SHIFT
William Bricken
December 1988

To Lou Kauffman

I've read your article "Reflections on Reflexivity". Delightfully broad in scope, fundamentally relevant. Now I understand the snippets about SHIFT that you have sent in various notes. Toggling between theory and meta-theory is a fundamental issue for AI. Several good theses have addressed the issue, Brian Smith's 3LISP is the best. I believe the content you are thinking about has a deep contribution to CS, just like LoF revolutionizes logical computation. I'll work on it over the next few months. Applications might develop in the treatment of quantification (all variables are universally quantified by default, SHIFT to get existential), in machine learning (programs are run by default, SHIFT to get them as descriptions), and of course, in user interface (SHIFT to get to the user).

I spent a lot of time worrying about how to represent the top level of a Losp program. In a list-based implementation, it's very handy to have a container around the entire expression, as in (A B). Reading for logic, however, the container is unnatural since it negates. Trying to return multiple values,

A B

or a Void

<void>

are also problems in an implementation. The algebraic version of the shift arithmetic is an elegant solution. The new computational axioms are:

```
( ) A = ( )
A (A) = A ( )
(((A)) B) = (A B)        formerly ((A)) = A
```

By enforcing

(( )) =/=> <void>

both of my problems are solved. And this maneuver points out a representational inconsistency. In

A (A) = A ( )

the space inside the parens can have an arbitrary structure in it, and most folks write this as

$$A\ (A\ B) = A\ (B)$$

But how to represent algebraically

$$(((A))) = (A)$$

The B is mandatory:

$$(((A))\ B) = (A\ B).$$

I have some difficulty understanding the [ ] notation (Section 5), and consequently the proof that

$$N! = [N!]$$

in Section 6. The problem might be in the introduction of N. I am also not clear about the difference between > and = in the pattern

$$A > [A] = Ex, \quad x = A.$$