SETL Newsletter # 143

J.T. Schwartz
February 3, 1975

'Arguments from Use' in the Proof of Relationships of Inclusion and Membership

The methods for establishing relationships of inclusion and membership which are outlined in Newletter 130 are 'arguments from definition', i.e., start by asserting that the output of an operation has certain inclusion/membership properties if the inputs are assumed to have certain corresponding properties of the same kind. It is well worth noting that, in addition to arguments of this general sort, there exists a significant 'argument from use' or 'backward agrument' which can be used to refine an analysis of inclusion and membership in SETL programs. The prototypical case is shown in the code line

(1) `f(r(x)) = f(r(x)) + 1;`

which we assume to appear in some SETL program in which f is known to be a tabulated integer-valued mapping, and where r is an expression without side effects. If only forward analysis were used, the statement (1) might be assumed to enlarge the domain of f; but as a matter of fact it does not. This may be seen as follows:

Since the integer 1 is added to f(r(x)), presumably without error, the value f(r(x)) must be different from Ω; thus r(x) must belong to the domain of f.

A related case is seen in the iteration

$$(\forall x \in s)\ f(x) = f(x) + 1;;$$

if this code is correct, s must be contained in the domain of f.

The general argument is this: if an ivariable x appears as the second argument of a map retrieval operation

$$o = f(x);$$

and if it is known (presumably from type analysis) that Ω is not an acceptable value for o, then must definitely belong to the domain of f.

This property can then be carried over to o- and ivariables of the same program P by the following line of reasoning: properties known for ivariables i' can be propagated back to source ovariables o' if all the variable occurences linked to o' (by the interoccurence linking function uu(o')) have the property in question, and if o' is not linked (by uu(o')) either to a redefinition of its variable or to a program exit. In the case of properties like $i' \in \nabla_1 f$ and $o' \in \nabla_1 f$, we must also be sure that no path from o' to one of its uses i' intersects an operation which can add to the domain $\nabla_1 f$. This argument may most often be employed when i' has o' as its only definition, and when the collection of paths connecting i' and o' is especially simple.