Linear Logic, types and languages

Jen-Shin Lin

October 22, 2016

The linear logic has been introduced in 1987 by Jean-Yves Girards [5, 6]. It's a logic system in which truth is considered priced. In other words, very predicates can be used exactly once only.

Abramsky [1] and Benton et al. [2, 3] have provided computational interpretations for both of classical and intuitionistic linear logic. Lots of efforts have been put on studying and manitulating linear logic in the way of sequent calculus. In the end, with some help of Curry-Howard correspondence, several term assignment system and linear lambda claculus have been introduced. Additionally Benton and Wadler [4] have also given a categorial semantics for linear lambda calculus and shown that it can be translate into Moggi's computation model [7].

Wadler [11] has considered neither classical logic nor sequent calculus and started directly with intuitionistic linear logic as natural deduction rules. With introducing several operator and context labels, he has introduced a linear lambda calculus together with a simple linear type system.

Simona Ronchi della Rocca and Luca Roversi [9] have presented another study in which they have designed another lambda calculus as a fully typed language, $\Lambda_!$, for intuitionistic linear logic and defined a categorical semantics as well.

In the end a linear language would provide us some advantages to managing resources as plenty of previous works mentioned. Those linear languages have a relatively naive operational semantics and provide good memory management properties, however, they will re-compute whenever a non-linear variable is required. In the other hand, if one can memorize the results but this will sacrifice some management properties. To solve this problem Turner and Wadler [10] have combined these two interpretations.

The linear logic includes several new connectives: *multiplicative conjunction*, *additive conjunction*, *additive disjunction* and *multiplicative disjunction*. However, traditionally in programming language design the third one will be omitted since there is no simple human institution for it. Therefore previouly designed languages seem like something missing and incompleted in some way. Novitzka and Mihalyi [8] has given a meaning in real world for all four connectives and presented how time and space can be introduced into linear logic and how this can be used in terms of resource-orienteds programming.

We design an experimental functional language syntax in agda. To avoid the difficulty of managing variable names, we applyed de Bruijn index and locally nameless. As in the most of above works, we introduce several new operators for linear operation. In other words, our language can be executed in normal mode or linear mode. Additionally, for supporting both of normal lambda calculus, in which variable can be used as many times as usuall, and linear lambda calculus, we separate normal context into a dual-context design which provides us the capability of manipulating normal variable and linear variable differently. *Source code: https://dl.dropboxusercontent.com/u/11966021/src/LinearTyped.zip.* Besides, for ready ourselve for adding type system, we also implement simple (but not linear) typing rules in terms of natural deduction and sequent calculus. *Source code: https://dl.dropboxusercontent.com/u/11966021/src/SimplyTyped.zip.*

References

[1] Abramsky, S. Computational interpretations of linear logic. *Theoretical Computer Science 111* (1992), 3–57.

[2] Benton, N., Bierman, G., and Paiva, V. Term assignment for intuitionistic linear logic. Tech. rep., 1992.

[3] Benton, N., Bierman, G. M., Hyland, J. M. E., and de Paiva, V. A term calculus for intuitionistic linear logic. In *Proceedings of the International Conference on Typed Lambda Calculi and Applications* (1993), M. Bezem and J. F. Groote, Eds., Springer-Verlag LNCS 664, pp. 75–90.

[4] Benton, N., and Wadler, P. Linear logic, monads, and the lambda calculus. In *In 11 th LICS* (1995), IEEE Computer Society Press, pp. 420–431.

[5] Girard, J.-Y. Linear logic. *Theor. Comput. Sci. 50*, 1 (Jan. 1987), 1–102.

[6] Girard, J.-Y. Linear logic: Its syntax and semantics, 1995.

[7] Moggi, E. Notions of computation and monads. *Inf. Comput. 93*, 1 (July 1991), 55–92.

[8] Novitzka, V., and Mihalyi, D. Resource-oriented programming based on linear logic. In *in Acta Polytechnica Hungarica 2007* (2007), pp. 157–166.

[9] Ronchi della Rocca, S., and Roversi, L. Lambda calculus and intuitionistic linear logic. In *Logic Colloquium '94* (Clermont Ferrand, France, 1994).

[10] Turner, D. N., and Wadler, P. Operational interpretations of linear logic, 1998.

[11] Wadler, P. A taste of linear logic. In *Mathematical Foundations of Computer Science* (Gdańsk, Poland, 1993), A. M. Borzyszkowski and S. Sokolowski, Eds., Springer-Verlag LNCS 781, pp. 185–210.