System F and the Proof theory of Second-Order Arithmetic

Patrick Nicodemus

May 4, 2019

Minimal Logic ($\to$) - Natural Deduction

Axioms:
introduction and
elimination

$$\dfrac{\begin{matrix}\mathcal{D}_1 \\ A\end{matrix} \qquad \begin{matrix}\mathcal{D}_2 \\ A \to B\end{matrix}}{B} \to \mathrm{E}$$

$$\dfrac{\begin{matrix}[A]^u \\ \mathcal{D} \\ B\end{matrix}}{A \to B} \to \mathrm{I}, u$$

Sample proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \to (B \to C)^u \qquad A^w}{B \to C} \qquad \dfrac{A \to B^v \qquad A^w}{B}}{C}}{A \to C} \to \mathrm{I}, w}{(A \to B) \to (A \to C)} \to \mathrm{I}, v}{A \to (B \to C) \to ((A \to B) \to (A \to C))} \to \mathrm{I}, u$$

Minimal Logic ($\to$) - Curry-Howard Correspondence

Axioms: introduction and elimination

$$\dfrac{\begin{matrix}\mathcal{D}_1 \\ A\end{matrix} \qquad \begin{matrix}\mathcal{D}_2 \\ A \to B\end{matrix}}{B} \to \mathsf{E}$$

$$\dfrac{\begin{matrix}[A]^u \\ \mathcal{D} \\ B\end{matrix}}{A \to B} \to \mathsf{I}, u$$

Typing rules for the simply typed lambda calculus: application and abstraction

$$\frac{\Gamma \vdash N : A \qquad \Gamma \vdash M : A \to B}{\Gamma \vdash MN : B}$$

$$\frac{\Gamma, u : A \vdash M : B}{\Gamma \vdash \lambda u : A.M : A \to B}$$

Lambda Calculus Terms Code Proofs

Sample proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \to (B \to C)^u \qquad A^w}{B \to C} \qquad \dfrac{A \to B^v \qquad A^w}{B}}{C}}{A \to C}\;\to\mathrm{I}, w}{(A \to B) \to (A \to C)}\;\to\mathrm{I}, v}{A \to (B \to C) \to ((A \to B) \to (A \to C))}\;\to\mathrm{I}, u$$

In λ-calculus:

$$\dfrac{\dfrac{\dfrac{\dfrac{u : A \to (B \to C), w : A \vdash uw : B \to C \qquad v : A \to B, w : A \vdash vw : B}{u : A \to (B \to C), w : A, v : A \to B \vdash uw(vw) : C}}{u : A \to (B \to C), v : A \to B \vdash \lambda w.uw(vw) : A \to C}}{u : A \to (B \to C) \vdash \lambda w.\lambda v.uw(vw) : (A \to B) \to (A \to C)}}{\vdash \lambda u.\lambda v.\lambda w.uw(vw) : (A \to (B \to C)) \to (A \to B) \to (A \to C)}$$

Normal Proof Trees

$$\dfrac{\dfrac{\begin{matrix}[A]^u \\ \mathcal{D} \\ B\end{matrix}}{A \to B}\,\to\mathrm{I},\ u \qquad \begin{matrix}\mathcal{D}' \\ A\end{matrix}}{A}$$

vs.

$$\begin{matrix}\mathcal{D}' \\ A \\ \mathcal{D} \\ B\end{matrix}$$

Curry-Howard isn't just *static*- it's *dynamic*

$$\dfrac{\dfrac{\begin{array}{c}[A]^u\\ \mathcal{D}\\ B\end{array}}{A \to B}\ {\to}\mathrm{I},\ u \qquad \begin{array}{c}\mathcal{D}'\\ A\end{array}}{A}$$

vs.

$$\begin{array}{c}\mathcal{D}'\\ A\\ \mathcal{D}\\ B\end{array}$$

$$\dfrac{\dfrac{u : A \vdash M : B}{\vdash \lambda u : A.M : A \to B} \qquad \vdash N : A}{\vdash (\lambda u : A.M)N : B}$$

vs.

$$\dfrac{}{\vdash M[u/N] : B}$$

"Normalization" of a proof corresponds to β-reduction of λ-terms.

Consistency: Does B have a proof?
Does B have a *normal* proof?

Second Order Logic

Add relational variables R to predicate logic; superscripts R^n denote arity. $\forall^2, \exists^2$ quantify over relational variables.

$$\frac{A}{\forall^2 Y^n, A[X^n/Y^n]}\ \forall^2 I$$

$$\frac{\forall^2 X^n, A}{A[X^n/\lambda x_1, \ldots, x_n.B]}\ \forall^2 E$$

$$\frac{A[X^n/\lambda x_1, \ldots, x_n.B]}{\exists^2 X^n, A}\ \exists^2 I$$

$$\frac{\begin{matrix}[A[X^n/Y^n]]^u \\ \mathcal{D} \\ C\end{matrix} \qquad \exists^2 X^n, A}{C}\ \exists^2 E$$

Redundancy

These redefinitions code the proof rules into the logic.

- $\bot := \forall^2 X^0 . X^0$
- $A \wedge B := \forall^2 X^0, (A \to (B \to X)) \to X$
- $A \vee B := \forall^2 X^0, (A \to X) \to ((B \to X) \to X)$
- $\exists y, A(y) := \forall^2 X^0 .(\forall y.(A(y) \to X) \to X)$
- $\exists^2 Y^n, A. := \forall^2 X^0 (\forall Y^n .(A \to X) \to X)$

So without loss of generality, sentences of $Ni^2$are:

- atomic formulas - relations on terms, propositional constants: $t_1 = t_2, R(t_1, \ldots, t_n), X(t_1, \ldots, t_n)$
- inductively, those constructed out of $\to, \forall, \forall^2$

Notions of "proof detour" in the language $(\to, \forall, \forall^2)Ni^2$

$$\dfrac{\dfrac{\dfrac{\mathcal{D}}{A}}{\forall y, A}}{A[y/t]}$$

vs.

$$\dfrac{\mathcal{D}[y/t]}{A[y/t]}$$

$$\dfrac{\dfrac{\dfrac{\mathcal{D}}{A}}{\forall Y^n, A}}{A[X^n/\lambda\vec{x}.B]}$$

vs.

$$\dfrac{\mathcal{D}[X^n/\lambda\vec{x}.B]}{A[X^n/\lambda\vec{x}.B]}$$

Second order propositional logic

- We restrict to a subsystem Ip^2 of Ni^2 - forget about constants, terms, n-ary relations with $n \geq 1$.
- Only propositional constants and propositional variables.
- $\forall^2 X \exists^2 Y, (X \to Y) \to ((X \to Y) \to Y)$

$\lambda 2$ aka System F

Types:

- A family of propositional constants
- A family of propositional variables $X, Y, \ldots$
- If A, B are types, $A \to B$ is a type
- If A is a type, and X is a type variable, $\forall^2 X.A$ is a type (sometimes $\prod X.A$)

Example: $\forall^2 X.X \to ((X \to X) \to X)$ is a type

Terms:

- For each type A one has type variables, $x : A$
- If $t : B$, and t depends on x of type A, then $\lambda x.t : A \to B$
- If $t : B$ and B depends on type variable X, then $\Lambda X.t : \forall^2 X.A$

Example: $\Lambda X.\lambda x : X.\lambda f : X \to X.fffx$ is a term of the type above

Parametric Polymorphism

- ListInt, ListBool, ListWidget,...
- These should all have the same basic functionality: append new element, concatenate element, etc.
- They should all have essentially the same code, independent of the type of object in the list.
- Solution: type variables (=polymorphism). Define List α, where α is a type variable.
- Function signatures - this.append(α x) ...

Reduction Laws

- $(\lambda x : A.M)N \xrightarrow{\beta} M[x/N]$
- $(\Lambda X.t)B = t[X/B]$

Computational Content

- Define $\mathbf{N}_X = X \to ((X \to X) \to X)$
- for $n \in \mathbb{N}$, let $\overline{n}_X = \lambda x : X.\lambda f : X \to X.f^n(x)$
- Theorem: The Church numerals are the only terms in normal form of type $\mathbf{N}_X$.
- Define $\mathbf{N} = \forall^2 X.N_X$
- define $\overline{n} = \Lambda X.\overline{n}_X$.
- What functions are computable?

Intuitionistic second-order arithmetic ("analysis") $\mathbf{HA}^2$

- Answer: iff it is a recursive function provably total in PA^2.
- LEM is "equiconsistent"
- Intuitionistic systems agree with classical ones on an interesting class of theorems - say, proving recursive functions are total.
-