

```

BEGIN                                              L4400010
RECORD NODE(INTEGER T,C;
            REFERENCE(NODE) G;
            REFERENCE(NODE,ATCM) D);
RECORD ATCM(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,SNDM;
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 VARIABLE;
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;
  STEMPC:=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 *** FIN DE SUIVANTG ***;
PROCEDURE SUIVANTD( REFERENCE(NODE) VALUE N1);
BEGIN
  REFERENCE(NODE) N2;
  N2:=D(N1);
  WHILE T(N2)=3 DO
    BEGIN
      C(N2):=C(N2)-1;
    END;

```

```
IF C(N2)>0 THEN C(G(N2)):=C(G(N2))+1;
D(N1):=G(N2);
N2:=G(N2);
END;
```

L4401110
L4401120
L4401130
L4401140
L4401150

```
END;
COMMENT *** FIN DE SUIVANTD ***;
```

L4401160
L4401170

```
COMMENT *** COPIE ***;
```

L4401180

```
REFERENCE(NODE) PROCEDURE COPIE (REFERENCE(NODE) VALUE N1);
```

```
BEGIN
```

L4401190
L4401200

```
REFERENCE(NODE) N2;
```

L4401210
L4401220

```
IF T(N1)=2 THEN
```

L4401230
L4401240

```
    BEGIN
```

L4401250
L4401260

```
        N2:=G(N1);
        C(N2):=C(N2)+1;
```

L4401270
L4401280

```
    END
```

L4401290
L4401300

```
    ELSE
```

L4401310
L4401320

```
        BEGIN
```

L4401330
L4401340

```
            SUIVANTD(N1);
```

L4401350
L4401360

```
            D(N2):=DUPLIQUE(D(N1));
            ELSE
```

L4401370
L4401380

```
                BEGIN
```

L4401390
L4401400

```
                    SUIVANTG(N1);
```

L4401410
L4401420

```
                    G(N2):=COPIE(G(N1));
                    END;
```

L4401430
L4401440

```
                END;
```

L4401450
L4401460

```
V2
```

L4401470
L4401480

```
END;
```

L4401490
L4401500

```
COMMENT *** FIN DE COPIE ***;
```

L4401510
L4401520

```
COMMENT *** DUPLIQUE ***;
```

L4401530
L4401540

```
REFERENCE(NODE) PROCEDURE DUPLIQUE (REFERENCE(NODE) VALUE N1);
```

```
BEGIN
```

L4401550
L4401560

```
REFERENCE (NODE) N2;
```

L4401570
L4401580

```
N2:=NODE( T(N1), 1 ,NULL,NULL);
```

L4401590
L4401600

```
IF T(N1)=2 THEN
```

L4401610
L4401620

```
    BEGIN
```

L4401630
L4401640

```
        G(N1):=N2;
```

L4401650
L4401660

```
        G(N2):=N2;
```

L4401670
L4401680

```
        D(N2):=D(N1);
```

L4401690
L4401700

```
        IST:=IST+1;
```

L4401710
L4401720

```
        STACK(IST):=N1;
```

L4401730
L4401740

```
    END;
```

L4401750
L4401760

```
N2
```

L4401770
L4401780

```
END;
```

L4401790
L4401800

```
COMMENT *** FIN DE DUPLIQUE ***;
```

L4401810
L4401820

```
COMMENT *** COPY ***;
```

L4401830
L4401840

```
REFERENCE(NODE) PROCEDURE COPY(REFERENCE(NODE)VALUE N1);
```

```
BEGIN
```

L4401850
L4401860

```
REFERENCE(NODE) N2;
```

L4401870
L4401880

```
INTEGER I;
```

L4401890
L4401900

```
IST:=0;
```

L4401910
L4401920

```

N1:=COPY(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;
COMMENT *** FIN DE PNODE***;
COMMENT *** SUITE DES PROCEDURES ***;
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 NEVAR(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;

```

L4401560
L4401570
L4401580
L4401590
L4401700
L4401710
L4401720
L4401730
L4401740
L4401750
L4401760
L4401770
L4401780
L4401790
L4401800
L4401810
L4401820
L4401830
L4401840
L4401850
L4401860
L4401870
L4401880
L4401890
L4401900
L4401910
L4401920
L4401930
L4401940
L4401950
L4401960
L4401970
L4401980
L4401990
L4402000
L4402010
L4402020
L4402030
L4402040
L4402050
L4402060
L4402070
L4402080
L4402090
L4402100
L4402110
L4402120
L4402130
L4402140
L4402150
L4402160
L4402170
L4402180
L4402190
L4402200

```

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{ISCAN|1} EST UN CHIFFRE ****;
ISCAN:=ISCAN-1;
VAR
END;
COMMENT *****FIN DE EXPRESSION*****;
*****;
REFERENCE(NODE) PROCEDURE SCANNEXP(INTEGER VALUE IBSVAR);
BEGIN
  REFERENCE(NODE) N1,N2,N3,N4;
  REFERENCE(ATOM) A1;
  INTEGER IB,IL;
COMMENT
* PROCEDURE CREAT LA SUITE DES NOEUDS LAMBDA ET
* DE LA VARIABLE QUI 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:=CREATETYPE(VNAME);
      A2:=ATOM(VNAME,TY,TRUE);
      N3:=NODE{2,0,NULL,A2};
      G(N3):=N3;
      NFVAR{IFVAR}:=N3;
      A2:=ATOM(VNAME,TY,TRUE);
      N3:=NODE{2,1,NULL,A2};
    END;
  END;
END;

```

```

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 BVARIALE;
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,VARIABLE);
        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
      END;
  END;

```

```

        ELSE GOTO RESTAL;

END;
ELSE
COMMENT * VARIABLE LIFE *;
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
*;
ATOM: 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 LI 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
N4:=SCANNEXP(IBVAR);
SCANN;
IF (EXP=",") OR (EXP=")") THEN
BEGIN
N3:=NODE(0,1,N3,N4);
IF EXP="," THEN GOTO BOUC ;

```

LA403310
LA403320
LA403330
LA403340
LA403350
LA403360
LA403370
LA403380
LA403390
LA403400
LA403410
LA403420
LA403430
LA403440
LA403450
LA403460
LA403470
LA403480
LA403490
LA403500
LA403510
LA403520
LA403530
LA403540
LA403550
LA403560
LA403570
LA403580
LA403590
LA403600
LA403610
LA403620
LA403630
LA403640
LA403650
LA403660
LA403670
LA403680
LA403690
LA403700
LA403710
LA403720
LA403730
LA403740
LA403750
LA403760
LA403770
LA403780
LA403790
LA403800
LA403810
LA403820
LA403830
LA403840
LA403850

FIGURE ALGORITHM 12

CAMBRIDGE MONITOR SYSTEM

```

        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(L1):=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
SCI: ISCAN:=ISCAN+1;
IF ISCAN>79 THEN
STTYP:="." ELSE
BEGIN
IF LINE(ISCAN|1)=" " THEN GOTO SCI;
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;
CRI: FST(LT2):=READTYPE;
IF STTYP="," THEN
BEGIN
LT2:=RST(LT2):=LTYP;
GOTO CRI;
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(011));
VARIABLE:=(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);
  GET 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 SCANNER;

```

LA40441
LA40442
LA40443
LA40444
LA40445
LA40446
LA40447
LA40448
LA40449
LA40450
LA40451
LA40452
LA40453
LA40454
LA40455
LA40456
LA40457
LA40458
LA40459
LA40460
LA40461
LA40462
LA40463
LA40464
LA40465
LA40466
LA40467
LA40468
LA40469
LA40470
LA40471
LA40472
LA40473
LA40474
LA40475
LA40476
LA40477
LA40478
LA40479
LA40480
LA40481
LA40482
LA40483
LA40484
LA40485
LA40486
LA40487
LA40488
LA40489
LA40490
LA40491
LA40492
LA40493
LA40494
LA40495

```

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 SCANNFCR ***
PROCEDURE ECREXP(REFERENCE(NODE) VALUE N1);
BEGIN
INTEGER IPILE;
REFERENCE(NODE) ARRAY PILE(1::10);
PILE:=0;
ECR2:IF T(N1)=2 THEN
  BEGIN
    EXP:=NAME(D(N1));
    SCANNFCR;
  END
  ELSE
COMMENT ON A SJIT UN NOEUD LAMBDA SOIT UN NOEUD GAMMA **;
  BEGIN
    IF T(N1)=1 THEN
      COMMENT * CAS NOEUD LAMBDA *
    *;
    BEGIN
      EXP:="**";
      SCANNFCR;
    ECR1: EXP:=NAME(D(D(N1)));
      EXP:=NAME(D(D(N1))):=NEWVAR(EXP(0|1));
      SCANNFCR;
      SUIVANTG(N1);
      N1:=G(N1);
      IF T(N1)=1 THEN GOTO ECR1;
      EXP:=". ";
      SCANNFCR;
      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);
  END;

```

```

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;
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
            WRITEDN(CHAR);
            IEXP:=IEXP+1;
            CHAR:=LINE(IEXP+1);
          END;
      END
      ELSE
      BEGIN
        WRITEDN(CHAR);
        IEXP:=IEXP+1;
      END;
    END;
  END;
END;

```

LAM0551
LAM0552
LAM0553
LAM0554
LAM0555
LAM0556
LAM0557
LAM0558
LAM0559
LAM0560
LAM0561
LAM0562
LAM0563
LAM0564
LAM0565
LAM0566
LAM0567
LAM0568
LAM0569
LAM0570
LAM0571
LAM0572
LAM0573
LAM0574
LAM0575
LAM0576
LAM0577
LAM0578
LAM0579
LAM0580
LAM0581
LAM0582
LAM0583
LAM0584
LAM0585
LAM0586
LAM0587
LAM0588
LAM0589
LAM0590
LAM0591
LAM0592
LAM0593
LAM0594
LAM0595
LAM0596
LAM0597
LAM0598
LAM0599
LAM0600
LAM0601
LAM0602
LAM0603
LAM0604
LAM0605

```

END;
COMMENT FIN DE WRITEEXP ****;
COMMENT ***** ****;
PROCEDURE CONVERSION(REFERENCE(NODE) VALUE N1);
BEGIN
INTEGER I1,I2;
REFERENCE(NODE) N2;
I1:=0;
ETIO: 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 ETIO;
          END
        END
      ELSE
        BEGIN
          SUIVANTD(N1);
          N1:=D(N1);
          GOTO ETIO;
        END;
      END;
COMMENT *** LE SOMMET G(N1) N'EST PAS UN LAMBDA ***;
BEGIN
  SUIVANTD(N1);
  N1:=D(N1);
  GOTO ETIO;
END;
END;
COMMENT ***LE NOEUD EST SOIT UN LAMBDA SOIT UN ATOM ***;
ELSE
  IF T(N1)=1 THEN
    BEGIN
      N1:=G(N1);
      GOTO ETIO;
    END;
  END;

```

LAM06060
LAM06070
LAM06080
LAM06090
LAM06100
LAM06110
LAM06120
LAM06130
LAM06140
LAM06150
LAM06160
LAM06170
LAM06180
LAM06190
LAM06200
LAM06210
LAM06220
LAM06230
LAM06240
LAM06250
LAM06260
LAM06270
LAM06280
LAM06290
LAM06300
LAM06310
LAM06320
LAM06330
LAM06340
LAM06350
LAM06360
LAM06370
LAM06380
LAM06390
LAM06400
LAM06410
LAM06420
LAM06430
LAM06440
LAM06450
LAM06460
LAM06470
LAM06480
LAM06490
LAM06500
LAM06510
LAM06520
LAM06530
LAM06540
LAM06550
LAM06560
LAM06570
LAM06580
LAM06590
LAM06600

FILE: UNIFY ALGOLW P1

CAMBRIDGE MONITOR SYSTEM

```
        END;
IF I1=0 THEN
  BEGIN
    N1:=PILE(I1);
    PILE(I1):=NULL;
    I1:=I1-1;
    SUIVANTD(N1);
    V1:=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(V1)=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;
        SJIVANTG(N2);
        N2:=G(N2);
        G3TD LAT;
      END ELSE
        BEGIN
          LT2:=RST(LT2):=LTYP(TYPE(D(D(N2))),NULL);
          I1:=I1+1;
          SJIVANTG(N2);
          N2:=G(N2);
          G3TD LAT;
        END;
DJT: LT3:=TARG(TYPE(D(N2)));
FOR I3:=1 JNTIL 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=NNULL 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(LIST?T) 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 OUTSIMPI;
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                                LAM07710
  BEGIN                                     LAM07720
    IF ~LF2 THEN RIG:=TRUE;
    DSF:=LISTPT(T1,T2,DSF);
    GOTO NEXTPAIR;
  END ELSE
  IF LF2 THEN                                LAM07750
    BEGIN                                     LAM07760
      RIG:=TRUE;
      DSF:=LISTPT(T2,T1,DSF);
      GOTO NEXTPAIR;
    END;
HDG:=HDGV(T1);                            LAM07830
IF HDG~=HDGN(T2) THEN                    LAM07840
  BEGIN                                     LAM07850
    TN:=TRUE;
    GOTO OUTSIMP2;
  END;
N1:=NINI(T1);                            LAM07890
N2:=NINI(T2);                            LAM07900
FOR I1:=1 UNTIL HDG DO                  LAM07910
  BEGIN                                     LAM07920
    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);                            LAM07990
  IF N3~=HD(T2) THEN                    LAM08000
    BEGIN                                     LAM08010
      IF G(N3)~=HD(T2) THEN
        BEGIN                                     LAM08020
          TFN:=TRUE;
          GOTO OUTSIMP2;
        END;
    END;
  COMMENT SAME HEADING ;
  N1:=COPY(N1);
  N3:=NINI(T1);
  FOR I1:=1 STEP 1 UNTIL HDG DO      LAM08110
    G(D(N3)):=D(N3);
    I1:=ARGN(T1);
    WHILE I1>0 DO                      LAM08120
      BEGIN                                     LAM08130
        DS:=LISTPT(MAKETERM(D(N1)),MAKETERM(D(N2)),DS);
        N1:=G(N1);
        N2:=G(N2);
        I1:=I1-1;
      END;
    GOTO NEXTPAIR;
  DJTSIMP1: IF ~RIG THEN TSN:=TRUE     LAM08220
    ELSE
    BEGIN                                     LAM08230
      DISAGREE(J1):=DSF;
    END;

```

FILE: JNIFY ALGOL W P1

CAMBRIDGE MONITOR SYSTEM

L408260
L408270
L408280
L408290
L408300
L408310
L408320
L408330
L408340
L408350
L408360
L408370
L408380
L408390
L408400
L408410
L408420
L408430
L408440
L408450
L408460
L408470
L408480
L408490
L408500
L408510
L408520
L408530
L408540
L408550
L408560
L408570
L408580
L408590
L408600
L408610
L408620
L408630
L408640
L408650
L408660
L408670
L408680
L408690
L408700
L408710
L408720
L408730
L408740
L408750
L408760
L408770
L408780
L408790
L408800

```
FROZEN(U1):=F;  
END;  
JJTSIMP2:  
IF (TFV 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 SUBSTITUTED 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 CONSTRUCTS 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(NEVAR("H"),TYP(TRES(T1),TARG(T1)),TRUE);  
    COMMENT THIS IS THE NEW VARIABLE H;  
N2:=NODE(2,1,NULL,AT1);  
G(N2):=N2;  
WHILE IK 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 *** * * * * * * * * * * * * * * * * *;  
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *;  
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *;  
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);
I>1:=ARGN(TE1);
I>2:=ARGN(TE2);
IF IN2<IN1 THEN GOTO EXITMATCH;
IN:=IN2-IN1;
COMMENT ****
IMITATION
*****
;

N2:=NINI(TE2);
FOR II:=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 I<IN THEN
BEGIN
COMMENT FIRST CASE OF IMITATION
TERM TO BE SUBSTITUTED TO F:
*W1...WP1V(N1+1)...VN2.HD(TE2)(E1,...,EP2) ;
N3:=NULL;
II:=IP1;
LT1:=TARG(TYPE(D(HD(TE1))));;
WHILE I<II DO
BEGIN
COMMENT CREATION OF THE HEADING OF THE TERM TO BE SUBSTITUTED; LAM0935

```

```

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 NOTTYPE(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

```

FILE: UNIFY ALGOLW P1

CAMBRIDGE MONITOR SYSTEM

```
NRES:=N3:=NODE(1,1,NULL,N4)
ELSE
  N3:=G(N3):=NODE(1,1,NULL,N4);
LT1:=RST(LT1);
I1:=I1-1;
END;
V4:=HD(TE2);
I1:=IP2-IP1+IK;
LT1:=TARG(TYPE(D(N4)));
WHILE 0<I1 DO
  BEGIN
    V4:=V3DE(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);
FAIL-IMI: 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#W1 ,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 II:=1 UNTIL IK DO
      BEGIN
        COMMENT FIRST WE CHECK CONDITION (2) ;
        LT2:=TARG(TYPE(D(HD(TE1)))); 
        FOR I3:=1 UNTIL II-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);
FAI_CASEA:
  END;
  COMMENT END OF LOOP ON I1;
END;
COMMENT END OF LOOP ON IK;
COMMENT CASEB: WE SUBSTITUTE A TERM WITH HEADING *WI ... 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

```

LAM10460
LAM10470
LAM10480
LAM10490
LAM10500
LAM10510
LAM10520
LAM10530
LAM10540
LAM10550
LAM10560
LAM10570
LAM10580
LAM10590
LAM10600
LAM10610
LAM10620
LAM10630
LAM10640
LAM10650
LAM10660
LAM10670
LAM10680
LAM10690
LAM10700
LAM10710
LAM10720
LAM10730
LAM10740
LAM10750
LAM10760
LAM10770
LAM10780
LAM10790
LAM10800
LAM10810
LAM10820
LAM10830
LAM10840
LAM10850
LAM10860
LAM10870
LAM10880
LAM10890
LAM10900
LAM10910
LAM10920
LAM10930
LAM10940
LAM10950
LAM10960
LAM10970
LAM10980
LAM10990
LAM11000

```

        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
J= THE I-TH ARGUMENT OF F;
FOR IK:=IN STEP -1 UNTIL 1 DO
BEGIN
LT1:=TARG(TYPE(D(HD(TE1)))); LAM11010
FOR I2:=1 UNTIL IK+IP1 DO LT1:=RST(LT1); LAM11020
COMMENT LT1 CONTAINS THE P1+K+1 ... Q1-TH ARGUMENT
TYPES OF F; LAM11030
I2:=IQ-IK-IP1; LAM11040
COMMENT WE CHECK CONDITION 2; LAM11050
LT3:=LT2; LAM11060
FOR I3:=1 UNTIL I2 DO LAM11070
    IF LT3=NULL THEN GOTO FAILCASEB LAM11080
        ELSE LT3:=RST(LT3); LAM11090
IM:=0; LAM11100
WHILE LT3 ≠ NULL DO LAM11110
BEGIN LAM11120
    IM:=IM+1; LAM11130
    LT3:=RST(LT3); LAM11140
END; LAM11150
FOR I3:=1 UNTIL IM DO LAM11160
    LT2:=RST(LT2); LAM11170
COMMENT WE TAKE OFF THE M GAMMAS; LAM11180
FOR I3:=1 UNTIL I2 DO LAM11190
    IF ¬EQTYP(FST(LT1),FST(LT2)) THEN GO TO FAILCASEB LAM11200
        ELSE LAM11210
        BEGIN LAM11220
            LT1:=RST(LT1); LAM11230
            LT2:=RST(LT2); LAM11240
        END; LAM11250
COMMENT WE BUILD THE HEADING; LAM11260
LT1:=TARG(TYPE(D(HD(TE1)))); LAM11270
NRES:=N3:=NULL; LAM11280
FOR I3:=1 UNTIL IK+IP1 DO LAM11290
BEGIN LAM11300
    N4:=NODE(2,IM+1,NULL,ATOM("W",FST(LT1),TRUE)); LAM11310
    G(N4):=N4; LAM11320
    IF I3=1 THEN NRES:=N3:=NODE(1,1,NULL,N4) LAM11330
        ELSE N3:=G(N3):=NODE(1,1,NULL,N4); LAM11340
    LT1:=RST(LT1); LAM11350
END; LAM11360
N4:=NRES; LAM11370
FOR I3:=1 UNTIL I1-1 DO N4:=G(N4); LAM11380
N4:=D(N4); LAM11390
G(N4):=G(N4)+1; LAM11400
COMMENT N4 IS THE HEAD WI; LAM11410
LT1:=TARG(TYPE(D(N4))); LAM11420
FOR I3:=1 UNTIL IM DO LAM11430
BEGIN LAM11440
    N4:=NODE(0,1,N4,MGNODE(NRES,IK+IP1,FST(LT1))); LAM11450
    LT1:=RST(LT1); LAM11460
END; LAM11470
LAM11480
LAM11490
LAM11500
LAM11510
LAM11520
LAM11530
LAM11540
LAM11550

```

FILE: UNIFY ALGOLW P1

CAMBRIDGE MONITOR SYSTEM

```
G(N3):=N4;  
RESLT:=LISTNODE(NRES,RESLT);  
END;  
COMMENT END OF LOOP ON IK;
```

LAMI1560
LAMI1570
LAMI1580
LAMI1590

FAILCASEB:

```
END;  
COMMENT END OF LOOP ON II;
```

EXITMATCH:=RESLT

END;

```
COMMENT **** END OF MATCH ****;  
*****;
```

LAMI1600
LAMI1610
LAMI1620
LAMI1630
LAMI1640
LAMI1650
LAMI1660
LAMI1670

COMMENT

```
*****CONSTRUCTION DES SUCCESEURS D'UN NOEUD DE L'ARBRE  
D'UNIFICATION
```

LAMI1680
LAMI1690
LAMI1700

*****;

LAMI1710

REFERENCE(LISTSUC) PROCEDURE SUCCTREE (REFERENCE(UNITREE,TTN) VALUE U1);

LAMI1720

BEGIN

COMMENT THIS PROCEDURE RETURNS THE LIST OF THE SUCCESSORS NODES
OF U1;

LAMI1730
LAMI1740

REFERENCE(UNITREE)U2;

LAMI1750

REFERENCE(LISTPT) LPT1,LPT2;

LAMI1760

REFERENCE(NODE)N1,N2,N3;

LAMI1770

REFERENCE(TERM) T1,T2;

LAMI1780

REFERENCE(LISTSUC) LS1;

LAMI1790

REFERENCE(LISTNODE) LN1;

LAMI1800

LPT1:=DISAGREE(U1);

LAMI1810

LN1:=NULL;

LAMI1820

WHILE FLEXIBLE(TRIGHT(LPT1)) DO

LAMI1830

 LPT1:=PTREST(LPT1);

LAMI1840

 LVI:=MATCH(TLEFT(LPT1),TRIGHT(LPT1));

LAMI1850

 IF LN1=NULL THEN GOTO EXITSUCCTREE;

LAMI1860

 N1:=HD(TLEFT(LPT1));

LAMI1870

 WHILE LN1=NULL DO

LAMI1880

 BEGIN

LAMI1890

 N2:=NODEINI(LN1);

LAMI1900

 WRITE("TO ",NAME(D(N1)), " MATCH SUBSTITUTES");

LAMI1910

 WRITEEXP(N2);

LAMI1920

 LN1:=NODEREST(LN1);

LAMI1930

 LPT2:=DISAGREE(U1);

LAMI1940

 F:=FROZEN(U1);

LAMI1950

 LPT1:=NULL;

LAMI1960

 WHILE LPT2=NULL DO

LAMI1970

 BEGIN

LAMI1980

 T1:=TLEFT(LPT2);

LAMI1990

 T2:=TRIGHT(LPT2);

LAMI2000

 LPT2:=PTREST(LPT2);

LAMI2010

 N3:=NINI(T1);

LAMI2020

 N3:=SUBSTI(N1,N2,N3);

LAMI2030

 T1:=MAKETERM(N3);

LAMI2040

 N3:=NINI(T2);

LAMI2050

 N3:=SUBSTI(N1,N2,N3);

LAMI2060

 T2:=MAKETERM(N3);

LAMI2070

 LPT1:=LISTPT(T1,T2,LPT1);

LAMI2080

 END;

LAMI2090

LAMI2100

```

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)TTNU1,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;
NEXTPAIR: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 NEXTPAIR;
  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;
NEXTNODE: IF INODEMAX=0 THEN GOTO EXXIT;
  INODEMAX:=INODEMAX-1;

```

FILE: UNIFY ALGOLW PI

CAMBRIDGE MONITOR SYSTEM

```
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 ... TPN")
      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 TPN ");
      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;
END>ROG:
END.
```

LAM1266
LAM1267
LAM1258
LAM1269
LAM1270
LAM1271
LAM1272
LAM1273
LAM1274
LAM1275
LAM1276
LAM1277
LAM1278
LAM1279
LAM1280
LAM1281
LAM1282
LAM1283
LAM1284
LAM1285
LAM1286
LAM1287
LAM1288
LAM1289
LAM1290
LAM1291
LAM1292
LAM1293
LAM1294
LAM1295
LAM1296
LAM1297
LAM1298
LAM1299
LAM1300
LAM1301
LAM1302
LAM1303
LAM1304
LAM1305
LAM1306
LAM1307
LAM1308
LAM1309
LAM1310
LAM1311
LAM1312
LAM1313