

A CASE STATEMENT FOR SETL.

IT HAS REPEATEDLY BEEN POINTED OUT THAT LABEL VARIABLES AND COMPUTED GOTO-S ARE UNAPPEALING CONSTRUCTS THAT HAVE NO PLACE IN A LANGUAGE LIKE SETL, ESPECIALLY IN VIEW OF THE POWERFUL CONTROL STRUCTURES IT OTHERWISE PROVIDES. WE PROPOSE TO ADD TO SETL A VARIANT OF THE -CASE- STATEMENT OF ALGOL AND PASCAL, AND TO DO AWAY ALTOGETHER WITH LABEL VARIABLES .

SYNTAX.

```

<CASESTAT>  → CASE. <EXPN> OF. <TBLOCK*1> <ELSEB> <ENDER>
<TBLOCK>    → ( <TAG> <COMTAG*0> ) : <BLOCK>
<TAG>       → <*NAME>
             → <*CONST>
<COMTAG>    → , <TAG>
<ELSEB>     → ELSE. <BLOCK>

```

EXAMPLE.

```

CASE. (TYPE.X) OF.
  (INT. ) : X = X+1 ;
  (REAL.) : X = X+1.;
  (CHAR.) :
    CASE. (+X) OF.
      (1) : X = 2*X ;
      (2) : X = DEC.X ; QUIT CASE.(TYPE.X) ;
    ELSE. X = X(1:2) ;
    END CASE.;
  (BLANK., OM.) : NOOP ;
  ELSE. X = +X+1 ;
END. CASE. ;

```

SEMANTICS.

<EXPN> IS EVALUATED , AND CONTROL TRANSFERS TO THE BLOCK, ONE OF WHOSE TAGS IS THE VALUE OF <EXPN>. AFTER EXECUTION OF THAT BLOCK, CONTROL TRANSFERS TO THE FIRST STATEMENT FOLLOWING THE RANGE OF THE -CASE- STATEMENT. THIS BASIC FLOW CAN BE MODIFIED IN THE FOLLOWING WAYS :

- A) A -QUIT- STATEMENT WITHIN THE RANGE OF THE CASE STATEMENT IS A BRANCH OUT OF THAT RANGE.
- B) CASE STATEMENTS CAN BE NESTED, AS SHOWN ABOVE. BY SUFFIXING THE -QUIT- STATEMENTS IN THE USUAL WAY WITH THE FIRST FEW TOKENS FROM THE SCOPE OPENER, IT IS POSSIBLE TO BRANCH OUT OF SEVERAL CASE STATEMENTS AT ONCE.
- C) IF THE VALUE OF <EXPN> CORRESPONDS TO NONE OF THE TAGS , THEN CONTROL BRANCHES TO THE -ELSE- CLAUSE. THIS CLAUSE IS OPTIONAL. IT IS HOWEVER A FATAL ERROR IF IT IS ABSENT, AND THE EVALUATION OF <EXPN> YIELDS NONE OF THE SPECIFIED TAGS.
- D) NOTE THAT TAGS ARE NOT LABELS, AND CANNOT BE THE TARGETS OF GOTO-S. EACH SUB-CASE BLOCK IS ENTERED ONLY THROUGH THE CASE HEADER. A BLOCK MAY CONTAIN LABELS, BUT THESE ARE LOCAL TO THAT BLOCK, AND CANNOT BE USED AS TARGETS OF GOTO-S EITHER FROM OTHER SUB-CASE BLOCKS OR FROM OUTSIDE THE RANGE OF THE CASE.

TAGS.

THE TAGS ARE EITHER ATOMIC COMPILE-TIME CONSTANTS (1, 3.1415, *TAG*, SET, ,ETC), OR NAMES. ALL THE TAGS OF A CASEBODY MUST BE OF THE SAME TYPE.

IF NAMES ARE USED AS TAGS, THESE NAMES ARE TREATED AS COMPILE-TIME BLANK ATOMS, LOCAL TO THE PROCEDURE WITHIN WHICH THEY APPEAR. THEY CAN BE USED AS OTHER BLANK ATOMS, I.E. COMPARED FOR EQUALITY, ASSIGNED TO OTHER VARIABLES, AND PLACED IN SETS AND TUPLES. THEY CANNOT HOWEVER BE THE TARGETS OF ASSIGNMENTS. THE SAME TAGS CAN BE USED WITHIN DIFFERENT CASE STATEMENTS, AS THEY CAN ONLY BE REACHED BY ENTERING THE CASE HEADER.

IT IS AT TIMES CONVENIENT TO MAKE TAGS AVAILABLE TO OTHER PROCEDURES, TO INCLUDE THEM ACROSS MODULES, ETC. THIS CAN BE ACHIEVED BY HAVING THEM APPEAR IN A GLOBAL DECLARATION BLOCK. THEY MUST BE DECLARED TO BE OF TYPE -BLANK-. NOTE HOWEVER THAT UNLIKE BLANK VARIABLES, TAGS ARE NOT INITIALIZED BY A CALL TO NEWAT., BUT INITIALIZED BY THE COMPILER AS SOON AS THEY ARE SEEN AS TAGS WITHIN SOME CASE STATEMENT.

EXAMPLE.

```

MODULE. FIND;
DECLARE THIS, THAT , THEOTHER :BLANK. ;

DEFINE WHAT(OBJ) ;

CASE. SOME(OBJ) OF.

  (THIS)      :      ...
  (THAT)      :      ...
  (THEOTHER) :      ...
  ELSE        :      ...

END CASE. ; END ;

DEFINEF SOME(OBJ) ;
....ELSE RETURN THAT ; END;
END;

```

```

#THE WORLD IS EVERYTHING
  THAT IS THE CASE#
    L.WITTGENSTEIN
    TRACTATUS, PRGP.1.

```

