SETL Newsletter Number 11 (Continuation of Number 7)

December 22, 1970
David Shields

25. ARBITRARY ELEMENT OF NULL SET IS UNDEFINED ATOM.

The statement "arb = $\ni \underline{\text{nl}}$" results in assignment of as value of arb; however, note that "$\Omega \in \underline{\text{nl}}$" has as value $\underline{f}$.

26. REDUCTION OPERATOR.

A new monadic operator of the form

(a) $[op, \forall x_1 \in e_1,\ \ x_2 \in e_2(x_1) \ldots,\ \ x_n \in e_n(x_1,\ldots,x_{n-1}) |$

$C(x_1,x_2, \ldots x_n)] s(x_1,x_2, \ldots, x_n)$

is defined as follows: to obtain the value val, set val $= \Omega$; obtain $x_1, x_2, \ldots, x_n$ in the standard way; for each set of x's for which the boolean expression $C(x_1,x_2, \ldots, x_n)$ has value $\underline{t}$, evaluate the expression $s(x_1,x_2, \ldots, x_n)$. After the first evaluation of s, set val = s; after succeeding evaluations of s, set val = val op s, when op is a binary operator. Continue until no more x's are obtained. More succinctly, we define this in SETL by

```
(b)  times = 0; val = $\Omega$ ;
     ($\forall x_1 \in e_1, x_2 \in e_2(x_1), \ldots, x_n \in e_n(x_1, \ldots x_{n-1}) |$
     $c(x_1, x_2, \ldots x_n)$)
     if times eq 0 then times = 1; val = $s(x_1, x_2, \ldots, x_n)$;
     else val = val op $s(x_1, x_2, \ldots, x_n)$;; end $\forall x_1$;
```

For example, the expression

(c) $$\text{ext} = \sum_{i=1}^{n} \max(a_i, \min_{\substack{1 \leq j \leq m \\ b_j > 0}} b_j)$$

could be written (in SETL) as

$$\text{(d)}\quad ext = [+,\ 1 \leq \forall i \leq n]\ a(i)\ \underline{max}$$
$$[\underline{min},\ 1 \leq \forall j \leq m \mid b(j)\ \underline{gt}\ 0]\ b(j);$$

27. UNDERLINES. Underline only operators in prefix and infix form, and the special symbols.

$\underline{nl}\quad \underline{t}\quad \underline{f}\quad \underline{true}\quad \underline{false}\quad \underline{null}\quad \underline{nullc}$

28. ASSIGNMENT FORM OF EXISTENTIAL OPERATOR. Any occurrence of "$\exists x$", where x is a variable, in a boolean expression, may be replaced by "$\exists [x]$", in which case x is assigned as value Ω, if no such x exists, or else the value of x for which "$\exists x$" has value $\underline{t}$. Note that in expressions of form "$\exists x$" we view x as a dummy variable, and in "$\exists [x]$" we view x as a SETL variable, since it is assigned a value. Note that in expressions of the form ",first $\underline{compop1}$ $\exists$ [x] $\underline{compop2}$ last," the "trial" values are taken in order from first to last; where first and last have integer values, compop1, compop2 are comparison operators (see (22), Newsletter Number 7). For example, to set i to indicate the last positive element (if any) in a sequence a of integers, write

$$\#a \geq \exists [i] \geq 1 \mid a(x)\ \underline{gt}\ 0 \ldots$$

29. NOTE THAT NO AUTOMATIC CLOSING BY END STATEMENT. Unclosed inner loops are not automatically closed by an $\underline{end}$ statement. For example,

$$(\forall x \in s)\ (\forall y \in t)\ z = fun(x,y);;\ end\ \forall x;$$

is correct, while

$(\forall x \in s)$ $(\forall y \in t)$ z = fun(x,y); end $\forall$x;

is not (the unclosed y loop is not closed up by "end $\forall$x" statement).

30. QUIT STATEMENTS. Loops may contain statements of the form "quit;" which result in a branch to the statement immediately following the loop. That is

(a) $(\forall x1 \in e1, \forall x_2 \in e_2(x_1), \ldots,$

$x_n \in e_n(x_1, x_2, \ldots, x_{n-1}) \mid C(x_1, \ldots, x_n)$

blockf; quit; blockl; end $\forall$x; next;

is equivalent to

(b) $(\forall x1 \in e1, \ldots \mid C(x_1, \ldots, x_n)$ til done; blockf; go to done;

blockl; [done] next;

31. CONTINUE STATEMENT. Iteration blocks may contain statements of the form

continue; or continue var;

which are to be interpreted as a branch to end of block with the SETL variable var as its leftmost iteration variable; or to the innermost block for the "continue;" statement. For example,

(a) $(\forall x \in a)$ s1; $(\forall y \in b)$ s2; if cond then continue
else s3; s4; end $\forall$y
s5; end $\forall$x;

is equivalent to

(b) $(\forall x \in a)$ s1; $(\forall y \in b)$ s2; if cond then go to end y;
else s3; s4;

```
          [endy]end ∀y; s5; end ∀x;
```

when the si's denote SETL statements.

32. NEW SYMBOLS FOR SELECTION OPERATORS. The symbols ✳ and - are no longer legal as selection operators. Use hd (read "head") for the former ✳ and tl (read "tail") for the former - . For example, "hd ⟨a,b⟩" has value a, "tl ⟨a,b⟩" has value b. Note that hd and tl are monadic operators, and thus should be underlined.

33. DOING OPTION FOR WHILE STATEMENTS. A while statement of the form

```
     (a)  (while cond doing blocka) blockb; end while
```

when cond is a Boolean expression and block a, block b are SETL blocks, is defined to be equivalent to

```
     (b)  (while cond) block b; [cont] block a; end while;
```

A "continue" statement in block b will result in transfer to first statement in block a. For example,

```
     var=0; i=1; (while i lt 100 doing i =i+1)
                if a(i) lt 0 then continue;;
     var = var+a(i); end while.
```

is equivalent to

```
     var=0; i=1 (while i lt 100) if a(i) lt 0 then go to cont;;
     var = var+a(i); [cont] i=i+1; end while;
```