

Editorial

Welcome to the *Journal of Functional Programming*! Our purpose is simple: to provide an archival journal for the publication of scholarly papers involving all aspects of the functional programming paradigm. We sincerely hope that you, the reader, will be able to participate with us in what we feel is an exciting, important, and long-overdue enterprise.

The field of functional programming has come a long way since Backus' Turing Award Lecture in 1978, and the appearance of *JFP* seems quite natural to us. Professional activity alone is a testament to the vitality of the field. Two conferences – the *ACM Symposium on Lisp and Functional Programming* and the *ACM/IFIP Conference on Functional Programming Languages and Computer Architecture* – have been meeting in alternate years since 1980. Workshops covering all aspects of the field from implementation concerns on one hand to purely theoretical concerns on the other have taken place at a pace so frequent that it would be difficult to list them all here. And aside from these dedicated conferences, countless papers on functional programming have appeared in related conferences and workshops.

Yet the growth is clearly not abating. An IFIP Working Group on Functional Programming (WG2.8) was formed in 1988. At least a dozen books on functional programming are now available, almost all appearing within the past few years. Courses on functional programming are becoming commonplace in academia, and new and better functional languages and implementations continue to appear.

Given all this activity, the appearance of this journal should be a surprise to no one – it is a natural step in the growth of an exciting and important field.

Scope

Our scope is not modest: anything relating to functional programming! However, if forced to be specific, we would group topics into four categories.

Foundations of functional programming includes topics such as formal semantics, abstract interpretation, combinators and lambda calculi, term and graph rewriting, logic, type theory, and category theory.

Implementation of functional languages includes compilation strategies for both uniprocessors and parallel processors; design of novel architectures such as dataflow and graph reduction machines; systems such as garbage collection and I/O; and environment issues such as debugging, profiling, and configuration management.

Linguistics includes the study of specific functional languages and language features such as non-determinism, side effects, and logical variables; methodologies that facilitate reasoning, proof and transformation of programs, including partial

evaluation and program synthesis; and the relationship to other programming paradigms.

Finally, under

Applications we consider the use of functional languages in solving real-world problems, either in isolation or as a basis for embedded systems, reports of practical experience, programming techniques, and prototyping.

Of course, the list is not exhaustive. In particular, although the hallmark of functional programming is perhaps its emphasis on sound *principles*, it is also true that these very same principles have had significant influence on other programming paradigms, and we see this as a Good Thing. Thus we intend to include articles on both ‘pure’ and ‘impure’ functional languages, as long as the principles are evident and worthy in their own right.

Format

JFP will obviously concentrate on the publication of scholarly papers within the scope mentioned above. Papers may be technical, tutorial, or survey in nature. In addition to full papers, we will also publish short papers which might briefly describe an interesting idea whose timely publication seems desirable (although no formal distinction will be made between long and short papers). A unique aspect of the publication format will be the inclusion of a ‘capsule review’ with each paper, written by someone on the editorial board, for the purpose of giving an outsider’s summary of the highlights of the paper. (This format is modelled on a similar format used by DEC System Research Laboratory for their technical reports.)

We will also on occasion publish technical reviews of books and papers that we feel would be of interest to the community. Correspondence on previously published papers will also be transmitted as necessary.

Finally, to really spice things up, we intend to have two regular columns in the journal: The first, *Functional Pearls*, will contain examples of interesting programs, derivations, applications, etc., and for the first few issues will be written by Richard Bird. The second column, *Pearls in Theory*, will contain interesting theoretical problems involving the lambda calculus, types, etc., and for the first few issues will be written by Henk Barendregt. We hope that you enjoy these challenging and entertaining columns.

Final Remarks

It has been very encouraging to witness the enormous growth in interest in functional programming over the last decade. Yet future exploitation will depend on more than ‘just’ great ideas, technical brilliance, and robust implementations – it will require *good marketing* (not to mention a little patience)! Hopefully *JFP*, through the timely publication of our community’s efforts, will facilitate this process, document our progress, and stimulate new directions for further research. We look forward to the opportunities thus granted to us.

HENK BARENDREGT PAUL HUDAK JOHN HUGHES PHILIP WADLER