

Foreword

It is our great pleasure to present this special issue, originating from the *6th International Conference on the Mathematics of Program Construction*, held in July 2002 at Schloss Dagstuhl in Germany. Authors of the very best papers presented at the conference have been asked to submit substantially revised and extended versions for inclusion in this issue. The papers have been subjected to the usual refereeing process, culminating in the six papers included here.

The Mathematics of Program Construction series of conferences is concerned with the use of crisp, clear mathematics in the discovery and design of algorithms and in the development of corresponding software and hardware. The six papers included in this volume cover only a fraction of this field, ranging from Kleene Algebra to refinement, and from functional programming to abstract interpretation.

Two papers are concerned with developments of Kleene Algebra. Dexter Kozen gave an invited conference talk on “Some results in dynamic model theory”, with the full paper presented in this issue for the first time. This paper illustrates how Kleene Algebra with Tests (KAT) can be used to give a natural semantics for studying the dynamic properties of models. Joakim von Wright’s conference paper introduced Demonic Refinement Algebra as an extension of KAT for total correctness; the paper “Towards a refinement algebra” included here takes this a step further into a General Refinement Algebra that covers both angelic and demonic non-determinism.

Wim Hesselink presented the theory of refinement using eternity variables in his conference paper, a full version of which will appear in ACM TOCL. The original motivation for, and the application of, this work is presented in his contribution to this volume, entitled “Using eternity variables to specify and prove a serializable database interface”.

Shin-Cheng Mu and Richard Bird study a class of inverse functions that can be characterised explicitly in “Theory and application of inverting functions as folds”. Ralf Hinze, Johan Jeuring and Andres Löh discuss fundamental ideas behind the advanced functional programming language Generic Haskell in their paper “Type-indexed data types”. Finally, Kevin and Roland Backhouse show an application of ideas, germinated in the generic and functional programming community, to abstract interpretation, in their paper “Safety of abstract interpretation for free, via logical relations and Galois connections”.

Our thanks go to the MPC program committee for helping us make the selection of papers for this special issue, to the authors and referees of the papers for their timely cooperation, and to the managing editors of SCP for their support in producing this special issue.

Eerke Boiten
University of Kent
Canterbury, UK
E-mail address: E.A.Boiten@kent.ac.uk

Bernhard Möller
University of Augsburg
Augsburg, Germany
E-mail address: Bernhard.Moeller@informatik.uni-augsburg.de