SILVIA GHILEZAN
UNIVERSITY OF NOVI SAD, SERBIA
GSILVIA@UNS.NS.AC.RS

SILVIA LIKAVEC
UNIVERSITÀ DI TORINO, ITALY
LIKAVEC@DI.UNITO.IT

Reducibility method and resource control

The basic relationship between logic and computation is given by the Curry-Howard correspondence [4] between simply typed λ-calculus and intuitionistic natural deduction. This connection can be extended to other calculi and logical systems. The *resource control* lambda calculus, $\lambda_{\textcircled{R}}$ [2], is an extension of the λ-calculus with explicit operators for erasure and duplication, which brings the same correspondence to intuitionistic natural deduction with explicit structural rules of weakening and contraction on the logical side [1]. It corresponds to the $\lambda_{\mathsf{cw}}lcw$-calculus of Kesner and Renaud [5]. In $\lambda_{\textcircled{R}}$ in every subterm every free variable occurs exactly once, and every binder binds (exactly one occurrence of) a free variable.

The main computational step is β reduction. But there are also reductions which perform propagation of contraction into the expression and reductions which extract weakening out of expressions. This discipline allows to optimize the computation by delaying duplication of terms and by performing erasure of terms as soon as possible.

Our intersection type assignment system $\lambda_{\textcircled{R}}\cap$ integrates intersection into logical rules, thus preserving syntax-directedness of the system. We assign restricted form of intersection types to terms, namely strict types, therefore minimizing the need for pre-order on types. By using this intersection type assignment system we prove that terms in the calculus enjoy strong normalisation if and only if they are typeable. We prove that terms typeable in $\lambda_{\textcircled{R}}$-calculus are strongly normalising by adapting the reducibility method for explicit resource control operators.

The reducibility method is a well known framework for proving reduction properties of λ-terms typeable in different type systems [3]. It was introduced by Tait [6] for proving the strong normalization property for the simply typed lambda calculus. Its main idea is to relate terms typeable in a certain type assignment system and terms satisfying certain realizability properties (e.g., strong normalisation, confluence). To this aim we interpret types by suitable sets of λ-terms called saturated sets, based on the sets of strongly normalizing terms. Then we obtain the soundness of type assignment with respect to these interpretations. As a consequence of soundness we have that every term typeable in the type system belongs to the interpretation of its type. This is an intermediate step between the terms typeable in a type system and strongly normalizing terms. Hence, the necessary notions for the reducibility method are: type interpretation, variable, reduction, expansion, weakening and contraction properties (which lead to the definition of saturated sets), term valuation, and soundness of the type assignment. Suitable modified reducibility method leads to uniform proofs of other reduction properties of $\lambda_{\textcircled{R}}$-terms, such as confluence or standardization.

References

1. G. Gentzen, "Untersuchungen über das logische Schließen", *Mathematische Zeitschrift*, **39**, 1935, 176–210, 405–431.

2. ghilivetlesclika11 S. Ghilezan, J. Ivetić, P. Lescanne, and S. Likavec, "Intersection types for the resource control lambda calculi", *8th International Colloquium on Theoretical Aspects of Computing, ICTAC '11, Lecture Notes in Computer Science*, **6916**, 2011, 116–134.

3. S. Ghilezan and S. Likavec, "Reducibility: A Ubiquitous Method in Lambda Calculus with Intersection Types", *2nd International Workshop on Intersection Types and Related Systems ITRS '02, Electronic Notes in Theoretical Computer Science*, **70**, 2003.

4. W. A. Howard., "The formulas-as-types notion of construction", In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490, Academic Press, London, 1980.

5. D. Kesner and F. Renaud, "The prismoid of resources", *34th International Symposium on Mathematical Foundations of Computer Science, MFCS '09, Lecture Notes in Computer Science*, **5734**, 2009, 464–476.

6. W. W. Tait., "Intensional interpretations of functionals of finite type I", *Journal of Symbolic Logic*, **32**, 1967, 198–212.