Purifying Natural Deduction Using Sequent Calculus

Aaron Stump

Computational Logic Center
CS, The University of Iowa

Funding from NSF CAREER.
Verified Programming

Thesis
The ability to state and prove properties of code is the crucial missing technology in the evolution of software.

- Stronger guarantees $\Rightarrow$ less monitoring $\Rightarrow$ higher performance.
- Ability to trust software opens up new applications.
- Confirmed quality helps open source, app stores, etc.
- Verification is a tool we don’t have.
The **GURU** Verified Programming Language

- Functional language
- Dependently typed programs
- General recursion
- Notation for theorems, proofs about programs
- Unaliased mutable state
- Resource management layer
- Type/Proof-checker, compiler to C
- No concurrency
- Aliasing for mutable state in progress

[www.guru-lang.org](http://www.guru-lang.org)
Practical Proof Theory

- How to prove your logic is consistent?
- Basic strategy:
  1. Identify subset of proofs which obviously are ok.
  2. Define rewrite rules to transform any proof to one in the ok form.
  3. Prove rules are (strongly or weakly) normalizing.
- By Curry-Howard isomorphism:
  - Proofs are \( \lambda \)-terms.
  - Proof normalization is \( \beta \)-reduction.
- Reducibility proofs (logical relations) are powerful, elegant.
- Do not work well with disjunctions, existentials.
Reducibility for Conjunction

Proof terms $p ::= (p_1, p_2) \mid p.1 \mid p.2$

$\Gamma \vdash p_1 : \phi_1 \quad \Gamma \vdash p_2 : \phi_2$

$\Gamma \vdash (p_1, p_2) : \phi_1 \land \phi_2 \quad \land I$

$\Gamma \vdash p : \phi_1 \land \phi_2 \quad i \in \{1, 2\}$

$\Gamma \vdash p.i : \phi_i \quad \land E$

Reducibility is “hereditary normalization”, defined by eliminations.

- $Red_\phi$ is set of reducible terms of type $\phi$.
- $p \in Red_b \Leftrightarrow SN(p)$, for base types $b$.
- $p \in Red_{\phi_1 \land \phi_2} \Leftrightarrow p.1 \in Red_{\phi_1}$ and $p.2 \in Red_{\phi_2}$.
- $p \in Red_{\phi_1 \rightarrow \phi_2} \Leftrightarrow \forall p' \in Red_{\phi_1}, (p \; p') \in Red_{\phi_2}$
What Goes Wrong with Disjunction

Proof terms $p ::= \langle 1, p \rangle \mid \langle 2, p \rangle \mid \text{case}(p)(x.p_1, x.p_2)$

\[
\Gamma \vdash p : \phi_i \quad i \in \{1, 2\} \quad \because I
\]

\[
\Gamma \vdash \langle i, p \rangle : \phi_1 \land \phi_2
\]

\[
\Gamma \vdash p : \phi_1 \lor \phi_2 \quad \Gamma, x : \phi_1 \vdash p_1 : \psi \quad \Gamma, x : \phi_2 \vdash p_2 : \psi \quad \therefore E
\]

\[
\Gamma \vdash \text{case}(p)(x.p_1, x.p_2) : \psi
\]

Attempt to define reducibility fails:

$p \in \text{Red}_{\phi_1 \lor \phi_2} \iff \forall \psi, \ p_1, p_2 \in \text{Red}_\psi, \ \text{case}(p)(x.p_1, x.p_2) \in \text{Red}_\psi$

Not legal to appeal to $\text{Red}_\psi$. 
A Way Forward

- Problem with ∨E:
  - to use $p : \phi$, need $p' : \psi$, where $\psi$ unrelated to $\phi$.
  - breaks definition of reducibility.

- But compare sequent calculus rules:

\[
\frac{\Gamma, \phi_1 \vdash \psi \quad \Gamma, \phi_2 \vdash \psi}{\Gamma, \phi_1 \lor \phi_2 \vdash \psi} \quad \text{L}\lor \\
\frac{\Gamma, \phi_1, \phi_2 \vdash \psi}{\Gamma, \phi_1 \land \phi_2 \vdash \psi} \quad \text{L}\land
\]

- Term assignment for sequent calculus is strange.

\[
\frac{\Gamma, y : \phi_1, z : \phi_2 \vdash p : \psi}{\Gamma, x : \phi_1 \land \phi_2 \vdash [x.1/y, x.2/z]p : \psi} \quad \text{L}\land
\]

- Limited by old view of “natural” deduction.
A Direct Term Assignment

- Left rules correspond to eliminations.
- Why insist that the context $\Gamma$ holds just variables?
- Proposal:
  - Assign terms to sequent calculus directly.
  - Devise new terms for $\lor E$, $\exists E$.
  - Allow $\Gamma$ to hold terms.
Elimination Rules

\[ \Gamma, p.1 : \phi_1, p.2 : \phi_2 \vdash p' : \psi \]

\[ \frac{}{\Gamma, p : \phi_1 \land \phi_2 \vdash p' : \psi} \quad \text{L}^\land \]

\[ \Gamma, (p a) : [a/x] \phi \vdash p' : \psi \]

\[ \frac{}{\Gamma, p : \forall x. \phi \vdash p' : \psi} \quad \text{L}^\forall \]

\[ p : \phi \vdash p : \phi \quad \text{Ax} \]

\[ \frac{}{\Gamma \vdash p' : \psi} \quad \text{LW} \]

\[ \frac{}{\Gamma, p : \phi \vdash [p]p' : \psi} \quad \text{LC} \]
We have separated logical terms \((p.(i))\) from structural \((p_1 \parallel p_2)\).

Logical terms have \(\beta\)-reductions:

\[
\begin{align*}
(p_1, p_2).i \rightsquigarrow p_i \\
\langle i, p \rangle.(i) \rightsquigarrow t \\
\langle i, p \rangle.(3 - i) \rightsquigarrow abort
\end{align*}
\]

Structural terms have commuting conversions:

\[
\begin{align*}
(p_1 \parallel p_2).i \rightsquigarrow (p_1.i) \parallel (p_2.i) \\
abort \parallel p \rightsquigarrow p
\end{align*}
\]

Simple unsound typing rules suffice for reducibility.

\[
\frac{\Gamma \vdash p : \phi_1 \lor \phi_2}{\Gamma \vdash p.i : \phi_i} \lor E
\]
Towards Pure Natural Deduction

- Define natural deduction rules.
  \[ S ::= \Gamma \vdash \Delta \mid S \parallel S \]
  \[ \Delta ::= t_1 : \phi_1, \ldots, t_n : \phi_n \]

- Example derivation:
  \[
  \begin{align*}
    u : \phi_1 \lor \phi_2 \\
    u.(1) : \phi_1 \parallel u.(2) : \phi_2 \\
    \vdots \quad \vdots \\
    p_1 : \psi \parallel p_2 : \psi \\
    p_1 \parallel p_2 : \psi
  \end{align*}
  \]

- Completeness proved (open derivations \( S \rightharpoonup S' \)).

Goal: Pure Natural Deduction.
  - All rules are either direct logical rules or structural.
  - Consistency proved by reducibility.
  - Decidable equational theory, including commuting conversions.