SETL NEWSLETTER NO. 182A (SUPERCEDES NL 182)
LEONARD I. VANEK
APRIL 28, 1977

LINKAGE CONVENTIONS FOR THE SETL OPTIMIZER

IN ORDER TO FACILITATE BOTH THE LINKING OF OCCURRENCES OF VARIABLES BETWEEN SUBROUTINES AND THE RESTORATION OF STACKED VALUES, IT IS USEFUL TO EMPLOY TWO SETS OF RESERVED TEMPORARY NAMES IN CONJUNCTION WITH DUMMY ASSIGNMENTS TO AND FROM THESE TEMPORARIES. ONE SUCH SET OF TEMPORARIES (P1, P2, ...) IS USED TO TRANSMIT PARAMETER VALUES FROM PROCEDURE CALLS TO THE PROCEDURE(S) BEING CALLED. THE OTHER SET (S1, S2, ...) IS USED FOR LINKING THE VALUES OF STACKED VARIABLES FROM THE POINT WHERE THEY ARE STACKED TO THE POINT WHERE THEY ARE RETRIEVED.

ASSIGNMENTS TO THESE TEMPORARIES WILL ACCOMPANY IMPLICIT OR EXPLICIT PUSHES OF THE VALUE BEING ASSIGNED, WHILE ASSIGNMENTS FROM THEM CORRESPOND TO POPS. THE RELEVANT OPCODES, ALONG WITH THEIR REAL AND VIRTUAL SEMANTICS, ARE GIVEN BELOW.

```
OP¬PLIST(L1,L2,L3,L4) MARKS START OF CALLING SEQUENCE
OP¬ARGIN(TMP,VAR,F)   PUSH VALUE OF VAR AND SET TMP = VAR
                      F = IF VAR RETURNS A VALUE THEN 1 ELSE 0
OP¬ARGOUT(VAR,TMP,-)  SET VAR = TMP AND POP VALUE INTO VAR
OP¬PARMIN(VAR,TMP,F)  DUMMY ASSIGNMENT FOR FORMAL PARAMETERS
                      F = IF VAR RETURNS A VALUE THEN 1 ELSE 0
OP¬PARMOUT(TMP,VAR,-) DUMMY ASSIGNMENT FOR RETURN PARAMETERS
OP¬FREE(N,-,-)        POP N STACK FRAMES W/O ASSIGNING CONTENTS
AUX¬ASN(X,Y,-)        DUMMY ASSIGNMENT OF Y TO X
OP¬CALL(M,N,SUBR)     CALL SUBROUTINE NAMED BY SUBR, WHICH
                      HAS N PARAMETERS, M OF WHICH CAN BE
                      USED TO RETURN VALUES
OP¬OF(VAR,X,FUNC)     CALL ONE-PARAMETER FUNCTION NAMED BY FUNC
OP¬OFN(VAR,N,FUNC)    CALL N-PARAMETER FUNCTION NAMED BY FUNC
```

IN THE ABOVE OPERATIONS, THE PUSH AND POP ARE ACTIONS WHICH ARE ACTUALLY TO BE EXECUTED BUT ARE IGNORED BY THE OPTIMIZER. THE ASSIGNMENTS TO AND FROM TMP, ON THE OTHER HAND, NEVER REALLY TAKE PLACE, BUT THE OPTIMIZER TREATS THEM AS IF THEY DO OCCUR. BECAUSE THE SPECIAL TEMPORARIES ARE USED ONLY IN DUMMY ASSIGNMENTS, THEY DO NOT HAVE TO EXIST AT EXECUTION TIME, BUT THEY MUST BE TREATED AS REAL BY THE OPTIMIZER.

WE FIRST ADDRESS OURSELVES TO DESCRIBING THE LINKAGE CODE REQUIRED FOR SUBROUTINES. THIS DIFFERS FROM THAT USED FOR FUNCTION CALLS BECAUSE ONLY SUBROUTINES CAN MODIFY PARAMETERS, WHILE FUNCTIONS MUST RETURN A VALUE.

TO ILLUSTRATE PARAMETER LINKAGE, CONSIDER THE FOLLOWING EXAMPLE:

```
      .
      .
   SUBR(A,B,C=);    $    C IS MODIFIED BY SUBR
      .
      .
   DEFINE SUBR(X,Y,Z=);
      DECLARE L: STACK;    $    M IS DECLARED TO BE GLOBAL STACKED
         .
         .
      RETURN;
         .
         .
   END SUBR;
      .
      .
```

THE CALL WOULD BE COMPILED AS FOLLOWS:

```
    OP¬PLIST(1,2,3,4)   MARK START OF CALLING SEQUENCE
 1: OP¬ASN(TA,A,-)          CONVERT-IN ASSIGNMENTS
    OP¬ASN(TB,B,-)          /
    OP¬ASN(TC,C,-)         /
 2: OP¬ARGIN(P1,TA,0)       PUSH PARAMETERS
    OP¬ARGIN(P2,TB,0)       /
    OP¬ARGIN(P3,TC,1)      /
 3: OP¬CALL(1,3,SUBR)      CALL SUBROUTINE
    OP¬FREE(-,-,-)         RELEASE STACK FRAME FOR READ/ONLY ARGUMENT
    OP¬FREE(-,-,-)         RELEASE STACK FRAME FOR READ/ONLY ARGUMENT
    OP¬ARGOUT(TD,P3,-)     POP RETURNED PARAMETER VALUE
    OP¬ASN(C,TD,-)         CONVERT-OUT ASSIGNMENT
 4:      ...
```

AND THE SUBROUTINE HAS THE FOLLOWING ENTRY AND EXIT CODE:

```
   OP¬SUBRSTRT(SUBR,-,-)  MARK SUBROUTINE FOR RELOCATION
   OP¬PARMIN(X,P1,0)        LINK ARGUMENT AND PARAMETER VALUES
   OP¬PARMIN(Y,P2,0)        /
   OP¬PARMIN(Z,P3,1)       /
   AUX¬ASN(S1,X,-)            SAVE VALUES OF STACKED VARIABLES
   AUX¬ASN(S2,Y,-)           /
   AUX¬ASN(S3,Z,-)           /
   AUX¬ASN(S4,L,-)           /
   AUX¬ASN(S5,M,-)          /
   OP¬ASN(L,OM,-)            RE-INITIALIZE STACKED VARIABLES
   OP¬ASN(M,OM,-)           /
      .
      .
   OP¬GO(-,-,EXIT)
      .
      .
```

```
      .
      .
EXIT:AUX¬ASN(M,S5,-)          POP OLD VALUE OF STACKED VARIABLES
   AUX¬ASN(L,S4,-)           /
   AUX¬ASN(Z,S3,-)          /
   AUX¬ASN(X,S2,-)         /
   AUX¬ASN(Y,S1,-)        /
   OP¬PARMOUT(P3,Z,-)     SAVE RETURN PARAMETER VALUE
   OP¬SUBREND(SUBR,-,-)   RETURN
```

THE STACKED-VARIABLE TEMPORARIES (S1, S2, ...) ARE USED ONLY LOCALLY WITHIN PROCEDURES TO UNDO THE ASSIGNMENTS EXECUTED ON ENTRY WHICH REDEFINE THE STACKED VARIABLES AND PARAMETERS. THE CORRESPONDING S-TEMPORARIES AT EITHER END OF THE PROCEDURE ARE LINKED VIA FFROM, AS ARE ORDINARY TEMPORARIES, EVEN THOUGH THERE MAY BE ≠INTERVENING≠ REDEFINITIONS OF THEM INSIDE PROCEDURES CALLED BY THIS SUBROUTINE.

WITH FUNCTION CALLS, THERE IS NO NEED TO LINK PARAMETERS BACK FROM THE FUNCTION TO THE CALLING STATEMENT BECAUSE PARAMETERS OF FUNCTIONS ARE READ ONLY, BUT THERE IS A NEED TO LINK THE RETURN FUNCTION VALUE BACK TO THE CALLING PROGRAM. AN ADDITIONAL VALUE-RECEIVING PARAMETER IS USED FOR THIS PURPOSE. FOR EXAMPLE, THE FOLLOWING SETL CODE

```
        .
        .
    Y = F(A,B);
        .
        .
    DEFINEF F(X,Y);
            .
            .
       RETURN W;
            .
            .
    END F;
```

WOULD BE COMPILED AS IF IT WERE

```
        .
        .
    F(A,B,Y=);
        .
        .
    DEFINE F(X,Y,RT=);
            .
            .
       RT = W;   RETURN;
            .
            .
    END F;
```

THE TEMPORARY VARIABLE -RT- IS A SPECIAL PARAMETER WHICH IS ALWAYS USED FOR PASSING BACK RETURN VALUES.

FOR CALLING A FUNCTION WHICH MAY ALSO BE A MAP OR OTHER INDEXABLE OBJECT, THE OPCODES OP¬OF AND OP¬OFN ARE USED. THESE OPCODES INCLUDE IN THEIR SEMANTICS THE ASSIGNMENT OF THE FUNCTION VALUE AND (IN THE CASE OF OP¬OF)

ANY NECESSARY STACK MANIPULATION, SO THE ARGIN AND ARGOUT OPERATIONS CAN BE REPLACED BY DUMMY AUX¬ASN QUADRUPLES. ON THE UNDERSTANDING THAT PROCEDURE VARIABLES AND EXTERNAL PROCEDURES WILL ALWAYS HAVE FORMAL PARAMETERS OF TYPE GENERAL, THE CONVERT-IN AND CONVERT-OUT ASSIGNMENTS ARE OMITTED.

WHEN THE TYPE OF F IS UNKNOWN, THE CALL Y = F(X) EXPANDS TO THE FOLLOWING INTERMEDIATE-LEVEL QUADRUPLES. SINCE F IS NOT A PROCEDURE CONSTANT, THERE IS NO NEED TO USE THE OP¬PLIST QUADRUPLE TO BEGIN THE CALLING SEQUENCE.

```
AUX¬ASN(P1,X,0)
AUX¬ASN(P2,Y,1)
OP¬OF(Y,X,F)
AUX¬ASN(TY,P2,-)
AUX¬DIS(Y,Y,TY)
```

AND THE LAST TWO INSTRUCTIONS CAN BE SHORTENED TO

```
AUX¬DIS(Y,Y,P2).
```

IN THIS EXPANSION, THE FIRST DEFINITION OF Y CAPTURES THE VALUES RESULTING FROM INDEXING A MAP, TUPLE, OR STRING, AND P2 CAPTURES RETURN VALUES FROM FUNCTIONS. THEN AFTER THE AUX¬DIS OPERATION, Y UNITES THESE TWO SETS OF VALUES.

THE CALLING SEQUENCE FOR MULTIPARAMETER INVOCATIONS OF OBJECTS WHICH CAN BE EITHER FUNCTIONS OR MAPS IS ILLUSTRATED BY THE EXAMPLE

```
Y = F(A,B)
```

WHICH EXPANDS TO

```
AUX¬ARB(T,F,-)
AUX¬TL(T,T,-)
AUX¬TL(T,T,-)
OP¬ARGIN(P1,A,0)
OP¬ARGIN(P2,B,0)
OP¬ARGIN(P3,Y,1)
OP¬OFN(Y,3,F)
AUX¬DIS(Y,T,P3)
OP¬FREE(3,-,-)
```

IN THIS CODE SEQUENCE THE PSEUDO-TEMPORARY, T, CAPTURES VALUES FROM INDEXING MAPS, AND TY CAPTURES FUNCTION RETURN VALUES. NOTE THAT THE OPTIMIZER TREATS THE ACTUAL CALL (OP¬OFN) AS A NO-OP, AND OBTAINS ALL OF ITS INFORMATION ABOUT THE RESULT OF THE OPERATION FROM AUXILIARY DUMMY OPERATIONS.

ALTHOUGH THE STATEMENT Y = F≤A,B≥ CANNOT BE A FUNCTION CALL, FOR CONSISTENCY IT IS EXPANDED USING THE SAME CONVENTIONS AS WERE USED FOR THE PREVIOUS CASE. THUS, IT BECOMES

```
AUX¬ARB(T,F,-)
AUX¬TL(T,T,-)
AUX¬TL(T,T,-)
AUX¬SET(T,T,-)
OP¬ARGIN(P1,A,0)
OP¬ARGIN(P2,B,0)
OP¬ARGIN(P3,Y,1)
OP¬OFAN(Y,3,F)
AUX¬ASN(Y,T,-)
OP¬FREE(3,-,-)
```

THIS TIME THE AUX¬ASN OPERATION, WHICH REPLACES THE AUX¬DIS, MAKES THE ASSIGNMENT TO Y BY THE OP¬OFAN OPERATION APPEAR TO THE OPTIMIZER TO BE A USELESS ASSIGNMENT SINCE T

IS ASSIGNED TO Y BEFORE Y IS EVER USED AGAIN. THE AUX¬SET OPERATION SIMULATES THE CREATION OF A SINGLETON-SET.

THE STATEMENT Y = F[A,B] MAY ALSO APPEAR WITH F BEING EITHER A FUNCTION OR A MAP. FOR EACH OF THESE INTERPRETATIONS, A VERY DIFFERENT SEQUENCE OF CODE IS REQUIRED. IF F IS A MAP, THEN THE QUADRUPLES FOR THE FOLLOWING SETL CODE ARE USED:

```
T1 = F[A];        Y = T1[B];
```

HOWEVER, IF F IS A PROCEDURE, WE MUST ITERATE THROUGH ALL ELEMENTS OF A AND B WHILE CALLING F AND INSERTING INTO Y, AS IN THE FOLLOWING SETL CODE

```
Y = NL;
(∨ T1 ↱ A, T2 ↱ B)
   Y = Y WITH F(T1,T2);
END ∨;
```

THE INTERMEDIATE-LEVEL QUADRUPLES FOR THIS STATEMENT MAKE USE OF A SPECIAL TYPE-TESTING OPCODE, OP¬IFPROC. THE CONVERT-IN AND CONVERT-OUT ASSIGNMENTS ARE OMITTED AGAIN BECAUSE F IS EITHER A PROCEDURE VARIABLE OR EXTERNAL PROCEDURE.

```
        OP¬IFPROC(F,-,LAB1)
        OP¬OFB(T1,A,F)
        OP¬OFB(Y,B,T1)
        OP¬GOTO(-,-,LAB2)
LAB1:   OP¬ASN(Y,NL,-)
        OP¬NEXT(T1,A,LAB2)
        OP¬NEXT(T2,B,LAB2)
        OP¬PLIST(0,2,3,4)
   2:   OP¬ARGIN(P1,T1,0)
        OP¬ARGIN(P2,T2,0)
        OP¬ARGIN(P3,T3,1)
   3:   OP¬CALL(1,3,F)
        OP¬ARGOUT(T3,P3,-)
        OP¬FREE(-,-,-)
        OP¬FREE(-,-,-)
   4:   OP¬WITH(Y,Y,T3)
        OP¬GOTO(-,-,LAB1)
LAB2:       NEXT QUADRUPLE
```