ラムダ計算の型問題について

– On type-related problems of λ-calculi –

藤田 憲悦　(群馬大学)*

1. はじめに

ラムダ計算 [5] は，Turing 機械と同様に万能な計算モデルである [24]．ラムダ計算は ”関数 ”の一般的な理論であり，その背景にある思想は創造当時の Church 自身による次の文章に表れている．

『Underlying the formal calculi which we shall develop is *the concept of a function*, ····. A function is *a rule of correspondence* by which when anything is given (as argument) another thing (the value of the function for that argument) may be obtained. That is, a function is an operation ····. In particular it is not excluded that *one of the elements of the range of arguments of a function f should be the function f itself*. This possibility has frequently been denied, ····. Here, however, we regard the operation or rule of correspondence, which constitutes the function, as being first given, ····.』

(Church [7] より引用：*Italic face* は著者による)

この理論は，現在ではタイプフリー・ラムダ計算と呼ばれている．そして，計算理論のみならず関数型プログラミング言語などの基礎の一つになっている．

一方，型付きラムダ計算の体系もある．ここで，型は ”関数 ”の定義域や値域に相当する概念を形式化したものである．型付きラムダ計算も論理・数学の基礎付けにとどまらず，プログラミング言語や定理証明器の基礎理論として研究されてきた．ラムダ計算の研究は，構文論と数学的モデルを扱う意味論とに大別される．構文論の研究の中でも，”関数の計算 ”的観点から，正規化定理，合流性 (Church-Rosser の定理)，Subject reduction 性など動的性質に関する研究と，型検査，型推論，Inhabitation など静的性質に関する研究に分類されるであろう．特に，型付きラムダ計算の静的性質は，型付きプログラミング言語や定理証明システムにおける ”計算 ”と独立な性質の解析のために不可欠である．さらに，論理式 A はラムダ計算の型と解釈（同一視）でき，型 A を持つラムダ式 M はその論理式の証明と解釈され得る [13]．この解釈は，カリー・ハワード同型 (または，Propositions-as-types，proofs-as-terms など) と呼ばれ，論理とラムダ計算との橋渡しをしている．

型付きラムダ計算の体系には，Curry 流 [8] と Church 流 [6] の二種類がある．Church 流のラムダ式は，その中に型の情報がコメントされた式になっている．これによって，Church 流のラムダ式の型検査問題が通常は容易になる．さらに，Church 流のラムダ式が型を持つならばその型は唯一つ存在することになり，ラムダ式は証明記述言語に適している．他方，Curry 流のラムダ式は型情報を全く含まず，タイプフリーのラムダ式と同じである．これにより，型から解放されてラムダ式（プログラム）を書くことができる．そして，式にきちんと型が付くかどうかの判断は，別途定義される型割り当てアルゴリズムに任せることになる．

ラムダ計算の型体系が 1 階の体系 (simply typed lambda-calculus) の場合，Curry 流と Church 流のスタイルの違いは型検査・型推論問題の可解性に影響を及ぼさない．

Keywords: 2nd-order λ-calculus, type-checking, type-inference, inhabitation.

* 〒 376-8515 群馬県桐生市天神町 1-5-1　群馬大学 大学院工学研究科
e-mail: fujita@cs.gunma-u.ac.jp
web: http://www.cs.gunma-u.ac.jp/~fujita/

しかし，型体系を 2 階の体系（2 階直観主義論理）に拡張した場合には大きな違いが生じてくる．例えば，Church 流の型検査問題は決定可能であるが，Curry 流のその問題は決定不能 (Wells (1999) [25]) となってしまう．さらに，Curry 流と Church 流の中間的な構造を持つ式やそれらを組み合わせた型問題 [4] なども考えることができる．このような中間構造を持つラムダ式の型問題を統一的な枠組みの中で扱った研究として Fujita-Schubert (2000) [10] がある．その研究動機は，2 階ラムダ計算の型問題の観点から決定可能・決定不能の境界構造を解明することにあった．そこでは，中間構造の一つであるドメインフリー (domain-free) ラムダ式の型検査・型推論問題 [1] が決定不能であることが示された．最近になり，もう一つの中間構造であるタイプフリー (type-free) ラムダ式の型問題 [2] も決定不能であることが明らかになった [12].

本稿では，これまで得られた結果を踏まえて，2 階ラムダ計算の型問題，特に，型検査，型推論，証明可能性 (Inhabitation) 問題に関連する結果を紹介する．なお，ラムダ計算全般にわたる詳細については，例えば，Barendregt [1, 2] をご参照願いたい.

2. Second-order lambda-calculus $\lambda 2$ in five styles

We introduce second-order lambda-calculus $\lambda 2$ in the Church, domain-free, hole-application, type-free, and Curry styles, which are denoted by Ch, Df, [], Tf and Cu, respectively [10]. Types, λ-terms in s-style for $s \in \{\mathrm{Ch}, \mathrm{Df}, [], \mathrm{Tf}, \mathrm{Ch}\}$, and inference rules for each style are defined as follows:

- $\lambda 2$-types:
$$A ::= X \mid (A \to A) \mid \forall X.A$$
- Contexts:
$$\Gamma ::= \langle\ \rangle \mid x{:}A, \Gamma$$
- $\lambda 2$-terms in Church-style:
$$M ::= x \mid (\lambda x{:}A.M) \mid (MM) \mid (\Lambda X.M) \mid (M[A])$$

  Inference rules for Church-style:
$$\frac{}{\Gamma, x{:}A \vdash_{\mathrm{Ch}} x : A}\ (\mathrm{var})$$
$$\frac{\Gamma, x{:}A_1 \vdash_{\mathrm{Ch}} M : A_2}{\Gamma \vdash_{\mathrm{Ch}} \lambda x{:}A_1.M : A_1 \to A_2}\ (\to I) \qquad \frac{\Gamma \vdash_{\mathrm{Ch}} M_1 : A_1 \to A_2 \quad \Gamma \vdash_{\mathrm{Ch}} M_2 : A_1}{\Gamma \vdash_{\mathrm{Ch}} M_1 M_2 : A_2}\ (\to E)$$
$$\frac{\Gamma \vdash_{\mathrm{Ch}} M : A}{\Gamma \vdash_{\mathrm{Ch}} \Lambda X.M : \forall X.A}\ (\forall I)^{\star} \qquad \frac{\Gamma \vdash_{\mathrm{Ch}} M : \forall X.A}{\Gamma \vdash_{\mathrm{Ch}} M[A_1] : A[X := A_1]}\ (\forall E)$$
  where $(\forall I)^{\star}$ denotes that the eigenvariable condition $X \notin \mathrm{FV}(\Gamma)$ is imposed on the application.
- Domain-free style $\lambda 2$-terms:
$$M ::= x \mid (\lambda x.M) \mid (MM) \mid (\Lambda X.M) \mid (M[A])$$
- Hole-application style $\lambda 2$-terms:
$$M ::= x \mid (\lambda x{:}A.M) \mid (MM) \mid (\Lambda X.M) \mid (M[])$$

[1] この問題は Barth-Sørensen の論文 (1997) [3] で未解決問題として提示されていた.
[2] この問題は Pfenning の論文 (1993) [19] で未解決のまま残されていた.

- Type-free style $\lambda 2$-terms:

$$M ::= x \mid (\lambda x.M) \mid (MM) \mid (\Lambda.M) \mid (M[])$$

Inference rules for domain-free, hole-application, and type-free styles are defined similarly.

- $\lambda 2$-terms in Curry-style:

$$M ::= x \mid (\lambda x.M) \mid (MM)$$

Inference rules for Curry-style:

$$\frac{}{\Gamma, x{:}A \vdash_{\mathrm{Cu}} x : A}\ (\mathrm{var})$$

$$\frac{\Gamma, x{:}A_1 \vdash_{\mathrm{Cu}} M : A_2}{\Gamma \vdash_{\mathrm{Cu}} \lambda x.M : A_1 \to A_2}\ (\to I) \qquad \frac{\Gamma \vdash_{\mathrm{Cu}} M_1 : A_1 \to A_2 \quad \Gamma \vdash_{\mathrm{Cu}} M_2 : A_1}{\Gamma \vdash_{\mathrm{Cu}} M_1 M_2 : A_2}\ (\to E)$$

$$\frac{\Gamma \vdash_{\mathrm{Cu}} M : A}{\Gamma \vdash_{\mathrm{Cu}} M : \forall X.A}\ (\forall I)^\star \qquad \frac{\Gamma \vdash_{\mathrm{Cu}} M : \forall X.A}{\Gamma \vdash_{\mathrm{Cu}} M : A[X := A_1]}\ (\forall E)$$

where $(\forall I)^\star$ denotes the eigenvariable condition $X \notin \mathrm{FV}(\Gamma)$.

Definition 1 *We define a binary relation on the styles, such that* $\mathrm{Cu} < \mathrm{Tf} < \mathrm{Df} < \mathrm{Ch}$ *and* $\mathrm{Tf} < [] < \mathrm{Ch}$. *We also write* $<$ *for the transitive closure. For styles* $s, t \in \{\mathrm{Cu}, \mathrm{Tf}, [], \mathrm{Df}, \mathrm{Ch}\}$ *with* $s < t$, *an erasure* $|\cdot|_s^t$ *is defined naturally from* t*-style* $\lambda 2$*-terms to* s*-style* $\lambda 2$*-terms, such that* $|x|_{\mathrm{Df}}^{\mathrm{Ch}} = x$, $|\lambda x{:}A.M|_{\mathrm{Df}}^{\mathrm{Ch}} = \lambda x.|M|_{\mathrm{Df}}^{\mathrm{Ch}}$, $|\Lambda X.M|_{\mathrm{Tf}}^{\mathrm{Df}} = \Lambda.|M|_{\mathrm{Tf}}^{\mathrm{Df}}$, $|M[A]|_{\mathrm{Tf}}^{\mathrm{Df}} = |M|_{\mathrm{Tf}}^{\mathrm{Df}}[]$, $|\Lambda.M|_{\mathrm{Cu}}^{\mathrm{Tf}} = |M|_{\mathrm{Cu}}^{\mathrm{Tf}}$, $|M[]|_{\mathrm{Cu}}^{\mathrm{Tf}} = |M|_{\mathrm{Cu}}^{\mathrm{Tf}}$, *and so on.*

Proposition 1 (Uniqueness of types for Church-style) *If* $\Gamma \vdash_{\mathrm{Ch}} M : A_1$ *and* $\Gamma \vdash_{\mathrm{Ch}} M : A_2$ *then we have* $A_1 \equiv A_2$.

Proposition 2 (Erasure and lifting) *Let* $s, t \in \{\mathrm{Cu}, \mathrm{Tf}, [], \mathrm{Df}, \mathrm{Ch}\}$ *with* $s < t$.

1. *If* $\Gamma \vdash_t M : A$ *then* $\Gamma \vdash_s |M|_s^t : A$.
2. *If* $\Gamma \vdash_s M : A$ *then there exists* t*-style* $\lambda 2$*-term* N *such that* $|N|_s^t = M$ *and* $\Gamma \vdash_t N : A$.

Proposition 3 (Generation lemma) *Let* $s \in \{\mathrm{Tf}, \mathrm{Df}, [], \mathrm{Ch}\}$.

1. *If* $\Gamma \vdash_s x : A$ *then* $\Gamma(x) = A$.
2. *If* $\Gamma \vdash_s \lambda x{:}A_0.M : A_1$ *then* $\Gamma, x{:}A_0 \vdash_s M : A_2$ *and* $A_1 = (A_0 \to A_2)$ *for some* A_2, *provided that* $s \geq []$.
3. *If* $\Gamma \vdash_s \lambda x.M : A_1$ *then* $\Gamma, x{:}A_0 \vdash_s M : A_2$ *and* $A_1 = (A_0 \to A_2)$ *for some* A_0, A_2, *provided that* $s \leq \mathrm{Df}$.
4. *If* $\Gamma \vdash_s M_1 M_2 : A_1$ *then* $\Gamma \vdash_s M_1 : A_0 \to A_1$ *and* $\Gamma \vdash_s M_2 : A_0$ *for some* A_0.
5. *If* $\Gamma \vdash_s \Lambda X.M : A_1$ *then* $\Gamma \vdash_s M : A_2$ *and* $A_1 = \forall X.A_2$ *together with* $X \notin \mathrm{FV}(\Gamma)$ *for some* A_2, *provided that* $s > \mathrm{Tf}$.
6. *If* $\Gamma \vdash_{\mathrm{Tf}} \Lambda.M : A_1$ *then* $\Gamma \vdash_{\mathrm{Tf}} M : A_2$ *and* $A_1 = \forall X.A_2$ *together with* $X \notin \mathrm{FV}(\Gamma)$ *for some* A_2.

7. *If* $\Gamma \vdash_s M[A] : A_1$ *then* $\Gamma \vdash_s M : \forall X.A_2$ *and* $A_1 = A_2[X := A]$ *for some* A_2, *provided that* $s \geq \mathrm{Df}$.

8. *If* $\Gamma \vdash_s M[] : A_1$ *then* $\Gamma \vdash_s M : \forall X.A_2$ *and* $A_1 = A_2[X := A]$ *for some* A, A_2, *provided that* $s \leq []$.

Definition 2 (Type-related problems parameterized with styles or systems)

1. *Type checking problem of s-style terms denoted by* $\Gamma \vdash_s M : A$? *or* $\mathrm{TCP}(s)$*:*

   Given an s-style λ*-term* M, *a type* A, *and a context* Γ, *is the judgement* $\Gamma \vdash_s M : A$ *derivable?*

2. *Type inference problem of s-style* λ*-terms denoted by* $\Gamma \vdash_s M :?$ *or* $\mathrm{TIP(s)}$*:*

   Given an s-style λ*-term* M *and a context* Γ, *is there a type* A *such that* $\Gamma \vdash_s M : A$ *is derivable?*

3. *Strong type inference problem of s-style terms denoted by* $?, \Gamma \vdash_s M :?$ *or* $\mathrm{STIP(s)}$*:*

   Given an s-style λ*-term* M *and a context* Γ_0, *are there a context* Γ *and a type* A *such that* $\Gamma_0, \Gamma \vdash_s M : A$ *is derivable?*

4. *Typability problem of s-style terms denoted by* $? \vdash_s M :?$ *or* $\mathrm{TP(s)}$*:*

   Given an s-style λ*-term* M, *are there a context* Γ *and a type* A *such that* $\Gamma \vdash_s M : A$ *is derivable?*

5. *Inhabitation problem of s-style terms denoted by* $\vdash ? : A$ *or* $\mathrm{IHP(s)}$*:*

   Given a type A, *is there a closed* λ*-term* M *in s-style such that* $\vdash_s M : A$*?*

Proposition 4 (Reductions between type-related problems) 1. $\mathrm{TCP}(s) \hookrightarrow \mathrm{TIP}(s)$ *for* $s \in \{\mathrm{Cu}, \mathrm{Tf}, \mathrm{Df}, [], \mathrm{Ch}\}$

2. $\mathrm{TIP}(s) \hookrightarrow \mathrm{TCP}(s)$ *for* $s \in \{\mathrm{Cu}, \mathrm{Tf}, \mathrm{Df}\}$

3. $\mathrm{TIP}(s) \hookrightarrow \mathrm{TCP}(s)$ *for* $s \in \{\mathrm{Tf}, []\}$

4. $\mathrm{TIP}(s) \hookrightarrow \mathrm{TP}(s)$ *for* $s \in \{[], \mathrm{Ch}\}$

5. $\mathrm{TIP}(\mathrm{Df}) \hookrightarrow \mathrm{TP}(\mathrm{Df})$

6. $\mathrm{TP}(s) \hookrightarrow \mathrm{TIP}(s)$ *for* $s \in \{\mathrm{Cu}, \mathrm{Tf}, \mathrm{Df}\}$

7. $\mathrm{STIP}(s) \hookrightarrow \mathrm{TIP}(s)$ *for* $s \in \{\mathrm{Cu}, \mathrm{Tf}, \mathrm{Df}\}$

8. $\mathrm{STIP}(s) \hookrightarrow \mathrm{TP}(s)$ *for* $s \in \{[], \mathrm{Ch}\}$

9. $\mathrm{IHP}(s) \longleftrightarrow \mathrm{IHP}(t)$ *for* $s, t \in \{\mathrm{Cu}, \mathrm{Tf}, \mathrm{Df}, [], \mathrm{Ch}\}$

Proof.

1. $\Gamma \vdash_s M : A$ if and only if $\Gamma, z{:}A \to Z \vdash_s zM : B$ for some B.

2. $\Gamma \vdash_s M : B$ for some B if and only if $\Gamma, z{:}Z \vdash_s (\lambda v.z)M : Z$.

3. $\Gamma \vdash_s M : B$ for some B if and only if $\Gamma, z{:}\forall X.(X \to Z) \vdash_s z[]M : Z$.

4. Let $\Gamma = \{a_1 : A_1, \ldots, a_n : A_n\}$. $\Gamma \vdash_s M : B$ for some B if and only if $\Sigma \vdash_s z(\lambda a_1 : A_n \ldots \lambda a_n : A_n . M) : B$ for some B and some Σ.

5. Let $\Gamma = \{a_1 : A_1, \ldots, a_n : A_n\}$. $\Gamma \vdash_s M : B$ for some B if and only if $\Sigma \vdash_s M_0 : B$ for some B and some Σ, where $M_0 = z_0(z_1(z[\forall X.X]))(z_1 z)(z[(A_1 \to \cdots \to A_n \to Y) \to Y](\lambda a_1 \ldots \lambda a_n . y M))$. Note that if M_0 is typable then type of z should be a universal type, to say, $\forall X.\mathsf{F}(X)$, such that $\mathsf{F}(\forall X.X) \doteq \forall X.\mathsf{F}(X)$, where F is a second-order variable with arity 1. Observe that the only solution to the unification equation is $[\mathsf{F} := \lambda x.x]$, which implies that type of z is $\forall X.X$.

6. Let $\{x_1, \ldots, x_n\} = \mathrm{FV}(M)$. $\Sigma \vdash_s M : B$ for some B and some Σ if and only if $\vdash_s \lambda x_1 \ldots \lambda x_n . M : B$ for some B.

7. See the case 6 above. 8. See the case 4 above.

9. By Proposition 2 (erasure and lifting). □

3. Domain-free $\lambda 2$ and ML

We introduce a fragment of domain-free $\lambda 2$, called domain-free ML, for which type-related problems become undecidable [10], and over which domain-free $\lambda 2$ is conservative.

- Monotypes $\tau ::= X \mid (\tau \to \tau)$ • Polytypes $\sigma ::= \tau \mid \forall X.\sigma$ • Contexts $\Gamma ::= \langle\ \rangle \mid x : \sigma, \Gamma$

- Terms

  $M ::= x \mid (\lambda x.M) \mid (MM) \mid (x[\tau_1] \cdots [\tau_n])$

- Type assignment rules for domain-free ML:

$$\frac{\Gamma(x) = \forall X_1 \cdots X_n . \tau \quad (n \geq 0)}{\Gamma \vdash_{\mathrm{dfML}} x[\tau_1] \cdots [\tau_n] : \tau[X_1 := \tau_1, \ldots, X_n := \tau_n]}\ (\mathrm{var})$$

$$\frac{\Gamma, x : \tau_1 \vdash_{\mathrm{dfML}} M : \tau_2}{\Gamma \vdash_{\mathrm{dfML}} \lambda x.M : \tau_1 \to \tau_2}\ (\to I) \qquad \frac{\Gamma \vdash_{\mathrm{dfML}} M_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash_{\mathrm{dfML}} M_2 : \tau_1}{\Gamma \vdash_{\mathrm{dfML}} M_1 M_2 : \tau_2}\ (\to E)$$

- Well-formed expressions denoted by T, U for second-order unification are defined from type variables, $\to$, and second-order functional variable $\mathsf{F}^{(n)}$ with arity $n \geq 1$.

$$T, U ::= X \mid \mathsf{F}^{(n)} \tau_1 \cdots \tau_n \mid (T \to U)$$

- The set of variables of well-formed unification expressions is defined by Uvar:

  $\mathrm{Uvar}(X) = \emptyset$, $\mathrm{Uvar}(\mathsf{F}^{(n)} \tau_1 \cdots \tau_n) = \{\mathsf{F}\}$, $\mathrm{Uvar}(T_1 \to T_2) = \mathrm{Uvar}(T_1) \cup \mathrm{Uvar}(T_2)$
  The set of constants is defined by Const:

  $\mathrm{Const}(X) = \{X\}$; $\mathrm{Const}(\mathsf{F}\tau_1 \ldots \tau_n) = \emptyset$; $\mathrm{Const}(T_1 \to T_2) = \mathrm{Const}(T_1) \cup \mathrm{Const}(T_2)$

- Let T_1 and T_2 be well-formed expressions, and $\{\mathsf{F}_1^{(k_1)}, \ldots, \mathsf{F}_n^{(k_n)}\} = \mathrm{Uvar}(T_1, T_2)$.

  A unification problem $T_1 \doteq T_2$ is solvable if there exists a substitution $S = [\mathsf{F}_1^{(k_1)} := \lambda X_1 \ldots X_{k_1} . U_1, \ldots, \mathsf{F}_n^{(k_n)} := \lambda X_1 \ldots X_{k_n} . U_n]$ with $\mathrm{Uvar}(U_i) = \emptyset$ for each i, such that $S(T_1) =_\beta S(T_2)$.

Theorem 1 (Schubert [21]) *The second-order unification problem on well-formed expressions (simple instances) is undecidable.*

Given $T_1 \doteq T_2$, then define an encoding $M_{T_{1,2}}$ and a signature Σ as follows:
$\Sigma = \Sigma_0 \cup \Sigma_1$ together with
$\Sigma_1 = \{x_{\mathsf{F}} : \forall X_1 \ldots X_n.\mathsf{F}X_1 \ldots X_n \mid \mathsf{F}^{(n)}$ is a second-order variable in T_1 or $T_2\}$,
and $\Sigma_0 = \{x : X \mid X \in \mathrm{Const}(T_1, T_2)\}$.
$M_{T_{1,2}} = \lambda z_1.\lambda z_2.\lambda f.fz_1(fz_2(\lambda g.g[\![T_1]\!]^{z_1}_{\Sigma}[\![T_2]\!]^{z_2}_{\Sigma}))$ where

1. Case T of X:
   - $[\![X]\!]^z_\Sigma = \lambda f.fz(fx(\lambda g.g))$ where $\Sigma(x) = X$,
   - $(\!| X |\!) = (X \to (X' \to X') \to X' \to X') \to X' \to X'$
     where X' is a fresh type variable;
2. Case T of $\mathsf{F}^{(n)}\tau_1 \ldots \tau_n$:
   - $[\![\mathsf{F}\tau_1 \ldots \tau_n]\!]^z_\Sigma = \lambda f.fz(f(x[\tau_1]\ldots[\tau_n])(\lambda g.g))$
     where $\Sigma(x) = \forall X_1 \ldots X_n.\mathsf{F}X_1 \ldots X_n$,
   - $(\!| \mathsf{F}\tau_1 \ldots \tau_n |\!) = ((\mathsf{F}\tau_1 \ldots \tau_n) \to (X \to X) \to X \to X) \to X \to X$
     where X is a fresh type variable;
3. Case T of $(T_1 \to T_2)$:
   - $[\![T_1 \to T_2]\!]^z_\Sigma = \lambda z_1.\lambda z_2.\lambda f.f(zz_1)(fz_2(\lambda g.g([\![T_1]\!]^{z_1}_\Sigma)([\![T_2]\!]^{z_2}_\Sigma)))$,
   - $(\!| T_1 \to T_2 |\!) = T_1 \to T_2 \to (T_2 \to A \to A) \to A$
     where $A = ((\!| T_1 |\!) \to (\!| T_2 |\!) \to X) \to X$ with fresh X.

Lemma 1 *For any substitution S, we have $S(\Sigma), z{:}S(T) \vdash_{\mathrm{dfML}} [\![T]\!]^z_\Sigma : S(\!| T |\!)$.*

Proposition 5 *$S(T_1) =_\beta S(T_2)$ if and only if $\Gamma, \Sigma_0 \vdash_{\mathrm{dfML}} M_{T_{1,2}} : \tau$ for some Γ, τ.*

Note that an encoding $M_{T_{1,2}}$ is in β-normal, see [10] for the details.

- Types with rank-k denoted by $R(k)$ are defined as follows:
  $R(0) ::= \tau$ (monotypes)
  $R(k+1) ::= R(k) \mid R(k) \to R(k+1) \mid \forall X.R(k+1)$

Proposition 6 (Conservative extension) *Let M_{nf} be a term in β-normal form of domain-free* ML. *$\Gamma \vdash_{\mathrm{dfML}} M_{\mathrm{nf}} : \tau$ if and only if $\Gamma \vdash_{\mathrm{Df}} M_{\mathrm{nf}} : \tau$.*

Theorem 2 (Domain-free $\lambda 2$) *1.* TCP(Df), TIP(Df) *and* TP(Df) *are all equivalent and undecidable for domain-free $\lambda 2$.*

2. In particular, STIP(Df) *is undecidable at rank-*2 *(even for* ML*).* TIP(Df) *is undecidable at rank-*2*, and* TCP(Df) *is undecidable at rank-*3*.*

Proof. Following Propositions 4 and 5, we show that STIP(dfML) is reduced to TIP(Df) for $\lambda 2$ as follows:
$\Sigma_0, \Gamma \vdash_{\mathrm{dfML}} M_{\mathrm{nf}} : \tau$ for some Γ, τ
$\iff \Sigma_0, \Gamma \vdash_{\mathrm{Df}} M_{\mathrm{nf}} : \tau$ for some Γ, τ by Proposition 6 (Conservativity)
$\iff \Sigma_0 \vdash_{\mathrm{Df}} \lambda\vec{x}.M_{\mathrm{nf}} : \sigma_1 \to \cdots \to \sigma_n \to \tau$ for some $\sigma_1, \ldots, \sigma_n, \tau$ by Proposition 3
Note that $(\sigma_1 \to \cdots \to \sigma_n \to \tau) \in R(2)$. Moreover, we have the following:
$\Gamma_0 \vdash_{\mathrm{Df}} M : \rho$ for some $\rho \in R(2)$ iff $\Gamma_0, z{:}Z \vdash_{\mathrm{Df}} (\lambda v.z)M : Z$
Here, remarked that $\lambda v.z : (\rho \to Z) \in R(3)$ for some $\rho \in R(2)$. □

4. Type-free $\lambda 2$

We first introduce second-order unification for simple instances and simple instances in flat form, which are both undecidable. Then the undividable unification problem in flat form is reduced to TCP(Tf) for type-free $\lambda 2$ [12].

4.1. Second-order unification in flat form

- Expressions for unification problems are defined from first-order variables denoted by X and constants denoted by C, together with $\to$ and second-order functional variables denoted by $\mathsf{F}^{(n)}$ with arity n:

$$A, B ::= X \mid C \mid (A \to B) \mid \mathsf{F}^{(n)} A_1 \cdots A_n$$

  The set of variables in unification expressions is defined by FVu:

  $\mathrm{FVu}(X) = \{X\}$, $\mathrm{FVu}(C) = \emptyset$, $\mathrm{FVu}(A \to B) = \mathrm{FVu}(A) \cup \mathrm{FVu}(B)$,

  $\mathrm{FVu}(\mathsf{F}^{(n)} A_1 \cdots A_n) = \{\mathsf{F}\} \cup \bigcup_i \mathrm{FVu}(A_i)$

- A second-order unification problem for simple instances E_s is defined by the following two equations:

$$\begin{aligned} \mathsf{F}A_1 \cdots A_n &\doteq (\mathsf{F}A'_1 \cdots A'_n) \to A'_0 \\ \mathsf{G}B_1 \cdots B_m &\doteq (\mathsf{G}B'_1 \cdots B'_m) \to (\mathsf{F}A''_1 \cdots A''_n) \end{aligned}$$

  where $\mathrm{FVu}(A_i, A'_i, A''_i, B_j, B'_j) = \emptyset$ for each i, j.

Theorem 3 (Schubert [21]) *The second-order unification problem for simple instances is undecidable.*

- Simple instances with root restriction E_{sr} is defined as follows:

$$\begin{aligned} \mathsf{F}A_1 \ldots A_n &\doteq X_{\mathsf{F}A'_1 \ldots A'_n} \to A'_0 & \qquad \mathsf{F}A'_1 \ldots A'_n &\doteq X_{\mathsf{F}A'_1 \ldots A'_n} \\ \mathsf{G}B_1 \ldots B_m &\doteq Y_{\mathsf{G}B'_1 \ldots B'_m} \to X_{\mathsf{F}A''_1 \ldots A''_n} & \qquad \mathsf{G}B'_1 \ldots B'_m &\doteq Y_{\mathsf{G}B'_1 \ldots B'_m} \\ & & \qquad \mathsf{F}A_1 \ldots A_n &\doteq X_{\mathsf{F}A''_1 \ldots A''_n} \end{aligned}$$

  where $X_{\mathsf{F}A'_1 \ldots A'_n}$, $X_{\mathsf{F}A'_1 \ldots A'_n}$, $Y_{\mathsf{G}B'_1 \ldots B'_m}$, $X_{\mathsf{F}A''_1 \ldots A''_n}$, $Y_{\mathsf{G}B'_1 \ldots B'_m}$, and $X_{\mathsf{F}A''_1 \ldots A''_n}$ are all fresh first-order variables.

- Simple instances in flat form[3] is defined as follows, and all the whole is denoted by E_{sf}:

$$\begin{aligned} X_{A_1 \ldots A_n} &\doteq X_{\mathsf{F}' A_1 \ldots A_n} \to A'_0 \\ \mathsf{F}' X_1 \ldots X_n &\doteq X_{A_1 \ldots A_n} \to A_1 \to \cdots \to A_n \to o \\ \mathsf{F}' C_1 \ldots C_n &\doteq X'_{A_1 \ldots A_n} \to C_1 \to \cdots \to C_n \to o \end{aligned}$$

  where F' is a fresh second-order variable with n-arity; $X_i, X_{A_1 \ldots A_n}, X_{\mathsf{F}' A_1 \ldots A_n}, X'_{A_1 \ldots A_n}$ are fresh first-order variables; $C_1, \ldots, C_n$ are pair wise distinct constants; and o is a distinguished constant.

Lemma 2 *The simple instances E_s are solvable if and only if the simple instances in the flat form E_{sf} are solvable.*

Proposition 7 ([11]) *The second-order unification problem for simple instances in flat form is undecidable.*

[3] Here we show only the flat form with respect to the first equation $\mathsf{F}A_1 \ldots A_n \doteq X_{\mathsf{F}A'_1 \ldots A'_n} \to A'_0$ of E_{sr}. E_{sf} consists of totally $3 * 5 = 15$ equations including two second-order variables.

4.2. Reduction from flat form to TCP(Tf), TIP(Tf) and TP(Tf)

First-order variables and constants in unification problems are interpreted as type variables of $\lambda 2$. Suppose that we have one-to-one mappings between unification expressions and term variables of $\lambda 2$, for which we write x_A, y_A, and so on. We define a context $\Gamma_o = \{x_o : o, y_{o_1} : o \to o, \ldots, y_{o_k} : o^k \to o\}$ for some fixed k.

Definition 3 (Encoding of 1st-order unification)

- *Encoding M_A of a first-order expression A:*
  1. *Case A of a variable X:* $M_X = y_{o_1}(y_X x_X)$
  2. *Case A of a constant C:* $M_C = y_{o_1}(y_C x_C)$
  3. *Case A of $(A_1 \to A_2)$:*
     $M_{A_1 \to A_2} = y_{o_4}(y_{A_2}(x_{A_1 \to A_2} x_{A_1}))(y_{A_1 \to A_2} x_{A_1 \to A_2}) M_{A_1} M_{A_2}$
- *Encoding M_E of a first-order unification problem $E = \{A \doteq B\} \cup E_1$:*
  $M_E = y_{o_5}(y_A x_A)(y_A x_B) M_A M_B M_{E_1}$ *and* $M_\emptyset = x_o$

Note that $\mathrm{FV}(M_A) \subseteq \mathrm{dom}(\Gamma_o) \cup \{x_B, y_B \mid B \text{ is a sub-expression of } A\}$. Let S be a substitution and A be an expression. Then remarked that we have $\Gamma \vdash_{\mathrm{Tf}} M_A : o$ for any $\Gamma \supseteq \Gamma_o$, such that $\Gamma(x_B) = S(B)$ and $\Gamma(y_B) = S(B) \to o$ for each sub-expression B of A.

Lemma 3 *Let E be a finite set of first-order unification equations. A substitution S solves E if and only if there exists $\Gamma \supseteq \Gamma_o$ such that $\Gamma \vdash_{\mathrm{Tf}} M_E : o$, where $\Gamma(x_B) = S(B)$ and $\Gamma(y_B) = S(B) \to o$ for each sub-expression B of equations in E.*

As a shorthand, we define $\lambda 2$-terms such that $M[]^{n+1} = (M[])[]^n$, $M[]^0 = M$ and $\Lambda^{n+1}.M = \Lambda^n.(\Lambda.M)$, $\Lambda^0.M = M$.

Definition 4 (Encoding of 2nd-order unification) *Given a second-order unification problem $E = E_0 \cup E_1$, where E_0 consists of the following equations in flat form:*
$\mathsf{F}X_1 \ldots X_n \doteq B_1$ *with* $B_1 = (X \to A_1 \to \cdots \to A_n \to o)$
$\mathsf{F}C_1 \ldots C_n \doteq B_2$ *with* $B_2 = (X' \to C_1 \to \cdots \to C_n \to o)$,
then we define an encoding M_E as follows:
$y_{o_9}(y_{\mathsf{F}} x_{\mathsf{F}})(y_{\mathsf{F}}(\Lambda^n.\lambda v_1 \ldots \lambda v_{n+1}.x_o))(y_{B_1} x_{B_1})(y_{B_1}(x_{\mathsf{F}}[]^n))(y_{B_2} x_{B_2})(y_{B_2}(x_{\mathsf{F}}[]^n)) M_{B_1} M_{B_2} M_{E_1}$

Proposition 8 *Let E be a finite set of second-order unification problem in flat form. A substitution S solves E if and only if there exists a context $\Gamma \supseteq \Gamma_o$ such that $\Gamma \vdash_{Tf} M_E : o$, where $\Gamma(x_{\mathsf{F}}) = \forall Z_1 \ldots Z_n.S(\mathsf{F})Z_1 \ldots Z_n$ and $\Gamma(y_{\mathsf{F}}) = \forall Z_1 \ldots Z_n.S(\mathsf{F})Z_1 \ldots Z_n \to o$ for each second-order variable $\mathsf{F}^{(n)}$ in E.*

Theorem 4 (Type-free $\lambda 2$) TCP(Tf), TIP(Tf) *and* TP(Tf) *are all undecidable for type-free $\lambda 2$.*

Proof. Let E be a flat form, $\vec{y} = \mathrm{FV}(M_E) \backslash (\{x_o\} \cup \vec{x}_C)$ where $\vec{x}_C = \{x_{C_1}, \ldots, x_{C_m}\}$ is all the constants in equations in E, and $\Gamma = \Gamma_o \cup \{x_C : C \mid C \text{ is a constant in equations in } E\}$. Then the flat form is reduced to TCP and TIP, as follows:

A flat form E is solvable
$\iff \Gamma, v : \forall X.(X \to o) \vdash_{\mathrm{Tf}} v[](\lambda \vec{y}.M_E) : o$
$\iff \Gamma, v : \forall X.(X \to o) \vdash_{\mathrm{Tf}} v[](\lambda \vec{y}.M_E) : A$ for some type A
Next, let Γ_E be

$\{f{:}(\forall C_1 \cdots \forall C_{m+1}.(C_{i_1} \to \cdots \to C_{i_{m+1}} \to W_1)) \to W_2, g{:}W_1\}$,
where $(i_1, \ldots, i_{m+1})$ is a permutation of $(1, \ldots, m+1)$. Let $\hat{M}_E$ be
$f(\Lambda^{m+1}.\lambda x_{C_1} \ldots \lambda x_{C_m}.\lambda x_o.(\lambda v.g)(\lambda \vec{y}.M_E))$,
where $\mathrm{FV}(\hat{M}_E) = \{f, g\}$. Then we can define a term N_f typable under some Γ', such that $\Gamma'(f) = (\forall C_1 \cdots \forall C_{m+1}.(C_{i_1} \to \cdots \to C_{i_{m+1}} \to W_1)) \to W_2$, see also [12] for the details. Therefore, the flat form is reduced to TP as follows:
E is solvable $\Longleftrightarrow \Gamma_E \vdash_{\mathrm{Tf}} \hat{M}_E : W_2 \Longleftrightarrow vN_f\hat{M}_E$ is typable $\square$

5. Hole-application $\lambda 2$

Inference rules for hole-application $\lambda 2$ are recalled and listed as follows:

$$\frac{}{\Gamma \vdash_{\mathrm{hole}} x : \Gamma(x)}\ (\mathrm{var})$$

$$\frac{\Gamma, x{:}A_1 \vdash_{\mathrm{hole}} M : A_2}{\Gamma \vdash_{\mathrm{hole}} \lambda x{:}A_1.M : A_1 \to A_2}\ (\to I) \qquad \frac{\Gamma \vdash_{\mathrm{hole}} M_1 : A_1 \to A_2 \quad \Gamma \vdash_{\mathrm{hole}} M_2 : A_1}{\Gamma \vdash_{\mathrm{hole}} M_1 M_2 : A_2}\ (\to E)$$

$$\frac{\Gamma \vdash_{\mathrm{hole}} M : A}{\Gamma \vdash_{\mathrm{hole}} \Lambda X.M : \forall X.A}\ (\forall I)^* \qquad \frac{\Gamma \vdash_{\mathrm{hole}} M : \forall X.A}{\Gamma \vdash_{\mathrm{hole}} M[] : A[X := B]}\ (\forall E)$$

5.1. TP for hole-application $\lambda 2$

Theorem 5 (Hole-application $\lambda 2$) *TP([]) is undecidable for hole-application $\lambda 2$.*

Proof. Given a second-order unification problem $E = E_0 \cup E_1$, where E_0 consists of the following equations in flat form:
$\mathsf{F}X_1 \ldots X_n \doteq B_1$ with $B_1 = (X \to A_1 \to \cdots \to A_n \to o)$
$\mathsf{F}C_1 \ldots C_n \doteq B_2$ with $B_2 = (X' \to C_1 \to \cdots \to C_n \to o)$,
then we define an encoding H_E as follows:
$y_{o_{11}}(y_{\mathsf{F}}x_{\mathsf{F}})(y_{\mathsf{F}}(\Lambda^n.(\Lambda X.\Lambda X_1 \ldots \Lambda X_n.\lambda x{:}X.\lambda x_1{:}X_1 \ldots \lambda x_n{:}X_n.x_o)[]^{n+1})$
$(y_{B_1}x_{B_1})(y_{B_1}(x_{\mathsf{F}}[]^n))(y_{B_1}((\Lambda X.\lambda x{:}X.\lambda x_1{:}A_1 \ldots \lambda x_n{:}A_n.x_o)[]))$
$(y_{B_2}x_{B_2})(y_{B_2}(x_{\mathsf{F}}[]^n))(y_{B_2}((\Lambda X'.\lambda x'{:}X'.\lambda x_1{:}C_1 \ldots \lambda x_n{:}C_n.x_o)[]))M_{B_1}M_{B_2}M_{E_1}$
Let E be a flat form, and $\Sigma = \Gamma_o \cup \{x_C{:}C \mid C$ is a constant in equations in $E\}$. Now the flat form is reduced to TP([]) as follows:
A flat form E is solvable
$\Longleftrightarrow \Sigma, \Gamma_y \vdash_{\mathrm{hole}} H_E : o$ for some Γ_y
$\Longleftrightarrow \Gamma_y \vdash_{\mathrm{hole}} \lambda v{:}\forall X.(X \to o).v[](\lambda \vec{y}_{o^n}{:}(\vec{o} \to o).\lambda \vec{x}{:}\vec{C}.H_E) : A$ for some A and Γ_y $\square$

5.2. TCP and TIP for hole-application $\lambda 2$

TCP([]) and TIP([]) are equivalent, and we provide a type inference algorithm $\mathsf{type}(\Gamma; M)$ for a context Γ and a hole-application term M, where we write S for a substitution for unification variables:

1. $\mathsf{type}(\Gamma; x) = \Gamma(x)$
2. $\mathsf{type}(\Gamma; \lambda x{:}A.M) = (A \to \mathsf{type}(\Gamma, x{:}A; M))$
3. $\mathsf{type}(\Gamma; MN) =$
   let $B_1 = \mathsf{type}(\Gamma; M)$ and
   let $B_2 = \mathsf{type}(\Gamma; N)$ and $S = \mathsf{unify}(B_1, B_2 \to \alpha)$
   in $S(\alpha)$ (* α is a fresh unification variable *)

Styles	TCP		TIP		TP
Church	yes	$\hookrightarrow$	yes	$\hookrightarrow$	no [21]
Hole-application	Yes	$\longleftrightarrow$	Yes	$\hookrightarrow$	No
Domain-free	No	$\longleftrightarrow$	No	$\longleftrightarrow$	No [10]
Type-free	No	$\longleftrightarrow$	No	$\hookleftarrow$	No [12]
Curry	no	$\longleftrightarrow$	no	$\hookleftarrow$	no [25]

Figure 1: Decidability of TCP, TIP, and TP for $\lambda 2$

$? \vdash M :?$	Church	Domain-free	Type-free	Curry
$\lambda 2$	no	No	No	no [25]
rank 2	no	No (STIP)		yes [15]
ML	no [21]	No (STIP)		yes [16]

Figure 2: Decidability of TP parameterized with ranks

4. $\mathsf{type}(\Gamma; \Lambda X.M) =$

   let $B = \mathsf{type}(\Gamma; M)$ and $X \notin \mathrm{FV}(\Gamma)$

   in $\forall X.S(B)$ (* S is an arbitrary substitution for unification variables *)

5. $\mathsf{type}(\Gamma; M[]) =$

   let $B = \mathsf{type}(\Gamma; M)$ and $S = \mathsf{unify}(\forall X.\mathsf{F}(X), B)$

   in $S(\mathsf{F})(\beta)$ (* β is a fresh unification variable, and F is a fresh 2nd-order unification variable with arity 1 *)

Proposition 9 (Soundness and completeness of type) *1. If* $\mathsf{type}(\Gamma; M) = A$ *then* $\Gamma \vdash_{\mathrm{hole}} M : A$.

2. Given a context Γ *and a term* M*, let* A *be a type such that* $\Gamma \vdash_{\mathrm{hole}} M : A$*. Then we have* $\mathsf{type}(\Gamma; M) = B$ *such that* $A = S(B)$ *for some substitution* S *for unification variables.*

Proof. By induction on the structure of M. □

6. Summary of results

We summarize the results on the type-related problems, where "yes" means that a problem is decidable and "no" means undecidable. "No" denotes the results obtained in [10, 12], and "Yes" is obtained in this article. Figure 1 shows the decidability results for $\lambda 2$ and relations on the type-related problems. Figure 2, 3, and 4 show the decidability results for TP, TIP, and TCP, respectively, parameterized with type-ranks.

7. Inhabitation problems for fragments of $\lambda 2(\neg, \to, \wedge, \vee, \exists)$

We write $\lambda 2(\neg, \to, \wedge, \vee, \exists)$ for second-order intuitionistic propositional logic consisting of $\neg, \to, \wedge, \vee$ and $\exists$, i.e., a second-order $\forall$-free fragment. In particular, a $(\neg, \wedge, \exists)$-fragment of $\lambda 2(\neg, \to, \wedge, \vee, \exists)$ is introduced in [9], and it is established that a typing relation correspondence between $\lambda 2(\to, \forall)$ and $\lambda 2(\neg, \wedge, \exists)$, as follows:

$\Gamma \vdash M$:?	Church	Domain-free	Type-free	Curry
$\lambda 2$	yes	No	No	no [25]
rank 2	yes	No		yes [15]
ML	yes	yes	yes	yes

Figure 3: Decidability of TIP parameterized with ranks

$\Gamma \vdash M : A$?	Church	Domain-free	Type-free	Curry
$\lambda 2$	yes	No	No	no
rank 3	yes	No		no [25]
rank 2	yes			yes [17]
ML	yes	yes	yes	yes

Figure 4: Decidability of TCP parameterized with ranks

$\Gamma \vdash M : A$ in $\lambda 2(\rightarrow, \forall)$ if and only if $\neg\Gamma^* \vdash \lambda a.M^* : \neg A^*$ in $\lambda 2(\neg, \wedge, \exists)$ under a negative translation $*$. However, this correspondence does not imply the undecidable result of IHP for $\lambda 2(\neg, \wedge, \exists)$, where it is undecidable for $\lambda 2(\rightarrow, \forall)$ [18, 22]. Indeed, IHP of $\lambda 2(\neg, \wedge, \exists)$ should be decidable [20, 23].

On the other hand, $\lambda\mu 2(\neg, \rightarrow, \wedge, \vee, \exists)$ denotes second-order classical propositional logic consisting of $\neg, \rightarrow, \wedge, \vee$ and $\exists$, i.e., $\lambda 2(\neg, \rightarrow, \wedge, \vee, \exists)$ plus the reductio ad absurdum rule: If $\Gamma, \neg A \vdash_{\lambda\mu} \bot$ then $\Gamma \vdash_{\lambda\mu} A$. In this section, we show the decidable results of IHP for fragments of $\lambda 2(\neg, \rightarrow, \wedge, \vee, \exists)$, which can be proved only by an elementary method.

Proposition 10 (Glivenko's theorem) *$\Gamma \vdash A$ in $\lambda\mu 2(\neg, \rightarrow, \wedge, \vee, \exists)$ if and only if $\neg\neg\Gamma \vdash \neg\neg A$ in $\lambda 2(\neg, \rightarrow, \wedge, \vee, \exists)$.*

Proof. By induction on the derivation. □

7.1. Two values semantics of 2nd-order classical propositional logic

Let v be a valuation from the set of propositional variables to the set of truth values $\{\mathtt{0}, \mathtt{1}\}$, and naturally extended to a function from the set of formulae to $\{\mathtt{0}, \mathtt{1}\}$. We write $v(X : b)$ for an updated function, such that $v(X : b)[\![Y]\!] = v[\![Y]\!]$ for $Y \not\equiv X$ and $v(X : b)[\![Y]\!] = b$ for $Y \equiv X$. We write $\Gamma \models A$ if for any v, $v[\![B]\!] = \mathtt{1}$ for each $B \in \Gamma$ implies $v[\![A]\!] = \mathtt{1}$.

1. $v[\![\bot]\!] = \mathtt{0}$, $v[\![\top]\!] = \mathtt{1}$
2. $v[\![\neg A]\!] = \mathtt{1} - v[\![A]\!]$
3. $v[\![A_1 \wedge A_2]\!] = \min\{v[\![A_1]\!], v[\![A_2]\!]\}$
4. $v[\![A_1 \vee A_2]\!] = \max\{v[\![A_1]\!], v[\![A_2]\!]\}$
5. $v[\![A_1 \rightarrow A_2]\!] = \max\{\mathtt{1} - v[\![A_1]\!], v[\![A_2]\!]\}$
6. $v[\![\exists X.A]\!] = \max\{v(X : \mathtt{0})[\![A]\!], v(X : \mathtt{1})[\![A]\!]\}$
7. $v[\![\forall X.A]\!] = \min\{v(X : \mathtt{0})[\![A]\!], v(X : \mathtt{1})[\![A]\!]\}$.

Lemma 4 *1. If* $v[\![\exists X.A]\!] = \mathtt{1}$ *then* $v[\![A[X := A[X := \top]]]\!] = \mathtt{1}$.

2. If $v[\![A[X := A[X := \bot]]]\!] = \mathtt{1}$ *then* $v[\![\forall X.A]\!] = \mathtt{1}$.

Proposition 11 (Decidability of $(\neg, \wedge, \exists)$, $(\neg, \wedge, \rightarrow, \exists)$-fragments) *1. Let A be a formula of $(\neg, \wedge, \exists)$-fragment. If $\models A$ then $\vdash A$ in $\lambda 2(\neg, \wedge, \exists)$.*

2. Let A be a formula of $(\neg, \wedge, \rightarrow, \exists)$-fragment and $\mathrm{FV}(A) = \emptyset$. If $\Gamma \models A$ then $\Gamma \vdash A$ in $\lambda 2(\neg, \wedge, \rightarrow, \exists)$.

Proof. By induction on the length $l(A)$ of a formula A, together with Proposition 10 (Glivenko's theorem) and Lemma 4:

1. $l(\bot) = l(\top) = l(X) = 1$, $l(\neg A) = l(A) + 1$;
2. $l(A_1 \rightarrow A_2) = l(A_1 \wedge A_2) = l(A_1) + l(A_2) + 1$;
3. $l(\exists X.A) = l(A[X := \alpha]) + 1$ where $l(\alpha) = l(A)$. □

Even for a second-order intuitionistic fragment $\lambda 2(\neg, \wedge, \exists, \forall)$ with $\forall$, the inhabitation problem is proved to be decidable, see Sakagawa [20] and Tatsuta, *et al.* [23] for the details. To the best of our knowledge, IHP for $\lambda 2(\rightarrow, \exists)$ in general still remains open.

References

[1] H. P. Barendregt: *The lambda Calculus. Its Syntax and Semantics*, North-Holland, second, revised edition, 1984.

[2] H. P. Barendregt: *Lambda calculi with types*, In S. Abramskty, *et al.* editors, *Handbook of Logic in Computer Science*, Vol II, pp. 117–309, Oxford University Press, 1992.

[3] G. Barthe, M. H. Sørensen: *Domain-Free Pure Type Systems*, Lecture Notes in Computer Science 1234, pp. 9–20, 1997.

[4] H.-J. Boehm: *Partial polymorphic type inference is undecidable*, Proc. IEEE 26th Annual Symposium on Foundations of Computer Science, pp. 339–345, 1985.

[5] A. Church: *A set of postulates for the foundation of logic*, Annals of Math. 33(2), pp. 346–366, 1932 and 34(4), pp. 839–864, 1933.

[6] A. Church: *A formulation of the simple theory of types*, J. Symbolic Logic 5, pp. 56–68, 1940.

[7] A. Church: *The Calculi of Lambda-Conversion*, Princeton University Press, 1941.

[8] H. B. Curry: *Functionality in combinatory logic*, Proc. Nat. Acad. Science USA, 20, pp. 584–590, 1934.

[9] K. Fujita: *CPS-translation as adjoint*, Theoretical Computer Science 411 (2), pp. 324–340, 2010.

[10] K. Fujita, A. Schubert: *Partially typed terms between Church-style and Curry-style*, Lecture Notes in Computer Science 1872, pp. 505-520, 2000.

[11] K. Fujita, A. Schubert: *Existential type systems with no types in terms*, Lecture Notes in Computer Science 5608, pp. 112-125, 2009.

[12] K. Fujita, A. Schubert: *The undecidability of type related problems in type-free style System* F, Leibniz International Proceedings in Informatics 6, pp. 103–118, 2010.

[13] H. Tonino, K. Fujita: *On the adequacy of representing higher order intuitionistic logic as a pure type system*, Ann. Pure Appl. Logic 57, pp. 251–276, 1992.

[14] A. J. Kfoury, J. Tiuryn, P. Urzyczyn: *The undecidability of the semi-unification problem*, STOC '90: Proc. 22nd Annual ACM Symposium on Theory of Computing, pp. 468–476, 1990.

[15] A. J. Kfoury, J. Tiuryn: *Type Reconstruction in Finite Rank Fragments of the Second-Order λ-Calculus*, Information and Computation 98, pp. 228–257, 1992.

[16] A. J. Kfoury, J. Tiuryn, P. Urzyczyn: *An Analysis of ML Typability*, Journal of the Association for Computing Machinery, Vol. 41, No. 2, pp. 368–398, 1994.

[17] A. J. Kfoury, J. B. Wells: *A direct algorithm for type inference in the rank-2 fragment of the second-order λ-calculus*, Proc. ACM LISP and Functional Programming, pp. 196–207, 1994.

[18] M. H. Löb: *Embedding first order predicate in fragments of intuitionistic logic*, J. Symbolic Logic 41 (4), pp. 705–718, 1976.

[19] F. Pfenning: *On the undecidability of partial polymorphic type reconstruction*, Fundamenta Informaticae 19 (1,2), pp. 185–199, 1993.

[20] 坂川 航: **記号が制限された古典/直観主義２階命題論理**, 東京工業大学修士論文, 2009.

[21] A. Schubert: *Second-order unification and type inference for Church-style polymorphism*, POPL '98: Proc. 25th ACM Symposium on Principles of Programming Languages, pp. 279–288, 1998.

[22] M. H. Sørensen, P. Urzyczyn: *Lectures on the Curry-Howard Isomorphism*, Volume 149, Studies in Logic and the Foundations of Mathematics, Elsevier Science Inc., 2006.

[23] M. Tatsuta, K. Fujita, R. Hasegawa, H. Nakano: *Inhabitation of polymorphic and existential types*, Ann. Pure Appl. Logic 161, pp. 1390–1399, 2010.

[24] A. M. Turing: *Computability and λ-definability*, J. Symbolic Logic 2, pp. 153–163, 1937.

[25] J. B. Wells: *Typability and type checking in system* F *are equivalent and undecidable*, Ann. Pure Appl. Logic 98, pp. 111–156, 1999.