Eight Topics This Talk Is Not About

Chad E. Brown

Czech Technical University in Prague

April 2022

Outline

Higher-Order Logic

Alonzo Church

Peter B. Andrews

- Church created the simply typed λ-calculus version of higher-order logic in 1940.
- Andrews pioneered research in automated theorem proving in higher-order logic for many decades. (TPS)

TPS - Connection Method

Journal of Automated Reasoning **5**: 257–291, 1989.
© 1989 *Kluwer Academic Publishers. Printed in the Netherlands.*

On Connections and Higher-Order Logic

PETER B. ANDREWS
Mathematics Department, Carnegie-Mellon University, Pittsburgh, PA 15213, U.S.A.

(Received: 13 December 1988)

The vpform of this is:

$$\left[\begin{array}{c} \left[\begin{array}{c} P \\ \sim Q \end{array}\right] \vee \left[\begin{array}{c} Q \\ \sim P \end{array}\right] \vee R \\ \sim R \vee \left[\begin{array}{c} \sim P \vee Q \\ \sim Q \vee P \end{array}\right] \\ \left[\left[\begin{array}{c} P \\ \left[\begin{array}{c} Q \\ \sim R \end{array}\right] \vee \left[\begin{array}{c} R \\ \sim Q \end{array}\right] \end{array}\right] \vee \left[\begin{array}{c} \sim Q \vee R \\ \sim R \vee Q \\ \sim P \end{array}\right] \right] \end{array}\right]$$

"vpform" = "vertical path form"

TPS - Connection Method

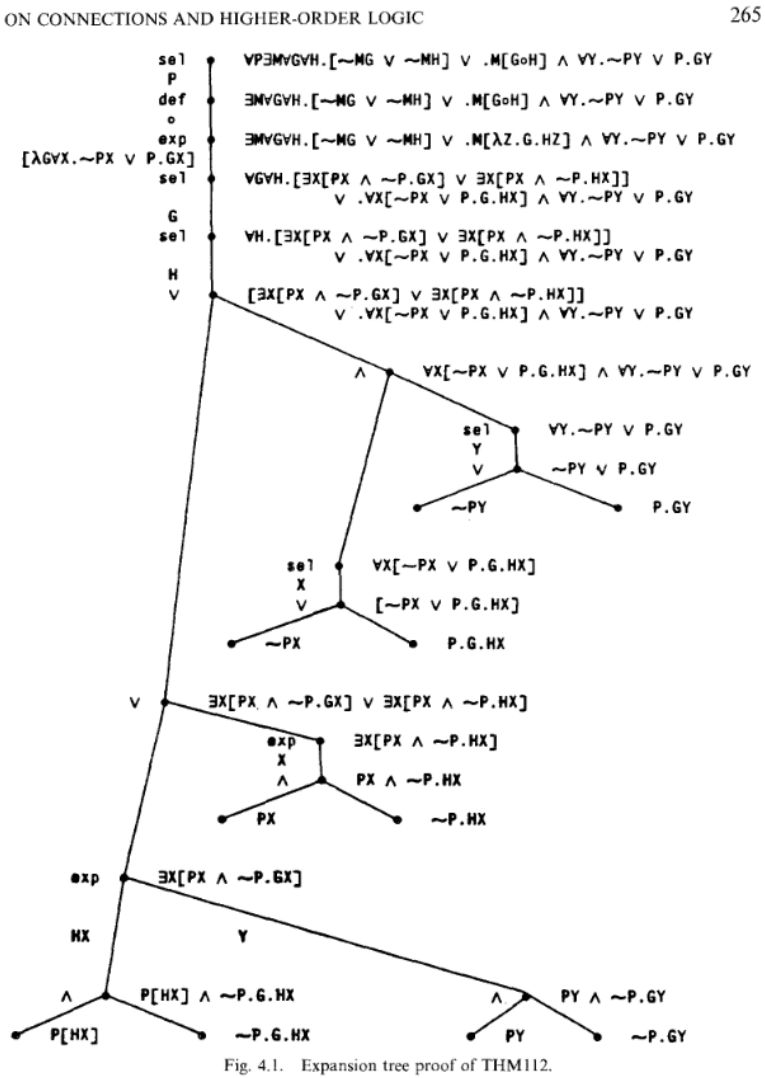


Fig. 4.1. Expansion tree proof of THM112.

TPS - Primitive Substitutions

$$\lambda X_{o\alpha\xi}\lambda Y_{o\beta}\lambda Z_{\alpha}.\sim R^{1}_{o\alpha(o\beta)(o\alpha\xi)}XYZ$$

$$\lambda X_{o\alpha\xi}\lambda Y_{o\beta}\lambda Z_{\alpha}.R^{2}_{o\alpha(o\beta)(o\alpha\zeta)}XYZ \wedge R^{3}_{o\alpha(o\beta)(o\alpha\xi)}XYZ$$

$$\lambda X_{o\alpha\xi}\lambda Y_{o\beta}\lambda Z_{\alpha}.R^{4}_{o\alpha(o\beta)(o\alpha\zeta)}XYZ \vee R^{5}_{o\alpha(o\beta)(o\alpha\zeta)}XYZ$$

$$\lambda X_{o\alpha\zeta}\lambda Y_{o\beta}\lambda Z_{\alpha}\exists W_{\varphi}R^{6}_{o\varphi\alpha(o\beta)(o\alpha\xi)}XYZW$$

$$\lambda X_{o\alpha\zeta}\lambda Y_{o\beta}\lambda Z_{\alpha}\forall W_{\varphi}R^{7}_{o\varphi\alpha(o\beta)(o\alpha\xi)}XYZW$$

$$\lambda X_{o\alpha\zeta}\lambda Y_{o\beta}\lambda Z_{\alpha}Y.R^{8}_{\beta\alpha(o\beta)(o\alpha\zeta)}XYZ$$

$$\lambda X_{o\alpha\zeta}\lambda Y_{o\beta}\lambda Z_{\alpha}X[R^{9}_{\zeta\alpha(o\beta)(o\alpha\zeta)}XYZ].R^{10}_{\alpha\alpha(o\beta)(o\alpha\zeta)}XYZ$$

$$\lambda X_{o\alpha\zeta}\lambda Y_{o\beta}X.R^{11}_{\zeta(o\beta)(o\alpha\xi)}XY$$

Fig. 10.1. Primitive substitutions for $R_{o\alpha(o\beta)(o\alpha\xi)}$.

Higher-Order Logic

- Simple Types (no evil type variables)
  - ι (individuals)
  - o (propositions/booleans/truth values)
  - $\alpha \to \beta$ (function types)
- Simply typed λ-terms with some logic:
  - Typed Variables x
  - Typed Constants c
  - Applications $s\ t$
  - Abstractions $\lambda x.s$
  - Implications $s \to t$
  - Universal quantifiers $\forall x.s$
  - Optional choice: $\varepsilon x.s$
- $\beta\eta$-equivalence (unification is hard)
- Propositions are terms of type o.
- Some propositions are provable.
- Some propositions are valid in all Henkin models.

Outline

Satallax

About 10 years ago I worked on a higher-order theorem prover Satallax. It won the TH0 division of CASC for most years of the 2010s.

- Complete tableau calculus (in the Hintikka, Beth, Smullyan, Fitting sense) for higher-order logic with a choice operator.
- Instantiation based – used *no* unification in the basic calculus.
- Had interesting restriction on quantifiers at base types: only instantiate with *discriminating* terms.
- Able to reason with equations without rewriting deeply inside terms.
- People still think I work on this, though I haven't in years.

Satallax

$$\forall x.f\ x = x$$

$$p\ (f\ (f\ a))$$

$$\neg p\ a$$

Satallax

$$\forall x.f\ x = x$$

$$p\ (f\ (f\ a))$$

$$\neg p\ a$$

$$f\ (f\ a) \neq a$$

Satallax

$$\forall x.f\ x = x$$

$$p\ (f\ (f\ a))$$

$$\neg p\ a$$

$$f\ (f\ a) \neq a$$

$$f\ a = a$$

Satallax

$$\forall x.f\ x = x$$

$$p\ (f\ (f\ a))$$

$$\neg p\ a$$

$$f\ (f\ a) \neq a$$

$$f\ a = a$$

$$\begin{array}{l|l} f\ (f\ a) \neq f\ a & a \neq a \\ f\ a \neq a & \end{array}$$

Outline

Higher-Order Logic

Satallax

Lash

Set Theory

Proof Terms

Egal

Proofgold

Megalodon

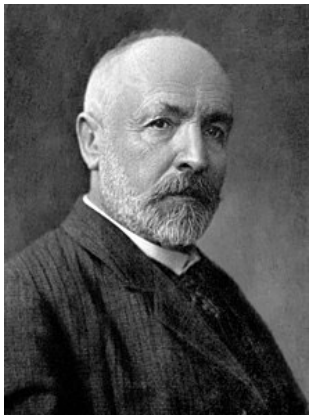
Main Topic

Lash

- Lash is a new implementation of Satallax's calculus.
- Cezary Kaliszyk reimplemented terms/$\beta\eta$-normalization in C
- ...with perfect sharing.
- He also reimplemented important data structures like priority queues in C.
- "Better" than Satallax already, but it's still early days.

Outline

Set Theory



Georg Cantor

Ernst Zermelo

No one shall expel us from the paradise that Cantor has created. - David Hilbert

Set Theory

- Popular foundation for mathematics
- Natural choice for formalizers of mathematics
- The Mizar people knew this in the 1970s already.
- ZFC (and TG) are not finitely axiomatizable in first-order
- ...but higher-order versions are!

Outline

Proof Terms

- λ-calculus / type theory gives a natural notion of proof terms and proof checking.
- "Curry-Howard"
- de Bruijn independently knew and implemented this in the late 1960s.


Nicolaas Govert de Bruijn

- AUTOMATH68 was the first real proof checker.

Proof Terms

A simple case is proof terms for (natural deduction proofs) in simple type theory.

Example:

$$\lambda p : o.\lambda u : p.u$$

is a proof of

$$\forall p : o.p \to p$$

Proof Terms

It's easy to give a calculus for a higher-order set theory with proof terms using only old well-understood ideas.

Set Theory
(Cantor, late 19th century;
Zermelo 1908; Fraenkel 1930s
Tarski 1930s;
Grothendieck 1970s)

Higher-order logic (Church 1940)

Checkable proof terms (de Bruijn 1968)

Outline

Egal

- My old interactive prover circa 2013 - 2018 for higher-order set theory with proof terms.
- For 5 months in 2014 used this to do a bitcoin treasure hunt.
- I prove a theorem (about basic set theory), use the proof to determine a private key, put some bitcoin at the address and challenged people to rediscover the same proof as me.

Egal

- My old interactive prover circa 2013 - 2018 for higher-order set theory with proof terms.
- For 5 months in 2014 used this to do a bitcoin treasure hunt.
- I prove a theorem (about basic set theory), use the proof to determine a private key, put some bitcoin at the address and challenged people to rediscover the same proof as me.
- Almost *10 people* participated!

Egal

This might have been one of them:

Charles Hoskinson (Bitshares, Ethereum, IOHK/IOG, Cardano)

- IOHK project (2015-2017): Qeditas
- Project for a theorem proving blockchain
- Egal code used for the checker (but with evil polymorphism added!)
- First person to prove a theorem with *any* correct proof gets the "bounty."
- ...never launched?

Outline

Proofgold

- Working extension of Qeditas (but without evil polymorphism) since 2020.
- A blockchain with a cryptocurrency...
- but also supports publishing formal math.
- No email or orcID required!
- Over 10K published proofs of theorems.
- So far about 800K Proofgold bars exist (with a cap of about 12M).
- Almost 300K of these are bounties on theorems.
- Proofs could be published by anyone, even nonhumans.

Outline

Megalodon

- Next generation of Egal
- ITP for higher-order set theory
- Allowed to assume as proven anything proven in the Proofgold chain
- Can produce Proofgold documents to publish into the chain, extending the library.

Outline

Surreal Numbers

- Example formalization: Conway's Surreal Numbers.
- John Conway. On Numbers and Games. 1976.
- If L is a set of surreal numbers and R is a set of surreal numbers and $x < y$ for every $x \in L$ and $y \in R$, then there is a "first" surreal z such that $L < z < R$ (pointwise).

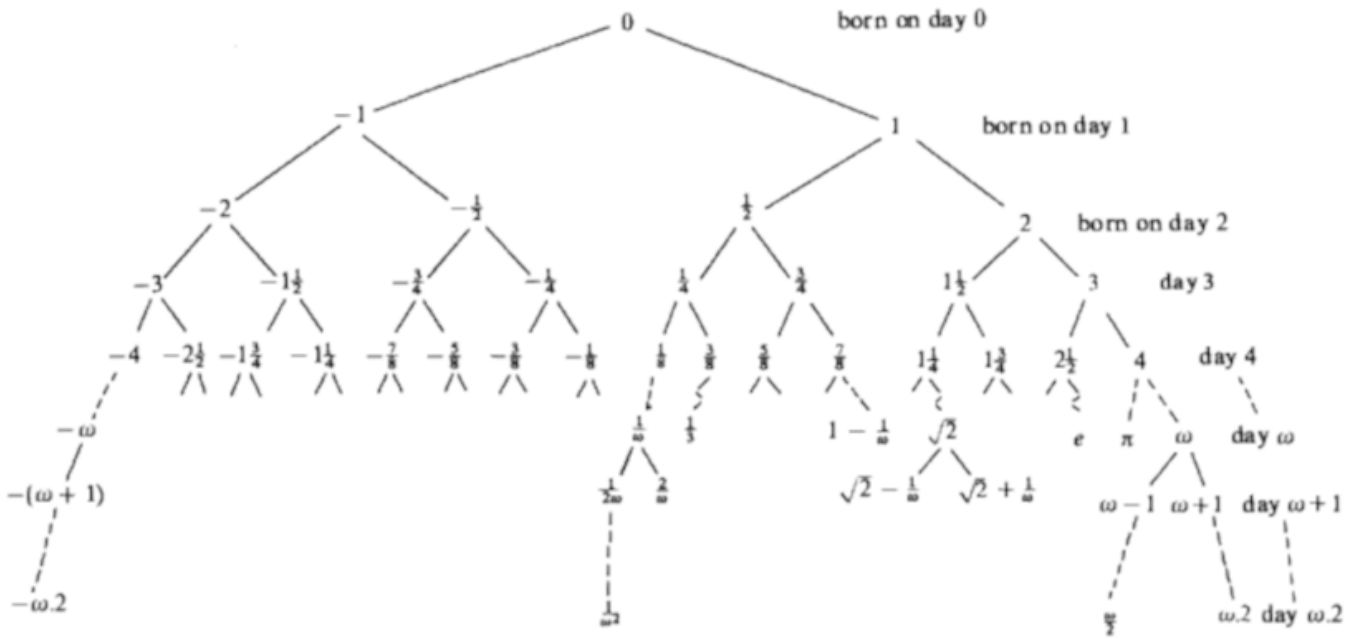


FIG. 0. When the first few numbers were born.

Surreal Numbers

- Nice property: $\omega \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}$
- How to prove $\forall x \in \mathbb{R}.x \neq 0 \rightarrow \exists y \in \mathbb{R}.xy = 1$?

Surreal Numbers

- Nice property: $\omega \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}$
- How to prove $\forall x \in \mathbb{R}.x \neq 0 \rightarrow \exists y \in \mathbb{R}.xy = 1$?
- First prove $\forall x \in \mathbb{R}.0 < x \rightarrow \exists y \in \mathbb{R}.xy = 1$?

Surreal Numbers

- Nice property: $\omega \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}$
- How to prove $\forall x \in \mathbb{R}.x \neq 0 \rightarrow \exists y \in \mathbb{R}.xy = 1$?
- First prove $\forall x \in \mathbb{R}.0 < x \rightarrow \exists y \in \mathbb{R}.xy = 1$?
- Before that prove $\forall x \in \mathbb{R}.0 < x < 1 \rightarrow \exists y \in \mathbb{R}.xy = 1$?

Surreal Numbers

- Nice property: $\omega \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}$
- How to prove $\forall x \in \mathbb{R}.x \neq 0 \rightarrow \exists y \in \mathbb{R}.xy = 1$?
- First prove $\forall x \in \mathbb{R}.0 < x \rightarrow \exists y \in \mathbb{R}.xy = 1$?
- Before that prove $\forall x \in \mathbb{R}.0 < x < 1 \rightarrow \exists y \in \mathbb{R}.xy = 1$?
- All published in Proofgold, so it "knows about" the reals.