SETL Newsletter # 132

J. Schwartz
June 3, 1974

Some Optimisations using type information.

Several useful optimisations come into reach once the typefinding analysis described in A. Tenenbaum's thesis, cf. also in Newsletter 71 has been carried out. This Newsletter enumerates a few of the possibilities; more will doubtless be noted in the course of time. If 'type' information is combined with 'operator application'information, additional optimisations, which will be described in a later newsletter, become possible.

1. Vectors and Sets of Reals and Integers.

Let V be a vector all of whose components are known to be real. Then V can be represented by an array of word-length real quantities (much as it would be in FORTRAN) rather than by an array of root-words each of which points to a two-word heap block. This speeds up access and cuts by a factor of 3 the amount of space required. Occurences of Ω can be represented using some bit-pattern (such as 'indefinite') that is illegal for reals. This same technique can be used wherever the typefinder establishes that only reals can appear in some particular component of a tuple, or in some particular component of all the tuples of a particular set. Both this technique and some of the others to be described may require that suitable'conversions' be interpolated at points where objects represented in some special form are made parts of composites whose components are represented in a more general SETL style.

Sets of reals can be represented by hashtables which contain the reals directly, rather than holding pointers to them in the ordinary SETL manner.

Unoccupied slots can contain 'indefinites' represented in the manner described in the preceeding paragraph. To avoid the insertion of pointers into hashtable words, a 'rehashing' rather than a 'chaining' search mode can be used.

Obviously a similar technique can be used in connection with variables whose values are known to be real.

Much the same approach can be used to handle variables whose values are known to be integers, or arrays or sets all of whose elements are known to be integers. Of course, some complications result from the fact that integers can either be long or short. Integers appearing in contexts known to be 'integer only' can be represented by full-word quantities with a single bit, the low-bit of the word, reserved as a long-short flag. From this representation of an integer, its true machine-level representation should be recoverable on most machines simply by performing an end-off arithmetic shift. Items flagged as 'long' can have a representation, involving a pointer, much closer to the ordinary SETL representation. In 'integer only' contexts like those we envisage, the quantity Ω can be represented in the 'long' form, but with a special pointer. Integers within sets known to contain integers only can be hashed in the manner described above for reals. Note that the approach suggested here could in principle be used as a standard technique for handling SETL integers, and might lead to somewhat faster arithmetic routines.

2. Subexpressions free of side effects within Boolean expressions.

The typefinder will be able to distinguish subexpressions whose evaluation is guaranteed not to produce side effects. Then, in expressions like e_1 or e_2 (resp. e_1 and e_2) evaluation of e_2 can be bypassed if e_1 evaluates to t (resp. f).

Note that in a SETL context this will almost always save time, since the evaluation even of the simplest SETL expressions is apt to require at least several machine cycles.