Rewriting Approach to Type Assignment

Peng Fu, Aaron Stump

September 30, 2011

Background I: Basic Formalism

- Term rewriting concepts
  - Reduction relation: $x_1 \to x_2$.
  - Reflexive transitive closure: $x_1 \stackrel{*}{\to} x_2$
- Lambda calculus
  - Pure lambda terms: $t \ ::= \ x \mid \lambda x.t \mid tt$
  - Beta-Reduction: $(\lambda x.t)t' \to_\beta [t'/x]t$

Background II: Simply Typed Lambda Calculus

Church-Style

- Terms: $t \ ::= x \mid \lambda x : T.t \mid tt$
- Types: $T \ ::= \ B \mid T_1 \Rightarrow T_2$
- Typing Context: $\Gamma ::= \cdot \mid \Gamma, x : T$
- Reduction: $(\lambda x : T.t)t' \to_\beta [t'/x]t$
- Typed Terms(Type Assignment):

$$\frac{(x : T) \in \Gamma}{\Gamma \vdash x : T} \qquad \frac{\Gamma, x : T_1 \ \vdash t : T_2}{\Gamma \vdash \lambda x : T.t : T_1 \Rightarrow T_2}$$

$$\frac{\Gamma \vdash t_1 : T_1 \Rightarrow T_2 \qquad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t_1 \ t_2 : T_2}$$

Background III: Simply Typed Lambda Calculus

Curry-Style

- Terms: $t ::= x \mid \lambda x.t \mid tt$
- Types: $T ::= B \mid T_1 \Rightarrow T_2$
- Typing Context: $\Gamma ::= \cdot \mid \Gamma, x : T$
- Reduction: $(\lambda x.t)t' \to_\beta [t'/x]t$
- Type assignment:

$$\frac{(x : T) \in \Gamma}{\Gamma \vdash x : T} \qquad \frac{\Gamma, x : T_1 \vdash t : T_2}{\Gamma \vdash \lambda x.t : T_1 \Rightarrow T_2}$$

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

Rewrite rules for Church-Style STLC

- *mixed terms*: $m ::= x \mid \lambda x : T.\, m \mid m\, m' \mid B \mid T \Rightarrow m$
- *standard terms*: $t ::= x \mid \lambda x : T.\, t \mid t\, t'$
- Idea: from $\Gamma \vdash t : T$ to $\Gamma t \stackrel{*}{\to} T$. Here Γ means substitue term variables with its types.
- Rewriting rules for typing:
  $\hat{m}[(T \Rightarrow m)\ T] \ \to_{\epsilon}\ \hat{m}[m]$
  $\hat{m}[\lambda x : T.\, m] \ \to_{\lambda}\ \hat{m}[T \Rightarrow [T/x]m]$
- Example: $\lambda x : T.x \to_{\lambda} T \Rightarrow T$.

Rewrite rules for Curry-Style STLC

- *mixed terms*: $m ::= x \mid \lambda x.\, m \mid m\, m' \mid B \mid T \Rightarrow m$
- *standard terms*: $t ::= x \mid \lambda x.\, t \mid t\, t'$
- Reductions:
  $\hat{m}[(T \Rightarrow m)\ T] \ \to_{\epsilon}\ \hat{m}[m]$
  $\hat{m}[\lambda x.\, m] \ \to_{\lambda}\ \hat{m}[T \Rightarrow [T/x]m]$
- Example: $\lambda x.x \to_{\lambda} T \Rightarrow T$.
- Notice: $\to_{\lambda}$ is infinite branching.

Curry System F

We can extend curry-STLC to curry-sytem F by adding new types and type assignments.

- Types: $T \ ::= \ ... \mid \forall X.T$
- New typing rules: $\dfrac{\Gamma \vdash t : T \quad X \notin FV(\Gamma)}{\Gamma \vdash t : \forall X.T}$ $\dfrac{\Gamma \vdash t : \forall X.T'}{\Gamma \vdash t : [T/X]T'}$
- Two temptative new rewriting rules:
  $\hat{m}[\forall X.m] \to_\iota \hat{m}[[T/X]m]$
  $\hat{m}[m] \to_\pi \hat{m}[\forall X.m]$
  Assuming $m \ ::= \ ... \ |\forall X.m$.
- Problem: $[x : X]x \equiv X \to_\pi \forall X.X$, but $x : X \not\vdash x : \forall X.X$.

Rewriting Simulation for Curry System F: an attempt

Keep the typing context Γ around.

- Pretypes: $P ::= T \mid \Gamma t \mid \Gamma T \mid P_1 P_2 \mid P_1 \to P_2 \mid \lambda x.P \mid \forall X.P$
- New rewrite rules:
  $\hat{P}[(\lambda x.P)] \rightsquigarrow_\lambda \hat{P}[T \to [x : T]P]$.

  $\hat{P}[(T \to P)T] \rightsquigarrow_\epsilon \hat{P}[P]$.

  $\hat{P}[P] \rightsquigarrow_\pi \hat{P}[\forall X.P]$, where $X \notin \Gamma V(P)$ and $P \notin$ **Types**.

  $\hat{P}[\forall X.P] \rightsquigarrow_\iota \hat{P}[[T/X]P]$.

  $\hat{P}[\Gamma x] \rightsquigarrow_s \hat{P}[\Gamma T]$, where $(x : T) \in \Gamma$.

  $\hat{P}[\Gamma T] \rightsquigarrow_r \hat{P}[T]$.
- Example: $[x : X]x \rightsquigarrow_s [x : X]X \rightsquigarrow_r X$.

Future Works and Conclusion

- Show $\Gamma \vdash t : T$ iff $\Gamma t \stackrel{*}{\to} T$.
- Prove type preservation for curry system F using rewriting approach.
- For more details, talk to me afterward.
- Thank you very much for listening.