

Preface to the special issue: isomorphisms of types and invertibility of lambda terms

RALPH MATTHES and SERGEI SOLOVIEV

*Institut de Recherche en Informatique de Toulouse (IRIT),
C.N.R.S. et Université Paul Sabatier (Toulouse III),
118 route de Narbonne, F-31062 Toulouse Cedex 9, France*

Received 4 March 2008

Isomorphisms of types are computational witnesses of logical equivalence with additional properties. The types/formulas A and B are isomorphic if there are functions (in a certain formalism) $f : A \rightarrow B$ and $g : B \rightarrow A$ such that $g \circ f$ and $f \circ g$ are equal in a certain sense to the identity on A and B , respectively. Typical such formalisms are extensions of simply typed λ -calculus, with $\beta\eta$ -convertibility as equality relation. Another view of a pair of functions $f : A \rightarrow B$ and $g : B \rightarrow A$ (besides establishing the logical equivalence of A and B) is that f is invertible with left-inverse g , and it is then natural to relax the above symmetric condition to just $g \circ f$ being equal to the identity on A . In this situation, A is called a retract of B , which is thus a natural generalisation of the notion of an isomorphism, while both these notions are refinements of the concept of logical equivalence in operational terms, that is, in terms of computable functions.

The study of invertibility of lambda terms from these different perspectives and the closely related subject of subtyping occupy an important place in type theory. It is related to number theory, algebra and category theory, and it has applications to information retrieval systems, automatic code generation, data transformations and coding.

The Institut de Recherche en Informatique de Toulouse (IRIT) hosted the Second International Workshop on Isomorphisms of Types (WIT 2005) in October 2005. The subsequent call for papers for this special issue of *Mathematical Structures in Computer Science* was open to contributions in any of these directions, and three of the four papers presented in this special issue are by participants of WIT 2005. These articles are all single-authored, as is the fourth article by Aleksy Schubert, who participated in the previous workshop WIT 2002, which was also held at IRIT. Since his results fit even better within the scope of WIT 2005, this special issue may justifiably be seen as the post-proceedings of WIT 2005.

The post-proceedings of WIT 2002 were published in 2005 (*Mathematical Structures in Computer Science* **15** (5)), and were mostly concerned with the isomorphisms of types. The articles included in the current issue concern the larger domain described above.

The only paper in this issue devoted to isomorphisms of types in the original sense is *Curry-style type isomorphisms and game semantics* by Joachim de Lataillade, which uses game semantics to give a characterisation of all type isomorphisms with respect to $\beta\eta$ -equality in second-order polymorphic λ -calculus with typing in the style of Curry, that is, without type annotations on the λ -abstractions. A crucial discovery is a new equational

axiom for type isomorphism that is absent from the version with Church-style typing. An appropriate semantic construction for its treatment forms the technical part of these findings.

The paper *An insertion operator preserving infinite reduction sequences* by David Chemouil introduces an operator (or, rather a class of operators) that inserts a reduction into the reduction sequences in such a way that infinite sequences are preserved. This (seemingly technical) result has important applications to the proofs of termination in λ -calculi extended by ‘non-standard’ reductions, which is illustrated by the turning of equations that are only provable by induction into rewrite rules that are added to the framework rules, thus avoiding a traversal of the datatype at hand. The results may have very interesting applications in improving the efficiency of proof assistants.

One of the longstanding candidates for the notion of a subtype was the notion of the retract of a type. Progress in this direction was slowed down by technical difficulties, in particular, by the absence of a good description of retracts, except in some very limited cases. In his paper *On the building of affine retractions*, Aleksy Schubert studies the interesting notion of *affine* retractions. An algorithm to decide affine retractability for simply typed λ -calculus is given in the form of an inference system. It is shown that the problem is, in general, NP-complete, where NP-hardness is established by a reduction to 3-SAT, which requires a deep analysis of the proposed inference system. For the types of order less than 3, a polynomial algorithm is suggested.

The paper *Coercions in a polymorphic type system* by Zhaohui Luo is about subtyping considered as an abbreviation mechanism: if A is considered a subtype/subset of B by way of a function $c : A \rightarrow B$, the type-checking procedure of a flexible programming language should allow the application of functions $f : B \rightarrow C$ to an argument a of type A , with the understanding that the intended term is $f(ca)$ and, likewise, allow the application of $g : A \rightarrow B_2$ to $b : B_1$ when $B = B_1 \rightarrow B_2$, here with the intended meaning $(cg)b$. The contribution of the article is an extension of the polymorphic system of Hindley and Milner by these two forms of coercive subtyping, with a proof of the soundness and completeness of the appropriate extension of the Hindley–Milner type-inference algorithm.

We are grateful to the authors of this issue for their contributions, their patience with the editors and their responsiveness to the suggestions by the referees. The work of the anonymous referees was exemplary, including a real wealth of constructive feedback. This investment in others’ (our authors’) publications is very gratefully acknowledged. We would also like to express our thanks for the financial aid we obtained for the workshop WIT 2005 from the European Union FP6-2002-IST-C Coordination Action 510996 ‘Types for Proofs and Programs’, from the department MathSTIC of the Centre National de la Recherche Scientifique (C.N.R.S.) and from the Université Paul Sabatier in Toulouse. Finally, we, the guest editors, wish to thank Giuseppe Longo, the editor-in-chief of this journal, for having given us the opportunity to prepare this special issue.