

STANFORD ARTIFICIAL INTELLIGENCE PROJECT
Memo No. 8
(Preliminary)

September 26, 1963.

STORAGE CONVENTIONS IN LISP 2

by John McCarthy

Abstract: Storage conventions and a basic set of functions for LISP 2 are proposed. Since the memo was written, a way of supplementing the features of this system with the unique storage of list structure using a hash rule for computing the address in a separate free storage area for lists has been found.

The research reported here was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (SI-183).

As in LISP 1.5, marking proceeds in both directions in a list word so that all words accessible from the pushdown list are marked. Relocatable addresses in program are not traced further however.

2. Listing unmarked words. A linear sweep is made through the free storage area and every word not marked in the mark table is put into a chain. Each word of the chain has the number of unmarked words preceding it put in its address part.

3. Relocating addresses. In another linear sweep through storage each word that has a relocatable address or decrement indicated in the mark table is modified. The amount of modification is determined by looking in the location referred to and counting backwards till an unmarked word is found. This word has the amount of modification in its address part.

4. Moving. A third linear sweep moves each word the required amount.

Associated with this memory structure are a number of functions. As in LISP 1.5 we have car, cdr, and cons. Supplementing car [x] and cdr [x] is cwr [x] (contents of the word in register) whose value is the full word contents of the register whose address is x. Words in a block specified by a layout word are obtained by functions like car [x + 3] or cwr [x + 3] 1.

Supplementing cons [x; y] and corresponding to the first three types of layout words listed above we have the following functions:

$$mk1 [n; type; w_1; \dots; w_n]$$

whose value is the location of a layout word of the first kind, labelled with type in its address part and followed by n words with contents $w_1; \dots; w_n$;

$$mk2 [n; type; a_1; d_1; \dots; a_n; d_n]$$

whose value is the location of a layout word labelled with type and followed by n words whose address and decrement parts are specified by the arguments $a_1, d_1, \dots, a_n, d_n$

$$mk3 \left[n; type; r_1, \dots, r_n; \begin{matrix} w_1 \\ a_1; d_1 \end{matrix}; \dots; \begin{matrix} w_n \\ a_n; d_n \end{matrix} \right]$$

Here the r_1, \dots, r_n specify the relocatability of the n following words and the w's or a-d pairs give the entries themselves. We do not define a mk4 function, at least for the present.

The analogues of rplaca and rplacd will exist only in the program feature which will be as like ALGOL 60 as is reasonable. One will write in the M-language

$$\begin{aligned} cwr [x + 3] & : = 3.14 \\ \text{or } cwadr [x + 3] & : = 4.2 \\ \text{or } caddr [x] & : = (PLUS A B) \end{aligned}$$