On fine structures between Church-style and Curry-style
\(\lambda 2\)-terms

Ken-etsu Fujita (Gunma University)

December 14, 2012

Abstract

We introduce a class of 2nd-order \(\lambda\)-terms with fine structures between so called Church-style and Curry-style. Here, \(\lambda\)-terms in the style of Curry are considered as atomic, and we adopt four term-constructors: (i) Domains (D) for \(\lambda\)-abstraction, (ii) Lambdas (\(\Lambda\)) for type-abstraction, (iii) Holes ([]) for type-application, and (iv) Types (\([A]\)) to be filled into a hole. Then applying the term-constructors to Curry-style provides the set of 12 styles of \(\lambda 2\)-terms in total, where Church-style can be regarded as a top and Curry-style is a bottom. We examine which term-constructor determines decidability of type-checking and type-inference problems of \(\lambda 2\)-terms. This study reveals fine boundaries between decidability and undecidability of the type-related problems.

1 Introduction

Second-order \(\lambda\)-terms in the style of Church consist of variables, applications, \(\lambda\)-abstractions, type applications and type-abstractions [2].

\[ M ::= x | MM | \lambda x:A.M | M[A] | AX \]

On the other hand, \(\lambda\)-terms in the style of Curry is the same as those of type-free \(\lambda\)-calculus. As a natural combinatorial problem, we can consider \(\lambda\)-terms with fine structures between Curry-style and Church-style. From the viewpoint of components of \(\lambda\)-terms, we take (i) domains of \(\lambda\)-abstraction, (ii) type abstractions \(\Lambda\), (iii) holes [ ] to be filled with a type, and (iv) type information (polymorphic instance) to be inserted into a hole, as primitive term constructors for fine structures. We write D, \(\Lambda\), [], and \([A]\), respectively, for the constructors. Then, based on the Curry-style, the following 12 styles (structures) for \(\lambda\)-terms can be defined as a combination of four constructors. We write \(ST\) for the set of 12 styles, as follows:

- Church-style [2] denoted by Ch has constructors (D, \(\Lambda\), \([A]\))
- Domain-free style [4] denoted by Df has ( \(\Lambda\), \([A]\))
- Type-free style [7] denoted by Tf has ( \(\Lambda\), [ ])
- Hole-application style [8] denoted by Hole has (D, \(\Lambda\), [ ])
- (D, [A]), (D, [ ]), (D, \(\Lambda\), ( [A]), ( [ ]), (D ), ( \(\Lambda\) )
- Curry-style [2] denoted by Cu has ( )

The fine structures between Curry-style and Church-style are presented in the following picture, see Figure 1. Upper arrows on the cubes denote adding domains of \(\lambda\)-abstraction, where we only depict one upper arrow among a total of 6 upper arrows in the picture. Four right arrows on the left cube
denote adding holes [], other right arrows on the right cube denote adding polymorphic instance [A], and six back arrows denote adding type abstractions $\Lambda$.

An order is defined on ST: Curry-style is the bottom, Church-style is the top, and $s < t$ if we have an arrow from $s$-style to $t$-style for $s, t \in ST$.

The picture shows that terms on the upper plane contain domains of $\lambda$-abstraction, where the set of styles on the upper plane is denoted by UpP. On the other hand, the set of styles on the lower plane is denoted by LwP. Terms on the back plane contain type abstractions $\Lambda$ where the set of styles on the back plane is denoted by BkP, and terms on the middle plane contain holes [] where the set of styles on the middle plane is denoted by MiP. Terms on the rightmost plane contain polymorphic instance [A], where the set of styles on the right plane is denoted by RiP. The set of styles on the leftmost plane is denoted by LeP.

The first problem is how to define inference rules for each system. The second problem is how to define reduction rules for each system. For this, we call a system normal, if the system contains both $\Lambda$ and either [] or [A], or contains neither $\Lambda$ nor []. Namely, systems of Ch, Hole, Df, Tf, (D), and Cu are normal.

We study decision problems parametrized by $\lambda$-terms with an intermediate structure of the cubes, and investigate critical conditions for the decidability property from the viewpoint of the constructors (D, $\Lambda$, [], and [A]). In this paper, as decision problems we adopt the type checking (TCP), type inference (TIP), and typability (TP) problems for second-order $\lambda$-terms with fine structures. Then we examine what constructor determines essentially (un)decidability of the problems.

2 Preliminary

Definition 1 (Type-related problems parameterized with styles)

1. Type checking problem of s-style terms denoted by TCP(s):
   Given an s-style $\lambda$-term $M$, a type $A$, and a context $\Gamma$, determine whether $\Gamma \vdash_{s} M : A$.

2. Type inference problem of s-style $\lambda$-terms denoted by TIP(s):
   Given an s-style $\lambda$-term $M$ and a context $\Gamma$, determine whether $\Gamma \vdash_{s} M : A$ for some type $A$.

3. Typability problem of s-style terms denoted by TP(s):
   Given an s-style $\lambda$-term $M$, determine whether $\Gamma \vdash_{s} M : A$ for some context $\Gamma$ and type $A$. 
Proposition 1 (Reductions between type-related problems)

1. TCP(s) \rightarrow TIP(s) for any s \in ST.
2. TIP(s) \leftarrow TCP(s) for any s \in LwP \cup MiP \cup LeP.
3. TIP(s) \leftarrow TP(s) for any s \in UpP \cup \{Df, ([A])\}.
4. TP(s) \leftarrow TIP(s) for s \in LwP.

Proof. 1. \Gamma \vdash_{s} M : A if and only if \Gamma, z : A \rightarrow Z \vdash_{s} zM : B for some B, where z, Z are fresh variables.

2. Let s \in LwP. \Gamma \vdash_{s} M : B for some B if and only if \Gamma, z : Z \vdash_{s} (\lambda v. z)M : Z, where z, v, Z are fresh variables with z \neq v.

Let s \in MiP. \Gamma \vdash_{s} M : B for some B if and only if \Gamma, z : \forall X. (X \rightarrow Z) \vdash_{s} z[]M : Z, where z, Z are fresh variables.

Let s \in LeP. \Gamma \vdash_{s} M : B for some B if and only if \Gamma, z : \forall X. (X \rightarrow Z) \vdash_{s} zM : Z, where z, Z are fresh variables.

3. Let s \in UpP. Let \Gamma = \{a_{1} : A_{1}, \ldots, a_{n} : A_{n}\} and z be a fresh variable. \Gamma \vdash_{s} M : B for some B if and only if \Sigma \vdash_{s} z(\lambda a_{1} : A_{1} \ldots \lambda a_{n} : A_{n}. M) : B for some B and some \Sigma.

Let s \in \{Df, ([A])\}. 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 \Sigma, where M_{0} = z_{0}(z_{1}(z_{2}(\forall X. X)))((z_{2}(z_{1}(z_{2}(\forall X. X))))(z_{1}(z_{2})))((A_{1} \rightarrow \cdots \rightarrow A_{n} \rightarrow Y) \rightarrow Y)(\lambda a_{1} \ldots \lambda a_{n}. M)), and z_{0}, z_{1}, z, y, Y are fresh variables.

If \Gamma \vdash_{s} M : B for some B, then M_{0} is typable. Because type of z is assigned to \forall X. X.

In turn, if M_{0} is typable then type of z should be a universal type, to say, \forall X. F(X), where F is a second-order variable with arity 1. From consistent typability of the two occurrences of z_{1}, we have the following unification equation [5]:

\[ F(\forall X. X) \equiv \forall X. F(X) \]

Observe that the only solution to the unification equation is [F := \lambda x. x], i.e., the identity function, which implies that type of z is \forall X. X. Hence, we can recover the context \Gamma.

4. Let \{x_{1}, \ldots, x_{n}\} = FV(M). \Sigma \vdash_{s} M : B for some B and some \Sigma if and only if \Gamma \vdash_{s} \lambda x_{1} \ldots \lambda x_{n}. M : B for some B.

We summarize already known results on the problems for \lambda 2. Table 1 shows the decidability results and relations on the type-related problems. Here, "yes" means that a problem is decidable and "no" undecidable. TCP and TIP have the boundaries between hole-application and domain-free. Compared with Church-style, TIP remains decidable even after deleting polymorphic instance information on application of (\forall E). However, on application of (\rightarrow I), deleting polymorphic domains makes TIP undecidable. Therefore, polymorphic domains are considered as the most essential information for (un)decidable TIP. In this paper, we examine System (D), Curry-style with explicit domains.
3 System (D): Curry-style plus explicit domains

We introduce System (D) and study the type-inference and type-checking problems of the system.

- Types

\[ X \in \text{TypeVars} \]
\[ A \in \text{Types ::= } X | (A \rightarrow A) | \forall X.A \]

- Terms:

\[ M \in \text{Terms ::= } x | (\lambda x : A.M) | (MM) \]

- Reduction rule:

\[ (\lambda x : AM)N \rightarrow_{\beta} M[x := N] \]

- Inference rules:

\[ \frac{\Gamma, x : A \vdash_{D} M : A}{\Gamma \vdash_{D} x : A} \quad (\text{var}) \]
\[ \frac{\Gamma, x : A_{1} \vdash_{D} M : A_{2}}{\Gamma \vdash_{D} \lambda x : A_{1}.M : A_{1} \rightarrow A_{2}} \quad (\rightarrow I) \]
\[ \frac{\Gamma \vdash_{D} M_{1} : A_{1} \rightarrow A_{2}, \Gamma \vdash_{D} M_{2} : A_{1}}{\Gamma \vdash_{D} M_{1}M_{2} : A_{2}} \quad (\rightarrow E) \]
\[ \frac{\Gamma \vdash_{D} M : A}{\Gamma \vdash_{D} M : \forall X.A} \quad (\forall I)^{*} \]
\[ \frac{\Gamma \vdash_{D} M : \forall X.A}{\Gamma \vdash_{D} M[X := A_{1}] : A[X := A_{1}]} \quad (\forall E) \]

where \((\forall I)^{*}\) denotes the eigenvariable condition \(X \notin \text{FV}(\Gamma)\).

Definition 2 (Removing vacuous-\(\forall\))

1. \[ \|x\| = x, \|\lambda x : A.M\| = \lambda x : \|A\|, \|M\|, \|M_{1}M_{2}\| = \|M_{1}\|\|M_{2}\|, \]
2. \[ \|X\| = X, \|A_{1} \rightarrow A_{2}\| = \|A_{1}\| \rightarrow \|A_{2}\|, \]
\[ \|\forall X.A\| = \forall X.\|A\| \text{ for } X \in \text{FV}(A) \]
\[ \|\forall X.A\| = \|A\| \text{ for } X \notin \text{FV}(A) \]
3. \[ \|\Gamma\|(x) = \|\Gamma(x)\| \]

If \(\|A\| = A\) then we say \(A\) has no vacuous \(\forall\).

Lemma 1

1. \[ \|A[X := B]\| = \|A\|[X := \|B\|] \]
2. \[ \|M[X := B]\| = \|M\|[X := \|B\|] \]
3. \[ \text{FV}(A) = \text{FV}(\|A\|) \]

Proof. By induction on the structure of \(A\) or \(M\).

Proposition 2

1. If \(\Gamma \vdash_{D} M : A\) then \(\|\Gamma\| \vdash_{D} \|M\| : \|A\|\).
2. If \(\Gamma \vdash_{D} M : A\) where each application of \((\forall I)\) is not vacuous in the derivation, then for any \(\Gamma', M', A'\) with \(\|\Gamma'\| = \Gamma, \|M'\| = M, \text{ and } \|A'\| = A\) we have \(\Gamma' \vdash_{D} M' : A'\).

Proof. First observe that given \(A\), then any \(B\) such that \(\|B\| = A\) is generated by the following steps with fresh type variables \(\bar{Z}\): (1) Case \(A = X : B = \forall \bar{Z}.X\), (2) Case \(A = (A_{1} \rightarrow A_{2}) : B = \forall \bar{Z} . (A'_{1} \rightarrow A'_{2})\), (3) Case \(A = \forall X.A_{1} : B = \forall \bar{Z} . \forall X.A'_{1}\). By induction on the derivation, we show only the case 2.

1. Case of \(\Gamma \vdash x : \Gamma(x)\):

For any \(\Gamma', M', A'\) with \(\|\Gamma'\| = \Gamma, \|M'\| = x, \text{ and } \|A'\| = \Gamma(x)\), we have \(M' \equiv x \text{ and } A' \equiv \Gamma'(x)\), and then we have \(\Gamma' \vdash x : \Gamma'(x)\).
2. \( \Gamma \vdash MN : B \) from \( \Gamma \vdash M : A \rightarrow B \) and \( \Gamma \vdash N : A \):

From the induction hypotheses, we have \( \Gamma' \vdash M' : C' \) for any \( \Gamma', M', C' \) such that \( \|\Gamma'\| = \Gamma \), \( \|M'\| = M \), \( \|C'\| = A \rightarrow B \), and we have \( \Gamma' \vdash N' : A' \) for any \( \Gamma', N', A' \) such that \( \|\Gamma'\| = \Gamma \), \( \|N'\| = N \), \( \|A'\| = A' \). Here, \( C' \) should be in the form of \( \forall Z. (A' \rightarrow B') \) where \( \|A'\| = A \) and \( \|B'\| = B \). Then from \( \Gamma' \vdash M' : A' \rightarrow B' \) and \( \Gamma' \vdash N' : A' \), we have \( \Gamma' \vdash M'N' : B' \) for any \( \Gamma', M'N', B' \) such that \( \|\Gamma'\| = \Gamma \), \( \|M'N'\| = MN \), \( \|B'\| = B \).

3. \( \Gamma \vdash \lambda x : A. M : A \rightarrow B \) from \( \Gamma, x : A \vdash M : B \):

From the induction hypothesis, we have \( \Gamma', x : A' \vdash M' : B' \) for any \( \Gamma', A', M', B' \) such that \( \|\Gamma', A'\| = \Gamma, A, \|M'\| = M, \|B'\| = B \). Then we have \( \Gamma', x : A' \vdash \lambda x : A'. M' : \forall Z. (A' \rightarrow B') \) where \( \|\Gamma'\| = \Gamma, \|\lambda x : A'. M'\| = \lambda x : A. M, \) and \( \|\forall Z. (A' \rightarrow B')\| = A \rightarrow B \).

4. \( \Gamma \vdash M : \forall X. A \) from \( \Gamma \vdash M : A \) where \( X \notin \text{FV}(\Gamma) \) and \( X \in \text{FV}(A) \):

From the induction hypothesis, we have \( \Gamma', x : A' \vdash M' : B' \) for any \( \Gamma', A', M', B' \) such that \( \|\Gamma', A'\| = \Gamma, \|M'\| = M, \|A'\| = A \). Then from \( X \notin \text{FV}(\Gamma') \), we have \( \Gamma' \vdash M' : \forall Z. \forall X. A' \), where \( \|\forall Z. \forall X. A'\| = \forall X. \|A'\| = \forall X. A \).

5. \( \Gamma \vdash M[X := B] : A[X := B] \) from \( \Gamma \vdash M : \forall X. A \):

From the induction hypothesis, we have \( \Gamma', x : A' \vdash M'[X := B'] : B' \) for any \( \Gamma', A', M', B' \) such that \( \|\Gamma', A'\| = \Gamma, \|M'[X := B']\| = M[X := B] \) and \( \|B'\| = B \). Then we have \( \Gamma' \vdash M'[X := B'] : \forall Z. \forall X. A'[X := B'] \) where \( \|M'[X := B']\| = M[X := B] \) and \( \|\forall Z. \forall X. A'[X := B']\| = \forall X. \|A'[X := B']\| = A[X := B] \).

Lemma 2 (Permutation for bund variables) If \( \Gamma \vdash_D M : \forall X. \forall Y. A \) then \( \Gamma \vdash_D M : \forall Y. \forall X. A \).

Lemma 3 (Substitution lemma 1) If \( \Gamma \vdash_D M : A \) then \( \Gamma[X := B] \vdash_D M[X := B] : A[X := B] \).

Proof. By induction on the derivation. \( \square \)

Lemma 4 (Substitution lemma 2) If \( \Gamma, x : A \vdash_D M : B \) and \( \Gamma \vdash_D N : A \), then \( \Gamma \vdash_D M[x := N] : B \).

Proof. By induction on the first derivation. \( \square \)

Definition 3 ((\forall I)(\forall E)-reduction for (D)) Let \( X \notin \text{FV}(\Gamma) \).

\[
\frac{\Gamma \vdash M : A}{\Gamma \vdash \forall X. A} \quad (\forall I)^* \quad \frac{\Gamma \vdash M : \forall X. A} {\Gamma \vdash M[X := B] : A[X := B]} \quad (\forall E)
\]

Under this definition, we consider only derivations without \((\forall I)(\forall E)\)-redexes. This property is also called the INST-before-GEN property [11]. From now on, we consider derivations for \( \Gamma \vdash_D M : A \) with no vacuous \( \forall \) and the INST-before-GEN property. It is also remarked that \((\forall E)\) may be applied only after \((\forall I), (\rightarrow E), \) or \((\forall E)\).

Definition 4 (Elimination-Introduction relation) 1. \( A \leq F B \) defined \( \Gamma \vdash_D B \) is derived from \( \Gamma \vdash_D A \) by successive application of \((\forall E)\) including null application for some term.

2. \( A \leq_{I(\Gamma)} B \) defined \( \Gamma \vdash_D B \) is derived from \( \Gamma \vdash_D A \) by successive application of \((\forall I)\) including null application for some term, where the eigenvariable condition holds w.r.t. \( \Gamma \).

3. \( A \leq_{F(\Gamma)} B \) defined \( A \leq F C \) and \( C \leq_{I(\Gamma)} B \) for some type \( C \).
For instance, $\forall X. (X \to X) \leq E_{I(\Gamma)} \forall X. \forall Y. ((X \to Z \to Y) \to (X \to Z \to Y))$ where $X,Y \notin \text{FV}(\Gamma)$.

We also write $\Gamma \vdash_{\text{D}} M : A \leq E_{I(\Gamma)} N : B$, if $\Gamma \vdash_{\text{D}} M : A$ derives $\Gamma \vdash_{\text{D}} N : B$ under the relation $A \leq E_{I(\Gamma)} B$. In this case, we have $M : A \leq E_{I(\Gamma)} S(M) : B$ for some substitution $S$ for type variables by the effect of application of $(\forall E)$.

**Lemma 5** ($\leq E_{I(\Gamma)}$) Let $m,n \geq 0$, and neither $A$ nor $B$ has $\forall$ as a top-symbol, and $Y_1, \ldots, Y_m \notin \text{FV}(\Gamma)$. $\forall X_1 \ldots X_n, A \leq E_{I(\Gamma)} \forall Y_1 \ldots Y_m, B$ if and only if $S(A) = B$ for some substitution $S$ with $\text{dom}(S) = \{X_1, \ldots, X_n\}$.

**Proof.** ($\implies$): Suppose $\forall X_1 \ldots X_n, A \leq E_{I(\Gamma)} \forall Y_1 \ldots Y_m, B$. Then $\forall X_1 \ldots X_n, A \leq E_{I(\Gamma)} \forall X_1 := A_1, \ldots, X_n := A_n = S(A) = B$ for some $S$, since $B \leq I(\Gamma) \forall Y_1 \ldots Y_m, B$. Hence, $S(A) = B$ for some $S$.

($\Longleftarrow$): Suppose that $S(A) = B$ for some $S$. Then $\forall X_1 \ldots X_n, A \leq S(A) = B \leq I(\Gamma) \forall Y_1 \ldots Y_m, B$ where each $Y_i \notin \text{FV}(\Gamma)$. 

**Remark 1** Given $A, B, \Gamma$, then it is decidable to check whether $A \leq E_{I(\Gamma)} B$ holds or not.

**Lemma 6** (partial order) Let $A, B, C$ be types with no vacuous-$\forall$.

1. $A \leq E_{I(\Gamma)} A$
2. If $A \leq E_{I(\Gamma)} B$ and $B \leq E_{I(\Gamma)} C$ then $A \leq E_{I(\Gamma)} C$.
3. If $A \leq E_{I(\Gamma)} B$ and $B \leq E_{I(\Gamma)} A$ then $A \equiv B$.

**Proof.** (2) If $\Gamma \vdash_{\text{D}} A \leq E_{I(\Gamma)} B$ and $\Gamma \vdash_{\text{D}} B \leq E_{I(\Gamma)} C$, and then we have $\Gamma \vdash_{\text{D}} A \leq E_{I(\Gamma)} C$. Moreover, if $\Gamma \vdash_{\text{D}} M_1 : A \leq E_{I(\Gamma)} M_2 : B$ and $\Gamma \vdash_{\text{D}} M_2 : B \leq E_{I(\Gamma)} M_3 : C$, and then we have $\Gamma \vdash_{\text{D}} M_1 : A \leq E_{I(\Gamma)} M_3 : C$.

(3) Let $A = \forall X_1 \ldots X_n, A'$ and $B = \forall Y_1 \ldots Y_m, B'$, where $X_1, \ldots, X_n \in \text{FV}(A')$ and $Y_1, \ldots, Y_m \in \text{FV}(B')$. Then $S_1(A') = B'$ and $S_2(B') = A'$ for some $S_1, S_2$ with $\text{dom}(S_1) = \{X_1, \ldots, X_n\}$ and $\text{dom}(S_2) = \{Y_1, \ldots, Y_m\}$. That is, $A'$ and $B'$ are variant, and hence $S_1, S_2$ are bijective. Then $n = m$ and $\forall X_1 \ldots X_n, A' \equiv \forall Y_1 \ldots Y_m, B'$ under permutation for bound variables.

Note that if we have vacuous-$\forall$, then $\forall X Y. X \leq E_{I(\Gamma)} \forall Z. Z$ and $\forall Z. Z \leq E_{I(\Gamma)} \forall X Y. X$, but $\forall X Y. X \neq \forall Z. Z$.

**Lemma 7** (Generation lemma for System (D)) 1. If $\Gamma \vdash x : A$ then $\Gamma(x) \leq E_{I(\Gamma)} A$.

2. If $\Gamma \vdash \lambda x : A. M : B$, then there exist $B_1$ such that $\Gamma, x : A \vdash M : B_1$ and $A \rightarrow B_1 \leq E_{I(\Gamma)} B$.

3. If $\Gamma \vdash M_1 M_2 : A$, then there exist $B_1, B_2, N_1$ such that $\Gamma \vdash N_1 : B_1 \rightarrow B_2$ and $\Gamma \vdash M_2 : B_1$ and $N_1 M_2 : B_2 \leq E_{I(\Gamma)} M_1 M_2 : A$.

**Proof.** By case analysis with the Elimination-Introduction property.

1. Suppose that $\Gamma \vdash x : A$.

   We should start with $\Gamma \vdash x : \Gamma(x)$, and then the only way to derive $\Gamma \vdash x : A$ is that $\Gamma(x) \leq E_{I(\Gamma)} A$.

2. Suppose that $\Gamma \vdash \lambda x : A_1. M : A_2$.

   Under the Elimination-Introduction property, the only way to derive $\Gamma \vdash \lambda x : \Gamma \times \Gamma M : A_1 M : B$ and $\Gamma \rightarrow B \leq E_{I(\Gamma)} A_2$ for some $B$. Here, we cannot apply $(\forall E)$ for $\Gamma \rightarrow B \leq E_{I(\Gamma)} A_2$.

3. Suppose that $\Gamma \vdash M_1 M_2 : A$.

   Under the Elimination-Introduction property, the only way to derive $\Gamma \vdash M_1 M_2 : A$ is that $\Gamma \vdash N_1 : B_1 \rightarrow B_2$ and $\Gamma \vdash M_2 : B_1$ and $N_1 M_2 : B_2 \leq E_{I(\Gamma)} M_1 M_2 : A$ for some $N_1, B_1, B_2$. Here, we may apply $(\forall E)$ for $N_1 N_2 : B_2 \leq E_{I(\Gamma)} M_1 M_2 : A$, if $B_2 = \forall \vec{X}. B_2'$ for some $B_2'$. Then $\vec{X}$ cannot be free in $N_2$ and hence $N_2 \equiv M_2$. 

\[ \square \]
Definition 5  
1. $(\lambda x:A.M)N \rightarrow_{\beta} M[x := N]$  
2. If $M \rightarrow_{\beta} N$ then $RM \rightarrow_{\beta} RN$, $MR \rightarrow_{\beta} NR$, and $\lambda x:A.M \rightarrow_{\beta} \lambda x:A.N$.

Lemma 8 (Abstraction) If $M \rightarrow_{\beta} N$ and $S(M') = M$ for a substitution $S$ for type variables, then there exists a term $N'$ such that $M' \rightarrow_{\beta} N'$ and $S(N') = N$.

Proof. By induction on the derivation of $M \rightarrow_{\beta} N$. □

Proposition 3 (Subject reduction) If $\Gamma \vdash M : A$ and $M \rightarrow_{\beta} N$, then $\Gamma \vdash N : A$.

Proof. By induction on the derivation of $M \rightarrow_{\beta} N$, together with generation lemma.

- Case of $\Gamma \vdash (\lambda x:A.M)N : B$ and $(\lambda x:A.M)N \rightarrow_{\beta} M[x := N]$:

$$
\Gamma, x:A' \vdash M', B_1 \quad \Gamma \vdash \lambda x:A'.M' : A' \rightarrow B_1 \quad (\rightarrow I) \quad \Gamma \vdash N : A' \quad (\rightarrow E) \quad \Gamma \vdash (\lambda x:A'.M')N : B_1 \leq_{E}^{\iota(\Gamma)} B \quad \Gamma \vdash (\lambda x:A.M)N : B
$$

where $S(M') = M$ for some substitution $S$, and $A' = A$ since if $B = \forall \vec{X}.B'$ for some $B'$ then $\vec{X}$ cannot be free in $A'$.

- Case of $\Gamma \vdash RM : B$ and $RM \rightarrow RN$ from $M \rightarrow N$:

We also have $R'M \rightarrow R'N$ from $M \rightarrow N$ where $S(R') = R$ for a substitution $S$.

$$
\Gamma \vdash R' : B_2 \rightarrow B_1 \quad \Gamma \vdash M : B_2 \quad (\rightarrow E) \quad \Gamma \vdash R'M : B_1 \leq_{E}^{\iota(\Gamma)} B \quad \Gamma \vdash RM : B
$$

From the induction hypothesis, we have $\Gamma \vdash N : B_2$, and then $\Gamma \vdash R'N : B_1 \leq_{I(\Gamma)}^{E} RN : B$.

- Case of $\Gamma \vdash MR : B$ and $MR \rightarrow NR$ from $M \rightarrow N$:

$$
\Gamma \vdash M' : B_2 \rightarrow B_1 \quad \Gamma \vdash R : B_2 \quad (\rightarrow E) \quad \Gamma \vdash M' : B_1 \leq_{E}^{\iota(\Gamma)} B \quad \Gamma \vdash MR : B
$$

Since $S(M') = M$ for some substitution $S$, we have $M' \rightarrow N'$ and $S(N') = N$ for some $N'$. From the induction hypothesis, we have $\Gamma \vdash N' : B_2 \rightarrow B_1$, and then $\Gamma \vdash N'R : B_1 \leq_{I(\Gamma)}^{E} NR : B$.

- Case of $\Gamma \vdash \lambda x:A.M : B$ and $\lambda x:A.M \rightarrow \lambda x:A.N$ from $M \rightarrow N$:

$$
\Gamma, x:A \vdash M : B_1 \quad \Gamma \vdash \lambda x:A.M : A \rightarrow B_1 \leq_{I(\Gamma)}^{E} B \quad (\rightarrow I) \quad \Gamma \vdash \lambda x:A.M : B
$$

From the induction hypothesis, we have $\Gamma, x:A \vdash N : B_1$, and then $\Gamma \vdash \lambda x:A.N : A \rightarrow B_1 \leq_{I(\Gamma)}^{E} B$.

Remark 2 If $\lambda x:A.(Mx) : A \rightarrow B$, then $\lambda x:A.(Mx) \rightarrow_{\eta} M' : A_1 \rightarrow B_1$ that is a contravariant such that $A \leq_{E}^{I} A_1$ and $B_1 \leq_{E}^{I} B$. For instance, we have $x : (A \rightarrow \forall X.X) \vdash \lambda a : A.xa : A \rightarrow Z$. Then we have $\lambda a : A.xa \rightarrow_{\eta} x : A \rightarrow \forall X.X$. 

Theorem 1 (Strong normalization) If $\Gamma \vdash_{D} M : A$ then $M$ is strongly normalizing.

Proof. Suppose $\Gamma \vdash_{D} M : A$ then $\Gamma \vdash_{\text{C}} |M| : A$ and the Curry-term $|M|$ is strongly normalizing, where $|\cdot|$ is a forgetful mapping from (D)-terms to Curry-style terms. For (D)-terms $M, N$, if $M \rightarrow_{\beta} N$ then $|M| \rightarrow_{\beta} |N|$.

Theorem 2 (Church-Rosser) $\lambda 2$-terms in the style of (D) are Church-Rosser with respect to $\rightarrow_{\beta}$. 

Proof. By the use of parallel reduction.

Remark 3 Note that $\lambda x : B. (\lambda x : A.x) x \rightarrow_{\beta} \lambda x : B.x$ and $\lambda x : B. (\lambda x : A.x) x \rightarrow_{\eta} \lambda x : A.x$. This implies that $\rightarrow_{\beta}$ and $\rightarrow_{\eta}$ are not commutative. Note also that well-typed terms are Church-Rosser w.r.t. $\rightarrow_{\beta}$, from the strong normalization property, weak Church-Rosser, and Newman’s lemma. Another proof is that type-annotated terms in the style of (D) are Church-Rosser together with the subject reduction property.

Proposition 4 (Reductions between type-related problems) 

1. TCP $\hookrightarrow$ TIP: 
$\Gamma \vdash M : A$ iff $\Gamma, z : (A \rightarrow Z) \vdash zM : B$ for some $B$, where $z, Z$ are fresh variables.

2. TIP $\hookrightarrow$ TCP:
$\Gamma \vdash M : B$ for some $B$ iff $\Gamma, z : \forall X. (X \rightarrow Z) \vdash zM : Z$, where $z, Z$ are fresh variables.

3. TIP $\hookrightarrow$ TP: Let $\Gamma = \{x_{1} : A_{1}, \ldots, x_{n} : A_{n}\}$ and $\text{Dom}(\Gamma) = \text{FV}(M)$.
$\Gamma \vdash M : B$ for some $B$ iff $\Sigma \vdash \lambda x_{1} : A_{1} \ldots \lambda x_{n} : A_{n}M : B$ for some $\Sigma, B$.

Definition 6 (Normal forms of (D)-terms)

$$ N \in \text{NF} ::= V | \lambda x : A.N $$
\[ V ::= x | VN \]

Proposition 5 Let $N \in \text{NF}$. If $\Gamma \vdash_{D} M : A$ with the the Elimination-Introduction property, then each application of the rule $(\forall E)$ in the derivation can be restricted to the following form:

$$ \frac{\Gamma \vdash N : \forall X.B}{\Gamma \vdash N : A[X := B]} \ (\forall E') $$

Proof. By induction on the derivation of normal forms, together with the generation lemma.

1. Case of $\Gamma \vdash \lambda x : A.N : B$

From the generation lemma, we have the following derivation:

$$ \frac{\Gamma, x : A \vdash N : C}{\Gamma \vdash \lambda x : A.N : A \rightarrow C \leq_{l(\Gamma)} B} $$

From the induction hypothesis, we have a derivation for $\Gamma, x : A \vdash N : B$, where the derivation may contain only $(\forall E')$ instead of $(\forall E)$.

2. Case of $\Gamma \vdash xN_{1} \ldots N_{n} : B$

From the generation lemma, for some $A_{1}, B_{2}$ we have $x : \Gamma(x) \leq_{E} A_{1} \rightarrow B_{2}$ where $(\forall E')$ may be applied, and $\Gamma \vdash N_{1} : A_{1}$ where each application of $(\forall E)$ can be restricted to $(\forall E')$ by the induction hypothesis. Then for some $A_{2}, B_{3}, N'_{1}$, we have $\Gamma \vdash xN_{1} : B_{2} \leq_{E} xN'_{1} : A_{2} \rightarrow B_{3}$ and $\Gamma \vdash N_{2} : A_{2}$. Here, $xN'_{1} : A_{2} \rightarrow B_{3}$ is obtained from $xN_{1} : B_{2}$ by consecutive application of $(\forall E)$. That is, $B_{2}$ is in the form of $\forall X_{2}B'_{2}$ for some $B'_{2}$, and $X_{2}$ cannot appear in $N_{1}$ as free
type variables. Hence, a chain of applications of \((\forall E)\) can be replaced with \((\forall E')\), so that we have \(xN_1 = xN_1\). In addition, \((\forall E)\) can be restricted to \((\forall E')\) in the derivation of \(\Gamma \vdash N_2 : A_2\) by the induction hypothesis. Following this argument, we have a chain of applications of \((\forall E):\)

\[
\Gamma(x) \leq^E (A_1 \rightarrow B_2), B_2 = \forall \bar{x}_2.B'_2 \leq^E (A_2 \rightarrow B_3), \ldots, B_n = \forall \bar{x}_n.B'_n \leq^E (A_n \rightarrow B_{n+1}),
\]

such that \(x : \Gamma(x) \leq^E x : (A_1 \rightarrow \forall \bar{x}_2.B'_2)\) and \(N_1 : A_1\) where \(\bar{x}_2 \notin \text{FV}(N_1)\),

\[
xN_1 : \forall \bar{x}_2.B'_2 \leq^E xN_1 : (A_2 \rightarrow \forall \bar{x}_3.B'_3)\) and \(N_2 : A_2\) where \(\bar{x}_3 \notin \text{FV}(N_1N_2)\),
\]

\[
\vdots,
\]

\[
xN_1 \ldots N_{n-1} : \forall \bar{x}_n.B'_n \leq^E (A_n \rightarrow \forall \bar{x}_{n+1}.B'_{n+1})\) and \(N_n : A_n\) where \(\bar{x}_{n+1} \notin \text{FV}(N_1 \ldots N_n)\),
\]

and

\[
xN_1 \ldots N_n : \forall \bar{x}_{n+1}.B'_{n+1} \leq^E (\Gamma) xN_1 \ldots N_n : B.
\]

Thus, each application of \((\forall E)\) in the derivation of \(\Gamma \vdash xN_1 \ldots N_n : B\) can be replaced with \((\forall E')\).

We divide the set of type variables into two countable sets: \(\text{TVars}\) for the usual type variables and \(\text{UVars}\) for type variables called unification variable.

\[
\text{TypeVars} = \text{TVars} \cup \text{UVars}
\]

The syntax of output types \(\hat{A}\) of type inference is defined as follows:

\[
\hat{A}, \hat{B} \in \text{Output} ::= X | \alpha | (\hat{A} \rightarrow \hat{B}) | \forall X.\hat{A}
\]

where \(X \in \text{TVars}\) is a type variable, \(\alpha \in \text{UVars}\) is a type variable also called a unification variable.

A unification procedure for the multiset \(E\) of unification equations is defined as usual by the following transformation rules, which give a most general unifier:

1. \(\{\hat{A} \doteq \hat{A}\} \cup E \Rightarrow E\)
2. \(\{\alpha \doteq \hat{A}\} \cup E \Rightarrow \{\alpha \doteq \hat{A}\} \cup E[\alpha := \hat{A}]\) if \(\alpha \notin \text{UVars}(\hat{A})\)
3. \(\{A_1 \rightarrow A_2 \doteq B_1 \rightarrow B_2\} \cup E \Rightarrow \{A_1 \doteq B_1, A_2 \doteq B_2\} \cup E\)
4. \(\{\forall X.\hat{A} \doteq \forall X.\hat{B}\} \cup E \Rightarrow \{\hat{A} \doteq \hat{B}\} \cup E\)

Here, we consider type inference of terms in the style of (D) where a given term is a normal form.

**Definition 7 (Type inference for (D): non-deterministic version for normal case)**

1. \(\text{type}(\Gamma; x) = \Gamma(x)\)
2. \(\text{type}(\Gamma; \lambda x.A.M) = (A \rightarrow B), \text{where type}(\Gamma, x : A; M) \leq^E_{I(\Gamma; A)} B\)
3. \(\text{type}(\Gamma; M_1.M_2) = B_2, \text{where type}(\Gamma; M_1) \leq^E B_1 \rightarrow B_2 \text{ and type}(\Gamma; M_2) \leq^E_{I(\Gamma)} B_1 \text{ for some } B_1\)

As a shorthand, we write \(\bar{x} : \hat{A}\) for \(x_1 : A_1, \ldots, x_n : A_n\), and \(\forall \bar{x}.\hat{A}\) for \(\forall X_1, \ldots X_n.\hat{A}\) \((n \geq 0)\). By deleting \(\forall \bar{X}\) at strictly positive positions, we use the following notation \(\preceq:\)

\[
\forall X_1(A_1 \rightarrow \forall X_2(A_2 \rightarrow \cdots \rightarrow \forall X_n(A_n \rightarrow A) \cdots)) \succeq (A_1 \rightarrow \forall X_2(A_2 \rightarrow \cdots \rightarrow \forall X_n(A_n \rightarrow A) \cdots))
\]

\[
\succeq (A_1 \rightarrow (A_2 \rightarrow \cdots \rightarrow \forall X_n(A_n \rightarrow A) \cdots)) \succeq \cdots \succeq (A_1 \rightarrow (A_2 \rightarrow \cdots \rightarrow (A_n \rightarrow A) \cdots)).
\]

**Definition 8 (Type-inference for (D): deterministic version for normal case)**

1. \(\text{type}(\Gamma; x) = \Gamma(x)\)
2. \(\text{type}(\Gamma; \lambda \bar{x}.\hat{A}V) = (\hat{A} \rightarrow \text{type}(\Gamma; \bar{x} : \hat{A}; V)) \) where \(\bar{x} : \hat{A}\) denotes \(x_1 : A_1, \ldots, x_n : A_n\) \((n \geq 1)\)
3. \( \text{type}(\Gamma; x_{N_{1}} \ldots N_{n}) = A[\bar{X} := \bar{B}] \), where we set \\
\( \Gamma(x) = \forall \bar{X}_{1}(A_{1} \rightarrow \forall \bar{X}_{2}(A_{2} \rightarrow \cdots \rightarrow \forall \bar{X}_{n}(A_{n} \rightarrow A) \cdot \cdot \cdot)) \), \( \bar{X} = \bar{X}_{1} \ldots \bar{X}_{n} \), and \( \bar{B} = \bar{B}_{1} \ldots \bar{B}_{n} \) \\
\((n \geq 1)\)

(a) Case of \( N_{i} = V_{1} \):

There exist some \( \bar{B}_{1} \) such that \( \text{type}(\Gamma; V_{1}) \leq E_{I(\Gamma)} A_{1}[\bar{X}_{1} := \bar{B}_{1}] \).

(b) Case of \( N_{i} = \lambda \bar{y}: \bar{C}.V_{1} \) where \( \bar{C} = C_{1}, \ldots, C_{k} \) (\( k \geq 1 \)):

Let \( \bar{C} \rightarrow \text{type}(\Gamma; \bar{y}: \bar{C}; V_{1}) \) be \( \text{type}(\Gamma; N_{i}) \). There exist some \( \bar{B}_{1}, D_{1} \) such that \( A_{1}[\bar{X}_{1} := \bar{B}_{1}] \geq (C_{1} \rightarrow \cdots \rightarrow C_{k} \rightarrow D_{1}) \) and \( \text{type}(\Gamma; \bar{y}: \bar{C}; V_{1}) \leq E_{I(\Gamma, \bar{C})} D_{1} \).

(c) Case of \( N_{i} = V_{i} \) (\( 1 < i \leq n \)):

There exist some \( \bar{B}_{i} \) such that \( \text{type}(\Gamma; V_{i}) \leq E_{I(\Gamma)} A_{i}[\bar{X}_{i} := \bar{B}_{1} \ldots \bar{B}_{i}] \).

(d) Case of \( N_{i} = \lambda \bar{y}: \bar{C}.V_{i} \) (\( 1 < i \leq n \)) where \( \bar{C} = C_{1}, \ldots, C_{k} \) (\( k \geq 1 \)):

Let \( \bar{C} \rightarrow \text{type}(\Gamma; \bar{y}: \bar{C}; V_{i}) \) be \( \text{type}(\Gamma; N_{i}) \). There exist some \( \bar{B}_{1}, D_{i} \) such that \( A_{i}[\bar{X}_{1} \ldots \bar{X}_{i} := \bar{B}_{1} \ldots \bar{B}_{i}] \geq (C_{1} \rightarrow \cdots \rightarrow C_{k} \rightarrow D_{1}) \) and \( \text{type}(\Gamma; \bar{y}: \bar{C}; V_{i}) \leq E_{I(\Gamma, \bar{C})} D_{i} \).

Remark 4

1. Although the cases of \( N_{i} \) are included in those of \( N_{i} \) (\( i \geq 1 \)), we write the first cases for readability.

2. We use the notation \( \bar{A} \rightarrow \text{type}(\Gamma, \bar{x}: \bar{A}; V) \) for \( \text{type}(\Gamma; \lambda \bar{x}: \bar{A}.V) \). If a given term is in the form of \( \lambda \bar{x}: \bar{A}.V \), then the expression \( \bar{A} \rightarrow \text{type}(\Gamma, \bar{x}: \bar{A}; V) \) simply means that \( A_{1} \rightarrow \cdots \rightarrow A_{n} \rightarrow \text{type}(\Gamma, \bar{x}: \bar{A}; V) \) where \( \bar{A} = (A_{1}, \ldots, A_{n}) \).

Lemma 9

1. It is decidable to verify whether the condition in the case of \( N = V \) of type, i.e., \( \text{type}(\Gamma; N) \leq E_{I(\Gamma)} A[\bar{X} := \bar{B}] \) for some \( \bar{B}, \bar{D} \), holds or not.

2. It is decidable to verify whether the condition in the case of \( N = \lambda \bar{y}: \bar{C}.V \) of type, i.e., \( A[\bar{X} := \bar{B}] \) for some \( \bar{B}, \bar{D} \) such that \( \text{type}(\Gamma; \bar{y}: \bar{C}; V) \leq E_{I(\Gamma, \bar{C})} D \), holds or not.

Proof. 1. The condition that \( \text{type}(\Gamma; N) \leq E_{I(\Gamma)} A[\bar{X} := \bar{B}] \) for some \( \bar{B} \) can be verified by first order unification as follows, see also Lemma 5: Let \( \forall \bar{Y}.C = \text{type}(\Gamma; N) \) (\( C \) has no \( \forall \) as a type-symbol), \( \forall \bar{Z}.A' = A \) (\( A' \) has no \( \forall \) as a type-symbol), and \( \bar{a}, \bar{b} \) be fresh unification variables. Then solve the unification equation such that \( \forall \bar{Y}.C = \bar{a} \). If the unification equation is solvable under a unifier \( S \), then we set \( \bar{B} = S(\bar{a}) \).

2. The condition that \( A[\bar{X} := \bar{B}] \geq (C_{1} \rightarrow \cdots \rightarrow C_{k} \rightarrow D) \) for some \( \bar{B}, \bar{D} \) can be verified by first order unification as follows: Let \( \bar{a}, \bar{b} \) be fresh unification variables, and \( A' \) be obtained from \( A \) by removing \( \forall \bar{X} \) at strictly positive positions just like that \( \forall \bar{X}_{1}(A_{1} \rightarrow \forall \bar{X}_{2}(A_{2} \rightarrow \cdots \rightarrow \forall \bar{X}_{n}(A_{n} \rightarrow A) \cdot \cdot \cdot)) \). Then solve the unification equation such that \( A'[\bar{X} := \bar{B}] = (C_{1} \rightarrow \cdots \rightarrow C_{k} \rightarrow \bar{a}) \). If the unification equation is solvable under a unifier \( S \), then we can check whether \( \text{type}(\Gamma, \bar{x}: \bar{C}; V) \leq E_{I(\Gamma, \bar{C})} S(\bar{a}) \) as in the previous case. Let \( \forall \bar{Y}.E = \text{type}(\Gamma, \bar{x}: \bar{C}; V) \), and \( \bar{a} \) be fresh unification variables. Then solve the unification equation \( E[\bar{a} := \bar{a}] = S(\bar{a}) \). Now suppose that the equation is solvable under a unifier \( T \). Next, we recover \( \forall \bar{X} \) to be removed for \( \geq \) under the variable conditions \( I(\Gamma), I(\Gamma, \bar{C}_{1}), \ldots, I(\Gamma, \bar{C}_{1}, \ldots, \bar{C}_{k-1}) \). Finally, we set \( \bar{B} = T(S(\bar{a})) \) and \( D = T(S(\bar{a})) \).

Proposition 6 (Soundness of type)

If \( \text{type}(\Gamma; N) = A \) then we have \( \Gamma \vdash N : A \).

Proof. The soundness is proved by induction on the length of a term.
1. Case $N$ of $x$:
   We always have $\Gamma \vdash x : \text{type}(\Gamma; x)$.

2. Case $N$ of $\lambda \vec{x} : A.V$:
   From the induction hypothesis, we have $\Gamma, \vec{x} : A \vdash V : \text{type}(\Gamma, \vec{x} : A; V)$. Then $\Gamma \vdash \lambda \vec{x} : A.V : (A \to \text{type}(\Gamma, \vec{x} : A; V))$, and $\text{type}(\Gamma ; \lambda \vec{x} : A.V) = (A \to \text{type}(\Gamma, \vec{x} : A; V))$.

3. Case $N$ of $x N_1 \ldots N_n$:
   Let $\Gamma(x) = \forall \vec{X}_1 (A_1 \to \forall \vec{X}_2 (A_2 \to \cdots \to \forall \vec{X}_n (A_n \to A) \cdot \cdot))$, and $\vec{X} = \vec{X}_1 \ldots \vec{X}_n$.

   (a) Case $N_i$ of $V_i$ ($1 \leq i \leq n$):
   From the induction hypothesis, we have $\Gamma \vdash V_i : \text{type}(\Gamma; V_i)$, and from the assumption, we also have $\Gamma[A_1[\vec{X}_1 \ldots \vec{X}_i := \vec{B}_1 \ldots \vec{B}_i] \succeq (C_1 \to \cdots \to C_k \to D_1)$ and $\text{type}(\Gamma; \vec{y} : \vec{C}; V_i) \leq_{I(\Gamma, \vec{C})}^{E} D_i$ for some $\vec{B}_i, D_i$. Then we have $\Gamma \vdash x N_1 \ldots N_i : \forall \vec{X}_1 (A_1 \to \cdots \to \forall \vec{X}_n (A_n \to A) \cdot \cdot)[\vec{X}_1 \ldots \vec{X}_i := \vec{B}_1 \ldots \vec{B}_i]$. Then we have $\Gamma \vdash x N_1 \ldots N_i : \forall \vec{X}_1 (A_1 \to \cdots \to \forall \vec{X}_n (A_n \to A) \cdot \cdot)[\vec{X}_1 \ldots \vec{X}_i := \vec{B}_1 \ldots \vec{B}_i]$.

   (b) Case $N_i$ of $\lambda \vec{y} : \vec{C}.V_i$ ($1 \leq i \leq n$):
   From the induction hypothesis, we have $\Gamma, \vec{y} : \vec{C} \vdash V_i : \text{type}(\Gamma; V_i)$, and from the assumption, we also have $\Gamma[A_1[\vec{X}_1 \ldots \vec{X}_i := \vec{B}_1 \ldots \vec{B}_i] \succeq (C_1 \to \cdots \to C_k \to D_1)$ and $\text{type}(\Gamma, \vec{y} : \vec{C}; V_i) \leq_{I(\Gamma, \vec{C})}^{E} D_i$ for some $\vec{B}_i, D_i$. Then we have $\Gamma \vdash \forall \vec{Z}_i (C_1 \to \cdots \to C_k \to D_i)$, and moreover $\Gamma \vdash N_i : \forall \vec{Z}_i (C_1 \to \cdots \to \forall \vec{Z}_k (C_k \to D_i))$ under the variable condition, where each $\forall \vec{Z}_i$ is the deleted quantifiers on the condition $A_i[\vec{X}_1 \ldots \vec{X}_i := \vec{B}_1 \ldots \vec{B}_i] \succeq (C_1 \to \cdots \to C_k \to D_i)$. Hence, we have $\Gamma \vdash x N_1 \ldots N_i : \forall \vec{X}_i (A_i \to \cdots \to \forall \vec{X}_n (A_n \to A) \cdot \cdot)[\vec{X}_1 \ldots \vec{X}_i := \vec{B}_1 \ldots \vec{B}_i]$. In this way, we have $\Gamma \vdash x N_1 \ldots N_n : A[\vec{X}_1 \ldots \vec{X}_n := \vec{B}_1 \ldots \vec{B}_n]$ and $\text{type}(\Gamma; x N_1 \ldots N_n) = A[\vec{X}_1 \ldots \vec{X}_n := \vec{B}_1 \ldots \vec{B}_n]$.

Proposition 7 (Completeness of type) Given a context $\Gamma$ and a normal term $N$, let $A$ be a type such that $\Gamma \vdash N : A$. Then we have $\text{type}(\Gamma; V) \leq_{I(\Gamma)}^{E} A$ if $N = V$, and $A \succeq (B_1 \to \cdots \to B_n \to C)$ for some $C$ such that $\text{type}(\Gamma, \vec{x} : \vec{B}; V) \leq_{I(\Gamma, \vec{B})}^{E} C$ if $N = \lambda \vec{x} : \vec{B}.V$.

Proof. The completeness is proved by induction on the derivation with the generation lemma and the Elimination-Introduction property.

1. We have $\Gamma \vdash x : \Gamma(x) \leq_{I(\Gamma)}^{E} \Gamma(x)$.

2. $\Gamma \vdash x N_1 \ldots N_n : A$

   Let $\Gamma(x) = \forall \vec{X}_1 (A_1 \to \forall \vec{X}_2 (A_2 \to \cdots \to \forall \vec{X}_n (A_n \to A) \cdot \cdot))$, and $\vec{X} = \vec{X}_1 \ldots \vec{X}_n$. Then from the generation lemma, we have $\Gamma \vdash x N_1 \ldots N_n : A[\vec{X} := \vec{B}] \leq_{I(\Gamma)}^{E} A$ for some $\vec{B}$, where $\text{type}(\Gamma; x N_1 \ldots N_n) = A[\vec{X} := \vec{B}]$.

3. $\Gamma \vdash \lambda \vec{x} : \vec{C}.V : A$

   From the generation lemma, we have $\Gamma \vdash \lambda \vec{x} : \vec{C}.V : C_1 \to A_1 \leq_{I(\Gamma)}^{E} A$ for some $A_1$, such that $\Gamma, x_1 : C_1 \vdash \lambda \vec{x} : \vec{C}.V : A_1$. Then we also have $\Gamma, x_1 : C_1 \vdash \lambda \vec{x} : \vec{C}.V : C_2 \to A_2 \leq_{I(\Gamma, C_1)}^{E} A_1$ for some $A_2$. Following similar reasoning, we have $\Gamma, x_1 : C_1, \ldots, x_n : C_{n-1} \vdash \lambda \vec{x} : \vec{C}.V : \vec{C} \to \cdots \to \lambda \vec{x} : \vec{C}.V : \vec{C} \to \cdots \to \lambda \vec{x} : \vec{C}.V : \vec{C}$.
For some $A_n$, such that $\Gamma, \vec{x} : \vec{C} \vdash V : A_n$ where $\text{type}(\Gamma, \vec{C}; V) \leq^E_{I(\Gamma, \vec{C})} A_n$ by the induction hypothesis. Now we have the following relations:

$$
C_n \rightarrow A_n \leq^E_{I(\Gamma, C_1, \ldots, C_{n-1})} A_{n-1}
$$

$$
C_2 \rightarrow A_2 \leq^E_{I(\Gamma, C_1)} A_1
$$

$$
C_1 \rightarrow A_1 \leq^E_{I(\Gamma)} A
$$

Namely there are some quantifiers $\forall \vec{X}_i$, such that $A = \forall \vec{X}_1. (C_1 \rightarrow A_1), A_1 = \forall \vec{X}_2. (C_2 \rightarrow A_2), \ldots, A_{n-1} = \forall \vec{X}_n. (C_n \rightarrow A_n)$. Hence, we have $A \geq (C_1 \rightarrow \cdots \rightarrow C_n \rightarrow A_n)$ and $\text{type}(\Gamma, \vec{C}; V) \leq^E_{I(\Gamma, \vec{C})} A_n$.

4. $\Gamma \vdash N : \forall X. A$ from $\Gamma \vdash N : A$ where $X \not\in \text{FV}(\Gamma)$

(a) Case $N$ of $V$:

From the induction hypothesis, we have $\text{type}(\Gamma; V) \leq^E_{I(\Gamma)} A$ and $A \leq^E_{I(\Gamma)} \forall X. A$. Then $\text{type}(\Gamma; V) \leq^E_{I(\Gamma)} \forall X. A$.

(b) Case $N$ of $\lambda \vec{x} : \vec{B}. V$:

From the induction hypothesis, we have $A \geq (B_1 \rightarrow \cdots \rightarrow B_n \rightarrow C)$ for some $C$ such that $\text{type}(\Gamma, \vec{x} : \vec{B}; V) \leq^E_{I(\Gamma, \vec{B})} C$. Then we also have $\forall X. A \geq (B_1 \rightarrow \cdots \rightarrow B_n \rightarrow C)$.

5. $\Gamma \vdash N : A[X := D]$ from $\Gamma \vdash N : \forall X. A$

(a) Case $N$ of $V$:

From the induction hypothesis, we have $\text{type}(\Gamma; V) \leq^E_{I(\Gamma)} \forall X. A$ and $\forall X. A \leq^E_{I(\Gamma)} A[X := D]$. Then $\text{type}(\Gamma; V) \leq^E_{I(\Gamma)} A[X := D]$ from the transitivity.

(b) Case $N$ of $\lambda \vec{x} : \vec{B}. V$:

This case is impossible under the Elimination-Introduction property, since $\Gamma \vdash \lambda \vec{x} : \vec{B}. V : \forall X. A$ should be introduced by $(\forall I)$. $\square$

Next, we define a type inference algorithm in general. For this, the notion of generalization of types is introduced.

**Definition 9 (Generalization)**: Given a type $A$, then define the set of generalization of $A$, denoted by $\text{Gen}(A)$ such that for each $P \in \text{Gen}(A)$, we have $S(P) \equiv A$ for some substitution $S$.

1. $\text{Gen}_\Delta(X) = \{ X^{id} \}$ if $X \not\in \Delta$
2. $\text{Gen}_\Delta(X) = \{ X \}$ if $X \in \Delta$
3. $\text{Gen}_\Delta(A \rightarrow B) = \{ Z^{[Z:=A\rightarrow B]} \}$

$$\cup \{ P_1 \rightarrow P_2 \mid P_1 \in \text{Gen}_\Delta(A_1), P_2 \in \text{Gen}_\Delta(A_2) \} \cup \text{merge}(\text{Gen}_\Delta(A_1), \text{Gen}_\Delta(A_2))$$

where $Z$ is a fresh variable, if $\text{FV}(A \rightarrow B) \not\subseteq \Delta$

4. $\text{Gen}_\Delta(A \rightarrow B) = \{ P_1 \rightarrow P_2 \mid P_1 \in \text{Gen}_\Delta(A_1), P_2 \in \text{Gen}_\Delta(A_2) \} \cup \text{merge}(\text{Gen}_\Delta(A_1), \text{Gen}_\Delta(A_2))$

where $Z$ is a fresh variable, if $\text{FV}(A \rightarrow B) \subseteq \Delta$

5. $\text{Gen}_\Delta(\forall X. A) = \{ Z^{[Z:=\forall X. A]} \} \cup \{ \forall X. P \mid P \in \text{Gen}_{\Delta \cup \{X\}}(A) \}$ where $Z$ is fresh, if $\text{FV}(\forall X. A) \not\subseteq \Delta$

6. $\text{Gen}_\Delta(\forall X. A) = \{ \forall X. P \mid P \in \text{Gen}_{\Delta \cup \{X\}}(A) \}$, if $\text{FV}(\forall X. A) \subseteq \Delta$
7. $\text{merge}(\text{Gen}_\Delta(A), \text{Gen}_\Delta(B)) = \{P_A \to P_B \mid P_A \text{ contains } Z_1^{[Z_1 := C]} \text{ and } P_B \text{ contains } Z_2^{[Z_2 := C]}$ for some $P_1 \in \text{Gen}_\Delta(A)$ and $P_2 \in \text{Gen}_\Delta(B)$, and

- $P_A$ is obtained from $P_1$ by replacing some occurrences of $Z_1^{[Z_1 := C]}$ in $P_1$ with $Z_2^{[Z_2 := C]}$, and

- $P_B$ is obtained from $P_2$ by replacing some occurrences of $Z_2^{[Z_2 := C]}$ in $P_2$ with $Z_1^{[Z_1 := C]}$,

where $Z$ is a fresh variable.

Here, $\Delta$ in $\text{Gen}_\Delta(A)$ denotes the set of bound type-variables in $\text{FV}(A)$, such that for each $X \in \Delta$ we have some context $C \neq [\ ]$ with $\forall X.C[A]$. Given a term $M$, and we write $\text{Atype}(M)$ for the multiset of annotated types in $M$, to say $[A_1, \ldots, A_n]$. Then we have generalizations of each type $[\text{Gen}(A_1), \ldots, \text{Gen}(A_n)]$. Next define the set of terms, denoted by $\text{Gen}(M)$, such that $\text{Gen}(M) = \{M[Z_1, \ldots, Z_n] \mid Z_1 \in \text{Gen}(A_1), \ldots, Z_n \in \text{Gen}(A_n)\}$, where $M[Z_1, \ldots, Z_n]$ is a term obtained from $M$ by replacing each occurrence $A_i$ in $M$ with $Z_i \in \text{Gen}(A_i)$. For each term $N \in \text{Gen}(M)$ we have $S(N) = M$ for some substitution $S$ for type variables in $N$. That is, each term $N \in \text{Gen}(M)$ is a term where annotated types in $M$ are generalized.

We show some examples, where we may omit the identity substitution $\text{id}$.

- $\text{Gen}(X \to Y) = [(X^{\text{id}} \to Y^{\text{id}}), Z^{[Z := (X \to Y)]}]$

- $\text{Gen}((X \to X) \to X \to X) = [(X \to X) \to X \to X, Z_1^{[Z_1 := X \to X]} \to X \to X, (X \to X) \to Z_2^{[Z_2 := X \to X]},$
  $\quad Z_3^{[Z_3 := X \to X]}, Z_1^{[Z_1 := X \to X]} \to Z_2^{[Z_2 := X \to X]}, Z_3^{[Z_3 := X \to X]}]$

- $\text{Gen}(\forall X.(X \to X)) = [\forall X.(X \to X), Z^{[Z := \forall X.(X \to X)]}]$

- $\text{Gen}(\forall X.(X \to Y \to Y) = [\forall X.(X \to Y \to Y), \forall X.(X \to Z^{[Z := Y \to Y]}), Z^{[Z := \forall X.(X \to Y \to Y)]}]$

- Let $B \equiv (\forall X.(X \to X)) \to \forall X.(X \to X)$. $\text{Gen}(B) = [(\forall X.(X \to X)) \to \forall X.(X \to X), Z_1^{[Z_1 := \forall X.(X \to X)]} \to \forall X.(X \to X),$
  $\quad Z_2^{[Z_2 := \forall X.(X \to X)]} \to Z_1^{[Z_1 := \forall X.(X \to X)]}, Z^{[Z := \forall X.(X \to X)]}, \forall X.(X \to X) \to Z_2^{[Z_2 := \forall X.(X \to X)]}, Z_3^{[Z_3 := \forall X.(X \to X)]}]$

Note that $\text{Gen}_\Delta(A)$ is a finite set of types, and then $\text{Gen}_\Delta(M)$ is also a finite set of terms. We always have $A \in \text{Gen}_\Delta(A)$ and $\text{id}(A) = A$, and hence $M \in \text{Gen}_\Delta(M)$.

**Definition 10** (Type inference for Curry with explicit domains: Non-deterministic version)

1. Type($\Gamma; x$) = $\Gamma(x)$

2. Type($\Gamma; \lambda x : A.M$) = ($A \to B$), where Type($\Gamma; x : A; N) \leq_E [\Gamma(A)$, $M : B$ for some $N \in \text{Gen}(M)$

3. Type($\Gamma; M_1 M_2$) = $B_2$, where Type($\Gamma; N_1)$ $\leq_E M_1 : B_1$ to $B_2$ and Type($\Gamma; N_2)$ $\leq_E [\Gamma(M_1), M_2 : B_1$ for some $B_1$ and some $N_1 \in \text{Gen}(M_1), N_2 \in \text{Gen}(M_2)$

**Proposition 8** (Soundness and completeness of non-deterministic Type)

1. If Type($\Gamma; M$) = $A$ then $\Gamma \vdash M : A$.

2. Given a context $\Gamma$ and a term $M$, let $A$ be a type such that $\Gamma \vdash M : A$. Then we have Type($\Gamma; N) \leq_E [\Gamma(M) : A$ for some $N \in \text{Gen}(M)$.
Proof. The soundness is proved by induction on the length of $M$.

1. Type($\Gamma; x) = \Gamma(x)$:
   We have $\Gamma \vdash x : \Gamma(x)$.

2. Type($\Gamma; \lambda x : A.M) = A \rightarrow B$, where Type($\Gamma; x : A; N) \leq^{E}_{I(\Gamma)} M : B$ for some $N \in \text{Gen}(M)$:
   From the induction hypothesis, we have $\Gamma; x : A \vdash N : \text{Type}(\Gamma; x : A; N) \leq^{E}_{I(\Gamma)} M : B$, and then $\Gamma \vdash \lambda x : A.M : (A \rightarrow B) = \text{Type}(\Gamma; x : A; M)$.

3. Type($\Gamma; M_{1}M_{2}) = B_{2}$, where Type($\Gamma; N_{1}) \leq^{E} M_{1} : B_{1} \rightarrow B_{2}$ and Type($\Gamma; N_{2}) \leq^{E}_{I(\Gamma)} M_{2} : B_{1}$ for some $N_{1} \in \text{Gen}(M_{1})$:
   From the induction hypotheses, we have $\Gamma \vdash N_{1} : \text{Type}(\Gamma; N_{1}) \leq^{E} M_{1} : B_{1} \rightarrow B_{2}$ and $\Gamma \vdash N_{2} : \text{Type}(\Gamma; N_{2}) \leq^{E}_{I(\Gamma)} M_{2} : B_{1}$. Then $\Gamma \vdash M_{1}M_{2} : B_{2} = \text{Type}(\Gamma; M_{1}M_{2})$.

The completeness is by induction on derivation.

- Case of $\Gamma \vdash x : \Gamma(x)$:
  We always have Type($\Gamma; x) = \Gamma(x) \leq^{E}_{I(\Gamma)} M(x)$.

- $\Gamma \vdash \lambda x : A.M : A \rightarrow B$ from $\Gamma; x : A \vdash M : B$:
  From the induction hypothesis, we have Type($\Gamma; x : A; N) \leq^{E}_{I(\Gamma,A)} M : B$ for some $N \in \text{Gen}(M)$, and then Type($\Gamma; \lambda x : A.M) = A \rightarrow B$.

- $\Gamma \vdash M_{1}M_{2} : B_{2}$ from $\Gamma \vdash M_{1} : B_{1} \rightarrow B_{2}$ and $\Gamma \vdash M_{2} : B_{1}$:
  From the induction hypotheses, we have Type($\Gamma; N_{1}) \leq^{E}_{I(\Gamma)} M_{1} : B_{1} \rightarrow B_{2}$ and Type($\Gamma; N_{2}) \leq^{E}_{I(\Gamma)} M_{2} : B_{1}$ for some $N_{1} \in \text{Gen}(M_{1})$. Then we have Type($\Gamma; M_{1}M_{2}) = B_{2}$.

- $\Gamma \vdash M : \forall X.A$ from $\Gamma \vdash M : A$ where $X \notin \text{FV}(\Gamma)$:
  From the induction hypothesis, we have Type($\Gamma; N) \leq^{E}_{I(\Gamma)} M : A$ for some $N \in \text{Gen}(M)$, and then $M : A \leq^{E}_{I(\Gamma)} M : \forall X.A$ since $X \notin \text{FV}(\Gamma)$. Hence, we have Type($\Gamma; N) \leq^{E}_{I(\Gamma)} M : \forall X.A$ for some $N \in \text{Gen}(M)$.

- $\Gamma \vdash M[X := B] : A[X := B]$ from $\Gamma \vdash M : \forall X.A$:
  From the induction hypothesis, we have Type($\Gamma; N) \leq^{E}_{I(\Gamma)} M : \forall X.A$ for some $N \in \text{Gen}(M)$. Then we also have $M : \forall X.A \leq^{E} M[X := B] : A[X := B]$, and hence Type($\Gamma; N) \leq^{E}_{I(\Gamma)} M[X := B] : A[X := B]$ for some $N \in \text{Gen}(M)$ from the transitivity.

\[ \square \]

References


