# Weak Linearization of the Lambda-Calculus

Sandra Alves and Mário Florido

University of Porto
Department of Computer Science & LIACC
R. do Campo Alegre 823, 4150-180 Porto, Portugal
e-mail: {sandra,amf}@ncc.up.pt

We identify a restricted class of terms of the lambda calculus, here called weak linear, that includes the linear lambda-terms keeping their good properties of strong normalization, non-duplicating reductions and typability in polynomial time. The advantage of this class over the linear lambda-calculus is the possibility of transforming general terms into weak linear terms with the same normal form. We present such transformation and prove its correctness by showing that it preserves normal forms.

A $\lambda$-term $M$ is *weak linear* if in any reduction sequence of $M$, when there is a contraction of a $\beta$-redex $(\lambda x.P)Q$, then $x$ occurs free in $P$ at most once, i.e., when a function $\lambda x.P$ is applied, its formal parameter $x$ must occur at most once in the function body. For example the term $\lambda x.xx$ is weak linear because it is a non-linear $\lambda$-abstraction which is never applied. The term $(\lambda x.xx)I$ is not weak linear because it has a redex where the function is not linear. For weak linear terms, only functions that can be applied to an argument in the reduction process are required to be linear. Notice that our definition does not refer only to $\beta$-redexes $(\lambda x.P)Q$ that are subterms of the original term $M$, but to abstractions $\lambda x.P$ that are going to be the function part of a $\beta$-redex in the reduction of $M$. For example the term $M \equiv ((\lambda x.x)(\lambda x.xx))k$ is not weak linear although it does not have any subterm of the form $(\lambda x.P)Q$ with $x$ occurring more than once in $P$. The problem is that there is a redex of this form (in this case, $(\lambda x.xx)k$), in a reduction sequence from $M$.

To deal with transformation of redexes which will appear during the reduction process (the *virtual redexes*) our transformation uses *legal paths*, [2, 3], because this notion provides a formal characterization of the intuitive notion of *virtual redex*. This contribution is also significant for the methodology it develops. What we set up is a new use of legal paths for complex term transformation.

For a more complete discussion, see [1].

## References

1. Sandra Alves and Mario Florido. Linearization by program transformation. In *Proceedings of the International Symposium on Logic-based Program Synthesis and Transformation (LOPSTR 2003)* , 2003.
2. Andrea Asperti, Vincent Danos, Cosimo Laneve, and Laurent Regnier. Paths in the lambda-calculus. In *Logic in Computer Science*, pages 426–436, 1994.
3. Andrea Asperti and Cosimo Laneve. Paths, computations and labels in the lambda-calculus. *Theoretical Computer Science*, 142(2):277–297, 1995.