

## EDITORIAL

### 1 Practice and experience papers

In the end, research on functional languages does little good unless they are used to write something other than compilers for functional languages. However, if one scans a typical functional programming conference or journal, one mainly sees papers on twists in language design, speed-ups in compiled code, clever new analyses, or refinements to semantic models. It is much less common to see a paper that considers a functional language as a tool to some other practical end. We would like to see this change.

The *Journal of Functional Programming* carries, and will continue to carry, articles on all aspects of functional programming from lambda calculus theory to language design to implementation. But we have specially sought, and will continue to seek, papers on functional programming practice and experience.

Research and papers on practice and experience sometimes receive less attention because they are perceived as possessing less academic content. So we want to remind potential authors that we have published a number of papers on this topic in the past, and to spell out the criteria we apply to such papers.

A ‘practice and experience’ paper might

- Describe the use of a functional language in an industrial setting. For example, the March 1997 issue described Natural Expert, a functional language used to access databases.
- Describe a particular application written in a functional language. There is an upcoming paper about a distributed communication system written in ML, and the July 1995 issue was devoted entirely to applications.
- Describe the use of functional programming in a new application area. We have accepted papers on using Haskell to describe fuzzy logic (forthcoming), using SML to specify TeX’s layout algorithm (September 1997), and using Haskell to score music (May 1996).
- Describe new algorithms, data structures, or libraries, written in a functional language. Recent examples include papers on functional arrays (September 1997), parser combinators (May 1996), and the constrained shortest-path problem (January 1996).
- Describe environments and tools related to the use of functional languages. An upcoming paper discusses an Emacs mode for indenting Haskell programs, and a paper in January 1993 described an interactive environment for Lazy ML.
- Suggest an interesting way to teach either functional programming or an application area. The January 1993 issue was devoted to functional languages

in education, including an article on teaching functional programming to business majors.

- Give a careful critique of a language or a programming style. Appel's 'Critique of Standard ML' (October 1993) is an outstanding example.

Such papers don't have to make novel contributions to either functional programming or to the application area. An application may be of interest because of (rather than in spite of) being entirely straightforward, since others might hesitate to write a similar application in a functional language without evidence that it would be tractable.

Papers should clearly summarise their contributions. Is there a new technique – or is the point that the application is straightforward, and no new technique is required? Did it make a difference writing in a functional style – or could the same application be written the same way in an imperative language? What lessons were learned? Were there any reusable programming techniques? And so on.

Lastly, the paper should not be sloppy. In particular, it must give an account of the application area that would be regarded as well-informed, up to date, and accurate by an expert in that field.

These sorts of papers can be hard to get published in conferences, because they tend to be a little long, and because they may not report crisp new research results. JFP is delighted to publish them, provided they meet the criteria above. So write on!

## 2 Online supporting material

For some (but not all) papers it may be useful to accompany the paper with source code, data, or other material, in machine-readable form. This is particularly appropriate for papers describing programming techniques, combinator libraries, abstract data types, and modest applications. It may be less appropriate for tutorials, surveys, theoretical work, or large implementations.

From early 1998, you can now supply online supporting material for your paper. Here's how it works:

- You provide a set of one or more files; one of them must be 'index.html', an HTML home page for the material. We will arrange a link from the abstract of your paper (available in a couple of clicks from the JFP web site) to the 'index.html' file that you provide. The printed copy will also say how to find the material.
- We permanently maintain all the files you supply, at the JFP web site. The 'index.html' home page should contain links (direct or indirect) to all the other files. Some of these other files might be tar'd up bundles for convenient download; whatever you provide we'll use.
- Once your paper, and its accompanying material is published, the Web-accessible material will not be changed. You may, at any time, ask to *add* new material to your 'index.html' page, but you cannot delete or change any existing material. The whole point of a journal is to capture a particular,

self-consistent snapshot. The source code (or whatever) that you provide is mutually consistent with the printed material and, because we keep it, it will remain so. For clarity, you may wish to date the original pages, and any new material that is added.

- That is not to say that errors cannot be corrected. Your subsequent additions (if any) to the online supplementary material may certainly take the form of errata, correcting errors in the paper, or providing new, better, download bundles.
- Your 'index.html' file can also contain a link back to your own home site, where readers can find fuller information, or more up-to-date source code; but any such link must be clearly identified as going outside the JFP archive.

The first paper to take advantage of this is O'Neill and Burton's paper 'A new method for functional arrays' (September 1997). We'd like to encourage more.

*(Received 31 July 1998)*

SIMON PEYTON JONES  
and PHIL WADLER