TAKEUCHI83

Takeuchi, I., Okuno, H., and Ohsato, N.; Tao--A Harmonic Mean of LISP, PROLOG and Smalltalk; SIGPLAN Notices 18, 7 (July 1983) pp. 65-74.

The authors have developed a new language which is essentially a dialect of LISP. They clearly belong to the "kitchen sink" school of language design since they have taken LISP and thrown in many of the features of PROLOG and Smalltalk. The language, called Tao, will be used on a microcoded Lisp machine.

The authors take the pattern matching ("unification") and logic programming from PROLOG, but by putting it into LISP, they remove many of the advantages. It is no longer simple (having acquired LISP syntax), and in order to make it compatible, they have introduced a flurry of different variable types, quotes, etc. Since their machine is purely sequential, they have eliminated the parallel aspects of PROLOG also.

Along with logic programming from PROLOG, Tao contains object-oriented programming from the MIT LISP Machine (ZetaLisp). These objects, called Flavours, are somewhat different from Smalltalk's. Of course, there is a different syntax for dealing with objects in Tao.

The authors attempt to justify their new language with a number of questionable claims. They "question whether it is natural for humans to write in [PROLOG's] logical forms" where presumably LISP's forms are more natural. Tao is a complicated language, they say, "since it reflects natural, but rather complicated ways how [sic] a human thinks." Finally, they make the amazing claim:

> A programming language design is similar to the design for an unrestrictedly high-performance automobile engine; the designer should do his best to achieve the highest performance and richest functionalities within his design environment, without considering air pollution or drivability, etc.