Artificial Intelligence Project--RLE and MIT Computation Center
Memo 19--LISP II Garbage Collector

by

Daniel J. Edwards

Motivation

The present LISP free storage control program, the garbage collector, has a severe limitation in that it can handle well only list structure. LISP II will be able to handle arrays, binary programs and other quantities, therefore the garbage collector will have to be able to recognize these quantities and control free storage accordingly. Since arrays and binary programs require blocks of contiguous free storage, the garbage collector must be able to relocate items to be saved in order to coalesce the isolated blocks of items discarded into one contiguous block. The LISP II Relocating Garbage Collector as currently envisioned will operate in the following way.

Marking

Phase one will be a trace through all active list structure, placing marks in a bit table for each word to be saved. In addition, each word to be saved will place a mark in the bit table if its address and/or decrement is a relocatable quantity. Each type of LISP quantity will have its own tracing routine, rules for determining whether the address and/or decrement is relocatable and rules governing the references to other types of LISP quantities. Hence, if when tracing a certain type of LISP quantity, a reference to another type of LISP quantity is encountered, the tracing routine for the second type of quantity will be called on. This will permit the mixing in free storage of LISP quantities in most any fashion the user may wish.

Collection

The collection phase will be a linear sweep through the bit table picking out the words to be collected and entering in to the corresponding location in free storage a pointer to the last word collected in the decrement and putting in the address the number of words collected so far. This quantity in the address will be the local relocation number (i.e., the amount by which everything above it will be moved during the moving phase). If no relocation is desired, the garbage collection may be terminated at this point as a free storage list a la LISP I will have been set up.

Relocation

This phase will be a linear sweep through the bit table hunting for each relocatable address and decrement. When a relocatable quantity is found, the corresponding address or decrement will be looked up in the bit table and a backwards scan initiated for the first free word. The address of this free word will give the local relocation number which will be subtracted from the original address or decrement to give the new address or decrement for that relocatable quantity.

Moving

The final phase will be a linear sweep through the bit table and free storage where every item to be saved is moved down into the free words. After the last item to be saved is moved, the rest of the cells will be a new free storage list in one block.

Conclusion

The free storage control scheme proposed here will provide the necessary flexability to allow LISP II to handle many types of quantities. Since this is only a proposal, any comments, suggestions and criticisms you may have will be considered before actual coding is done.