An Induction Measure on λ-terms and Its Applications

by

Hongwei Xi
Department of Mathematical Sciences
Carnegie Mellon University
Pittsburgh, PA 15213

Research Report No. 96-192 2
July, 1996

An Induction Measure on λ-terms and Its Applications

by

Hongwei Xi
Department of Mathematical Sciences
Carnegie Mellon University
Pittsburgh, PA 15213

Research Report No. 96-192
July, 1996

An Induction Measure on λ-terms and Its Applications

Hongwei Xi
Department of Mathematical Sciences
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Email: hwxi@cs.cmu.edu

August 27, 1995

Abstract

A useful induction measure on λ-terms is presented here. Combining leftmost reduction with subterm reduction, we introduce a new notion called $\mathcal{H}$-reduction for untyped λ-calculus. Since a subterm reduction is only performed on a term when it is in an empty context, the $\mathcal{H}$-reduction is really a relation in a more rigorous sense. We then prove the equivalence between strong normalisability and $\mathcal{H}$-normalisability, which is essentially a bridge linking $\mathcal{H}$-reduction to various strong normalisation problems. Exploiting the new notion, we present some simplified proofs for several fundamental theorems such as finiteness of developments, the conservation theorem for λK-calculus, and the strong normalisation theorem for simply typed λ-calculus. Also a simplified proof of the characterisation theorem on perpetual redexes in [BK82] is included. Compared with other proofs in the literature, all presented proofs are quite concise and straightforward. In the case of the conservation therorem, the proof is also quite perspicacious. Finally, we give a brief comparison between $\mathcal{H}$-reduction and other methods such as perpetual strategies. We claim $\mathcal{H}$-reduction is a clean presentation of many similar ideas mentioned in the literature.

1. Introduction

In λ-calculus or some other rewriting systems, an induction measure on terms usually plays a pivotal role in the proofs of various theorems related to strong normalisation or termination. The current research starts with the following simple observation.

As an example, the conservation theorem in [CR36] states an interesting property for λI-calculus, i.e., a term is strongly normalisable if it has a normal form. A naïve extension of this theorem to λK-calculus, namely, the usual λ-calculus, would fail. For instance, $(\lambda x.\lambda y.y)(\omega\omega)$, where $\omega = \lambda x.xx$, is not strongly normalisable but has a normal form $\lambda y.y$. Nonetheless, there is a slight modification which makes the theorem valid in λ-calculus. Roughly speaking, when a β-redex $(\lambda x.u)v$ is contracted, we have to check that v has a normal form if x does not occur free in u. A similar idea can also be found in [Bar76], where a perpetual strategy is introduced to prove the conservation theorem for λK-calculus, and also in [BK82].

With this observation, we introduce a notion of hybrid reduction, or $\mathcal{H}$-reduction. A term t $\mathcal{H}$-reduces to t' if either t' is a proper subterm of t or t' is obtained from reducing the *leftmost* β-redex in t. Then we prove the equivalence between strong normalisability and $\mathcal{H}$-normalisability, which enables us to deal with various problems related to strong normalisation in an innovative fashion. First a new proof of the *finiteness of developments* theorem is presented, which is quite concise and straightforward when compared with other proofs in the literature, though I feel the one in [Hin78] is more perspicacious. We then demonstrate a short and sharp proof for the *conservation theorem* in λK calculus with the help of the *finiteness of developments* theorem. Exploiting the very same technique, we give a much simplified proof of the main result in [BK82], namely, the characterisation of perpetual redexes. Turning our attention to typed system, we present a syntactic proof of the strong normalisation theorem for simply typed λ-calculus.

Lastly, some relation between $\mathcal{H}$ and other methods is mentioned. We conclude that $\mathcal{H}$-reduction is an effective way to handle many problems involved with strong normalisation, and we expect more applications of this method.

The layout of the paper is as follows.

- The notions and basics are explained in Section 2.
- In Section 3, the equivalence between strong normalisability and $\mathcal{H}$-normalisability is established.
- In Section 4, a fundamental theorem in λ-calculus, finiteness of developments, is proven via the notion of $\mathcal{H}$-reduction.
- In Section 5, the conservation theorem for λK-calculus is presented.
- In Section 6, The behavior of β_K-redexes is investigated, and a related theorem is presented, which yields the characterisation of perpetual redexes given in [BK82].
- In Section 7, a syntactic proof of the strong normalisation theorem for pure simply typed λ-calculus is given.
- Lastly, some remarks on η-reduction are drawn, and some conclusions on $\mathcal{H}$-reduction are given.

2. Notions, Terminology and Basics

We give a brief explanation on the notions and terminology used in this paper. Most details, which could not be included here, can be found in [Bar84].

Definition 1 *(Pure λ-terms) The set Λ of λ-terms is defined inductively as follows.*

- *(variable) There are infinitely many variables $x, y, z, \ldots$ in Λ.*
- *(abstraction) If $t \in \Lambda$ then $(\lambda x.t) \in \Lambda$.*
- *(application) If $t_0, t_1 \in \Lambda$ then $(t_0 t_1) \in \Lambda$.*

$[u/x]v$ stands for substituting u for all free occurrences of x in v. α-conversion or renaming bound variables may have to be performed in order to avoid naming collisions. Also substitution properties such as Lemma 2.2.16 in [Bar84] will be assumed.

Definition 2 *(β-redex, β-reduction and β-normal form) A term of form $(\lambda x.u)v$ is called a β-redex, and $[v/x]u$ is called the contractum of the redex; $t \rightsquigarrow_\beta t'$ or $t \rightsquigarrow t'$ stands for a β-reduction step where t' is obtained from replacing some redex in t with its contractum; a β-normal form is a term in which there is no β-redex.*

"β-" is often omitted if this causes no confusion or ambiguity. In addition, similar notions such as other redexes or reductions will probably not be defined explicitly later if they are very analogous to the previous ones.

Usually there are many different redexes in a term t; a redex r_1 in t is left to another redex r_2 in t if the first symbol of r_1, namely the first λ, is left to that of r_2.

Given a kind of reduction R, $\rightsquigarrow_R$ stands for a single step of the reduction; $\rightsquigarrow_R^n$ stands for n steps of the reduction, where n could be 0; $\rightsquigarrow_R^*$ stands for some number of steps of the reduction.

Definition 3 *(Reduction Tree) Given a term t, a root node $n(t)$ is created to which t is attached; if t can be reduced to t', then a child node of $n(t)$ is created to which t' is attached; a tree, possibly infinite, is constructed in this way, and is called the reduction tree of t; each path starting from the root of the tree stands for a possible reduction sequence from t.*

For different reductions, there are different reduction trees accordingly. All reduction trees in this paper are finitely branched, namely, each node has at most finitely many children. By König Lemma, we know that a reduction tree must be a finite tree if all of its paths are of finite length.

Definition 4 *(Strong Normalisability) A term t is strongly normalisable if every β-reduction sequence starting from t is finite. In other words, t has a finite β-reduction tree.*

Let $\mathcal{S}(t)$ denote the height of the β-reduction tree of t. $\mathcal{S}(t) < \infty$ means t is strongly normalisable while $\mathcal{S}(t) = \infty$ conveys that t has an infinite β-reduction tree.

Definition 5 *(Subterm Reduction) A term is reduced to its subterms according to the following rules.*

$$(\lambda x.t_0) \rightsquigarrow_s t_0 \qquad t_1 t_2 \rightsquigarrow_s t_1 \qquad t_1 t_2 \rightsquigarrow_s t_2$$

Remark The terminology is abused a little here. The subterm reduction can only be performed on a term t when t is in the empty context. Besides, no term should be referred to as a redex with respect to subterm reduction.

Definition 6 *(Leftmost Reduction and Hybrid Reduction) A reduction $t \rightsquigarrow_l t^*$ is called leftmost reduction if the β-redex contracted in this step is the leftmost one among all redexes in t; a reduction is a hybrid reduction $\rightsquigarrow_{\mathcal{H}}$ if it is either a subterm reduction $\rightsquigarrow_s$ or a leftmost reduction $\rightsquigarrow_l$.*

Note that the leftmost reduction is different from the head reduction, which will be defined in Section 7. They coincide only if the reduced term is not in head normal form. "$\mathcal{H}$-" will be attached to names of notions associated with hybrid reduction. Now we are ready to establish a relation between hybrid reduction and strong normalisation.

Definition 7 *A term t is $\mathcal{H}$-normalisable if every hybrid reduction sequence starting from t is finite.*

Similarly, the $\mathcal{H}$-reduction tree $\mathcal{T}$ of a term t can be constructed according to $\mathcal{H}$-reduction. $\mathcal{T}$ is a finite tree if and only if every $\mathcal{H}$-reduction sequence starting from t is of finite length. This time we do not need König Lemma since each node in $\mathcal{T}$ has at most three children. Let $\mathcal{H}(t)$ be the height of $\mathcal{T}$ if $\mathcal{T}$ is finite. $\mathcal{H}(t) = \infty$ means that the $\mathcal{H}$-reduction tree of t is infinite. In addition, we define $\mathcal{H}_s(t) = max\{\mathcal{H}(t_s) \mid t \rightsquigarrow_s t_s\}$ and $\mathcal{H}_l(t) = \mathcal{H}(t^*)$ where $t \rightsquigarrow_l t^*$.

Proposition 8 *We have the following properties.*

1. $\mathcal{H}(x) = 0$ *for any variable* x,
2. $\mathcal{H}(\lambda x.t) = 1 + \mathcal{H}(t)$,
3. $\mathcal{H}(t) = 1 + max\{\mathcal{H}_s(t), \mathcal{H}_l(t)\}$ *for any application* t, *and*
4. *if* s *is a subterm of* t *then* $\mathcal{H}(s) \leq \mathcal{H}(t)$.

Notice $1 + \infty = \infty$ *is adopted.*

Proof All these can be proven by straightforward induction on $\mathcal{H}(t)$. ∎

In the following presentation, the main strategy to prove $\mathcal{H}(t) < \infty$ is to show $\mathcal{H}_s(t) < \infty$ and $\mathcal{H}_l(t) < \infty$.

3. Equivalence between Strong normalisability and $\mathcal{H}$-normalisability

First we prove the easier direction: strong normalisability implies $\mathcal{H}$-normalisability.

Lemma 9 *($\mathcal{S}$-$\mathcal{H}$) A term t is $\mathcal{H}$-normalisable if it is strongly normalisable.*

Proof Assume t to be strongly normalisable. We proceed by induction on $\mathcal{S}(t)$. The proof is based on the following observations.

- $\mathcal{S}(t') < \mathcal{S}(t)$ if $t \rightsquigarrow_l t'$, and
- $\mathcal{S}(t') \leq \mathcal{S}(t)$ if $t \rightsquigarrow_s t'$.

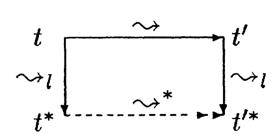
Since subterm reduction cannot go on forever, for every term t there is a number $n(t)$ such that $\mathcal{S}(t_1) < \mathcal{S}(t)$ whenever $t \rightsquigarrow^n_{\mathcal{H}} t_1$ for some $n \geq n(t)$. This implies $\mathcal{H}(t) < \infty$. ■

Before moving forward, let us introduce another useful notion, whose formal definition can be found in [Bar84]. Let $\mathcal{R}$ be a set of redexes in a term t, $r = (\lambda x.u)v \in \mathcal{R}$, and $t \rightsquigarrow t'$ after r is contracted. This reduction step affects redexes r' in $\mathcal{R}$ in the following way.

- r' is r. Then r' has no residual in t'.
- r' is in v. All copies of r' in $[v/x]u$ are called the residuals of r' in t';
- r' is in u. Then $[v/x]r'$ in $[v/x]u$ is the only residual of r' in t';
- r' contains r. Then the residual of r' is the term obtained from replacing r in r' with $[v/x]u$.
- Otherwise, r' is not affected, and is its own residual in t'.

The following lemma, virtually equivalent to Lemma 13.2.5(i) in [Bar84], is needed in the proof of the upcoming $\mathcal{H}$-$\mathcal{S}$ lemma. I present a proof here since this lemma plays a crucial role in this paper.

Lemma 10 *If $t \rightsquigarrow_l t^*$ and $t \rightsquigarrow t' \rightsquigarrow_l t'^*$, then $t^* \rightsquigarrow^* t'^*$.*



Proof Let $r_l = (\lambda x.u_l)v_l$ be the leftmost redex in t, and $r = (\lambda x.u)v$ be the redex contracted in step $t \rightsquigarrow t'$. We have the following cases.

- $(\lambda x.u_l)v_l$ is $(\lambda x.u)v$. Then $t^* \rightsquigarrow^* t'^*$ since $t^* = t' \rightsquigarrow_l t'^*$.

- $(\lambda x.u_l)v_l$ is left to $(\lambda x.u)v$. Now we have the following subcases.
  - $(\lambda x.u)v$ is in v_l. After $(\lambda x.u)v$ is contracted, v_l is reduced to some v'. Now $(\lambda x.u_l)v'$ is the leftmost redex in t', which gets contracted in $t' \rightsquigarrow_l t'^*$. Hence t'^* is obtained from replacing $(\lambda x.u_l)v_l$ in t with $[v'/x]u_l$. On the other hand, t^* is obtained from replacing $(\lambda x.u_l)v_l$ in t with $[v_l/x]u_l$. Reducing all residuals of r we have $t^* \rightsquigarrow^* t'^*$.
  - $(\lambda x.u)v$ is in u_l or $(\lambda x.u)v$ is separate from $(\lambda x.u_l)v_l$. In this case, t'^* can be obtained from contracting the only residual of r in t^*.

■

Lemma 11 *($\mathcal{H}$-$\mathcal{S}$) If t is $\mathcal{H}$-normalisable then t is strongly normalisable.*

Proof We define a measure $\mathcal{M}$ as follows.

$$\mathcal{M}(t) = \begin{cases} \mathcal{S}(t_0) + \mathcal{S}(t_1) & \text{if } t = t_0 t_1; \\ 0 & \text{otherwise.} \end{cases}$$

To make the induction work, we show $\mathcal{H}(t') \leq \mathcal{H}(t)$ for any $t \rightsquigarrow t'$ and $\mathcal{S}(t) < \infty$ simultaneously by induction on on the lexicographic order $\langle \mathcal{H}(t), \mathcal{M}(t) \rangle$.

- t is in normal form. Then $\mathcal{S}(t) = 0 < \infty$.
- $t = \lambda x.u$. This case is easily verified by induction hypothesis.
- $t = t_0 t_1$. Let $t \rightsquigarrow_l t^*$ and $t \rightsquigarrow t'$. By induction hypothesis, $\mathcal{S}(t_i) < \infty$ for $i = 0, 1$.
  - $t^* = t'$. Then $\mathcal{H}(t') = \mathcal{H}(t^*) < \mathcal{H}(t)$. By induction hypothesis, $\mathcal{S}(t') < \infty$.
  - $t' = t'_0 t'_1$ where $t_i \rightsquigarrow t'_i$ and $t_{1-i} = t'_{1-i}$ for some $i \in \{0, 1\}$. Hence $\mathcal{M}(t') < \mathcal{M}(t)$. Notice for $i = 0, 1$ $\mathcal{H}(t'_i) \leq \mathcal{H}(t_i)$ by induction hypothesis. By Lemma 10, there exists t'^* such that $t' \rightsquigarrow_l t'^*$ and $t^* \rightsquigarrow^* t'^*$. Since $\mathcal{H}(t^*) < \mathcal{H}(t)$, induction hypothesis yields $\mathcal{H}(t'^*) \leq \mathcal{H}(t^*)$. Therefore, we have obtained

    $$\mathcal{H}(t') = 1 + max\{\mathcal{H}(t'_0), \mathcal{H}(t'_1), \mathcal{H}(t'^*)\} \leq 1 + max\{\mathcal{H}(t_0), \mathcal{H}(t_1), \mathcal{H}(t^*)\} = \mathcal{H}(t),$$

    which, with $\mathcal{M}(t') < \mathcal{M}(t)$, yields $\mathcal{S}(t') < \infty$ by induction hypothesis. Therefore, $\mathcal{S}(t) < \infty$ since $\mathcal{S}(t') < \infty$ for all $t \rightsquigarrow t'$.

■

Using $\mathcal{H}$-reduction often yields clarity in a proof and, more importantly, brings a great deal of easiness in thinking. This claim becomes more and more clear when the following proofs unfold.

4. Finiteness of Developments

The main theorem in this section states that developments, a special kind of β-reduction sequences, are always finite. The theorem plays a pivotal role in some proofs of Church-Rosser theorem. A version of this theorem for λI-calculus was proved in [CR36]; for λK-calculus, it was proved by Schroer in [Sch65] and independently by Hyland in [Hyl73] and by Hindley in [Hin78]. Since this is essentially a strong normalisation for β_0-reduction defined below, we intend to give a proof based on hybrid reduction, which is close to the one given by Hindley but with a greater clarity.

Definition 12 *(λ_0-terms)*

- *(variable) There are infinitely many variables $x, y, z, \ldots$ in Λ_0.*
- *(abstraction) If $t \in \Lambda_0$ then $(\lambda x.t) \in \Lambda_0$.*
- *(application) If $t_0, t_1 \in \Lambda_0$ then $(t_0 t_1) \in \Lambda_0$.*
- *(β_0-redex) If $t_0, t_1 \in \Lambda_0$ then $(\lambda_0 x.t_0)t_1 \in \Lambda_0$.*

Intuitively, given a β-redex $r = (\lambda x.u)v$, we can mark r to obtain a β_0-redex $r_0 = (\lambda_0 x.u)v$; given a λ-term t and a set $\mathcal{R}$ of redexes in t, $t_{\mathcal{R}}$ is the λ_0-term obtained from marking all redexes in $\mathcal{R}$. We often use t to stand for $t_{\mathcal{R}}$ if this causes no confusion.

Definition 13 *(β_0-reduction) Given a β_0-redex $r_0 = (\lambda_0 x.u)v$, $[v/x]u$ is the contractum of r_0; $t \rightsquigarrow_0 t'$ stands for a single step of β_0-reduction in which t' is obtained from replacing a β_0-redex in t with its contractum.*

Note that λ_0-terms are closed under β_0-reductions.

Definition 14 *(Development) Given a λ-term t and a set $\mathcal{R}$ of redexes in t; let $t_{\mathcal{R}}$ be the λ_0-term obtained from marking all the redexes in $\mathcal{R}$ into β_0-redexes, then a β_0-reduction sequence starting from $t_{\mathcal{R}}$ is called a development of $(t, \mathcal{R})$; a complete development of $(t, \mathcal{R})$ is a β_0-reduction sequence from $t_{\mathcal{R}}$ to some β_0-normal form.*

Notice that $\lambda_0 x.xx$ is not a legal λ_0-term: the first λ cannot get marked since there is no redex involved. As an example, let $t = (\lambda x.xx)(\lambda x.xx)$. Marking the only redex in t, we obtain $t_0 = (\lambda_0 x.xx)(\lambda x.xx)$. Note

$$t_0 = (\lambda_0 x.xx)(\lambda x.xx) \rightsquigarrow_0 (\lambda x.xx)(\lambda x.xx) = t,$$

which has no β_0-redex.

A leftmost β_0-reduction $t \rightsquigarrow_{l_0} t'$ is the one in which the leftmost β_0-redex gets contracted; subterm reduction, denoted by $\rightsquigarrow_{s_0}$ here, needs a slight modification:

$$t = (\lambda_0 x.u_0)t_1 \rightsquigarrow_{s_0} \lambda x.u_0$$

since $\lambda_0 x.u_0$ is not a legal λ_0-term; the hybrid reduction is called $\mathcal{H}_0$-reduction in this section, and $\mathcal{H}_0(t)$, $\mathcal{H}_{0,s}(t)$ and $\mathcal{H}_{0,l}(t)$ are defined accordingly.

Observation Given $t = t_0 t_1$, where t is not a β_0-redex. It is clear that $\mathcal{H}_0(t_i) < \infty$ for $i = 0, 1$ implies $\mathcal{H}_0(t) < \infty$. This is simply due to the fact that there cannot be any interactions between t_0 and t_1.

Since Lemma 10 is still true for β_0-reduction, we have the following $\mathcal{H}_0$-$\mathcal{S}_0$ lemma.

Lemma 15 *($\mathcal{H}_0$-$\mathcal{S}_0$) A term t is strongly β_0-normalisable if it is $\mathcal{H}_0$-normalisable.*

Lemma 16 *Given u, v such that $\mathcal{H}_0(u) < \infty$ and $\mathcal{H}_0(v) < \infty$, then $\mathcal{H}_0([v/x]u) < \infty$.*

Proof The proof proceeds by induction on $\mathcal{H}_0(u)$.

- u is a variable. Then the case is trivial.
- $u = \lambda y.u_0$. By induction hypothesis, $\mathcal{H}_0([v/x]u_0) < \infty$. Hence

$$\mathcal{H}_0([v/x]u) = \mathcal{H}_0(\lambda y.[v/x]u_0) = 1 + \mathcal{H}_0([v/x]u_0) < \infty.$$

- $u = (\lambda_0 y.u_0)u_1$. Then $u \leadsto_{l_0} u^* = [u_1/y]u_0$, yielding $\mathcal{H}_0(u^*) < \mathcal{H}_0(u)$. By induction hypothesis, $\mathcal{H}_0([v/x]u^*) < \infty$. Note $[v/x]u = (\lambda_0 y.[v/x]u_0)([v/x]u_1)$, and we have $\mathcal{H}_{0,s}([v/x]u) < \infty$ by induction hypothesis. Notice $[v/x]u \leadsto_{l_0} [([v/x]u_1)/y]([v/x]u_0) = [v/x]u^*$ according to some property of substitution, yielding $\mathcal{H}_{0,l}([v/x]u) < \infty$. Hence

$$\mathcal{H}_0([v/x]u) = 1 + max\{\mathcal{H}_{0,l}([v/x]u), \mathcal{H}_{0,s}([v/x]u)\} < \infty.$$

- $u = u_0 u_1$. Notice $[v/x]u = ([v/x]u_0)([v/x]u_1)$. Induction hypothesis yields $\mathcal{H}_0([v/x]u_i) < \infty$ for $i = 0, 1$. By the above observation, $\mathcal{H}_0([v/x]u) < \infty$. ■

Theorem 17 *(Finiteness of Developments) All developments are finite.*

Proof It suffices to show that all λ_0-terms are strongly β_0-normalisable. By the $\mathcal{H}_0$-$\mathcal{S}_0$ lemma, this is equivalent to proving $\mathcal{H}_0(t) < \infty$ for every λ_0-term t. The proof proceeds by induction on the structure of t.

- t is a variable. The case is trivial.
- $t = \lambda x.u$. By the induction hypothesis, $\mathcal{H}_0(t) = 1 + \mathcal{H}_0(u) < \infty$.
- $t = (\lambda_0 x.u_0)t_1$, where u_0, t_1 are λ_0-terms. By induction hypothesis, $\mathcal{H}_0(u_0) < \infty$ and $\mathcal{H}_0(t_1) < \infty$. Lemma 16 yields $\mathcal{H}_{0,l}(t) = \mathcal{H}_0([t_1/x]u_0) < \infty$. Therefore, we have

$$\mathcal{H}_0(t) = 1 + max\{\mathcal{H}_{0,l}(t), 1 + \mathcal{H}_0(u_0), \mathcal{H}_0(t_1)\} < \infty.$$

- $t = t_0 t_1$. By induction hypothesis, $\mathcal{H}_0(t_i) < \infty$ for $i = 0, 1$. The above observation yields $\mathcal{H}_0(t) < \infty$. ■

The use of $\mathcal{H}$-reduction helps facilitate the proof significantly. Compared with other proofs in the literature, the conciseness of this proof can certainly be noticed. The theorem can be proven virtually in the same say if one conducts an induction on $\mathcal{S}(t)$. The difference is that one has to perform an induction on $\mathcal{S}(u_0)$ and $\mathcal{S}(t_1)$ in the case $t = (\lambda x.u_0)t_1$, which makes the proof less attractive. Nonetheless, even such a proof is fresh to the author. Later we will mention that η-reduction can be handled straightforwardly in the setting of $\mathcal{H}$-reduction, yielding that all λ_0-terms are strongly $\beta_0\eta$-normalisable.

Lastly, I would like to mention that for any $(t, \mathcal{R})$ all the complete developments of $(t, \mathcal{R})$ end with the same term. This result will be used in our following proofs, and its explanation can be readily found in [Bar84].

5. Conservation Theorems

One reason of combining leftmost reduction with subterm reduction is to exploit some useful properties of standard reduction sequence. The following proof of the conservation theorem is an example of such an exploitation.

Definition 18 *(Standard reduction sequence) Given a reduction sequence $t \rightsquigarrow^n t'$, and let r_i be the contracted redex in the ith reduction step for $i = 1, \ldots, n$; if r_j is not a residual of some redex left to r_i for all $1 \leq i < j \leq n$, then the given reduction sequence is called a standard reduction sequence; a development is standard if it is a standard reduction sequence.*

Remark Given t and a set $\mathcal{R}$ of redexes in t, we can always form a standard complete development of $(t, \mathcal{R})$ by contracting leftmost β_0-redexes first.

Definition 19 *(β_I-redex) $r = (\lambda x.u)v$ is a β_I-redex if variable x does occur free in u; $t \rightsquigarrow_I t'$ stands for a β_I-reduction in which the contracted redex is a β_I-redex; $t \rightsquigarrow_{l_I} t'$ stands for a leftmost reduction in which the contracted redex is a β_I-redex.*

Observation For any terms s and v, $\mathcal{H}(s) \leq \mathcal{H}([v/x]s)$. This is because for any $\mathcal{H}$-reduction sequence starting from s there is a corresponding one starting from $[v/x]s$: v can be treated as if it were a variable.

Lemma 20 *Let $\mathcal{R}$ be a set of β_I redex in t and $t \rightsquigarrow^* t_N$ is a standard complete development of $(t, \mathcal{R})$, then $\mathcal{H}(t_N) < \infty$ implies $\mathcal{H}(t) < \infty$.*

Proof Assume $t \rightsquigarrow^n t_N$. The proof proceeds by induction on the lexicographic order of $\langle \mathcal{H}(t_N), n \rangle$.

- $n = 0$. This is trivial since $t = t_N$.
- $n > 0$. Let r be the first contracted redex in $t \rightsquigarrow t' \rightsquigarrow^{n-1} t_N$. By induction hypothesis, $\mathcal{H}(t') < \infty$. Now we conduct a case analysis on the structure t.

- $t = \lambda x.u$. The case is easily verified by induction hypothesis.
- $t = t_0 t_1$. We first prove $\mathcal{H}_s(t) < \infty$.
  * r is t. Then $t = (\lambda x.u_0)t_1$ for some u_0 and $t' = [t_1/x]u_0$. By the observation, we know $\mathcal{H}(u_0) \leq \mathcal{H}(t') < \infty$, yielding $\mathcal{H}(t_0) = 1+\mathcal{H}(u_0) < \infty$. Since r is a β_I-redex, t_1 is a subterm of t', yielding $\mathcal{H}(t_1) \leq \mathcal{H}(t') < \infty$. Therefore, $\mathcal{H}_s(t) < \infty$.
  * r is not t. Since the development is standard and complete, t_N must be of form $t_N^0 t_N^1$, where $t_i \leadsto^* t_N^i$ are standard complete developments for $i = 0, 1$. By induction hypothesis, $\mathcal{H}(t_i) < \infty$ for $i = 0, 1$. Hence $\mathcal{H}_s(t) < \infty$

  Let r_l be the leftmost β-redex in t, and we show $\mathcal{H}_l(t) < \infty$.
  * r_l is r. Then $\mathcal{H}_l(t) = \mathcal{H}(t') < \infty$.
  * r_l is not r. By the definition of residuals, it can be readily verified that r_l has only one residual r_N in t_N, which is the leftmost β-redex in t_N. Let $t_N \leadsto_l t_N^*$, then $\mathcal{H}(t_N^*) < \mathcal{H}(t_N)$. Consider the standard complete development of $(t, \mathcal{R} \cup \{r_l\})$: $t \leadsto_l t^* \leadsto^* t_N^*$, where $t^* \leadsto^* t_N^*$ is a standard complete development of all the residuals of the redexes in $\mathcal{R}$. It is a routine verification that all the residuals are β_I-redexes since all the redexes in $\mathcal{R}$ are β_I-redexes. By induction hypothesis, $\mathcal{H}_l(t) = \mathcal{H}(t^*) < \infty$.

  Therefore, $\mathcal{H}(t) = 1 + max\{\mathcal{H}_s(t), \mathcal{H}_l(t)\} < \infty$.

■

Theorem 21 *(Conservation Theorem) If $t \leadsto_I t'$ then strong normalisability of t' implies strong normalisability of t.*

Proof This is just a corollary of Lemma 20, where $\mathcal{R}$ is a singleton set. ■

Now it becomes clear that an induction on $\mathcal{H}$ makes it sufficient to study only the residuals of inner redexes generated by $\mathcal{H}$-reductions. The essence of the above proof can be summarised in one sentence, namely, the residuals of a β_I-redex generated by $\mathcal{H}$-reductions are still β_I-redexes. The proof of Lemma 20 would fail if $\mathcal{S}$ were used instead of $\mathcal{H}$ since the residuals of β_I-redexes are not necessarily β_I-redexes after the contraction of an arbitrary β-redex. This is basically the reason why the proof of the conservation theorem for the λI-calculus in [Bar84] can not be simply extended to a proof of the conservation theorem for the λK-calculus.

With this observation, we are ready to explore further on this subject.

6. Perpetual Redexes

It is certainly interesting to know what happens if we reduce some β-redexes which are not β_I-redexes.

Definition 22 *(β_K-redex) $r = (\lambda x.u)v$ is a β_K-redex if x does not occur free in u; $t \leadsto_K t'$ is a β_K-reduction in which the contracted redex is a β_K-redex; $t \leadsto_{l_K} t'$ stands for a leftmost reduction in which the contracted redex is a β_K-redex.*

The following lemma can be found in [Bar84] as Lemma 13.4.5.

Lemma 23 *(β_K-conservation) Given $t \rightsquigarrow_{l_K} t'$ in which a β_K-redex $r = (\lambda x.u)v$ gets contracted. If both t' and v are strongly normalisable, then t is strongly normalisable.*

Proof A simple induction on $\mathcal{H}(t)$ yields the result. ∎

Notice that the contracted β_K-redex must be a leftmost redex in order to apply the above lemma; otherwise, counterexamples can be found easily. For those who are interested in β_K-conservation, [BK82] gives a much more detailed analysis on this subject.

Definition 24 *A β_K-redex $r = (\lambda x.u)v$ is a β_P-redex if $\mathcal{S}(u) < \infty$ implies $\mathcal{S}(v) < \infty$; a β_P-redex r in t is t-special if $[\vec{t}/\vec{x}]r$ are β_P-redexes for any list of strongly normalisable terms $\vec{t} = t_1, \ldots, t_n$, where x is a list of all the variables which are free in r but bound in t; a redex r is special if it is $(\lambda y_1 \ldots \lambda y_m.r)$-special, where y_i for $1 \leq i \leq m$ are all the free variables in r.*

Evidently, we can use $\mathcal{H}$ instead of $\mathcal{S}$ in the above definition without changing its meaning. Also we assume bound variables are chosen distinctly from free variables to make the above definition hygienic.

Proposition 25 *Given t and its leftmost redex $r_l = (\lambda x.u)v$, where $\mathcal{H}(v) < \infty$ and all the free variables in v are free in t; if $r \neq r_l$ is a t-special redex in t and $t \rightsquigarrow_l t^*$, then all the residuals of r in t^* are t^*-special.*

Proof Let r_s be a residual of r in t^*.

- r_s is of form r. This case is trivial.
- r_s is of form $[v/x]r$. r_s is a β_P-redex since $\mathcal{H}(v) < \infty$. Let $\vec{x} = x_1, \ldots, x_n$ be all the free variables which are free in r_s but bound in t^*, then no free variables in v are in $\vec{x}$. With some property of substitution, it can be easily verified that r_s is a t^*-special redex.

∎

Lemma 26 *Given t and a set $\mathcal{R}$ of t-special redexes in t; If $t \rightsquigarrow^* t_N$ is a standard complete development of $(t, \mathcal{R})$, then $\mathcal{H}(t_N) < \infty$ implies $\mathcal{H}(t) < \infty$.*

Proof Assume $\mathcal{H}(t_N) < \infty$ and $t \rightsquigarrow^n t_N$. We proceed by induction on $\langle \mathcal{H}(t_N), n \rangle$, lexicographically ordered. The following proof is quite similar to the proof of Lemma 20.

- $n = 0$. This is trivial since $t = t_N$.
- $n > 0$. Let us conduct a case analysis on the structure of t.
  - $t = \lambda x.u$. This case is easily verified by induction hypothesis.

- $t = t_0 t_1$. Let $t \rightsquigarrow t' \rightsquigarrow^* t_N$, and r is the t-special redex contracted in the first step. We first establish $\mathcal{H}_s(t) < \infty$.
  * r is t. Then $t = (\lambda x.u_0)t_1$ and $t' = u_0$. By induction hypothesis, $\mathcal{H}(u_0) = \mathcal{H}(t') < \infty$. Since r is t-special, $\mathcal{H}(u_0) < \infty$ implies $\mathcal{H}(t_1) < \infty$. Thus, $\mathcal{H}_s(t) = max\{1 + \mathcal{H}(u_0), \mathcal{H}(t_1)\} < \infty$.
  * r is not t. Then t_N must be of form $t_N^0 t_N^1$, where $t_i \rightsquigarrow^* t_N^i$ are standard complete developments for $i = 0, 1$. By induction hypothesis, $\mathcal{H}(t_i) < \infty$ for $i = 0, 1$.

  Thus, $\mathcal{H}_s(t) = max\{\mathcal{H}(t_0), \mathcal{H}(t_1)\} < \infty$. It is easy to see that t must be of one of the following forms.
  * $t = au_1 \dots u_n$ for some atom a. Since $\mathcal{H}(u_i) \leq \mathcal{H}_s(t) < \infty$, it is obvious that $\mathcal{H}(t) < \infty$.
  * $t = r_l u_1 \dots u_n$ for some leftmost β-redex $r_l = (\lambda x.u)v$. Let r_N be the only residual of r_l in t_N, then r_N is the leftmost β-redex in t_N. Let $t_N \rightsquigarrow_l t_N^*$, and we have $\mathcal{H}(t_N^*) < \mathcal{H}(t_N)$. Consider the standard complete development of $(t, \mathcal{R} \cup \{r_l\})$: $t \rightsquigarrow_l t^* \rightsquigarrow^* t_N^*$, where $t^* \rightsquigarrow^* t_N^*$ is the standard complete development of the residuals of all the redexes in $\mathcal{R}$. Note that all the free variables in v are free in t. Also v is a proper subterm of t, yielding $\mathcal{H}(v) \leq \mathcal{H}_s(t) < \infty$. By Proposition 25, all residuals of the redexes in $\mathcal{R}$ are t^*-special. Hence $\mathcal{H}_l(t) = \mathcal{H}(t^*) < \infty$ by induction hypothesis, yielding $\mathcal{H}(t) = 1 + max\{\mathcal{H}_s(t), \mathcal{H}_l(t)\} < \infty$.

■

To make sure that contracting a β_K-redex r in t does not change the strong normalisability of t, all we really need is that all the residuals of r are β_P-redexes in all the $\mathcal{H}$-reduction sequences from t. Since this is difficult to verify, we require that r be t-special. The following corollary is a slight variation of Corollary 26 in [BK82]

Corollary 27 *Given a term $t = t[r]$ where $r = (\lambda x.u)v$ is a t-special redex and $t[\ ]$ is a context; if $\mathcal{H}(t[u]) < \infty$ then $\mathcal{H}(t[r]) < \infty$.*

Proof The is a special case of Lemma 26, where $\mathcal{R}$ is a singleton set. ■

Definition 28 *A redex r with contractum c is perpetual if $\mathcal{S}(t[c]) < \infty$ implies $\mathcal{S}(t[r]) < \infty$ for any context $t[\ ]$.*

Theorem 29 *A redex $r = (\lambda x.u)v$ is perpetual if and only if r is a β_I-redex or r is a special redex.*

Proof If r is a redex in t and r is special, then r is t-special. Applying Theorem 21 and Corollary 27, we can see that the only case left is to verify that a perpetual β_K-redex $r = (\lambda x.u)v$ is special. Suppose there exist $\vec{t} = t_1, \dots, t_n$ such that $\mathcal{S}(t_i) < \infty$ for $i = 1, \dots, n$, $\mathcal{S}([\vec{t}/\vec{x}]u) < \infty$ and $\mathcal{S}([\vec{t}/\vec{x}]v) = \infty$, where $\vec{x} = x_1 \dots, x_n$ is a list of all the free variables in r. Let $t[\ ] = (\lambda x_1 \dots \lambda x_n.[\ ])t_1 \dots t_n$, and we have $\mathcal{S}(t[u]) < \infty$ while $\mathcal{S}(t[r]) = \infty$. This contradicts that r is a perpetual redex. Hence the therorem has been justified. ■

Compared with the proof in [BK82], which uses a perpetual strategy, this proof simply shows what happens to the residuals of t-special redexes in a $\mathcal{H}$-reduction. If one notices that a term without head normal form implies that any substitution instances of the term have no head normal forms, this method can be readily adapted to give a syntactic proof of Berry's sequentiality theorem, where the standardisation theorem is needed but $\mathcal{H}$ plays no role. The main difference between induction on $\mathcal{H}$ and a perpetual strategy lies in that the former brings out inner redexes using $\mathcal{H}$-reductions while the latter enters a term to find them.

7. Simply Typed λ-calculus

In this section, we intend to give a syntactic proof of the strong normalisation theorem for pure simply typed λ-calculus, which exhibits an elegant solution to a crucial lemma in the proof of the strong normalisation theorem for the labelled λ-calculus in [Daa80].

Definition 30 *(Simple Types and Simply Typed Terms) Types are formulated in the following way.*

- *Atomic types are types.*
- *If U and V are types then $U \to V$ is a type.*

Simply typed terms are defined inductively as follows.

- *(variable) For each type U, there are infinitely many variables $x^U, y^U, \ldots$ of that type.*
- *(abstraction) If v is of type V then $\lambda x^U.v$ is of type $U \to V$.*
- *(application) If u is of type $U \to V$ and v is of type U, then uv is of type V.*

We often omit the type superscript of a variable if this causes no confusion or ambiguity. Also β-reduction for simply typed λ-calculus is essentially the same as β-reduction for untyped λ-calculus. We intend to prove the next theorem in this section.

Theorem 31 *(Strong Normalisation for Simply Typed λ-Calculus) Every term in simply typed lambda calculus is strongly normalisable.*

The method used below originates from Turing's work according to [Gan80]. A detailed account of it can also be found in [And71]. Though the method produces a straightforward proof of a *weak* normalisation theorem for simply typed λ-calculus, we have not found a proof of the *strong* normalisation theorem for simply typed λ-calculus given in this fashion in the literature. The closest ones are found in [deGr93] and [KW94], where *controlling erasure* technique is used.

Definition 32 *(Head redex and Head Reduction) Given $t = \lambda x_1 \ldots \lambda x_m.ru_1 \ldots u_n$ where r is a redex. We call r the head redex in t; $t \leadsto_h t'$ stands for a head reduction step in which r gets contracted; a head normal form is a term without head redex in it.*

Definition 33 *The complexity $|T|$ of a type T is defined as follows.*

$$|T| = \begin{cases} 0 & \text{if } T \text{ is atomic;} \\ max\{1 + |T_0|, |T_1|\} & \text{if } T = T_0 \to T_1. \end{cases}$$

The rank $|t|$ of an application $t = t_0 t_1$ is defined as $|T_1|$ where T_1 is the type of t_1.

Lemma 34 *Given a simply typed term $t = t_0 t_1$. If $\mathcal{S}(t_i) < \infty$ for $i = 0, 1$, then $\mathcal{S}(t) < \infty$.*

Proof We proceed to prove $\mathcal{H}(t) < \infty$ by induction on the lexicographic order $\langle |t|, \mathcal{H}(t_0), \mathcal{H}(t_1) \rangle$. The $\mathcal{S}$-$\mathcal{H}$ lemma implies $\mathcal{H}(t_i) < \infty$ for $i = 0, 1$ since $\mathcal{S}(t_i) < \infty$ for $i = 0, 1$. If t is in normal form, then $\mathcal{H}(t) < \infty$. Let us assume $t \leadsto_l t^*$, and argue that $\mathcal{H}(t^*) < \infty$.

- $t^* = t_0^* t_1^*$ where $t_i \leadsto_l t_i^*$ and $t_{1-i} = t_{1-i}^*$ for some $i \in \{0, 1\}$. Hence $\mathcal{H}(t_i^*) < \mathcal{H}(t_i)$ and $\mathcal{H}(t_{1-i}^*) = \mathcal{H}(t_{1-i})$. By induction hypothesis, $\mathcal{H}(t^*) < \infty$.
- $t_0 = \lambda x.u$. Then $t^* = [t_1/x]u$. We have the following subcases.
  - $u = \lambda y.u_0$. Since $\mathcal{H}(\lambda x.u_0) < \mathcal{H}(t_0)$, by induction hypothesis, $\mathcal{H}((\lambda x.u_0)t_1) < \infty$. Thus, $\mathcal{H}(t^*) = \mathcal{H}(\lambda y.[t_1/x]u_0) = 1 + \mathcal{H}([t_1/x]u_0) < \mathcal{H}((\lambda x.u_0)t_1) < \infty$.
  - $u = v_0 v_1$. $\mathcal{H}((\lambda x.v_i)t_1) < \infty$ for $i = 0, 1$ by induction hypothesis. This yields $\mathcal{H}([t_1/x]v_i) < \infty$ for $i = 0, 1$. Hence $\mathcal{H}_s(t^*) < \infty$. Now we show $\mathcal{H}(t^*) < \infty$.
    * $u \leadsto_h u^*$. Then $t^* \leadsto_l [t_1/x]u^*$. By induction hypothesis, $\mathcal{H}((\lambda x.u^*)t_1) < \infty$ since $\mathcal{H}(\lambda x.u^*) < \mathcal{H}(t_0)$. Thus, $\mathcal{H}_l(t^*) = \mathcal{H}([t_1/x]u^*) < \infty$. Note $\mathcal{H}(t^*) = 1 + max\{\mathcal{H}_s(t^*), \mathcal{H}_l(t^*)\} < \infty$.
    * u is of form $au_1 \ldots u_n$, where a is a variable. Let $u_i' = [t_1/x]u_i$ for $i = 1, \ldots, n$. By induction hypothesis, for $i = 1, \ldots, n$, $\mathcal{H}(u') < \mathcal{H}((\lambda x.u_i)t_1) < \infty$ since $\mathcal{H}(\lambda x.u_i) < \mathcal{H}(t_0)$.
      - a is not x. Then $t^* = au_1' \ldots u_n'$. It is easy to verify $\mathcal{H}(t^*) < \infty$ since $\mathcal{H}(u_i') \leq \mathcal{H}_s(t^*) < \infty$ for $i = 1, \ldots, n$.
      - a is x. Then $t^* = t_1 u_1' \ldots u_n'$. If $n = 0$ then $\mathcal{H}(t^*) = \mathcal{H}(t_1) < \infty$. Otherwise, it is easy to verify $|t^*| < |t|$. By induction hypothesis, $\mathcal{H}_l(t^*) < \infty$. Thus, $\mathcal{H}(t^*) = 1 + max\{\mathcal{H}_s(t^*), \mathcal{H}_l(t^*)\} < \infty$.

Therefore, $\mathcal{H}(t) = 1 + max\{\mathcal{H}(t_0), \mathcal{H}(t_1), \mathcal{H}(t^*)\} < \infty$. This yields $\mathcal{S}(t) < \infty$ by the $\mathcal{H}$-$\mathcal{S}$ lemma. ∎

It is feasible to give a proof of Lemma 34 without using $\mathcal{H}$, but the complexity of such a proof increases significantly. For instance, if we would like to prove $\mathcal{S}(t^*) < \infty$ directly, then we have to show $\mathcal{S}(t^{*\prime}) < \infty$ for all $t^* \leadsto t^{*\prime}$. The difficulty arises if the contracted redex in $t^* \leadsto t^{*\prime}$ is not a residual of some redex in t. A solution to this problem is given in [Daa80], which can also be found as Exercise 15.4.8 in [Bar84].

Proof (of theorem 31) We prove every term t is strongly normalisable by induction on the structure of t.

- t is a variable. Then t is strongly normalisable.
- $t = \lambda x.u$. t is strongly normalisable since u is by induction hypothesis.
- $t = t_0 t_1$. By induction hypothesis, t_i are strongly normalisable for $i = 0, 1$. Applying lemma 34, we have $\mathcal{S}(t) < \infty$, i.e., t is strongly normalisable.

■

For those who are familiar with the labelled λ-calculi introduced by Hyland, Wadsworth and Lévy, it is obvious that this proof also works in that setting with a slight twist on the definition of $|t|$. Comparing this proof with the proof of the strong normalisation theorem for labelled λ-calculus in [Daa80], we can notice that the underline strategies resemble each other. Our proof is very short since the $\mathcal{H}$-$\mathcal{S}$ has captured the essential idea in [Daa80]. The major drawback with the proof is that it can hardly be extended to other stronger systems such as system **F**.

8. η-reduction

Let $\lambda x.tx \rightsquigarrow_\eta t$ be an η-reduction where x does not occur free in t. Since Lemma 10 is obviously still true in the presence of η-reduction we conclude that $\mathcal{H}$-$\mathcal{S}$ lemma still holds even if η-reduction is taken into consideration. This fact can be used to judge that all results, such as finiteness of development and the conservation theorem, also hold if $\beta\eta$-reduction is used. The only reason that we exclude η-reduction in our proofs is to enhance the comprehensibility of the proofs. If a term $s = \lambda x.tx$ itself is an η-redex, then the contractum t of s can be obtained from two steps of subterm reduction. In other words, the case of η-reduction can simply be handled by induction hypothesis in the structural induction proofs, yielding virtually unchanged new proofs.

9. Related Work

$\mathcal{H}$-reduction bears a great resemblance to the reduction strategy used in [Gog94]. The main difference is that we start our work in the untyped λ-calculus while Goguen works in a typed setting. It is also easy to reveal by a direct comparison that some intimate relation exists between $\mathcal{H}$-reduction and the perpetual strategies in [Bar76] and [BK82]. A related idea of transforming strong normalisation into weak normalisation can also be found in [Ned73], [Klo80], [deGr93] and [KW94].

$\mathcal{H}$-reduction brings out inner redexes or their residuals by leftmost and subterm reductions. It is usually easier to prove $\mathcal{H}(t) < \infty$ than $\mathcal{S}(t) < \infty$ for a given λ-term.

Perpetual strategies spot the crucial places where reductions may change the strong normalisability of a term. They are often intuitive but can involve too many syntactic details.

Controlling erasure reduces $(\lambda x.\lambda y.u)v$ to $\lambda y.(\lambda x.u)v$ so that one can avoid contracting β_K-redexes while keep reducing β_I-redexes. If this method works for a system, one can usually give a direct proof of the strong normalisation theorem for that system. The proof of Theorem 31 is such an example.

We have seen that all the proofs presented are quite short and straightforward. $\mathcal{H}$-reduction is a flexible and helpful tool handling strong normalisations, especially, when it is combined with other ideas such as standard reduction sequences. Besides, the $\mathcal{H}$-$\mathcal{S}$ lemma is also a clean presentation of many similar ideas such as perpetual reduction strategies mentioned in the literature: to make sure if contracting the leftmost redex changes the strong normalisability of a term.

10. Conclusion

We have seen, through various examples, that the notion of $\mathcal{H}$-reduction establishes a useful induction measure in the proofs of many theorems related to strong normalisations. The new proof for the *finiteness of developments* theorem is quite concise and straightforward, compared with others in the literature. The new proof of the conservation theorem for λK-calculus really brings out the essence of the theorem, which enables us to present a much simplified proofs for the characterisation theorem on perpetual redexes given in [BK]. To demonstrate the versatility of the method, we also present a proof of the strong normalisation theorem for the simply typed λ-calculus, which can be readily transformed into a proof of the strong normalisation theorem for labelled λ-calculi. Above all, I feel that $\mathcal{H}$-reduction eases thinking to a great extent when one deals with problems related to strong normalisations. It summarises a key idea used in many related proofs in the literature. Instead of handling inner redexes directly, $\mathcal{H}$-reduction allows us to wait until they become leftmost redexes in a reduction sequence. I have also tried this method on various semantic proofs of strong normalisation theorems for various typed λ-calculi, but the result turns out to be much less satisfactory since the simplification is very minor if there is any. This should not be surprising since the semantic proofs often treat leftmost redexes and inner redexes equally with no distinction. Lastly, We expect more applications of $\mathcal{H}$-reduction coming out.

11. Acknowledgement

I thank Peter Andrews, Frank Pfenning and Richard Statman for helpful talks and suggestions on the presentation of the paper.

References

[And71] P.B. Andrews (1971), Resolution in type theory, *J. Symbolic Logic 36*, pp. 414-432.

[Bar76] H.P. Barendregt et al. (1976), Some notes on lambda reduction, *Preprint No. 22, University of Utrecht, Department of mathematics*, pp. 13-53.

[Bar84] H.P. Barendregt (1984), The Lambda Calculus: Its Syntax and Semantics, *North-Holland publishing company, Amsterdam.*

[BK82] J.A. Bergstra and J.W. Klop (1982), Strong normalisation and perpetual reductions in the lambda calculus, *J. Inform. Process. Cybernet. 18 (718)*, pp. 403-417.

[CR36] A. Church and J.B. Rosser (1936), Some properties of conversions, *Trans. Amer. Math. Soc. 39*, pp. 472-482.

[Daa80] D.T. van Daalen, The language theory of Automath, *dissertation, Technological University Eindhoven.*

[deGr93] P. de Groote (1993), The conservation theorem revisited, *Int'l conf. Typed lambda calculi and applications*, vol. 664 of LNCS, pp. 163-178.

[Gan80] R.O. Gandy (1980), An early proof of normalisation by A.M. Turing, *To: H.B. Curry: Essays on combinatory logic, lambda calculus and formalism*, Academic press, pp. 453-456.

[Gan80a] R.O. Gandy (1980), Proofs of Strong Normalisation, *To: H.B. Curry: Essays on Combinatory logic, lambda calculus and formalism*, Academic press, pp. 457-478.

[Gir72] J.-Y. Girard (1972), Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur, *Thèse de doctorat d'etat, Université Paris VII.*

[Gir89] J.-Y. Girard et al. (1989), Proofs and types, *Cambridge Press*, 176 pp.

[Göd58] K. Gödel (1958), Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes, *Dialectica 12*, pp. 280-287.

[Gog94] Healfdene Goguen (1994), A typed operational semantics for type theory, *Ph.D. thesis, Dept. of Comp. Sci., The Univ. of Edinburgh.*

[Hin78] J.R. Hindley (1978), Reductions of residuals are finite, *Trans. Amer. Math. Soc. 240*, pp. 345-361.

[Hyl73] J.M.E. Hyland (1973), A simple proof of the Church-Rosser theorem, *Typescript, Oxford University*, 7 pp.

[Hyl76] J.M.E. Hyland (1976), A syntactic characterisation of the equality in some models of the λ-calculus, *J. London Math. Soc. (2) 12*, pp. 361-370.

[Klo80] J.W. Klop (1980), Combinatory reduction systems, *Ph.D. thesis, CWI, Amsterdam, Mathematical center tracts, No. 127.*

[KW94] A.J. Kfoury and J.B. Wells (1994), New notions of reduction and non-semantic proofs of β-strong normalisation in typed λ-calculi, *Tech. Rep. 94-104, Computer Science Department, Boston University.*

[Lév75] J.-J. Lévy, An algebraic interpretation of λ-β-K-calculus and a labelled λ-calculus, *Proceedings of the Symposium held in Rome, Lecture Notes in Computer Science 37, edited by C. Böhm, Springer-Verlag, Berlin,* pp. 147-165.

[Ned73] R.P. Nederpelt (1973), Strong normalisation in a typed lambda calculus with lambda structured types, *Ph.D. thesis, Technische Hogeschool Eindhoven.*

[Pra71] D. Prawitz (1971), Ideas and results of proof theory, Proceedings of the 2nd scandinavian logic symposium, *editor J.E. Fenstad, North-Holland Publishing Company, Amsterdam.*

[Sch65] D.E. Schroer (1965), The Church-Rosser Theorem, *Dissertation, Cornell university, Ithaca, New York, USA*.

[Tai67] W. Tait (1967), Intensional Interpretations of funtionals of finite type I, *J. symbolic logic 32*, pp. 198-212.

[Wad76] C.P. Wadsworth (1976), The relation between computational and denotational properties for Scott's D_∞-models of λ-calculus.

[Vri85] R. de Vrijer (1985), A direct proof of the finite developments theorem, *Journal of Symbolic Logic*, 50:339-343.