Guest Editorial: Special Issue on Implicit Computational Complexity

This special issue of *ACM Transactions on Computational Logic* is devoted to implicit computational complexity, following the *Workshop on Implicit Computational Complexity* held in Marseilles in February 2006 as part of the special thematic session on *Geometry of Computation (GEOCAL'06)* at the Centre International de Rencontres Mathématiques. The articles selected were submitted responding to a public call-for-papers after the workshop and have undergone the *TOCL* standard refereeing process.

The area of Implicit Computational Complexity (ICC) has emerged from various propositions to use logic and formal methods to provide languages for complexity-bounded computation. It aims at studying the computational complexity of programs without referring to a particular machine model, external measuring conditions, or particular interpretations, only considering language restrictions or logical/ computational principles entailing complexity properties. Several approaches have been explored for this purpose such as restrictions on primitive recursion and ramification, rewriting systems, linear logic, types and lambda-calculus. They originally often relied on the functional programming paradigm. ICC's two main objectives are:

(1) to find natural implicit logical characterizations of functions of various complexity classes, and
(2) to design systems suitable for static verification of program complexity.

The latter goal in particular requires characterizations which are general enough to validate commonly used algorithms.

The contributions to this special issue illustrate the vitality of this research domain with recent trends corresponding to both objectives just mentioned. The first two contributions can be classified as addressing objective (1). In the article "A New Function Algebra of EXPTIME Functions by Safe Nested Recursion", Arai and Eguchi exploit the recursion-based approach in order to characterize the complexity class of exponential time functions. The article "Context Semantics, Linear Logic, and Computational Complexity" by Dal Lago is devoted to linear logic and its light variants, and explores the use of context semantics for establishing their time complexity bounds. The rest of the contributions address objective (2) The article "Extending the LOOP Language with Higher-Order Procedural Variables" by Crolard, Polonowski and Valarcher explores the relations between functional and imperative languages in the framework of Godel's system T. Then, there are three articles, which are

DOI 10.1145/1555746.1555747 http://doi.acm.org/10.1145/1555746.1555747

related by their goal, namely, the attempt to apply ICC to real Programs. The article "Sup-Interpretations, a Semantic Method for Static Analysis of program resources" by Marion and Péchoux presents some methods, based on semantics, for establishing complexity criteria for functional programs. In the article "A Flow Calculus of *mwp*-Bounds for Complexity Analysis" Jones and Kristiansen consider imperative programs and build a matrix-based methodology to measure bounds on the size of values computed during the execution. Finally, in "Resource Control Graphs", Moyen introduces a new notion of control graphs for studying the termination and space bound properties of programs.

We would like to thank the organizers of *GEOCAL'06*, Thomas Ehrhard and Laurent Régnier, for the opportunity of organizing the workshop and the stimulating scientific context offered by this event. We are grateful to Vladimir Lifschitz for accepting the proposal for a special issue of *TOCL* on ICC. Many thanks also to the referees, who have provided expert evaluation and comments. Finally, we wish to acknowledge the support of the organizations without whom *GEOCAL'06* would not have been possible: CIRM, Université Aix-Marseille II, CNRS, ACI Nouvelles Interfaces des Mathématiques.

Patrick Baillot
Jean-Yves Marion
Simona Ronchi Della Rocca
Guest Editors