Upper Bounds for Standardisations and an Application

by

Hongwei Xi

Department of Mathematics
Carnegie Mellon University
Pittsburgh, PA 15213

Research Report No. 96-189 2
May, 1996

Upper Bounds for Standardisations and an Application

Hongwei Xi

Mathematics Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Telephone: +1 412 268-1439 Fax: +1 412 268-6380

Email: hwxi@cs.cmu.edu

Upper Bounds for Standardisations and an Application

Hongwei Xi

Mathematics Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Telephone: +1 412 268-1439 Fax: +1 412 268-6380

Email: hwxi@cs.cmu.edu

Abstract

We first present a new proof for the standardisation theorem, a fundamental theorem in λ-calculus. Since our proof is largely built upon structural induction on lambda terms, we can extract some bounds for the number of reduction steps in the standard reduction sequences obtained from transforming any given reduction sequences. This result sharpens the standardisation theorem and establishes a link between lazy and eager evaluation orders in the context of computational complexity. As an application, we establish a super exponential bound for the number of reduction steps in reduction sequences from any given simply typed λ-terms.

1. Introduction

The standardisation theorem of Curry and Feys [CF58] is a very useful result, stating that if u β-reduces to v for λ-terms u and v, then there is a *standard* reduction from u to v. Using this theorem, we can readily prove the normalisation theorem, i.e., a λ-term has a normal form if and only if the leftmost reduction sequence from the term is finite. The importance of lazy evaluation in functional programming language largely comes from the normalisation theorem. Moreover, the standardisation theorem can be viewed as a syntactic version of sequentiality theorem in [Ber78]. For instance, it can be readily argued that *parallel or* is inexpressible in λ-calculus by using the standardisation theorem. In fact, a syntactic proof of the sequentiality theorem can be given with the help of the standardisation theorem.

There have been many proofs of the standardisation theorem in the literature such as the ones in [Mit79], [Klo80], and [Bar84]. In the present proof we intend to find a bound for standardisations, namely, to measure the number of steps in the standard reduction sequence obtained from a given reduction sequence. This method presents a concise and more accurate formulation of the standardisation theorem. As an application, we establish a super exponential bound for the number of reduction steps in reduction sequences from any given simply typed λ-terms. This not only strengthens the strong normalisation theorem in the simply typed λ-calculus, but also yields more understanding on $\mu(t)$, the number of steps in a longest reduction sequence from a simply typed λ-term t. Since $\mu(t)$ can often be used as an induction order, its structure plays a key role in understanding related inductive proofs.

The structure of the paper is as follows.

- The notions and basics are explained in Section 2.
- In Section 3, our proof of the standardisation theorem is presented.
- Some upper bounds for standardisations are extracted in Section 4.
- In Section 5, we establish a bound for the number of reduction steps in reduction sequences from any given simply typed λ-terms.
- Finally, some related work is mentioned and a few remarks are drawn in Section 6.

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 *(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; variables are the only subterms of themselves.*
- *(abstraction) If $t \in \Lambda$ then $(\lambda x.t) \in \Lambda$; u is a subterm of $(\lambda x.t)$ if u is $(\lambda x.t)$ or a subterm of t.*
- *(application) If $t_0, t_1 \in \Lambda$ then $(t_0 t_1) \in \Lambda$; u is a subterm of $(t_0 t_1)$ if u is $t_0 t_1$ or a subterm of t_i for some $i \in \{0, 1\}$.*

The set $\mathrm{FV}(t)$ of free variables in t is defined as follows.

$$\mathrm{FV}(t) = \begin{cases} \{x\} & \textit{if } t = x \textit{ for some variable } x; \\ \mathrm{FV}(t_0) - \{x\} & \textit{if } t = \lambda x.t_0; \\ \mathrm{FV}(t_0) \cup \mathrm{FV}(t_1) & \textit{if } t = t_0 t_1. \end{cases}$$

The set Λ_I of λI-terms is a subset of Λ such that, for every term $t \in \Lambda_I$, if $(\lambda x.t_0)$ is a subterm of t then $x \in \mathrm{FV}(t_0)$.

$[v/x]u$ stands for substituting v for all free occurrences of x in u. α-conversion or renaming bounded variables may have to be performed in order to avoid name collisions. Rigorous definitions are omitted here.

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; $t \rightsquigarrow t'$ 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. Let $\rightsquigarrow^n$ stand for n steps of β-reduction, and $\rightsquigarrow^*$ stand for some steps of reduction, which could be 0. 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 is left to that of r_2.

Definition 3 *(Multiplicity) Given a redex $r = (\lambda x.u)v$; the multiplicity $m(r)$ of r is the number of occurrences of the variable x in u.*

Definition 4 *(Reduction sequence) Given a β-redex r in t; $t \overset{r}{\rightsquigarrow} u$ stands for the reduction step in which redex r gets contracted; $r_1 + \cdots + r_n$ stands for a reduction sequence of the following form.*

$$t_0 \overset{r_1}{\rightsquigarrow} t_1 \overset{r_2}{\rightsquigarrow} \cdots \overset{r_n}{\rightsquigarrow} t_n$$

Conventions $\sigma, \tau, \ldots$ range over reduction sequences; $\sigma : t \leadsto^* t'$ or $t \overset{\sigma}{\leadsto}{}^* t'$ stands for a reduction sequence from t to t'; $|\sigma|$ is the length of σ, namely, the number of reduction steps in σ, which might be 0.

Definition 5 *(Concatenation) Given $\sigma : t_0 \leadsto^* t_1$ and $\tau : t_1 \leadsto^* t_2$; $\sigma + \tau$ stands for the concatenation of σ and τ, namely, $\sigma + \tau : t_0 \overset{\sigma}{\leadsto}{}^* t_1 \overset{\tau}{\leadsto}{}^* t_2$.*

Conventions Let $\sigma : u \leadsto^* v$ and $C[\,]$ be a context, then σ can also be regarded as the reduction sequence which reduces $C[u]$ to $C[v]$ in the obvious way. In other words, we may use σ to stand for $C[\sigma]$.

Before moving forward, let us introduce the concept of *residuals* of redexes. The rigorous definition of this notion can be found in [Hue94]. Let $\mathcal{R}$ be a set of redexes in a term t, $r = (\lambda x.u)v$ in $\mathcal{R}$, and $t \overset{r}{\leadsto} t'$. 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 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' (rename bounded variables in u suitably if necessary);
- 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'.

Definition 6 *(Developments) Given a λ-term t and a set $\mathcal{R}$ of redexes in t; if $\sigma : t \leadsto^* u$ contracts only redexes in $\mathcal{R}$ or their residuals, then σ is a development.*

3. The Proof of Standardisation Theorem

Standardisation theorem of Curry and Feys [CF58] states that any reduction sequence can be standardised in the following sense.

Definition 7 *(Standard Reduction Sequence) Given a reduction sequence*

$$\sigma : t = t_0 \overset{r_0}{\leadsto} t_1 \overset{r_1}{\leadsto} t_2 \overset{r_2}{\leadsto} \cdots$$

σ is standard if for all $0 \leq i < j$, r_j is not a residual of some redex left to r_i; a reduction sequence $\sigma : t \leadsto^* t'$ is standardisable if there exists a standard reduction sequence $\sigma_s : t \leadsto^* t'$.

Lemma 8 *Given $t \overset{\sigma_s}{\leadsto}{}^* u \overset{r}{\leadsto} v$, where σ_s is standard and r is a residual of some redex r_t in t; if no redexes left to r_t or their residuals are contracted in σ_s, then we can construct a standard reduction sequence $t \overset{\tau}{\leadsto}{}^* v$ with $|\tau| \leq 1 + \max\{m(r), 1\} \cdot |\sigma_s|$.*

Proof Let us proceed by structural induction on t.

- $t = \lambda x.t_0$. By induction hypothesis on t_0, this case is trivial.
- $t = t_0 t_1$, and r_t is not t. $\sigma_s = \sigma_0 + \sigma_1$, where $t_i \overset{\sigma_i}{\leadsto}^* u_i$ for $i = 0, 1$, and $u = u_0 u_1$. Note r_t is in t_i for some $i \in \{0, 1\}$, and thus, $v = v_0 v_1$, where $u_i \overset{r}{\leadsto} v_i$ and $u_{1-i} = v_{1-i}$. By induction hypothesis, we can construct a standard reduction sequence $t_i \overset{\tau_i}{\leadsto}^* v_i$ with $|\tau_i| \leq 1 + \max\{m(r), 1\} \cdot |\sigma_i|$. Note that $i = 1$ implies $|\sigma_0| = 0$. Let $\tau = \tau_0 + \sigma_1$ if $i = 0$ or τ_1 if $i = 1$, then $t \overset{\tau}{\leadsto}^* v$ is standard. It can be readily verified that $|\tau| \leq 1 + \max\{m(r), 1\} \cdot |\sigma_s|$.
- $t = (\lambda x.t_0)t_1$, and r_t is t. $\sigma_s = \sigma_0 + \sigma_1$, where $t_i \overset{\sigma_i}{\leadsto}^* u_i$ for $i = 0, 1$, and $r = u = (\lambda x.u_0)u_1$. Hence, $v = [u_1/x]u_0$. Let $\sigma_0 = r_1 + \cdots + r_n$, and $\sigma_0^* = r_1^* + \cdots + r_n^*$, where $r_j^* = [t_1/x]r_j$ for $j = 1, \ldots, n$. Then $\sigma_0^* : [t_1/x]t_0 \leadsto^* [t_1/x]u_0$ is standard. Notice that $\sigma_0^* + \sigma_1 + \cdots + \sigma_1$ is a reduction sequence which reduces $[t_1/x]t_0$ to $v = [u_1/x]u_0$, where σ_1 occurs $m(r)$ times and each σ_1 reduces one occurrence of t_1 in $[t_1/x]u_0$ to u_1. If a redex contracted in some σ_1 is left to some r_j^*, then all redexes contracted in that σ_1 are left to that r_j^*. Hence, we can move that σ_1 to the front of r_j^*. In this way, we can construct a standard reduction sequence from $[t_1/x]t_0$ to $v = [u_1/x]u_0$ in the following form.

  $$\sigma_s^* = \cdots + r_1^* + \cdots + \ldots + \cdots + r_n^* + \cdots,$$

  where $\cdots$ stands for a reduction sequence of form $\sigma_1 + \cdots + \sigma_1$, which may be empty, and r_j^* may also denote their corresponding residuals. Hence $\tau = r_t + \sigma_s^*$ is a standard reduction sequence from t to v. Notice

  $$|t| = 1 + |\sigma_s^*| = 1 + |\sigma_0^*| + m(r) * |\sigma_1| \leq 1 + \max\{m(r), 1\} \cdot |\sigma_s|.$$

■

Lemma 9 *Given $t \overset{\sigma_s}{\leadsto}^* u \overset{r}{\leadsto} v$, where σ_s is standard; then we can construct a standard reduction sequence $t \overset{\tau}{\leadsto}^* v$ with $|\tau| \leq 1 + \max\{m(r), 1\} \cdot |\sigma_s|$.*

Proof The proof proceeds by induction on $|\sigma_s|$. Let $\tau = r$ if $|\sigma_s| = 0$. Now assume $\sigma_s = r' + \sigma_s'$, where $t \overset{r'}{\leadsto} t' \overset{\sigma_s'}{\leadsto}^* u$. Now we have two cases.

- r is a residual of some redex in t which is left to r'. By Lemma 8, we are done.
- r is not a residual of any redexes in t which are left to r'. By induction hypothesis, we can construct a standard reduction sequence $t' \overset{\tau'}{\leadsto}^* v$ with $|\tau'| \leq 1 + \max\{m(r), 1\} \cdot |\sigma_s'|$. Let $\tau = r' + \tau'$, then it can be readily verified that τ is standard according to the construction. Note $|\tau| = 1 + |\tau'| \leq 1 + \max\{m(r), 1\} \cdot (1 + |\sigma_s'|) = 1 + \max\{m(r), 1\} \cdot |\sigma_s|$. ■

Theorem 10 *(Standardisation) Every finite β-reduction sequence is standardisable.*

Proof Given $t \overset{\sigma}{\leadsto}^* v$, let us proceed by induction on $|\sigma|$. Assume $\sigma = \sigma' + r$, where $t \overset{\sigma'}{\leadsto}^* u \overset{r}{\leadsto} v$. By induction hypothesis, we can construct a standard reduction sequence $t \overset{\sigma_s'}{\leadsto}^* u$. Hence, Lemma 9 yields the result. ■

4. The Upper Bounds

It is clear from the previous proofs that we actually have an algorithm to transform any reduction sequences into standard ones. Let std(σ) denote the standard reduction sequence obtained from transforming a given reduction sequence σ, and we are ready to give some upper bounds for the number of reduction steps in std(σ).

Theorem 11 *(Standardisation with bound) Given a reduction sequence* $t \overset{\sigma}{\leadsto}^* u$, *where* $\sigma = r_0 + r_1 + \cdots + r_n$ *for some* $n \geq 1$, *then there exists a standard reduction sequence* $t \overset{\sigma_s}{\leadsto}^* u$ *with* $|\sigma_s| \leq (1 + \max\{m(r_1), 1\}) \cdots (1 + \max\{m(r_n), 1\})$.

Proof Let $\sigma_0 = r_0$, $\sigma_i = r_0 + r_1 + \cdots + r_i$ and $l_i = |\text{std}(\sigma_i)|$ for $i = 1, \ldots, n$. By Lemma 9, we have $l_{i+1} \leq 1 + \max\{m(r_{i+1}), 1\} \cdot l_i$ for $i = 0, 1, \ldots, n-1$ according to the proof of Theorem 10. Note $1 \leq l_i$, and thus, for $i = 0, 1, \ldots, n-1$,

$$l_{i+1} \leq 1 + \max\{m(r_{i+1}), 1\} \cdot l_i \leq (1 + \max\{m(r_{i+1}), 1\}) \cdot l_i.$$

Since $l_0 = 1$, this yields $l_n \leq (1 + \max\{m(r_1), 1\}) \cdots (1 + \max\{m(r_n), 1\})$. Note $\sigma = \sigma_n$. Let $\sigma_s = \text{std}(\sigma_n)$, then we are done. ∎

Clearly, this simple bound is not very tight. With a closer study, a tighter but more complex bound can be given in the same fashion. Unlike many earlier proofs in the literature, our proof of the standardisation theorem does not use the *finiteness of developments theorem.* In this respect, our proof is similar to the one in [Tak95]. As a matter of fact, Theorem 11 can be modified to show that all developments are finite, following the application in the next section. We will not pursue in this direction since the work in [dV85] has produced an exact bound for finiteness of developments.

Given $t \leadsto^n u$, we can also give a bound in terms of n and the complexity of t defined below.

Definition 12 *The complexity* $|t|$ *of a term* t *is defined inductively as follows.*

$$|t| = \begin{cases} 1 & \text{if } t \text{ is a variable;} \\ 1 + |t_0| & \text{if } t = \lambda x.t_0; \\ |t_0| + |t_1| & \text{if } t = t_0 t_1. \end{cases}$$

Proposition 13 *If* $t \leadsto u$ *then* $|u| < |t|^2$.

Proof A structural induction on t yields the result. ∎

Corollary 14 *If* $t \leadsto^n u$, *then there is a standard reduction sequence* $t \overset{\sigma_s}{\leadsto}^* u$ *with* $|\sigma_s| < |t|^{2^n}$.

Proof This clearly holds if $n = 1$. Now assume $r_0 + r_1 + \cdots + r_{n-1} : t \rightsquigarrow^n u$. By Proposition 13, we have $|r_i| < |t|^{2^i}$ for $i = 1, \ldots, n-1$, which yields $1 + \max\{m(r_i), 1\} \leq |t|^{2^i}$ for $i = 1, \ldots, n-1$. By Theorem 11, we can construct a standard reduction sequence $\sigma_s : t \rightsquigarrow^* u$ with $|\sigma_s| \leq |t|^{2^1} \cdots |t|^{2^{n-1}} < |t|^{2^n}$. ∎

Now we introduce a lemma which will be used in the next section.

Lemma 15 *If $\sigma : t \rightsquigarrow^* u$ is a development, then $|u| < 2^{|t|}$.*

Proof This can be verified by a structural induction on t. ∎

5. An Application

It is a well-known fact that the simply typed $\lambda^{\rightarrow}$-calculus enjoys strong normalisation property. In this section, as an application of our previous result, we will present an upper bound for the lengths of all the reduction sequences from any given $\lambda^{\rightarrow}$-term t. Among various proofs showing the strong normalisation property of $\lambda^{\rightarrow}$, a few, such as the ones in [Gan80a] and [Sch91], present some superexponential upper bounds for longest reduction sequences from given $\lambda^{\rightarrow}$-terms. Gandy invents an semantic approach in [Gan80a], which is called *functional interpretations* and has its traces in many following papers such as [Sch82], [Pol94] and [Kah95]. In [Sch91], Schwichtenberg adopts a syntactic approach from [How80], which bases on cut elimination in intuitionistic logic.

Compared with other related methods in the literature, our following syntactic method is not only innovative but also yields an quite intelligible and tight bound. It also exhibits a nice way to transform strong normalisation into weak normalisation in $\lambda^{\rightarrow}$, simplifying a much involved transformation in [Sch91]. Therefore, the new transformation has its own value in this respect. We start with a weak normalisation proof due to Turing according to [Gan80], which can also be found in many other literatures such as [And71] and [GLT89].

5.1. A bound for $\lambda^{\rightarrow}I$-terms

Since the leftmost reduction sequence from any λI-term t is a longest one among all reduction sequences from t, it goes straightforward to establish a bound for $\lambda^{\rightarrow}I$-terms if we can find any normalisation sequences for them. In order to get a tighter bound, the key is to find shorter normalisation sequences.

Definition 16 *(Simple Types and $\lambda^{\rightarrow}$-terms) Types are formulated in the following way.*

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

$\lambda^{\rightarrow}$*-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 *and* x *does occur free in* 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. On the other hand, superscripts may be used to indicate the types of $\lambda^{\rightarrow}$-terms.

Definition 17 *The rank* $\rho(T)$ *of a simple type* T *is defined as follows.*

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

The rank $\rho(r)$ *of a redex* $r = (\lambda x^U.v^V)u^U$ *is* $\rho(U \to V)$*, and the rank of a term* t *is*

$$\rho(t) = \begin{cases} \langle 0, 0\rangle \text{ if } t \text{ is in } \beta\text{-normal form; or} \\ \langle k = \max\{\rho(r) : r \text{ is a redex in } t\}, \text{ the number of redexes } r \text{ in } t \text{ with } \rho(r) = k\rangle. \end{cases}$$

The ranks of terms are lexically ordered.

Notice that a redex has a redex rank, which is a number, and also has a term rank, which is a pair of numbers.

Observations Now let us observe the followings.

- If $t \rightsquigarrow t'$ and redex r' in t' is a residual of some redex r in t, then $\rho(r') = \rho(r)$.
- Given $t = t[r]$ with $\rho(t) = \langle k, n\rangle$, where $r = (\lambda x^V.u^U)v^V$ is a redex with $\rho(r) = k$ and no redexes in r have rank k. Then $\rho(t') < \rho(t)$ for $t \rightsquigarrow t' = t[[v/x]u]$. This can be verified by counting the number of redexes in t' with rank k. It is easy to see that any redex in t' which is not a residual must have rank $\rho(U)$ or $\rho(V)$, which is less than k. Hence, a redex in r' with rank k must be a residual of some redex r_1 in t with rank k. Note r_1 has only one residual in t' since r_1 is not in r. This yields $\rho(t') < \rho(t)$ since $\rho(t')$ is either $\langle k, n-1\rangle$ or $\langle k', n'\rangle$ for some $k' < k$.

Lemma 18 *Given* t *with* $\rho(t) = \langle k, n\rangle$ *for some* $k > 0$*; then we can construct a development* $\sigma : t \rightsquigarrow^* u$ *such that* $|\sigma| = n$ *and* $\rho(u) = \langle k', n'\rangle$ *for some* $k' < k$.

Proof Following the observations, we can always reduce innermost redexes with rank k until there exist no redexes with rank k. This takes n steps and reaches a term with a less rank. ■

Definition 19 *Let functions 2_k for $k = 0, 1, \ldots$ be defined as follows.*

$$2_k(x) = \begin{cases} x & \text{if } k = 0; \\ 2^{2_{k-1}(x)} & \text{if } k > 0. \end{cases}$$

Also we define

$$m(\sigma) = \begin{cases} 1 & \text{if } |\sigma| = 0; \\ (1 + \max\{m(r_1), 1\}) \cdots (1 + \max\{m(r_n), 1\}) & \text{if } \sigma = r_1 + \cdots + r_n. \end{cases}$$

Clearly, $m(\sigma_1 + \sigma_2) = m(\sigma_1)m(\sigma_2)$.

Theorem 20 *If t is a $\lambda^{\rightarrow}$-term with $\rho(t) = \langle k, n\rangle$ for some $k > 0$, then there exists $\sigma : t \rightsquigarrow^* u$ such that u is in β-normal form and $m(\sigma) < 2_1(\sum_{i=1}^{k}(2_{i-1}(|t|))^2)$.*

Proof By Lemma 18, there exists a development $\sigma' : t \rightsquigarrow^* t'$ with $|\sigma'| = n$ and $\rho(t') = \langle k', n'\rangle$ for some $k' < k$. Let $\sigma' = r_1 + \cdots + r_n$, then $1 + m(r_n) < 2^{|t|}$ by Lemma 15. Hence, $m(\sigma') < 2^{n|t|} < 2^{|t|^2}$ since $n < |t|$. Now let us proceed by induction on k.

- $k = 1$. Since t' is in normal form, let $\sigma = \sigma'$ and we are done.
- $k > 1$. By induction hypothesis, there exists $\sigma'' : t' \rightsquigarrow^* u$ such that u is in β-normal form and $m(\sigma'') < 2_1(\sum_{i=1}^{k-1}(2_{i-1}(|t'|))^2)$. Let $\sigma = \sigma' + \sigma''$, then

$$m(\sigma) = m(\sigma')m(\sigma'') < 2_1(|t|^2)2_1(\sum_{i=1}^{k-1}(2_{i-1}(|t'|))^2) < 2_1(\sum_{i=1}^{k}(2_{i-1}(|t|))^2)$$

since $|t'| < 2^{|t|}$ by Lemma 15. ■

It is a well-known fact that the leftmost reduction sequence from a λI-term t is a longest one if t has a normal form.

Corollary 21 *Given any simply typed $\lambda^{\rightarrow}I$-term t with $\rho(t) = \langle k, n\rangle$; every reduction sequence from t is of length less than $2_{k+1}(|t|)$.*

Proof It can be verified that the result holds if $|t| \leq 3$. For $|t| > 3$, we have $2_1(\sum_{i=1}^{k}(2_{i-1}(|t|))^2) \leq 2_{k+1}(|t|)$. By Theorem 20, there exists $\sigma : t \rightsquigarrow^* u$ such that u is in β-normal form and $m(\sigma) < 2_{k+1}(|t|)$. This yields that $\mathrm{std}(\sigma) < 2_{k+1}(|t|)$ by Theorem 11. Since t is a $\lambda^{\rightarrow}I$-term, the leftmost reduction reduction sequence from t is a longest one. This concludes the proof. ■

Notice that the leftmost reduction sequence from t may not yield a longest one if t is not a $\lambda^{\rightarrow}I$-term. Therefore, the proof of Corollary 21 cannot go through directly for all $\lambda^{\rightarrow}$-terms.

5.2. A bound for $\lambda^{\rightarrow}$-terms

Our following method is to transform a $\lambda^{\rightarrow}$-term t into a $\lambda^{\rightarrow}I$-term $\mathcal{T}(t)$ such that $\mu(t) \leq \mu(\mathcal{T}(t))$. Since we have already established a bound for $\mathcal{T}(t)$, this bound certainly works for t.

Lemma 22 *Given $t = ru_1 \dots u_n$ and $t_0 = ([v/x]u)u_1 \dots u_n$, where $r = (\lambda x.u)v$; if t_0 and v are strongly normalisable, then t is strongly normalisable and $\mu(t) \leq 1 + \mu(t_0) + \mu(v)$.*

Proof Let $\sigma : t \rightsquigarrow^* t^*$ be a reduction sequence, and we verify that $|\sigma| \leq 1 + \mu(t_0) + \mu(v)$. Clearly, we can assume that redex r or some residual of r has been contracted in σ. Then $\sigma = \sigma_1 + r' + \sigma_2$ is of the following form.

$$t \overset{\sigma_1}{\rightsquigarrow}{}^* (\lambda x.u')v'u'_1 \dots u'_n \overset{r'}{\rightsquigarrow} ([v'/x]u')u'_1 \dots u'_n \overset{\sigma_2}{\rightsquigarrow}{}^* t^*,$$

where $\sigma_1 = \sigma_u + \sigma_v + \sigma_{u_1} + \dots + \sigma_{u_n}$ for $u \overset{\sigma_u}{\rightsquigarrow}{}^* u'$, $v \overset{\sigma_v}{\rightsquigarrow}{}^* v'$, $u_1 \overset{\sigma_{u_1}}{\rightsquigarrow}{}^* u'_1$, ..., and $u_n \overset{\sigma_{u_n}}{\rightsquigarrow}{}^* u'_n$. Let $\tau_1 : [v/x]u \rightsquigarrow^* [v'/x]u$ be the reduction sequence which reduces each occurrence of v in $[v/x]u$ to v' by following σ_v, and $\tau_2 : [v'/x]u \rightsquigarrow^* [v'/x]u'$ be the reduction sequence which reduces $[v'/x]u$ to $[v'/x]u'$ by following σ_u. Clearly, $|\tau_1| = m(r)|\sigma_v|$ and $|\tau_2| = |\sigma_u|$. Also let $\tau : ([v/x]u)u_1 \dots u_n \rightsquigarrow^* t^*$ be $\tau_1 + \tau_2 + \sigma_{u_1} + \dots + \sigma_{u_n} + \sigma_2$, then $|\tau| \leq \mu(t_0)$ by definition. Note

$$|\sigma| = |\sigma_1 + r' + \sigma_2| = |\sigma_u| + |\sigma_{u_1}| + \dots + |\sigma_{u_n}| + 1 + |\sigma_2| + |\sigma_v| \leq 1 + |\tau| + |\sigma_v|.$$

By definition, $|\sigma_v| \leq \mu(v)$. Hence, $|\sigma| \leq 1 + \mu(t_0) + \mu(v)$. ■

Definition 23 *(Transformation) To facilitate the presentation, we assume that there exist constants $\langle , \rangle$ of type $U \rightarrow (V \rightarrow U)$ for all types U and V. Let $\langle u, v \rangle$ denote $\langle , \rangle uv$.*

$$\mathcal{T}(t) = \begin{cases} t & t \text{ is a variable;} \\ \lambda x \lambda y_1^{U_1} \dots \lambda y_m^{U_m}.\langle \mathcal{T}(t_0)y_1 \cdots y_m, x\rangle & t = \lambda x.t_0, \text{ where } t_0 \text{ has type } U_1 \rightarrow \dots \rightarrow U_m \rightarrow V, \\ & \text{and } V \text{ is atomic.} \\ \mathcal{T}(t_0)\mathcal{T}(t_1) & t = t_0t_1. \end{cases}$$

Clearly, $\langle , \rangle$ can always be replaced by a free variable of the same type without changing the normalisability of terms.

Proposition 24 *For every $\lambda^{\rightarrow}$-term t of type T, we have the followings.*

1. *$\mathcal{T}(t)$ is a $\lambda^{\rightarrow}I$-term of type T;*
2. *$\mathcal{T}([u/x^U]t) = [\mathcal{T}(u)/x^U]\mathcal{T}(t)$ for any $\lambda^{\rightarrow}$-term u of type U;*
3. *$\mu(t) \leq \mu(\mathcal{T}(t))$.*

Proof (1) and (2) can be readily proven by structural induction on t. By (1) and Corollary 21, we know $\mu(\mathcal{T}(t))$ exists for every $\lambda^{\rightarrow}$-term t. We now proceed to show (3) by induction on $\mu(\mathcal{T}(t))$ and the structure of $\mathcal{T}(t)$, lexicalgraphically ordered.

- $t = \lambda x.u$. By induction, $\mu(t) = \mu(u) \leq \mu(\mathcal{T}(u)) \leq \mu(\mathcal{T}(t))$.
- $t = au_1 \ldots u_n$, where a is some variable. Note $\mu(\mathcal{T}(t)) = a\mathcal{T}(u_1) \ldots \mathcal{T}(u_n)$. By induction hypothesis, $\mu(t) = \mu(u_1) + \ldots + \mu(u_n) \leq \mu(\mathcal{T}(u_1)) + \ldots \mu(\mathcal{T}(u_n)) = \mu(\mathcal{T}(t))$.
- $t = ru_1 \ldots u_n$, where $r = (\lambda x.u)v$. By definition, $\mathcal{T}(t) = \mathcal{T}(r)\mathcal{T}(u_1) \ldots \mathcal{T}(u_n)$, and $\mathcal{T}(r) = \lambda x \lambda y_1 \ldots \lambda y_m.\langle \mathcal{T}(u) y_1 \ldots y_m, x\rangle \mathcal{T}(v)$. Hence,

$$\mathcal{T}(r) \rightsquigarrow \lambda y_1 \ldots \lambda y_m.\langle ([\mathcal{T}(v)/x]\mathcal{T}(u)) y_1 \ldots y_m, \mathcal{T}(v)\rangle.$$

Since $\mathcal{T}(u) y_1 \ldots y_m$ is of atomic type, $m \geq n$. This yields

$$\mathcal{T}(t) \rightsquigarrow^* \lambda y_{m-n+1} \ldots \lambda y_m.\langle (([\mathcal{T}(v)/x]\mathcal{T}(u))\mathcal{T}(u_1) \ldots \mathcal{T}(u_n) y_{m-n+1} \ldots y_m, \mathcal{T}(v)\rangle.$$

By (2), $[\mathcal{T}(v)/x]\mathcal{T}(u) = \mathcal{T}([v/x]u)$, and thus,

$$([\mathcal{T}(v)/x]\mathcal{T}(u))\mathcal{T}(u_1) \ldots \mathcal{T}(u_n) = \mathcal{T}(([v/x]u)u_1 \ldots u_n).$$

By induction hypothesis, $\mu(([v/x]u)u_1 \ldots u_n) \leq \mu(\mathcal{T}(([v/x]u)u_1 \ldots u_n))$. Therefore, by Lemma 22,

$$\mu(t) \leq 1 + \mu(([v/x]u)u_1 \ldots u_n) + \mu(v) \leq 1 + \mu(\mathcal{T}(([v/x]u)u_1 \ldots u_n)) + \mu(\mathcal{T}(v)) \leq \mu(\mathcal{T}(t)).$$

■

Corollary 25 *Given any simply typed $\lambda^{\rightarrow}$-term t with $\rho(t) = \langle k, n\rangle$; every reduction sequence from t is of length less than $2_{k+1}((2k+3)|t|)$.*

Proof Given a subterm $\lambda x.u$ of type $U = U_1 \rightarrow \cdots \rightarrow U_m \rightarrow V$ in t, where V is atomic, we can simply transform $\lambda x.u$ into $\lambda x.\mathcal{T}(u)$ if $k < \rho(U)$ since no redexes with rank greater than k can occur in any reduction sequence of t; if $\rho(U) \leq k$, we have

$$|\mathcal{T}(\lambda x.u)| = |\lambda x \lambda y_1 \ldots \lambda y_m.\langle \mathcal{T}(u) y_1 \ldots y_m, x\rangle| = |\mathcal{T}(u)| + 2m + 3 \leq |\mathcal{T}(u)| + 2k + 3.$$

Thus, it can be readily shown that $|\mathcal{T}(t)| \leq (2k+3)|t|$. Also it can verified that, if $\rho(t) = \langle k, n\rangle$ for some k and n then $\rho(\mathcal{T}(t)) = \langle k, n\rangle$ by the definition. By Corollary 21, we have $\mu(\mathcal{T}(t)) < 2_{k+1}((2k+3)|t|)$. This yields $\mu(t) < 2_{k+1}((2k+3)|t|)$ by Proposition 24 (3). ■

6. Related work and Conclusion

For those who know the strong equivalence relation $\cong$ on reductions in [Bar84], originally due to Berry and Lévy, it can be verified that $\sigma \cong \mathrm{std}(\sigma)$ for all reduction sequences σ.

There is a short proof of the standardisation theorem due to Mitschke [Mit79], which analyses the relation between head and internal reductions. It shows any reduction sequence can be transformed into one which starts with head reductions followed by internal reductions. In this formulation, it is not easy to extract a bound from the proof. There are also two proofs due to Klop [Klo80], to which the present proof bears some connection. Though all these proofs aim at commuting the contracted leftmost redexes to the front, our proof uses an entirely different strategy to show the termination of such commutations. While Klop focuses on the strong equivalence relation $\cong$, we establish Lemma 8 by a structural induction without using the finiteness developments theorem. This naturally yields an upper bound for standardisations.

In our application, an upper bound is given for the lengths of reduction sequences in $\lambda^{\rightarrow}$. This is a desirable result since $\mu(t)$, the length of a longest reduction sequence from t, can often be used as an induction order in many proofs. Gandy mentions a similar bound in [Gan80a] but details are left out. His semantic method, which aims at giving strong normalisation proof, is utterly different from ours. Schwichtenberg presents a similar bound in [Sch91] using an approach adapted from [How80]. His method of transforming $\lambda^{\rightarrow}$-terms into $\lambda^{\rightarrow}I$-terms closely relates to our presented method but is very much involved. It seems that his entire proof is less transparent, and therefore, obscures the merits in it. In addition, the proof of *finiteness of developments* theorem by de Vrijer [dV85] yields an exact bound for the lengths of developments, and thus, is casually related to our proof of the standardisation theorem with bound.

In Gentzen's sequent calculus, there exists a similar bound for the sizes of cut-free proofs obtained from cut elimination. Mints [Min79] (of which I have only learned the abstract) gives a way of computing the maximum length of a reduction from the length of a standard reduction sequence. In this respect, our work can be combined with his to show the maximum length of a reduction sequence from the length of an *arbitrary* one. This also motivates our planning to establish a similar bound for the first-order λ-calculus with dependent types. On the other hand, Statman [Sta79] suggests that a lower bound for $\mu(t)$ have the same superexponential form, and this makes it a challenging task to sharpen our presented bound for $\mu(t)$ though it seems to be greatly exaggerated.

7. Acknowledgement

I thank Frank Pfenning, Peter Andrews and Richard Statman for their support and for providing me a nice working environment. I also thank some anonymous referees for their criticisms and suggestions on a draft of this paper, which have certainly enhanced its quality to a large extent.

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.*

[Ber78] G. Berry (1978), Séquentialité de l'évaluation formelle des λ-expressions, *Proc. 3-e Colloque International sur la Programmation, Paris.*

[Chu41] A. Church, (1941), The calculi of lambda conversion, *Princeton University Press, Princeton.*

[CF58] H.B. Curry and R. Feys (1958), Combinatory Logic, *North-Holland Publishing Company, Amsterdam.*

[dV85] R. de Vrijer (1985), A direct proof of the finite developments theorem, *Journal of Symbolic Logic*, 50:339-343.

[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*, edited by J.P. Seldin and J.R. Hindley, 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*, edited by J.P. Seldin and J.R. Hindley, Academic press, pp. 457-478.

[GLT89] J.-Y. Girard et al. (1989), Proofs and types, *Cambridge Press*, 176 pp.

[Hue94] Gérard Huet (1994), Residual Theory in λ-Calculus: A Formal Development, *Journal of Functional Programming vol. 4*, pp. 371–394.

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

[How80] W. Howard (1980), Ordinal analysis of terms of finite type, *Journal of Symbolic Logic*, 45(3):493-504.

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

[Kah95] Stefan Kahrs (1995), Towards a Domain Theory for Termination Proofs, *Laboratory for Foundation of Computer Science*, 95-314, Department of Computer Science, The University of Edinburgh.

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

[Lév78] J.-J. Lévy (1978), Réductions correctes et optimales dans le lambda calcul, *Thèse de doctorat d'état, Université Paris VII.*

[Min79] G.E. Mints (1979), A primitive recursive bound of strong normalisation for predicate calculus (in Russian with English abstract), *Zapiski Naucnyh Seminarov Leningradskogo Otdelenija Matematiceskogo Instituta im V.A. Steklova Akademii Nauk SSSR (LOMI) 88*, pp. 131-135.

[Mit79] G. Mitschke (1979), The standardization theorem for the λ-calculus, *Z. Math. Logik Grundlag. Math. 25*, pp. 29-31.

[Pol94] J. van de Pol (1994), Strict functionals for termination proofs, *Lecture Notes in Computer Science 902*, edited by J. Heering, pp. 350-364.

[Sta79] Richard Statman (1979), The typed λ-calculus is not elementary, *Theoretical Computer Science 9*, pp. 73-81.

[Sch82] H. Schwichtenberg (1982), Complexity of normalisation in the pure typed lambda-calculus, *The L.E.J. Brouwer Centenary Symposium*, edited by A.S. Troelstra and D. van Dalen, North-Holland publishing company, pp. 453-457.

[Sch91] H. Schwichtenberg (1991), An upper bound for reduction sequences in the typed lambda-calculus, *Archive for Mathematical Logic*, 30:405-408.

[Tak95] Masako Takahashi (1995), Parallel Reductions in λ-Calculus, *Information and Computation 118*, pp. 120–127.

[Wad76] C.P. Wadsworth (1976), The relation between computational and denotational properties for Scott's D_∞-models of λ-calculus, *SIAM Journal of Computing*, 5(3):488-521.