Weak Normalization for the Simply-Typed Lambda-Calculus in Twelf

Andreas Abel

ProgLog, Chalmers
Göteborg, Sweden
June 9, 2004

Work supported by: GKLI (DFG), TYPES & APPSEM-II (EU), CoVer (SFF)

Twelf

- Logical framework based on the Edinburgh LF (dependently-typed \( \lambda \)-calculus)
- Propositions-as-types, derivations-as-objects
- Higher-order abstract syntax
- Terms: abstraction, application, user-def. constants
- Terms considered up to \( \beta\eta \)-equality
- No user-def. reduction rules: all functions parametric
- Types: dependent function, user-def. type family constants
- Logic programming through proof search
Simply Typed λ-Calculus (STL)

- Syntax.

\[ r, s, t, u ::= x \mid \lambda x.t \mid rs \] untyped terms
\[ A, B, C ::= \ast \mid A \rightarrow B \] simple types
\[ \Gamma ::= \diamond \mid \Gamma, x:A \] contexts

- Type assignment \( \Gamma \vdash t : A \).

\[ (x:A) \in \Gamma \quad \frac{\Gamma \vdash x : A}{\Gamma \vdash \lambda x.t : A \rightarrow B} \quad \text{of}_{\text{lam}} \]
\[ \frac{\Gamma \vdash r : A \rightarrow B \quad \Gamma \vdash s : A}{\Gamma \vdash rs : B} \quad \text{of}_{\text{app}} \]

- Weak head reduction \( t \rightarrow_w t' \).

\[ (\lambda x.t)s \rightarrow_w [s/x]t \quad \text{beta} \]
\[ r \rightarrow_w r' \quad rs \rightarrow_w r's \quad \text{appl} \]

Representation of Syntactic Objects in Twelf

- Representation of simple types.

\[ \text{ty} : \text{type.} \]
\[ \ast : \text{ty.} \]
\[ => : \text{ty -> ty -> ty.} \]

- Representation of \( \lambda \)-terms.

\[ \text{tm} : \text{type.} \]
\[ \text{lam} : (\text{tm -> tm}) \rightarrow \text{tm.} \]
\[ \text{app} : \text{tm -> tm -> tm.} \]
• HOAS = represent object variables by framework variables.

\[
\text{twice} = \lambda f : \text{tm} . \lambda x : \text{tm} . \text{app } f (\text{app } f x).
\]

Slide 5

Representation of Judgements without Hypotheses

• Representation of weak head reduction.

\[
\rightarrow^w : \text{tm} \rightarrow \text{tm} \rightarrow \text{type}.
\]

Slide 6

\[
\begin{align*}
\text{beta} & : \text{app } (\lambda \text{am } T) \ S \rightarrow^w T \ S. \\
\text{apl} & : \ R \rightarrow^w \ R' \rightarrow \text{app } R \ S \rightarrow^w \text{app } R' \ S.
\end{align*}
\]

• Substitution in object theory is application of the framework.
Representation of Judgements with Hypotheses

- Representation of typing relation: Think in natural deduction trees.

\[
\frac{\ \ x \ \ }{A \ \ } \quad \frac{A \to B \ \ A \ \ }{B \ \ } \quad \frac{A \to B}{A \to B} 
\]

\[\text{of}_\text{app}\]

\[\text{of}_\text{lam}\]

- Typing assumption is represented as hypothetical judgement.

\[
of : \text{tm} \to \text{ty} \to \text{type}.
\]

\[
of_\text{lam} : (\{x:\text{tm}\} \ x \ of \ A \to (T \ x) \ of \ B) \to (\text{lam} \ [x:\text{tm}] \ T \ x) \ of \ (A \Rightarrow B).
\]

\[
of_\text{app} : R \ of \ (A \Rightarrow B) \to S \ of \ A \to (\text{app} \ R \ S) \ of \ B.
\]

Weak Head Reduction is Closed under Substitution

- Lemma: If \( t \longrightarrow_w t' \) then \([u/y]t \longrightarrow_w [u/y]t'\).
- Proof: By induction on the derivation of \( t \longrightarrow_w t' \).

  - Case \( (\lambda x.t) \ s \longrightarrow_w [s/x]t \). W.l.o.g. \( x \neq y \) and \( x \) not free in \( u \). Then,

\[
[u/y)((\lambda x.t) \ s) \quad = \quad (\lambda x.[u/y]t) \ [u/y]s
\]

\[
\longrightarrow_w \quad [[u/y][s/x][u/y]t \quad = \quad [u/y][s/x]t.
\]

  - Case \( r \ s \longrightarrow_w r' \ s \) with \( r \longrightarrow_w r' \). By ind. hyp.,

\[
[u/y]r \longrightarrow_w [u/y]r'. \quad \text{Hence,}
\]

\[
[u/y](r \ s) \quad = \quad ([u/y][u/y]s)
\]

\[
\longrightarrow_w \quad ([u/y][u/y]r') \ [u/y]s \quad = \quad [u/y](r' \ s)
\]

\[\square\]
Representation of Theorems and Proofs

- A theorem is represented as a functional relation.

\[
\text{subst}_\text{red} : \{U : \text{tm}\} \rightarrow \{y : \text{tm}\} \rightarrow T y \rightarrow T' y
\rightarrow T U \rightarrow T' U \rightarrow \text{type.}
\%
\text{mode subst}_\text{red} +U +P -P'.
\]

- Its proof is represented as a logic program which implements the relation.

\[
\begin{align*}
\text{subst}_\text{red}_\beta & : \text{subst}_\text{red} U ([y] \beta) \beta. \\
\text{subst}_\text{red}_\text{appl} & : \text{subst}_\text{red} U ([y] \text{appl} (P y)) (\text{appl} P') \\
& \quad \leftarrow \text{subst}_\text{red} U P P'. \\
\%	ext{terminates P (subst} _\text{red} _P_).
\end{align*}
\]

- Function must be total to represent a valid proof.
- This requires termination and coverage of all possible inputs.
A Formalized Proof of Weak Normalization for the STL

- Structure of a normalization proof:
  1. Define a relation $t \Downarrow A$ which is closed under application.
  2. Show: If $t : A$ then $t \Downarrow A$.
  3. Show: If $t \Downarrow A$ then $t$ is normalizing.

Slide 11

- Tait and crowd: $t \Downarrow A$ is a logical relation (semantical).
- Joachimski and Matthes (2004): $t \Downarrow A$ is a finitary inductive definition.

Inductive Characterization of Weakly Normalizing Terms

- $\Gamma \vdash t \Downarrow A$: $t$ is weakly normalizing of type $A$.
- $\Gamma \vdash t \downarrow^x A$: $t$ is wn and neutral of type $A$.
- Rules:

\[
\frac{(x:A) \in \Gamma}{\Gamma \vdash x \downarrow^x A} \quad \frac{\Gamma \vdash r \downarrow^x A \rightarrow B \quad \Gamma \vdash s \downarrow^x B}{\Gamma \vdash r \cdot s \Downarrow B} \quad \text{wne_app}
\]

\[
\frac{\Gamma \vdash r \downarrow^x B}{\Gamma \vdash r \Downarrow B} \quad \text{wn_ne}
\]

\[
\frac{\Gamma, x:A \vdash t \Downarrow B}{\Gamma \vdash \lambda x.t \Downarrow A \rightarrow B} \quad \text{wn_lam} \quad \frac{r \rightarrow^\omega r' \quad \Gamma \vdash r' \Downarrow A}{\Gamma \vdash r \Downarrow A} \quad \text{wn_exp}
\]
**Difficult: Closure under Application**

- Lemma: Let $D :: \Gamma \vdash s \Downarrow A$.
  1. If $E :: \Gamma \vdash r \Downarrow A \rightarrow C$ then $\Gamma \vdash r s \Downarrow C$.
  2. If $E :: \Gamma, x : A \vdash t \Downarrow C$, then $\Gamma \vdash [s/x]t \Downarrow C$.
  3. If $E :: \Gamma, x : A \vdash t \Downarrow C$, then $\Gamma \vdash [s/x]t \Downarrow C$
    and $C$ is a subexpression of $A$.
  4. If $E :: \Gamma, x : A \vdash t \Downarrow^y C$ with $x \neq y$, then $\Gamma \vdash [s/x]t \Downarrow^y C$.

- Proof: Simultaneously by main induction on type $A$ (for part 3) and side induction on the derivation $E$.
- Similar to Girard, Lafont and Taylor (1989): Lexicographic induction on highest degree (=type) of a redex and the number of redexes of highest degree.

**Closure under Application and Substitution in Twelf**

- Representation of lemma as 4 type families.
- “$C$ is a subexpression of $A$” expressed by \texttt{\%reduces C \leq A}.
- Mutual lexicographic termination order.
Soundness of Inductive Characterization

- Simple induction: $t \downarrow A$ for every typed term $t : A$.
- Lemma (Soundness): If $t \downarrow A$ then $t \rightarrow^{\ast} v$ for some $v$.
- Requires characterization of valued and properties of reduction.
- Technical, but well understood.

Tait-Style Proofs in Twelf?

- Heart of Tait’s proof is the rule:

\[
\forall s. \quad s \downarrow A \Rightarrow r \downarrow B \\
\hline
r \downarrow A \rightarrow B
\]

- Literal encoding in Twelf...

\[
\{S:tm\} \quad \text{wn} \quad S \ A \rightarrow \text{wn} \quad (\text{app} \quad R \ S \ B) \rightarrow \text{wn} \quad R \ (A \Rightarrow B).
\]

- ...means something else:

if for a fresh term $S$ for which we assume $\text{wn} \ S \ A$ it holds
that $\text{wn} \ (\text{app} \ R \ S) \ B$, then $\text{wn} \ R \ (A \Rightarrow B)$.

- Problem: Tait’s infinitary premise is not expressible.
Strong Normalization in Twelf?

- Classical definition of \textit{strongly normalizing}: no infinite reduction sequences.
- No good in a constructive setting.
- Inductive definition of \textit{strongly normalizing}: wellfounded part of reduction relation.

\[
\forall t', t \rightarrow t' \Rightarrow \text{sn } t' \\
\text{sn } t.
\]

- Suffers likewise from an infinitary premise.

Conclusion

- Normalization for a proof-theoretically weak object theory directly implementable in Twelf.
- Limits for normalization proofs: expressiveness of Twelf, termination checker.

Slide 18

- Conjecture 1: Infinitary premises not expressible in Twelf.
- Conjecture 2: Strong normalization not expressible in Twelf.
- Conjecture 3: Proof-theoretical strength of Twelf bounded by arithmetic.
Related Work

- Berghofer and Nipkow: Joachimski and Matthes’ proof in Isabelle.