Constructive approach to relevant and affine term calculi

Silvia Ghilezan* Jelena Ivetić* Pierre Lescanne
Faculty of Technical Sciences
University of Novi Sad, Serbia
gsilvia@uns.ac.rs jelenaivetic@uns.ac.rs
École Normal Supérieure de Lyon
University of Lyon, France
pierre.lescanne@ens-lyon.fr

Silvia Likavec*
Dipartimento di Informatica
Università di Torino, Italy
likavec@di.unito.it

In this paper, we propose a new way to obtain a computational interpretation of some substructural logics, starting from an intuitionistic (i.e. constructive) term calculi with explicit control of resources.

Substructural logics [1] are a wide family of logics obtained by restricting or rejecting some of Gentzen's structural rules, such as weakening, contraction and exchange. The most well known substructural logic is the linear logic of Girard [3], in which, due to the absence of contraction and weakening, each formula appears exactly once in the theorem. The other well known substructural logics are the relevant logic (the one without weakening), the affine logic (without contraction) and the Lambek calculus (without all three structural rules).

From the computational point of view, structural rules of weakening and contraction are closely related to to the control of available resources (i.e. term variables). More precisely, contraction corresponds to the duplication of the variable that is supposed to be used twice in a term, while weakening corresponds to the erasure of an useless variable. These concepts were implemented into several extensions of the lambda calculus [4,5,6,2].

Here, we use the resource control lambda calculus $\lambda_{\circledR}$, proposed in [2], as a starting point for obtaining computational interpretations of implicative fragments of some substructural logics, namely relevant and affine logic. The corresponding formal calculi are obtained by syntactic restrictions, along with modifications of the reduction rules and the type assignment system. The proposed approach is simpler than the one obtained via linear logic.

The *pre-terms* of $\lambda_{\circledR}$ are given by the following abstract syntax:

$$\text{Pre-terms} \qquad f \quad ::= \quad x \,|\, \lambda x.f \,|\, ff \,|\, x \odot f \,|\, x <^{x_1}_{x_2} f$$

where x ranges over a denumerable set of term variables, $\lambda x.f$ is an *abstraction*, ff is an *application*, $x \odot f$ is a *weakening* and $x <^{x_1}_{x_2} f$ is a *contraction*. $\lambda_{\circledR}$-terms are derived from the set of pre-terms by inference rules, that informally specify that bound variables must actually appear in a term and that each variable occurs at most once. Operational semantics of $\lambda_{\circledR}$-calculus is defined by four groups of reduction rules and some equivalencies. The main computational step is the standard β reduction, executed by substitution defined as meta-operator. The group of (γ) reductions performs propagation of contraction into the term. Similarly, (ω) reductions extract weakening out of the terms. This discipline allows us to optimize the computation by delaying duplication of variables on the one hand, and by performing erasure of variables as soon as possible on the other. Finally, the rules in $(\gamma\omega)$ group explain the interaction between explicit resource operators that are of different nature.

The simple types are introduced to the $\lambda_{\circledR}$-calculus in the following figure.

*Partially supported by the Ministry of Education and Science of Serbia, projects III44006 and ON174026

Submitted to:
CM:FP 2013

$$\frac{}{x:A\vdash x:A}\ (Ax)$$

$$\frac{\Gamma,x:\alpha\vdash M:\beta}{\Gamma\vdash\lambda x.M:\alpha\to\beta}\ (\to_I) \qquad \frac{\Gamma\vdash M:\alpha\to\beta \quad \Delta\vdash N:\beta}{\Gamma,\Delta\vdash MN:\beta}\ (\to_E)$$

$$\frac{\Gamma,x:\alpha,y:\alpha\vdash M:\beta}{\Gamma,z:\alpha\vdash z<^x_y M:\beta}\ (Cont) \qquad \frac{\Gamma\vdash M:\beta}{\Gamma,x:\alpha\vdash x\odot M:\beta}\ (Weak)$$

In the obtained system $\lambda_{\circledR}\to$, weakening is explicitly control by the choice of the axiom, while the control of the contraction is managed by implementing context-splitting style (i.e. requiring that Γ,Δ represents disjoint union of the two bases).

Modifications of the $\lambda_{\circledR}\to$ system can provide the computational interpretation of some substructural logics, different from the usual approach via linear logic. For instance, if one excludes the $(Weak)$ rule but preserves the axiom that controls the introduction of variables, the resulting system would correspond to the logic without weakening and with explicit control of contraction i.e. to the variant of implicative fragment of relevance logic. Similarly, if one excludes the $(Cont)$ rule, but preserves context-splitting style of the rest of the system, correspondence with the variant of the logic without contraction and with explicit control of weakening i.e. implicative fragment of affine logic is obtained. Naturally, these modifications also require certain restrictions on the syntactic level, changes in the definition of terms and modifications of operational semantics as well.

We also proposed intersection type assignment systems for both the $\lambda_{\circledR}$-calculus and its substructural restrictions, and proved that those systems completely characterize strongly normalising terms of all three calculi.

Although the proposed systems may be considered naive due to the fact that they only correspond to implicative fragments of relevant and affine logics and therefore are not able to treat characteristic split conjunction and disjunction connectives, they could be useful as a simple and neat logical foundation for the programming specific relevant and affine programming languages.

References

1. P.Schroeder-Heister and K.Došen: *Substructural Logics*, Oxford University Press (1993).
2. S.Ghilezan, J.Ivetić, P.Lescanne and S.Likavec: Intersection types for the resource control lambda calculi. In A. Cerone and P. Pihlajasaari: *8th International Colloquium on Theoretical Aspects of Computing, ICTAC '11*, LNCS Vol. 6916, pages 116–134. Springer (2011).
3. J-Y.Girard: Linear logic. In: *Theoretical Computer Science*, Vol. 50(1), pages 1-102, North Holland (1987).
4. D.Kesner and S.Lengrand: Resource operators for lambda-calculus. In: *Information and Computation*, Vol. 205(4):419–473 (2007).
5. P. Lescanne and D.Žunić: Classical proofs essence and diagrammatic computation. In: *Proceedings of International Conference on Numerical Analysis and Applied Mathematics - ICNAAM 2011* AIP Conf. Proc., Vol. 1389, pages 792–797 (2011).
6. V. van Oostrom: Net-calculus. Course notes, http://www.phil.uu.nl/ oostrom/oudonderwijs/cmitt/00-01/net.ps (2001).