A Rewriting Formulation of Curry-Style System F

Peng Fu[1] **and Aaron Stump**[1]

1 **Department of Computer Science, The University of Iowa**
14 Maclean Hall, Iowa City, USA
`peng-fu@uiowa.edu, aaron-stump@uiowa.edu`

Abstract

This paper show how to extend the rewriting formulation for type systems due to Kuan et al. to Curry-style system F. The extension utilizes a new technique, namely the use of typing contexts as part of the syntax of terms. We show the equivalence between the new rewriting formulation and Curry-style system F, and give direct proofs of the type preservation and strong normalization theorems for the rewriting formulation.

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

Category Regular Research Paper

1 Introduction

The idea of using rewriting to simulate typing was proposed by Kuan, MacQueen and Findler [6]. Instead of jugding if a term has a type by the typing rules, we can rewrite a term to its types by applying the rewrite rules. The traditional presentation of type system is *relational* in the sense that it tries to decide if the relation $\Gamma \vdash t : T$ holds. While the rewriting-based presentation is *functional* in the sense that it tries to compute a type giving the typing context and the term, namely, $(\Gamma t) \overset{*}{\rightsquigarrow} T$, we will explain the meaning of Γt later.

Kuan provides a rewriting presentation for Chuch-style system F, which is a direct extension of his version for simply typed lambda calculus [5]. His version use the typing context Γ as an instance of substitution, i.e. Γt means substituting all the free term variables in t with their types according to the Γ. Thus Kuan's rewriting formulation does not maintain the typing context environment during the process of rewriting. This makes it hard to simulate the curry-style typing rule below:

$$\frac{\Gamma \vdash t : T \quad X \notin FV(\Gamma)}{\Gamma \vdash t : \forall X.T}\ \forall\textit{-intro}$$

The $\forall$-intro rule above imposes a restriction $X \notin FV(\Gamma)$ which involves checking all the free type variables in the typing context. As mentioned in Stump et al.'s paper [7], the eager use of typing context as substitution also poses particular difficulties when we try to give a rewriting formulation for dependent type systems. For this reason, we develop a new rewriting formulation for Curry-style system F that contains typing context as part of the syntax of the rewriting obejcts, more specificly, Γt will be a syntactical objects in our formulation, Γ does not act as a substitution. We prove our rewriting formulation is equivalent to the traditional presentation of Curry-style system F.

It is possible to apply the rewriting techniques to prove properties about the type systems within the framework of rewriting formulation. Stump et al. [7] show how to use decreasing diagram to get the type preservation results. However, their method requires the rewrite rules that simulate the typing to be confluence. We will see this property does not hold for Curry-style type systems in general. However, we still manage to show type preservation and strong normalization in this paper. The major proof method we used to show metatheorems is by induction on the structure of the term t, which corresponds to the proof by induction on typing derivation for traditional presentation of type system.

In this paper, we first discuss the difficulties of adapting Kuan et al.'s version to Curry style system F(section 2.2). Then we give a new set of rewrite rules to simulate Curry style system F(section 2.3). The difference between our version and Kuan et al.'s is we build the typing context into the syntax of the rewriting objects, instead of treating it as an instance of substitution. We show the equivalence between our rewriting formulation and the standard presentation of Curry-style system F by proving the so call soundness and completeness theorems(section 2.5). We show how to prove type preservation (section 3.1) and strong normalization (section 3.2) in our framework. Several related works(including what we had mentioned so far) and future works are discussed in section 4.

2 Curry-Style System F

Standard presentation of Curry-style typing systems can be found in Barendregt's book [2]. We present a rewriting formulation for Curry-style simply typed lambda calculus(STLC) first. Then we show how the difficulties arise when we try to extend this formulation to Curry-style system F. Finally, a new rewriting formulation is proposed and its important properties are stated and proved.

2.1 Simply Typed Lambda Calculus

▶ **Definition 2.1.** The syntax of STLC is defined as follows:

$$\begin{array}{llll} \textit{types} & T & ::= & X \mid T \to T \\ \textit{terms} & t & ::= & x \mid t\ t \mid \lambda x.t \\ \textit{typing contexts} & \Gamma & ::= & \cdot \mid \Gamma, x : T \end{array}$$

▶ **Definition 2.2.** The relational presentation of STLC is defined as follow:

$$\frac{(x:T) \in \Gamma}{\Gamma \vdash x : T}\ \textit{Var} \qquad \frac{\Gamma, x : T_1 \ \vdash t : T_2}{\Gamma \vdash \lambda x.t : T_1 \to T_2}\ \to\textit{-intro}$$

$$\frac{\Gamma \vdash t_1 : T_1 \to T_2 \qquad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t_1\ t_2 : T_2}\ \to\textit{-elim}$$

▶ **Definition 2.3.** Here is the definition of mixed terms and reduction context.

$$\begin{array}{l} \textit{mixed terms } m \ ::= \ X \mid x \mid mm \mid \lambda x.m \mid T \to m \\ \textit{reduction context } E \ ::= \ * \mid \lambda x.E \mid Em \mid mE \mid T \to E \mid \forall X.E. \end{array}$$

Mixed terms will be the rewriting objects. We can see that mixed terms include terms and types of STLC. Let Γt means the mixed term obtained by substituting all the free term variables in t with their corresponding types in Γ. The process of typing now becomes starting from Γt, keep applying rewrite rules on the mixed terms till we get a type T.

▶ **Definition 2.4.** Now we give the rewriting formulation for STLC–$\lambda^{\rightsquigarrow}$:

$$\begin{array}{l} E[\lambda x.m] \rightsquigarrow_{\lambda} E[T \to [T/x]m] \\ E[(T \to m)T] \rightsquigarrow_{\epsilon} E[m] \end{array}$$

This formulation is an variant of Kuan's Church version. Notice that $\lambda^{\leadsto}$ is infinitely branching due to the rule $\leadsto_\lambda$. We can have $[T/y](\lambda x.xy) \equiv (\lambda x.xT) \leadsto_\lambda T \to (TT)$, which is a stuck term in $\lambda^{\leadsto}$. This is the reason why $\lambda^{\leadsto}$ is not confluence. We use $\leadsto$ as a short hand for $\leadsto_\lambda \cup \leadsto_\epsilon$. And $\overset{*}{\leadsto}$ means the reflective and transitive closure of $\leadsto$.

Now the jugdement $\cdot \vdash \lambda x.x : Y \to Y$ becomes a rewriting process: $\lambda x.x \leadsto_\lambda Y \to [Y/x]x \equiv Y \to Y$. We state the following proposition without proving it.

▶ **Proposition 2.5.** $\Gamma \vdash t : T$ *iff* $\Gamma t \overset{*}{\leadsto} T$.

The left-to-right of this proposition is called soundness theorem, while the right-to-left is called completeness theorem. This proposition is used to establish the equivalence of $\lambda^{\leadsto}$ and STLC.

2.2 Curry-Style System F

The syntax of Curry-style system F(CSSF) is almost the same as STLC, except we need to add $\forall X.T$ to the definition of types. The typing rules for CSSF are the ones in STLC together with two extra rules in definition 2.6.

▶ **Definition 2.6.** we add the following rules to STLC.

$$\frac{\Gamma \vdash t : T \quad X \notin FV(\Gamma)}{\Gamma \vdash t : \forall X.T}\ \forall\text{-}intro \qquad \frac{\Gamma \vdash t : \forall X.T}{\Gamma \vdash t : [T'/X]T}\ \forall\text{-}elim$$

We expand the notion of mixed term by adding $\forall X.m$ to the definition 2.3 and add $\forall X.E$ to the reduction context accordingly. Intuitively, we would want to add the following two rewrite rules to get a rewriting simulation of CSSF:

$$E[m] \leadsto_\pi E[\forall X.m]$$
$$E[\forall X.m] \leadsto_\iota E[[T/X]m]$$

However, this is not a complete simulation because that will allow reductions like: $\lambda x.x \leadsto_\lambda Y \to Y \leadsto_\pi Y \to \forall Y.Y$. But we known that $\cdot \nvdash \lambda x.x : Y \to \forall Y.Y$ in CSSF. We want a rewriting simulation such that: $\Gamma \vdash t : T$ iff $\Gamma t \overset{*}{\leadsto} T$.

One would want to change the $\leadsto_\pi$ to something likes $E[m] \leadsto_\pi E[\forall X.m], X \notin FV(E)$. Here $FV(E)$ means the set of free type variables in the reduction context. This will give us the reduction $\lambda x.x \leadsto_\lambda Y \to Y \not\leadsto_\pi Y \to \forall Y.Y$. It seems to be a possible fix.

But under careful inspection, we can see that first, the reduction is not *compatible with the reduction context*, i.e. if we have $Y \leadsto_\pi \forall Y.Y$, then we do not get $Y \to Y \leadsto_\pi Y \to \forall Y.Y$. This is undesirable since we actually need compatible with reduction context to prove the soundness theorem, namely, if $\Gamma \vdash t : T$, then $\Gamma t \overset{*}{\leadsto} T$. Second, we are still allowing this kind of reduction $[x : Y]x \equiv Y \leadsto_\pi \forall Y.Y$. But we know $x : Y \nvdash x : \forall Y.Y$.

At this point we can see that the problem is due to the eager use of typing context as substitution, e.g. Γt as a result of substituting all the free term variables in t to their types according to Γ. The rule $\lambda x.m \leadsto_\lambda T \to [T/x]m$ build into the idea of using typing context as substitution as well. We notice that this eager use of typing context as substitution is because the definition of mixed terms contain no information about the typing context.

This leads us to consider an alternative definition of mixed terms such that it contains information about the typing context. It turns out that our new formulation provides a satisfiable solution to the problems we mentioned before.

2.3 A New Rewriting Formulation

▶ **Definition 2.7.** Now we define *pretypes* instead of mixed terms.

$$\begin{array}{l}\textit{pretypes } P \ ::= X \mid \Gamma X \mid \Gamma x \mid PP \mid \lambda x.P \mid T \to P \mid \forall X.P. \\ \textit{reduction contexts } E \ ::= \ [] \mid E\ P \mid P\ E \mid \lambda x.E \mid \forall X.E \mid T \to E\end{array}$$

We can see that the structure of the pretypes is similar to the mixed terms in definition 2.3. Types are included in pretypes. The main difference between pretypes and mix terms is that we include two special objects, namely, Γx and ΓX. Notice from now on, we will not treat typing context Γ as instance of substitution like before. So intuitively, we can veiw Γx as a pair (Γ, x), ΓX as (Γ, X). Since pretypes now contain typing context information, we can see it will be able to deal with the problems we mentioned in section 2.2.

Notice that we want to distinguish X from $[\cdot]X$, and x from $[\cdot]x$. They belong to different syntactical category.

▶ **Definition 2.8.** Now we will define rewrite rules on pretypes.

$E[(\lambda x.P)] \leadsto_\lambda E[T \to [x : T] \cdot P]$.
$E[(T \to P)T] \leadsto_\epsilon E[P]$.
$E[P] \leadsto_\pi E[\forall X.P]$, where $X \notin \Gamma V(P)$ and $P \notin$ **Types**.
$E[\forall X.P] \leadsto_\iota E[[T/X]P]$.
$E[\Gamma x] \leadsto_s E[\Gamma T]$, where $(x : T) \in \Gamma$.
$E[\Gamma T] \leadsto_r E[T]$.

We can see that in the rule $\leadsto_\lambda$, instead of use $[x : T]$ as substitution, we use it as a piece of typing context. $[x : T] \cdot P$ means adding the typing context $[x : T]$ to every typing contexts in P, we call it *contextual action*(definition 2.9).

▶ **Definition 2.9.** This is the definition of contextual action.

$[x : T] \cdot X \equiv X$.
$[x : T] \cdot \Gamma y \equiv [\Gamma, x : T]y$ if $x \notin dom(\Gamma)$. Else if $x \in dom(\Gamma)$, then $[x : T] \cdot \Gamma y \equiv \Gamma y$.
$[x : T] \cdot \Gamma Y \equiv [\Gamma, x : T]Y$ if $x \notin dom(\Gamma)$. Else if $x \in dom(\Gamma)$, then $[x : T] \cdot \Gamma Y \equiv \Gamma Y$.
$[x : T] \cdot (P_1P_2) \equiv ([x : T] \cdot P_1)([x : T] \cdot P_2)$
$[x : T] \cdot (T' \to P) \equiv T' \to [x : T] \cdot P$
$[x : T] \cdot \lambda y.P \equiv \lambda y.([x : T] \cdot P)$. This will invoke renaming if nessesary.
$[x : T] \cdot \forall X.P \equiv \forall X.([x : T] \cdot P)$. This will invoke renaming if nessesary.

Now let us look at the rule $\leadsto_\pi$. We do add a restriction for this rule, but we know that this restriction is not about the reduction context. We simply require P is not a type and the type variable X we are going to generalize is not in the set of free type variables in the typing context of P–$\Gamma V(P)$(definition 2.10). So we can get compatibility with the reduction context. We can see that since we have a notion of typing context in our pretypes, we have the luxury to refer to that in this rule.

▶ **Definition 2.10.** We define $\Gamma V(P)$ inductively:

$\Gamma V(X) = \emptyset$.
$\Gamma V(\Gamma x) = FV(\Gamma)$.
$\Gamma V(\Gamma X) = FV(\Gamma)$.
$\Gamma V(P_1 P_2) = \Gamma V(P_2) \cup \Gamma V(P_1)$
$\Gamma V(T \to P) = \Gamma V(P)$
$\Gamma V(\lambda x.P) = \Gamma V(P)$
$\Gamma V(\forall X.P) = \Gamma V(P) - \{X\}$

For the rule $\leadsto_s$, it corresponds to the Var rule in definition 2.2. We use this rule to substitute the term variable with its type in Γ. So in a sense, the use of typing context as substitution can be defered by the use of this rewrite rule in our formulation.

The rule $\leadsto_r$ is letting us get rid of the typing context. The important thing here is that once we get rid of the typing context, we get a type T, which is can only be further reduced by the rule $\leadsto_\iota$. The intuition for $\leadsto_\iota$ is the $\forall$ elimination, but we have to define $[T/X]P$ in definition 2.13. We can extend the notion of Γx and ΓX to Γt and ΓT(definition 2.11, definition 2.12).

▶ **Definition 2.11.** We extend Γx to Γt.

$\Gamma(t_1 t_2) \equiv (\Gamma t_1)(\Gamma t_2)$.
$\Gamma(\lambda x.t) \equiv \lambda x.(\Gamma t)$. This will invoke term variable renaming as nessesary.

▶ **Definition 2.12.** We extend ΓX to ΓT.

$\Gamma(T_1 \to T_2) \equiv T_1 \to (\Gamma T_2)$.
$\Gamma(\forall X.T) \equiv \forall X.(\Gamma T)$. This will invoke type variable renaming as nessesary.

▶ **Definition 2.13.** We define substitution $[T/X]P$ on pretypes P:

$[T/X](\Gamma x) \equiv ([T/X]\Gamma)x$, where $[T/X]\Gamma$ means apply the substitution $[T/X]$ on each type in Γ.
$[T/X]X \equiv T$.
$[T/X]\Gamma X' \equiv ([T/X]\Gamma)([T/X]X')$.
$[T/X](T' \to P) \equiv [T/X]T' \to [T/X]P$
$[T/X]P_1 P_2 \equiv ([T/X]P_1)([T/X]P_2)$
$[T/X]\lambda x.P \equiv \lambda x.[T/X]P$
$[T/X]\forall Y.P \equiv \forall Y.([T/X]P)$. This will invoke renaming and capture avoiding if nessesary.

The following example simulate the typing of $y : T \vdash (\lambda x.x)y : T$.

$[y : T]((\lambda x.x)y) \equiv (\lambda x.[y : T]x)([y : T]y) \leadsto_\lambda (T \to [x : T, y : T]x)([y : T]y) \leadsto_s (T \to [x : T, y : T]x)([y : T]T) \leadsto_r (T \to [x : T, y : T]x)T \leadsto_\epsilon [x : T, y : T]x \leadsto_s [x : T, y : T]T \leadsto_r T$.

The following example simulate the typing of $x : Y \vdash x : \forall Z.Y$.

$[x : Y]x \leadsto_\pi \forall Z.([x : Y]x) \leadsto_s \forall Z.([x : Y]Y) \leadsto_r \forall Z.Y$.

2.4 Properties

We will give several lemmas stating properties about our new rewriting formulation. We will use $\leadsto$ as a short hand for $\leadsto_\lambda \cup \leadsto_\epsilon \cup \leadsto_\pi \cup \leadsto_\iota \cup \leadsto_s \cup \leadsto_r$. And $\overset{*}{\leadsto}$ is the reflective and transitive closure of $\leadsto$. We will use $\forall X^n.P$ as a short hand for $\forall X_n.\forall X_{n-1}...\forall X_1.P$ and

$\forall X^0.P \equiv P$. We use $FV(\Gamma)$ to denote all the free type variable in Γ. Some of the proofs can be found in the appendix.

▶ **Lemma 2.14** (Congruence). *If $P \overset{*}{\leadsto} P'$, then $E[P] \overset{*}{\leadsto} E[P']$.*

It is straightforward by the definition of $P \leadsto P'$.

▶ **Lemma 2.15** (Closed Under Substitution). *If $P \leadsto P'$, then for any type level substitution δ, we have $\delta P \leadsto \delta P'$.*

We can prove this by induction on the derivation of $P \leadsto P'$.

▶ **Lemma 2.16** (Compatible with Contextual Action). *If $P \leadsto P'$, then $[x : T] \cdot P \leadsto [x : T] \cdot P'$.*

We can prove this by induction on the derivation of $P \leadsto P'$. The three lemmas above establish basic intuition for our rewriting formulation.

▶ **Lemma 2.17.** *If $\Gamma t \overset{*}{\leadsto} \forall Y^m.P$, then $\{Y_1, ..., Y_m\} \cap FV(\Gamma) = \emptyset$. Assuming modulo alpha equivalence.*

Proof. By induction on the length of $\Gamma t \overset{*}{\leadsto} \forall Y^m.P$.

Base Case: $m = 0$ and $P \equiv \Gamma t$. So it is the case.

Step Case: $\Gamma t \overset{*}{\leadsto} P' \leadsto \forall Y^m.P$. Now case split on the last step $\leadsto$:

$P' \equiv \forall Y^{m-1}.P \leadsto_\pi \forall Y^m.P$. By IH, we have $\{Y_1, ..., Y_{m-1}\} \cap FV(\Gamma) = \emptyset$. And by the restriction on the $\leadsto_\pi$, we know that $Y_m \notin \Gamma V(P)$. We can use renaming to make sure $Y_m \notin FV(\Gamma)$. Thus we have $\{Y_1, ..., Y_m\} \cap FV(\Gamma) = \emptyset$.

$P' \equiv \forall Y^m.P \leadsto_\iota \forall Y^{m-1}.[U/Y_m]P$. By IH, we have $\{Y_1, ..., Y_m\} \cap FV(\Gamma) = \emptyset$. Thus $\{Y_1, ..., Y_{m-1}\} \cap FV(\Gamma) = \emptyset$.

$P' \equiv \forall Y^m.P_1 \leadsto \forall Y^m.P$, where $P_1 \leadsto P$. By IH, we have $\{Y_1, ..., Y_m\} \cap FV(\Gamma) = \emptyset$. So it is the case.

◀

Remarks: This lemma is saying that the type variable $Y_1, ..., Y_m$ we generalize along with the reductions are not in the set $FV(\Gamma)$.

▶ **Lemma 2.18** (Abstraction Inversion). *If $\forall X^n.(\lambda x.P) \overset{*}{\leadsto} T$, then there are T_1, P', m such that $\forall X^n.\lambda x.P \overset{*}{\leadsto} \forall Y^m.\lambda x.P' \leadsto_\lambda \forall Y^m.(T_1 \to [x : T_1] \cdot P') \overset{*}{\leadsto} T$ and $\forall X^n.P \overset{*}{\leadsto} P'$.*

Proof. By induction on the length of $\forall X^n.(\lambda x.P) \overset{*}{\leadsto} T$.

Base Case: It is impossible to arise.

Step Case: $\forall X^n.(\lambda x.P) \leadsto P' \overset{*}{\leadsto} T$. Case split on the first step $\leadsto$.

If $\forall X^n.(\lambda x.P) \leadsto_\lambda \forall X^n.(T_1 \to [x : T_1] \cdot P) \overset{*}{\leadsto} T$. So it is the case. In this case, $\forall X^n.P \overset{*}{\leadsto}_\iota P$.

If $\forall X^n.\lambda x.P \leadsto_\pi \forall X^{n+1}.\lambda x.P \overset{*}{\leadsto} T$. By IH, we have $\forall X^{n+1}.\lambda x.P \overset{*}{\leadsto} \forall Y^m.\lambda x.P' \leadsto_\lambda \forall Y^m.(T_1 \to [x : T_1] \cdot P') \overset{*}{\leadsto} T$ and $\forall X^{n+1}.P \overset{*}{\leadsto} P'$. Thus we have $\forall X^n.P \leadsto_\pi \forall X^{n+1}.P \overset{*}{\leadsto} P'$.

If $\forall X^n.\lambda x.P \leadsto_\iota \forall X^{n-1}.\lambda x.[U/X_n]P \overset{*}{\leadsto} T$. By IH, we have $\forall X^{n-1}.\lambda x.[U/X_n]P \overset{*}{\leadsto} \forall Y^m.\lambda x.P' \leadsto_\lambda \forall Y^m.(T_1 \to [x : T_1] \cdot P') \overset{*}{\leadsto} T$ and $\forall X^{n-1}.[U/X_n]P \overset{*}{\leadsto} P'$. Thus we have $\forall X^n.P \leadsto_\iota \forall X^{n-1}.[U/X_n]P \overset{*}{\leadsto} P'$.

If $\forall X^n.\lambda x.P \leadsto \forall X^n.\lambda x.P'' \overset{*}{\leadsto} T$, where $P \leadsto P''$. By IH we have $\forall X^n.\lambda x.P'' \overset{*}{\leadsto} \forall Y^m.\lambda x.P' \leadsto_\lambda \forall Y^m.(T_1 \to [x : T_1] \cdot P') \overset{*}{\leadsto} T$ and $\forall X^n.P'' \overset{*}{\leadsto} P'$. Thus $\forall X^n.P \leadsto \forall X^n.P'' \overset{*}{\leadsto} P'$.

◀

Remarks: If the reduction sequence is of the form $\forall X^n.(\lambda x.P) \overset{*}{\leadsto} T$, then we know that there must be a $\leadsto_\lambda$ reduction appears in that sequence. And we can extract another sequence $\forall X^n.P \overset{*}{\leadsto} P'$ from this sequence.

▶ **Lemma 2.19** (Arrow Inference). *If $\forall X^n.(T \to P) \overset{*}{\leadsto} T'$, then $T' \equiv \forall Y^m.(T_1 \to T_2)$, $\delta T \equiv T_1$, $\delta P \overset{*}{\leadsto} T_2$ for some type level substitution δ.*

Proof. By induction on the length of $\forall X^n.(T \to P) \overset{*}{\leadsto} T'$.

Base Case: $\forall X^n.(T \to P) \equiv T'$. It is the case.

Step Case: $\forall X^n.(T \to P) \leadsto P' \overset{*}{\leadsto} T'$. Case split on the first step.

If $\forall X^n.(T \to P) \leadsto_\pi \forall X^{n+1}.(T \to P) \overset{*}{\leadsto} T'$. By IH, $T' \equiv \forall Y^m.(T_1 \to T_2)$ and $\delta T \equiv T_1, \delta P \overset{*}{\leadsto} T_2$. So it is the case.

If $\forall X^n.(T \to P) \leadsto_\iota \forall X^{n-1}.([U/X_n]T \to [U/X_n]P) \overset{*}{\leadsto} T'$. By IH, we have $\delta([U/X_n]T) \equiv [\delta U/X_n]\delta T \equiv T_1$ and $\delta([U/X_n]P) \equiv [\delta U/X_n]\delta P \overset{*}{\leadsto} T_2$. So it is the case.

If $\forall X^n.(T \to P) \leadsto \forall X^n.(T \to P'') \overset{*}{\leadsto} T'$, where $P \leadsto P''$. By IH, we have $\delta T \equiv T_1$ and $\delta P'' \overset{*}{\leadsto} T_2$. So $\delta P \overset{*}{\leadsto} \delta P'' \overset{*}{\leadsto} T_2$ by compatible with substitution(lemma 2.15). Thus it is the case.

◀

Remarks: If we have a reduction sequence that is of the form $\forall X^n.(T \to P) \overset{*}{\leadsto} T'$, then we can specify the form of T'. We can see from the proof above that if $X \in dom(\delta)$, then either $X \in \{X_1, ..., X_n\}$ or X is generated by the rule $\leadsto_\pi$.

▶ **Lemma 2.20** (Application Inversion). *If $\forall X^n.P_1P_2 \overset{*}{\leadsto} T$, then there exists P', m, T_1 such that $\forall X^n.P_1P_2 \overset{*}{\leadsto} \forall Y^m.(T_1 \to P')T_1 \leadsto_\epsilon \forall Y^m.P' \overset{*}{\leadsto} T$. Also we have $\forall X^n.P_1 \overset{*}{\leadsto} T_1 \to P'$ and $\forall X^n.P_2 \overset{*}{\leadsto} T_1$.*

Proof. By induction on the length of $\forall X^n.P_1P_2 \overset{*}{\leadsto} T$.

Base Case: Impossible to arise.

Step Case: $\forall X^n.P_1P_2 \leadsto P_3 \overset{*}{\leadsto} T$. Case Split on $\leadsto$:

If $P_1 \equiv T_1 \to P'$ and $P_2 \equiv T_1$, then $\forall X^n.P_1P_2 \equiv \forall X^n.(T_1 \to P')T_1 \leadsto_\epsilon P_3 \equiv \forall X^n.P' \overset{*}{\leadsto} T$. Also $\forall X^n.(T_1 \to P') \overset{*}{\leadsto} T_1 \to P'$ and $\forall X^n.T_1 \overset{*}{\leadsto} T_1$. Thus it is the case.

If $\forall X^n.P_1P_2 \leadsto_\pi \forall X^{n+1}.P_1P_2 \overset{*}{\leadsto} T$. By IH, $\forall X^{n+1}.P_1P_2 \overset{*}{\leadsto} \forall Y^m.(T_1 \to P')T_1 \leadsto_\epsilon \forall Y^m.P' \overset{*}{\leadsto} T$,$\forall X^{n+1}.P_1 \overset{*}{\leadsto} T_1 \to P'$ and $\forall X^{n+1}.P_2 \overset{*}{\leadsto} T_1$. Thus we have $\forall X^n.P_1 \leadsto_\pi \forall X^{n+1}.P_1 \overset{*}{\leadsto} T_1 \to P'$ and $\forall X^n.P_2 \leadsto_\pi \forall X^{n+1}.P_2 \overset{*}{\leadsto} T_1$. So it is the case.

If $\forall X^n.P_1P_2 \leadsto_\iota \forall X^{n-1}.[U/X_n](P_1P_2) \overset{*}{\leadsto} T$. By IH, we have $\forall X^{n-1}.[U/X_n](P_1P_2) \overset{*}{\leadsto} \forall Y^m.(T_1 \to P')T_1 \leadsto_\epsilon \forall Y^m.P' \overset{*}{\leadsto} T$, $\forall X^{n-1}.[U/X_n]P_1 \overset{*}{\leadsto} T_1 \to P'$ and $\forall X^{n-1}.[U/X_n]P_2 \overset{*}{\leadsto} T_1$. Thus we have $\forall X^n.P_1 \leadsto_\iota \forall X^{n-1}.[U/X_n]P_1 \overset{*}{\leadsto} T_1 \to P'$ and $\forall X^n.P_2 \leadsto_\iota \forall X^{n-1}.[U/X_n]P_2 \overset{*}{\leadsto} T_1$.

If $\forall X^n.P_1P_2 \leadsto \forall X^n P_1'P_2 \overset{*}{\leadsto} T$, where $P_1 \leadsto P_1'$. By IH, we have $\forall X^n P_1'P_2 \overset{*}{\leadsto} \forall Y^m.(T_1 \to P')T_1 \leadsto_\epsilon \forall Y^m.P' \overset{*}{\leadsto} T$, $\forall X^n.P_1' \overset{*}{\leadsto} T_1 \to P'$ and $\forall X^n.P_2 \overset{*}{\leadsto} T_1$. Thus $\forall X^n.P_1 \leadsto \forall X^n.P_1' \overset{*}{\leadsto} T_1 \to P$. So it is the case.

◀

Remarks: This lemma is telling us that there must be a $\leadsto_\epsilon$ reduction within the sequence $\forall X^n.P_1P_2 \overset{*}{\leadsto} T$. And we can split out two sequences $\forall X^n.P_1 \overset{*}{\leadsto} T_1 \to P'$ and $\forall X^n.P_2 \overset{*}{\leadsto} T_1$ from this sequence.

▶ **Lemma 2.21.** *If $\forall X^n.P \overset{*}{\leadsto} T$, then there exists δ such that $\delta P \overset{*}{\leadsto} T$.*

Proof. By induction on the length of $\forall X^n.P \overset{*}{\leadsto} T$.

Base Case: Obvious.

Step Case: $\forall X^n.P \leadsto P' \overset{*}{\leadsto} T$. Case split on the first step.

$\forall X^n.P \leadsto_\pi \forall X^{n+1}.P \equiv P' \overset{*}{\leadsto} T$. By IH, $\delta P \overset{*}{\leadsto} T$, so it is the case.

$\forall X^n.P \leadsto_\iota \forall X^{n-1}.[U/X_n]P \equiv P' \overset{*}{\leadsto} T$. By IH, $\delta([U/X_n]P) \overset{*}{\leadsto} T$. So it is the case.

$\forall X^n.P \leadsto \forall X^n.P' \overset{*}{\leadsto} T$, where $P \leadsto P'$. By IH, we have $\delta P' \overset{*}{\leadsto} T$. Thus $\delta P \leadsto \delta P' \overset{*}{\leadsto} T$. So it is the case.

◀

Remarks: We can see from the proof above that if $X \in dom(\delta)$, then either $X \in \{X_1, ..., X_n\}$ or X is generated by the rule $\leadsto_\pi$.

2.5 Soundness and Completeness

The theorems in this section establish the equivalence of Curry-style system F and our rewriting formulation.

▶ **Theorem 2.22** (Soundness). *If $\Gamma \vdash t : T$, then $\Gamma t \overset{*}{\leadsto} T$.*

The proof for this theorem is by induction on the derivation of $\Gamma \vdash t : T$.

▶ **Theorem 2.23** (Completeness). *If $\Gamma t \overset{*}{\leadsto} T$, then $\Gamma \vdash t : T$.*

Proof. By induction on the structure of t.

Base Case: $t \equiv x$.

So we have $\Gamma x \overset{*}{\leadsto} T$. It must be that $\Gamma x \overset{*}{\leadsto}_{\pi,\iota} \forall X^n.\Gamma x \leadsto_s \forall X^n.\Gamma T' \overset{*}{\leadsto}_{\pi,\iota} T$, where $(x : T') \in \Gamma$. So by the restriction on $\leadsto_\pi$, we have $\Gamma \vdash x : T$.

Step Case: $t \equiv t_1 t_2$.

We have $(\Gamma t_1)(\Gamma t_2) \overset{*}{\leadsto} T$. By lemma 2.20, we have $(\Gamma t_1)(\Gamma t_2) \overset{*}{\leadsto} \forall Y^n.(T_1 \to P')T_1 \leadsto_\epsilon \forall Y^n.P' \overset{*}{\leadsto} T$, $\Gamma t_1 \overset{*}{\leadsto} T_1 \to P$, $\Gamma t_2 \overset{*}{\leadsto} T_1$ and $\forall Y^n.P' \overset{*}{\leadsto} T$. By lemma 2.17, we know that $\{Y_1, ..., Y_n\} \cap FV(\Gamma) = \emptyset$. By lemma 2.21, we have $\delta P \overset{*}{\leadsto} T$, and if $X \in dom(\delta)$, then $X \in \{Y_1, ..., Y_n\}$ or X is generated by $\leadsto_\pi$. So $dom(\delta) \cap FV(\Gamma) = \emptyset$. So we have $\delta(\Gamma t_1) \equiv \Gamma t_1 \overset{*}{\leadsto} \delta T_1 \to \delta P \overset{*}{\leadsto} \delta T_1 \to T$. And we also have $\delta(\Gamma t_2) \equiv \Gamma t_2 \overset{*}{\leadsto} \delta T_1$. By IH, we have $\Gamma \vdash t_1 : \delta T_1 \to T$ and $\Gamma \vdash t_2 : \delta T_1$. So we have $\Gamma \vdash t_1 t_2 : T$.

Step Case: $t \equiv \lambda x.t'$.

We have $\lambda x.(\Gamma t') \overset{*}{\leadsto} T$. By lemma 2.18, we have $\lambda x.\Gamma t' \overset{*}{\leadsto} \forall Y^n.(\lambda x.P) \leadsto_\lambda \forall Y^n.(T_1 \to [x : T_1] \cdot P) \overset{*}{\leadsto} T$ and $\Gamma t' \overset{*}{\leadsto} P$. By lemma 2.17, $\{Y_1, ..., Y_n\} \cap FV(\Gamma) = \emptyset$. We also have $[\Gamma, x : T_1]t' \overset{*}{\leadsto} [x : T_1] \cdot P$ by compatible with contextual action. By lemma 2.19, we have $T \equiv \forall Z^n.(T_3 \to T_4)$, $\delta T_1 \equiv T_3$ and $[x : \delta T_1] \cdot \delta P \overset{*}{\leadsto} T_4$, and if $X \in dom(\delta)$, then $X \in \{Y_1, ..., Y_n\}$ or X is generated by $\leadsto_\pi$. Thus $dom(\delta) \cap FV(\Gamma) = \emptyset$. So we have $\delta([\Gamma, x : T_1]t') \equiv [\Gamma, x : \delta T_1]t' \overset{*}{\leadsto} [x : \delta T_1] \cdot \delta P \overset{*}{\leadsto} T_4$. Thus we have $[\Gamma, x : T_3]t' \overset{*}{\leadsto} T_4$. By IH, we have $\Gamma, x : T_3 \vdash t' : T_4$. Thus by lemma 2.17, we know $\{Z_1, ..., Z_n\} \cap FV(\Gamma) = \emptyset$. Thus we have $\Gamma \vdash \lambda x.t' : \forall Z^n.(T_3 \to T_4)$.

◀

3 Meta-theorems

Now we already get the equivalence of the rewriting formulation and Curry-style system F. We would like to see if we can prove some standard meta-theorems for our rewriting formulation. The canonical proof method we use in our rewriting formulation is by induction on the structure of term t, one can inspect the proof of theorem 2.23, lemma 3.1 and theorem 3.9 to see this pattern. This proof pattern resembles the proof by induction on the derivation of $\Gamma \vdash t : T$ in traditional presentation of type system.

3.1 Type Preservation

▶ **Lemma 3.1.** *If* $[\Gamma, x : T_1]t_1 \overset{*}{\leadsto} T$ *and* $\Gamma t_2 \overset{*}{\leadsto} T_1$*, then* $\Gamma([t_2/x]t_1) \overset{*}{\leadsto} T$.

Proof. By induction on the structure of t_1.

Base Case: $t_1 \equiv x$. We have $[\Gamma, x : T_1]x \overset{*}{\leadsto} T$ and $\Gamma t_2 \overset{*}{\leadsto} T_1$. We know that $T \equiv \forall X^n.T_1$. By lemma 2.17, $\{X_1, ..., X_n\} \cap FV([\Gamma, x : T_1]) = \emptyset$. Thus $\Gamma([t_2/x]x) \equiv \Gamma t_2 \overset{*}{\leadsto} \forall X^n.(\Gamma t_2) \overset{*}{\leadsto} \forall X^n.T_1 \equiv T$. So it is the case.

Step Case: $t_1 \equiv \lambda y.t'$. We have $[\Gamma, x : T_1](\lambda y.t') \overset{*}{\leadsto} T$ and $\Gamma t_2 \overset{*}{\leadsto} T_1$. By lemma 2.18, we have $[\Gamma, x : T_1](\lambda y.t') \equiv \lambda y.[\Gamma, x : T_1]t' \overset{*}{\leadsto} \forall X^n.\lambda y.P \leadsto_\lambda \forall X^n.(T_x \to [y : T_x] \cdot P) \overset{*}{\leadsto} T$ and $[\Gamma, x : T_1]t' \overset{*}{\leadsto} P$. By lemma 2.19, we have $T \equiv \forall Z^m.(T_a \to T_b)$. And we have a type substitution δ, where $dom(\delta) \cap (FV(\Gamma) \cup FV(T_1)) = \emptyset$, such that $\delta([y : T_x] \cdot P) \overset{*}{\leadsto} T_b$ and $\delta T_x \equiv T_a$. So by compatible with contextual action, we have $[\Gamma, x : T_1, y : T_x]t' \overset{*}{\leadsto} [y : T_x] \cdot P$. By closed under substitution, we have $[\Gamma, x : T_1, y : \delta T_x]t' \overset{*}{\leadsto} [y : \delta T_x] \cdot \delta P \overset{*}{\leadsto} T_b$. So we have $\Gamma(\lambda y.[t_2/x]t') \equiv \lambda y.\Gamma([t_2/x]t') \leadsto_\lambda \delta T_x \to [\Gamma, y : \delta T_x]([t_2/x]t')$. By IH, we have $[\Gamma, y : \delta T_x]([t_2/x]t') \overset{*}{\leadsto} T_b$. By lemma 2.17, $\{Z_1, ..., Z_n\} \cap FV([\Gamma, x : T_1]) = \emptyset$. So we have $\Gamma(\lambda y.[t_2/x]t') \equiv \lambda y.\Gamma([t_2/x]t') \leadsto_\lambda \delta T_x \to [\Gamma, y : \delta T_x]([t_2/x]t') \overset{*}{\leadsto} \forall Z^m.(\delta T_x \to [\Gamma, y : \delta T_x]([t_2/x]t')) \overset{*}{\leadsto} \forall Z^m.(\delta T_x \to T_b) \equiv \forall Z^m.(T_a \to T_b)$. So it is the case.

Step Case: $t_1 \equiv t_a t_b$. We have $[\Gamma, x : T_1](t_a t_b) \overset{*}{\leadsto} T$ and $\Gamma t_2 \overset{*}{\leadsto} T_1$. By lemma 2.20, we have $[\Gamma, x : T_1](t_a t_b) \overset{*}{\leadsto} \forall Y^n.(T_x \to P)T_x \leadsto_\epsilon \forall Y^n.P \overset{*}{\leadsto} T$, $[\Gamma, x : T_1]t_a \overset{*}{\leadsto} T_x \to P$ and $[\Gamma, x : T_1]t_b \overset{*}{\leadsto} T_x$. By lemma 2.21, we know there is a type substitution δ, where $dom(\delta) \cap (FV(\Gamma) \cup FV(T_1)) = \emptyset$, such that $\delta P \overset{*}{\leadsto} T$. Thus $[\Gamma, x : T_1]t_a \overset{*}{\leadsto} \delta T_x \to \delta P \overset{*}{\leadsto} \delta T_x \to T$ and $[\Gamma, x : T_1]t_b \overset{*}{\leadsto} \delta T_x$. By IH, we have $\Gamma([t_2/x]t_a) \overset{*}{\leadsto} \delta T_x \to T$ and $\Gamma([t_2/x]t_b) \overset{*}{\leadsto} \delta T_x$. Thus $\Gamma([t_2/x]t_a)[t_2/x]t_b \overset{*}{\leadsto} (\delta T_x \to T)\delta T_x \leadsto_\epsilon T$. So it is the case.

◀

Remarks: This lemma corresponds to this theorem: If $\Gamma, x : T_1 \vdash t_1 : T$ and $\Gamma \vdash t_2 : T_1$, then $\Gamma \vdash [t_2/x]t_1 : T$.

▶ **Theorem 3.2** (Type Preservation). *If $\Gamma t \overset{*}{\leadsto} T$ and $t \leadsto_\beta t'$, then $\Gamma t' \overset{*}{\leadsto} T$.*

Proof. We prove this theorem by induction on the derivation of $t \leadsto_\beta t'$. We will treat only the case that $t \equiv (\lambda x.t_1)t_2$ and $t' \equiv [t_2/x]t_1$. Since we know that $\Gamma(\lambda x.t_1)t_2 \overset{*}{\leadsto} T$. By lemma 2.20, we know that $\Gamma\lambda x.t_1 \overset{*}{\leadsto} T_x \to P$, $\Gamma t_2 \overset{*}{\leadsto} T_x$ and $\forall Y^m.P \overset{*}{\leadsto} T$. By lemma 2.21, there is a type level subsitution δ, where $dom(\delta) \cap FV(\Gamma) = \emptyset$, such that $\delta P \overset{*}{\leadsto} T$. Thus we have $\Gamma\lambda x.t_1 \overset{*}{\leadsto} \delta T_x \to T$ and $\Gamma t_2 \overset{*}{\leadsto} \delta T_x$. By lemma 2.18, $\lambda x.\Gamma t_1 \overset{*}{\leadsto} \forall Z^q.\lambda x.P_1 \leadsto_\lambda \forall Z^q.(T_a \to [x : T_a] \cdot P_1) \overset{*}{\leadsto} \delta T_x \to T$ and $\Gamma t_1 \overset{*}{\leadsto} P_1$. By lemma 2.19, we have a type level substitution δ', where $dom(\delta') \cap FV(\Gamma) = \emptyset$, such that $\delta' T_a \equiv \delta T_x$, and $[x : \delta' T_a] \cdot \delta' P_1 \overset{*}{\leadsto} T$. Thus $[\Gamma, x : \delta T_x]t_1 \overset{*}{\leadsto} [x : \delta' T_a] \cdot \delta' P_1 \overset{*}{\leadsto} T$. By lemma 3.1, we have $\Gamma([t_2/x]t_1) \overset{*}{\leadsto} T$. So it is the case.

◀

The proof of this theorem in traditional type system can be found in Barendregt's book [2]. Barendregt defines a special kind of ordering on tyeps: $T \geq T'$ iff $T' \equiv \forall X.T$ or $T \equiv \forall X.T_1, T' \equiv [T_2/X]T_1$. Then he states a kind of inversion lemmas (he call them *Generation lemmas*) based on his ordering. e.g. if $\Gamma \vdash \lambda x.t : T$, then there are T_1, T_2 such that $T_1 \to T_2 \geq T$ and $\Gamma, x : T_1 \vdash t : T_2$. He prove a special property for his ordering, nameley, if $T_1 \to T_2 \geq T_1' \to T_2'$, then exists a type substitution δ such that $\delta(T_1 \to T_2) \equiv T_1' \to T_2'$. Finally he is able to prove type preservation with these tools he developed.

While in our rewriting formulation, all the important properties are already proved in section 2.4, previously we used them to show the completeness theorem. We can reuse these lemmas to prove type preservation. Later will see we can again reuse them to prove strong normalization.

3.2 Strong Normalization

We borrow Girard's reducibility candidates [3] directly. We develop a new form of type soundness theorem and we are able to prove it using the properties in section 2.4.

▶ **Definition 3.3** (Reducibility Candidate). A reducibility candidate $\mathcal{R}$ is a set of terms that satisfies the following conditions:

CR 1 If $t \in \mathcal{R}$,then t is strong normalizing.
CR 2 If $t \in \mathcal{R}$ *and* $t \leadsto t'$, then $t' \in \mathcal{R}$.
CR 3 If t is neutral, and for any t' such that $t \leadsto t'$ *and* $t' \in \mathcal{R}$, then $t \in \mathcal{R}$.

Notice that a term is neutral iff it is of the form x, tt'.

▶ **Definition 3.4** (Environment Mapping)**.** Let $\Re$ be the set of all reducibility candidates. Let $\mathcal{V}$ be the set of type variables. Let $FV(T)$ be the set of free type variable in T. Apparently $FV(T) \subseteq \mathcal{V}$. Let ϕ be an environment function: $\mathcal{V} \to \Re$. e.g. $\phi(X) \in \Re$. We usually write $\phi[\mathcal{R}/X]$ to mean $\phi[\mathcal{R}/X]$ is exactly like ϕ except $\phi(X) = \mathcal{R}$.

▶ **Definition 3.5** (Reducibility Sets)**.** The reducibility set $[\![T]\!]_\phi$ is defined inductively as follows.

$t \in [\![X]\!]_\phi$ iff $t \in \phi(X)$.
$t \in [\![T_1 \to T_2]\!]_\phi$ iff $(\forall u \in [\![T_1]\!]_\phi \Rightarrow (t\ u) \in [\![T_2]\!]_\phi)$.
$t \in [\![\forall X.T]\!]_\phi$ iff $t \in [\![T]\!]_{\phi[\mathcal{R}/X]}$ for any $\mathcal{R} \in \Re$.

▶ **Lemma 3.6.** $[\![T]\!]_\phi \in \Re$.

▶ **Lemma 3.7.** *If* $u \in [\![T]\!]_\phi$ *and* $[u/x]t \in [\![T']\!]_\phi$*, then* $\lambda x.t \in [\![T \to T']\!]_\phi$*.*

The definitions and lemmas listed above are quite standard and can be found in Girard's book [3].

▶ **Definition 3.8.** We define the relation $\sigma \in [\Gamma]_\phi$ inductively as follows:

$$\frac{}{\emptyset \in [.]_\phi} \qquad \frac{\sigma \in [\Gamma]_\phi \quad t \in [\![T]\!]_\phi}{\sigma \cup [t/x] \in [\Gamma, x : T]_\phi}$$

Remarks: When we write $[\Gamma]_\phi$, we mean a set of σ such that $\sigma \in [\Gamma]_\phi$. If for any $X \in FV(\Gamma)$, $\phi(X) = \phi'(X)$, then $[\Gamma]_\phi = [\Gamma]_{\phi'}$.

▶ **Theorem 3.9** (Type Soundness)**.** *If* $\Gamma t \overset{*}{\rightsquigarrow} T$ *and* $\sigma \in [\Gamma]_\phi, \sigma t \in [\![T]\!]_\phi$*.*

Proof. By induction on the structure of t.

Base Case: $t \equiv x$.

So we have $\Gamma x \overset{*}{\rightsquigarrow} T$. So $\overset{*}{\rightsquigarrow}$ can only be a combination of $\rightsquigarrow_r, \rightsquigarrow_s, \rightsquigarrow_\pi, \rightsquigarrow_\iota$. Due to the restriction on the $\rightsquigarrow_\pi$. $\rightsquigarrow_\iota$ can not change the type in Γ. So we have $(x : T') \in \Gamma$, where $\forall Y^n.T' \equiv T$. Notice here $\{Y_1, ..., Y_n\} \cap FV(T') = \emptyset$. Since $\sigma \in [\Gamma]_\phi$, we have $\sigma(x) \in [\![T']\!]_\phi = [\![\forall Y^n.T']\!]_\phi$.

Step Case: $t \equiv t_1 t_2$.

We have $(\Gamma t_1)(\Gamma t_2) \overset{*}{\rightsquigarrow} T$. By lemma 2.20, we have $(\Gamma t_1)(\Gamma t_2) \overset{*}{\rightsquigarrow} \forall Y^n.(T_1 \to P')T_1 \rightsquigarrow_\epsilon \forall Y^n.P' \overset{*}{\rightsquigarrow} T$, $\Gamma t_1 \overset{*}{\rightsquigarrow} T_1 \to P$, $\Gamma t_2 \overset{*}{\rightsquigarrow} T_1$ and $\forall Y^n.P' \overset{*}{\rightsquigarrow} T$. By lemma 2.21, we have $\delta P \overset{*}{\rightsquigarrow} T$ and $dom(\delta) \cap FV(\Gamma) = \emptyset$. So we have $\delta(\Gamma t_1) \equiv \Gamma t_1 \overset{*}{\rightsquigarrow} \delta T_1 \to \delta P \overset{*}{\rightsquigarrow} \delta T_1 \to T$. And we also have $\delta(\Gamma t_2) \equiv \Gamma t_2 \overset{*}{\rightsquigarrow} \delta T_1$. By IH, we have $\sigma t_1 \in [\![\delta T_1 \to T]\!]_\phi$ and $\sigma t_2 \in [\![\delta T_1]\!]_\phi$. So $\sigma(t_1 t_2) \in [\![T]\!]_\phi$.

Step Case: $t \equiv \lambda x.t'$.

We have $\lambda x.(\Gamma t') \overset{*}{\rightsquigarrow} T$. By lemma 2.18, we have $\lambda x.\Gamma t' \overset{*}{\rightsquigarrow} \forall Y^n.(\lambda x.P) \rightsquigarrow_\lambda \forall Y^n.(T_1 \to [x : T_1] \cdot P) \overset{*}{\rightsquigarrow} T$ and $\Gamma t' \overset{*}{\rightsquigarrow} P$. Thus $[\Gamma, x : T_1]t' \overset{*}{\rightsquigarrow} [x : T_1] \cdot P$ by compatible with contextual action. By lemma 2.19, we have $T \equiv \forall Z^n.(T_3 \to T_4)$, $\delta T_1 \equiv T_3$ and $[x : \delta T_1] \cdot \delta P \overset{*}{\rightsquigarrow} T_4$ and

$dom(\delta) \cap FV(\Gamma) = \emptyset$. So we have $\delta([\Gamma, x : T_1]t') \equiv [\Gamma, x : \delta T_1]t' \overset{*}{\rightsquigarrow} [x : \delta T_1] \cdot \delta P \overset{*}{\rightsquigarrow} T_4$. Thus we have $[\Gamma, x : T_3]t' \overset{*}{\rightsquigarrow} T_4$. Let $u \in [\![T_3]\!]_\phi$, thus $\sigma[u/x] \in [\Gamma, x : T_3]$. By IH, $\sigma[u/x]t' \in [\![T_4]\!]_\phi$. So $\lambda x.\sigma t' \in [\![T_3 \to T_4]\!]_\phi$ by lemma 3.7.

Now we want to show $\lambda x.\sigma t' \in [\![\forall Z^n.(T_3 \to T_4)]\!]_\phi$. By definition, we want to show $\lambda x.\sigma t' \in [\![T_3 \to T_4]\!]_{\phi[\mathcal{R}_1/Z_1,...,\mathcal{R}_n/Z_n]}$ for any $\mathcal{R}_1, ..., \mathcal{R}_n$. By lemma 2.17, we know that $FV(\Gamma) \cap \{Z_1, ..., Z_n\} = \emptyset$. Thus $[\Gamma]_\phi = [\Gamma]_{\phi[\mathcal{R}_1/Z_1,...,\mathcal{R}_n/Z_n]}$. Thus $\sigma \in [\Gamma]_{\phi[\mathcal{R}_1/Z_1,...,\mathcal{R}_n/Z_n]}$.So we have $\sigma[u/x] \in [\Gamma, x : T_3]_{\phi[\mathcal{R}_1/Z_1,...,\mathcal{R}_n/Z_n]}$ for any $u \in [\![T_3]\!]_{\phi[\mathcal{R}_1/Z_1,...,\mathcal{R}_n/Z_n]}$. So we have $\sigma[u/x]t' \in [\![T_4]\!]_{\phi[\mathcal{R}_1/Z_1,...,\mathcal{R}_n/Z_n]}$. Thus we have $\lambda x.\sigma t' \in [\![T_3 \to T_4]\!]_{\phi[\mathcal{R}_1/Z_1,...,\mathcal{R}_n/Z_n]}$. So it is the case.

◀

The proof of this theorem is by induction on the structure of t, and the proof does not need the helper lemma like: $[\![[T/X]T']\!]_\phi = [\![T']\!]_{[[\![T]\!]_\phi/X]\phi}$, which is need in the traditional presentation of Curry-style system F.

4 Related Works and Conclusions

Kuan discusses the difficulty of showing completeness theorem in [4], namely, if $\Gamma t \overset{*}{\rightsquigarrow} T$, then $\Gamma \vdash t : T$. He proved it by constructing a CEK machine formulation as an intermediate formulation bewteen rewriting formulation and traditional presentation. While a more direct proof based on rewriting formulation can be found in Stump et al.'s paper [7]. Notice in their rewriting formulation, Γ is treated as substitution.

Using reduction to defer substitution is not new, this technique can be traced back to the idea of explicit subsitution for lambda calculus [1]. The way we treat the typing context reflects the idea of explicit substitution.

As mentioned by Stump et al. [7], for dependent type system poses difficulties for Kuan's rewriting formulation. e.g. $\cdot \vdash \lambda x : \mathsf{Nat}.(\mathsf{Join}\ xx) : (\Pi x : \mathsf{Nat}.x = x)$. This can not be simulated by the rewrite rules $\lambda x : T.t \rightsquigarrow \Pi x : T.[T/x]t$ and $\mathsf{Join}\ tt \rightsquigarrow t = t$. Just as Stump et al.'s remarks:

"Π-bound variables must play a dual role. When computing a dependent function type $\Pi x : T.T'$ from an abstraction $\lambda x : T.t$, we may need to abstract x to T , as for STLC; but we may also need to leave it unabstracted, since with dependent types, x is allowed to appear in the range type T. "

We conjecture the idea of adding typing context may provide a solution for this issue. So naturally, one future direction of our works would be to see if the ideas developed in this paper work for dependent type systems and pure type systems.

References

1 M. Abadi, L. Cardelli, P. L. Curien, and J. J. Levy. Explicit substitutions. In *POPL '90: Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 31–46, New York, NY, USA, 1990. ACM.

2 H. Barendregt. Lambda Calculi with Types. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*. Oxford University Press, 1992.

3 Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*. Cambridge University Press, New York, NY, USA, 1989.

4 George Kuan. A rewriting semantics for type inference. Technical Report TR-2007-03, University of Chicago, 2007.

5 George Kuan. Type checking and inference via reductions. In Matthias Felleisen, Robert Bruce Findler, and Matthew Flatt, editors, *Semantics Engineering with PLT Redex*. MIT Press, 2009.

6 George Kuan, David MacQueen, and Robert Bruce Findler. A rewriting semantics for type inference. In Rocco De Nicola, editor, *Programming Languages and Systems, 16th European Symposium on Programming, ESOP 2007*, volume 4421, March 2007.

7 Aaron Stump, Garrin Kimmell, and Roba El Haj Omar. Type Preservation as a Confluence Problem. In Manfred Schmidt-Schauß, editor, *22nd International Conference on Rewriting Techniques and Applications (RTA'11)*, volume 10 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 345–360, Dagstuhl, Germany, 2011. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

A Proofs

▶ **Lemma 1.1** (Congruence). *If $P \overset{*}{\leadsto} P'$, then $E[P] \overset{*}{\leadsto} E[P']$.*

This lemma is straightforward by the definition of reductions.

▶ **Lemma 1.2** (Closed Under Substitution). *If $P \leadsto P'$, then for any type level substitution δ, we have $\delta P \leadsto \delta P'$.*

Proof. By induction on the derivation of $P \leadsto P'$.

Case:

$$\frac{}{\lambda x.P \leadsto_\lambda T \to [x : T] \cdot P'}$$

We have $\lambda x.\delta P \leadsto_\lambda \delta T \to [x : \delta T] \cdot (\delta P) \equiv \delta(T \to [x : T] \cdot P)$. So it is the case.

Case:

$$\frac{}{(T \to P)T \leadsto_\epsilon P}$$

We have $\delta((T \to P)T) \equiv (\delta T \to \delta P)\delta T \leadsto_\epsilon \delta P$. So it is the case.

Case:

$$\frac{}{\forall X.P \leadsto_\iota [T/X]P}$$

We have $\forall X.\delta P \leadsto_\iota [T/X](\delta P) \equiv \delta([T/X]P)$. Assuming modulo renaming and capture avoiding.

Case:

$$\frac{X \notin \Gamma V(P) \quad P \notin \textbf{Types}}{P \leadsto_\pi \forall X.P}$$

$\delta P \leadsto_\pi \forall X.(\delta P) \equiv \delta(\forall X.P)$, since $\delta P \notin$ **Types** and $X \notin \Gamma V(\delta P)$.

Case:

$$\frac{(x : T) \in \Gamma}{\Gamma x \rightsquigarrow_s \Gamma T}$$

$\delta\Gamma x \rightsquigarrow_s \delta\Gamma\delta T$, since $(x : \delta T) \in \delta\Gamma$.

Case:

$$\frac{}{\Gamma T \rightsquigarrow_r T}$$

$\delta\Gamma\delta T \rightsquigarrow_r \delta T$. So it is the case.

Case:

$$\frac{P \rightsquigarrow P'}{\lambda x.P \rightsquigarrow \lambda x.P'}$$

By IH, we have $\delta P \rightsquigarrow \delta P'$. Thus we have $\lambda x.\delta P \equiv \delta(\lambda x.P) \rightsquigarrow \lambda x.\delta P' \equiv \delta(\lambda x.P')$.

Case:

$$\frac{P_2 \rightsquigarrow P_2'}{P_1 P_2 \rightsquigarrow P_1 P_2'}$$

By IH,we have $\delta P_2 \rightsquigarrow \delta P_2'$. So $\delta P_1 \delta P_2 \rightsquigarrow \delta P_1 \delta P_2'$. So it is the case.

The other cases are similar. ◀

▶ **Lemma 1.3** (Compatible with TypContext Action). *If $P \rightsquigarrow P'$, then $[x : T] \cdot P \rightsquigarrow [x : T] \cdot P'$.*

Proof. By induction on the derivation of $P \rightsquigarrow P'$.

Base Case 1: $\lambda x.P \rightsquigarrow_\lambda T_1 \to [x : T_1] \cdot P$. We also have $[y : T] \cdot (\lambda x.P) \equiv \lambda x.([y : T] \cdot P) \rightsquigarrow_\lambda T_1 \to [x : T_1] \cdot ([y : T] \cdot P) \equiv [y : T] \cdot (T_1 \to [x : T_1] \cdot P)$. So it is the case.

Base Case 2: $(T \to P)T \rightsquigarrow_\epsilon P$. We also have $[x : T'] \cdot ((T \to P)T) \equiv (T \to [x : T'] \cdot P)T \rightsquigarrow_\epsilon [x : T'] \cdot P$.

Base Case 3: $P \rightsquigarrow_\pi \forall X.P$. Then $[x : T] \cdot P \rightsquigarrow_\pi \forall X.[x : T] \cdot P \equiv [x : T] \cdot (\forall X.P)$.

Base Case 4: $\forall X.P \rightsquigarrow_\iota [U/X]P$. Then $[x : T]\cdot(\forall X.P) \equiv \forall X.([x : T]\cdot P) \rightsquigarrow_\iota [x : T]\cdot([U/X]P)$.

Base Case 5: $\Gamma x \rightsquigarrow_s \Gamma T$, where $(x : T) \in \Gamma$. It is the case.

Base Case 6: $\Gamma T \rightsquigarrow_r T$. Obvious it is the case.

Step Case: $P \equiv E[P_1] \rightsquigarrow P' \equiv E[P_2]$, where $P_1 \rightsquigarrow P_2$. We need to show $[x : T] \cdot E[P_1] \rightsquigarrow [x : T] \cdot E[P_2]$. By case split on the form of E:

If $E \equiv T \to E$. Then $[x:T] \cdot (T' \to P_1) \equiv T' \to [x:T] \cdot P_1$. By IH, we have $[x:T] \cdot P_1 \leadsto [x:T] \cdot P_2$. Thus $T' \to [x:T] \cdot P_1 \leadsto T' \to [x:T] \cdot P_2$.

If $E \equiv EP''$. Then $[x:T] \cdot (P_1 P'') \equiv [x:T] \cdot P_1([x:T] \cdot P'')$. By IH, we have $[x:T] \cdot P_1 \leadsto [x:T] \cdot P_2$. Thus $[x:T] \cdot P_1([x:T] \cdot P'') \leadsto [x:T] \cdot P_2([x:T] \cdot P'')$.

If $E \equiv \forall X.E$. Then $[x:T] \cdot (\forall X.P_1) \equiv \forall X.([x:T] \cdot P_1)$. By IH, we have $[x:T] \cdot P_1 \leadsto [x:T] \cdot P_2$. Thus $\forall X.([x:T] \cdot P_1) \leadsto \forall X.([x:T] \cdot P_2)$.

If $E \equiv \lambda y.E$. Then $[x:T] \cdot (\lambda y.P_1) \equiv \lambda y.([x:T] \cdot P_1)$. By IH, we have $[x:T] \cdot P_1 \leadsto [x:T] \cdot P_2$. Thus $\lambda y.([x:T] \cdot P_1) \leadsto \lambda y.([x:T] \cdot P_2)$.

◀

▶ **Theorem 1.4.** *If $\Gamma \vdash t : T$, then $\Gamma t \overset{*}{\leadsto} T$.*

Proof. By induction on the derivation of $\Gamma \vdash t : T$.

Base Case:

$$\frac{(x:T) \in \Gamma}{\Gamma \vdash x : T}\ T\_Var$$

We know that $\Gamma x \leadsto_s \Gamma T \leadsto_r T$, where $(x:T) \in \Gamma$. So it is the case.

Step Case:

$$\frac{\Gamma, x:T_1 \ \vdash t : T_2}{\Gamma \vdash \lambda x.t : T_1 \to T_2}\ \to\_intro$$

We have $\Gamma \lambda x.t \leadsto_\lambda T_1 \to [\Gamma, x:T_1]t$. By IH, we know that $[\Gamma, x:T_1]t \overset{*}{\leadsto} T_2$. By congruence lemma, we know that $\Gamma \lambda x.t \equiv \lambda x.\Gamma t \overset{*}{\leadsto} T_1 \to [\Gamma, x:T_1]t \overset{*}{\leadsto} T_1 \to T_2$.

Step Case:

$$\frac{\Gamma \vdash t_1 : T_1 \to T_2 \quad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t_1\ t_2 : T_2}\ \to\_elim$$

By IH, we know that $\Gamma t_1 \overset{*}{\leadsto} T_1 \to T_2$ and $\Gamma t_2 \overset{*}{\leadsto} T_1$. Thus we have $\Gamma t_1 t_2 \equiv (\Gamma t_1)(\Gamma t_2) \overset{*}{\leadsto} (T_1 \to T_2)T_1 \leadsto_\epsilon T_2$. Thus it is the case.

Step Case:

$$\frac{\Gamma \vdash t : T \quad X \notin FV(\Gamma)}{\Gamma \vdash t : \forall X.T}\ \forall\_intro$$

By IH, we know that $\Gamma t \leadsto T$. And $\Gamma t \leadsto_\pi \forall X.(\Gamma t) \overset{*}{\leadsto} \forall X.T$, where $X \notin FV(\Gamma)$.

Step Case:

$$\frac{\Gamma \vdash t : \forall X.T}{\Gamma \vdash t : [T'/X]T}\ \forall\_elim$$

By IH, we have $\Gamma t \overset{*}{\rightsquigarrow} \forall X.T \rightsquigarrow_{\iota} [T'/X]T$. Thus it is the case.

◀