

LISP II Project
Information International, Inc.
Memo 1B

Corrections to Memo 1A

March 8, 1965

Michael Levin

This correction sheet is intended to keep Memo 1A current. More corrections will be issued from time to time, and a new memo released when they become too numerous. The corrections are numbered according to the paragraphs of the original.

1. The legal characters are letters, digits, and the period (.). The first character must be a letter. The @ is not used.

1.1. The reserved identifiers which the user must stay away from unless knowledgeable are those that end with a period.

1.2. Change TAIL@ to TAIL

2.1. Quoting must be distinguished from the string delimiter #.

Examples:

source language

internal language

3
#AB#
'X
'(U V 2.7)

3
#AB#
(QUOTE X)
(QUOTE (U V 2.7))

2.3. Change to (COND p₁ e₁ ... p_n e_n)

2.4. In the source language, left arrow has very high left precedence, and very low right precedence.

2.5. (FROM A) has the value...

2.8. There are two additional locative expressions.

If e is any expression, then CAR(e), CDR(e), CAAR(e), etc. are locative expressions. The statement (SET (CAR A) B) has the same effect as (RPLACA A B) has in LISP 1.5.

Let e be an expression of type SYMBOL. The value of this expression is transmitted as a pointer. WORD.(e) if used for value produces an OCTAL integer which is the contents of the word pointed to in memory. When WORD.(e) is used as a locative, it stores the OCTAL type value produced on the right side of the SET into the position specified by the pointer which is the value of e.

WORD. and two primitives that transfer between pointers and integers are needed to write a garbage collector in LISP.

3. Additional statements are: comit, try, select, declare, empty.

3.2. The value of a compound statement is the value of the last non-go statement to be executed. If a compound statement is quit by means of a go to an external label, then it has no value.

The value of a block is determined the same way, unless it is quit by means of a RETURN. There will be a mechanism similar to RETURN that goes to the end of a compound statement.

3.5. Internal language for SELECT: Delete the period after E.

3.1. FOR statements.

The source language is not completely defined at present.

Internal language:

```

(POR <locative> <for element> n+1 <statement> )
<for element> ::= ( <p1> <p2> <p3> ) | ( <p4> <p3> )
<p1> ::= <expression>
<p2> ::= STEP <expression> | RESET <expression> | <empty>
<p3> ::= UNTIL <a-expr> | WHILE <B-expr> | UNLESS <B-expr> | <empty>
<p4> ::= IN <expression> | ON <expression>

```

The expression in p₁ and p₄ get executed exactly once. The expressions following the words STEP and UNTIL are static. The others are dynamic.

The control variable is not implicitly declared at the level of the FOR statement. The value of the FOR statement is the value of the control variable at exit.

3.8. Add a right parenthesis in text. (GO (A I J))

4.2. In SQUARE, the declaration of X as real is redundant. Variables implicitly have the type of their function.

5. At the Stanford meeting, we had decided that undeclared variables would be considered local, and that free variables must be declared FREE. This appears to be unworkable without seriously altering the language. The problem is that one may be inside a LAMBDA, but not inside a PROG. In this case, the use of an undeclared variable cannot be made local unless a PROG is added inside the LAMBDA. This language modification seems messy. Furthermore, if one wants the variable to be free, there is no place to say this, for the only declaration is the LAMBDA one where every variable

mentioned must correspond to a parameter. There is the additional question of whether an unmentioned free variable inside a compound statement should be declared at the level of the statement (turning it into a block) or at an outer block level.

I am proposing that we return to the original idea of declaring all variables including local ones, and leaving free variables undeclared. This has the logical advantage that declaration always means "set up space for it" rather than meaning "don't set up space but pick it up free."

The declare statement is still in effect.

When (DECLARE $d_1 \dots d_n$) is encountered, the list which is cdr of the statement is added to the declarations of the innermost block declaration level currently in effect.

5.1. Change GLOBAL to FLUID

5.2. In the third line from top, the variable after (FOR is U not V.

BEGIN FOR U←1 STEP 1 UNTIL N DO V←V+ SQUARE (A[U]);

5.5. Function declarations have effect only within their lexical scope. In this way, they are like local variables. They may be referenced outside of this scope by means of tailing. In this way, they are like our variables.

5.6. It is necessary to isolate a section of program containing FLUID variables from all other parts. Let $D_1, D_2 \dots$ be declarations. Then (SECTION $D_1, D_2 \dots$) is the same, but isolates the FLUID variables in these declarations from variables of the same name outside the SECTION.

6. TRY:

(TRY <statement> <locative> <statement>)

The first statement gets executed. If EXIT is not encountered, then nothing else happens. If EXIT(ξ) is encountered, then:

<locative> ← EXIT(ξ)

and the second statement is executed.