;
            N2
        END;
COMMENT ***************************;
COMMENT *****FIN DE COPIE********;
COMMENT ****  DUPLIQUE  **********;
    REFERENCE(NODE) PROCEDURE DUPLIQUE (REFERENCE(NODE) VALUE N1);
    BEGIN
    REFERENCE (NODE) N2;
    N2:=NODE( T(N1), 1 ,NULL,NULL);
    IF T(N1)=2 THEN
        BEGIN
        G(N1):=N2;
        G(N2):=N2;
        D(N2):=D(N1);
        IST:=IST+1;
        STACK(IST):=N1;
        END;
    N2
    END;
COMMENT ***************************
**********FIN DE DUPLIQUE *********
**********************************;
REFERENCE(NODE)PROCEDURE COPY(REFERENCE(NODE)VALUE N1);
    BEGIN
    REFERENCE(NODE)N2;
    INTEGER I;
    IST:=0;
```

```
    N1:=COPIE(N1);
    FOR I:=1 UNTIL IST DO
          BEGIN
          N2:=STACK(I);
          STACK(I):=NULL;
          G(N2):=N2;
          END;
    N1
    END;
COMMENT ****************************
*********FIN DE COPY **************
**********************************;

PROCEDURE PNODE(REFERENCE(NODE) VALUE N1);
     BEGIN
     WRITE(N1,T(N1),C(N1),G(N1),D(N1));
     IF T(N1)¬=2 THEN
          BEGIN
          PNODE(G(N1));
          IF T(N1)¬=3 THEN PNODE(D(N1));
          END;
     END;
COMMENT **********FIN DE PNODE**********;
COMMENT **********************************************;
REFERENCE (NODE)  PROCEDURE CREATENODE;
    BEGIN
    REFERENCE(TYP)TY;
    REFERENCE(NODE) N1;
    INTEGER ISCAN;
    STRING(1) EXP;
    STRING(80) LINE;
    STRING(6) ARRAY  BVAR(1::20) ;
    REFERENCE(NODE) ARRAY  NBVAR(1::20);
COMMENT *** SUITE DES PROCEDURES  *****
        ***
        ***;
    PROCEDURE SCANN;
    BEGIN
    SC1:
    ISCAN:=ISCAN+1;
    IF ISCAN=79 THEN
         EXP:=";"
           ELSE
         BEGIN
         EXP:=LINE(ISCAN|1);
         IF EXP=" " THEN GOTO SC1;
         END;
    END;
COMMENT ****************************
*********FIN DE SCANN ************
**********************************;
    STRING(6) PROCEDURE EXPRESSION;
    BEGIN
    INTEGER I;
    STRING(6) VAR;
```

```
    SCANN;
    I:=0;
    VAR:=" ";
    VAR(I|1):=LINE(ISCAN|1);
    SCANN;
    WHILE DECODE(EXP(0|1))>  239 DO
          BEGIN
          I:=I+1;
          VAR(I|1):=LINE(ISCAN|1);
          SCANN;
          END;
COMMENT LINE(ISCANN|1) EST UN CHIFFRE           **;
    ISCAN:=ISCAN-1;
    VAR
    END;
COMMENT ********************************
***************FIN DE EXPRESSION********
**************************************;
REFERENCE(NODE) PROCEDURE SCANNEXP(INTEGER VALUE IBVAR);
    BEGIN
    REFERENCE(NODE) N1,N2,N3,N4;
    REFERENCE(ATOM) A1;
    INTEGER IB,IL;
COMMENT
* PROCEDURE CREANT LA SUITE DES NOEUDS LAMBDA ET
* DE LA VARIABLEQUI LUI EST LIE
* LA VALEUR RETOURNEE EST CELLE DU SOMMET LE PLUS
* BAS DANS L ARBRE
*;
    REFERENCE(NODE) PROCEDURE HEADING;
    BEGIN
    REFERENCE(NODE) N2,N3,N4;
    REFERENCE(ATOM) A2;
    STRING(6) VNAME;
    PROCEDURE AFFECT;
          BEGIN
               IL:=0;
          WHILE IL<IFVAR DO
               BEGIN
               IL:=IL+1;
               IF VNAME=FVAR(IL) THEN
                    BEGIN
                    TY:=TYPE(D(NFVAR(IL)));
                    GOTO FOUND;
                    END;
               END;
          IFVAR:=IFVAR+1;
          FVAR(IFVAR):=VNAME;
          TY:=CREATETYP(VNAME);
          A2:=ATOM(VNAME,TY,TRUE);
          N3:=NODE(2,0,NULL,A2);
          G(N3):=N3;
          NFVAR(IFVAR):=N3;
FOUND:    A2:=ATOM(VNAME,TY,TRUE);
          N3:=NODE(2,1,NULL,A2);
```

```
        G(N3):=N3;
        IBVAR:=IBVAR+1;
        BVAR(IBVAR):=VNAME;
        NBVAR(IBVAR):=N3;
        N4:=NODE(1,1,NULL,N3);
        VNAME:="      ";
        END;
COMMENT **** FIN DE AFFECT  ***;
COMMENT * CREATION D UN SOMMET LAMBDA ET DE SON ARGUMENT
**;
    VNAME:=EXPRESSION;
    IF DECODE(VNAME(0|1))<129 THEN ERREUR(1);
    AFFECT;
    N1:=N4;
HEA:    VNAME:=EXPRESSION;
    N2:=N4;
    IF VNAME(0|1)¬="." THEN
        BEGIN
        IF DECODE(VNAME(0|1))<128 THEN ERREUR(2);
        AFFECT;
        IF G(N1)=NULL THEN G(N1):=N4 ELSE G(N2):=N4;
        GOTO HEA;
        END;
    N2
    END;
COMMENT **** FIN DE HEADING ***
**
**;
    STRING(6) VAR;
    PROCEDURE BVARIABLE;
    BEGIN
    IB:=IBVAR+1;
RESTAB: IB:=IB-1;
IF IB<1 THEN
        BEGIN
        COMMENT * VARIABLE LIBRE ;
        IL:=0;
RESTAL  :         IL:=IL+1;
        IF IL>IFVAR THEN
            BEGIN
            COMMENT * NOUVELLE VARIABLE LIBRE *;
            IFVAR:=IL;
            TY:=CREATETYP(VAR);
            A1:=ATOM(VAR,TY,VARIABL);
            N3:=NODE(2,1,NULL,A1);
            G(N3):=N3;
            NFVAR(IL):=N3;
            FVAR(IL):=VAR;
            END
                 ELSE
            IF VAR=FVAR(IL) THEN
                BEGIN
                N3:=NFVAR(IL);
                C(N3):=C(N3)+1;
                END
```

```
                                        ELSE GOTO RESTAL;
          END
         ELSE
                COMMENT * VARIABLE LIEE *;
                IF BVAR(IB)=VAR THEN
                     BEGIN
                     N3:=NBVAR(IB);
                     C(N3):=C(N3)+1;
                     END
                                 ELSE GOTO RESTAB;
     END;
COMMENT *******************************
*********FIN DE BVARIABLE**************
************* *************************;
COMMENT CORPS DE SCANNEXP ************;
     N1:=NULL;
     N2:=NULL;
     SCANN;
          IF EXP="*" THEN
               BEGIN
               IF LINE(ISCAN+1|1)=" " THEN ERREUR(3);
               N2:=HEADING;
               SCANN;
               END;
COMMENT
* ON A CREE HEADING
*;
ATOME:    IF DECODE(EXP)>   128 THEN
COMMENT   ON A UN ATOM     **;
          BEGIN
          ISCAN:=ISCAN-1;
          VAR:=EXPRESSION;
          BVARIABLE;
          END
                                        ELSE
          IF EXP="(" THEN
               BEGIN
               N3:=SCANNEXP(IBVAR);
               SCANN;
               END   ELSE ERREUR(4);
          SCANN;
          IF (EXP=")") OR(EXP=",")OR(EXP=";") THEN
COMMENT ON A LU UN ATOM ON CHERCHE SI ON A (*X)A OU
        (*X)A(E1,E2,...,EN)                                *;
               ISCAN:=ISCAN-1
                                                  ELSE
COMMENT ATOM EST SUIVI D'ARGUMENT  ;
               IF EXP="(" THEN
                    BEGIN
BOUC:               N4:=SCANNEXP(IBVAR);
                      SCANN;
                      IF (EXP=",") OR (EXP=")") THEN
                           BEGIN
                           N3:=NODE(0,1,N3,N4);
                           IF EXP="," THEN GOTO BOUC ;
```

```
                     END ELSE ERREUR(5);
                END ELSE ERREUR(6);
   IF N1=NULL THEN N1:=N3
                  ELSE G(N2):=N3;
   N1
   END;
COMMENT  * FIN DE SCANNEXP **
**
**;
ISCAN:=-1;
WRITE("ENTER YOUR TERM:");
WRITE(" ");
READCARD(LINE);
N1:=SCANNEXP(0);
C(N1):=0;
N1
END;
COMMENT ***FIN DE CREATENODE ***;
COMMENT    *****************     ;
REFERENCE(TYP)PROCEDURE CREATETYP(STRING(6)VALUE VAR);
    BEGIN
    INTEGER I1,ISCAN;
    REFERENCE(TYP)TY1;
    STRING(80)LINE;
    STRING(1)STTYP;
    COMMENT    ***********
               ***********
               ***********;
    PROCEDURE SCANN;
        BEGIN
    SC1: ISCAN:=ISCAN+1;
        IF ISCAN>79 THEN
        STTYP:="." ELSE
            BEGIN
            IF LINE(ISCAN|1)=" " THEN GOTO SC1;
            STTYP:=LINE(ISCAN|1);
            END;
        END;
    COMMENT    ***********
               ***********
               *FIN DE SCANN;
    REFERENCE(TYP)PROCEDURE READTYPE;
               BEGIN
               REFERENCE(TYP)T1;
               REFERENCE(LTYP)LT1,LT2;
               SCANN;
               IF STTYP="(" THEN
                   BEGIN
                   LT1:=LT2:=LTYP;
                   CR1: FST(LT2):=READTYPE;
                        IF STTYP="," THEN
                            BEGIN
                            LT2:=RST(LT2):=LTYP;
                            GOTO CR1;
                            END;
```

```
                  IF STTYP="-" THEN
                      BEGIN
                      SCANN;
                      IF (STTYP ¬="I")AND(STTYP ¬="O") THEN
                                      ERREUR(7);
                      T1:=TYP(STTYP,LT1);
                      RST(LT2):=NULL;
                      SCANN;
                      IF STTYP¬=")" THEN ERREUR(8);
                      SCANN;
                      END
                              ELSE ERREUR(9);
                  COMMENT CAS (A1,A2,...,AP-  ;
              END
                      ELSE
          IF((STTYP="I")OR(STTYP="O"))THEN
              BEGIN
              T1:=TYP(STTYP,NULL);
              SCANN;
              END  ELSE ERREUR(10);
          T1
          END;
          COMMENT **********
                  ***********
                  **FIN DE READTYPE;
    COMMENT CORPS DE CREATETYP;
    I1:=DECODE(VAR(0|1));
    VARIABL:=(I1>227)OR(I1=198)OR(I1=199)OR(I1=200);
    WRITE("ENTER TYPE OF ATOM :",VAR);
    WRITE(" ");
    READCARD(LINE);
    ISCAN:=-1;
    TY1:=READTYPE;
    TY1
    END;
COMMENT ***********************
    ***********************
    ***FIN DE CREATETYP;
PROCEDURE ERREUR(INTEGER VALUE I);
BEGIN
WRITE(" ERREUR NO: ",I);
GOTO ENDPROG;
END;
COMMENT **********;
COMMENT * PROCEDURE POUR ECRIRE LA LAMBDA EXPRESSION ***;
PROCEDURE WRITEEXP(REFERENCE(NODE) VALUE N1);
BEGIN
INTEGER ISCAN,IEXP;
STRING(200) LINE;
STRING(6) EXP;
STRING(1) CHAR;
COMMENT
**
**;
PROCEDURE SCANVECR;
```

```
    BEGIN
    IEXP:=0;
    IF ISCAN>194 THEN ERREUR(11);
    IEXP:=0;
    WHILE EXP(IEXP|1) ¬= " " DO
          BEGIN
          LINE(ISCAN+IEXP|1):=EXP(IEXP|1);
          IEXP:=IEXP+1;
          IF IEXP > 5 THEN GOTO ENDSCAN;
          END;
ENDSCAN:  ISCAN:=ISCAN+IEXP;
    END;
COMMENT ******************************
************FIN DE SCANNECR ***********
*************************************;
PROCEDURE ECREXP(REFERENCE(NODE) VALUE N1);
BEGIN
INTEGER IPILE;
REFERENCE(NODE) ARRAY PILE(1::10);
IPILE:=0;
ECR2:IF T(N1)=2 THEN
    BEGIN
    EXP:=NAME(D(N1));
    SCANNECR;
    END
        ELSE
COMMENT ON A SOIT UN NOEUD LAMBDA SOIT UN NOEUD GAMMA **;
    BEGIN
    IF T(N1)=1 THEN
COMMENT * CAS NOEUD LAMBDA *
*;
          BEGIN
          EXP:="*";
          SCANNECR;
ECR1:     EXP:=NAME(D(D(N1)));
          EXP:=NAME(D(D(N1))):=NEWVAR(EXP(0|1));
          SCANNECR;
          SUIVANTG(N1);
          N1:=G(N1);
          IF T(N1)=1 THEN GOTO ECR1;
          EXP:=".";
          SCANNECR;
          IF T(N1)=2 THEN GOTO ECR2;
          END;
ECR0:     IF T(N1)=0 THEN
          BEGIN
          WHILE T(N1)=0 DO
                BEGIN
                IPILE:=IPILE+1;
                PILE(IPILE):=N1;
                SUIVANTG(N1);
                N1:=G(N1);
                END;
COMMENT * A GAUCHE ON DOIT AVOIR UN ATOM SINON ERREUR **;
         IF T(N1)¬=2 THEN ERREUR(12);
```

```
          EXP:=NAME(D(N1));
          SCANNECR;
          EXP:="(";
          SCANNECR;
          WHILE IPILE¬=0 DO
               BEGIN
               N1:=PILE(IPILE);
               SUIVANTD(N1);
               ECREXP(D(N1));
               IPILE:=IPILE-1;
               IF IPILE¬=0 THEN
                    BEGIN
                    EXP:=",";
                    SCANNECR;
                    END;
               END;
         EXP:=")";
         SCANNECR;
         END;
COMMENT * FIN DU CAS  ... A(E1,...EN)  **;
     END;
END;
COMMENT ** FIN DE ECREXP  ***;
COMMENT**
**
**;
LINE:=" ";
ISCAN:=0;
CONVERSION(N1);
WHILE T(N1)=3 DO N1:=G(N1);
WRITE("         ");
ECREXP(N1);
IEXP:=0;
WHILE IEXP<=ISCAN DO
     BEGIN
     CHAR:=LINE(IEXP|1);
      IF CHAR="0" THEN
          BEGIN
          WHILE CHAR="0" DO
               BEGIN
               IEXP:=IEXP+1;
               CHAR:=LINE(IEXP|1);
               END;
          WHILE 240<=DECODE(CHAR) DO
               BEGIN
               WRITEON(CHAR);
               IEXP:=IEXP+1;
               CHAR:=LINE(IEXP|1);
               END;
          END     ELSE
          BEGIN
          WRITEON(CHAR);
          IEXP:=IEXP+1;
          END;
     END;
```

```
END;
COMMENT FIN DE WRITEEXP *****;
COMMENT ****************** ***;
PROCEDURE CONVERSION(REFERENCE(NODE) VALUE N1);
BEGIN
INTEGER I1,I2;
REFERENCE(NODE) N2;
I1:=0;
ETI0: IF T(N1)¬=2 THEN  SUIVANTG(N1);
ETI1: IF T(N1)=0 THEN
         BEGIN
         WHILE T(G(N1))=0 DO
                   BEGIN
                   I1:=I1+1;
                   PILE(I1):=N1;
                   N1:=G(N1);
                   SUIVANTG(N1);
                   END;
         IF T(G(N1))=1 THEN
                   BEGIN
                   ETI2: IF C(G(N1))¬=1 THEN
                        BEGIN
                        IF C(D(G(N1)))>1 THEN
                             BEGIN
                             C(G(N1)):=C(G(N1))-1;
                             G(N1):=COPY(G(N1));
                             END;
                        END;
                    REDUCTION(N1);
                   WHILE T(N1)=3 DO N1:=G(N1);
                   IF (T(N1)=1) AND (I1¬=0) THEN
                        BEGIN
                        N1:=PILE(I1);
                PILE(I1):=NULL;
                        SUIVANTG(N1);
                        I1:=I1-1;
                        GOTO ETI2;
                        END
                               ELSE
                   GOTO ETI0;
         END
                  ELSE
COMMENT  **** LE SOMMET G(N1) N'EST PAS UN LAMBDA *****;
         BEGIN
              SUIVANTD(N1);
         N1:=D(N1);
         GOTO ETI0;
         END;
         END
COMMENT ***LE NOEUD EST SOIT UN LAMBDA SOIT UN ATOM ***;
         ELSE
         IF T(N1)=1 THEN
                   BEGIN
                   N1:=G(N1);
                   GOTO ETI0;
```

```
                    END;
          IF I1¬=0 THEN
                    BEGIN
                    N1:=PILE(I1);
                    PILE(I1):=NULL;
                    I1:=I1-1;
                    SUIVANTD(N1);
                    N1:=D(N1);
                    GOTO ETIO;
                    END;
END;
COMMENT ***FIN DE CONVERSION *****;
REFERENCE(TERM) PROCEDURE MAKETERM(REFERENCE(NODE) VALUE N1);
    BEGIN
    INTEGER I1,I2;
    REFERENCE(NODE) N2;
    REFERENCE(LTYP) LT1,LT2,LT3;
    LT2:=LTYP;
    LT1:=RST(LT2);
    I1:=0;
    I2:=0;
    WHILE(T(N1)=3) DO N1:=G(N1);
    N2:=N1;
LAT:IF T(N2)=2 THEN GOTO OUT
                     ELSE
    IF T(N2)=0 THEN
         BEGIN
      I2:=I2+1;
    SUIVANTG(N2);
      N2:=G(N2);
      GOTO LAT;
      END ELSE
      BEGIN
    LT2:=RST(LT2):=LTYP(TYPE(D(D(N2))),NULL);
      I1:=I1+1;
    SUIVANTG(N2);
      N2:=G(N2);
      GOTO LAT;
      END;
OUT:LT3:=TARG(TYPE(D(N2)));
    FOR I3:=1 UNTIL I2 DO
    LT3:=RST(LT3);
    RST(LT2):=LT3;
   TERM(I1,I2,N2,N1,TYP(TRES(TYPE(D(N2))),LT1))
    END;
COMMENT *** FIN DE MAKETERM *****;
COMMENT ***************************;
REFERENCE(NODE)PROCEDURE SUBSTI(REFERENCE(NODE)VALUE NX,NEX,N1);
    BEGIN
    REFERENCE(NODE)N2,N3;
    INTEGER I1;
    G(NX):=COPY(NEX);
    C(NEX):=0;
    N2:=COPY(N1);
    G(NX):=NX;
```

```
    CONVERSION(N2);
    N2
    END;
COMMENT *************
******** FIN DE SUBSTI ********
******************************;
    COMMENT *****************
            *****************
            *****************;
LOGICAL PROCEDURE FLEXIBLE(REFERENCE(TERM) VALUE T1);
    BEGIN
    REFERENCE(LISTNODE)F1;
    REFERENCE(NODE)N,N1;
    INTEGER I1;
    LOGICAL FLX;
    FLX:=FALSE;
    N1:=HD(T1);
    IF ¬BVARE(D(N1)) THEN
         GOTO OUTFAL;
    F1:=F;
    WHILE F1¬=NULL DO
         BEGIN
         IF N1=NODEINI(F1) THEN GOTO OUTFAL;
         F1:=NODEREST(F1);
         END;
    N:=NINI(T1);
    FOR I1:=1 UNTIL HDGN(T1) DO
         IF N1=D(N)  THEN GOTO OUTFAL
                     ELSE N:=G(N);
    FLX:=TRUE;
    OUTFAL:
    FLX
    END;
    COMMENT *********
              *******
          FIN DE FLEXIBLE;
REFERENCE(UNITREE,TTN) PROCEDURE SIMPLIFY
                (REFERENCE(UNITREE,TTN) VALUE U1);
    BEGIN
    REFERENCE(TERM) T1,T2;
    REFERENCE(NODE)N1,N2,N3;
    REFERENCE(LISTPT) DS,DSF;
    INTEGER HDG,I1;
    LOGICAL RIG,TSN,TFN,LF1,LF2;
    DS:=DISAGREE(U1);
    DSF:=NULL;
    F:=FROZEN(U1);
    RIG:=TFN:=TSN:=FALSE;
NEXTPAIR: IF DS=NULL THEN GOTO OUTSIMP1;
    T1:=TLEFT(DS);
    T2:=TRIGHT(DS);
    LF1:=FLEXIBLE(T1);
    LF2:=FLEXIBLE(T2);
    DS:=PTREST(DS);
COMMENT ON TEST SI (T1,T2) EST RIGID,RIGID OU N'IMPORTE QUOI ;
```

```
    IF LF1 THEN
        BEGIN
        IF ¬LF2 THEN RIG:=TRUE;
        DSF:=LISTPT(T1,T2,DSF);
        GOTO NEXTPAIR;
        END   ELSE
        IF LF2 THEN
            BEGIN
            RIG:=TRUE;
            DSF:=LISTPT(T2,T1,DSF);
            GOTO NEXTPAIR;
            END;
    HDG:=HDGN(T1);
    IF HDG¬=HDGN(T2) THEN
        BEGIN
        TFN:=TRUE;
        GOTO OUTSIMP2;
        END;
    N1:=NINI(T1);
    N2:=NINI(T2);
    FOR I1:=1 UNTIL HDG DO
        BEGIN
        G(D(N1)):=D(N2);
        COMMENT ON RENOME LES VARIABLES LIEES DE T1 EN CELLES DE T2;
        F:=LISTNODE(D(N2),F);
        N1:=G(N1);
        N2:=G(N2);
        END;
    N3:=HD(T1);
    IF N3¬=HD(T2) THEN
        BEGIN
        IF G(N3)¬=HD(T2) THEN
            BEGIN
            TFN:=TRUE;
            GOTO OUTSIMP2;
            END;
        END;
    COMMENT SAME HEADING ;
    N1:=COPY(N1);
    N3:=NINI(T1);
    FOR I1:=1 STEP 1 UNTIL HDG DO
        G(D(N3)):=D(N3);
    I1:=ARGN(T1);
    WHILE I1>0 DO
        BEGIN
        DS:=LISTPT(MAKETERM(D(N1)),MAKETERM(D(N2)),DS);
        N1:=G(N1);
        N2:=G(N2);
        I1:=I1-1;
        END;
    GOTO NEXTPAIR;
OUTSIMP1: IF ¬RIG THEN TSN:=TRUE
         ELSE
        BEGIN
        DISAGREE(J1):=DSF;
```

```
         FROZEN(U1):=F;
         END;
OUTSIMP2:
    IF (TFN OR TSN) THEN U1:=TTN(TSN);
    U1
    END;
COMMENT ********************
    FIN DE SIMPLIFY *****;
COMMENT ***************
    MATCH
*************************;
REFERENCE(LISTNODE) PROCEDURE MATCH(REFERENCE(TERM) VALUE TE1,TE2);
BEGIN
COMMENT MATCH RETURNS A LIST OF NODES TO BE SUBSTITUDED FOR
      HEAD OF TE1 ;
INTEGER IN1,IN2,IP1,IP2,IN,IK,I1,IM,I2,I3,IQ;
REFERENCE(LISTNODE)F1;
REFERENCE(NODE)N1,N2,N3,N4,NRES;
REFERENCE(TYP)T1;
REFERENCE(LTYP)LT1,LT2,LT3;
REFERENCE(LISTNODE)RESLT;
REFERENCE(NODE) PROCEDURE MGNODE(REFERENCE(NODE) VALUE N1;
              INTEGER VALUE I;REFERENCE(TYP)VALUE T1);
    BEGIN
    COMMENT THIS PROCEDURE CONSTUCTS THE MOST GENERAL TERM
            CONTAINING I BOUND VARIABLES WHOSE BINDER STARTS IN N1.
            T1 CONTAINS THE TYPE OF THE
            TERM TO BE CONSTRUCTED;
    REFERENCE(LTYP)LT2,LT3;
    REFERENCE(NODE)N2;
    REFERENCE(ATOM)AT1;
    LT2:=LT3:=NULL;
    AT1:=ATOM(NEWVAR("H"),TYP(TRES(T1),TARG(T1)),TRUE);
         COMMENT THIS IS THE NEW VARIABLE H;
    N2:=NODE(2,1,NULL,AT1);
    G(N2):=N2;
    WHILE 0<I DO
         BEGIN
         N2:=NODE(0,1,N2,D(N1));
         IF LT2=NULL THEN LT2:=LT3:=LTYP(TYPE(D(D(N1))),NULL)
                   ELSE LT3:=RST(LT3):=LTYP(TYPE(D(D(N1))),NULL);
         N1:=G(N1);
         I:=I-1;
         END;
    IF LT2¬=NULL THEN
         BEGIN
         RST(LT3):=TARG(T1);
         TARG(TYPE(AT1)):=LT2;
         END;
    N2
    END;
COMMENT ***********************************
************* FIN DE MGNODE *************
****************************************;
LOGICAL PROCEDURE EQTYP(REFERENCE(TYP)VALUE TYP1,TYP2);
```

```
    BEGIN
    COMMENT THIS PROCEDURE TESTS WHETHER  TWO TYPES ARE EQUAL;
    LOGICAL BRES;
    REFERENCE(LTYP)LT1,LT2;
    BRES:=FALSE;
    IF TRES(TYP1)¬=TRES(TYP2)THEN GOTO XEQ;
    LT1:=TARG(TYP1);
    LT2:=TARG(TYP2);
    WHILE LT1¬=NULL DO
          BEGIN
          IF LT2 =NULL THEN GOTO XEQ;
          IF ¬EQTYP(FST(LT1),FST(LT2)) THEN GOTO XEQ;
          LT1:=RST(LT1);
          LT2:=RST(LT2);
          END;
    IF LT2¬=NULL THEN GOTO XEQ;
    BRES:=TRUE;
    XEQ:BRES
    END;
          COMMENT END OF EQTYP ********
                    ***********************;
          COMMENT ******  BODY OF MATCH *******;
RESLT:=NULL;
IN1:=HDGN(TE1);
IN2:=HDGN(TE2);
IP1:=ARGN(TE1);
IP2:=ARGN(TE2);
IF IN2<IN1  THEN GOTO EXITMATCH;
IN:=IN2-IN1;
COMMENT *********
         IMITATION
         *********
;
N2:=NINI(TE2);
FOR I1:=1 UNTIL IN1 DO
     COMMENT ON TEST SI LA TETE DE E2 EST DANS
              LES N1 PREMIERES VARIABLES LIEES DE E2;
     IF HD(TE2)=D(N2) THEN GOTO PROJECTION
                      ELSE N2:=G(N2);
F1:=F;
WHILE F1¬=NULL DO
     COMMENT IS HEAD OF TE2 A FROZEN VARIABLE;
     IF HD(TE2)=NODEINI(F1) THEN GOTO PROJECTION
                            ELSE F1:=NODEREST(F1);
IF 0<IN THEN
     BEGIN
     COMMENT FIRST CASE OF IMITATION
              TERM TO BE SUBSTITUDED TO F:
              *W1...WP1V(N1+1)...VN2.HD(TE2)(E1,...,EP2) ;
     N3:=NULL;
     I1:=IP1;
     LT1:=TARG(TYPE(D(HD(TE1))));
     WHILE 0<I1 DO
          BEGIN
          COMMENT CREATION OF THE HEADING OF THE TERM TO BE SUBSTITUTED;
```

```
        N4:=NODE(2,IP2+1,NULL,
                      ATOM("W",FST(LT1),TRUE));
        G(N4):=N4;
        IF I1=IP1 THEN
            NRES:=N3:=NODE(1,1,NULL,N4)
                ELSE
            N3:=G(N3):=NODE(1,1,NULL,N4);
        LT1:=RST(LT1);
        I1:=I1-1;
        END;
    I1:=IN;
    WHILE 0<I1 DO
        BEGIN
        C(D(N2)):=C(D(N2))+1;
        IF N3=NULL THEN NRES:=N3:=NODE(1,1,NULL,D(N2))
                 ELSE N3:=G(N3):=NODE(1,1,NULL,D(N2));
        N2:=G(N2);
        I1:=I1-1;
        END;
    N4:=HD(TE2);
    I1:=IP2;
    LT1:=TARG(TYPE(D(N4)));
    WHILE 0<I1 DO
        BEGIN
        N4:=NODE(0,1,N4,MGNODE(NRES,IP1+IN,FST(LT1)));
        LT1:=RST(LT1);
        I1:=I1-1;
        END;
    IF N3=NULL THEN NRES:=N4
                ELSE G(N3):=N4;
    RESLT:=LISTNODE(NRES,NULL);
    END
       ELSE
    BEGIN
    COMMENT  CASE IN=0 ;
    IF IP2<IP1 THEN I1:=IP1-IP2
                ELSE I1:=0;
    FOR IK:=I1 UNTIL IP1 DO
        BEGIN
        COMMENT CHECK CONDITION (2) ;
        LT1:=TARG(TYPE(D(HD(TE1))));
        LT2:=TARG(TYPE(D(HD(TE2))));
        FOR I1:=1 UNTIL IK DO LT1:=RST(LT1);
        FOR I1:=1 UNTIL IP2-IP1+IK DO LT2:=RST(LT2);
        FOR I1:=1 UNTIL IP1-IK DO
            IF ¬EQTYP(FST(LT1),FST(LT2)) THEN GOTO FAILIMI;
        I1:=IK;
        NRES:=N3:=NULL;
        LT1:=TARG(TYPE(D(HD(TE1))));
        WHILE 0<I1 DO
                BEGIN
            N4:=NODE(2,IP2-IP1+IK+1,NULL,
                          ATOM("W",FST(LT1),TRUE));
                G(N4):=N4;
            IF I1 =IK THEN
```

```
                         NRES:=N3:=NODE(1,1,NULL,N4)
                         ELSE
                         N3:=G(N3):=NODE(1,1,NULL,N4);
               LT1:=RST(LT1);
               I1:=I1-1;
               END;
          N4:=HD(TE2);
          I1:=IP2-IP1+IK;
          LT1:=TARG(TYPE(D(N4)));
          WHILE 0<I1 DO
               BEGIN
               N4:=NODE(0,1,N4,MGNODE(NRES,IK,FST(LT1)));
               LT1:=RST(LT1);
               I1:=I1-1;
               END;
          IF N3=NULL THEN
               NRES:=N4
                     ELSE
                    G(N3):=N4;
          RESLT:=LISTNODE(NRES,RESLT);
FAILIMI:  END;
          COMMENT FIN DE LA BOUCLE SUR IK ;
      END;
PROJECTION:
COMMENT  **********
              PROJECTION
         **********;
IQ:=0;
LT2:=TARG(TYPE(D(HD(TE1))));
WHILE LT2¬=NULL DO
     BEGIN
     IQ:=IQ+1;
     LT2:=RST(LT2);
     END;
COMMENT CASE A:
          WE SUBSTITUTE A TERM WITH HEADING:
          *W1 ...WK.WI  ,1<=I<=K<=P1  ;
FOR IK:=1 UNTIL IP1 DO
     BEGIN
     LT1:=TARG(TYPE(D(HD(TE1))));
     FOR I2:=1 UNTIL IK DO LT1:=RST(LT1);
     COMMENT LT1 CONTAINS THE K+1 ... Q1 ARGUMENTS TYPE OF F ;
     I2:=IQ-IK;
     FOR I1:=1 UNTIL IK DO
          BEGIN
          COMMENT FIRST WE CHECK CONDITION (2) ;
          LT2:=TARG(TYPE(D(HD(TE1))));
          FOR I3:=1 UNTIL I1-1 DO LT2:=RST(LT2);
          IF TRES(FST(LT2))¬=TRES(TYPE(D(HD(TE1)))) THEN
               GOTO FAILCASEA;
          COMMENT WE CHECK THAT TERMINAL TYPES ARE COMPATIBLE;
          LT2:=TARG(FST(LT2));
          COMMENT LT2 CONTAINS THE LIST OF THE TYPE OF THE ITH ARGUMENT
                  OF F;
          LT3:=LT2;
```

```
        FOR I3:=1 UNTIL I2 DO
             IF LT3=NULL THEN GOTO FAILCASEA
                        ELSE LT3:=RST(LT3);
        IM:=0;
        WHILE LT3¬=NULL DO
             BEGIN
             IM:=IM+1;
             LT3:=RST(LT3);
             END;
        FOR I3:=1 UNTIL IM DO
             LT2:=RST(LT2);
             COMMENT WE TAKE OFF THE M GAMMAS;
        FOR I3:=1 UNTIL I2 DO
             IF ¬EQTYP(FST(LT1),FST(LT2)) THEN GOTO FAILCASEA
                                              ELSE
                  BEGIN
                  LT1:=RST(LT1);
                  LT2:=RST(LT2);
                  END;
             COMMENT WE BUILD THE HEADING;
        LT1:=TARG(TYPE(D(HD(TE1))));
        NRES:=N3:=NULL;
        FOR I3:=1 UNTIL IK DO
             BEGIN
             N4:=NODE(2,IM+1,NULL,ATOM("W",FST(LT1),TRUE));
             G(N4):=N4;
             IF I3=1 THEN NRES:=N3:=NODE(1,1,NULL,N4)
                     ELSE N3:=G(N3):=NODE(1,1,NULL,N4);
             LT1:=RST(LT1);
             END;
        N4:=NRES;
        FOR I3:=1 UNTIL I1-1 DO N4:=G(N4);
        N4:=D(N4);
        C(N4):=C(N4)+1;
        COMMENT N4 IS THE HEAD WI;
        LT1:=TARG(TYPE(D(N4)));
        FOR I3:=1 UNTIL IM DO
             BEGIN
             N4:=NODE(0,1,N4,MGNODE(NRES,IK,FST(LT1)));
             LT1:=RST(LT1);
             END;
        G(N3):=N4;
        RESLT:=LISTNODE(NRES,RESLT);
FAILCASEA:
        END;
        COMMENT END OF LOOP ON I1;
    END;
        COMMENT END OF LOOP ON IK;
    COMMENT CASEB: WE SUBSTITUTE A TERM WITH HEADING *W1 ... WP1
        V(N1+1) ... V(N1+K).WI;
    FOR I1:=1 UNTIL IP1 DO
        BEGIN
        LT2:=TARG(TYPE(D(HD(TE1))));
        FOR I3:=1 UNTIL I1-1 DO LT2:=RST(LT2);
        IF TRES(FST(LT2)) ¬= TRES(TYPE(D(HD(TE1)))) THEN
```

```
                GOTO FAILCASEB;
   COMMENT WE CHECK THAT TERMINAL TYPES ARE CONSISTENT;
   LT2:=TARG(FST(LT2));
   COMMENT LT2 CONTAINS THE LIST OF THE TYPES OF THE ARGUMENTS
           OF THE I-TH ARGUMENT OF F;
   FOR IK:=IN STEP -1 UNTIL 1 DO
       BEGIN
       LT1:=TARG(TYPE(D(HD(TE1))));
       FOR I2:=1 UNTIL IK+IP1 DO LT1:=RST(LT1);
       COMMENT LT1 CONTAINS THE P1+K+1 ... Q1-TH ARGUMENT
                                TYPES OF F;
       I2:=IQ-IK-IP1;
       COMMENT WE CHECK CONDITION 2);
       LT3:=LT2;
       FOR I3:=1 UNTIL I2 DO
           IF LT3=NULL THEN GOTO FAILCASEB
                     ELSE LT3:=RST(LT3);
       IM:=0;
       WHILE LT3 ¬= NULL DO
           BEGIN
           IM:=IM+1;
           LT3:=RST(LT3);
           END;
       FOR I3:=1 UNTIL IM DO
           LT2:=RST(LT2);
       COMMENT WE TAKE OFF THE M GAMMAS;
       FOR I3:=1 UNTIL I2 DO
           IF ¬EQTYP(FST(LT1),FST(LT2)) THEN GO TO FAILCASEB
                                     ELSE
               BEGIN
               LT1:=RST(LT1);
               LT2:=RST(LT2);
                   END;
       COMMENT WE BUILD THE HEADING;
       LT1:=TARG(TYPE(D(HD(TE1))));
       NRES:=N3:=NULL;
       FOR I3:=1 UNTIL IK+IP1 DO
           BEGIN
           N4:=NODE(2,IM+1,NULL,ATOM("W",FST(LT1),TRUE));
           G(N4):=N4;
           IF I3=1 THEN NRES:=N3:=NODE(1,1,NULL,N4)
                 ELSE N3:=G(N3):=NODE(1,1,NULL,N4);
           LT1:=RST(LT1);
           END;
       N4:=NRES;
       FOR I3:=1 UNTIL I1-1 DO N4:=G(N4);
       N4:=D(N4);
       C(N4):=C(N4)+1;
       COMMENT N4 IS THE HEAD WI;
       LT1:=TARG(TYPE(D(N4)));
       FOR I3:=1 UNTIL IM DO
           BEGIN
           N4:=NODE(0,1,N4,MGNODE(NRES,IK+IP1,FST(LT1)));
           LT1:=RST(LT1);
           END;
```

```
            G(N3):=N4;
            RESLT:=LISTNODE(NRES,RESLT);
            END;
            COMMENT END OF LOOP ON IK;
FAILCASEB:
        END;
        COMMENT END OF LOOP ON I1;
    EXITMATCH:RESLT
    END;
COMMENT  ******************************
         *******END OF MATCH***********
         ******************************;
COMMENT
      *******CONSTRUCTION DES SUCCESSEURS D'UN NOEUD DE L'ARBRE
                D'UNIFICATION                                  **
      *****************************************************;
REFERENCE(LISTSUC) PROCEDURE SUCCTREE (REFERENCE(UNITREE,TTN) VALUE U1);
    BEGIN
    COMMENT THIS PROCEDURE RETURNS THE LIST OF THE SUCCESSORS NODES
            OF U1;
    REFERENCE(UNITREE)U2;
    REFERENCE(LISTPT) LPT1,LPT2;
    REFERENCE(NODE)N1,N2,N3;
    REFERENCE(TERM) T1,T2;
    REFERENCE(LISTSUC) LS1;
    REFERENCE(LISTNODE) LN1;
    LPT1:=DISAGREE(U1);
    LS1:=NULL;
    WHILE FLEXIBLE(TRIGHT(LPT1)) DO
        LPT1:=PTREST(LPT1);
    LN1:=MATCH(TLEFT(LPT1),TRIGHT(LPT1));
    IF LN1=NULL THEN GOTO EXITSUCCTREE;
    N1:=HD(TLEFT(LPT1));
    WHILE LN1¬=NULL DO
        BEGIN
        N2:=NODEINI(LN1);
        WRITE("TO ",NAME(D(N1)), " MATCH SUBSTITUTES");
        WRITEEXP(N2);
        LN1:=NODEREST(LN1);
        LPT2:=DISAGREE(U1);
        F:=FROZEN(U1);
        LPT1:=NULL;
        WHILE LPT2¬=NULL DO
            BEGIN
            T1:=TLEFT(LPT2);
            T2:=TRIGHT(LPT2);
            LPT2:=PTREST(LPT2);
            N3:=NINI(T1);
            N3:=SUBSTI(N1,N2,N3);
            T1:=MAKETERM(N3);
            N3:=NINI(T2);
            N3:=SUBSTI(N1,N2,N3);
            T2:=MAKETERM(N3);
            LPT1:=LISTPT(T1,T2,LPT1);
            END;
```

```
      U2:=UNITREE(LPT1,F,U1,NULL);
      U2:=SIMPLIFY(U2);
      LS1:=LISTSUC(U2,LS1);
      END;
   EXITSUCCTREE:
      SUCC(U1):=LS1;
   LS1
   END;
   COMMENT
        FIN DE SUCCTREE
   ************************;

COMMENT *********
   MAIN ****;
PROCEDURE MAIN;
   BEGIN
   STRING(80) MOT;
   LOGICAL LL;
   REFERENCE(NODE)N1,N2;
   REFERENCE(UNITREE,TTN)U1,U2;
   REFERENCE(LISTSUC) LS1,LS2,LS3;
   REFERENCE(LISTPT)LIPT;
   REFERENCE(TERM)T1,T2;
   INTEGER IPAIR,INODEMAX;
   IFVAR:=0;
   FOR IPAIR:=1 UNTIL 10 DO WRITE(" ");
   IPAIR:=0;
   LIPT:=NULL;
NXTPAIR:IPAIR:=IPAIR+1;
   WRITE(" PAIR NO",IPAIR);
   WRITE("FIRST TERM");
   N1:=CREATENODE;
   CONVERSION(N1);
   T1:=MAKETERM(N1);
   WRITE("SECOND TERM");
   N2:=CREATENODE;
   CONVERSION(N2);
   T2:=MAKETERM(N2);
   LIPT:=LISTPT(T1,T2,LIPT);
   WRITE("ANY MORE PAIRS?  (WRITE Y OR N)");
   WRITE(" ");
   READCARD(MOT);
   IF MOT="Y" THEN GOTO NXTPAIR;
   F:=NULL;
   U1:=UNITREE(LIPT,F,NULL,NULL);
   U1:=SIMPLIFY(U1);
   WRITE("HOW MANY NODES ALLOWED ?");
   WRITE(" ");
   READ(INODEMAX);
   LS1:=LISTSUC(U1,NULL);
   LS2:=NULL;
   U1:=NULL;
   GOTO WRNDE;
NXTNODE: IF INODEMAX=0 THEN GOTO EXXIT;
   INODEMAX:=INODEMAX-1;
```

```
    IF LS2=NULL THEN GOTO EXXIT;
    U1:=UNITRINI(LS2);
    LS2:=SUCREST(LS2);
    IF U1 IS TTN THEN GOTO NXTNODE;
    LS1:=SUCCTREE(U1);
WRNDE:IF LS1=NULL THEN WRITE(U1," HAS NO SUCCESSOR ... TFN")
                 ELSE IF LS2=NULL THEN LS2:=LS1
                                 ELSE
       BEGIN
       LS3:=LS2;
       WHILE SUCREST(LS3)¬=NULL DO LS3:=SUCREST(LS3);
       SUCREST(LS3):=LS1;
       END;
    WHILE LS1¬=NULL DO
       BEGIN
       U2:=UNITRINI(LS1);
       WRITE(U2," FOLLOWS ",U1);
       LS1:=SUCREST(LS1);
       IF U2 IS TTN THEN
          BEGIN
          IF SUCCESS(U2) THEN WRITE(U2," IS TSN ")
                         ELSE WRITE(U2," IS TFN ");
          GOTO NEXTSUC;
          END;
       LIPT:=DISAGREE(U2);
          WHILE LIPT¬=NULL DO
          BEGIN
          T1:=TLEFT(LIPT);
          T2:=TRIGHT(LIPT);
          LIPT:=PTREST(LIPT);
          WRITE ("THE NEXT PAIR TO UNIFY IS");
          WRITEEXP(NINI(T1));
          WRITEEXP(NINI(T2));
          END;
       NEXTSUC:
       END;
    WRITE("TIME SPENT IN JIFFIES=",TIME(1));
    GOTO NXTNODE;
    END;
COMMENT ********************
    ********************
    FIN DE MAIN;
EXXIT:WRITE("EXIT");
WRITE("TIME SPENT IN JIFFIES=",TIME(1));
NEXTV:=0;
MAIN;
ENDPROG:
END.
```