STRONG NORMALIZATION OF THE SECOND-ORDER SYMMETRIC LAMBDA–MU CALCULUS

YORIYUKI YAMAGATA

Abstract. Parigot [12] suggested symmetric structural reduction rules to ensure unique representation of data types. We prove strong normalization of the second-order $\lambda\mu$-calculus with such rules.

1. Introduction

Ever since its introduction by Parigot [11], the $\lambda\mu$-calculus seems to have been quite successful as an idealized programming language, especially for modeling of control operators. The $\lambda\mu$-calculus has continuation semantics (Groote [5]) and an exact correspondence to the call-by-name control-operator calculus (Groote [6]). As shown by Streicher and Reus [17], it is naturally derived from the Krivine machine.

Parigot [11] was also motivated by the possibility of extracting witnesses from classical proofs of Σ^0_1-formulae. Unfortunately, the reduction rules of the $\lambda\mu$-calculus do not appear to be sufficient for this purpose. For example, let $A(x)$ be an atomic formula of arithmetic. We represent $\exists x A(x)$ as $\forall X(\forall x(A(x) \to X) \to X)$ in the second-order language, where the variable X ranges over propositions. We expect a closed, normal deduction of $\exists x A(x)$ to somehow contain the unique first-order term t such that $A(t)$ holds. However, consider the following scenario: Suppose that $A(t)$ holds but $A(u)$ does not hold, and let M be a deduction of $A(t)$ represented as a $\lambda\mu$-term. Then

$$\Lambda X.\lambda\alpha.\mu\beta.[\beta]\alpha u(\mu\gamma.[\beta]\alpha t M) \tag{1}$$

is a closed, normal deduction of $\exists x A(x)$, but it apparently contains two terms (t and u). Moreover, u is not the witness for $\exists x A(x)$. This suggests that further reduction is needed to extract the witness.

In order to solve a similar problem on normal forms of the natural-number type, Parigot [12] proposed new reduction rules $M(\mu\alpha.N) \Rightarrow \mu\beta.N[M^*/\alpha]$, where $N[M^*/\alpha]$ is defined by inductively replacing all occurrences of $[\alpha]L$ in N with $[\alpha]M(L[M^*/\alpha])$. We refer collectively to these new rules and the structural reduction rules of the $\lambda\mu$-calculus as *symmetric structural reduction rules*, and we call the $\lambda\mu$-calculus with symmetric structural reduction the *symmetric* $\lambda\mu$-calculus. Ong and Stewart [10] subsequently defined the call-by-value $\lambda\mu$-calculus $\lambda\mu_v$, which uses symmetric structural reduction, and showed that it can encode various control structures in computer programs.

In this article, we prove strong normalization of the second-order predicate symmetric $\lambda\mu$-calculus and show that a closed, normal deduction of $\exists x A(x)$ for an atomic formula $A(x)$ contains the unique first-order term t which satisfies $A(t)$. Ong and Stewart [10] mentioned that strong normalization of the $\lambda\mu_v$-calculus can be proved by the method of reducibility candidates. Nakazawa [8] gave a proof of strong normalization using CPS-translation. We improve on their result by removing the call-by-value restriction on the reduction strategy and showing that witnesses can always be extracted from deductions of Σ^0_1-formulae.

In the presence of symmetric structural reduction, the final result of a computation may depend on the reduction strategy employed. However, this nondeterminism in our calculus is manageable in the sense that if a program is of type $\exists x A(x)$, then all values computed with that program are guaranteed to satisfy the specification given by $A(x)$. Since symmetric structural reduction reflects the symmetric nature of classical logic, the non-determinism in our $\lambda\mu$-calculus may be an inherent property of classical logic.

Strong normalization and extraction of witnesses for Σ^0_1-formulae are known to hold of the symmetric lambda calculus for Peano arithmetic introduced By Barbanera and Berardi [3]. By use of an extraction operator, witnesses for Σ^0_1-formulae can also be extracted in Parigot's $\lambda\mu$-calculus as shown by Parigot [12] and the second-order sequent calculus as shown by Danos et al. [4]. Compared to the symmetric lambda calculus for Peano arithmetic, however, our approach has the advantage that the symmetric $\lambda\mu$-calculus is an extension of the $\lambda\mu$-calculus, which is in turn an extension of the λ-calculus. Furthermore, unlike the case of the $\lambda\mu$-calculus and the second-order sequent calculus, our approach does not require use of an extraction operator.

The main technical result presented in this article is strong normalization. Nondeterminism of our calculus precludes the possibility of CPS-translation. Further, the symmetric nature of our structural reduction appears to preclude direct adaptation of Parigot's proof [13] of strong normalization for the original $\lambda\mu$-calculus. Instead, we adapt Barbanera and Berardi's method [3, 2] of using transfinite induction for defining reducibility candidates. The requirements for reducibility candidates are naturally obtained from the notion of validity introduced by Prawitz [15].

2. Symmetric $\lambda\mu$-calculus

The logic treated in this paper is second-order classical predicate logic. We formulate second-order predicate logic as a many-sorted logic whose domains of quantification are either first-order objects, or predicates over first-order objects. To denote first-order objects, we introduce *first-order terms*, which are constructed using constants $c_1, c_2, \ldots$, function symbols $f_1, f_2, \ldots$, and *first-order variables* $x_1, x_2, \ldots$. Formally, the syntax of a first-order term t is specified by the BNF notation

$$t ::= c_i \mid x_j \mid f_k t_1 \cdots t_{r_k}, \qquad (2)$$

where c_i is a constant, x_j is a first-order variable, f_k is a function symbol for a function with arity r_k, and $t_1, \ldots, t_{r_j}$ are first-order terms. The symbols x, y, z are used to denote first-order variables, and t, u, v are used to denote first-order terms.

In this article, we refer to formulae as *types.* A type either consists of just $\perp$ (the absurdity) or has the form A or $\neg A$ (the negation of A) for some A that contains neither $\neg$ nor $\perp$. (For example, there is no type such as $A \to \perp$.) We call such an A a *proposition.* To emphasize the restriction placed on negation, we use the symbol $\bullet$ for $\neg$. For a proposition A, we define $\bullet A$ by the involution $\bullet\bullet A = A$. The symbols $A, B, A_1, \ldots$ are used to denote propositions, while the symbols $C, D, C_1, \ldots$ are used for types.

Table 1. $\lambda\mu$-terms and their types

$$\frac{}{a_i : A_i} \qquad \frac{[\alpha^C]}{\alpha^C : C}$$

$$\frac{M : \bullet A \quad N : A}{[M]N : \bot}\,[\,] \qquad \frac{\begin{array}{c}[\alpha^A]\\ \vdots\ \mathcal{D}_1\\ M : \bot\end{array}}{\mu\alpha^A.M : \bullet A}\,\mu\alpha \qquad \frac{\begin{array}{c}[\alpha^{\bullet A}]\\ \vdots\ \mathcal{D}_2\\ M : \bot\end{array}}{\mu\alpha^{\bullet A}.M : A}\,\mu\alpha$$

$$\frac{M : A \to B \quad N : A}{MN : B}\,\text{app.} \qquad \frac{\begin{array}{c}[\alpha^A]\\ \vdots\ \mathcal{D}\\ M : B\end{array}}{\lambda\alpha^A.M : A \to B}\,\lambda\alpha$$

$$\frac{M : \forall x A}{Mt : A[t/x]}\,\text{app.}^1 \qquad \frac{\begin{array}{c}\vdots\ \mathcal{D}_3\\ M : A\end{array}}{\lambda x.M : \forall x A}\,\lambda^1$$

$$\frac{M : \forall X A}{MT : A[T/X]}\,\text{app.}^2 \qquad \frac{\begin{array}{c}\vdots\ \mathcal{D}_4\\ M : A\end{array}}{\lambda X.M : \forall X A}\,\lambda^2$$

$\mathcal{D}_3$ has no open assumptions which have x as a free predicate variable. Similarly, $\mathcal{D}_4$ has no open assumptions which have X as a free first-order variable.

Intuitively, propositions are the objects affirmed or denied, and types are assertions which affirm or deny propositions or show that a contradiction arises. This view is proposed by Stewart [16]. Formally, propositions are built up by using the following symbols and connectives: $=$, to express equality between first-order objects; *n-ary predicate variables* X_i^n, to form n-ary predicates over first-order objects; $\to$, to express implication; and $\forall$, to indicate universal quantification over first-order objects and predicates. The syntax of a proposition A is specified by the BNF notation

$$A ::= t_1 = t_2 \mid X_i^n t_1 \cdots t_n \mid A_1 \to A_2 \mid \forall x A \mid \forall X_i^n A, \tag{3}$$

where $t_1, \ldots, t_n$ are first-order terms. We call a proposition of the form $t_1 = t_2$ an *atomic* proposition. X, Y, Z are used to denote predicate variables. Logical connectives and quantifiers other than $\to$ and $\forall$ are defined by using second-order constructs. For example, $\exists x A(x)$ is defined as $\forall X(\forall x(A(x) \to X) \to X)$, and $A \wedge B$ as $\forall X((A \to B \to X) \to X)$.

Definition 1. *An (n-ary)* abstraction term *is a term of the form $\lambda x_1 \cdots x_n.A$ for some proposition A. The abstraction term $\lambda x_1 \cdots x_n.A$ denotes the n-ary predicate P defined by $Px_1 \ldots x_n \leftrightarrow A(x_1, \ldots, x_n)$. If T is the abstraction term $\lambda x_1 \cdots x_n.A$, then* substitution of *T for the n-ary predicate variable X in B, which is denoted by $B[T/X]$, is defined by replacing all occurrences of $Xt_1 \cdots t_n$ in which X is free in B with $A[t_1, \ldots, t_n/x_1, \ldots, x_n]$.*

Definition 2. *To denote deductions, we employ what we call* $\lambda\mu$-terms. *If a $\lambda\mu$-term M denotes a deduction of some type C, we say that M* is of type *C. As shown*

Table 1, we define the set of $\lambda\mu$-terms by relating them to various deductions, each of which is presented in the form of a tree. Whenever an assumption of type C (where C is a type other than $\perp$) appears in a $\lambda\mu$-term, we call that assumption a $\lambda\mu$-variable of type C. *We denote $\lambda\mu$-variables of type C by Greek letters, with C as a superscript: $\alpha^C, \beta^C, \ldots$. We choose the particular set* $\mathbf{Ax}$ *of propositions as axioms. For each axiom A_i, there is a deduction of A_i without any premise and inference. We denote such a deduction by* $\lambda\mu$-constant a_i.

In Table 1, C stands for a type other than $\perp$ (and $M : C$ means that M is of type C); A and B stand for propositions. When the inference discharges some assumption, the discharged assumption is written to the right of the name of the inference in the diagrams below. (In the expression $\mu\alpha$, for instance, μ is the name of the inference and α is the discharged assumption.) A_i denotes the i-th axiom in $\mathbf{Ax}$. *The deduction of the axiom A_i is denoted by the constant a_i. λ and μ bind the $\lambda\mu$-variable α^A in $\lambda\alpha^A$ and $\mu\alpha^A$, respectively, and similarly for the $\lambda\mu$-variable $\alpha^{\bullet A}$. In addition, λ binds the first-order variable x in λx, and the predicate variable X in λX. $FV(M)$ denotes the set of all free $\lambda\mu$-variables in the $\lambda\mu$-term M. If M has no free $\lambda\mu$-variables (but possibly has free first-order variables and/or free predicate variables), we say that M is* closed. *If a $\lambda\mu$-term M is constructed by rule* app., app.[1]*, or* app.[2]*, we say that M* is constructed by application*, or that M* has a form of application*, or simply that M* is application.

If we translate denial ($\bullet$) as negation ($\neg$), all the above rules are valid in classical logic. Conversely, well-typed terms of Parigot's $\lambda\mu$-calculus can be translated into $\lambda\mu$-terms of the sort given above, by replacing μ-variables of type A (where A is what we are calling a proposition) with $\lambda\mu$-variables of type $\bullet A$. Hence, the set of classically valid propositions is exactly the same as the set of propositions which are inhabited by closed $\lambda\mu$-terms.

The reason for use of the "Church-style formulation," that is, incorporation of typing information into the $\lambda\mu$-terms, is that in the proof of strong normalization, we seem to need the fact that every $\lambda\mu$-term defined in this way denotes a unique deduction.

Parigot [14] proposed a single-conclusion system for the $\lambda\mu$-calculus which is similar to our system but allows for negation ($\neg$)—which is exactly the same as denial ($\bullet$) for us—to be applied to subformulae as well.

For a type C other than $\perp$, and a $\lambda\mu$-term N of type C, *substitution* of N for the $\lambda\mu$-variable α^C in a $\lambda\mu$-term M is defined by replacing each free occurrence of α^C in M with N; the resulting $\lambda\mu$-term is denoted by $M[N/\alpha^C]$. To substitute a first-order term t for the first-order variable x in a $\lambda\mu$-term M, first do the following: for every type C, and every $\lambda\mu$-variable α^C in M that is outside the binding constructs of x, replace α^C in M with $\alpha^{C[t/x]}$; and then replace all free occurrences of x in first-order terms in M with t. The $\lambda\mu$-term obtained by this procedure is denoted by $M[t/x]$. Substitution of an abstraction term T for the predicate variable X in M is defined similarly, and the resulting $\lambda\mu$-term is denoted by $M[T/X]$.

Definition 3. *We define transformations of $\lambda\mu$-terms by the rules given below. To show that each of these transformations preserves the type of a $\lambda\mu$-term, we give a part of a corresponding deduction for each $\lambda\mu$-term that appears in the rules. The symbol $\mathcal{D}$ is used to denote the common part of deductions before and after a transformation. Note that substitution of the $\lambda\mu$-term N for the $\lambda\mu$-variable α in* *the $\lambda\mu$-term M corresponds concatenation of two deduction $\mathcal{D}_1$ and $\mathcal{D}_2$ in the place of the assumption α, if N corresponds $\mathcal{D}_1$ and M corresponds $\mathcal{D}_2$.*

For rule R (where R denotes any of the rules given below), the compatible closure of the binary relation defined by R is called the one-step reduction relation by R *or* one-step reduction by R *(and is denoted by $\Rightarrow^1_R$). The union, over all rules R, of such one-step reduction relations is called the* one-step reduction relation *or* one-step reduction *(and is denoted by $\Rightarrow^1$). The reflexive and transitive closure of the one-step reduction relation is called the* reduction relation, *or simply* reduction *(and is denoted by $\Rightarrow$). Similarly, the reflexive and transitive closure of $\Rightarrow^1_R$ is denoted by $\Rightarrow_R$.*

If there is no rule that could be applied to a $\lambda\mu$-term M, we say that M is normal. *If $M \Rightarrow N$ and N is normal, we say that N is a* normal form *of M. The length of the sequence $M \Rightarrow^1 M_1 \Rightarrow^1 \cdots \Rightarrow^1 M_n$ of one-step reductions is defined to be n. Let $w(M)$ be the length of the longest sequence of one-step reductions, if such exists; otherwise, $w(M)$ is undefined. M is* strongly normalizable *if and only if $w(M)$ is defined.*

λ-reduction rule:

$$\frac{\dfrac{\begin{matrix}[\alpha^A]\\ \vdots\ \mathcal{D}\\ M:B\end{matrix}}{\lambda\alpha.M:A\to B}\,\lambda\alpha \qquad N:A}{(\lambda\alpha.M)N:B} \quad\Rightarrow^1_\lambda\quad \begin{matrix}N:A\\ \vdots\ \mathcal{D}\\ M[N/\alpha]:B\end{matrix} \tag{4}$$

λ_1-reduction rule:

$$\frac{\dfrac{\begin{matrix}\vdots\ \mathcal{D}\\ M:A\end{matrix}}{\lambda x.M:\forall xA}}{(\lambda x.M)t:A[t/x]} \quad\Rightarrow^1_{\lambda_1}\quad \begin{matrix}\vdots\ \mathcal{D}[t/x]\\ M[t/x]:A[t/x]\end{matrix} \tag{5}$$

$\mathcal{D}[t/x]$ is the deduction obtained from $\mathcal{D}$ by substitution of t for x in types and $\lambda\mu$-terms in $\mathcal{D}$.

λ_2-reduction rule:

$$\frac{\dfrac{\begin{matrix}\vdots\ \mathcal{D}\\ M:A\end{matrix}}{\lambda X.M:\forall XA}}{(\lambda X.M)T:A[T/X]} \quad\Rightarrow^1_{\lambda_2}\quad \begin{matrix}\vdots\ \mathcal{D}[T/X]\\ M[T/X]:A[T/X]\end{matrix} \tag{6}$$

The definition of $\mathcal{D}[T/X]$ is analogous to that of $\mathcal{D}[t/x]$. The λ-, λ_1-, and λ_2-rules are just the $\to$- and $\forall$-contraction rules of natural deduction. Also, they are extensions of β-reduction in the λ-calculus for predicate logic.

μ-reduction rule: *This rule appeared in Parigot* [14], *Stewart* [16]. *There are two cases, μ_L and μ_R. The μ_L-rule is similar to the $\neg$-contraction rule of natural deduction.*

$$\frac{\dfrac{\begin{matrix}[\alpha^A]\\ \vdots\ \mathcal{D}\\ M:\bot\end{matrix}}{\mu\alpha.M:\bullet A}\,\mu\alpha \qquad N:A}{[\mu\alpha.M]N:\bot} \quad\Rightarrow^1_{\mu_L}\quad \begin{matrix}N:A\\ \vdots\ \mathcal{D}\\ M[N/\alpha]:\bot\end{matrix} \tag{7}$$

(8)
$$\dfrac{M:\bullet A \qquad \dfrac{\begin{matrix}[\alpha^{\bullet A}]\\ \vdots\ \mathcal{D}\\ N:\bot\end{matrix}}{\mu\alpha.N:A}\ \mu\alpha}{[M]\mu\alpha.N:\bot} \quad \Rightarrow^1_{\mu_R} \quad \begin{matrix}M:\bullet A\\ \vdots\ \mathcal{D}\\ N[M/\alpha]:\bot\end{matrix}$$

ζ-reduction rule: *Again, there are two cases, ζ_L and ζ_R. The former appeared in Parigot [14]. If we think of $\bullet$ as negation, then Andou's reduction for $\bot_c$ [1] also resembles the ζ_L-rule. The ζ-rules are needed for the encoding of symmetric structural reduction, as shown later. The ζ_1- and ζ_2-rules are analogous to the ζ_L-rules.*

(9)
$$\dfrac{\dfrac{\begin{matrix}[\alpha^{\bullet(A\to B)}]\\ \vdots\ \mathcal{D}\\ M:\bot\end{matrix}}{\mu\alpha.M:A\to B}\ \mu\alpha \qquad N:A}{(\mu\alpha.M)N:B} \quad \Rightarrow^1_{\zeta_L} \quad \dfrac{\begin{matrix}\dfrac{\dfrac{[\beta^{\bullet B}] \qquad \dfrac{[\gamma^{A\to B}] \quad N:A}{\gamma N:B}}{[\beta](\gamma N):\bot}}{\mu\gamma.[\beta](\gamma N):\bullet(A\to B)}\ \mu\gamma\\ \vdots\ \mathcal{D}\\ M[\mu\gamma.[\beta](\gamma N)/\alpha]:\bot\end{matrix}}{\mu\beta.M[\mu\gamma.[\beta](\gamma N)/\alpha]:B}\ \mu\beta$$

(10)
$$\dfrac{M:A\to B \qquad \dfrac{\begin{matrix}[\alpha^{\bullet A}]\\ \vdots\ \mathcal{D}\\ N:\bot\end{matrix}}{\mu\alpha.N:A}\ \mu\alpha}{M(\mu\alpha.N):B} \quad \Rightarrow^1_{\zeta_R} \quad \dfrac{\begin{matrix}\dfrac{\dfrac{[\beta^{\bullet B}] \qquad \dfrac{M:A\to B \quad [\gamma^{A}]}{M\gamma:B}}{[\beta](M\gamma):\bot}}{\mu\gamma.[\beta](M\gamma):\bullet A}\ \mu\gamma\\ \vdots\ \mathcal{D}\\ N[\mu\gamma.[\beta](M\gamma)/\alpha]:\bot\end{matrix}}{\mu\beta.N[\mu\gamma.[\beta](M\gamma)/\alpha]:B}\ \mu\beta$$

ζ_1-reduction rule:

(11)
$$\dfrac{\dfrac{\begin{matrix}[\alpha^{\bullet\forall xA}]\\ \vdots\ \mathcal{D}\\ M:\bot\end{matrix}}{\mu\alpha.M:\forall xA}}{(\mu\alpha.M)t:A[t/x]} \quad \Rightarrow^1_{\zeta_1} \quad \dfrac{\begin{matrix}\dfrac{\dfrac{[\beta^{\bullet A[t/x]}] \qquad \dfrac{[\gamma^{\forall xA}]}{\gamma t:A[t/x]}}{[\beta](\gamma t):\bot}}{\mu\gamma.[\beta](\gamma t):\bullet\forall xA}\ \mu\gamma\\ \vdots\ \mathcal{D}\\ M[\mu\gamma.[\beta](\gamma t)/\alpha]:\bot\end{matrix}}{\mu\beta.M[\mu\gamma.[\beta](\gamma t)/\alpha]:A[t/x]}\ \mu\beta$$

ζ_2-reduction rule:

(12)
$$\dfrac{\dfrac{\begin{matrix}[\alpha^{\bullet\forall XA}]\\ \vdots\ \mathcal{D}\\ M:\bot\end{matrix}}{\mu\alpha.M:\forall XA}}{(\mu\alpha.M)T:A[T/X]} \quad \Rightarrow^1_{\zeta_2} \quad \dfrac{\begin{matrix}\dfrac{\dfrac{[\beta^{\bullet A[T/X]}] \qquad \dfrac{[\gamma^{\forall XA}]}{\gamma T:A[T/X]}}{[\beta](\gamma T):\bot}}{\mu\gamma.[\beta](\gamma T):\bullet\forall XA}\ \mu\gamma\\ \vdots\ \mathcal{D}\\ M[\mu\gamma.[\beta](\gamma T)/\alpha]:\bot\end{matrix}}{\mu\beta.M[\mu\gamma.[\beta](\gamma T)/\alpha]:A[T/X]}\ \mu\beta$$

$\bullet$-reduction rule: *The $\bullet$-, $\bullet_1$-, and $\bullet_2$-rules are similar to Prawitz's reduction for $\bot_c$. The idea is to reduce the complexity of the type of the assumption discharged by the μ-inference if the type is a denial.*

$$\dfrac{\begin{matrix}[\alpha^{\bullet(A\to B)}]\\ \vdots\ \mathcal{D}\\ M:\bot\end{matrix}}{\mu\alpha.M : A\to B}\,\mu\alpha \quad \Rightarrow^1_\bullet \quad \dfrac{\dfrac{\begin{matrix}\dfrac{\dfrac{[\beta^{\bullet B}] \quad \dfrac{[\gamma^{A\to B}]\quad[\delta^A]}{\gamma\delta : B}}{[\beta](\gamma\delta):\bot}}{\mu\gamma.[\beta](\gamma\delta):\bullet(A\to B)}\,\mu\gamma \\ \vdots\ \mathcal{D}\\ M[\mu\gamma.[\beta](\gamma\delta)/\alpha]:\bot\end{matrix}}{\mu\beta.M[\mu\gamma.[\beta](\gamma\delta)/\alpha]:B}\,\mu\beta}{\lambda\delta.\mu\beta.M[\mu\gamma.[\beta](\gamma\delta)/\alpha]:A\to B}\,\lambda\delta \tag{13}$$

$\bullet_1$-reduction rule:

$$\dfrac{\begin{matrix}[\alpha^{\bullet\forall xA}]\\ \vdots\ \mathcal{D}\\ M:\bot\end{matrix}}{\mu\alpha.M : \forall xA} \quad \Rightarrow^1_{\bullet_1} \quad \dfrac{\dfrac{\begin{matrix}\dfrac{\dfrac{[\beta^{\bullet A}] \quad \dfrac{[\gamma^{\forall xA}]}{\gamma x : A}}{[\beta](\gamma x):\bot}}{\mu\gamma.[\beta](\gamma x):\bullet\forall xA}\,\mu\gamma \\ \vdots\ \mathcal{D}\\ M[\mu\gamma.[\beta](\gamma x)/\alpha]:\bot\end{matrix}}{\mu\beta.M[\mu\gamma.[\beta](\gamma x)/\alpha]:A}\,\mu\beta}{\lambda x.\mu\beta.M[\mu\gamma.[\beta](\gamma x)/\alpha]:\forall xA} \tag{14}$$

$\bullet_2$-reduction rule: *Let T_0 be the abstraction term $\lambda x_1\cdots x_n.Xx_1\cdots x_n$, where n is the arity of X.*

$$\dfrac{\begin{matrix}[\alpha^{\bullet\forall XA}]\\ \vdots\ \mathcal{D}\\ M:\bot\end{matrix}}{\mu\alpha.M : \forall XA} \quad \Rightarrow^1_{\bullet_2} \quad \dfrac{\dfrac{\begin{matrix}\dfrac{\dfrac{[\beta^{\bullet A}] \quad \dfrac{[\gamma^{\forall XA}]}{\gamma T_0 : A}}{[\beta](\gamma T_0):\bot}}{\mu\gamma.[\beta](\gamma T_0):\bullet\forall XA}\,\mu\gamma \\ \vdots\ \mathcal{D}\\ M[\mu\gamma.[\beta](\gamma T_0)/\alpha]:\bot\end{matrix}}{\mu\beta.M[\mu\gamma.[\beta](\gamma T_0)/\alpha]:A}\,\mu\beta}{\lambda X.\mu\beta.M[\mu\gamma.[\beta](\gamma T_0)/\alpha]:\forall XA} \tag{15}$$

S-reduction rule: *This rule is analogous to η-reduction in the λ-calculus. If the $\lambda\mu$-variable $\alpha^{\bullet A}$ is not free in M, the following reduction is possible.*

$$\dfrac{\dfrac{[\alpha^{\bullet A}] \quad \begin{matrix}\vdots\ \mathcal{D}\\ M:A\end{matrix}}{[\alpha]M:\bot}}{\mu\alpha.[\alpha]M:A}\,\mu\alpha \quad \Rightarrow^1_S \quad \begin{matrix}\vdots\ \mathcal{D}\\ M:A\end{matrix} \tag{16}$$

There is a certain amount of redundancy in these reduction rules. ζ_L-reduction is derivable from $\bullet$- and λ-reduction. ζ_1- and ζ_2-reduction are derivable from $\bullet_1$-, $\bullet_2$-, and λ-reduction. Despite this redundancy, we choose to exhibit all the rules here in order to make clear the symmetric nature of the ζ-rule.

We can easily see that our calculus is an extension of Parigot's $\lambda\mu$-calculus. The λ-rule is the same as β-reduction in the λ-calculus. The μ-rule is an extension of Parigot's renaming rule $[\alpha]\mu\beta.M \Rightarrow M[\alpha/\beta]$. Using μ and our ζ-rules, we can derive the symmetric structural reduction rules from Parigot [12] that we referred to in Section 1:

$$\begin{aligned} (\mu\alpha.\ \cdots[\alpha]N\cdots)L &\Rightarrow_\zeta \mu\beta.\ \cdots[\mu\gamma.[\beta](\gamma L)]N\cdots \\ &\Rightarrow_\mu \mu\beta.\cdots[\beta](NL)\cdots \end{aligned} \tag{17}$$

$$\begin{aligned} N(\mu\alpha.\ \cdots[\alpha]L\cdots) &\Rightarrow_\zeta \mu\beta.\ \cdots[\mu\gamma.[\beta](N\gamma)]L\cdots \\ &\Rightarrow_\mu \mu\beta.\ \cdots[\beta](NL)\cdots \end{aligned} \tag{18}$$

In this way, we can formalize structural reduction by usual substitution.

Our S-reduction rule is equivalent to the (S_2)-rule in Parigot [12]. S-reduction and $\bullet$-, $\bullet_1$-, and $\bullet_2$-reduction are needed for the extraction of witnesses.

3. Extraction of witnesses from Σ^0_1-formulae

Since our calculus can simulate Parigot's symmetric structural reduction, $\lambda\mu$-terms of the natural-number type are reduced to Church numerals as mentioned by Parigot [12]. See Nour [9] for a complete proof. However, this does not necessarily mean that witnesses can be extracted from proofs of Σ^0_1-formulae. It is instructive to see how the standard method used in intuitionistic logic fails, even for Σ^0_1-formulae. For more about witness extraction of intuitionistic logic, for example see Girard et al. [7].

In the following discussion, we assume that 0 is included in the constants, and S is included in the unary function symbols. 0 stands for zero in the natural numbers and S stands for the successor function. The formula $N(x)$ is defined as $\forall X(X0 \to (\forall y(Xy \to XSy) \to Xx))$.

Consider the deduction $\mathcal{D}$ of $\exists x(N(x) \wedge A(x))$ where $A(x)$ is an atomic proposition, shown in Fig. 1.

Figure 1. Deduction of $\exists x(N(x) \wedge A(x))$

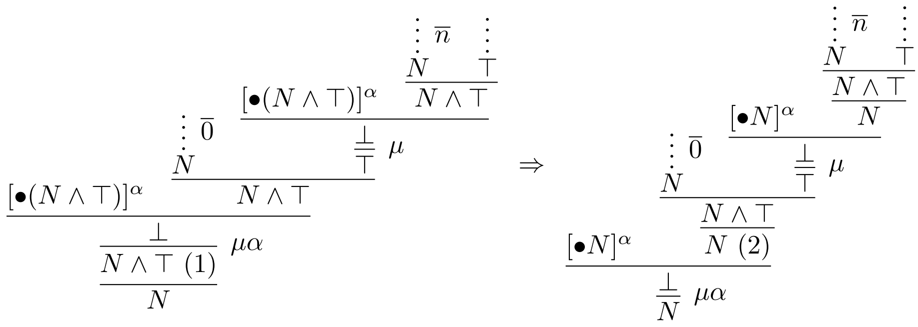
$$\dfrac{\dfrac{[\bullet\exists x(N(x)\wedge A(x))]^\alpha \qquad \dfrac{\dfrac{\begin{array}{c}\vdots\ \mathcal{D}_0\\ N(0)\end{array} \qquad \dfrac{\dfrac{[\bullet\exists x(N(x)\wedge A(x))]^\alpha \qquad \dfrac{\begin{array}{c}\vdots\ \mathcal{D}_n\\ N(n)\end{array} \quad \begin{array}{c}\vdots\\ A(n)\end{array}}{\exists x(N(x)\wedge A(x))}}{\bot}}{A(0)}\,\mu}{N(0)\wedge A(0)}}{\exists x(N(x)\wedge A(x))}}{\bot}}{\exists x(N(x)\wedge A(x))}\,\mu\alpha$$

$\mathcal{D}$ is not normal, but further reduction would involve only α and the $\exists$ quantifier. In particular, there is no way to eliminate $\mathcal{D}_0$.

Dropping the predicative part and supplying $\wedge$-elimination yields the left-hand diagram in Figure 2, which reduces to the right-hand diagram in the figure by reduction of redex (1). (N is the natural-number type, and $\overline{0}$ and $\overline{n}$ are the Church numerals for 0 and n, respectively.)

If we reduce redex (2), the right-hand diagram becomes $\mu\alpha.[\alpha]\overline{0}$, which reduces to $\overline{0}$. However, there is no guarantee that $A(0)$ holds.

FIGURE 2. Propositional part and its reduction



The cause of the problem appears to stem from the presence of conjunction. Hence, we consider only $\lambda\mu$-terms of type $\exists x A(x)$ for some atomic proposition $A(x)$ and show that their normal forms end with introduction of the $\exists$ quantifier. Since $\exists x A(x)$ is derivable from $\exists x(N(x) \wedge A(x))$, this approach is sufficient for extraction of witnesses for Σ^0_1-formulae.

Definition 4. *If the set* **Ax** *of axioms satisfies the following conditions, we say that* **Ax** *is a* Post system*.*

(1) *Each* $A \in \mathbf{Ax}$ *is of the form* $p_1 \to (p_2 \to (\cdots \to (p_{n-1} \to p_n)\cdots))$*, where* $p_1, \ldots, p_n$ *are atomic propositions. For* $n = 2$*, 3, and 4, what this means is that these propositions are of the form* $p_1 \to p_2$*,* $p_1 \to (p_2 \to p_3)$*, and* $p_1 \to (p_2 \to (p_3 \to p_4))$*, respectively.*

(2) *If* $A \in \mathbf{Ax}$*, then for each first-order variable* x *and first-order term* t*, the proposition* $A[t/x]$ *obtained by substituting* t *for* x *is also an element of* **Ax***.*

Note that in intuitionistic logic, if all non-logical axioms constitute a Post system, all the rules of inference in a closed, normal deduction of an atomic proposition are elimination rules. We will see later that a similar property holds of classical logic in the presence of symmetric structural reduction rules. By restricting quantifiers over the natural numbers to the predicate $N(x) \equiv \forall X(X0 \to (\forall y(Xy \to XSy) \to Xx))$ and replacing $\perp$ (the symbol for absurdity) in the language of arithmetic with $0 = S0$, second-order Peano Arithmetic can be formalized by a Post system.

Theorem 1. *Assume that* **Ax** *is a Post system. Let* $A(x)$ *be an atomic proposition, and let* M *be a closed, normal* $\lambda\mu$*-term of type* $\exists x A(x)$*. Then* M *has the form* $\lambda X.\lambda\alpha.\alpha t N$*, where* t *is a first-order term and* N *is a closed, normal* $\lambda\mu$*-term of type* $A(t)$*.*

To prove the theorem, we need the following lemma.

Lemma 1. *Assume that* **Ax** *is a Post system. Let* M *be a normal* $\lambda\mu$*-term whose type is an atomic proposition. Assume that every* $\lambda\mu$*-variable of* M *either is of type* $\forall x(A(x) \to X)$ *for some proposition* A *or is a denial* $\bullet B$*. Assume, furthermore, that* M *does not begin with* μ*. Then* M *is closed.*

Proof. By induction on the construction of M.

Case I. M is a $\lambda\mu$-variable. This case cannot occur, since $\forall x(A(x) \to X)$ and $\bullet B$ are not atomic propositions.

Case II. M is an axiom. Then M is closed.

Case III. M is constructed by application. Then M has the form $M_0\Sigma_1\Sigma_2 \dots \Sigma_n$ where M_0 is an axiom or a $\lambda\mu$-variable and Σ_i is either a $\lambda\mu$-term, a first-order term, or an abstraction term. Suppose that M_0 is a $\lambda\mu$-variable. For application to be possible, and since the type of M_0 is either $\forall x(A(x) \to X)$ or a denial, M_0 must be of type $\forall x(A(x) \to X)$. Therefore, the type of M contains the predicate variable X, which contradicts the assumption that M is of an atomic proposition. Hence, M_0 is an axiom. By the constraints given in Definition 4, Σ_i is a $\lambda\mu$-term whose type is an atomic proposition. Since M is normal, Σ_i cannot begin with μ (otherwise, ζ-reduction could be applied). By the induction hypothesis, Σ_i has no free $\lambda\mu$-variables. Hence, M is closed.

Case IV. M begins with λ. This case cannot occur, since M is of an atomic proposition.

Case V. M is of the form $[\alpha]M_0$. Then M is of type $\bot$. This contradicts the assumption that M is of an atomic proposition. □

Remark. Using this lemma, we can show that a closed, normal $\lambda\mu$-term L of an atomic proposition is constructed by axioms and application. The only way in which the proof differs from the proof in intuitionistic logic is in the case where L contains the symbols μ and [ ]. Because of the ζ-reduction rules, these symbols can appear only at the beginning of a $\lambda\mu$-term. So suppose that $L \equiv \mu\alpha.[\alpha]M$ and that L is a closed, normal $\lambda\mu$-term of an atomic proposition. M cannot begin with μ, otherwise we could apply μ-reduction to $\mu\alpha.[\alpha]M$. Hence we can apply the above lemma to M. Since M has no free $\lambda\mu$-variables, S-reduction can be applied to L, in contradiction to our assumption about L. Hence, neither μ nor [ ] appears in L. The rest of the proof is similar to the case of intuitionistic logic.

Proof. Proof of the Theorem By the type of M, M could begin with either λ or μ. However, if M began with μ, a $\bullet_2$-reduction rule could be applied to M. Hence M is of the form $\lambda X.M_1$. By similar reasoning, M_1 begins with λ. Hence, M is of the form $\lambda X.\lambda\alpha^{\forall y(A(y)\to X)}.M_2$. By the type of M_2, M_2 either begins with μ or can be constructed by application. We consider these two possibilities.

Case I. M_2 is constructed by application. Let $M_2 \equiv M_3\Sigma_1\Sigma_2 \dots \Sigma_n$ where M_3 is not constructed by application. M_3 is either an axiom a_i for some integer i or a $\lambda\mu$-variable, otherwise we could apply one of $\lambda, \lambda_1, \lambda_2, \zeta, \zeta_1, \zeta_2$-reduction rules to M_2. On the other hand, M_3 cannot be an axiom since the type of M_3 contains a predicate variable X but this is impossible. Since the only open $\lambda\mu$-variable of M_2 has type $\forall y(A(y) \to X)$, M_2 has a form $\alpha t N$ where t is a first-order term and N is a $\lambda\mu$-term with type $A(t)$.

N cannot begin with μ, otherwise we could apply ζ-reduction to the redex $(\alpha t)N$. Hence, we can apply the previous lemma to N and infer that N is closed. So the conclusion of the theorem is satisfied.

Case II. M_2 begins with μ. Then M_2 is of the form $\mu\beta.[\beta]M_3$. M_3 cannot begin with μ, otherwise μ-reduction could be applied to M_2. Since the type of M_3 is atomic, M_3 is constructed by application. By reasoning similar to that used above, M_3 is of the form $\alpha t N$. By the previous lemma, N is closed. Hence, S-reduction can be applied to $\mu\beta.[\beta]\alpha t N$. This contradicts the normality of M. Therefore, this case cannot occur. □

4. Strong normalization

This section is devoted to the proof of strong normalization. We first show that strong normalization for the full calculus can be reduced to strong normalization of reductions that do not involve use of the ζ_L-, ζ_1-, ζ_2-, and S-rules. As noted before, the ζ_L-, ζ_1-, and ζ_2-rules can be derived from the $\bullet$-, $\bullet_1$-, and $\bullet_2$-rules and the λ-, λ_1-, and λ_2-rules. In the case of the S-rules, we exploit the fact that S-reduction can be postponed until after other kinds of reduction have been carried out.

Let $\Rightarrow_*$ and $\Rightarrow_*^1$ denote reduction and one-step reduction without use of the S-, ζ_L-, ζ_1-, and ζ_2-rules.

Proposition 1. *Let M, N, K be $\lambda\mu$-terms, and assume that $M \Rightarrow_S^1 N \Rightarrow_*^1 K$. Then there is a $\lambda\mu$-term L such that $M \neq L$, $M \Rightarrow_* L$, and $L \Rightarrow_S K$. (Note that L could be identical to K.)*

Proof. M can be written as $E[\mu\alpha.[\alpha]M_1]$, with α not free in M_1. We will check all possible reduction rules R other than S, ζ_L, ζ_1, and ζ_2 (that is, we will set R to each such rule in turn, substituting its name for $*$ in $N \Rightarrow_*^1 K$, and then show that there exists $L \neq M$ such that $M \Rightarrow_* L \Rightarrow_S K$, where the $*$ in $M \Rightarrow_* K$ stands for any rule other than S, ζ_L, ζ_1, and ζ_2).

Case I. $R = \lambda$. In this case, N and K can be written as $E[M_1] = E'[(\lambda\beta.N_1)N_2]$ and $E'[N_1[N_2/\beta]]$, respectively. There are several possibilities for the relationship of M_1 to N_1 and N_2. First, we consider the case where M_1 contains the $\lambda\mu$-term $(\lambda\beta.N_1)N_2$. Let M_1 be $E''[(\lambda\beta.N_1)N_2]$. Then we can take L to be $E[\mu\alpha.[\alpha]E''[N_1[N_2/\beta]]]$. Next, we consider the case where N_1 or N_2 contains M_1. If N_1 contains M_1, let N_1 be $E''[M_1]$, in which case we can take L to be $E'[(E''[\mu\alpha.[\alpha]M_1])[N_2/\beta]]$. If N_2 contains M_1, let N_2 be $E''[M_1]$, in which case we can take L to be $E[N_1[E''[\mu\alpha.[\alpha]M_1]/\beta]]$. Finally, we consider the case where $M_1 = \lambda\beta.N_1$. The following reduction is possible.

$$\begin{array}{lll}(\mu\alpha.[\alpha]\lambda\beta.N_1)N_2 & \Rightarrow_\bullet & (\lambda\delta.\mu\alpha.[\mu\gamma.[\alpha](\gamma\delta)]\lambda\beta.N_1)N_2 \\ & \Rightarrow_\lambda & \mu\alpha.[\mu\gamma.[\alpha](\gamma N_2)]\lambda\beta.N_1 \\ & \Rightarrow_\mu & \mu\alpha.[\alpha](\lambda\beta.N_1)N_2 \\ & \Rightarrow_\lambda & \mu\alpha.[\alpha]N_1[N_2/\beta] \\ & \Rightarrow_S & N_1[N_2/\beta]\end{array}$$

Here, we can take L to be $E'[\mu\alpha.[\alpha]N_1[N_2/\beta]]$. These three cases cover all the possibilities in which M_1 and $(\lambda\beta.N_1)N_2$ have an overlap. If there is no overlap between M_1 and $(\lambda\beta.N_1)N_2$, the conclusion of the proposition is obvious.

Case II. $R = \lambda_1$ or λ_2. The proof of this case is similar to the case $R = \lambda$.

Case III. $R = \mu$. Assume that $N = E[M_1] = E'[[N_1]\mu\beta.N_2]$ and $K = E'[N_2[N_1/\beta]]$. (The proof for the case where N is of the form $E'[[\mu\beta.N_1]N_2]$ and K is of the form $E'[N_1[N_2/\beta]]$ is similar.) Just as in case I, there are four possibilities, corresponding to the relative positions of M, N_1, and N_2. Only for the case $M_1 = \mu\beta.N_2$ is the proof non-trivial. In this case, we can use K itself as L, since the following reduction is possible.

$$\begin{array}{lll}[N_1](\mu\alpha.[\alpha](\mu\beta.N_2)) & \Rightarrow_\mu & [N_1](\mu\beta.N_2) \\ & \Rightarrow_\mu & N_2[N_1/\beta]\end{array}$$

Case IV. $R = \zeta_R$. Assume that N is of the form $E'[N_1(\mu\beta.N_2)]$, and that K is of the form $E'[\mu\beta.N_2[\mu\gamma.[\beta]\gamma N_1/\beta]]$. As before, the only non-trivial case is that of $M_1 = \mu\beta.N_2$. Since the following reduction is possible, we can choose L to be $E'[\mu\alpha.[\alpha](\mu\beta.N_2[\mu\gamma.[\beta]N_1\gamma/\beta])]$.

$$\begin{array}{lll} N_1(\mu\alpha.[\alpha](\mu\beta.N_2)) & \Rightarrow_{\zeta_R} & \mu\alpha.[\mu\gamma.[\alpha](N_1\gamma)](\mu\beta.N_2) \\ & \Rightarrow_{\mu} & \mu\alpha.[\alpha](N_1(\mu\beta.N_2)) \\ & \Rightarrow_{\zeta_R} & \mu\alpha.[\alpha](\mu\beta.N_2[\mu\gamma.[\beta]N_1\gamma/\beta]) \\ & \Rightarrow_{S} & \mu\beta.N_2[\mu\gamma.[\beta]\gamma N_1/\beta] \end{array}$$

Case v. $R = \bullet$. Assume that $N = E'[\mu\beta.N_1]$ and that $\mu\beta.N_1$ has been reduced by application of the $\bullet$-rule to the outermost μ-symbol. Since either N_1 contains M_1 or M_1 contains $\mu\beta.N_1$, the conclusion of the proposition is obvious. The cases $R = \bullet_1$ and $R = \bullet_2$ are treated similarly. □

Proposition 2. *If a $\lambda\mu$-term M is strongly normalizable via $\Rightarrow_*$, then M is strongly normalizable via $\Rightarrow$.*

Proof. We prove the contrapositive, so assume that M is not strongly normalizable via $\Rightarrow$. By König's lemma, there is an infinite sequence $M \Rightarrow^1 M_1 \Rightarrow^1 M_2 \Rightarrow^1 \ldots$. For reasons stated earlier, we can assume that this sequence does not use the ζ_L-, ζ_1-, and ζ_2-rules. Since it is impossible that, from some point on, all these reductions are S-reductions (i.e., we cannot have $M_k \Rightarrow^1_S M_{k+1} \Rightarrow^1_S \cdots$ for some $k \geq 1$), there are infinitely many $\Rightarrow^1_*$ in the sequence. Using the proposition above, we can construct an arbitrarily long finite sequence of $\Rightarrow^1_*$ from M. Hence, M is not strongly normalizable via $\Rightarrow_*$. □

Next, we use the method of reducibility candidates to prove strong normalization via $\Rightarrow_*$. From this point on, $\Rightarrow^1_*$ and $\Rightarrow_*$ are written as $\Rightarrow^1$ and $\Rightarrow$ for simplicity. Our construction of reducibility follows the notion of *strong validity* introduced by Prawitz [15]. Strongly valid $\lambda\mu$-terms are either 1) $\lambda\mu$-terms that are constructed from strongly valid $\lambda\mu$-terms by introduction rules or 2) $\lambda\mu$-terms all of whose reduction sequences lead to $\lambda\mu$-terms that can be obtained as in 1). Strong validity of a $\lambda\mu$-term M with a free $\lambda\mu$-variable α of type C is defined by strong validity of $M[N/\alpha]$ for all strongly valid terms N of type C. For intuitionistic logic, this gives an inductive definition over construction of formulae. To adapt this idea, we had to specify what the introduction rules are. We decided that the μ-rule, as well as the λ-, λ_1-, and λ_2-rules, are the introduction rules for this purpose. For example, the $\lambda\mu$-term

$$\dfrac{\begin{array}{c}[\alpha : \bullet A] \\ \vdots \\ M : \bot\end{array}}{\mu\alpha.M : A} \tag{19}$$

is strongly valid if and only if M is strongly valid. M is strongly valid if and only if for all strongly valid terms N of type $\bullet A$, $M[N/\alpha]$ is strongly valid. There is circularity in our definition, since the notion of strongly validity over $\lambda\mu$-terms of type $\bullet A$ depends on the notion of strong validity over $\lambda\mu$-terms of type A. We resolve this circularity by transfinite induction up to the first uncountable ordinal, ω_1. We denote the set of $\lambda\mu$-variables and $\lambda\mu$-constants of type C as $\mathcal{V}_C$.

Definition 5.

(1) *For a set S of $\lambda\mu$-terms of type C, $Cl(S)$ is defined as the smallest set of $\lambda\mu$-terms of type C which satisfies the following conditions. (Cl is a closure operator.)*
   (a) $S, \mathcal{V}_C \subseteq Cl(S)$.
   (b) *Let Σ be either a $\lambda\mu$-term, a first-order term, or an abstraction term. If $M\Sigma$ is a $\lambda\mu$-term of type C and $N \in Cl(S)$ for all N such that $M\Sigma \Rightarrow^1 N$, then $M\Sigma \in Cl(S)$.*

   Note that if all the $\lambda\mu$-terms in S are strongly normalizable, then all the $\lambda\mu$-terms in $Cl(S)$ are strongly normalizable.
(2) *The set of strongly normalizable $\lambda\mu$-terms of type $\bot$ is also denoted by $\bot$.*
(3) *Let S be a set of $\lambda\mu$-terms of type A (resp. $\bullet A$). Then the set $\bullet S$ of $\lambda\mu$-terms of type $\bullet A$ (resp. A) is defined as*

$$\bullet S := \{\mu\alpha.M | \forall N \in S, M[N/\alpha] \in \bot\}. \tag{20}$$

   Note that if S is not empty and the type of $\lambda\mu$-terms of S is a proposition, then $\bullet S$ consists of strongly normalizable terms. This is because $\bullet, \bullet_1, \bullet_2$-reductions cannot be applied to the outermost μ of $\mu\alpha.M \in \bullet S$ and M is strongly normalizable by the definition of $\bullet$.
(4) *The operator $D(S)$ is defined as $Cl(S \cup \bullet\bullet S)$. Note that $\bullet\bullet$ is a monotone operator, as is D. For ordinals σ, we define D^σ as*

$$D^0(S) := S \tag{21}$$

$$D^\sigma(S) := D(\bigcup_{\tau<\sigma} D^\tau(S)). \tag{22}$$

   $D^{\omega_1}(S)$ is a fixed point of D, where ω_1 is the first uncountable ordinal. In particular, $\bullet\bullet D^{\omega_1}(S) \subseteq D^{\omega_1}(S)$.

Definition 6. *Let S_A and S_B be sets of $\lambda\mu$-terms of type A and B, respectively. Then the set $S_A \to S_B$ of $\lambda\mu$-terms of type $A \to B$ is defined by the following two-step process:*

$$L(S_A, S_B) := \{\lambda\alpha^A.M | \forall N \in S_A, M[N/\alpha^A] \in S_B\} \tag{23}$$

$$S_A \to S_B := D^{\omega_1}(L(S_A, S_B)) \tag{24}$$

Definition 7. *Let $(t_i)_{i\in I}$ be a non-empty family of first-order terms, and for each $i \in I$ let S_i be a set of $\lambda\mu$-terms of type $A[t_i/x]$. We define the set $\bigwedge_{\substack{i\in I\\ \text{1st-ord.}}} S_i$ of $\lambda\mu$-terms of type $\forall xA$ as follows.*

$$\prod_{\substack{i\in I\\ \text{1st-ord.}}} S_i := \{\lambda x.M | \forall i \in I, M[t_i/x] \in S_i\} \tag{25}$$

$$\bigwedge_{\substack{i\in I\\ \text{1st-ord.}}} S_i := D^{\omega_1}(\prod_{\substack{i\in I\\ \text{1st-ord.}}} S_i). \tag{26}$$

Though A does not appear in the notation, the definition of $\bigwedge_{\substack{i\in I\\ \text{1st-ord.}}} S_i$ may depend on the choice of A and x.

Definition 8. *Let $(T_i)_{i\in I}$ be a non-empty family of abstraction terms, and for each $i \in I$ let S_i be a set of $\lambda\mu$-terms of type $A[T_i/X]$. We define the set $\bigwedge_{\substack{i\in I\\ \text{abstr.}}} S_i$ of* *λμ-terms of type $\forall XA$ as follows.*

$$\prod_{\substack{i\in I\\ \text{abstr.}}} S_i \quad := \quad \{\lambda X.M | \forall i \in I, M[T_i/X] \in S_i\} \tag{27}$$

$$\bigwedge_{\substack{i\in I\\ \text{abstr.}}} S_i \quad := \quad D^{\omega_1}(\prod_{\substack{i\in I\\ \text{abstr.}}} S_i). \tag{28}$$

As in the previous definition, the definition of $\bigwedge_{\substack{i\in I\\ \text{abstr.}}} S_i$ may depend on the choice of A and X.

Definition 9. *For a proposition A, $\mathbf{R}_A$ is the smallest set which satisfies the conditions stated below. If $\mathcal{R} \in \mathbf{R}_A$, we say that $\mathcal{R}$ is a reducibility candidate of proposition A. In keeping with our distinction between types and propositions, reducibility candidates are defined only for propositions.*

1. *If A is atomic or of the form $Xt_1 \cdots t_n$, then $D^{\omega_1}(\mathcal{V}_A) \in \mathbf{R}_A$.*
2. *If $\mathcal{A} \in \mathbf{R}_A$ and $\mathcal{B} \in \mathbf{R}_B$, then $\mathcal{A} \to \mathcal{B} \in \mathbf{R}_{A\to B}$.*
3. *Let $(t_i)_{i\in I}$ be a non-empty family of first-order terms, and for each $i \in I$ let $\mathcal{R}_i$ be a reducibility candidate of type $A[t_i/x]$. Then $\bigwedge_{\substack{i\in I\\ \text{1st-ord.}}} \mathcal{R}_i \in \mathbf{R}_{\forall xA}$.*
4. *Let $(T_i)_{i\in I}$ be a non-empty family of abstraction terms, and for each $i \in I$ let $\mathcal{R}_i$ be a reducibility candidate of proposition $A[T_i/X]$. Then $\bigwedge_{\substack{i\in I\\ \text{abstr.}}} \mathcal{R}_i \in \mathbf{R}_{\forall XA}$.*

Note that a reducibility candidate $\mathcal{R}$ can be written as $D^{\omega_1}(S)$ for a set S of λμ-terms such that no element of S begins with μ.

Proposition 3. *Let $\mathcal{R}$ be a reducibility candidate.*

1. *$\mathcal{R}$ is non-empty, and all λμ-terms in $\mathcal{R}$ are strongly normalizable.*
2. *If $M \in \mathcal{R}$ and $M \Rightarrow_1 N$, then $N \in \mathcal{R}$.*
3. *For $M \in \bullet\mathcal{R}$ and $N \in \mathcal{R}$, $[M]N \in \bot$.*

The proof of the proposition uses induction on the construction of the candidate $\mathcal{R}$. Each step of the induction is broken up into Lemmata 4, 5, 6, and 7. Before proceeding with the proof of clauses 1 and 2 of the proposition, we show that clause 3 can be derived from clauses 1 and 2.

Lemma 2. *Let S be a set of λμ-terms which does not begin with μ. We assume that all terms contained in S has a same type. Let $P = D^{\omega_1}(S)$. Then we have $P = Cl(S \cup \bullet\bullet P)$.*

Proof. The inclusion from right to left is immediate, by the closure property of Cl and the fact that $\bullet\bullet P \subseteq P$ and $S = D^0(S) \subseteq D^{\omega_1}(S) = P$. To see the inclusion from left to right, we let $R = Cl(S \cup \bullet\bullet P)$ and prove that $D^\sigma(S) \subseteq R$ for all ordinals $\sigma \leq \omega_1$. The proof is by induction on σ. Clearly, $D^0(S)$ $(=S) \subseteq R$, so the base case holds. Let σ be an ordinal such that $0 < \sigma \leq \omega_1$. By the induction hypothesis, $D^\tau(S) \subseteq R$ for all $\tau < \sigma$. Let Q be $\bigcup_{\tau<\sigma} D^\tau(S)$. Then we have $Q \subseteq P$. By the monotonicity of the $\bullet\bullet$ operator, $\bullet\bullet Q \subseteq \bullet\bullet P$. Since $\bullet\bullet P \subseteq R$, we have $\bullet\bullet Q \subseteq R$. Hence $Q \cup \bullet\bullet Q \subseteq R$. By the closure property of Cl, $Cl(Q \cup \bullet\bullet Q) \subseteq R$, so $D^\sigma(S) = D(Q) = Cl(Q \cup \bullet\bullet Q) \subseteq R$. □

In the remainder of the paper, we sometimes use Lemma 2 without mention.

Lemma 3. *If a reducibility candidate $\mathcal{R}$ satisfies clauses 1 and 2 of Proposition 3, then for all $M \in \bullet\mathcal{R}$ and $N \in \mathcal{R}$, $[M]N \in \perp$.*

Proof. As noted in Definition 9, $\mathcal{R}$ can be written as $D^{\omega_1}(S)$ for a set S of $\lambda\mu$-terms such that no element of S begins with μ.

Assume $M \in \bullet\mathcal{R}$ and $N \in \mathcal{R}$. To complete the proof of the lemma, it suffices to show that every K such that $[M]N \Rightarrow^1 K$ is strongly normalizable. We consider all possibilities for the reduction of $[M]N$. We use induction on $w(M) + w(N)$. N is strongly normalizable by the clause 1 of Proposition 3. M is strongly normalizable since $\mathcal{R}$ is not empty. (See the clause 3 of Definition 5.)

Case I. K is of the form $[M']N'$, where $M \Rightarrow M'$ and $N \Rightarrow N'$. In this case, $M' \in \bullet\mathcal{R}$ by the definition of $\bullet$, and $N' \in \mathcal{R}$ by clause 2 of Proposition 3. By the induction hypothesis, $K \in \perp$, since $w(M') + w(N') < w(M) + w(N)$.

Case II. $M \equiv \mu\alpha.M_1$ and $K \equiv M_1[N/x]$. Since $M \in \bullet\mathcal{R}$ and $N \in \mathcal{R}$, $M_1[N/x] \in \perp$ by the definition of $\bullet$.

Case III. $N \equiv \mu\alpha.N_1$ and $K \equiv N_1[M/\alpha]$. Since $\mathcal{R} = Cl(S \cup \bullet\bullet\mathcal{R})$ and S does not contain a $\lambda\mu$-term beginning with μ, N must be an element of $\bullet\bullet\mathcal{R}$. Hence $N_1[M/\alpha] \in \perp$. □

Lemma 4. *If $\mathcal{R} = D^{\omega_1}(\mathcal{V}_A)$ for an atomic proposition A, then Proposition 3 holds.*

Proof. Non-emptiness and strong normalizability are easy. To prove the second clause, let σ be the least ordinal such that $M \in D^{\sigma}(\mathcal{V}_A)$. The proof is by induction on σ.

If $\sigma = 0$, then M is a $\lambda\mu$-variable and the conclusion of the lemma follows. If $\sigma > 0$, then either M has a form of application or $M \in \bullet\bullet\bigcup_{\sigma_1<\sigma} D^{\sigma_1}(\mathcal{V}_A)$. By the definitions of Cl and $\bullet$, the conclusion of the lemma holds in either case. (Note that $\bullet$-, $\bullet_1$-, and $\bullet_2$-reduction cannot be applied to the outermost μ.) □

Lemma 5. *Let $\mathcal{A}, \mathcal{B}$ be reducibility candidates that satisfy all three clauses of Proposition 3. Then $\mathcal{R} = \mathcal{A} \to \mathcal{B}$ is non-empty and the following hold of all $M \in \mathcal{R}$.*

(1) *M is strongly normalizable.*
(2) *If $M \Rightarrow^1 N$, then $N \in \mathcal{R}$.*
(3) *If $N \in \mathcal{A}$, then $MN \in \mathcal{B}$.*

Proof. Again, non-emptiness is easy. Let $D(\sigma)$ be $D^{\sigma}(L(\mathcal{A}, \mathcal{B}))$. We prove, by induction on ordinals $\sigma \leq \omega_1$, that for all $M \in D(\sigma)$, M is strongly normalizable and if $M \Rightarrow_1 N$, then $N \in D(\sigma)$. We show further that $MN \in \mathcal{B}$ for every $N \in \mathcal{A}$.

If $\sigma = 0$, then the conclusion follows from the definition of $L(\mathcal{A}, \mathcal{B})$ and the facts that $\mathcal{A}$ is not empty and $\mathcal{B}$ satisfies clauses 1 and 2 of Proposition 3. Assume $\sigma > 0$. First we prove that M is strongly normalizable and if $M \Rightarrow_1 N$, then $N \in D(\sigma)$. If we prove this for the case where $M \in \bullet\bullet\bigcup_{\sigma_1<\sigma} D(\sigma_1)$, the conclusion for the general case follows from the definition of Cl. Hence assume that M is of the form $\mu\alpha.M_1$. We denote $\bigcup_{\sigma_1<\sigma} D(\sigma_1)$ by S.

M_1 is strongly normalizable since $\bullet S$ is not empty. We further use induction on $w(M_1)$ and show that each $\lambda\mu$-term M' such that $M \Rightarrow^1 M'$ is strongly normalizable and that $M' \in D(\sigma)$. If the reduction does not consist of a $\bullet$-rule applied to the outermost μ, then $M' \equiv \mu\alpha.M_1' \in \bullet\bullet S \subseteq D(\sigma)$. By the induction hypothesis (for the induction on $w(M_1)$) and the fact that $w(M_1') < w(M_1)$, M' is strongly normalizable. Hence assume that M is reduced by the $\bullet$-rule applied to the outermost μ. Then $M' = \lambda\beta.\mu\gamma.M_1[\mu\delta.[\gamma]\delta\beta/\alpha]$. To prove that $M' \in D(\sigma)$ and M' is strongly normalizable, choose some $\sigma_1 < \sigma$ and let $N \in D(\sigma_1)$. In addition, let $L \in \mathcal{A}$ and $K \in \bullet\mathcal{B}$. Note that $NL \in \mathcal{B}$, by the induction hypothesis (for the induction on σ). Hence $[K]NL \in \bot$. Since $\sigma_1 < \sigma$ and N was an arbitrary element of $D(\sigma_1)$, we have $\mu\delta.[K]\delta L \in \bullet S$. Using the hypothesis that $\mu\alpha.M_1 \in \bullet\bullet S$, we see that $M_1[\mu\delta.[K]\delta L/\alpha] \in \bot$. Since $K \in \bullet\mathcal{B}$, we get $\mu\gamma.M_1[\mu\delta.[\gamma]\delta L/\alpha] \in \bullet\bullet\mathcal{B} \subseteq \mathcal{B}$. This means that $M' \in L(\mathcal{A},\mathcal{B})$, hence that M' is strongly normalizable and $M' \in \mathcal{D}(\sigma)$.

Next, we prove that $MN \in \mathcal{B}$ for each $N \in \mathcal{A}$. As noted in Definition 9, $\mathcal{A} = D^{\omega_1}(X)$ for some set X of $\lambda\mu$-terms. We can assume that X does not contain a $\lambda\mu$-term beginning with μ. Let τ be the least ordinal such that $N \in D^{\tau}(X)$. By induction on τ and $w(M) + w(N)$, we will prove that if $MN \Rightarrow^1 L$, then $L \in \mathcal{B}$. This is precisely the condition that $MN \in \mathcal{B}$.

Case I. $L \equiv M'N'$, and either ($M \Rightarrow^1 M'$ and $N \equiv N'$) or ($M \equiv M'$ and $N \Rightarrow^1 N'$). The conclusion follows from the induction hypothesis (for the induction on $w(M) + w(N)$).

Case II. $M \equiv \lambda\alpha.M_1$ and $L \equiv M_1[N/\alpha]$. Since $M \in L(\mathcal{A},\mathcal{B})$, the conclusion follows.

Case III. N is of the form $\mu\alpha.N_1$ and L is obtained from reduction of the outermost redex. Then L is of the form $\mu\beta.N_1[\mu\gamma.[\beta](M\gamma)/\alpha]$. Choose some $\tau_1 < \tau$, and let $Q \in D^{\tau_1}(X)$. Furthermore, let $P \in \bullet\mathcal{B}$. By the induction hypothesis (for the induction on τ), we have $MQ \in \mathcal{B}$. By reasoning similar to that used above, it follows that $\mu\gamma.[P](M\gamma) \in \bullet\bigcup_{\tau_1<\tau} D^{\tau_1}(X)$. Since N begins with μ, $N \in \bullet\bullet\bigcup_{\tau_1<\tau} D^{\tau_1}(X)$. We thus have $N_1[\mu\gamma.[P](M\gamma)/\alpha] \in \bot$, hence that $L \in \bullet\bullet\mathcal{B} \subseteq \mathcal{B}$. □

Lemma 6. *Let I be a non-empty index set such that for each $i \in I$, $\mathcal{R}_i$ is a reducibility candidate of proposition $A[t_i/x]$, and let $\mathcal{R} = \bigwedge_{\substack{i\in I\\ \mathrm{1st-ord.}}} \mathcal{R}_i$. Assume that for each $i \in I$, $\mathcal{R}_i$ satisfies all the clauses of Proposition 3. Then $\mathcal{R}$ is non-empty, and the following hold of all $M \in \mathcal{R}$.*

1. *M is strongly normalizable.*
2. *If $M \Rightarrow^1 N$, then $N \in \mathcal{R}$.*
3. *$Mt_i \in \mathcal{R}_i$.*

Proof. Again, non-emptiness is easy. Let $D(\sigma)$ be $D^{\sigma}(\prod_{\substack{i\in I\\ \mathrm{1st-ord.}}} \mathcal{R}_i)$. We prove, by induction on $\sigma \leq \omega_1$, that for all $M \in D(\sigma)$, M is strongly normalizable and if $M \Rightarrow_1 N$, then $N \in D(\sigma)$. We show further that $Mt_i \in \mathcal{R}_i$ for each $i \in I$.

If $\sigma = 0$, then the conclusion follows from the facts that I is not empty and for each $i \in I$, $\mathcal{R}_i$ satisfies all the clauses of Proposition 3. The fact that $Mt_i \in \mathcal{R}_i$ for each $i \in I$ is obtained from the definition of $\prod_{\substack{i\in I\\ \mathrm{1st-ord.}}} \mathcal{R}_i$. Assume $\sigma > 0$. First we prove that M is strongly normalizable and if $M \Rightarrow_1 N$, then $N \in D(\sigma)$. If we prove this for the case where $M \in \bullet\bullet\bigcup_{\sigma_1<\sigma} D(\sigma)$, the conclusion for the general case follows from the definition of Cl. Hence assume that M is of the form $\mu\alpha.M_1$. We denote $\bigcup_{\sigma_1<\sigma} D(\sigma_1)$ by S.

Since all $\lambda\mu$-terms in S are strongly normalizable by the induction hypothesis, M_1 is strongly normalizable. We further use induction on $w(M_1)$ and show that every M' such that $M \Rightarrow^1 M'$ is strongly normalizable and that $M' \in D(\sigma)$. If the reduction rule is not the $\bullet_1$-rule for the outermost μ, then $M' \equiv \mu\alpha.M_1' \in \bullet\bullet S \subseteq \mathcal{D}(\sigma)$. By the induction hypothesis (for the induction on $w(M_1)$) and the fact that $w(M_1') < w(M_1)$, M' is strongly normalizable. Hence assume that M is reduced by the $\bullet_1$-rule for the outermost μ. Then $M' = \lambda x.\mu\gamma.M_1[\mu\delta.[\gamma]\delta x/\alpha]$. Choose some $\sigma_1 < \sigma$, and let $N \in D(\sigma_1)$. Also, let $K \in \bullet\mathcal{R}_i$. By the induction hypothesis (for the induction on σ), $Nt_i \in \mathcal{R}_i$; hence, $[K]Nt_i \in \bot$. Since $\sigma_1 < \sigma$ and $N \in D(\sigma_1) \subseteq S$, we have $\mu\delta.[K]\delta t_i \in \bullet S$. Furthermore, $M_1[\mu\delta.[K]\delta t_i/\alpha] \in \bot$, because $\mu\alpha.M_1 \in \bullet\bullet S$. Since $K \in \bullet\mathcal{R}_i$, we see that $\mu\gamma.M_1[\mu\delta.[\gamma]\delta t_i/\alpha] \in \bullet\bullet\mathcal{R}_i \subseteq \mathcal{R}_i$. This means that $M' \in \prod_{\substack{i\in I \\ \text{1st-ord.}}} \mathcal{R}_i$, hence that M' is strongly normalizable and $M' \in D(\sigma)$.

Next, we prove that $Mt_i \in \mathcal{R}_i$. We will show that $N \in \mathcal{R}_i$ for all N such that $Mt_i \Rightarrow^1 N$. The proof is by induction on $w(M)$.

Case I. $N \equiv M't_i$ and $M \Rightarrow^1 M'$. By the induction hypothesis and the fact that $w(M') < w(M)$, the conclusion follows.

Case II. $M \equiv \lambda x.M_1$ and $N \equiv M_1[t_i/x]$. Since $M \in \prod_{\substack{i\in I \\ \text{1st-ord.}}} \mathcal{R}_i$, we have the conclusion.

There are no other possibilities for M and N. □

Lemma 7. *Let I be a non-empty index set such that for each $i \in I$, $\mathcal{R}_i$ is a reducibility candidate of proposition $A[T_i/X]$, and let $\mathcal{R} = \bigwedge_{\substack{i\in I \\ \text{abstr.}}} \mathcal{R}_i$. Assume that for each $i \in I$, $\mathcal{R}_i$ satisfies all three clauses of Proposition 3. Then $\mathcal{R}$ is non-empty, and the following hold of all $M \in \mathcal{R}$.*

1. *M is strongly normalizable.*
2. *If $M \Rightarrow^1 N$, then $N \in \mathcal{R}$.*
3. *$MT_i \in \mathcal{R}_i$.*

Proof. Analogous to the proof of the previous lemma. □

Proof. Proof of Proposition 3 The proof is by induction on the construction of $\mathcal{R}$, together with Lemmata 4, 5, 6, and 7. □

Using Proposition 3 and Lemmata 4, 5, 6 and 7, we can prove strong normalization by the method of reducibility candidates.

Definition 10. *For each abstraction term $T = \lambda x_1 \cdots x_n A$, a* complex *of kind T is a map which sends n-tuples of first-order terms $\langle t_1, \dots, t_n\rangle$ to elements of $\mathbf{R}_{A[t_1/x_1,\dots,t_n/x_n]}$. $\mathbf{C}_T$ denotes the set of all complexes of kind T. $\mathbf{C}^n$ denotes the union of $\mathbf{C}_T$ over all n-ary abstraction terms T.*

Definition 11. *Let ξ be a map which sends first-order variables to first-order terms, and n-ary predicate variables to elements of $\mathbf{C}^n$. We call such a map an* interpretation. *We extend ξ to all first-order terms t by*

$$\xi(t) = t[\xi(x_1)/x_1, \dots, \xi(x_n)/x_n], \tag{29}$$

where $\{x_1, \dots, x_n\} = FV(t)$. Moreover, we extend ξ to arbitrary types and abstraction terms as follows. Let $\bot$ be the set of all strongly normalizable $\lambda\mu$-terms of type $\bot$, let $\mathcal{V}_A$ be the set of all $\lambda\mu$-variables and constants of type A, and let $\mathcal{T}$ be the *set of all first-order terms.*

$$\xi(t_1 = t_2) = D^{\omega_1}(\mathcal{V}_{\xi(t_1)=\xi(t_2)}) \tag{30}$$

$$\xi(Xt_1 \cdots t_n) = \xi(X) < \xi(t_1), \dots, \xi(t_n) > \tag{31}$$

$$\xi(A \to B) = \xi(A) \to \xi(B) \tag{32}$$

$$\xi(\forall x A) = \bigwedge_{\substack{t \in \mathcal{T} \\ \text{1st-ord.}}} \xi[t/x](A) \tag{33}$$

$$\xi(\forall X^n A) = \bigwedge_{\substack{\mathcal{C} \in \mathbf{C}^n \\ \text{abstr.}}} \xi[\mathcal{C}/X^n](A) \tag{34}$$

$\xi(\bot)$ *and* $\xi(\bullet A)$ *for a proposition* A *are defined as* $\xi(\bot) = \bot$ *and* $\xi(\bullet A) = \bullet\xi(A)$ *respectively.*

$\xi[t/x]$ *is defined in such a way that for all first-order variables except* x*, it gives the same results as* ξ*; and for* x*,* $\xi[t/x](x) = t$*. The definition of* $\xi[\mathcal{C}/X^n]$ *is similar.*

$\xi(\lambda x_1 \cdots x_n A)$ *is defined as the complex which sends the tuple* $\langle t_1, \dots, t_n \rangle$ *to* $\xi[t_1/x_1, \dots, t_n/x_n](A)$.

In what follows, we use the abbreviation $\overline{e}$ for a finite sequence $e_1, \dots, e_n$.

Proposition 4. *Let* M *be a* $\lambda\mu$*-term of type* C *with free first-order variables* $x_1, \dots, x_l$*, free predicate variables* $X_1, \dots, X_m$*, and free* $\lambda\mu$*-variables* $\alpha_1^{C_1}, \dots, \alpha_n^{C_n}$*. Let* ξ *be an interpretation, and let* t_i *be* $\xi(x_i)$ *for* $i = 1, \dots, l$*. Assume that for each* j *with* $1 \le j \le m$*,* $\xi(X_j) \in \mathbf{C}_{T_j}$ *for some abstraction term* T_j*. Choose* $N_k \in \xi(C_k)$ *for* $k = 1, \dots, n$*. Then we have*

$$M[\,\overline{t}/\overline{x}][\overline{T}/\overline{X}][\overline{N}/\overline{\alpha}] \in \xi(C). \tag{35}$$

Proof. By induction on the construction of M.

Case I. $M \equiv a_i$ (an axiom). Since $\xi(C) = Cl(S \cup \bullet\bullet\,\xi(C))$ for some S and by definition of Cl, $a_i \in \xi(C)$.

Case II. $M \equiv \alpha_i$. The conclusion of the proposition is clear from the assumption that $N_i \in \xi(C_i)$.

Case III. $M \equiv \mu\alpha.M_1$. Here, $M_1[\,\overline{t}/\overline{x}][\overline{T}/\overline{X}][\overline{N}/\overline{\alpha}, N/\alpha] \in \bot$ for $N \in \xi(\bullet C)$, by the induction hypothesis. Hence we have $\mu\alpha.M_1[\,\overline{t}/\overline{x}][\overline{T}/\overline{X}][\overline{N}/\overline{\alpha}] \in \xi(C)$.

Case IV. $M \equiv [M_1]M_2$. The conclusion of the proposition follows from Proposition 3.

Case V. $M \equiv \lambda\alpha.M_1$, $M \equiv \lambda x.M_1$, or $M \equiv \lambda X.M_1$. The conclusion of the proposition follows from the construction of ξ. For example, let $M \equiv \lambda X.M_1$. Then C is of the form $\forall X A_1$. Let T be an abstraction term, and let $\mathcal{C}$ be a complex of type T. By the induction hypothesis applied to M_1, $M_1[\,\overline{t}/\overline{x}][\overline{T}/\overline{X}, T/X][\overline{N}/\overline{\alpha}] \in \xi[\mathcal{C}/X](A_1)$. Note that since the types of the individual components of $\overline{\alpha}$ do not contain X as a free predicate variable, substitution of T for X does not alter the types of the $\lambda\mu$-variables in $\overline{\alpha}$. By renaming the bound predicate variable X in M, we can safely assume that $\overline{N}$ does not contain a free occurrence of X. Hence, we can infer that $M_1[\,\overline{t}/\overline{x}][\overline{T}/\overline{X}][\overline{N}/\overline{\alpha}][T/X] \in \xi[\mathcal{C}/X](A_1)$. The conclusion follows from the definition of $\xi(\forall X A_1)$.

Case VI. $M \equiv M_1 M_2$, $M \equiv M_1 t$, or $M \equiv M_1 T$. The conclusion of the proposition follows from Lemmata 5, 6, and 7. For example, consider the case $M \equiv M_1 T$. Then $C \equiv A'[T/X^r]$ and M_1 is of type $\forall X^r A'$. (r is the arity of X.) Let $\tilde{T}$ be $T[\overline{t}/\overline{x}][\overline{T}/\overline{X}]$, and let $\tilde{M_1}$ be $M_1[\,\overline{t}/\overline{x}][\overline{T}/\overline{X}][\overline{N}/\overline{\alpha}]$. By the induction hypothesis,

$M_1 \in \bigwedge_{\mathcal{C}\in\mathbf{C}^r_{\text{abstr.}}} \xi[\mathcal{C}/X](A')$. By Lemma 7, we have $\tilde{M}_1\tilde{T} \in \xi[\xi(T)/X](A')$. Since $\tilde{M}_1\tilde{T} = M_1T[\overline{t}/\overline{x}][\overline{T}/\overline{X}][\overline{N}/\overline{\alpha}]$ and $\xi[\xi(T)/X](A') = \xi(A'[T/X])$, the conclusion of the proposition follows. □

In the above proposition, choose ξ so that $\xi(x_i) = x_i$ for $i = 1, \ldots, l$, and let $\xi(X_j) \in \mathbf{C}_{\lambda x_1 \cdots x_{k_j}.X_j x_1 \cdots x_{r_j}}$ for $j = 1, \ldots, m$, where r_j is the arity of X_j. Further, let N_k be α_k for $k = 1, \ldots, n$. Then we have $M \in \xi(C)$. By Proposition 3, we have the following theorem.

Theorem 2. *All $\lambda\mu$-terms are strongly normalizable.*

I am grateful to Ken-etsu Fujita, Ryu Hasegawa, Charles Stewart, and Makoto Tatsuta for their helpful comments and discussion. I also wish to thank Georgia Martin for carefully reading the manuscript and helping to improve the presentation.

References

[1] Yuuki Andou. A normalization-procedure for the first order classical natural deduction with full symbols. *Tsukuba Journal of Mathematics*, 19(1):153–162, 1995.

[2] F. Barbanera and S. Berardi. A strong normalization result for classical logic. *Ann. Pure Appl. Logic*, 76:99–116, 1995.

[3] F. Barbanera and S. Berardi. A symmetric lambda calculus for "classical" program extraction. *Inf. Comput.*, 125(2):103–117, 1996.

[4] V. Danos, J. B. Joinet, and H. Schellinx. A new deconstructive logic: linear logic. *J. Symb. Log.*, 62(3):755–807, 1997.

[5] Ph. de Groote. A cps-translation of the $\lambda\mu$-calculus. In *Trees in algebra and programming, CAAP '94*, number 787 in Lect. Notes Comput. Sci, pages 85–99. Springer-Verlag, 1994.

[6] Ph. de Groote. On the relation between $\lambda\mu$-calculus and the syntactic theory of sequential control. In *Logic programming and automated reasoning*, volume 822 of *Lect. Notes Comput. Sci*, pages 31–43. Springer-Verlag, 1994.

[7] J. Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1988.

[8] Koji Nakazawa. Confluency and strong normalizability of call-by-value $\lambda\mu$-calculus. *Theoret. Comput. Sci.*, 290:429–463, 2003.

[9] Karim Nour. La valeur d'un entier classique en $\lambda\mu$-calcul. *Arch. Math. Logic*, 36:461–473, 1997.

[10] C.-H. L. Ong and C. A. Stewart. A curry-howard foundation for functional computation with control. In *Proceedings of the 24th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 215–217. ACM press, 1997.

[11] M. Parigot. $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In A. Voronkov, editor, *Logic Programming and Automated Reasoning*, volume 624 of *Lecture Notes in Artificial Intelligence*, pages 190–201. Springer-Verlag, 1992.

[12] M. Parigot. Classical proofs as programs. In *Computational logic and proof theory*, volume 713 of *Lect. Notes Comput. Sci*, pages 263–276. Springer-Verlag, 1993.

[13] M. Parigot. Strong normalization for second order classical natural deduction. *J. Symb. Log.*, 62(4):1461–1479, 1997.

[14] M. Parigot. On the computational interpretation of negation. In P. Clote and H. Schwichtenberg, editors, *Computer Science Logic*, volume 1862 of *Lect. Notes Comput. Sci*, pages 472–484. Springer-Verlag, 2000.

[15] Dag Prawitz. Idea and results of proof theory. In *Proceedings of the Second Scandinavian Logic Symposium*, pages 237–309. North-Holland, 1971.

[16] Charles A. Stewart. On the formulae-as-types correspondence for classical logic. DPhil. thesis, Oxford University, 2000.

[17] Th. Streicher and B. Reus. Classical logic, continuation semantics and abstract machines. *Journal of Functional Programming*, 8(6):543–572, 1998.

National Institute of Advanced Industrial Science and Technology, Osaka, Japan
E-mail address: yoriyuki.yamagata@aisg.go.jp