STRONG NORMALIZATION OF PROGRAM-INDEXED LAMBDA CALCULUS

Abstract
A program-indexed typed λ-calculus, λ_{DL}, is introduced. λ_{DL} has a Curry-Howard correspondence with an intuitionistic variant of dynamic logics. The strong normalization theorem for λ_{DL} is proved.

1. Introduction

Dynamic logics (DLs) are known to be useful logics for program analysis, and have widely been studied by many researchers [5]. Natural deduction (ND) systems for DLs have been required to obtain good computational interpretations and practical applications. Some sound and complete ND-systems for DLs over the first-order language of Peano arithmetic were introduced by Honsell and Miculan [6]. As mentioned in [6], this approach was inspired by an unpublished paper by Stirling, where a ND-system for deterministic DL was sketched. The approach by Honsell and Miculan was motivated to give adequate encodings of DL and Hoare logic in an interactive proof development environment, Coq. In [6], some completeness and encoding results were obtained for two ND-systems S_{ND}(DL) and

*This paper is based on the conference presentation [7]. I would like to thank the referee for his or her valuable comments. By the comments, some errors in the previous version are corrected. This research was supported by the Alexander von Humboldt Foundation and the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (B) 20700015.
S_{ND}^D(DL). However, (strong) normalization and Curry-Howard correspondence for such a system have not yet been studied. Indeed, obtaining a Curry-Howard correspondence with DLs was remained as an open question [6]. The present paper provides a partial solution to this open question, investigating a modified fragment of S_{ND}(DL) and S_{aND}^D(DL).

The aim of this paper is then to obtain a strongly normalizable typed \( \lambda \)-calculus (and ND-system) for an intuitionistic DL with a Curry-Howard correspondence. To obtain such a framework, the logical and program inference rules in S_{ND}(DL) and S_{aND}^D(DL) are generalized and restricted. Firstly, the logical and program inference rules are indexed by a (possibly empty) sequence \([d_0][d_1] \cdots [d_n]\) of atomic program modalities. This indexing gives us a natural generalization of the standard simply typed \( \lambda \)-calculus, and allows us to obtain a simple strong normalization proof. Secondly, the infinitary inference rule for the program iteration operator \( \ast \):

\[
\begin{array}{c}
\{ [b; b; \cdots; b] \alpha \mid j \in \omega \} \\
[\cdot\ast\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\...
with demonic non-deterministic program operators. A dynamic \( \lambda \)-calculus was introduced by Kohlhase and Kuschert [8] in order to obtain a logical foundation of discourse theories. This calculus is, however, different from the standard approach to DLs in the sense that it does not use the program operators. Although as mentioned above, there are few results for ND-systems for DLs, there are some results for ND-systems for linear-time temporal logics (LTLs). For example, a judgment ND-system for a fragment of LTLs was introduced by Davies and Pfenning [2] in order to analyze staged computation in programming applications. A strongly normalizable ND-system for the until-free fragment of LTL was studied by Baratella and Masini [1] from the point of view of pure mathematical logic.

2. Program-indexed ND-system

Formulas are constructed from (countable) propositional variables, \( \rightarrow \) (implication), \( \land \) (conjunction), and \( [b] \) (program modal operator) where \( b \) is a program. Programs are constructed from (countable) atomic programs, \( \cup \) (non-deterministic choice), \( ; \) (composition) and \( * \) (finite iteration). Lower-case letters \( b, c, \ldots \) are used for programs and Greek lower-case letters \( \alpha, \beta, \ldots \) are used for formulas. The symbol \( \omega \) is used to represent the set of natural numbers. The symbol \( \omega_l \) where \( l \) is a fixed positive integer is used to represent the set \( \{ i \in \omega \mid i \leq l \} \). Lower-case letters \( i, j, \ldots \) are sometimes used for any natural numbers. For a program \( b \), an expression \( b^i \) with \( i \in \omega \) is defined inductively by \( b^0 \equiv \emptyset \) and \( b^{i+1} \equiv b^i ; b \). An expression \( [\emptyset]\alpha \) means \( \alpha \), and expressions \( [\emptyset ; b]\alpha \) and \( [b ; \emptyset]\alpha \) mean \( [b]\alpha \).

**Definition 1.** Formulas and programs are defined by the following grammar, assuming \( p \) and \( e \) represent propositional variables and atomic programs, respectively:

\[
\begin{align*}
\alpha &::= p \mid \alpha \rightarrow \alpha \mid \alpha \land \alpha \mid [b]\alpha \\
b &::= e \mid b \cup b \mid b ; b \mid b^n
\end{align*}
\]

The symbol APR is used to represent the set of all atomic programs including the empty program \( \emptyset \). An expression \( [d] \) is used to represent \( [d_0][d_1] \cdots [d_i] \) with \( i \in \omega, d_i \in \text{APR} \) and \( d_0 \equiv \emptyset \), i.e., \( [d] \) can be the empty sequence. Also, an expression \( \hat{d} \) is used to represent \( d_0 ; d_1 ; \cdots ; d_i \) with \( i \in \omega, d_i \in \text{APR} \) and \( d_0 \equiv \emptyset \).
Definition 2 (N\textsubscript{DL}). Let \( l \) be a fixed positive integer.

The inference rules of N\textsubscript{DL} are of the form: for any \( k \in \omega \) and any programs \( b \) and \( c \),

\[
\begin{array}{c}
\frac{[\hat{d}]\alpha}{[\hat{d}]\beta} (\rightarrow I) \quad \frac{[\hat{d}]\alpha \rightarrow \beta}{[\hat{d}]\alpha} (\rightarrow E) \\
\frac{[\hat{d}]\alpha_1 \land \alpha_2}{[\hat{d}]\alpha_1 \land \alpha_2} (\land I) \quad \frac{[\hat{d}]\alpha_1 \land \alpha_2}{[\hat{d}]\alpha_1} (\land E_1) \quad \frac{[\hat{d}]\alpha_1 \land \alpha_2}{[\hat{d}]\alpha_2} (\land E_2) \\
\frac{[\hat{d}]b \alpha}{[\hat{d}]b \cup c \alpha} (\cup I) \quad \frac{[\hat{d}]b \cup c \alpha}{[\hat{d}]b \alpha} (\cup E_1) \quad \frac{[\hat{d}]b \cup c \alpha}{[\hat{d}]c \alpha} (\cup E_2) \\
\frac{[\hat{d}]b \alpha}{[\hat{d}]b \alpha} (; I) \quad \frac{[\hat{d}]b \alpha}{[\hat{d}]b \alpha} (; E) \quad \frac{[\hat{d}]b \alpha}{\cup \{ [\hat{d}]b \alpha \}_{j \in \omega}} (\ast I) \quad \frac{[\hat{d}]b \alpha}{\cup \{ [\hat{d}]b \alpha \}_{j \in \omega}} (\ast E).
\end{array}
\]

Remark that \((\ast I)\) has \((l + 1)\)-premises, and the \( k \) appeared in \((\ast E)\) is in \( \omega \). This means that the formula of the form \([b^\ast \alpha] \) is interpreted as \( \alpha \land [b] \alpha \land [b^2] \alpha \land \cdots \land [b^l] \alpha \). If the unbounded versions of \((\ast I)\) and \((\ast E)\) by replacing \( \omega \) by \( \omega \) are assumed, then the underlying system N\textsubscript{DL} is more natural than N\textsubscript{DL}. But, the strong normalization result for N\textsubscript{DL} has not yet been obtained, and hence our result for N\textsubscript{DL} is regarded as a partial result.

Despite the restriction on the iteration number, N\textsubscript{DL} can derive some typical axioms for DLs, such as a program induction axiom. N\textsubscript{DL} allows us to obtain a simple treatment of the strong normalization and Curry-Howard correspondence. Such theoretical merits may not be obtained easily for N\textsubscript{DL}, because the unbounded iteration operator requires an infinite inference rule. To show the strong normalization theorem for N\textsubscript{DL}, handling a kind of infinite terms may be needed. Restricting the iteration number may imply not only the theoretical merits as mentioned above, but also some practical merits for describing finitely executable programs (i.e., non reactive programs) since there are issues in computer science where only a finite program sequence is of interest.
Strictly speaking, $N_{DL}$ is just the logic parameterized by a fixed concrete positive integer $l$, and hence, such an $l$-parameterized logic should precisely be denoted as e.g., $N_{DL[l]}$. But, since we don’t need to specify such an integer $l$ in the following discussion, we will use the abstract name “$N_{DL}$” instead of the concrete “$N_{DL[l]}$”. We also use the same positive integer $l$ through the following discussion.

The terminologies of the standard natural deduction system are used. The notions of proof (in $N_{DL}$), open and discharged assumptions of proof, and end-formula of proof are defined as usual. A formula $\alpha$ is said to be provable in $N_{DL}$ if there exists a proof in $N_{DL}$ with no open assumption whose end-formula is $\alpha$.

Although the reduction relation on the set of proofs in $N_{DL}$ can naturally be defined and the strong normalization theorem for $N_{DL}$ can also be shown, such a discussion is omitted here, since the strong normalization theorem will be proved for the corresponding typed $\lambda$-calculus $\lambda_{DL}$. By the Curry-Howard correspondence, the strong normalization theorem for $N_{DL}$ can be obtained from that of $\lambda_{DL}$.

The following proposition means that some typical axioms of DL are provable in $N_{DL}$. An expression $\alpha \leftrightarrow \beta$ represents the formula $(\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha)$.

**Proposition 3.** The following formulas are provable in $N_{DL}$: for any formula $\alpha$ and any programs $b, c$,

1. $[b \cup c] \alpha \leftrightarrow [b] \alpha \land [c] \alpha$,
2. $[b : c] \alpha \leftrightarrow [b] [c] \alpha$,
3. $[b^*] \alpha \rightarrow \alpha$,
4. $[b^*] \alpha \rightarrow [b] \alpha$,
5. $(\alpha \land [b^*] (\alpha \rightarrow [b] \alpha)) \rightarrow [b^*] \alpha$ (program induction),
6. $[b^*] \alpha \leftrightarrow \alpha \land [b] \alpha \land [b^2] \alpha \land \cdots \land [b^l] \alpha$ (finite iteration).

**Proposition 4.** The rule of the form: for any formula $\alpha$ and any program $b$,

$$
\begin{array}{c}
\alpha \\
[b] \alpha
\end{array}
$$

is admissible in $N_{DL}$.
Proof. (Sketch). Firstly, we show that the restricted rule (atomic-regu) which is obtained from (regu) by replacing a program \( b \) by an atomic program \( e \) is admissible in \( N_{DL} \). This fact can be proved by induction on the proofs of \( \alpha \). Next, we show that (regu) is admissible in \( N_{DL} \) by induction on a complexity \( \text{comp}(b) \) of \( b \) where \( \text{comp}(c^*) \) must be defined as (*): \( \text{comp}(c^*) \geq \text{comp}(c^j) \) for any \( j \in \omega_i \). In the case \( b \equiv e \) with \( e \in \text{APR} \), we use (atomic-regu), and in the case \( b \equiv c^* \), we use the property (*). Q.E.D.

3. Program-indexed typed \( \lambda \)-calculus

Terms are constructed from variables, a \( \lambda \)-abstraction \( \lambda \), an application operator \( \cdot \) (it is always omitted), a pairing function \( \langle, \rangle \), an \((l + 1)\)-ary pairing function \( \langle, \ldots, \rangle \), projection functions \( \pi_1, \pi_2, \ldots, \pi_{l+1} \) and two new constructors \( \iota \) and \( \iota^{-1} \). The intended meaning of \( \iota \) and \( \iota^{-1} \) is presented as the equations:

1. \( (\iota^{-1}eM[b,c][\alpha][b,c][\alpha]) = M[b,c][\alpha] \),
2. \( (\iota e^{-1}M[b,c][\alpha][b,c][\alpha]) = M[b,c][\alpha] \).

(Untyped) terms are defined as usual, and types are defined as the same way as in Definition 1, replacing “formulas” by “types”. Variables are denoted as \( x, x_n, y, \ldots, \), untyped terms are denoted as \( M, M_n, N, \ldots, \), types are denoted as \( \alpha, \beta, \gamma, \ldots, \), typed terms are denoted as \( M^\alpha, N^\beta, L^\gamma, \ldots, \). Typed terms are sometimes denoted as \( M, N, L, \ldots, \) by omitting the types. It is assumed that in a \( \lambda \)-term, the same variables do not occur simultaneously as both free and bound variables. It is also assumed that in a \( \lambda \)-term, there are no iterated occurrences of the same bound variable \( x \), such as \( \cdots \lambda x^\alpha, (\cdots \lambda x^\alpha, \cdots) \cdots \). An expression \( [N^\alpha/x^\alpha]M^\beta \) means, in a usual sense, the substitution of \( N^\alpha \) to a free variable \( x^\alpha \) in \( M^\beta \).

For the new constructor \( \iota' \in \{\iota, \iota^{-1}\} \), we also assume the condition \( [N^\alpha/x^\alpha](\iota'M^\beta)^\gamma = (\iota'[N^\alpha/x^\alpha]M^\beta)^\gamma \). To avoid the clash of bound variables by substitutions, \( \alpha \)-conversions are occasionally assumed.

Definition 5. The notion of degree \( d(b) \) of a program \( b \) is defined by:

1. \( d(c) = 1 \) for any atomic program \( c \),
2. \( d(b \cup c) = d(b) + d(c) + 1 \),
3. \( d(b^*) = (l \times d(b)) + l \).
The notion degree $d(\alpha)$ of a type $\alpha$ is defined by:

1. $d(p) = 1$ for any atomic type $p$,
2. $d(\alpha \circ \beta) = d(\alpha) + d(\beta) + 1$ where $\circ \in \{\rightarrow, \wedge\}$,
3. $d([b]\alpha) = d(b) + d(\alpha)$.

For example, we have $d([e ; e]p) = 4$, $d([e]e)p = 3$, $d(e^*) = 2 \times l - 1$ and $d(e^l) = (2 \times l) - 1$ where $e$ and $p$ are an atomic program and an atomic type, respectively. Remark that $d([b ; c]\alpha) > d([b]c\alpha)$ and $d([b^*]\alpha) > d([b^*]\alpha)$ for any programs $b, c$ and any type $\alpha$.

**DEFINITION 6.** Typed $\lambda$-terms for $\lambda_{DL}$ are defined as follows.

1. if $x[\alpha]d$ is a typed variable, then it is a typed $\lambda$-term.
2. if $x[\alpha]d$ is a typed variable and $M[\beta]d$ is a typed $\lambda$-term, then $(\lambda x[\alpha]d.M[\beta]d)(\alpha \rightarrow \beta)$ is a typed $\lambda$-term.
3. if $M[\alpha]d$ and $N[\beta]d$ are typed $\lambda$-terms, then $(M[\alpha]d(N[\beta]d))(\alpha \rightarrow \beta)$ is a typed $\lambda$-term.
4. if $M[\alpha]d$ and $N[\beta]d$ are typed $\lambda$-terms, then $(\lambda x[\alpha]d.N[\beta]d)(\alpha \wedge \beta)$ is a typed $\lambda$-term.
5. if $M[\alpha]d$ and $N[\beta]d$ are typed $\lambda$-terms, then $(\pi_1 M[\alpha]d, \pi_2 M[\beta]d)(\alpha \wedge \beta)$ are typed $\lambda$-terms.
6. if $M[\alpha]d$ and $N[\beta]d$ are typed $\lambda$-terms, then $(\lambda x[\alpha]d, N[\beta]d)(\alpha \wedge \beta)$ is a typed $\lambda$-term.
7. if $M[\alpha]d$ and $N[\beta]d$ are typed $\lambda$-terms, then $(\pi_1 M[\alpha]d, \pi_2 M[\beta]d)((\alpha \wedge \beta)\beta)$ are typed $\lambda$-terms.
8. if $M[\alpha]d$ and $N[\beta]d$ are typed $\lambda$-terms, then $(\lambda x[\alpha]d, N[\beta]d)((\alpha \wedge \beta)\beta)$ is a typed $\lambda$-term.
9. if $M[\alpha]d$ and $N[\beta]d$ are typed $\lambda$-terms, then $(\lambda x[\alpha]d, N[\beta]d)((\alpha \wedge \beta)\beta)$ is a typed $\lambda$-term.
10. if $M[\alpha]d$ and $N[\beta]d$ are typed $\lambda$-terms, then $(\pi_1 M[\alpha]d, \pi_2 M[\beta]d)((\alpha \wedge \beta)\beta)$ are typed $\lambda$-terms.
11. if $M[\alpha]d$ and $N[\beta]d$ are typed $\lambda$-terms, then $(\lambda x[\alpha]d, N[\beta]d)((\alpha \wedge \beta)\beta)$ is a typed $\lambda$-term.

**DEFINITION 7 (\lambda_{DL}).** The typed $\lambda$-calculus $\lambda_{DL}$ is defined by reductions for the typed $\lambda$-terms defined in Definition 6. In the following, the transformation process from the left hand side of $\rightarrow$ to the right hand side of $\rightarrow$ is called a reduction, and the term of the left hand side of $\rightarrow$ is called a redex.

1. $(\lambda x[\alpha]d.M[\beta]d)(\alpha \rightarrow \beta).N[\alpha]d \vdash \beta \rightarrow (N[\alpha]d.M[\beta]d)$.
2. $(\lambda x[\alpha]d.N[\beta]d)(\alpha \wedge \beta) \vdash \beta \rightarrow (N[\alpha]d.M[\beta]d)$.
3. \((\pi_3(M[\hat{d}]^\alpha, N[\hat{d}]^\beta)) \equiv N[\hat{d}]^\beta\).
4. \((\pi_1(M[\hat{d}]^\alpha, M[\hat{d}]^\gamma)) \equiv M[\hat{d}]^\alpha\).
5. \((\pi_2(M[\hat{d}]^\alpha, N[\hat{d}]^\gamma)) \equiv N[\hat{d}]^\gamma\).
6. \((\iota^{-1}(M[\hat{d}]^\alpha)[\iota^{-1}e]\hat{d})[\iota^{-1}e]\alpha \equiv M[\hat{d}]^\alpha\).
7. \((\pi_k(M[\hat{d}]^\alpha, M[\hat{d}]^\beta, ..., M[\hat{d}]^\gamma)) \equiv M[\hat{d}]^\gamma\).

8. (Compatible closure): if \(M \triangleright N\), then \(\lambda x.M \triangleright \lambda x.N\), \(ML \triangleright NL\), \(LM \triangleright LN\), \(\langle M, L \rangle \triangleright \langle N, L \rangle\), \(\langle L, M \rangle \triangleright \langle L, N \rangle\), \(\langle ..., M, ... \rangle \triangleright \langle ..., N, ... \rangle\), \(\pi_k M \triangleright \pi_k N\) with \(1 \leq k \in \omega_{\alpha+1} \) and \(\iota M \triangleright \iota N\) and \(\iota^{-1} M \triangleright \iota^{-1} N\).

Roughly speaking, the items 4–7 in Definition 7 correspond to the items 1, 2 and 6 in Proposition 3.

4. Strong normalization

**Definition 8.** A typed \(\lambda\)-term is said to be normal if it contains no redex. A sequence \(M_0^\alpha, M_1^\alpha, \ldots\) of typed \(\lambda\)-terms is called a reduction sequence if it satisfies the following conditions (1) \(M_i^\alpha \triangleright M_{i+1}^\alpha\) for all \(0 \leq i\) and (2) the last typed \(\lambda\)-term in the sequence is normal if the sequence is finite. A typed \(\lambda\)-term \(M^\alpha\) is called strongly normalizable if each reduction sequence starting from \(M^\alpha\) is terminated.

We now start to prove the strong normalization theorem for \(\lambda_{DL}\), using the method presented in [4]. In the following, SN means the set of all strongly normalizable typed \(\lambda\)-terms for \(\lambda_{DL}\), and TERM means the set of all typed \(\lambda\)-terms for \(\lambda_{DL}\). In order to show \(\text{TERM} \subseteq \text{SN}\) (i.e., the strong normalization theorem), we will define the set RED of reducible terms, and will show \(\text{TERM} \subseteq \text{RED} \subseteq \text{SN}\). First, we will show \(\text{RED} \subseteq \text{SN}\) by induction on the degree of a type, and second, will show \(\text{TERM} \subseteq \text{RED}\) by induction on the construction of a term.

**Definition 9.** The set RED of reducible terms of type \(\gamma\) (for \(\lambda_{DL}\)) is defined by induction on the (degree of) type \(\gamma\) as follows.

1. For any atomic type \(p\), \(M[\hat{d}]^p \in \text{RED}_{\hat{d}}\) iff \(M[\hat{d}]^p \in \text{SN}\).
2. \(\gamma[\hat{d}]^\alpha \rightarrow \beta \in \text{RED}_{\hat{d}}\) iff \(\forall N[\hat{d}]^\alpha \in \text{RED}_{\hat{d}}\) \(\gamma[\hat{d}]^\alpha \rightarrow \beta \in \text{RED}_{\hat{d}}\).
3. \(M^{[\alpha]}(\alpha \land \beta) \in \text{RED}_{\bar{d}[\alpha \land \beta]} \iff (\pi_1 M^{[\bar{d}]}(\alpha \land \beta))^{\bar{d}[\alpha]} \in \text{RED}_{\bar{d}[\alpha]} \) and 
\( (\pi_2 M^{[\bar{d}]}(\alpha \land \beta))^{\bar{d}[\beta]} \in \text{RED}_{\bar{d}[\beta]} \).

4. \(M^{[\bar{d}]}[b,c]^{\alpha} \in \text{RED}_{\bar{d}[b,c]} \alpha \iff (\pi_1 M^{[\bar{d}]}[b,c]^{\alpha})^{\bar{d}[b]^{\alpha}} \in \text{RED}_{\bar{d}[b]^{\alpha}} \) and 
\( (\pi_2 M^{[\bar{d}]}[b,c]^{\alpha})^{\bar{d}[c]^{\alpha}} \in \text{RED}_{\bar{d}[c]^{\alpha}} \).

5. \(M^{[\bar{d}]}[b]^{\alpha} : c^{\alpha} \in \text{RED}_{\bar{d}[b]}[b]^{\alpha} c^{\alpha} \iff (\iota^{-1} M^{[\bar{d}]}[b]^{\alpha} c^{\alpha})^{\bar{d}[b]^{\alpha}} \in \text{RED}_{\bar{d}[b]^{\alpha}} c^{\alpha} \).

6. \(M^{[\bar{d}]}[\alpha]^{\alpha} \in \text{RED}_{\bar{d}[\alpha]} \alpha \iff (\pi N M^{[\bar{d}]}[\alpha]^{\alpha})^{\bar{d}[\alpha]^{\alpha} k^{-1}} \in \text{RED}_{\bar{d}[\alpha]^{\alpha} k^{-1}} \) for all \(k\) with \(1 \leq k \in \omega_{\omega+1}\).

**Definition 10.** A typed \(\lambda\)-term \(M^{\alpha}\) for \(\lambda_{DL}\) is said to be neutral if \(M\) is one of the forms \(x, NP, \pi_k N\) with \(1 \leq k \in \omega_{\omega+1}\), and \(\iota^{-1} N\).

If \(M^{\alpha} \in \text{SN}\), then an expression \(v(M^{\alpha})\) means the least number which bounds the length of every reduction sequence beginning with \(M^{\alpha}\).

**Lemma 11.** For all typed \(\lambda\)-term \(M^{\alpha}\) for \(\lambda_{DL}\), \(M^{\alpha}\) satisfies the following four conditions.

- **CR1.** if \(M^{\alpha} \in \text{RED}_{\alpha}\), then \(M^{\alpha} \in \text{SN}\); 
- **CR2.** if \(M^{\alpha} \in \text{RED}_{\alpha}\) and \(M^{\alpha} \succ N^{\alpha}\), then \(N^{\alpha} \in \text{RED}_{\alpha}\); 
- **CR3.** if \(M^{\alpha}\) is neutral, then \(\forall N^{\alpha} \,[\text{if } M^{\alpha} \succ N^{\alpha} \text{ and } N^{\alpha} \in \text{RED}_{\alpha}\), 
then \(M^{\alpha} \in \text{RED}_{\alpha}\)].
- **CR4.** if \(M^{\alpha}\) is neutral and normal, then \(M^{\alpha} \in \text{RED}_{\alpha}\). Remark that (CR4) is a special case of (CR3).

**Proof.** By induction on the degree \(d(\alpha)\) of the type \(\alpha\). The cases \(\alpha \equiv [\bar{d}]_{p}\) (p: atomic), \(\alpha \equiv [\bar{d}](\beta \rightarrow \gamma)\) and \(\alpha \equiv [\bar{d}](\beta \land \gamma)\) can be proved in a similar way as in [4]. We show only the cases \(\alpha \equiv [\bar{d}]_{b}^{\alpha} c^{\beta}\) and \(\alpha \equiv [\bar{d}][b]^{\alpha} \beta\).

- **Case** \((\alpha \equiv [\bar{d}]_{b}^{\alpha} c^{\beta}\).  
  - **(CR1):** Suppose that \(M^{[\bar{d}]}_{b}^{\alpha} c^{\beta} \in \text{RED}_{\bar{d}[b]^{\alpha} c^{\beta}}\). Then, 
  \((\iota^{-1} M^{[\bar{d}]}_{b}^{\alpha} c^{\beta})^{\bar{d}[b]^{\alpha} c^{\beta}} \in \text{RED}_{\bar{d}[b]^{\alpha} c^{\beta}}\) by the definition of RED. 
  
  By the induction hypothesis of (CR1) with \(d([\bar{d}]_{b}^{\alpha} c^{\beta}) \succ d([\bar{d}]_{b}^{\alpha} c^{\beta})\), we obtain \((\iota^{-1} M^{[\bar{d}]}_{b}^{\alpha} c^{\beta})^{\bar{d}[b]^{\alpha} c^{\beta}} \in \text{SN}\). Moreover, we have 
  \(v((\iota^{-1} M^{[\bar{d}]}_{b}^{\alpha} c^{\beta})^{\bar{d}[b]^{\alpha} c^{\beta}}) \geq v(M^{[\bar{d}]}_{b}^{\alpha} c^{\beta})\), because from any reduction sequence \(M^{[\bar{d}]}_{b}^{\alpha} c^{\beta} \succ M^{[\bar{d}]}_{b}^{\alpha} c^{\beta} \succ M^{[\bar{d}]}_{b}^{\alpha} c^{\beta} \succ \cdots\), one can construct a reduction sequence \((\iota^{-1} M^{[\bar{d}]}_{b}^{\alpha} c^{\beta})^{\bar{d}[b]^{\alpha} c^{\beta}} \succ (\iota^{-1} M^{[\bar{d}]}_{b}^{\alpha} c^{\beta})^{\bar{d}[b]^{\alpha} c^{\beta}}\)
\[ \leadsto (\lambda_{\beta_1} M_1[b; c] b_\beta) \quad \ldots. \] So \( v(M_1[b; c] b_\beta) \) is finite, and hence \( M_1[b; c] b_\beta \in \text{SN} \).

(CR2): Suppose that \( M_1[b; c] b_\beta \succ N_1[b; c] b_\beta \). Then,
\[ (\lambda_{\beta_1} M_1[b; c] b_\beta) \succ (\lambda_{\beta_1} N_1[b; c] b_\beta) \]. By the hypothesis, we have \( M_1[b; c] b_\beta \in \text{RED}_{[\beta]}[b; c] b_\beta \), and hence \( (\lambda_{\beta_1} M_1[b; c] b_\beta) \succ (\lambda_{\beta_1} N_1[b; c] b_\beta) \) by the definition of \( \text{RED} \). By the induction hypothesis of (CR2) with \( d([\beta] b; c) \beta \succ d([\beta] b; c) \beta \), we obtain \( (\lambda_{\beta_1} N_1[b; c] b_\beta) \succ (\lambda_{\beta_1} N_1[b; c] b_\beta) \in \text{RED}_{[\beta]}[b; c] b_\beta \). Since \( M_1[b; c] b_\beta \) is neutral,
\[ (\lambda_{\beta_1} N_1[b; c] b_\beta) \succ (\lambda_{\beta_1} N_1[b; c] b_\beta) \] cannot itself be a redex. We thus obtain
\[ (\lambda_{\beta_1} M_1[b; c] b_\beta) \succ (\lambda_{\beta_1} N_1[b; c] b_\beta) \] and \( (\lambda_{\beta_1} N_1[b; c] b_\beta) \in \text{RED}_{[\beta]}[b; c] b_\beta \) because of the hypothesis \( N_1[b; c] b_\beta \in \text{RED}_{[\beta]}[b; c] b_\beta \) and the definition of \( \text{RED} \). Since \( \lambda_{\beta_1} M_1[b; c] b_\beta \) is neutral and all the typed \( \lambda \)-terms one step from \( \lambda_{\beta_1} M_1[b; c] b_\beta \) are in \( \text{RED}_{[\beta]}[b; c] b_\beta \), we can apply the induction hypothesis of (CR3) with \( d([\beta] b; c) \beta \succ d([\beta] b; c) \beta \), and obtain \( (\lambda_{\beta_1} M_1[b; c] b_\beta) \in \text{RED}_{[\beta]}[b; c] b_\beta \). Therefore we obtain \( M_1[b; c] b_\beta \in \text{RED}_{[\beta]}[b; c] b_\beta \) by the definition of \( \text{RED} \).

• Case (\( \alpha \equiv [\beta] b_\beta \)).

(CR1): Suppose \( M_1[b_\beta] b_\beta \in \text{RED}_{[\beta]}[b_\beta] b_\beta \). Then, by the definition of \( \text{RED} \), \( (\pi_k M_1[b_\beta] b_\beta) \in \text{RED}_{[\beta]}[b_\beta] b_\beta \) for all \( k \) with \( 1 \leq k < \omega_{\beta+1} \).

We have \( k = 1 \) and \( d([\beta] b_\beta) b_\beta = d([\beta] b_\beta) b_\beta \). Hence we can apply the induction hypothesis of (CR1), and obtain \( (\pi_k M_1[b_\beta] b_\beta) \in \text{SN} \). Moreover, we have \( v((\pi_k M_1[b_\beta] b_\beta) \in \text{SN} \). Because from any reduction sequence \( M_1[b_\beta] b_\beta \succ M_1[b_\beta] b_\beta \succ M_1[b_\beta] b_\beta \succ \ldots \), one can construct a reduction sequence \( (\pi_k M_1[b_\beta] b_\beta) \in \text{SN} \). So \( v(M_1[b_\beta] b_\beta) \) is finite, and hence \( M_1[b_\beta] b_\beta \in \text{SN} \).
(CR2): Suppose $M_{d}^{[b^{*}]}/d > N_{a}^{[b^{*}]}$. Then, $(\pi_{k}M_{d}^{[b^{*}]}_{d})^{[b^{k-1}]}/d > (\pi_{k}N_{a}^{[b^{*}]}_{d})^{[b^{k-1}]}/d$ for all $k$ with $1 \leq k \in \omega_{l+1}$. By the hypothesis, we have $M_{d}^{[b^{*}]}_{d} \in \text{RED}[d[b^{*}]]/d$, and hence $(\pi_{k}M_{d}^{[b^{*}]}_{d})^{[b^{k-1}]}/d \in \text{RED}[d[b^{k-1}]]/d$ by the definition of RED. We have $k - 1 \leq l$ and $d([d^{[b^{k-1}]}]/d) < d([d^{[b^{*}]}]/d)$. Hence we can apply the induction hypothesis of (CR2), and then obtain $(\pi_{k}N_{a}^{[b^{*}]}_{d})^{[b^{k-1}]}/d \in \text{RED}[d[b^{k-1}]]/d$. Thus, $N_{a}^{[d^{[b^{*}]}]}_{d} \in \text{RED}[d[b^{k-1}]]/d$ by the definition of RED.

(3R3): Let $M_{d}^{[b^{*}]}_{d}$ be neutral and suppose all the $N_{a}^{[d^{[b^{*}]}]}_{d}$ such that $M_{d}^{[b^{*}]}_{d} \in \text{RED}[d[b^{*}]]/d$. Since $M_{d}^{[b^{*}]}_{d}$ is neutral, $(\pi_{k}M_{d}^{[b^{*}]}_{d})^{[b^{k-1}]}/d \in \text{RED}[d[b^{k-1}]]/d$ for all $k$ with $1 \leq k \in \omega_{l+1}$ cannot itself be a redex. Thus, we obtain $(\pi_{k}M_{d}^{[b^{*}]}_{d})^{[b^{k-1}]}/d > (\pi_{k}N_{a}^{[b^{*}]}_{d})^{[b^{k-1}]}/d$ and $(\pi_{k}N_{a}^{[b^{*}]}_{d})^{[b^{k-1}]}/d \in \text{RED}[d[b^{k-1}]]/d$, because of the hypothesis $N_{a}^{[d^{[b^{*}]}]}_{d} \in \text{RED}[d[b^{k-1}]]/d$ and the definition of RED. We have that $(\pi_{k}M_{d}^{[b^{*}]}_{d})^{[b^{k-1}]}/d$ is neutral and all the typed $\lambda$-terms one step from $(\pi_{k}M_{d}^{[b^{*}]}_{d})^{[b^{k-1}]}/d$ are in RED[d[b^{k-1}]]/d, and that $k - 1 \leq l$ and $d([d^{[b^{k-1}]}]/d) < d([d^{[b^{*}]}]/d)$. Thus, we can apply the induction hypothesis of (CR3), and then obtain $(\pi_{k}M_{d}^{[b^{*}]}_{d})^{[b^{k-1}]}/d \in \text{RED}[d[b^{k-1}]]/d$. Therefore, we obtain $M_{d}^{[b^{*}]}_{d} \in \text{RED}[d[b^{k-1}]]/d$ by the definition of RED. Q.E.D.

By (CR1) of Lemma 11, we have RED $\subseteq$ SN. Using (CR1) – (CR4) in Lemma 11, we can prove the following lemma.

**Lemma 12.** The following conditions hold for $\lambda_{DL}$.

1. If $x^{[d]}/\alpha$ is a typed variable, then $x^{[d]}/\alpha \in \text{RED}[d][/\alpha]$.

2. For any $M_{d}^{[d^{[b^{*}]}]}_{d}$ in RED[d]d and any $N_{a}^{[d^{[b^{*}]}]}_{d}$ in RED[d][/a], if $[N_{a}^{[d^{[b^{*}]}]}/x^{[d]}][M_{d}^{[d^{[b^{*}]}]}_{d}] \in \text{RED}[d][/a]$, then $<\lambda x^{[d]}_{d_{\alpha}}, M^{[d^{[b^{*}]}]}_{d_{\alpha}}>/\alpha - /\beta \in \text{RED}[d][/a - /\beta]$.

3. If $N_{a}^{[d^{[b^{*}]}]}_{d} \in \text{RED}[d][/a]$ and $N_{a}^{[d^{[b^{*}]}]}_{d} \in \text{RED}[d][/a]$, then $<\lambda x^{[d^{[b^{*}]}]}_{d_{\alpha}}, x^{[d^{[b]}]}_{d_{\alpha}}>/\alpha - /\beta \in \text{RED}[d][/a - /\beta]$.

4. If $M_{d}^{[d^{[b^{*}]}]}_{d} \in \text{RED}[d][/a]$ and $N_{a}^{[d^{[b^{*}]}]}_{d} \in \text{RED}[d][/a]$, then $<\lambda x^{[d^{[b^{*}]}]}_{d_{\alpha}}, x^{[d^{[b^{*}]}]}_{d_{\alpha}}>/\alpha - /\beta \in \text{RED}[d][/a - /\beta]$.

5. If $M_{d}^{[d^{[b^{*}]}]}_{d} \in \text{RED}[d][/a - /\beta]$, then $<\lambda x^{[d^{[b^{*}]}]}_{d_{\alpha}}, x^{[d^{[b^{*}]}]}_{d_{\alpha}}>/\alpha - /\beta \in \text{RED}[d][/a - /\beta]$.
6. If \( M_0^{[\alpha]} \in RED_{\alpha} \), \( M_1^{[\beta][\alpha]} \in RED_{\beta[\alpha]} \), ..., \( M_i^{[\theta][\alpha]} \in RED_{\theta[\alpha]} \), then
\[ (M_0^{[\alpha]}, ..., M_i^{[\theta][\alpha]}) \in RED_{\theta[\alpha]}. \]

**Proof.** (1) is obvious by (CR4). (2) can be proved in a similar way as in [4]. (3)–(6) are similar. We thus show only (5).

(5): Suppose \( M[d]\overline{b}^\alpha \in RED_d \). We will show
\[ (tM[d]\overline{b}^\alpha)[\overline{b}\ni\alpha] \in RED_d \] i.e., it is enough to show
\[ \{\overline{b}\ni\alpha\} \in RED_d \] because of (CR1) and the hypothesis, we have \( M[d]b\ni\alpha \in SN \). Thus, we can consider \( v(M[d]b\ni\alpha) \). In the following, we show
\[ \{\overline{b}\ni\alpha\} \in RED_d \] by induction on \( v(M[d]b\ni\alpha) \). This typed \( \lambda \)-term converts (1) \( M[d]^\ni\alpha \) or (2)
\[ \{\overline{b}\ni\alpha\} \in RED_d \] where \( M[d]b\ni\alpha \). For the case (2), we obtain \( N[d]b\ni\alpha \in RED_d \) by (CR2), and we have \( v(M[d]b\ni\alpha) > v(N[d]b\ni\alpha) \). So we obtain
\[ \{\overline{b}\ni\alpha\} \in RED_d \] by induction hypothesis. In both cases, the neutral term \( \{\overline{b}\ni\alpha\} \) converts to reducible terms only, and by (CR3), it is reducible. Therefore
\[ \{\overline{b}\ni\alpha\} \in RED_d \]. Q.E.D.

An expression \([N_1^{\beta_1}/x_1^{\beta_1}, ..., N_n^{\beta_n}/x_n^{\beta_n}]M^\alpha \) denotes the simultaneous substitution. Using Lemma 12, we can prove the following lemma.

**Lemma 13.** Let \( M^\alpha \) be a typed \( \lambda \)-term for \( \lambda DL \). If \( N_1^{\beta_1} \in RED_{\beta_1} \), ..., \( N_n^{\beta_n} \in RED_{\beta_n} \), then \([N_1^{\beta_1}/x_1^{\beta_1}, ..., N_n^{\beta_n}/x_n^{\beta_n}]M^\alpha \in RED_{\alpha} \).

**Proof.** By induction on the construction of \( M \). Let \( \sigma \) be \([N_1^{\beta_1}/x_1^{\beta_1}, ..., N_n^{\beta_n}/x_n^{\beta_n}] \).

1. Case (\( M^\alpha \equiv x_1^{\beta_1} \)): Obvious, i.e., \( \sigma x_1^{\beta_1} \equiv N_1^{\beta_1} \in RED_{\beta_1} \).
2. Case (\( M^\alpha \equiv x^\alpha \) and \( x^\alpha \neq x_1^{\beta_1}, ..., x_n^{\beta_n} \)): By Lemma 12 (1).
3. Case (\( M^\alpha \equiv (\lambda x[d]^{\beta}, N[d]^{\gamma})^{[\theta][\gamma]} \)): By using Lemma 12 (2).
Case \((M^\alpha \equiv (N^\beta, L^\gamma)^\alpha)\) where \((,\) is a pairing \((,\) or an application): By the hypothesis of induction, we have \(\sigma N^\beta \in \text{RED}_\beta\) and \(\sigma L^\gamma \in \text{RED}_\gamma\). We thus obtain \(\sigma M^\alpha \equiv (\sigma N^\beta, \sigma L^\gamma)^\alpha \in \text{RED}_\alpha\) by Lemma 12 (3–4) or by the definition.

Case \((M^\alpha \equiv (\iota M^{d[\hat{b} : \hat{c}])^{\hat{c}})^{\alpha})\): By Lemma 12 (5). Similar to the case above.

Case \((M^\alpha \equiv \langle M^\alpha_0, M^\alpha_1, ..., M^\alpha_l \rangle)\): By using Lemma 12 (6). Similar to the case just above.

Case \((M^\alpha \equiv (\pi M^\beta)^\alpha\) where \(\pi\) is \(\pi_k\) with \(1 \leq k \in \omega_{\nu+1}\) or \(\iota^{-1}\)): By the hypothesis of induction, we have \(\sigma M^\beta \in \text{RED}_\beta\). This fact derives \((\pi \sigma M^\beta)^\alpha \in \text{RED}_\alpha\) by the definition. Therefore we obtain \(\sigma (\pi M^\beta)^\alpha \in \text{RED}_\alpha\). Q.E.D.

**Theorem 14** (Strong normalization) All typed \(\lambda\)-terms for \(\lambda_{DL}\) are strongly normalizable.

**Proof.** In Lemma 13, taking \(N_1 \equiv x_1, ..., N_n \equiv x_n\), we have \(M^\alpha \in \text{RED}_\alpha\) for any typed \(\lambda\)-term \(M^\alpha\) for \(\lambda_{DL}\), i.e., \(\text{TERM} \subseteq \text{RED}\). Since we already have \(\text{RED} \subseteq \text{SN}\), we obtain \(\text{TERM} \subseteq \text{SN}\). Q.E.D.

**References**


Waseda Institute for Advanced Study,
1-6-1 Nishi Waseda, Shinjuku-ku,
Tokyo 169-8050, JAPAN
e-mail: kamide@mtc.biglobe.ne.jp