Mathematical Problems of Computer Science 29, 2007, 5–15.

Analysis of Bounds for Lengths of Reductions in Typed λ-calculus

Tigran M. Galoyan

Institue for Informatics and Automation Problems of NAS of RA
e-mail: tigran.galoyan@gmail.com

Abstract

We analyze bounds for the lengths of arbitrary reduction sequences of terms in typed λ-calculus, consider some estimates obtained by the other authors and compare these estimates. The cut elimination and normalization algorithms are also investigated in this paper. Thereafter we refine the estimates achieved in [3] (for pure implicational logic only) by supplement of η-conversion and then we extend evaluations to first-order logic.

1 Preface

The key observation is that the number of nodes with conversions in the head reduction tree of a term bounds the length of any reduction sequence of that term. It is well known that the full reduction tree for any term of the typed λ-calculus is finite, and hence also any reduction sequence of that term is finite. The size of this tree yields a nontrivial bound on the maximal length of a reduction chain starting with the term, since the tree represents the worst case reductions.

In [7] the author considers head reduction trees of λ-I-terms. A term r is called a λ-I-term if for any subterm of the form $\lambda x.s$ one has $x \in FA(s)$, where $FA(s)$ is the set of free assumption variables in s. Here and from now on we use the word *term* for derivation terms (ref [8]) as we are interested in normalization of derivations. By normalization we mean a collection of algorithms transforming a given derivation into a certain normal form, i.e. it does not contain any "detour" (ref [6]). That is the reason why we use the notion FA - the set of free assumption variables (ref [8]).

In [3] the author considers head reduction trees of an arbitrary term in simple typed λ-calculus. Each node labeled with β-redex $(\lambda x.r)s$ will have two child-nodes, $r[x := s]$ and s. Hence, also in the case $x \notin FA(r)$ the head reduction tree controls conversions in s. The main difference of these two calculi investigated in [7] and in [3] is besides some refinements, that in [3] λ-terms of arbitrary level are derived (whereas in [7] λ-terms of 0 level are derived) and the estimate is independent from the arities of subterms of a term. In other paper [2] the authors develop a perspicuous method for classifying the derivation lengths of GÖDEL's T. Here the derivation length of a term r is the longest possible reduction sequence starting from r. Following ideas from [3] they extend the previous approach to GÖDEL's T, where terms may be constructed using **R** recursor. That is an expanded head reduction tree is assigned to each term $t \in T$. Furthermore, the head reduction trees are extended by a cut-rule and an appropriate miniaturization of Buchholz' Ω-rule which allow a simple embedding of any term of GÖDEL's T into the extended calculus.

There are also interesting researches of cut elimination, with distinguished estimates of the bound on the length of the resulting cut-free derivation, carried out in [6] and in [1].

2 Preliminaries

Let us first get acquainted with some necessary definitions and notions which will be useful after. By $r^\rho, s^\rho, t^\rho, p^\rho, q^\rho$... we denote derivation terms (simply typed lambda-terms), which are built from assumption variables $x^\rho, y^\rho, z^\rho \ldots$ by the introduction and elimination rules for $\to$ (for the moment we consider only implicational logic):

$\to^+$ - implication introduction - $(\lambda x^\rho r^\sigma)^{\rho\to\sigma}$;

$\to^-$ - implication elimination - $(s^{\rho\to\sigma} t^\rho)^\sigma$.

Though assumption variables and derivation terms have types (henceforth we use the word *type* for formula), for the convenience, basically they will be omitted, but implied.

The *length* $l(r)$ and the *height* $h(r)$ of a term r are defined recursively as:

$$\begin{array}{lcl} l(x) = 1 & & h(x) = 0 \\ l(\lambda x.r) = l(r) + 1 & \text{and} & h(\lambda x.r) = h(r) + 1 \\ l(rs) = l(r) + l(s) & & h(rs) = \max(h(r), h(s)) + 1 \end{array}$$

The *level* $lev(r^\sigma)$ of a term r^σ is defined to be the level $lev(\sigma)$ of its type σ, where the *level* of a type formula is defined recursively as: for a ground type ι (i.e. for an atomic formula) $lev(\iota) = 0$ and $lev(\rho \to \sigma) = \max(lev(\rho) + 1, lev(\sigma))$.

The *degree* $g(r)$ of a term r is defined to be the maximum of the levels of subterms of r.

With $d(r)$ we denote the maximum of lengths of arbitrary reduction sequences starting from r with respect to $\to_1$, the one step reduction, for the moment using just β-conversion rule.

With $ar(r)$ we denote the maximum of arities of free assumption variables of r. As usual the arity of a variable x is the maximal number of parameters it admits, e.g. if x has a type $\tau_1 \to (\tau_2 \to \ldots \to (\tau_n \to \tau)\ldots)$ and τ is a ground type, then $ar(x) = n$. We write $\tau_1, \ldots, \tau_n \to \tau$ for $(\tau_1 \to (\tau_2 \to \ldots \to (\tau_n \to \tau)\ldots))$. For instance, let x, y are variables of type $(\iota^n \to \iota) \to \iota$ and $\iota^n \to \iota$ respectively, then $ar(xy) = n$.

Terms of the form $(\lambda x.r)s$ are called convertible - $(\lambda x.r)s$ converts into $r_x[s]$. A convertible derivation $(\lambda x^\sigma.r)s$ is also called a *cut* with *cut-formula* σ. By the level of a cut we mean the level of its cut-formula. The *cut-rank* of a derivation r is the least number bigger than the levels of all cuts in r.

3 Obtained bounds for lengths of reductions

As we have already mentioned at the beginning of the paper the principal observation in [7] as well as in [3] is that the number of nodes with conversions in the head reduction tree of a term bounds the length of any reduction sequence of that term.

In [7] the author imposes some restrictions and hypotheses. Firstly he considers head reduction trees of λ-I-terms. This hypothesis is necessary, since in non-λ-I-terms subterms can disappear by means of conversions, and hence the head reduction tree may not show any trace of a conversion inside the term. For instance, the terms $(\lambda x.y)((\lambda x.p)q)$ and $(\lambda x.y)(p_x[q])$ have the same head reduction tree (consisting of one additional node labeled y). Afterwards, to reduce the general case to the case of λ-I-terms, the author introduces the notion of variant of a term. For this purpose, the author introduces dummy variables which turn the given term into a corresponding λ-I-term. Secondly, there is a restriction on the level of the term to be derived. The calculus is defined for terms of level 0.

For r of level 0 the relation $\vdash^{\alpha}_{m} r$ (to be read r is derivable with reduction branch of height $\leq \alpha$ and cut-rank $\leq m$) is defined inductively by

- (*β – Rule*) If $\vdash^{\alpha}_{m} r_x[s]\vec{t}$, then $\vdash^{\alpha+1}_{m} (\lambda x.r)s\vec{t}$.
- (*Variable Rule*) If $\vdash^{\alpha}_{m} t_i\vec{y_i}$ for $i = 1, \ldots, n$, then $\vdash^{\alpha+1}_{m} xt_1 \cdots t_n$. In particular, $\vdash^{\alpha+1}_{m} x$ for any α and m.
- (*Cut Rule*) If $\vdash^{\alpha}_{m} ry_1 \cdots y_n$ with $n \geq 1$ and $\vdash^{\alpha}_{m} t_i\vec{y_i}$ and $lev(t_i) < m$ for $i = 1, \ldots, n$, then $\vdash^{\alpha+1}_{m} rt_1 \cdots t_n$.

Note that $\vdash^{\alpha}_{0} r$ is generated by a uniquely determined rule. Hence the generation tree (with the α's stripped off) is uniquely determined; we call it the *expanded head reduction tree* of r (sometimes for brevity *head reduction tree*).

Based on the aforesaid the author gives the following estimate of bound for length of any reduction sequence for the defined calculus:

Theorem 1. *Let r be a term of the typed λ-calculus of level 0 with requirement that $ar(r) \geq 2$. Then the length of an arbitrary reduction sequence for r with respect to $\to_1$ is bounded by*

$$ar(r)^{2_{g(r)}(g(r)+2\cdot h(r)+2\cdot ar(r)+2)},$$

where $2_m(n)$ is recursively defined by

$$2_0(n) = n \ \ and \ \ 2_{m+1}(n) = 2^{2_m(n)}.$$

In [3] it is shown that the preceding bound can be improved to $2_{g(r)}(l(r))$, respectively $2_{g(r)+1}(h(r))$, since it is easy to show that $l(r) \leq 2^{h(r)}$. Here the author considers head reduction trees of an arbitrary term in simple typed λ-calculus. The main difference of this calculus compared with the one in [7] that λ-terms of arbitrary level are derived and that the width of the expanded head reduction tree is also controlled, as well as the estimate is independent from the arities of subterms of a term.

For λ-terms r of arbitrary level (and $\alpha, m < \omega$) $\vdash^{\alpha}_{m} r$ is defined inductively by

- ($\beta - Rule$) If $\vdash^{\alpha}_{m} r[x := s]\vec{t}$ and $\vdash^{\alpha}_{m} s$, then $\vdash^{\alpha+1}_{m} (\lambda x.r)s\vec{t}$.
- ($\beta_0 - Rule$) If $\vdash^{\alpha}_{m} r$, then $\vdash^{\alpha+1}_{m} \lambda x.r$.
- ($Variable\ Rule$) If $\vdash^{\alpha}_{m} t_i$ for $i = 1, \ldots, n$, then $\vdash^{\alpha+n}_{m} xt_1 \cdots t_n$. In particular, $\vdash^{\alpha}_{m} x$ for any variable x and $\alpha, m < \omega$.
- ($Cut\ Rule$) If $\vdash^{\alpha}_{m} r$, $lev(r) \leq m$ and $\vdash^{\alpha}_{m} t$, then $\vdash^{\alpha+1}_{m} rt$.

Based on the aforesaid the author gives an exact, refined estimate of bound for lengths of reductions for the defined calculus:

Theorem 2. *Let r be a term of the typed λ-calculus of arbitrary level. Then the expanded head reduction tree of r with $g(r) > 0$ has*

$$height \leq 2_{g(r)-1}(l(r)).$$

Respectively, for the quantity $\#r$, the number of nodes with conversions in it, which bounds the length on any reduction sequence of r, it follows

$$d(r) \leq \#r \leq 2^{2_{g(r)-1}(l(r))} = 2_{g(r)}(l(r)).$$

4 Cut elimination vs normalization

As yet we consider only the pure implicational logic by means of Gentzen's rules of natural deduction. In this section we consider another type of logical calculus, the *sequent calculus* introduced by Gentzen (Gentzen type formal system, henceforth GS). It treats with *sequents* $\Gamma \Rightarrow \varphi$, where Γ is a finite set of formulas. In other words, *sequent* is a formal expression of the form $\varphi_1, \ldots, \varphi_k \Rightarrow \sigma_1, \ldots, \sigma_m$, where $k, m \geq 0$ and $\varphi_1, \ldots, \varphi_k, \sigma_1, \ldots, \sigma_m$ are formulas (ref [5]). The $\varphi_1, \ldots, \varphi_k$ part standing at the left side of $\Rightarrow$ is called *antecedent*, respectively, the right part $\sigma_1, \ldots, \sigma_m$ is called *succedent*. The full sequent has the same interpretation for GS as the formula $\varphi_1 \wedge \cdots \wedge \varphi_k \supset \sigma_1 \vee \cdots \vee \sigma_m$ has for the propositional or predicate calculi (Hilbert type formal system, henceforth HS).

In [6] the author considers only the implicational fragment of the sequent calculus. The rules of the sequent calculus for pure implicational logic in [6] are slightly reformulated as compared with the corresponding rules given in [5]. Those are the following (we write Γ, φ for $\Gamma \cup \{\varphi\}$):

($Axiom$) $\vdash \Gamma, \varphi \Rightarrow \varphi$ for φ atomic formula.

($\rightarrow -right$) If $\vdash \Gamma, \varphi \Rightarrow \psi$, then $\vdash \Gamma \Rightarrow \varphi \rightarrow \psi$.

($\rightarrow -left$) If $\vdash \Gamma, \varphi \rightarrow \psi \Rightarrow \varphi$ and $\vdash \Gamma, \varphi \rightarrow \psi, \psi \Rightarrow \chi$, then $\vdash \Gamma, \varphi \rightarrow \psi \Rightarrow \chi$.

(Cut) If $\vdash \Gamma \Rightarrow \chi$ and $\vdash \Gamma, \chi \Rightarrow \varphi$, then $\vdash \Gamma \Rightarrow \varphi$.

The notation $\vdash S$, where S is a sequent, serves as an expression of the fact, that the sequent S is derivable in GS.

One of the main results obtained by Gentzen is that the system GS is equivalent to HS in terms, that for any formula φ, $\vdash \Rightarrow \varphi$ in GS iff $\vdash \varphi$ in HS. The generalization of this result is stated in [5] by means of the following two theorems (for brevity we do not give the complete versions of the theorems):

Theorem 3-1. *If* $\Gamma \vdash \varphi$ *in HS and all variables stay fixed, then* $\vdash \Gamma \Rightarrow \varphi$ *in GS.*

Let us give some notations, which will be used in theorem 3-2. Let φ be any fixed, closed formula. Let Θ be $\sigma_1, \ldots, \sigma_m$ and $m \geq 0$. And let us suppose that Θ' is $\sigma_1, \ldots, \sigma_{m-1}$ (Θ' is empty, if $m \leq 1$), Θ'' is σ_m, if $m \geq 1$, and $\neg(\varphi \supset \varphi)$, if $m = 0$, $\neg\Theta$ is $\neg\sigma_1, \ldots, \neg\sigma_m$ and $\neg\Theta'$ is $\neg\sigma_1, \ldots, \neg\sigma_{m-1}$ (is empty, if $m \leq 1$).

Theorem 3-2. *If* $\vdash \Gamma \Rightarrow \Theta$ *in GS, then in HS* $\Gamma, \neg\Theta' \vdash \Theta''$ *and all variables are fixed. (In particular, if* $\vdash \Gamma \Rightarrow \varphi$ *in GS, then* $\Gamma \vdash \varphi$ *in HS and all variables are fixed.)*

Just based on these theorems it is stated in [6] that the sequent calculus is equivalent to natural deduction, in the sense that $\vdash \Gamma \Rightarrow \varphi$ iff φ is derivable from Γ by means of the rules $\rightarrow^{+}$ and $\rightarrow^{-}$ and assumption rule. As we know a normal derivation of φ from Γ has the property that all formulas occurring in this derivation are subformulas of either φ or a formula in Γ. The same property holds for a cut-free derivation of the sequent $\Gamma \Rightarrow \varphi$. The fundamental theorem of Gentzen, or the normal form theorem, states that the cut rule can always be eliminated from a derivation of any sequent, if none of the variables in the sequent occurs simultaneously free and bound. The same cut elimination theorem is proved in [6] in such a way that is also obtained a bound on the length of the resulting cut-free derivation, in the form

$$2^{j(d)} \cdot l(d),$$

where $l(d)$ is the length of the original derivation and $j(d)$ is the maximum taken over all paths in d of the sum of the degrees of all cut formulas on the path. It must be mentioned that the notion of degree used here (denoted $deg(\varphi)$ for a formula φ) is rather peculiar and differs from the one defined in the *Preliminaries* section of this paper (denoted $g(r)$ for a term r). Formally, the author defines the relation $\vdash^{\alpha}_{m} \Gamma \Rightarrow \varphi$ (to be read $\Gamma \Rightarrow \varphi$ is derivable with height $\leq \alpha$ (number of applications of rules) and cut-rank $\leq m$) with α, m natural numbers inductively by the following rules:

$(Axiom)$ $\vdash^{\alpha}_{m} \Gamma, \varphi \Rightarrow \varphi$ for φ atomic formula.

$(\rightarrow -right)$ If $\vdash^{\alpha}_{m} \Gamma, \varphi \Rightarrow \psi$, then $\vdash^{\alpha+1}_{m} \Gamma \Rightarrow \varphi \rightarrow \psi$.

$(\rightarrow -left)$ If $\vdash^{\alpha}_{m} \Gamma, \varphi \rightarrow \psi \Rightarrow \varphi$ and $\vdash^{\alpha}_{m} \Gamma, \varphi \rightarrow \psi, \psi \Rightarrow \chi$, then $\vdash^{\alpha+1}_{m} \Gamma, \varphi \rightarrow \psi \Rightarrow \chi$.

(Cut) If $\vdash^{\alpha}_{m} \Gamma \Rightarrow \chi$ and $\vdash^{\alpha}_{m} \Gamma, \chi \Rightarrow \varphi$, then $\vdash^{\alpha+1}_{m+deg(\chi)} \Gamma \Rightarrow \varphi$.

Note. In principle, for the $(\rightarrow -left)$ rule the $\varphi \rightarrow \psi$ formula can be removed from the antecedent.

As is easy to see the obtained bound $2^{j(d)} \cdot l(d)$ mentioned above linearly depends on the length of the original derivation, whereas the obtained bound $2_{g(d)-1}(l(d))$ in [3], mentioned in the previous section, non-linearly depends on the length of the original derivation. A consequence of this fact that the bound on the length of the cut-free derivation given by the algorithm (defined in [6]) in terms of the original derivation is the following:

The indicated cut elimination algorithm $d \mapsto d^{cf}$ essentially differs from normalization $d \mapsto d^{nf}$ (ref [3]), in the sense that there cannot exist elementary translations from derivations in natural deductions to derivations in the sequent calculus $d \mapsto d^{seq}$ and vice versa $d \mapsto d^{nat}$, such that $d^{nf} = (((d^{seq})^{cf})^{nat}$. For then $d \mapsto d^{nf}$ would be elementary, which it is not (ref [6] - A lower bound).

Another cut-elimination method for Gentzen's sequent calculus (LK) is defined in [1]. It must be emphasized that here whole deduction system for first-order logic is considered, not only its pure implicational part as it is in [6]. First the cut-elimination is reduced to the (more general) problem of redundancy-elimination and then a resolution method is developed to handle the latter one. The first step consists in transforming a proof Φ with cuts into a cut-free proof Ψ of an extended end-sequent; more formally, it is defined a mapping T_{cut} which transforms a proof Φ of a sequent $S : \Gamma \Rightarrow \Delta$ with cut formulas $\varphi_1, \ldots, \varphi_n$ into a proof Ψ of the extended end-sequent $\forall(\varphi_1 \to \varphi_1) \wedge \cdots \wedge \forall(\varphi_n \to \varphi_n), \Gamma \Rightarrow \Delta$. This transformation (unlike "real" cut-elimination) is harmless in the sense that the time complexity is linear in size(Φ). The size of a proof Φ is defined by the number of symbol occurrences in Φ and is denoted by size(Φ). The new problem then consists in the elimination of the formula $B : \forall(\varphi_1 \to \varphi_1) \wedge \cdots \wedge \forall(\varphi_n \to \varphi_n)$ on the left-hand-side of the end-sequent. The method which is used for this purpose is more general in the sense that it eliminates also formulas B which are of different syntactical form; they only must be valid. The elimination of redundancy in proofs is performed by a resolution method. The final goal is to construct a cut-free proof of the original sequent. Note that in an intermediary step a proof of the original sequent with atomic cuts of the sequent will be obtained. In the last step the atomic cuts are eliminated and a cut-free proof is obtained. The complexity of the method is analyzed and it is shown that a non-elementary speed-up over Gentzen's method can be achieved. The length of the cut-free proof obtained by this method is

$$\leq 2^{d \cdot l(T_{cut}(\Phi)) \cdot l(\gamma)}$$

where d is an appropriate independent constant, γ is the resolution refutation of characteristic clauses $CL(T_{cut}(\Phi), \alpha)$ (ref [1]), $T_{cut}(\Phi)$ is a cut-free proof Ψ of the cut-extension of the extended end-sequent for which it is shown that

$$l(T_{cut}(\Phi)) \leq l(\Phi) + n \cdot r + k \cdot n^2$$

for a constant k independent of Φ (Φ is a proof with cut formulas $\varphi_1, \ldots, \varphi_n$) and $r = \max\{v_f(\varphi_i) \mid i = 1, \ldots, n\}$, where $v_f(\varphi_i)$ denotes the number of free variables in a formula φ_i. (If Φ is an LK proof then $l(\Phi)$ is defined as the number of sequents (i.e. nodes) occurring in Φ. $l(\Phi)$ is called the length of Φ.

More generally, the idea of the resolution method is the following: instead of showing that $\vdash \varphi_1, \ldots, \varphi_n \Rightarrow \psi$, we start with the set of sequents $(\Rightarrow \varphi_1), \ldots, (\Rightarrow \varphi_n), (\psi \Rightarrow)$ and try to prove from these the empty sequent $\Rightarrow$. The first stage is to reduce the sequents $(\Rightarrow \varphi_i)$ and the sequent $(\psi \Rightarrow)$ to clauses (sequents which contain only atomic formulas) using reductions which maintain the (un)satisfiability of the set of sequents. In other words, to start with the original set of sequents S, and using certain reductions to obtain a new set of sequents S' which is satisfiable if and only if S is satisfiable. After obtaining the clauses the question remains how to prove that the set of clauses is inconsistent (i.e. unsatisfiable), that is, how to prove the empty sequent $\Rightarrow$ from it.

5 Some refinements concerning the normalization

In this section we consider the calculus investigated in [3] which we have already observed in the "Obtained bounds for lengths of reductions" section of this paper. More precisely, in this section we make some refinements of the calculus mentioned above, by supplement of η-conversion and elimination of the $\beta_0 - Rule$ and $Variable - Rule$ from definition of the $\vdash^{\alpha}_{m} r$ relation, and show that the same results and estimates can be obtained without the use of these two rules. Thereafter we extend evaluations to first-order logic, since the investigations in [3] were done for pure implicational logic only.

From here on with $d(r)$ we denote the maximum of lengths of arbitrary reduction sequences starting from r with respect to $\rightarrow_1$, the one step reduction using β- and η-conversion rules. The calculus is defined for terms of arbitrary level.

For λ-terms r of arbitrary level the relation $\vdash^{\alpha}_{m} r$ (to be read r is derivable with reduction branch of height $\leq \alpha$ and cut-rank $\leq m$) with α, m natural numbers are defined inductively by

- ($\beta - Rule$) If $\vdash^{\alpha}_{m} r[x := s]\vec{t}$ and $\vdash^{\alpha}_{m} s$, then $\vdash^{\alpha+1}_{m} (\lambda x.r)s\vec{t}$.
- ($\eta - Rule$) If $\vdash^{\alpha}_{m} r$ and $x \notin FA(r)$, then $\vdash^{\alpha+1}_{m} \lambda x.rx$.
- ($Cut\ Rule$) If $\vdash^{\alpha}_{m} r$, $lev(r) \leq m$ and $\vdash^{\alpha}_{m} t$, then $\vdash^{\alpha+1}_{m} rt$.

The calculus allows a structural rule, i.e. if $\vdash^{\alpha}_{m} r$ and $\alpha \leq \alpha' < \omega$, $m \leq m' < \omega$, then $\vdash^{\alpha'}_{m'} r$.

Remark. Though we eliminate the *Variable Rule* used in the previous calculus, we may use its particular case ($\vdash^{\alpha}_{m} x$ for any variable x and $\alpha, m < \omega$) in spite of that elimination, because $\vdash^{\alpha}_{m} x$ by itself independently obvious, as it is evident from the meaning of $\vdash^{\alpha}_{m} r$.

We observe that $\vdash^{\alpha}_{0} r$ can be viewed as a tree which is generated in a unique way. As it is mentioned above we call this tree (with the α's stripped off) the expanded head reduction tree of r and we denote by $\#r$ its number of nodes. More precisely $\#r$ is defined by induction on $\vdash^{\alpha}_{0} r$ in the following way:

$$\begin{aligned} \#x &:= 0 \quad \text{(for any variable } x) \\ \#((\lambda x.r)s\vec{t}) &:= \#(r[x := s]\vec{t}) + 1 + \#s \\ \#(\lambda x.rx) &:= \#r + 1 \quad (\text{if } x \notin FA(r)) \end{aligned}$$

Lemma 1. $\#r = \#r[x := y]$.
Proof. The proof by induction on $\vdash^{\alpha}_{0} r$ is obvious.

Lemma 2. $\#(ry) \geq \#r$.
Proof. The proof by induction on $\vdash^{\alpha}_{0} ry$.

$$\begin{aligned} \#((\lambda x.r)s\vec{t}y) &= \#(r[x := s]\vec{t}y) + 1 + \#s \\ &\geq \#(r[x := s]\vec{t}) + 1 + \#s = \#((\lambda x.r)s\vec{t}) \\ \#((\lambda x.rx)y) &= \#(ry) + 1 + \#y = \#(ry) + 1 \\ &\geq \#r + 1 = \#(\lambda x.rx) \quad (\text{if } x \notin FA(r)) \end{aligned}$$

Main Lemma. *If $r \to_1 s$, then $\#r > \#s$.*
Proof. We show more general assertion. If $z \in FA(r)$, then

$$(\beta)\quad \#(r[z := (\lambda x.p)q]) > \#(r[z := p[x := q]])$$
$$(\eta)\quad \#(r[z := (\lambda x.px)]) > \#(r[z := p]) \quad \text{if } x \notin FA(p).$$

Let $t^* := t[z := (\lambda x.p)q]$ and $t' := t[z := p[x := q]]$ for assertion (β) and respectively $t^* := t[z := (\lambda x.px)]$ and $t' := t[z := p]$ for (η). We prove both assertions by induction on $\#r^*$.

$$\begin{aligned} \#((\lambda y.r)s\vec{t})^* &= \#(r[y := s]\vec{t})^* + 1 + \#s^* \\ &> \#(r[y := s]\vec{t})' + 1 + \#s' \\ &= \#((\lambda y.r)s\vec{t})' \end{aligned}$$

For " > " it is important that we have $z \in FA(r[y := s]\vec{t})$ or $z \in FA(s)$. This is the reason why we formulated the $\beta - Rule$ in the definition of $\vdash^{\alpha}_{m} r$ as we did.

$$\begin{aligned} \#(\lambda y.ry)^* &= \#r^* + 1 \\ &> \#r' + 1 \\ &= \#(\lambda y.ry)' \quad (\text{if } y \notin FA(r)) \end{aligned}$$

For " > " it is important that we have $z \in FA(r)$. Also we have to impose requirements $y \notin r^*$ and $y \notin r'$, since we need it for η-reductions $\lambda y.r^*y$ and $\lambda y.r'y$.

Now we undertake the cut elimination process.

Renaming Lemma. *If $\vdash^{\alpha}_{m} r$, then $\vdash^{\alpha}_{m} r[x := y]$.*
Proof. The proof by induction on $\vdash^{\alpha}_{m} r$ is obvious.

Appending Lemma. *If $\vdash^{\alpha}_{m} r$ and ry is a term, then $\vdash^{\alpha+1}_{m} ry$.*
Proof. The proof by induction on $\vdash^{\alpha}_{m} r$.

($\beta - Rule$) By induction hypothesis we have $\vdash^{\alpha+1}_{m} r[x := s]\vec{t}y$ and $\vdash^{\alpha}_{m} s$, hence $\vdash^{\alpha+2}_{m} (\lambda x.r)s\vec{t}y$ by the $\beta - Rule$.

($\eta - Rule$) The $\eta - Rule$ states: if $\vdash^{\alpha}_{m} r$ and $x \notin FA(r)$, then $\vdash^{\alpha+1}_{m} \lambda x.rx$, therefore rx is a term. Since $\vdash^{\alpha}_{m} r$ and rx is a term, then by induction hypothesis for x variable we obtain $\vdash^{\alpha+1}_{m} rx$, hence $\vdash^{\alpha+1}_{m} rx[x := y]$ by the Renaming Lemma. Note that we may use Renaming Lemma, since we have $(\lambda x.rx)y$ is a term, which means that variables x and y have the same type. Having $\vdash^{\alpha+1}_{m} y$ (according to the remark mentioned above) and $\vdash^{\alpha+1}_{m} rx[x := y]$ we apply the $\beta - Rule$ and obtain $\vdash^{\alpha+2}_{m} (\lambda x.rx)y$.

($Cut\ Rule$) We have $\vdash^{\alpha+1}_{m} rt$ and $lev(r) \le m$, hence $lev(rt) \le lev(r) \le m$. Having $\vdash^{\alpha+1}_{m} y$ (according to the remark mentioned above) we obtain $\vdash^{\alpha+2}_{m} rty$ applying the $Cut\ Rule$.

Estimate Lemma. *If $\vdash^{\alpha}_{0} r$, then $\#r \le 2^{\alpha}$.*
Proof. We show $\vdash^{\alpha}_{0} r \Rightarrow \#r \le 2^{\alpha} - 1$ by induction on $\vdash^{\alpha}_{0} r$.

($\beta - Rule$) $\#((\lambda x.r)s\vec{t}) = \#(r[x := s]\vec{t}) + 1 + \#(s) \le (2^{\alpha} - 1) + 1 + (2^{\alpha} - 1) \le 2^{\alpha+1} - 1$.

($\eta - Rule$) $\#(\lambda x.rx) = \#r + 1 \le (2^{\alpha} - 1) + 1 \le 2^{\alpha+1} - 1$.

Let us notice that the Estimate Lemma does not depend on the arity $ar(r)$ of the term r.

Substitution Lemma. *If* $\vdash_m^\alpha r$ *and* $\vdash_m^\beta s_j$ *and* $lev(s_j) \leq m,\ j = 1, \ldots, k$, *then* $\vdash_m^{\alpha+\beta} r[\vec{y} := \vec{s}]$.

Proof. The proof by induction on $\vdash_m^\alpha r$. We write t^* for $t[\vec{y} := \vec{s}]$.

($\beta - Rule$) By induction hypothesis we have $\vdash_m^{\alpha+\beta} r^*[x := s^*]\vec{t^*}$ and $\vdash_m^{\alpha+\beta} s^*$, hence $\vdash_m^{\alpha+\beta+1} (\lambda x.r^*)s^*\vec{t^*}$ by the $\beta - Rule$.

($\eta - Rule$) By induction hypothesis we have $\vdash_m^{\alpha+\beta} r^*$, hence $\vdash_m^{\alpha+\beta+1} (\lambda x.r^*x)$ by the $\eta - Rule$. The proof is indifferent to the fact whether $x \in \vec{y}$ or no, because $x \notin FA(r)$. An important requirement is $x \notin FA(s_j),\ j = 1, \ldots, k$, which can be obtained renaming x to unused variable z, i.e. $z \notin FA(s_j),\ j = 1, \ldots, k$.

($Cut - Rule$) By induction hypothesis we have $\vdash_m^{\alpha+\beta} r^*$ and $\vdash_m^{\alpha+\beta} t^*$ and $lev(r^*) \leq m$, thus $\vdash_m^{\alpha+\beta+1} r^*t^*$ by the $Cut - Rule$.

Cut Elimination Lemma. *If* $\vdash_{m+1}^\alpha r$, *then* $\vdash_m^{2^\alpha} r$.
Proof. We show $\vdash_{m+1}^\alpha r \Rightarrow \vdash_m^{2^\alpha - 1} r$ by induction on $\vdash_{m+1}^\alpha r$.

($\beta - Rule$) By induction hypothesis we have $\vdash_m^{2^\alpha-1} r[x := s]\vec{t}$ and $\vdash_m^{2^\alpha-1} s$, hence $\vdash_m^{2^\alpha} (\lambda x.r)s\vec{t}$ by the $\beta - Rule$ and $2^\alpha \leq 2^{\alpha+1} - 1$.

($\eta - Rule$) By induction hypothesis we have $\vdash_m^{2^\alpha-1} r$, hence $\vdash_m^{2^\alpha} (\lambda x.rx)$ by the $\eta - Rule$ and $2^\alpha \leq 2^{\alpha+1} - 1$.

($Cut - Rule$) By induction hypothesis we have $\vdash_m^{2^\alpha-1} r, \vdash_m^{2^\alpha-1} t$ and $lev(r) \leq m+1$, hence $lev(t) \leq m$. By the Appending Lemma we obtain $\vdash_m^{2^\alpha} ry$, thus $\vdash_m^{2^{\alpha+1}-1} rt$ by the Substitution Lemma.

Embedding Lemma. $g(r) \leq m+1$ *implies* $\vdash_m^{l(r)} r$.
Proof. We show $g(r) \leq m+1 \Rightarrow \vdash_m^{l(r)-1} r$ by induction on r.

(Case x). By the remark we obtain $\vdash_m^{l(x)-1} x$.

(Case λx.r). By induction hypothesis $\vdash_m^{l(r)-1} r$. It is obvious that the terms $\lambda x.r$ and r have the same number of β-redexes, thus $\vdash_m^{l(r)-1} (\lambda x.r)$, so $\vdash_m^{l(\lambda x.r)-1} (\lambda x.r)$. With regard to the η-redexes the term $\lambda x.r$ can contain the same number or plus one of η-redexes as compared with the term r, thus $\vdash_m^{l(r)} (\lambda x.r)$, so $\vdash_m^{l(\lambda x.r)-1} (\lambda x.r)$.

(Case ts). By induction hypothesis $\vdash_m^{l(t)-1} t$ and $\vdash_m^{l(s)-1} s$, thus the Appending Lemma yields $\vdash_m^{l(t)} ty$. Since $lev(t) \leq m+1$ we have $lev(s) \leq m$, hence $\vdash_m^{l(t)+l(s)-1} ts$ by the Substitution Lemma, so $\vdash_m^{l(ts)-1} ts$.

With the Embedding Lemma and the Cut Elimination Lemma it follows that the expanded head reduction tree of r with $g(r) > 0$ has the

$$height \leq 2_{g(r)-1}(l(r)).$$

The Estimate Lemma now shows

$$\#r \leq 2^{2_{g(r)-1}(l(r))} = 2_{g(r)}(l(r)).$$

Together with the Main Lemma this yields

$$d(r) \leq \#r \leq 2_{g(r)}(l(r)).$$

The use of the method of collapsing types to transfer the result concerning strong normalization (that is, any derivation r in implicational logic is strongly normalizable) from implicational logic to first-order logic is illustrated in [4]. Then the result is improved by a complement, which states:

Theorem 4. *Any derivation r in first-order logic is strongly normalizable. Moreover, the length of reduction sequence to obtain normal form of r is equal to the length of reduction sequence to obtain normal form of r^c, in other words, the same number of one-step reductions is needed to bring r and r^c to their normal forms.*

Let us notice that r^c is a term in implicational logic and it is called the collapse of the term r (ref [4]).

Now using the theorem 4 we can extend the result obtained in this section (the upper bound for the length of arbitrary reduction sequences for pure implicational logic) to include first-order logic. So we obtain, that in first-order logic any reduction sequence (by means of β- and η-conversions) for a term r is bounded by

$$2_{g(r^c)}(l(r^c)),$$

where r^c is the collapse of the term r, $l(r^c)$ and $g(r^c)$ are the length and degree of the term r^c respectively.

References

[1] Baaz, M., Leitsch, A.: Cut-Elimination and Redundancy-Elimination by Resolution. Journal of Symbolic Computation, 29:149–176, 2000.

[2] Beckmann, A., Weiermann, A.: Analyzing Gödel's T via expanded head reduction trees. Mathematical Logics Quarterly, 46:517–536, 2000.

[3] Beckmann, A.: Exact bounds for lengths of reductions in typed lambda-calculus. J. Symbolic Logic, 66:1277-1285, 2001.

[4] Galoyan, T.: Strong normalization for first-order logic. Transactions of the Institute for Informatics and Automation Problems of the NAS of RA. Mathematical Problems of Computer Science 28, pp. 45-50, 2007.

[5] Kleene, S.: Introduction to Metamathematics. D. van Nostrand, New York, 1952.

[6] Schwichtenberg, H.: Normalization. In F.L. Bauer, editor, Logic, Algebra and Computation. Proceedings of the International Summer School Marktoberdorf, Germany, July 25 - August 6, 1989, Series F: Computer and Systems Sciences, Vol. 79, pages 201-237. NATO Advanced Study Institute, Springer Verlag, Berlin, Heidelberg, New York, 1991.

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

[8] Schwichtenberg, H.: Proof theory. Lecture notes, Mathematisches Institut der Universität München, Germany, 1994.

Ռեդուկցիոն հաջորդականությունների երկարության սահմանների վերլուծությունը տիպականացված λ-հաշվում

Տ. Մ. Գալոյան

Ամփոփում

Աշխատանքում ուսումնասիրվում են ռեդուկցիոն հաջորդականությունների երկարության գնահատականները տիպականացված λ-հաշվի թերմերի համար։ Դիտարկվում և համեմատվում են այլ հեղինակների կողմից տրված գնահատականները։ Ձևափոխվում է [3]-ում տրված գնահատականի ապացույցը (մաքուր իմպլիկացիոն տրամաբանության համար)՝ ընդգրկելով նաև η-ռեդուկցիայի կանոնը։ Այնուհետև ստացված արդյունքը ընդհանրացվում է առաջին կարգի տրամաբանության համար։ Աշխատանքում հետազոտվում են նաև կրճատման գործողության արտաքսման և նորմալացման ալգորիթմները։