

Special issue on
**‘Semantics, Applications, and Implementation
of Program Generation’**

WALID TAHA

Department of Computer Science, Rice University, Houston, TX 77025, USA

This special issue of the *Journal of Functional Programming* follows up on the First International ACM SIGPLAN Workshop on the Semantics, Applications, and Implementation of Program Generators (SAIG 2000). The special issue contains eight full length papers, which were received based on an open call for papers. Six of these papers are substantially extended revisions of papers presented at the workshop itself.

Increasingly, we find both end users and software developers either building or using programs that produce other programs. Notable examples include systems like the Fastest Fourier Transform in the West (FFTW), which adaptively produces algorithms optimized for specific architectures, and like many generation- and translation-based implementations of domain-specific languages. At the same time, program generators introduce new research challenges: what are good engineering practices for exploiting program generation? How can we guarantee useful properties (such syntactic or type correctness) of generated programs, *before* they are generated? How do we reconcile the idea of program generation with traditional (and more well-understood) paradigms in various programming languages? This special issue addresses these challenges.

The selected papers cover a diverse spectrum of techniques and approaches, exposing many facets of program generation that deserve systematic study, as well as a range of techniques available for building generative software. The collection includes detailed experience reports from researchers who built significant program generation systems, novel uses of logic programming languages for the automatic synthesis of scientific programs, a successful application of partial evaluation to a security problem, a new approach to compiling domain specific languages via generative combinators, and a number of foundational studies into program transformation and multi-stage languages.

Since its inception, SAIG has continued to grow. Starting from the year 2002, SAIG is held jointly with the conference on Generative and Component Based Software Engineering (GCSE) as one unified conference called the International ACM SIGPLAN/SIGSOFT Conference on Generative Programming and Component Engineering (GPCE'02). Whereas the SAIG community focuses on formal programming language aspects of generation, the GCSE community approaches program automation from a contemporary software engineering viewpoint. Together, the combination of SAIG and GCSE is expected to provide simultaneously the depth

of theory and practice that one would expect in a premier research conference. Excellent submissions from *both* the programming languages and the software engineering communities will be essential for sustaining an intellectually sound conference with real relevance to the world around us. So keep an eye for the next call for papers!