SETL Newsletter # 184

Robert B.K. Dewar
March 18, 1977

Extending the Notion of Basing

This newsletter is concerned specifically with base sets consisting solely of atoms. Such base sets correspond roughly to mode definitions in ALGOL-68 for structures. The based maps describe associated data corresponding to fields of the ALGOL-68 structure, and the set of atoms in the base corresponds to the set of ref amode pointers which would appear in the corresponding ALGOL-68 program.

The current implementation of base sets in SETL provides two facilities which are not always used, and are not available in the analogous ALGOL-68 semantics:

1). The ability to locate an atom in the set by hashing.

2). The ability to iterate through the set.

These capabilities cause extra overhead in programs which do not require them, and we address the problem of describing the conditions under which they are not required and modifying the implementation to avoid the overhead.

The hashing is required only for conversion to element of base format. In a number of SETL programs, the only time such a conversion is required, is when newat is converted. Art Grand has already observed that the searching of hash chains can be avoided since newat must be unique. We can extend this observation to note that the hashing itself can be avoided. Bases for which the only conversion to element of format is of this type can be stored as a simple, unhashed linked list with no adverse consequences. This saves the hashing of the newat values and is an essentially trivial change to the library.

If this procedure is followed, then the purpose of storing the base elements in a linked list is to allow iteration through the base. Such iteration is required in response to an explicit iteration in the SETL program (e.g. an iteration through an associated base set), or in the garbage collector (for compacting the indices used for remote objects).

If there are no remotely based objects and no explicit base iterations are used, then there is no point in storing the base as a linked list. Instead, the element blocks can be allocated independantly with the EBLINK field not being used. This is also a case in which EBINDEX values are also unnecessary. If these transformations can be made, then the SETL statements:

```
repr  B: base set {newat},
      H,K,J:   ∈ B,
          LSON, RSON:  based map (∈ B) ∈ B,
          VAL:  based  map  (∈ B) int;
.
.
.
J = newat;
LSON(J) = K; RSON(J) = H;  VAL(J) = 4;
```

have similar semantics and identical implementation to the ALGOL-68 sequence:

```
mode node = struct (int val, ref node &son, rson);
ref node H, J, K;
.
.
.
J := heap node;
&son(J) := K; rson(J) := H; val(J) :=4;
```