Title
On fine structures between Church-style and Curry-style $\lambda$-terms (Proof theory and complexity)

Author(s)
Fujita, Ken-etsu

Citation
数理解析研究所講究録 (2013), 1832: 73-87

Issue Date
2013-04

URL
http://hdl.handle.net/2433/194853

Type
Departmental Bulletin Paper

Textversion
publisher

Kyoto University
On fine structures between Church-style and Curry-style λ2-terms

Ken-etsu Fujita (Gunma University)

December 14, 2012

Abstract

We introduce a class of 2nd-order λ-terms with fine structures between so called Church-style and Curry-style. Here, λ-terms in the style of Curry are considered as atomic, and we adopt four term-constructors: (i) Domains \( D \) for λ-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 λ2-terms in total, where Church-style can be regarded as a top and Curry-style is a bottom. We examine which term-constructors determine decidability of type-checking and type-inference problems of λ2-terms. This study reveals fine boundaries between decidability and undecidability of the type-related problems.

1 Introduction

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

\[
M ::= x \mid MM \mid \lambda x : A . M \mid M[A] \mid \Lambda X . M
\]

On the other hand, λ-terms in the style of Curry is the same as those of type-free λ-calculus. As a natural combinatorial problem, we can consider λ-terms with fine structures between Curry-style and Church-style. From the viewpoint of components of λ-terms, we take (i) domains of λ-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 λ-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 λ-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 Λ.

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 λ-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 Λ 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 Λ and either [] or [A], or contains neither Λ nor []. Namely, systems of CH, Hole, Df, Tf, (D), and Cu are normal.

We study decision problems parametrized by λ-terms with an intermediate structure of the cubes, and investigate critical conditions for the decidability property from the viewpoint of the constructors (D, Λ, [ ], and [A]). In this paper, as decision problems we adopt the type checking (TCP), type inference (TIP), and typability (TP) problems for second-order λ-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 λ-term $M$, a type $A$, and a context $\Gamma$, determine whether $\Gamma \vdash_s M : A$.

2. Type inference problem of $s$-style λ-terms denoted by TIP(s):
   
   Given an $s$-style λ-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 λ-term $M$, determine whether $\Gamma \vdash_s M : A$ for some context $\Gamma$ and type $A$. 
Table 1: Styles of \(\lambda 2\)-terms and decidability of TCP, TIP, TP

<table>
<thead>
<tr>
<th>Styles</th>
<th>TCP</th>
<th>TIP</th>
<th>TP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Church</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Hole-application</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Domain-free</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Type-free</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Curry</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

Proposition 1 (Reductions between type-related problems)

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

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

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

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

3. Let \(s \in\) UP. 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 \(M_{0} = \pi_{0}(\pi_{1}(z_{1}(\pi_{2}(\forall X.X)))((z_{1}z)(z([A_{1} \rightarrow \cdots \rightarrow A_{n} \rightarrow Y]((\lambda a_{1} \ldots \lambda a_{n} . y) M)))))\), 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 \(\{ x_{1}, \ldots, x_{n} \} = \text{FV}(M)\). \(\Sigma \vdash_{s} M : B\) 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**
  \[ X \in \text{TypeVars} \]
  \[ A \in \text{Types} ::= X \mid (A \rightarrow A) \mid \forall X.A \]

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

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

- **Inference rules:**
  \[ \frac{\Gamma,x:A_{1} \vdash_{D} M:A_{2}}{\Gamma \vdash_{D} \lambda x:A_{1}.M:A_{1} \rightarrow A_{2}} (arrow I) \]
  \[ \frac{\Gamma \vdash_{D} M_{1}:A_{1} \rightarrow A_{2} \Gamma \vdash_{D} M_{2}:A_{1}}{\Gamma \vdash_{D} M_{1}M_{2}:A_{2}} (arrow E) \]
  \[ \frac{\Gamma \vdash_{D} M:A}{\Gamma \vdash_{D} M:\forall X.A} (\forall I)^{\star} \]
  \[ \frac{\Gamma \vdash_{D} M:\forall X.A}{\Gamma \vdash_{D} M[X := A_{1}]:A[X := A_{1}]} (\forall E) \]

  where \((\forall I)^{\star}\) 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. \(\| X \| = X\), \(\| A_{1} \rightarrow A_{2} \| = \| A_{1} \| \rightarrow \| A_{2} \|\),
   \(\| \forall X.A \| = \forall X.\| A \|\) for \(X \notin \text{FV}(A)\),
   \(\| \forall X.A \| = \| A \|\) for \(X \in \text{FV}(A)\),
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. \(\text{FV}(A) = FV(\| A \|)\)

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

**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 \(\| \Gamma' \| = \Gamma\), \(\| M' \| = M\), 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 \(\hat{Z}\):

1. Case \(A = X\): \(B = \forall \hat{Z}.X\),
2. Case \(A = (A_{1} \rightarrow A_{2})\): \(B = \forall \hat{Z}.(A'_{1} \rightarrow A'_{2})\),
3. Case \(A = \forall X.A_{1}\): \(B = \forall \hat{Z}.\forall X.A'_{1}\). 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 \(\| \Gamma' \| = \Gamma\), \(\| M' \| = x\), and \(\| A' \| = \Gamma(x)\), we have \(M' \equiv x\) 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 \( \Gamma \vdash \lambda x : A . M : A \rightarrow B \):

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

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

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

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

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

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

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

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

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)\textit{(\forall E)}-reduction for (D)) Let \( X \not\in \text{FV}(\Gamma) \).

\[\begin{array}{c}
\Gamma \vdash M : A \\
\Gamma \vdash M : \forall X . A \\
\hline
\Gamma \vdash M[X := B] : A[X := B]
\end{array}\]

\((\forall I)^*\)

\[\begin{array}{c}
\Gamma \vdash \forall X . A \\
\hline
\Gamma \vdash M[X := B] : A[X := B]
\end{array}\]

\((\forall E)\)

Under this definition, we consider only derivations without \((\forall I)\textit{(\forall E)}\)-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 \((\text{var})\), \((\rightarrow E)\), or \((\forall E)\).

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

2. \( A \leq_{I(\Gamma)} B \) if \( \Gamma \vdash_D B \) 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. \( A \leq_E B \) if \( \Gamma \vdash_D C \) and \( C \leq_{I(\Gamma)} B \) for some type \( C \).
For instance, $\forall X.(X \rightarrow X) \leq^{E}_{I(\Gamma)} \forall X.\forall Y.((X \rightarrow Z \rightarrow Y) \rightarrow (X \rightarrow Z \rightarrow Y))$ where $X,Y \notin \text{FV}(\Gamma)$.

We also write $\Gamma \vdash_{D} M : A \leq^{E}_{I(\Gamma)} N : B$, if $\Gamma \vdash_{D} M : A$ derives $\Gamma \vdash_{D} N : B$ 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 \& 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.** $(\Rightarrow)$: Suppose $\forall X_{1} \ldots X_{n}.A \leq^{E}_{I(\Gamma)} \forall Y_{1} \ldots Y_{m}.B$. Then $\forall X_{1} \ldots X_{n}.A \leq^{E}_{I(\Gamma)} A[X_{1} := A_{1}, \ldots, X_{n} := A_{n}] = S(A) = B$ for some $S$, since $B \leq^{E}_{I(\Gamma)} \forall Y_{1} \ldots Y_{m}.B$. Hence, $S(A) = B$ for some $S$.

$(\Leftarrow)$: Suppose that $S(A) = B$ for some $S$. Then $\forall X_{1} \ldots X_{n}.A \leq^{E} S(A) = B \leq^{E}_{I(\Gamma)} \forall Y_{1} \ldots Y_{m}.B$ where each $Y_{i} \notin \text{FV}(\Gamma)$. $\square$

**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-\&.

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. $A \leq^{E}_{I(\Gamma)} B$ and $B \leq^{E}_{I(\Gamma)} A$ then $A \equiv B$.

**Proof.** (2) If $\Gamma \vdash_{D} A \leq^{E}_{I(\Gamma)} B$ and $\Gamma \vdash_{D} B \leq^{E}_{I(\Gamma)} C$, then we have $\Gamma \vdash_{D} A \leq^{E}_{I(\Gamma)} C$. Moreover, if $\Gamma \vdash_{D} M_{1} : A \leq^{E}_{I(\Gamma)} M_{2} : B$ and $\Gamma \vdash_{D} M_{2} : B \leq^{E}_{I(\Gamma)} M_{3} : C$, then we have $\Gamma \vdash_{D} M_{1} : A \leq^{E}_{I(\Gamma)} M_{3} : C$.

(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. $\square$

Note that if we have vacuous-\&, then $\forall X.Y.X \leq^{E}_{I} \forall Z.Z$ and $\forall Z.Z \leq^{E}_{I} \forall X.Y.X$, but $\forall X.Y.X \not= \forall Z.Z$.

**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 $\Gamma \vdash \lambda x : A_{1}.M : A_{2}$ is that $\Gamma, x : A_{1} \vdash M : B$ and $A_{1} \rightarrow B \leq^{E}_{I(\Gamma)} A_{2}$ for some $B$. Here, we cannot apply $(\forall E)$ for $A_{1} \rightarrow B \leq^{E}_{I(\Gamma)} A_{2}$.

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 $B_{2} = \forall X.B_{2}'$ for some $B_{2}'$. Then $\forall X$ 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 \(M \rightarrow_\beta N\). \(\Box\)

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 \((\lambda x : A. M)N \rightarrow_\beta M[x := N]\): 
  \[
  \Gamma, x : A' \vdash M' : B_1 \quad \Gamma \vdash \lambda x : A'. M' : A' \rightarrow B_1 \quad \Gamma \vdash N : A' \quad \Gamma \vdash \lambda x : A. M : A \rightarrow B_1 \quad \Gamma \vdash N[x := N] : B_1 \leq_{I(\Gamma)} B \]
  
  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 some substitution \(S\).

  \[
  \Gamma \vdash R' : B_2 \rightarrow B_1 \quad \Gamma \vdash R : B_2 \quad \Gamma \vdash M : B_2 \quad \Gamma \vdash RM : B \quad \Gamma \vdash M : B_2 \leq_{I(\Gamma)} B \]

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

- Case of \(\Gamma \vdash MR : B\) and \(MR \rightarrow NR\) from \(M \rightarrow N\): 
  \[
  \Gamma \vdash M' : B_2 \rightarrow B_1 \quad \Gamma \vdash R : B_2 \quad \Gamma \vdash M' : B_2 \quad \Gamma \vdash MR : B \quad \Gamma \vdash M' : B_2 \leq_{I(\Gamma)} B \]

  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 \(\Gamma \vdash N'R : B_1 \leq_{I(\Gamma)} B\).

- Case of \(\Gamma \vdash \lambda x : A. M : B\) and \(\lambda x : A. M \rightarrow \lambda x : A. N\) from \(M \rightarrow N\): 
  \[
  \Gamma, x : A \vdash A \vdash M : B_1 \quad \Gamma \vdash \lambda x : A. M : A \rightarrow B_1 \leq_{I(\Gamma)} B \quad \Gamma \vdash \lambda x : A. M : B \quad \Gamma \vdash \lambda x : A. N : A \rightarrow B_1 \leq_{I(\Gamma)} B \]

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

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 \(A \leq_{I} A_1\) and \(B_1 \leq_{I} B\). For instance, we have \(x : (A \rightarrow \forall X. X) \vdash \lambda a : A. x : A \rightarrow Z\). 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 $\Gamma \vdash_{Cu} |M| : A$ and the Curry-term $|M|$ is strongly normalizing, where $|.|$ is a forgetful mapping from (D)-terms to Curry-style terms. For (D)-terms $M, N$, if $M \rightarrow_B N$ then $|M| \rightarrow_{\beta} |N|$. \hfill \Box

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

Proof. By the use of parallel reduction. \hfill \Box

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 $\iff$ 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 $\iff$ 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 $\iff$ TP: Let $\Gamma = \{x_1 : A_1, \ldots, x_n : A_n\}$ and Dom($\Gamma$) = 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)

$$N \in \text{NF} ::= V \mid \lambda x : A.N$$

$$V ::= x \mid V \hspace{1cm} N$$

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:

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

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:

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

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 \(xN_{1} \equiv xN_{1}\). 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 \((\forall E)\):
\[
\Gamma(x) \leq^{E} (A_{1} \rightarrow B_{2}), B_{2} = \forall \vec{X}_{2}.B_{2}' \leq^{E} (A_{2} \rightarrow B_{3}), \ldots, B_{n} = \forall \vec{X}_{n}.B_{n}' \leq^{E} (A_{n} \rightarrow B_{n+1}),
\]

such that
\[
x : \Gamma(x) \leq^{E'} x : (A_{1} \rightarrow \forall \vec{X}_{2}.B_{2}') \text{ and } N_{1} : A_{1} \text{ where } \vec{X}_{2} \notin \text{FV}(N_{1}),
\]
\[
xN_{1} : \forall \vec{X}_{2}.B_{2}' \leq^{E} xN_{1} : (A_{2} \rightarrow \forall \vec{X}_{3}.B_{3}') \text{ and } N_{2} : A_{2} \text{ where } \vec{X}_{3} \notin \text{FV}(N_{1}N_{2}),
\]
\[
\ldots,
\]
\[
xN_{1} \ldots N_{n-1} : \forall \vec{X}_{n}.B_{n}' \leq^{E'} \Gamma \vdash xN_{1} \ldots N_{n} : B.
\]

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: TVars for the usual type variables and UVars for type variables called unification variable.

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

1. \(\text{type}(\Gamma; x) = \Gamma(x)\)
2. \(\text{type}(\Gamma; \lambda x : A.M) = (A \rightarrow B)\), where \(\text{type}(\Gamma, x : A; M) \leq^{E} B\) for some \(B\)
3. \(\text{type}(\Gamma; M_{1}M_{2}) = B_{2}\), where \(\text{type}(\Gamma; M_{1}) \leq^{E} B_{1} \rightarrow B_{2}\) and \(\text{type}(\Gamma; M_{2}) \leq^{E} B_{1}\)

As a shorthand, we write \(\vec{x} : \vec{A}\) for \(x_{1} : A_{1} \ldots x_{n} : A_{n}\), and \(\forall \vec{x}.A\) for \(\forall X_{1} \ldots X_{n}.A\) \((n \geq 0)\). By deleting \(\forall \vec{X}\) at strictly positive positions, we use the following notation \(\succeq\):
\[
\forall \vec{X}_{1}(A_{1} \rightarrow \forall \vec{X}_{2}(A_{2} \rightarrow \cdots \rightarrow \forall \vec{X}_{n}(A_{n} \rightarrow A) \cdots)) \succeq (A_{1} \rightarrow \forall \vec{X}_{2}(A_{2} \rightarrow \cdots \rightarrow \forall \vec{X}_{n}(A_{n} \rightarrow A) \cdots))
\]
\[
\succeq \ldots \succeq (A_{1} \rightarrow (A_{2} \rightarrow \cdots \rightarrow \forall \vec{X}_{n}(A_{n} \rightarrow A) \cdots))
\]

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

1. \(\text{type}(\Gamma; x) = \Gamma(x)\)
2. \(\text{type}(\Gamma; \lambda \vec{x} : \vec{A}.V) = (A \rightarrow \text{type}(\Gamma, \vec{x} : \vec{A}; V))\) where \(\vec{x} : \vec{A}\) denotes \(x_{1} : A_{1} \ldots x_{n} : A_{n}\) \((n \geq 1)\)
3. type(Γ; xN₁…Nₙ) = A[\vec{X} := \vec{B}], where we set
\[ \Gamma(x) = \forall \vec{X}_1(A_1 \rightarrow \forall \vec{X}_2(A_2 \rightarrow \cdots \rightarrow \forall \vec{X}_n(A_n \rightarrow A) \cdot \cdot)) \]
\[ \vec{X} = \vec{X}_1 \ldots \vec{X}_n, \text{ and } \vec{B} = \vec{B}_1 \ldots \vec{B}_n \]
\[ (n \geq 1) \]

(a) Case of N₁ = V₁:
There exist some \vec{B}_1 such that type(Γ; V₁) ≤⁴⁺₁(Γ) A₁[\vec{X}_1 := \vec{B}_1].

(b) Case of N₁ = λ\vec{y}: \vec{C}_1V₁ where \vec{C} = C₁, \ldots, Cₖ (k ≥ 1):
Let (\vec{C} \rightarrow type(Γ; y: \vec{C}; V₁)) be type(Γ; N₁). There exist some \vec{B}_1, D₁ such that A₁[\vec{X}_1 := \vec{B}_1] ≥ (C₁ → \cdots → Cₖ → D₁) and type(Γ; y: \vec{C}; V₁) ≤⁴⁺₁(Γ, \vec{C}) D₁.

(c) Case of N₁ = V₁ (1 < i ≤ n):
There exist some \vec{B}_i such that type(Γ; V₁) ≤⁴⁺₁(Γ) Aᵢ[\vec{X}_₁ \ldots \vec{X}_i := \vec{B}_₁ \ldots \vec{B}_i].

(d) Case of N₁ = λ\vec{y}: \vec{C}_iV₁ (1 < i ≤ n) where \vec{C} = C₁, \ldots, Cₖ (k ≥ 1):
Let (\vec{C} \rightarrow type(Γ; y: \vec{C}; V₁)) be type(Γ; N₁). There exist some \vec{B}_i, Dᵢ such that Aᵢ[\vec{X}_₁ \ldots \vec{X}_i := \vec{B}_₁ \ldots \vec{B}_i] ≥ (C₁ → \cdots → Cₖ → Dᵢ) and type(Γ; y: \vec{C}; V₁) ≤⁴⁺₁(Γ, \vec{C}) Dᵢ.

Remark 4
1. Although the cases of N₁ are included in those of Nₙ (i ≥ 1), we write the first cases for readability.
2. We use the notation \vec{A} → type(Γ, \vec{x}: \vec{A}; V) for type(Γ, λ\vec{x}: \vec{A}; V). If a given term is in the form of \lambda\vec{x}: \vec{A}; V, then the expression A → type(Γ, \vec{x}: \vec{A}; V) simply means that A₁ → \cdots → Aₙ → type(Γ, \vec{x}: \vec{A}; V) where \vec{A} = (A₁, \ldots, Aₙ).

Lemma 9
1. It is decidable to verify whether the condition in the case of N = V of type, i.e., type(Γ; N₁) ≤⁴⁺₁(Γ) A[\vec{X} := \vec{B}] for some \vec{B}, holds or not.
2. It is decidable to verify whether the condition in the case of N = \lambda\vec{y}: \vec{C}V₁ of type, i.e., A[\vec{X} := \vec{B}] ≥ (C₁ → \cdots → Cₖ → D) for some \vec{B}, D such that type(Γ; \vec{y}: \vec{C}; V) ≤⁴⁺₁(Γ, \vec{C}) D, holds or not.

Proof. 1. The condition that type(Γ; N₁) ≤⁴⁺₁(Γ) A[\vec{X} := \vec{B}] for some \vec{B} can be verified by first order unification as follows, see also Lemma 5: Let \forall \vec{Y}.C = type(Γ; N₁) (C has no \forall as a top-symbol), \forall \vec{Z}.A' = A (A' has no \forall as a top-symbol), and \vec{a}, \vec{b} be fresh unification variables. Then solve the unification equation such that C[\vec{Y} := \vec{b}] = A'[\vec{X} := \vec{a}]. If the unification equation is solvable under a unifier S, then we set \vec{B} = S(\vec{a}).

2. The condition that A[\vec{X} := \vec{B}] ≥ (C₁ → \cdots → Cₖ → D) for some \vec{B}, D can be verified by first order unification, as follows: Let \vec{a}, \vec{b}, \vec{c} be fresh unification variables, and A' be obtained from A by removing \forall \vec{X} at strictly positive positions just like that \forall \vec{X}_₁(A₁ → \forall \vec{X}_₂(A₂ → \cdots → \forall \vec{X}_n(A_n → A) \cdot \cdot)) ≥ (A₁ → (A₂ → \cdots → (Aₙ → A) \cdot \cdot)). Then solve the unification equation such that A'[\vec{X} := \vec{a}] = (C₁ → \cdots → Cₖ → \vec{c}). If the unification equation is solvable under a unifier S, then we can check whether type(Γ, \vec{x}: \vec{C}; V) ≤⁴⁺₁(Γ, \vec{C}) S(\vec{c}) as in the previous case. Let \forall \vec{Y}.E be type(Γ, \vec{x}: \vec{C}; V), and \vec{c} be fresh unification variables. Then solve the unification equation E[\vec{Y} := \vec{c}] = S(\vec{c}). Now suppose that the equation is solvable under a unifier T. Next, we recover \forall \vec{X} to be removed for ≥ under the variable conditions I(Γ), I(Γ, \vec{C}_1), \ldots, I(Γ, \vec{C}_₁, \ldots, \vec{C}_{k-1}). Finally, we set \vec{B} = T(S(\vec{c})) and D = T(S(\vec{c})).

Proposition 6 (Soundness of type) If type(Γ; N) = A then we have Γ ⊨ 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 $\Gamma \vdash \lambda \vec{x} : A.V : (A \rightarrow \text{type}(\Gamma, \vec{x} : A; V))$, and type$(\Gamma; \lambda \vec{x} : A.V) = (A \rightarrow \text{type}(\Gamma, \vec{x} : A; V))$.

3. Case $N$ of $x_{N_{1}} \ldots N_{n}$:
   Let $\Gamma(x) = \forall \vec{x}_{1}(A_{1} \rightarrow \forall \vec{x}_{2}(A_{2} \rightarrow \cdots \rightarrow \forall \vec{x}_{n}(A_{n} \rightarrow A))$, 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 $A_{N_{i}}[\vec{x}_{1} \ldots \vec{x}_{i} := B_{1} \ldots B_{i}] \succeq (C_{1} \rightarrow \cdots \rightarrow C_{k} \rightarrow D_{i})$ and type$(\Gamma; V_{i}) \leq_{I(\Gamma)} E A_{i}$ for some $B_{i}, D_{i}$. Then from the induction hypothesis, we have $\Gamma \vdash x_{N_{i}} \ldots N_{n} : A_{N_{i}}[\vec{x}_{1} \ldots \vec{x}_{i} := B_{1} \ldots B_{i}] \succeq (C_{1} \rightarrow \cdots \rightarrow C_{k} \rightarrow D_{i})$. Here, we have $\forall \vec{Z}_{i}(C_{1} \rightarrow \cdots \rightarrow \forall \vec{Z}_{k}(C_{k} \rightarrow D_{i})) = A_{N_{i}}[\vec{x}_{1} \ldots \vec{x}_{i} := B_{1} \ldots B_{i}]$. Hence, we have $\Gamma \vdash x_{N_{i}} \ldots N_{n} : \forall \vec{x}_{1}(A_{i} \rightarrow \cdots \rightarrow \forall \vec{x}_{n}(A_{n} \rightarrow A)) \succeq (B_{1} \ldots B_{i})$.
      In this way, we have $\Gamma \vdash x_{N_{1}} \ldots N_{n} : A[\vec{x}_{1} \ldots \vec{x}_{n} := B_{1} \ldots B_{n}]$ and type$(\Gamma; x_{N_{1}} \ldots N_{n}) = A[\vec{x}_{1} \ldots \vec{x}_{n} := B_{1} \ldots B_{n}]$. 

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 type$(\Gamma; V) \leq_{E} A$ if $N = V$, and $A \succeq (B_{1} \rightarrow \cdots \rightarrow B_{n} \rightarrow C)$ for some $C$ such that type$(\Gamma, \vec{x} : \vec{C}; V) \leq_{E} C$ if $N = \lambda \vec{x} : \vec{C} ; V$.

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

1. We have $\Gamma \vdash x : \text{type}(\Gamma; x)$.
2. $\Gamma \vdash x_{N_{1}} \ldots N_{n} : A$
   Let $\Gamma(x) = \forall \vec{x}_{1}(A_{1} \rightarrow \forall \vec{x}_{2}(A_{2} \rightarrow \cdots \rightarrow \forall \vec{x}_{n}(A_{n} \rightarrow A) \cdot \cdot))$, and $\vec{x} = \vec{x}_{1} \ldots \vec{x}_{n}$. Then from the generation lemma, we have $\Gamma \vdash x_{N_{1}} \ldots N_{n} : A_{0}[\vec{x} := B] \leq_{I(\Gamma)} A$ for some $B$, where type$(\Gamma; x_{N_{1}} \ldots N_{n}) = A_{0}[\vec{x} := B]$.
3. $\Gamma \vdash \lambda \vec{x} : \vec{C} . V : A$
   From the generation lemma, we have $\Gamma \vdash \lambda \vec{x} : \vec{C} . V : C_{1} \rightarrow A_{1} \leq_{I(\Gamma)} A$ for some $A_{1}$, such that $\Gamma, x_{1} : C_{1} \vdash \lambda \vec{x} : \vec{C} . V : A_{1}$. Then we also have $\Gamma, x_{1} : C_{1} \vdash \lambda \vec{x} : \vec{C} . V : C_{2} \rightarrow A_{2} \leq_{I(\Gamma, C_{1})} A_{1}$ for some $A_{2}$. Following similar reasoning, we have $\Gamma, x_{1} : C_{1}, \ldots, x_{n} : C_{n} \vdash \lambda x_{n} : C_{n} . V : \vec{C} \rightarrow$
$A_n \leq I(\Gamma, C_1, \ldots, C_{n-1}) A_{n-1}$ 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:

$$
C_n \rightarrow A_n \leq I(\Gamma, C_1, \ldots, C_{n-1}) A_{n-1} \\
C_2 \rightarrow A_2 \leq I(\Gamma, C_1) A_1 \\
C_1 \rightarrow A_1 \leq I(\Gamma) A
$$

Namely there are some quantifiers $\forall \vec{X}_i$, such that $A = \forall \vec{X}_1. (C_1 \rightarrow A_1)$, $A_1 = \forall \vec{X}_2. (C_2 \rightarrow A_2)$, $\ldots$, and $A_{n-1} = \forall \vec{X}_n. (C_n \rightarrow A_n)$. Hence, we have $A \succeq (C_1 \rightarrow \cdots \rightarrow C_n \rightarrow A_n)$ 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 $X \notin \text{FV}(\Gamma)$

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

5. $\Gamma \vdash N : [X := D]$ 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 $\forall X.A \leq E_{I(\Gamma)} [X := D]$. Then $\text{type}(\Gamma; V) \leq E_{I(\Gamma)} [X := D]$ 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 $(\forall I)$. □

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 $X \notin \Delta$
2. $\text{Gen}_\Delta(X) = \{X\}$ if $X \in \Delta$
3. $\text{Gen}_\Delta(A \rightarrow B) = \{Z^{[Z := A \rightarrow B]}\}$
   \[ \cup \{P_1 \rightarrow P_2 | P_1 \in \text{Gen}_\Delta(A_1), P_2 \in \text{Gen}_\Delta(A_2)\} \] 
   $\cup \text{merge}(\text{Gen}_\Delta(A_1), \text{Gen}_\Delta(A_2))$
   where $Z$ is a fresh variable, if $\text{FV}(A \rightarrow B) \not\subseteq \Delta$
4. $\text{Gen}_\Delta(A \rightarrow B) = \{P_1 \rightarrow P_2 | P_1 \in \text{Gen}_\Delta(A_1), P_2 \in \text{Gen}_\Delta(A_2)\} \cup \text{merge}(\text{Gen}_\Delta(A_1), \text{Gen}_\Delta(A_2))$
   where $Z$ is a fresh variable, if $\text{FV}(A \rightarrow B) \subseteq \Delta$
5. $\text{Gen}_\Delta(\forall X.A) = \{Z^{[Z := \forall X.A]}\} \cup \{\forall X.P | P \in \text{Gen}_{\Delta \cup \{X\}}(A)\}$ where $Z$ is fresh, if $\text{FV}(\forall X.A) \not\subseteq \Delta$
6. $\text{Gen}_\Delta(\forall X.A) = \{\forall X.P | P \in \text{Gen}_{\Delta \cup \{X\}}(A)\}$, if $\text{FV}(\forall X.A) \subseteq \Delta$
7. $\text{merge}(\text{Gen}_{\Delta}(A), \text{Gen}_{\Delta}(B)) = \{ P_A \rightarrow P_B \mid P_A \text{ contains } Z_1^{[Z_1=:\!C]} \text{ and } P_B \text{ contains } Z_2^{[Z_2=:\!C]} \}
   \text{ for some } P_1 \in \text{Gen}_{\Delta}(A) \text{ and } P_2 \in \text{Gen}_{\Delta}(B), \text{ and}
   \text{P_A is obtained from } P_1 \text{ by replacing some occurrences of } Z_1^{[Z_1=:\!C]} \text{ in } P_1 \text{ with } Z_2^{[Z_2=:\!C]}, \text{ and}
   \text{P_B is obtained from } P_2 \text{ by replacing some occurrences of } Z_2^{[Z_2=:\!C]} \text{ in } P_2 \text{ with } Z_1^{[Z_1=:\!C]},
   \text{ where } Z \text{ 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 $C \neq \emptyset$ with $\forall X.C[A]$.

Given a term $M$, and we write $\text{type}(M)$ 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 id.

- $\text{Gen}(X \rightarrow Y) = [([X^{\text{id}} \rightarrow Y^{\text{id}}], Z^{[Z=:\!(X \rightarrow Y)]}]
- \text{Gen}((X \rightarrow X) \rightarrow X \rightarrow X) = [(X \rightarrow X) \rightarrow X \rightarrow X, Z_1^{[Z_1=:\!(X \rightarrow X)]} \rightarrow X \rightarrow X, (X \rightarrow X) \rightarrow Z_2^{[Z_2=:\!(X \rightarrow X)], Z_1^{[Z_1=:\!(X \rightarrow X)]} \rightarrow Z_2^{[Z_2=:\!(X \rightarrow X)], Z_3^{[Z_3=:\!(X \rightarrow X \rightarrow X)]}}]
- \text{Gen}(\forall X.(X \rightarrow X)) = [\forall X.(X \rightarrow X), Z^{[Z=:\!(\forall X.(X \rightarrow X))]}]$
- $\text{Gen}(\forall X.(X \rightarrow Y \rightarrow Y) = [\forall X.(X \rightarrow Y \rightarrow Y), \forall X.(X \rightarrow Z^{[Z=:\!(Y \rightarrow Y)]}), Z^{[Z=:\!(\forall X.(Y \rightarrow Y))]}]
- \text{Let } B \equiv (\forall X.(X \rightarrow X)) \rightarrow \forall X.(X \rightarrow X).
   \text{Gen}(B) = [(\forall X.(X \rightarrow X)) \rightarrow \forall X.(X \rightarrow X), Z_1^{[Z_1=:\!(\forall X.(X \rightarrow X))]} \rightarrow \forall X.(X \rightarrow X), Z_2^{[Z_2=:\!(\forall X.(X \rightarrow X))]} \rightarrow \forall X.(X \rightarrow X), Z_3^{[Z_3=:\!(\forall X.(X \rightarrow X)) \rightarrow \forall X.(X \rightarrow 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. $\text{Type}(\Gamma; x) = \Gamma(x)$
2. $\text{Type}(\Gamma; \lambda x:A.M) = (A \rightarrow B)$, where $\text{Type}(\Gamma; x:A; N) \leq_{E_{\Gamma}} B$ for some $N \in \text{Gen}(M)$
3. $\text{Type}(\Gamma; M_1 M_2) = B_2$, where $\text{Type}(\Gamma; N_1) \leq_{E} M_1 : B_1 \rightarrow B_2$ and $\text{Type}(\Gamma; N_2) \leq_{E} M_2 : B_1$ 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 $\text{Type}(\Gamma; M) = A$ 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 $\text{Type}(\Gamma; N) \leq_{E_{\Gamma}} M : A$ for some $N \in \text{Gen}(M)$. 

Proof. The soundness is proved by induction on the length of \( M \).

1. \( \text{Type}(\Gamma; x) = \Gamma(x) \):
   
   We have \( \Gamma \vdash x : \Gamma(x) \).

2. \( \text{Type}(\Gamma; \lambda x : A . M) = A \to B \), where \( \text{Type}(\Gamma; x : A ; N) \leq E_{I(\Gamma)} M : B \) 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 \( \Gamma \vdash \lambda x : A . M : (A \to B) = \text{Type}(\Gamma; x : A ; M) \).

3. \( \text{Type}(\Gamma; M_1 M_2) = B_2 \), where \( \text{Type}(\Gamma; N_1) \leq E_{I(\Gamma)} M_1 : B_1 \to B_2 \) and \( \text{Type}(\Gamma; N_2) \leq E_{I(\Gamma)} M_2 : B_1 \) for some \( N_i \in \text{Gen}(M_i) \):
   
   From the induction hypotheses, we have \( \Gamma \vdash N_1 : \text{Type}(\Gamma; N_1) \leq E_{I(\Gamma)} M_1 : B_1 \to B_2 \) 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 \( \text{Type}(\Gamma; x) = \Gamma(x) \leq E_{I(\Gamma)} \Gamma(x) \).

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

- \( \Gamma \vdash M_1 M_2 : B_2 \) from \( \Gamma \vdash M_1 : B_1 \to B_2 \) and \( \Gamma \vdash M_2 : B_1 \):
  
  From the induction hypotheses, we have \( \text{Type}(\Gamma; N_1) \leq E_{I(\Gamma)} M_1 : B_1 \to B_2 \) and \( \text{Type}(\Gamma; N_2) \leq E_{I(\Gamma)} M_2 : B_1 \) for some \( N_i \in \text{Gen}(M_i) \). Then we have \( \text{Type}(\Gamma; M_1 M_2) = B_2 \).

- \( \Gamma \vdash M : \forall X . A \) from \( \Gamma \vdash M : A \) where \( X \not\in \text{FV}(\Gamma) \):
  
  From the induction hypothesis, we have \( \text{Type}(\Gamma; N) \leq E_{I(\Gamma)} M : A \) for some \( N \in \text{Gen}(M) \), and then \( M : A \leq I(\Gamma) M : \forall X . A \) since \( X \not\in \text{FV}(\Gamma) \). Hence, we have \( \text{Type}(\Gamma; N) \leq E_{I(\Gamma)} M : \forall X . A \) 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 \( \text{Type}(\Gamma; N) \leq E_{I(\Gamma)} M : \forall X . A \) for some \( N \in \text{Gen}(M) \). Then we also have \( M : \forall X . A \leq E_{I(\Gamma)} M[X := B] : A[X := B] \), and hence \( \text{Type}(\Gamma; N) \leq E_{I(\Gamma)} M[X := B] : A[X := B] \) for some \( N \in \text{Gen}(M) \) from the transitivity.

\[ \square \]

References


