Evident logic

Ben Sherman

CSAIL
Massachusetts Institute of Technology
Cambridge, MA
sherman@csail.mit.edu

Abstract

Many logical systems justify their consistency via proofs of cut elimination or normalization. However, these metatheoretic proofs inevitably use a transfinite induction which cannot be proved well-founded within the system itself. We present *evident logic*, a logical system with a normalization proof which does not require any induction that is stronger than what is available within the logical system itself. Via the Curry-Howard correspondence, this system may also be viewed a programming language for very efficient computations. We provide a typechecker and normalizer within Coq; all programs normalize in time constant with respect to their length. We discuss the potential advantages of working with weak foundations.

1. Introduction

Since Gentzen transformed the study of logical consistency, logicians have craned their necks upwards. Continuing the infantile competition of naming the larger number (and carrying well past ∞), it seems that the goal of logicians is to reach towards the heavens, building formal systems with higher and higher proof-theoretic ordinals, with the ultimate goal of reaching God (who supposedly lives near ω_1^{CK}) and resolving the fundamental question regarding the meaning of life: is the continuum hypothesis true?

This program follows the tired scheme of adding stronger and stronger inductive princples that reach closer and closer to inconsistency. It seems the scheme itself can be described by the transfinite iteration of the following process:

$$\begin{aligned} f(0) &= \text{recursion} \\ f(\mathsf{S}\ n) &= \text{induction-}f(n) \\ f(\mathsf{lim}\ \sigma) &= \text{higher ?}, \end{aligned}$$

though the problem of what is done with limit ordinals remains open[4].

However, this program fails to remain grounded; its relevance to computer scientists and engineers, who are used to looking down at their feet (at least in social situations), is dubious. However, it turns out that working downwards, observing the earth on which we stand, has its own rewards. Angiuli's artisanal type theory [1] demonstrates that the benefits of the approach of remaining grounded include a more organic, authentic, and meaningful experience. In fact, the notion of "artisanally-performed beta-reductions" largely anticipates the results of this work.

The grounded program cherishes the earth and studies its secrets. We observe the centipede, and pull off its legs, one by one, until it finally fails to function. Harvey Friedman pioneered this field, known as reverse mathematics, in the 1970s. It is at its heart a program in *conservation*; using only the principles that we truly need.

But contemporary mathematicians build skyscrapers of abstractions, paying little attention to their decadent waste. One such waste is the use a proof by contradiction when a direct proof is immediately available. Other instances are clunky definitions: Why is a function defined as a subset of ordered pairs satisfying certain existence and uniqueness conditions, when obviously the notion of a function is logically prior? Why are quotients implemented with equivalence classes? Nobody actually thinks of these objects as their definitions! In the Bourbaki foundations, the term representing the number 1 requires 4523659424929 symbols[6]. Had the Bourbaki foundations been accepted, who knows whether any symbols would remain for our grandchildren.

Of course, the grounded program, of establishing *meaning* in logic, can trace its roots back to Bertrand Russell and the *Principia mathematica* program, and our lord and savior, L.E.J. Brouwer and his intuitionism program. Russell coined the term "impredicativity" to represent all that is evil in the world, and committed himself to eliminating this scourge from the face of the Earth. However, Russell's program failed when Gödel's proved its impossibility in 1931 with his well-renowned incompleteness theorem.

Edward Nelson correctly identified the source of evil (i.e., impredicativity) leading to Gödel's incompleteness theorem: the very definition of the Peano induction scheme is in fact impredicative. Nelson's program of predicative arithmetic developed mathematics in a system which allowed recursion strictly weaker than in primitive recursive arithmetic (which is in turn weaker than Peano arithmetic)[7].

Bellantoni, Cook, and Leivant in 1992 [2] translated the notion of predicative recursion across the Curry-Howard correspondence and found that predicative recursion was precisely a logical characterization of polynomial-time computer programs. That is, those who conserve their logical principles conserve their computational resources as well. This groundbreaking result provided a connection between recursion theory and complexity theory.

Many logical systems justify their consistency using proofs of cut-elimination or normalization according to Gentzen's program of ordinal analysis. However, these proofs are unsatisfactory, because they use induction that cannot be proved well-founded within

DOI: http://dx.doi.org/10.1145/(to come)

Figure 1. Type formation rules for evident logic.

$$\frac{}{\top \text{ type}}\ \top\text{-FORM} \qquad \frac{}{\bot \text{ type}}\ \bot\text{-FORM}$$

$$\frac{A \text{ type} \qquad B \text{ type}}{A \vee B \text{ type}}\ \vee\text{-FORM} \qquad \frac{A \text{ type} \qquad B \text{ type}}{A \wedge B \text{ type}}\ \wedge\text{-FORM}$$

$$\frac{A \text{ type} \qquad B \text{ type}}{A \to B \text{ type}}\ \to\text{-FORM} \qquad \frac{A \text{ type} \qquad B \text{ type}}{\forall A, B \text{ type}}\ \forall\text{-FORM}$$

$$\frac{A \text{ type} \qquad B \text{ type}}{\exists A, B \text{ type}}\ \exists\text{-FORM} \qquad \frac{}{\mathbb{N} \text{ type}}\ \mathbb{N}\text{-FORM}$$

the system itself. This includes even consistency arguments for predicative recursion. As André Weil said, "Gentzen proved the consistency of arithmetic, i.e., induction up to the ordinal ω, by means of induction up to ϵ_0"[5]. But the conventional wisdom of the entire theory of ordinal analysis is that this dissatisfaction is in fact necessary.

The logical system presented in this paper, called *evident logic*, proves otherwise. It has a normalization argument which does not require the use of induction at all, and so its consistency is immediately evident. They key to evident logic is not to admit elimination rules (except for $\bot$) as valid rules of deduction.

2. Evident logic

Evident logic looks like almost every other logical system or programming language, but we produce every rule of inference here because it makes us look smart. In fact, what makes evident logic special is not the rules that it has, but the rules it lacks, so we wish the reader good luck in figuring that out.

Theorem 1 (Normalization)**.** *For every term $e : A$ there is a term $e' : A$ which is in normal form.*

Proof. Immediate, since there is no cut elimination rule (i.e., beta reduction rule), every term in evident logic is already in normal form. □

Theorem 2 (Consistency)**.** *There is no term of type $\bot$.*

Proof. Note that types of the hypotheses of all the proof rules are subformulae of the types of the conclusions, with the exception of the $\bot$-ELIM rule. Therefore, the only way to produce a term of type $\bot$ would be by $\bot$-ELIM, which requires that we already have a term of type $\bot$ on hand, so in fact the construction of a term of type $\bot$ is impossible. □

3. Discussion

3.1 Logic and computation

The fact that normalization arguments are unsatisfactory is in fact relevant to computation. For instance, one could easily write a computer program in a language supporting primitive recursion that ostensibly solves the game of chess. That is, one can write a program

$$e : \mathsf{black\_wins} \vee \mathsf{white\_wins} \vee \mathsf{draw}$$

where each type represents a proof that chess, played optimally, will have that result. The interpretation of disjunction in natural deduction is that one can prove a disjunction if and only if one can prove one of the disjuncts, but the chess example shows this

Figure 2. Introduction and elimination rules for evident logic.

$$\frac{}{\mathsf{unit} : \top}\ \top\text{-INTRO} \qquad \frac{e : \bot \qquad A \text{ type}}{\mathsf{abort}\ e : A}\ \bot\text{-ELIM}$$

$$\frac{p : A \qquad q : B}{\langle p, q\rangle : A \wedge B}\ \wedge\text{-INTRO} \qquad \frac{y : B}{\Lambda\, y : A \to B}\ \to\text{-INTRO}$$

$$\frac{p : A}{\mathsf{inl}\ p : A \vee B}\ \vee\text{-INTRO-L} \qquad \frac{q : B}{\mathsf{inr}\ q : A \vee B}\ \vee\text{-INTRO-R}$$

$$\frac{}{0 : \mathbb{N}}\ \mathbb{N}\text{-ZERO} \qquad \frac{n : \mathbb{N}}{\mathsf{succ}\ n : \mathbb{N}}\ \mathbb{N}\text{-SUCC}$$

$$\frac{x : A \qquad p : P}{\langle x, p\rangle : \exists A, P}\ \exists\text{-INTRO} \qquad \frac{A \text{ type} \qquad p : P}{\Lambda p : \forall A, P}\ \forall\text{-INTRO}$$

is blatantly false: we challenge any reader to provide a proof of any of the three disjuncts. The problem is that the number of steps in the computation n might be enormous relative to the size $|e|$ of the term e: the relation between n and $|e|$ is bounded only by the Ackermann function.

In evident logic, this defect is remedied, because if one could produce a term e' in evident logic with the type above, it would contain as a sub-term one of the three disjuncts. Therefore, proofs in evident logic are *more meaningful* than in systems with stronger foundations. This corresponds to the fact that terms in evident logic are equivalent to *very efficient computer programs*, which is the name we give for computer programs which run in constant time.

It is unfortunate that the field of complexity theory has given very little attention to the study of very efficient computer programs. Evident logic shows that the class of very efficient computer programs in fact has a deep and important logical structure that merits greater study. An algorithm which runs in constant time is an algorithm which is optimally fast, and optimally efficient, so we would hope that algorithm designers would strive to produce very efficient algorithms.

3.2 Meaning in logical systems

Brouwer's intuitionist program attempted to address what Bishop calls the "debasement of meaning" in contemporary mathematics[3]. In this view, "meaningful distinctions deserve to be maintained", and constructive mathematics preserves distinctions which classical mathematics lacks. In constructive mathematics, the positive quantifiers (disjunction and existence) have meaning that classical mathematics cannot express. By weakening the foundations, one observes more distinctions.

Nelson's predicative arithmetic again weakens the foundations and gives further distinctions. For instance, the unary natural numbers and binary natural numbers are no longer the same, as the binary natural numbers are a far more efficient representation. So one can exponentiate binary natural numbers but not unary natural numbers, as exponentiate unary natural numbers takes too long. However, even polynomial-time algorithms might be so slow that they are effectively useless. Evident logic allows one to distinguish between the constant-time algorithms and those which take longer, by only admitting the former.

3.3 Prepping for the inconsistapocalypse

Gödel's second incompleteness theorem shows that it is impossible to prove the consistency of arithmetic, and as a result the normaliza-

tion arguments (or *any* consistency arguments for that matter) for logical systems extending primitive recursive arithmetic can never be formulated within the theories themselves.

The obvious conclusion to draw is that primitive recursive arithmetic is probably inconsistent. Therefore, it is not a matter of *if* the inconsistapocalypse strikes, but *when*. Unfortunately, the majority of the mathematical community has its head in the sand; very few are preparing for the post-inconsistapocalyptic world. Fortunately, Harvey Friedman's reverse mathematics program provides a good field guide for which theorems will survive in different potential scenarios. Edward Nelson created by far the best post-inconsistapocalypse survival guide[7][8], demonstrating how to reconstruct a significant fraction of mathematics with only the mangled ruins that will remain after the rapture. Vladimir Voevodsky began prepping in 2010 by advocating for constructive type theory as a system whose theorems will hold even after the inconsistapocalypse[9]. In essence, Voevodsky argues that computer normalizers can convert terms from constructive type theories and elaborate them into terms which are truly incontrovertible. By incompleteness we cannot be sure that the normalization process terminates. We claim that evident logic represents precisely those terms which Voevodsky deems unassailable, and so any of these terms can be faithfully represented in terms of evident logic.

4. Conclusion and future work

Evident logic presents a logic where proofs are more meaningful than in other systems, and so the theorems proved are more true. This makes it an excellent bastion for when the inconsistapocalypse comes. Before the inconsistapocalypse or the AI takeover (whichever comes first) we can perhaps use computers to convert terms from other systems into terms of evident logic.

References

[1] C. Angiuli. Artisanal type theory. *Proceedings of SIGBOVIK 2015*, pages 17–19, 2015.

[2] S. Bellantoni and S. Cook. A new recursion-theoretic characterization of the polytime functions. *Computational complexity*, 2(2):97–110, 1992.

[3] E. Bishop. *Schizophrenia in contemporary mathematics*. American Mathematical Society, 1973.

[4] P. Dybjer. A general formulation of simultaneous inductive-recursive definitions in type theory. *The Journal of Symbolic Logic*, 65(02):525–549, 2000.

[5] J.-Y. Girard. *The Blind Spot: lectures on logic*. European Mathematical Society, 2011.

[6] A. R. Mathias. A term of length 4523659424929. *Synthese*, 133(1-2): 75–86, 2002.

[7] E. Nelson. *Predicative Arithmetic*. Princeton University Press, 1986. ISBN 9781400858927.

[8] E. Nelson. Warning signs of a possible collapse of contemporary mathematics. *Infinity (eds. Michael Heller, W. Hugh Woodin)*, pages 76–85, 2011.

[9] V. Voevodsky. What if current foundations of mathematics are inconsistent. *Video lecture commemorating the 80th anniversary of the Institute for Advanced Study (Princeton)(http://video. ias. edu/voevodsky-80th)*, 2010.