Hereditary Substitution for Stratified System F

Harley Eades
Advisor: Aaron Stump

Computer Science
The University of Iowa

Introduction to Simply Typed λ-Calculus

- A short history.
  - In the 1920's, Alonzo Church invented the lambda-calculus.
  - John McCarthy used the simply typed λ-calculus as the core of Lisp.
- The Language.
  - Terms: $t := x \mid \lambda x : \phi.t \mid t\ t$
  - Types: $\phi := b \mid \phi \to \phi$
- The Type System.
  - Consists of several type-checking rules.
  - Contexts, denoted by Γ, are used to keep track of the types of the free variables within a term.
  - We write $\Gamma \vdash t : \phi$ to denote, t has type ϕ w.r.t context Γ.
- Reduction Strategies.
  - Full β-reduction: $(\lambda x : \phi.t)t' \rightsquigarrow [t'/x]t$.
  - Call By Value: $(\lambda x : \phi.t)v \rightsquigarrow [v/x]t$, v is a value.

Some Simple Examples

- Example Terms.
  - Term 1: $\lambda x : b.x$
    - Type: $b \to b$
  - Term 2: $\lambda x : b \to b.\lambda u : b.x\ u$
    - Type: $(b \to b) \to b \to b$
- Example Computation.

$$
\begin{array}{ll}
 & (\lambda x : b \to b.\lambda u : b.x\ u)(\lambda x : b.x)z \\
\rightsquigarrow & (\lambda u : b.(\lambda x : b.x)\ u)z \\
\rightsquigarrow & (\lambda x : b.x)\ z \\
\rightsquigarrow & z
\end{array}
$$

Hereditary Substitution

- Like ordinary capture avoiding substitution.
- Except, if the substitution introduces a redex, then that redex is recursively reduced.
  - Example: $[(\lambda z : b.z)/x]^{b \to b} xy \; (\rightsquigarrow (\lambda z : b.z) y) = y$.
- Hereditary substitution is a terminating function.

(Weak) Normalization

Definition (Normal Terms)

We call a term t normal if and only if t does not contain a redex as a subterm.

Definition (Normalizing Type Theory)

We call a type theory normalizing if and only if for all terms $\Gamma \vdash t : \phi$, there exists a term n, such that, $t \rightsquigarrow^* n$ and n is normal.

- What is normalization?
- Normalization is a powerful property, implies logical consistency for proof systems based on lambda calculus.
- Proving normalization is difficult and often requires very complex arguments.

An Interpretation of Types and Type Soundness

- The interpretation of types, as we have defined them, are sets of terms with a common type.
  - Provides a semantics of the type system.
- Soundness of typing: if a term t is typeable with type ϕ, then t is in the interpretation of type ϕ.
  - Bridges gap between the syntax and the semantics.

Stratified System F

- A modified version of Girard's System F, created by Daniel Leivant.
- Substantially, weaker then System F.
  - The entire set of the primitive recursive functions are definable in System F.
  - Only a proper subset of the primitive recursive functions are definable in Stratified System F.
- The difference is that the types in Stratified System F are stratified into levels.
- The Language.

$$\begin{array}{lcl} \text{Terms: } t & := & x \mid \lambda x : \phi.t \mid tt \mid \Lambda X : K.t \mid t[\phi] \\ \text{Types: } \phi & := & X \mid \phi \to \phi \mid \forall X : K.\phi \\ \text{Kinds: } K & := & *_0 \mid *_1 \mid \ldots \end{array}$$

- The Reduction Rules (Full β-Reduction).

$$\begin{array}{lcl} (\Lambda X : *_p.t)[\phi] & \rightsquigarrow & [\phi/X]t \\ (\lambda x : \phi.t)t' & \rightsquigarrow & [t'/x]t \end{array}$$

Stratified System F - Type quantification

- Kind-checking rule.

$$\frac{\Gamma, X : *_q \vdash \phi : *_p}{\Gamma \vdash \forall X : *_q.\phi : *_{max(p,q)+1}}$$

- Type-checking rules.

$$\frac{\Gamma, X : *_p \vdash t : \phi}{\Gamma \vdash \Lambda X : *_p.t : \forall X : *_p.\phi} \qquad \frac{\Gamma \vdash t : \forall X : *_l.\phi_1 \quad \Gamma \vdash \phi_2 : *_l}{\Gamma \vdash t[\phi_2] : [\phi_2/X]\phi_1}$$

The Interpretation of Types

- In the following figure we define the interpretation of kindable types for normal terms.

- We extend this definition to non-normal terms as follows. A non-normal term t is in the interpretation of a type ϕ if and only if $t \rightsquigarrow^! n$ and $n \in [\![\phi]\!]_\Gamma$.

$$
\begin{array}{lcl}
x \in [\![\phi]\!]_\Gamma & \Leftrightarrow & \Gamma(x) = \phi \\
n_1 n_2 \in [\![\phi]\!]_\Gamma & \Leftrightarrow & \exists \phi'.n_1 \in [\![\phi' \to \phi]\!]_\Gamma \wedge n_2 \in [\![\phi']\!]_\Gamma \\
\lambda x : \phi_1.n \in [\![\phi]\!]_\Gamma & \Leftrightarrow & \exists \phi_2.\phi = \phi_1 \to \phi_2 \wedge n \in [\![\phi_2]\!]_{\Gamma, x:\phi_1} \\
\Lambda X : *_p.n \in [\![\phi]\!]_\Gamma & \Leftrightarrow & \exists \phi'.\phi = \forall X : *_p.\phi' \wedge n \in [\![\phi']\!]_{\Gamma, X:*_p} \\
n[\phi'] \in [\![\phi]\!]_\Gamma & \Leftrightarrow & \exists \phi'', l.\phi = [\phi'/X]\phi'' \wedge \Gamma \vdash \phi' : *_l \wedge n \in [\![\forall X : *_l.\phi'']\!]_\Gamma
\end{array}
$$

Figure: Interpretation of Kindable Types for Normal Terms

Well-Foundness of Ordering on Types

Definition (well-founded ordering on types)
The ordering $>_\Gamma$ is defined as the least relation satisfying the universal closures of the following formulas:

$$
\begin{array}{lll}
\phi_1 \to \phi_2 & >_\Gamma & \phi_1 \\
\phi_1 \to \phi_2 & >_\Gamma & \phi_2 \\
\forall X : *_l.\phi & >_\Gamma & [\phi'/X]\phi \text{ where } \Gamma \vdash \phi' : *_l.
\end{array}
$$

Theorem ($>_\Gamma$ is well-founded)
The ordering $>_\Gamma$ is well-founded on types ϕ such that $\Gamma \vdash \phi : *_l$ for some l.

Substitution for Interpretation of Types

- The proof of the following lemma requires a number of insights.
  - This proof is done by induction on the measure (ϕ, n') in lexicographic combination of $>_{\Gamma,\Gamma'}$ and the strict subexpression ordering.
  - The central idea of hereditary substitution.

Lemma (Substitution for the Interpretation of Types)
If $n' \in [\![\phi']\!]_{\Gamma,x:\phi,\Gamma'}$, $n \in [\![\phi]\!]_{\Gamma}$, then $[n/x]n' \rightsquigarrow^! \hat{n} \in [\![\phi']\!]_{\Gamma,\Gamma'}$ and if n' is not a λ-abstraction or a Λ-abstraction and $\hat{n}$ is, then $\phi \geq_{\Gamma,\Gamma'} \phi'$.

Concluding Normalization

- We are now in a position to conclude normalization for stratified system F.

Theorem (Type Soundness for the Interpretation of Types)
If $\Gamma \vdash t : \phi$ *then* $t \in [\![\phi]\!]_{\Gamma}$.

Closing Remarks

- Proof by hereditary substitution.
- Normalization proof for stratified system F.
- Thank you Prof. Stump.
- Thank you for listening!
- All information in this presentation and more can be found in our paper.

  Hereditary Substitution for Stratified System F. Harley D. Eades III and Aaron Stump. PSTT. Edinburgh. 2010.

  - Download: http://wwww.cs.uiowa.edu/∼heades/papers.html.