Koji Nakazawa
Hiroto Naya

Strong Reduction of Combinatory Calculus with Streams

Abstract. This paper gives the strong reduction of the combinatory calculus SCL, which was introduced as a combinatory calculus corresponding to the untyped Lambda-mu calculus. It proves the confluence of the strong reduction. By the confluence, it also proves the conservativity of the extensional equality of SCL over the combinatory calculus CL, and the consistency of SCL.

Keywords: combinatory logic, lambda-mu calculus, strong reduction, confluence

1. Introduction

The $\lambda\mu$-calculus was introduced by Parigot [7] as a term assignment for the classical natural deduction, and it has been widely studied because of its relationship to calculi with continuations and control operators. An untyped variant of the $\lambda\mu$-calculus, called the $\Lambda\mu$-calculus, is also interesting since Saurin showed that it enjoys Böhm's separation theorem and the standardization theorem [8, 9].

For the $\Lambda\mu$-calculus, an extension of the concept of λ-model, called *stream model*, and its algebraic characterization as an extension of the concept of combinatory algebra were given in [6]. The extended algebras were induced from the combinatory calculus SCL, which is an extension of the well-known combinatory calculus CL with extra syntax representing streams and additional combinators manipulating streams. They proved in [6] that SCL is equivalent to the $\Lambda\mu$-calculus.

This paper studies the syntactic structure of SCL. In particular, we extend the *strong reduction* of CL to SCL and prove its confluence. By the confluence, it is proved that the extensional equality of SCL is conservative over CL. The conservativity directly gives a syntactic proof of the consistency of the equational logic of SCL.

2. Strong Reduction of SCL

In this section, we will give the definition of SCL and a reduction relation on it, which is an extension of the strong reduction of the combinatory logic CL [3].

2.1. Combinatory Calculus SCL

We define the equational logic of SCL given in [6]. They showed that SCL is equivalent to the $\Lambda\mu$-calculus. The correspondence between SCL and the $\Lambda\mu$-calculus will be discussed in the next section in a stronger form than the result in [6].

Definition 2.1. 1. It is supposed that there are two sorts of variables: *term variables* ($x, y, \cdots \in \mathsf{Var_T}$) and *stream variables* ($\alpha, \beta, \cdots \in \mathsf{Var_S}$). The constants of SCL are

$$C ::= \mathsf{K}_0 \mid \mathsf{K}_1 \mid \mathsf{S}_0 \mid \mathsf{S}_1 \mid \mathsf{W}_1 \mid \mathsf{C}_{10} \mid \mathsf{C}_{11},$$

and *pre-terms* and *pre-streams* of SCL are defined by

$$T, U ::= C \mid x \mid T \cdot U \mid T \star \mathcal{S} \qquad\qquad \mathcal{S} ::= \alpha \mid T :: \mathcal{S}.$$

The binary function symbols $(\cdot)$ and $(\star)$ have the same associative strength, and both are left associative. For simplicity, these function symbols are often omitted like in the combinatory logic. For example, $T_1T_2\mathcal{S}_3T_4$ denotes $((T_1 \cdot T_2) \star \mathcal{S}_3) \cdot T_4$. Note that it does not make ambiguity since two kinds of variables are syntactically distinguished. The set of variables occurring in T is denoted by $FV(T)$. The set of *terms* (*streams*, respectively), which is denoted by $\mathsf{Term_{SCL}}$ ($\mathsf{Stream_{SCL}}$), is defined as the quotient of the pre-terms (pre-streams) by the least compatible equivalence such that

$$T \cdot U \star \mathcal{S} \equiv_{\mathsf{assoc}} T \star (U :: \mathcal{S}).$$

The set of terms containing no stream variable is denoted by $\mathsf{Term}^0_{\mathsf{SCL}}$. We also use the metavariables $T, U, \cdots$ for terms, and $\mathcal{S}, \cdots$ for streams. The substitutions $T[x := U]$ and $T[\alpha := \mathcal{S}]$ are defined in an usual way.

2. The *extensional equality* $=_{\mathsf{SCL}}$ on SCL-terms is the least compatible equivalence relation satisfying the following axioms and rules.

$$\begin{aligned}
\mathsf{K}_0T_1T_2 &=_{\mathsf{SCL}} T_1 & \qquad \mathsf{K}_1T_1\mathcal{S}_2 &=_{\mathsf{SCL}} T_1 \\
\mathsf{S}_0T_1T_2T_3 &=_{\mathsf{SCL}} T_1T_3(T_2T_3) & \qquad \mathsf{S}_1T_1T_2\mathcal{S}_3 &=_{\mathsf{SCL}} T_1\mathcal{S}_3(T_2\mathcal{S}_3) \\
\mathsf{C}_{10}T_1\mathcal{S}_2T_3 &=_{\mathsf{SCL}} T_1T_3\mathcal{S}_2 & \qquad \mathsf{C}_{11}T_1\mathcal{S}_2\mathcal{S}_3 &=_{\mathsf{SCL}} T_1\mathcal{S}_3\mathcal{S}_2 \\
\mathsf{W}_1T_1\mathcal{S}_2 &=_{\mathsf{SCL}} T_1\mathcal{S}_2\mathcal{S}_2
\end{aligned}$$

$$\frac{Tx =_{\mathsf{SCL}} Ux \quad x \notin FV(T) \cup FV(U)}{T =_{\mathsf{SCL}} U} \ (\zeta_T)$$

$$\frac{T\alpha =_{\mathsf{SCL}} U\alpha \quad \alpha \notin FV(T) \cup FV(U)}{T =_{\mathsf{SCL}} U} \ (\zeta_S)$$

3. The combinatory calculus CL is the subsystem of SCL defined by restricting the syntax as

$$C ::= \mathsf{K}_0 \mid \mathsf{S}_0, \qquad\qquad T, U ::= C \mid x \mid TU,$$

and the extensional equality of CL is defined by the axioms for K_0 and S_0 and the rule ζ_T. The set of terms of CL is denoted by $\mathsf{Term}_{\mathsf{CL}}$.

It is easy to see that every SCL-term is uniquely represented as a pre-term without $(::)$ by normalizing with respect to the assoc reduction $T(U :: \mathcal{S}) \to_{\mathsf{assoc}} TU\mathcal{S}$ on pre-terms. It is also easy to see that the equivalence $\equiv_{\mathsf{assoc}}$ is decidable in linear time by comparing the $\to_{\mathsf{assoc}}$-normal forms, which are characterized as $T ::= C \mid x \mid TU \mid T\alpha$. In the following some translations on $\mathsf{Term}_{\mathsf{SCL}}$ are defined as translations on the $\to_{\mathsf{assoc}}$-normal forms.

In the axioms and rules of the extensional equality, T_i and $\mathcal{S}_i$ are treated as pattern variables. Note that, when we define the patterns as pre-terms and pre-streams with pattern variables each of which occurs at most once, any pattern matching has at most one solution and is decidable in linear time.

2.2. Strong Reduction

The strong reduction of CL is defined from the weak reduction and the additional rule (ξ_T), which will be given below. For SCL, the weak reduction will be extended to the new combinators and the rule (ξ_S) for streams will be also introduced. However, it is not sufficient, and we need some extra rules, which will be given as the ϕ- and σ-rules corresponding to the rule called (fst) in [8] and propagation of structural substitutions of the $\Lambda\mu$-calculus, respectively.

The following are variants of the translations in [6].

Definition 2.2. 1. For $T \in \mathsf{Term}_{\mathsf{SCL}}$ and $x \in \mathsf{Var}_{\mathsf{T}}$, we define $\lambda^* x.T \in \mathsf{Term}_{\mathsf{SCL}}$ as follows:

$$\begin{aligned}
\lambda^* x.x &= \mathsf{S}_0\mathsf{K}_0\mathsf{K}_0 && \\
\lambda^* x.T &= \mathsf{K}_0 T && (x \notin FV(T)) \\
\lambda^* x.Tx &= T && (x \notin FV(T)) \\
\lambda^* x.TU &= \mathsf{S}_0(\lambda^* x.T)(\lambda^* x.U) && (x \in FV(TU)) \\
\lambda^* x.T\alpha &= \mathsf{C}_{10}(\lambda^* x.T)\alpha && (x \in FV(T)).
\end{aligned}$$

2. For $T \in \mathsf{Term_{SCL}}$ and $\alpha \in \mathsf{Var_S}$, we define $\mu^*\alpha.T \in \mathsf{Term_{SCL}}$ as follows:

$$\begin{aligned}
\mu^*\alpha.T &= \mathsf{K}_1 T && (\alpha \notin FV(T)) \\
\mu^*\alpha.TU &= \mathsf{S}_1(\mu^*\alpha.T)(\mu^*\alpha.U) && (\alpha \in FV(TU)) \\
\mu^*\alpha.T\alpha &= T && (\alpha \notin FV(T)) \\
\mu^*\alpha.T\alpha &= \mathsf{W}_1(\mu^*\alpha.T) && (\alpha \in FV(T)) \\
\mu^*\alpha.T\beta &= \mathsf{C}_{11}(\mu^*\alpha.T)\beta && (\alpha \neq \beta,\ \alpha \in FV(T)).
\end{aligned}$$

Definition 2.3 (Strong reduction). The *strong reduction* $\twoheadrightarrow_s$ on SCL-terms is defined as the least reflexive transitive compatible relation satisfying the following axioms and rules.

$$\begin{aligned}
\mathsf{K}_0 T_1 T_2 &\twoheadrightarrow_s T_1 && (\mathsf{K}_0) \\
\mathsf{K}_1 T_1 \mathcal{S}_2 &\twoheadrightarrow_s T_1 && (\mathsf{K}_1) \\
\mathsf{S}_0 T_1 T_2 T_3 &\twoheadrightarrow_s T_1 T_3 (T_2 T_3) && (\mathsf{S}_0) \\
\mathsf{S}_1 T_1 T_2 \mathcal{S}_3 &\twoheadrightarrow_s T_1 \mathcal{S}_3 (T_2 \mathsf{S}_3) && (\mathsf{S}_1) \\
\mathsf{W}_1 T_1 \mathcal{S}_2 &\twoheadrightarrow_s T_1 \mathcal{S}_2 \mathcal{S}_2 && (\mathsf{W}_1) \\
\mathsf{C}_{10} T_1 \mathcal{S}_2 T_3 &\twoheadrightarrow_s T_1 T_3 \mathcal{S}_2 && (\mathsf{C}_{10}) \\
\mathsf{C}_{11} T_1 \mathcal{S}_2 \mathcal{S}_3 &\twoheadrightarrow_s T_1 \mathcal{S}_3 \mathcal{S}_2 && (\mathsf{C}_{11}) \\
\mathsf{K}_1 T &\twoheadrightarrow_s \mathsf{K}_0(\mathsf{K}_1 T) && (\phi_1) \\
\mathsf{S}_1 T &\twoheadrightarrow_s \mathsf{S}_0(\mathsf{S}_0(\mathsf{K}_0 \mathsf{S}_1) T) && (\phi_2) \\
\mathsf{S}_1(\mathsf{K}_1 T) &\twoheadrightarrow_s \mathsf{S}_0(\mathsf{K}_0(\mathsf{S}_1(\mathsf{K}_1 T))) && (\phi_3) \\
\mathsf{W}_1 T &\twoheadrightarrow_s \mathsf{S}_0(\mathsf{K}_0 \mathsf{W}_1)(\mathsf{S}_0(\mathsf{S}_0(\mathsf{K}_0 \mathsf{S}_1) T) \mathsf{K}_1) && (\phi_4) \\
\mathsf{C}_{11} T &\twoheadrightarrow_s \mathsf{C}_{10}(\mathsf{S}_0(\mathsf{K}_0 \mathsf{C}_{11}) T) && (\phi_5) \\
\mathsf{C}_{10} T U &\twoheadrightarrow_s \mathsf{C}_{10}(\mathsf{S}_0 T(\mathsf{K}_0 U)) && (\sigma_0) \\
\mathsf{C}_{11} T U &\twoheadrightarrow_s \mathsf{C}_{11}(\mathsf{S}_1 T(\mathsf{K}_1 U)) && (\sigma_1)
\end{aligned}$$

$$\frac{T \twoheadrightarrow_s U}{\lambda^* x.T \twoheadrightarrow_s \lambda^* x.U}\ (\xi_T) \qquad \frac{T \twoheadrightarrow_s U}{\mu^* \alpha.T \twoheadrightarrow_s \mu^* \alpha.U}\ (\xi_S)$$

The equivalence closure of $\twoheadrightarrow_s$ is denoted by $=_s$.

Theorem 2.4. *For any* SCL*-terms* T *and* U, $T =_s U$ *iff* $T =_{\mathsf{SCL}} U$.

Proof. (If part) By induction on the definition of $T =_{\mathsf{SCL}} U$. The ζ-rules are the only nontrivial cases. For (ζ_T), suppose $Tx =_{\mathsf{SCL}} Ux$ for a fresh variable x, and then we have $Tx =_s Ux$ by IH. Hence we have $T = \lambda^* x.Tx =_s \lambda^* x.Ux = U$ by (ξ_T). The case of (ζ_S) is similarly proved.

(Only-if part) It is sufficient to prove that $T \twoheadrightarrow_s U$ implies $T =_{\mathsf{SCL}} U$. The case of the rule (ϕ_1) is proved as follows.

$$\mathsf{K}_1 T x \alpha = \mathsf{K}_1 T (x :: \alpha) =_{\mathsf{SCL}} T$$
$$\mathsf{K}_0 (\mathsf{K}_1 T) x \alpha =_{\mathsf{SCL}} \mathsf{K}_1 T \alpha =_{\mathsf{SCL}} T$$

Hence we have $\mathsf{K}_1 T x \alpha =_{\mathsf{SCL}} \mathsf{K}_0(\mathsf{K}_1 T) x \alpha$, and then $\mathsf{K}_1 T =_{\mathsf{SCL}} \mathsf{K}_0(\mathsf{K}_1 T)$ by the ζ-rules. The other cases are similarly proved. ■

2.3. Weak Reduction

The one-step weak reduction $\rightarrow_w$ of SCL is defined as the least compatible relation satisfying the first seven axioms (from K_0 to C_{11}) of the strong reduction, and the weak reduction $\twoheadrightarrow_w$ is the reflexive transitive closure of $\rightarrow_w$. The confluence of the weak reduction is easily proved by a generalized notion of complete development, which was independently introduced in [2] and [4].

THEOREM 2.5 ([2, 4]). *Let $(A, \rightarrow)$ be an abstract rewriting system, and $\twoheadrightarrow$ be the reflexive transitive closure of $\rightarrow$. Suppose that there exists a translation f on A such that, for any $a, b \in A$, if $a \rightarrow b$ then $b \twoheadrightarrow f(a) \twoheadrightarrow f(b)$. Then, $\twoheadrightarrow$ is confluent.*

THEOREM 2.6 (Confluence of weak reduction). *The weak reduction $\twoheadrightarrow_w$ on SCL is confluent.*

PROOF. Define the translation $(\cdot)^\dagger$ on $\rightarrow_{\mathsf{assoc}}$-normal forms as follows:

$$\begin{aligned}
(\mathsf{K}_0 T_1 T_2)^\dagger &= T_1^\dagger & (\mathsf{K}_1 T_1 \vec{T}_2 \alpha)^\dagger &= T_1^\dagger \\
(\mathsf{S}_0 T_1 T_2 T_3)^\dagger &= T_1^\dagger T_3^\dagger (T_2^\dagger T_3^\dagger) & (\mathsf{S}_1 T_1 T_2 \vec{T}_3 \alpha)^\dagger &= T_1^\dagger \vec{T}_3^\dagger \alpha (T_2^\dagger \vec{T}_3^\dagger \alpha) \\
(\mathsf{C}_{10} T_1 \vec{T}_2 \alpha T_3)^\dagger &= T_1^\dagger T_3^\dagger \vec{T}_2^\dagger \alpha & (\mathsf{C}_{11} T_1 \vec{T}_2 \alpha \vec{T}_3 \beta)^\dagger &= T_1^\dagger \vec{T}_3^\dagger \beta \vec{T}_2^\dagger \alpha \\
(\mathsf{W}_1 T_1 \vec{T}_2 \alpha)^\dagger &= T_1^\dagger \vec{T}_2^\dagger \alpha \vec{T}_2^\dagger \alpha & & \\
(T_1 T_2)^\dagger &= T_1^\dagger T_2^\dagger & & \text{(otherwise)} \\
(T_1 \alpha)^\dagger &= T_1^\dagger \alpha & & \text{(otherwise)},
\end{aligned}$$

where $\vec{T}$ denotes a finite sequence $(T_1, \cdots, T_n)$ of terms, $T'\vec{T}$ denotes the term $T' T_1 \cdots T_n$, and $\vec{T}^\dagger$ denotes $(T_1^\dagger, \cdots, T_n^\dagger)$. This translation is well-defined on $\mathsf{Term}_{\mathsf{SCL}}$, and satisfies $U \twoheadrightarrow_w T^\dagger \twoheadrightarrow_w U^\dagger$ for any $T \rightarrow_w U$. ■

3. Confluence of Strong Reduction

In this section, we show the confluence of the strong reduction of SCL. It is proved by the confluence of the $\Lambda\mu$-calculus, which has been proved in [10].

3.1. Untyped $\Lambda\mu$-Calculus

First, we recall the $\Lambda\mu$-calculus and correspondence between SCL and the $\Lambda\mu$-calculus. We follow the notation of [8].

Definition 3.1 ($\Lambda\mu$-calculus). The terms of the $\Lambda\mu$-calculus are defined by

$$M, N ::= x \mid \lambda x.M \mid MN \mid \mu\alpha.M \mid M\alpha.$$

As abbreviations, $\lambda x_1 x_2 \cdots x_n.M$ denotes $\lambda x_1.(\lambda x_2.(\cdots(\lambda x_n.M)\cdots))$ and similarly for μ. The juxtaposition MN and $M\alpha$ have the same associative strength and both are left associative. Variable occurrences of x and α are bound in $\lambda x.M$ and $\mu\alpha.M$, respectively. Variable occurrences which are not bound are called free, and $FV(M)$ denotes the set of variables freely occurring in M. The set of the $\Lambda\mu$-terms is denoted by $\mathsf{Term}_{\Lambda\mu}$, and the set of the $\Lambda\mu$-terms containing no free stream variable is denoted by $\mathsf{Term}^0_{\Lambda\mu}$. $M[x := N]$ and $M[\alpha := \beta]$ are usual capture-avoiding substitutions, and $M[P\alpha := PN\alpha]$ recursively replaces each subterm of the form $P\alpha$ in M by $PN\alpha$.

The reduction relation $\twoheadrightarrow_{\Lambda\mu}$ is the reflexive transitive closure of the least compatible relation such that

$$\begin{array}{rll}
(\lambda x.M)N \twoheadrightarrow_{\Lambda\mu} M[x := N] & & (\beta_T)\\
(\mu\alpha.M)\beta \twoheadrightarrow_{\Lambda\mu} M[\alpha := \beta] & & (\beta_S)\\
\lambda x.Mx \twoheadrightarrow_{\Lambda\mu} M & (x \notin FV(M)) & (\eta_T)\\
\mu\alpha.M\alpha \twoheadrightarrow_{\Lambda\mu} M & (\alpha \notin FV(M)) & (\eta_S)\\
\mu\alpha.M \twoheadrightarrow_{\Lambda\mu} \lambda x.\mu\alpha.M[P\alpha := Px\alpha] & & (fst)
\end{array}$$

The equivalence closure of $\twoheadrightarrow_{\Lambda\mu}$ is denoted by $=_{\Lambda\mu}$.

Contexts are defined by $K ::= []\alpha \mid K[[]M]$, and $K[M]$ is defined in a usual way. The substitution $M[P\alpha := K[P]]$ recursively replaces each subterm of the form $P\alpha$ in M by $K[P]$.

Saurin showed in [10] the confluence of the $\lambda\mu$-calculus for the terms without free stream variables.

THEOREM 3.2 (Confluence [10]). *1. For any* $M \in \mathsf{Term}^0_{\Lambda\mu}$*, if* $M \twoheadrightarrow_{\Lambda\mu} M_1$ *and* $M \twoheadrightarrow_{\Lambda\mu} M_2$ *hold, then there exists* M_3 *such that* $M_1 \twoheadrightarrow_{\Lambda\mu} M_3$ *and* $M_2 \twoheadrightarrow_{\Lambda\mu} M_3$*.*

2. For any $M_1, M_2 \in \mathsf{Term}^0_{\Lambda\mu}$*, if* $M_1 =_{\Lambda\mu} M_2$ *holds, then there exists* M_3 *such that* $M_1 \twoheadrightarrow_{\Lambda\mu} M_3$ *and* $M_2 \twoheadrightarrow_{\Lambda\mu} M_3$*.*

3.2. Correspondence between $\Lambda\mu$-Calculus and SCL

The relationship between the $\Lambda\mu$-calculus and SCL is formalized by the following translations.

DEFINITION 3.3. 1. The mapping M^* from $\mathsf{Term}_{\Lambda\mu}$ to $\mathsf{Term}_{\mathsf{SCL}}$ is defined by

$$\begin{aligned} x^* &= x \\ (\lambda x.M)^* &= \lambda^* x.M^* & (MN)^* &= M^*N^* \\ (\mu\alpha.M)^* &= \mu^*\alpha.M^* & (M\alpha)^* &= M^*\alpha. \end{aligned}$$

2. The mappings T_* from $\mathsf{Term}_{\mathsf{SCL}}$ to $\mathsf{Term}_{\Lambda\mu}$ and $\mathcal{S}_*$ from $\mathsf{Stream}_{\mathsf{SCL}}$ to contexts are defined by

$$\begin{aligned} (\mathsf{K}_0)_* &= \lambda xy.x & x_* &= x \\ (\mathsf{K}_1)_* &= \lambda x.\mu\alpha.x & (TU)_* &= T_*U_* \\ (\mathsf{S}_0)_* &= \lambda xyz.xz(yz) & (T\alpha)_* &= T_*\alpha \\ (\mathsf{S}_1)_* &= \lambda xy.\mu\alpha.x\alpha(y\alpha) \\ (\mathsf{C}_{10})_* &= \lambda x.\mu\alpha.\lambda y.xy\alpha & \alpha_* &= []\alpha \\ (\mathsf{C}_{11})_* &= \lambda x.\mu\alpha\beta.x\beta\alpha & (T :: \mathcal{S})_* &= \mathcal{S}_*[[]T_*]. \\ (\mathsf{W}_1)_* &= \lambda x.\mu\alpha.x\alpha\alpha \end{aligned}$$

LEMMA 3.4. *1.* $\mu^*\alpha.T \twoheadrightarrow_s \lambda^* x.\mu^*\alpha.T[\alpha := x :: \alpha]$ *holds by the* ϕ*-rules.*

2. $(\lambda^* x.T)[\alpha := \mathcal{S}] \twoheadrightarrow_s \lambda^* x.T[\alpha := \mathcal{S}]$ *holds by the* σ_0*-rule.*

3. $(\mu^*\beta.T)[\alpha := \mathcal{S}] \twoheadrightarrow_s \mu^*\beta.T[\alpha := \mathcal{S}]$ *holds by the* σ_1*-rule.*

PROOF. 1. By induction on T.

Case $\alpha \notin FV(T)$.

$$\begin{aligned} \mu^*\alpha.T &= \mathsf{K}_1 T \\ &\twoheadrightarrow_s \mathsf{K}_0(\mathsf{K}_1 T) && \text{(by } \phi_1\text{)} \\ &= \lambda^* x.\mu^*\alpha.T. \end{aligned}$$

Case $T = T'U'$ and $\alpha \in FV(T'U')$.

$$\begin{aligned}\text{LHS} &= \mathsf{S}_1(\mu^*\alpha.T')(\mu^*\alpha.U')\\ \text{RHS} &= \lambda^*x.\mathsf{S}_1(\mu^*\alpha.T'[\alpha := x :: \alpha])(\mu^*\alpha.U'[\alpha := x :: \alpha])\\ &= \mathsf{S}_0(\lambda^*x.\mathsf{S}_1(\mu^*\alpha.T'[\alpha := x :: \alpha]))(\lambda^*x.\mu^*\alpha.U'[\alpha := x :: \alpha])\end{aligned}$$

Here, note that α occurs in $(T'U')[\alpha := x :: \alpha]$, and $\mu^*\alpha.U'[\alpha := x :: \alpha]$ is not identical to x since $U'[\alpha := x :: \alpha]$ is not identical to $x\alpha$.

Subcase $\alpha \in FV(T')$. The variables x and α occur in $T'[\alpha := x :: \alpha]$, so we have

$$\begin{aligned}\text{LHS} &= \mathsf{S}_1(\mu^*\alpha.T')(\mu^*\alpha.U')\\ &\twoheadrightarrow_s \mathsf{S}_0(\mathsf{S}_0(\mathsf{K}_0\mathsf{S}_1)(\mu^*\alpha.T'))(\mu^*\alpha.U') && \text{(by } (\phi_2)\text{)}\\ &\twoheadrightarrow_s \mathsf{S}_0(\mathsf{S}_0(\mathsf{K}_0\mathsf{S}_1)(\lambda^*x.\mu^*\alpha.T'[\alpha := x :: \alpha]))\\ &\qquad\qquad (\lambda^*x.\mu^*\alpha.U'[\alpha := x :: \alpha]) && \text{(by IH)}\\ &= \mathsf{S}_0(\lambda^*x.\mathsf{S}_1(\mu^*\alpha.T'[\alpha := x :: \alpha]))(\lambda^*x.\mu^*\alpha.U'[\alpha := x :: \alpha])\\ &= \text{RHS},\end{aligned}$$

where $\mu^*\alpha.T'[\alpha := x :: \alpha]$ is not identical to x.

Subcase $\alpha \notin FV(T')$. We have $T'[\alpha := x :: \alpha] = T'$ and $x \notin FV(\mu^*\alpha.T'[\alpha := x :: \alpha])$, so we have

$$\begin{aligned}\text{LHS} &= \mathsf{S}_1(\mathsf{K}_1T')(\mu^*\alpha.U')\\ &\twoheadrightarrow_s \mathsf{S}_1(\mathsf{K}_1T')(\lambda^*x.\mu^*\alpha.U'[\alpha := x :: \alpha]) && \text{(by IH)}\\ &\twoheadrightarrow_s \mathsf{S}_0(\mathsf{K}_0(\mathsf{S}_1(\mathsf{K}_1T')))(\lambda^*x.\mu^*\alpha.U'[\alpha := x :: \alpha]) && \text{(by } (\phi_3)\text{)}\\ &= \text{RHS}.\end{aligned}$$

Case $T = T'\alpha$ and $\alpha \notin FV(T')$. LHS is T', whereas RHS is $\lambda^*x.\mu^*\alpha.T'(x :: \alpha) = \lambda^*x.\mu^*\alpha.T'x\alpha$, which is identical to T' by definition.

Case $T = T'\alpha$ and $\alpha \in FV(T')$. Note that, since $\alpha \in FV(T')$, the variables x and α occur in $T'[\alpha := x :: \alpha]$.

$$\begin{aligned}\text{LHS} &= \mathsf{W}_1(\mu^*\alpha.T')\\ &\twoheadrightarrow_s \mathsf{S}_0(\mathsf{K}_0\mathsf{W}_1)(\mathsf{S}_0(\mathsf{S}_0(\mathsf{K}_0\mathsf{S}_1)(\mu^*\alpha.T'))\mathsf{K}_1) && \text{(by } \phi_4\text{)}\\ &\twoheadrightarrow_s \mathsf{S}_0(\mathsf{K}_0\mathsf{W}_1)(\mathsf{S}_0(\mathsf{S}_0(\mathsf{K}_0\mathsf{S}_1)(\lambda^*x.\mu^*\alpha.T'[\alpha := x :: \alpha]))\mathsf{K}_1) && \text{(by IH)}\\ &= \lambda^*x.\mu^*\alpha.(T'[\alpha := x :: \alpha]x\alpha) = RHS.\end{aligned}$$

Case $T = T'\beta$, $\alpha \neq \beta$, and $\alpha \in FV(T')$.

$$\begin{aligned}
\mathrm{LHS} &= \mathsf{C}_{11}(\mu^*\alpha.T')\beta \\
&\twoheadrightarrow_s \mathsf{C}_{10}(\mathsf{S}_0(\mathsf{K}_0\mathsf{C}_{11})(\mu^*\alpha.T'))\beta && \text{(by } (\phi_5)\text{)} \\
&\twoheadrightarrow_s \mathsf{C}_{10}(\mathsf{S}_0(\mathsf{K}_0\mathsf{C}_{11})(\lambda^*x.\mu^*\alpha.T'[\alpha := x :: \alpha]))\beta && \text{(by IH)} \\
&= \lambda^*x.\mu^*\alpha.(T'[\alpha := x :: \alpha]\beta) = RHS.
\end{aligned}$$

2. By induction on T.
Case $T = T'\alpha$ and $x \in FV(T')$.

$$\begin{aligned}
\mathrm{LHS} &= (\mathsf{C}_{10}(\lambda^*x.T')\alpha)[\alpha := \mathcal{S}] \\
&= \mathsf{C}_{10}(\lambda^*x.T')[\alpha := \mathcal{S}]\mathcal{S} \\
&\twoheadrightarrow_s \mathsf{C}_{10}(\lambda^*x.T'[\alpha := \mathcal{S}])\mathcal{S} && \text{(by IH).}
\end{aligned}$$

Hence it is sufficient to prove that if $x \in FV(T'')$ then $\mathsf{C}_{10}(\lambda^*x.T'')\mathcal{S} \twoheadrightarrow_s \lambda^*x.(T''\mathcal{S})$ by (σ_0). It is proved by induction on $\mathcal{S}$. When $\mathcal{S} = \beta$, we have $\mathsf{C}_{10}(\lambda^*x.T'')\beta = \lambda^*x.(T''\beta)$ by definition. When $\mathcal{S} = U :: \mathcal{S}'$, we have

$$\begin{aligned}
\mathsf{C}_{10}(\lambda^*x.T'')(U :: \mathcal{S}') &= \mathsf{C}_{10}(\lambda^*x.T'')U\mathcal{S}' \\
&\twoheadrightarrow_s \mathsf{C}_{10}(\mathsf{S}_0(\lambda^*x.T'')(\mathsf{K}_0U))\mathcal{S}' && \text{(by } (\sigma_0)\text{)} \\
&= \mathsf{C}_{10}(\lambda^*x.T''U)\mathcal{S}' \\
&\twoheadrightarrow_s \lambda^*x.(T''U\mathcal{S}') && \text{(by IH)} \\
&= \lambda^*x.(T''(U :: \mathcal{S}')).
\end{aligned}$$

The other cases are proved by IH.
3. Similar to 2. ∎

Lemma 3.5. *1. For any $T \in \mathsf{Terms}_{\mathsf{SCL}}$, we have $(T_*)^* = T$.*
2. $T \twoheadrightarrow_s U$ implies $T_* =_{\Lambda\mu} U_*$.
3. $M \twoheadrightarrow_{\Lambda\mu} N$ implies $M^* \twoheadrightarrow_s N^*$.

Proof. 1. By induction on T. For example, in the case of $T = \mathsf{C}_{10}$, we have $\lambda^*y.(xy\alpha) = \mathsf{C}_{10}(\lambda^*y.xy)\alpha = \mathsf{C}_{10}x\alpha$, so we have $((\mathsf{C}_{10})_*)^* = (\lambda x.\mu\alpha.\lambda y.xy\alpha)^* = \lambda^*x.\mu^*\alpha.\lambda^*y.(xy\alpha) = \lambda^*x.\mu^*\alpha.(\mathsf{C}_{10}x\alpha) = \mathsf{C}_{10}$.

2. By Theorem 2.4, it is sufficient to show that $T =_{\mathsf{SCL}} U$ implies $T_* =_{\Lambda\mu} U_*$. It is straightforwardly proved in the same way as the proof in [6].

3. By induction on $M \twoheadrightarrow_{\Lambda\mu} N$, using Lemma 3.4. ∎

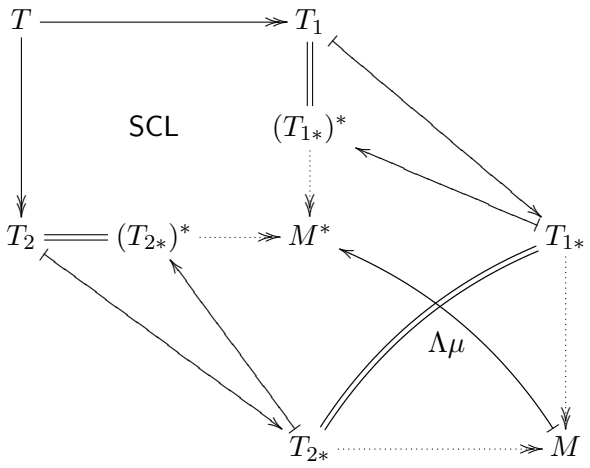

Figure 1. Proof of Theorem 3.6.1.

3.3. Confluence of Strong Reduction

Now the confluence of the strong reduction of SCL for terms without stream variables is proved from the confluence of the $\Lambda\mu$-calculus.

THEOREM 3.6 (Confluence of Strong Reduction). *1. For any $T \in \mathsf{Term}^0_{\mathsf{SCL}}$, if $T \twoheadrightarrow_s T_1$ and $T \twoheadrightarrow_s T_2$ hold, then there exists T_3 such that $T_1 \twoheadrightarrow_s T_3$ and $T_2 \twoheadrightarrow_s T_3$.*

2. For any $T_1, T_2 \in \mathsf{Term}^0_{\mathsf{SCL}}$, if $T_1 =_s T_2$ holds, then there exists T_3 such that $T_1 \twoheadrightarrow_s T_3$ and $T_2 \twoheadrightarrow_s T_3$.

PROOF. It is easy to see that $T \in \mathsf{Term}^0_{\mathsf{SCL}}$ implies $T_* \in \mathsf{Term}^0_{\Lambda\mu}$, and that $T \in \mathsf{Term}^0_{\mathsf{SCL}}$ and $T \twoheadrightarrow_s U$ imply $U \in \mathsf{Term}^0_{\mathsf{SCL}}$.

1. Suppose $T \twoheadrightarrow_s T_1$ and $T \twoheadrightarrow_s T_2$. By Lemma 3.5.2, we have $T_{1*} =_{\Lambda\mu} T_* =_{\Lambda\mu} T_{2*}$. Since T_{1*} and T_{2*} are in $\mathsf{Term}^0_{\Lambda\mu}$, by Theorem 3.2.2, there exists $M \in \mathsf{Term}_{\Lambda\mu}$ such that $T_{1*} \twoheadrightarrow_{\Lambda\mu} M$ and $T_{2*} \twoheadrightarrow_{\Lambda\mu} M$, so by Lemma 3.5.3, we have $(T_{1*})^* \twoheadrightarrow_s M^*$ and $(T_{2*})^* \twoheadrightarrow_s M^*$. By Lemma 3.5.1, we have $T_1 \twoheadrightarrow_s M^*$ and $T_2 \twoheadrightarrow_s M^*$. Figure 1 summarizes this proof.

2. Similar to 1, since $T_1 =_s T_2$ implies $T_{1*} =_{\Lambda\mu} T_{2*}$ by Lemma 3.5.2. ■

Note that the statement 2 does not directly follow from 1, since the set $\mathsf{Term}^0_{\mathsf{SCL}}$ is not closed under the inverse of the strong reduction.

4. Conservativity of SCL

In this section, we prove that the strong reduction of SCL is conservative over that of CL, and then the extensional equality of SCL is conservative over that of CL, that is, if two CL-terms are equal in SCL, then they are equal in CL. The conservativity of the extensional equality is proved by the confluence and the conservativity of the strong reduction.

The combinatory logic CL and its strong reduction is obtained by restricting SCL to the constants K_0 and S_0, the operation $(\cdot)$, and the reduction rules (K_0), (S_0), and (ξ_T). In this section, the strong reductions of SCL and CL are denoted by $\twoheadrightarrow_s^{\mathsf{SCL}}$ and $\twoheadrightarrow_s^{\mathsf{CL}}$, respectively. The extensional equality $=_{\mathsf{CL}}$ is the equivalence closure of $\twoheadrightarrow_s^{\mathsf{CL}}$.

Theorem 4.1 (Conservativity of Strong Reduction). *For any $T \in \mathsf{Term}_{\mathsf{CL}}$, if $T \twoheadrightarrow_s^{\mathsf{SCL}} U$ holds, then U also belongs to $\mathsf{Term}_{\mathsf{CL}}$ and $T \twoheadrightarrow_s^{\mathsf{CL}} U$ holds.*

Note that this is not trivial due to the ξ-rules. For any CL term T, we have $T = \mu^*\alpha.T\alpha$ for fresh α, so we have to show $\mu^*\alpha.U \in \mathsf{Term}_{\mathsf{CL}}$ for any U such that $T\alpha \twoheadrightarrow_s^{\mathsf{SCL}} U$.

In order to prove the theorem, we define auxiliary notions and show some properties.

Definition 4.2. 1. The constants K_1, S_1, W_1, C_{10}, and C_{11} are called *stream constants*.

2. The *variable contexts* are defined by $K ::= [] \mid K \cdot x \mid K\alpha$.

Lemma 4.3. *1. If $\lambda^*x.T$ is in $\mathsf{Term}_{\mathsf{CL}}$, then T also belongs to $\mathsf{Term}_{\mathsf{CL}}$.*

2. If $\lambda^*x.T = K[U\alpha]$ for a term $U \in \mathsf{Term}_{\mathsf{CL}}$ and a variable context K, then $T = K[U\alpha]x$ such that $x \notin FV(K[U\alpha])$.

3. If $\mu^*\alpha.T$ contains no stream constant, then $T = T'\alpha$ such that $\alpha \notin FV(T')$.

Proof. 1. By induction on T. Note that if T contains a stream variable, $\lambda^*x.T$ introduces the stream constant C_{10}.

2. The rules other than $\lambda^*x.T \cdot x = T$ do not match the form of $K[U\alpha]$ for $U \in \mathsf{Term}_{\mathsf{CL}}$.

3. If T is not of the form $T'\alpha$ ($\alpha \notin FV(T')$), then $\mu^*\alpha.T$ introduces a stream constant. ∎

Proof of Theorem 4.1. We simultaneously prove the following two statements for any CL-term T and any variable context K.

1. $T \twoheadrightarrow_s^{\mathsf{SCL}} U \Rightarrow (U \in \mathsf{CL}) \& (T \twoheadrightarrow_s^{\mathsf{CL}} U)$

2. $K[T\alpha] \twoheadrightarrow_s^{\mathsf{SCL}} U \Rightarrow (U = K[T'\alpha])\&(T \twoheadrightarrow_s^{\mathsf{CL}} T')$ for some $T' \in \mathsf{CL}$.

The cases we have to consider are only (K_0), (S_0), and the ξ-rules since neither T nor $K[T\alpha]$ contain any stream constant, and the cases of (K_0) and (S_0) are easily proved.

Case (ξ_T). For 1, suppose $T = \lambda^* x.T' \in \mathsf{Term}_{\mathsf{CL}}$, $U = \lambda^* x.U'$, and $T' \twoheadrightarrow_s^{\mathsf{SCL}} U'$. By Lemma 4.3.1, T' is also a CL-term, so $T' \twoheadrightarrow_s^{\mathsf{CL}} U'$ holds by IH of 1. Hence we have $\lambda^* x.T' \twoheadrightarrow_s^{\mathsf{CL}} \lambda^* x.U'$.

For 2, suppose $K[T\alpha] = \lambda^* x.T'$, $U = \lambda^* x.U'$, and $T' \twoheadrightarrow_s^{\mathsf{SCL}} U'$. By Lemma 4.3.2, $T' = K[T\alpha]x$ and $x \notin FV(K[T\alpha])$. By IH of 2 for $T' \twoheadrightarrow_s^{\mathsf{SCL}} U'$, U' must be $K[T''\alpha]x$ for some $T'' \in \mathsf{Term}_{\mathsf{CL}}$ such that $T \twoheadrightarrow_s^{\mathsf{CL}} T''$. Hence we have $U = K[T''\alpha]$ and $T \twoheadrightarrow_{s\,\mathsf{CL}} T''$.

Case (ξ_S). For 1, suppose $T = \mu^* \alpha.T' \in \mathsf{Term}_{\mathsf{CL}}$, $U = \mu^* \alpha.U'$, and $T' \twoheadrightarrow_s^{\mathsf{SCL}} U'$. By Lemma 4.3.3, we have $T' = T\alpha$ and $\alpha \notin FV(T)$. By IH of 2 for $T' \twoheadrightarrow_s^{\mathsf{SCL}} U'$, U' must be in the form of $U''\alpha$ and $T \twoheadrightarrow_s^{\mathsf{CL}} U''$, and then $U = U''$.

2 is similarly proved to the case (ξ_T).

Corollary 4.4 (Conservativity of Extensional Equality). *For any* CL*-terms* T *and* U*, if* $T =_{\mathsf{SCL}} U$ *holds, then* $T =_{\mathsf{CL}} U$.

Proof. Neither T nor U contain any stream variable since they are CL-terms. Suppose $T =_{\mathsf{SCL}} U$, and then $T =_s U$ holds, so there exists a SCL-term V such that $T \twoheadrightarrow_s^{\mathsf{SCL}} V$ and $U \twoheadrightarrow_s^{\mathsf{SCL}} V$ by Theorem 3.6.2. By Theorem 4.1, V is a CL-term and we have $T \twoheadrightarrow_s^{\mathsf{CL}} V$ and $U \twoheadrightarrow_s^{\mathsf{CL}} V$, and therefore $T =_{\mathsf{CL}} U$. ■

Corollary 4.5 (Consistency of SCL). *There exist two terms not related by* $=_{\mathsf{SCL}}$.

Proof. Since $\mathsf{S}_0 \neq_{\mathsf{CL}} \mathsf{K}_0$, we have also $\mathsf{S}_0 \neq_{\mathsf{SCL}} \mathsf{K}_0$ by the conservativity. ■

5. Concluding Remarks

We have given the strong reduction for the combinatory calculus SCL in [6] and proved its confluence. By the confluence, the extensional equality of SCL is conservative over CL and hence consistent.

We did not consider a constant representing the identity function, which is often denoted I in CL. However, we can straightforwardly add a new constant I_0 such that

$$\mathsf{I}_0 T =_{\mathsf{SCL}} T \qquad\qquad \mathsf{I}_0 T \twoheadrightarrow_s T,$$

and redefine $\lambda^* x.x$ as I_0 with little change of proofs in this paper. In particular, we do not have to change ϕ- nor σ-rules of the strong reduction. As discussed in [3] for CL, the variant containing I must have a merit to consider a correspondence between irreducible SCL-terms with respect to the strong reduction and normal $\Lambda\mu$-terms.

We proved the confluence SCL by means of the confluence of the $\Lambda\mu$-calculus, and it is similar to the confluence proof of the strong reduction of CL. In the case of CL, the direct proof of confluence which does not depend on that of the λ-calculus was the first TLCA open problem, and recently it was solved by David [1] and Minari [5] independently. To prove the confluence of SCL directly independent of the $\Lambda\mu$-calculus is still open.

References

[1] David, R. A direct proof of the confluence of combinatory strong reduction. *Theoretical Computer Science*, 410(42):4204–4215, 2009.

[2] Dehornoy, P. and van Oostrom, V. Z, proving confluence by monotonic single-step upperbound functions. In *Logical Models of Reasoning and Computation (LMRC-08)*, 2008. Slides available on `http://www.phil.uu.nl/ oostrom/publication/talk/lmrc060508.pdf`.

[3] Hindley, R. and Seldin, J.P. *Lambda-Calculus and Combinators, an Introduction.* Cambridge University Press, 2008.

[4] Komori, Y., Matsuda, N., and Yamakawa, F. A simplified proof of the church-rosser theorem. *Studia Logica*, 102(1):175–183, 2013.

[5] Minari, P. A solution to curry and hindley's problem on combinatory strong reduction. *Archive for Mathematical Logic*, 48(2):159–184, 2009.

[6] Nakazawa, K. and Katsumata, S. Extensional models of untyped Lambda-mu calculus. In *Fourth International Workshop on Classical Logic and Computation (CL&C'12)*, volume 97 of *Electric Proceedings in Theoretical Computer Science*, pages 35–47, 2012.

[7] Parigot, M. $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In *Proceedings of the International Conference on Logic Programming and Automated Reasoning (LPAR '92)*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer, 1992.

[8] Saurin, A. Separation with streams in the $\Lambda\mu$-calculus. In *20th Annual IEEE Symposium on Logic in Computer Science (LICS' 05)*, pages 356–365. IEEE Computer Society, 2005.

[9] Saurin, A. Standardization and Böhm trees for $\Lambda\mu$-calculus. In *Tenth International Symposium on Functional and Logic Programming (FLOPS 2010)*, volume 6009 of *Lecture Notes in Computer Science*, pages 134–149. Springer, 2010.

[10] Saurin, A. Typing streams in the $\Lambda\mu$-calculus. *ACM Transactions on Computational Logic*, 11:1–34, 2010.

Koji Nakazawa
Graduate School of Informatics
Kyoto University
Kyoto, Japan
knak@kuis.kyoto-u.ac.jp

Hiroto Naya
FORCIA, Inc.
Tokyo, Japan