

Addendum to
“New Notions of Reduction and
Non-Semantic Proofs of
 β -Strong Normalization in
Typed λ -Calculi”*

A. J. Kfoury
kfoury@cs.bu.edu
Dept. of Computer Science
Boston University

J. B. Wells
jbw@cs.bu.edu
Dept. of Computer Science
Boston University

March 31, 1995

Boston University Computer Science Department
Technical Report 95-007

This addendum to our technical report of December 1994 [KW94b] has several purposes:

1. To clarify some statements about the focus and purpose of the paper which were written unclearly.
2. To add information on research by others on the notion of reduction which we call γ -reduction and other similar transformations.
3. To discuss the closely-related research by P. de Groote of which we were unfortunately unaware. This discussion has three parts:
 - (a) A comparison of the way de Groote reduces the β -SN problem to a weak normalization problem which is very close to our method.
 - (b) A discussion of how de Groote’s proposed β -SN proof for the simply-typed λ -calculus fails and how our proof avoids the flaw in de Groote’s proof.
 - (c) A comparison of the different ways in which our paper and de Groote’s paper go beyond the simply-typed λ -calculus.
4. To discuss an earlier method by Klop for reducing a β -SN problem to a weak normalization problem, which could be extended to a β -SN proof for the intersection-type system by using the method of our paper.
5. To fix the incorrect proof of Lemma 3.2.

We will discuss these items in this order.

*This work is partly supported by NSF grant CCR-9113196.

1 Clarification

Although we were careful in our “Background and Motivation” section to specify that we were discussing β -SN proofs for type systems more powerful than the simply-typed λ -calculus, the wording in the rest of the paper was not so careful and a comment which was intended to go at the beginning of Section 5 was placed at the beginning of Section 4 instead. As a result, the paper erroneously made false claims about the history of proving the β -SN property for the simply-typed λ -calculus. Many readers have pointed out numerous methods for proving the β -SN property for the simply-typed λ -calculus that are not variations on the methods of Tait or Girard. The methods mentioned have included proofs by Gandy, Nederpelt, Klop, de Groote, de Vrijer, van de Pol and Schwichtenberg [Gan80b, Ned73, Klo80, dG93, dV87, vdPS95].

Our intention was to discuss proofs of the β -SN property for more powerful type systems such as system F, the system of intersection types, and the system of positive recursive types. These are the usual polymorphic extensions of the simply-typed λ -calculus. Although there is a proof of the β -SN property for the simply-typed λ -calculus in our paper, we consider this an exercise in preparation for extending the result to the intersection type discipline. The β -SN proof for the simply-typed λ -calculus is given to make it easier for the reader to understand the proof method.

Until now, the β -SN property has always been proven for polymorphic type systems using some variant of the semantic methods of Tait or Girard. All of the known proofs of the β -SN property for system F are based on Girard’s method, although de Groote has described a way (of which we were unaware) to use Scedrov’s simplification of the conditions on reducibility candidates, which were designed for a weak normalization proof, to prove strong normalization [dG93, Sce87]. The proofs of the β -SN property for the system of intersection types by Pottinger and Leivant are based on Girard’s reducibility candidates, although a more recent proof by van Bakel is based directly on Tait’s method without requiring the reducibility candidates [vB92]. The only published proof of β -SN for the system of positive-recursive types by Mendler uses reducibility candidates [Men91]. Since we were unaware of the proof by van Bakel, we must modify our claim that proofs of β -SN for polymorphic extensions of the simply-typed λ -calculus have used reducibility candidates to state instead that these proofs are based on either Tait’s or Girard’s method. We keep our claim that all such proofs have used semantic methods.

2 Related Research on γ -Reduction

Several readers have pointed out research which either used the notion of reduction which we call γ -reduction or a similar transformation. These transformations are size-preserving and act on the structure of λ -terms without changing their “meaning”. We will call these notions of reduction *structural reductions*. The notion of reduction we call γ (which we called θ_4 in [KW94a]) is the reduction relation satisfying this requirement:

$$((\lambda x.(\lambda y.N))P) \xrightarrow{\gamma} (\lambda y.((\lambda x.N)P))$$

There is another very similar reduction which we will call θ_1 in this addendum (to be consistent with our earlier paper [KW94a]):

$$(((\lambda x.N)P)Q) \xrightarrow{\theta_1} ((\lambda x.(NQ))P)$$

There is another structural reduction which is used for a different purpose, but which it is necessary to mention. We will define this reduction here and give it the name θ_3 (which is the name used in [KW94a]):

$$(N((\lambda x.P)Q)) \xrightarrow{\theta_3} ((\lambda x.(NP))Q)$$

Starting around 1989, a number of researchers independently began using the γ , θ_1 , and θ_3 reductions. The first reference of which we are aware is a Ph.D. thesis by Vidal in 1989 [Vid89], which we are told used something like γ and θ_1 . Also in 1989, Moggi used θ_3 along with many other transformations in a paper [Mog89]. In 1990, Kfoury, Tiurnyn, and Urzyczyn used θ_1 and θ_3 together with another transformation as part of the proof that typability in ML is DEXPTIME-complete [KTU90, KTU94]. A 1992 paper by Kfoury and Tiurnyn uses a variant of γ (denoted “ $()_L$ ”) in analyzing the rank-2 restriction of system F [KT92]. Also in 1992, Sabry and Felleisen introduced a generalization (denoted “ β_{ift} ”) of θ_1 and θ_3 [SF92]. This development by Sabry and Felleisen is especially interesting since the β_{ift} rule was derived from equivalences induced by continuation-passing style transformations. In 1993, de Groote used a restriction of θ_1 in a paper that will be discussed in more detail later in this addendum. Kfoury and Wells used restrictions of both γ , θ_1 , and θ_3 together with another transformation for a further analysis of typability in the rank-2 restriction of system F [KW94a]. In a 1994 paper, Regnier used both γ and θ_1 to aid in a proof that the perpetual reduction strategy (described in [Bar84, § 13.4]) finds a longest reduction path for a β -SN term [Reg94]. In a 1995 paper, Ariola, Felleisen, Maraist, Odersky, and Wadler use θ_1 and a restriction of θ_3 as part of an analysis of how to implement sharing in a real language interpreter in a way that directly corresponds to a formal calculus [AFM⁺95]. It is clear that many researchers are finding these structural reductions useful in many different types of research.

3 Relation to de Groote’s Method

P. de Groote’s 1993 paper [dG93] uses a method for reducing the β -SN problem to a weak normalization problem that is essentially the same as ours in spirit but differs in minor details. Instead of γ -reduction, de Groote uses a restriction of the earlier-mentioned θ_1 -reduction, which de Groote calls β_S . The restriction is to K-abstractions as follows:

$$(((\lambda x.M)N)O) \xrightarrow{\beta_S} ((\lambda x.MO)N) \quad \text{if } x \notin M$$

The most important difference between the two papers is that de Groote uses general $\beta_I\beta_S$ -reduction while we take advantage of a specific reduction strategy of $\beta_I\gamma$ -reduction which we call \star -reduction. Both our paper and de Groote’s paper achieve the nearly identical result that the problem of β -strong normalization is equivalent (respectively) to the problems of \star -normalization and $\beta_I\beta_S$ -normalization. The way de Groote achieves the result is quite different. First, de Groote shows that β_K -reduction steps can be postponed in a sequence of β -reduction and β_S -reduction steps, yielding the fact that if a $\beta_I\beta_S$ -descendent is β -SN, then the ancestor is β -SN as well. Then de Groote defines a calculus with labels to record the number of $\beta_I\beta_S$ -reduction steps that have occurred. A complex argument shows this calculus to be confluent. Since the sum of the labels in a term is a bound on the longest reduction sequence leading to that term, and since all reduction paths from a term with a normal form must eventually reach the normal form, this yields the desired result. Another difference is that we take advantage of the shape of γ -normal forms in many of our lemmas, but β_S -normal forms do not have such nice shapes and in any case de Groote’s method does not go to β_S -normal form after every β_I step.

3.1 Flaw in de Groote’s SN Proof for Simple Types.

Both de Groote’s paper and our paper contain β -SN proofs for the simply-typed lambda calculus. With regard to the simply-typed lambda calculus, de Groote’s 3-sentence proof sketch in Theorem 27 is incorrect. There are two ways to fix the proof, which we describe below.

Unlike in our paper, de Groote freely alternates between β_I -reduction and β_S -reduction, rather than going to β_S -normal form after every β_I step. As a result, de Groote’s proof must show a metric to decrease after every single step, whether β_I or β_S . For the simply typed lambda calculus, de Groote’s metric gives the “order” of a β_I or β_S redex as the size of the type of the abstraction of the redex. The paper claims that there is a $\beta_I\beta_S$ reduction strategy such that any newly created redexes are of lower order. The proof for the case of reducing a β_I redex is given (taken directly from the classic normalization proof by Turing [Gan80a]), although it only deals with the possible new β_I redexes. The proof for the case of reducing a β_S redex is not given.

Consider this β_S redex M :

$$M \equiv (((\lambda x.((\lambda y.P)Q))N)O) \quad \text{where } x \notin ((\lambda y.P)Q)$$

Suppose $y \notin P$. Then $((\lambda y.P)Q)$ is a β_K redex and is not counted by de Groote’s metric. (Note that a β -redex can not be part of a β_I -redex and also overlap with a β_S redex. In our paper, a β -redex can be part of both a β_I and a γ -redex.) Let the type of x be σ , the type of P be τ , and the type of y be ρ . The type of $(\lambda x.((\lambda y.P)Q))$ is $\sigma \rightarrow \tau$. After β_S reduction, we have this result:

$$M \xrightarrow{\beta_S} M' \equiv ((\lambda x.(((\lambda y.P)Q)O))N)$$

The subterm $(((\lambda y.P)Q)O)$ is now a new β_S redex (since $y \notin P$). The order of this new redex is the size of the type of $(\lambda y.P)$, which is $\rho \rightarrow \tau$. The order of the old redex M is the size of the type of $(\lambda x.((\lambda y.P)Q))$, which is $\sigma \rightarrow \tau$. The order of the newly created β_S redex could be much larger than the order of the contracted redex M , because ρ could be much larger than σ .

Any reduction strategy must eventually reduce the β_S redex M or a residual of M . In any residual of M , the abstraction will still look like $(\lambda x.((\lambda y.P')Q'))$ because $y \notin P$ (and therefore $y \notin P'$) and so the interior β_K redex will still be there. Thus, it is not proven that de Groote’s metric decreases at every step of $\beta_I\beta_S$ -reduction.

One way to fix de Groote’s proof is by not including the β_S as part of the metric, adopting a reduction strategy of performing one β_I -reduction step followed by reduction to β_S -normal form, and only checking the metric when the term is in β_S -normal form. It is worth observing that this is essentially what our paper does with \star -reduction.

Another way to fix de Groote’s proof, suggested by de Groote himself, is to let the order of a β_S -redex $((\lambda x.M)NO)$ be the size of the type of M instead of the size of the type of $(\lambda x.M)$. Instead of using the multiset ordering, de Groote’s proof calculates the order of a λ -term as a pair (n, m) where n is the highest order of an individual redex and m is the number of redexes of order m . Under this ordering and using the new definition of the order of a β_S -redex, β_I -reduction of the innermost redex of highest degree is strictly decreasing and β_S -reduction is non-increasing. By adding a proof of the strong normalization of β_S -reduction, the normalization of $\beta_I\beta_S$ -reduction can be proved.

3.2 Comparison on Polymorphic Type Systems.

Our paper extends our method to the system of intersection types, while de Groote’s paper discusses how one might apply the method to the systems of Barendregt’s λ -cube. With regard to the system

of intersection types, it would be possible to use de Groote’s “front end” with our “back end” to achieve the same result, but only after fixing the bug mentioned in the previous subsection.

Regarding the type systems of the λ -cube, de Groote discusses how to apply his method to each dimension of the cube separately. Our methods are similar enough that this discussion applies for both our method and de Groote’s method. Every system of the λ -cube allows terms to be applied to terms and the simplest system which allows only this is exactly the simply-typed λ -calculus. Travelling in one of the three directions in the λ -cube adds one of these three possibilities: terms applied to types, types applied to types, or types applied to terms.

Allowing terms to be applied to types yields system F, the second-order polymorphic typed λ -calculus. In this case, de Groote points out that Scedrov’s semantic method for proving β -normalization for system F, which uses simpler conditions on the reducibility candidates, can be used to prove normalization of $\beta_I\beta_S$ -reduction [Sce87]. It is important to observe that the proof of β -SN that de Groote proposes for system F is still semantic. Any proof of β -SN for system F will involve higher-order quantification, since this fact can not be proven within second-order Peano Arithmetic [GLT89, p. 114].

Allowing types to be applied to types results in β -reduction occurring at the type level. Unless the third possibility (types applied to terms) is also allowed, β -reduction at the type level is separate from β -reduction at the term level, and strong normalization can be proved separately for the two levels.

Allowing types to be applied to terms causes a bit of difficulty, since β_I -reduction can now erase subterms. Since this type system is in the Church style, a β_I -reduction step looks like this:

$$((\lambda x : \tau.M)N) \xrightarrow{\beta_I} M[x := N]$$

Since τ can contain a term, even though x occurs in M , some erasure occurs, unlike in ordinary β_I -reduction. This problem is solved by de Groote by using this alternate form of β -reduction:

$$((\lambda x : \tau.M)N) \xrightarrow{\beta} ((\lambda x : \tau.M[x := N])N)$$

Without also using β_S -reduction (or γ -reduction), this would prevent necessary new β -redexes from being formed. However, β_S -reduction (or γ -reduction) bypasses this problem. In fact, the entire presentation of both our paper and de Groote’s paper could be rewritten using only this form of β -reduction. A recent paper on a “call-by-need” λ -calculus [AFM⁺95] is entirely based on a very similar approach, except the β -reduction is even more restricted: only one instance of x is replaced instead of all of them!

4 Relation to Nederpelt’s and Klop’s Method

In Klop’s extensive Ph.D. thesis [Klo80, Chap. 1, § 8], a simple proof is given for the β -SN property which is inspired by an earlier proof by Nederpelt [Ned73]. Although this proof is not extended to polymorphic type systems, it has much in common with our method and de Groote’s method. Our method and de Groote’s method can be seen as ways of avoiding the erasure that occurs when K-redexes are reduced. The proofs of Nederpelt and Klop do not avoid or postpone reducing K-redexes, but instead retain the arguments that would otherwise be discarded. Klop’s proof in Chapter 1 is actually for the λ -calculus with Hyland-Wadsworth labels, but this implies the SN property for the simply-typed λ -calculus and it is easy to perform the proof directly for the

simply-typed λ -calculus. In Chapter 2 of his thesis, Klop generalizes the method (“reductions with memory”) to allow proving SN for regular CRSs (combinatory reduction systems).

Klop’s method reduces the problem of β -SN to the problem of weak normalization for the λI -calculus extended with a single constant P with the reduction rule $Pabc \longrightarrow P(ac)b$. The purpose of the constant P is to retain in its second argument the “memory” of arguments to β -redexes that might otherwise be erased. The weak β -normalization property of this calculus is proven in the standard way.

Perhaps the most interesting property of Klop’s method is that it appears that it can also be used as a “front end” with our “back end” to prove the β -SN property for the system of intersection types.

5 Corrections to Lemma 3.2

The proof for part 1 of Lemma 3.2 is simply wrong and should be replaced by the following:

Count the number of pairs of subterm occurrences P and Q in M such that P is an application, Q is an abstraction, P contains Q , and there is no subterm (RS) contained within P such that Q is contained within S . Every γ -reduction step reduces this count.

The proof for part 2 of Lemma 3.2 fails to note that it only proves weak confluence and depends on the result of part 1 for the full confluence result, from which the uniqueness of normal forms follows.

6 Acknowledgements

Pawel Urzyczyn spotted the wrong proof in part 1 of Lemma 3.2. Pawel Urzyczyn, Femke van Raamsdonk, Vincent van Oostrom, Matthias Felleisen, and Philippe de Groote mentioned much of the related research to us.

References

- [AFM⁺95] Z. M. Ariola, M. Felleisen, J. Maraist, M. Odersky, and P. Wadler. A call-by-need lambda calculus. In *Conf. Rec. 22nd Ann. ACM Symp. Princ. Program. Lang.* ACM, 1995.
- [Bar84] H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland, revised edition, 1984.
- [dG93] P. de Groote. The conservation theorem revisited. In *Int’l Conf. Typed Lambda Calculi and Applications*, vol. 664 of *LNCS*, pp. 163–178. Springer-Verlag, Mar. 1993.
- [dV87] R. de Vrijer. Exactly estimating functionals and strong normalization. *Indagationes Mathematicae*, 49(4):479–493, 1987.
- [Gan80a] R. O. Gandy. An early proof of normalization by A. M. Turing. In Seldin and Hindley [SH80], pp. 453–455.
- [Gan80b] R. O. Gandy. Proofs of strong normalization. In Seldin and Hindley [SH80], pp. 457–477.
- [GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Number 7 in Cambridge Tracts in Theor. Comp. Sci. Cambridge Univ. Press, 1989.
- [Klo80] J. W. Klop. *Combinatory Reduction Systems*. Number 127 in Mathematical Centre Tracts. Mathematisch Centrum, Amsterdam, 1980. Ph.D. Thesis.

- [KT92] A. J. Kfoury and J. Tiuryn. Type reconstruction in finite-rank fragments of the second-order λ -calculus. *Inf. Comput.*, 98(2):228–257, June 1992.
- [KTU90] A. J. Kfoury, J. Tiuryn, and P. Urzyczyn. An analysis of ML typability. In *15th Colloq. Trees Algebra Program.*, vol. 431 of *LNCS*, pp. 206–220. Springer-Verlag, 1990.
- [KTU94] A. J. Kfoury, J. Tiuryn, and P. Urzyczyn. An analysis of ML typability. *J. ACM*, 41(2):368–398, Mar. 1994. A preliminary version is [KTU90].
- [KW94a] A. J. Kfoury and J. B. Wells. A direct algorithm for type inference in the rank-2 fragment of the second-order λ -calculus. In *Proc. 1994 ACM Conf. LISP Funct. Program.*, 1994.
- [KW94b] A. J. Kfoury and J. B. Wells. New notions of reduction and non-semantic proofs of β -strong normalization in typed λ -calculi. Tech. Rep. 94-014, Comput. Sci. Dept., Boston Univ., 1994.
- [Men91] N. P. Mendler. Inductive types and type constraints in the second-order lambda calculus. *Annals of Pure and Applied Logic*, 51:159–172, 1991.
- [Mog89] E. Moggi. Computational lambda-calculus and monads. In *Proc. 4th Ann. Symp. Logic Comput. Sci.*, pp. 14–23, Pacific Grove, CA, U.S.A., June 5–8 1989. IEEE Comput. Soc. Press.
- [Ned73] R. P. Nederpelt. *Strong Normalization for a Typed Lambda Calculus with Lambda Structured Types*. PhD thesis, Technische Hogeschool Eindhoven, 1973.
- [Reg94] L. Regnier. Une équivalence sur les lambda-termes. *Theoretical Comput. Sci.*, 126:281–292, 1994. In French.
- [Sce87] A. Scedrov. Normalization revisited. In J. W. Gray and A. Scedrov, eds., *Proc. AMS Research Conference*, pp. 357–369. American Mathematical Society, 1987.
- [SF92] A. Sabry and M. Felleisen. Reasoning about programs in continuation-passing style. In *Proc. 1992 ACM Conf. LISP Funct. Program.*, pp. 288–298, 1992.
- [SH80] J. P. Seldin and J. R. Hindley, eds. *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*. Academic Press, 1980.
- [vB92] S. J. van Bakel. Complete restrictions of the intersection type discipline. *Theoretical Comput. Sci.*, 102(1):135–163, 1992.
- [vdPS95] J. van de Pol and H. Schwichtenberg. Strict functionals for termination proofs. In *2nd Int'l Conf. Typed Lambda Calculi and Applications*. Springer-Verlag, Apr. 1995.
- [Vid89] D. Vidal. *Nouvelles Notions de Réduction en Lambda-Calcul*. Thèse de Doctorat, Université de Nancy 1, Feb. 1989.