CS6180: Lecture 4

Robert Constable

August 31, 2017

1 Lecture Summary

Brief survey of proof styles

Styles for presenting proofs have been formalized, analysed, and compared as part of the subarea of logic called *proof theory*. There are many excellent textbooks on this subject such as *Intuitionism and Proof Theory* [15], *Combinators, λ–Terms, and Proof Theory* [21], *Basic Proof Theory* [23, 27], *Handbook of Proof Theory*[24], and *Natural Deduction* [19].

We will briefly discuss the following major styles, already mentioned in Lecture 3. In the previous lecture we examined *Hilbert style* proofs, illustrated by Kleene's axioms for first-order logic and elementary number theory. The number theory comes in two versions, *Peano Arithmetic, PA* that uses all of Kleene's axioms and *Heyting Arithmetic, HA* which omits the "dreaded rule 8," and is thus a constructive number theory. In this lecture we will introduce the notion of constructive and intuitionistic mathematics and introduce you to the most impactful ideas in these subareas of logic. These ideas are at the core of constructive type theory, and recently Nuprl has implemented intuitionistic type theory, an even more expressive theory based on the seminal ideas of L. E. J. Brouwer [11, 4, 3, 26, 25]. Constructive theories are sub-theories of Brouwer's intuitionistic mathematics which was formalized by Kleene and Vesley in their book *Foundations of Intuitionistic Mathematics* [12] and in earlier papers by Kleene [13, 14].

- Hilbert style (illustrated by Kleene's axioms and inference rules in Lecture 3).
- Natural deduction (illustrated in PLCV proofs in Lecture 3).
- Sequent calculus (Kleene covers in his Chapter 15, also Gentzen [9]).
- Refinement style (used in Nuprl [7], a top down organization of sequent style proofs, developed in the PhD thesis of Joseph L. Bates, *A Logic for Correct Program Development* [1]).
- Tableaux systems (used by Smullyan in his classic book First-Order Logic [20] and studied in CS4860 in the spring 2018 semester).
- Tactic based systems pioneered by Robin Milner in *Edinburgh LCF* [10].

The LCF proof system changed the author's way of designing and implementing proof systems. It led to the Nuprl proof style which we call *Refinement Logic*, a top down sequent calculus with tactics [8, 2].

1.1 Kleene's Axioms for Peano and Heyting Arithmetic

1a. $A \Rightarrow (B \Rightarrow A)$.

1b. $(A \Rightarrow B) \Rightarrow ((A \Rightarrow (B \Rightarrow C)) \Rightarrow (A \Rightarrow C))$.

Inference Rule 2:

$$\frac{A, A \Rightarrow B}{B}$$

3. $A \Rightarrow (B \Rightarrow A\&B)$.
4a. $(A\&B) \Rightarrow A$.
4b. $(A\&B) \Rightarrow B$.
5a. $A \Rightarrow (A \vee B)$.
5b. $B \Rightarrow (A \vee B)$.
6. $(A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \vee B) \Rightarrow C))$.
7. $(A \Rightarrow B) \Rightarrow ((A \Rightarrow \sim B) \Rightarrow \sim A)$.
8*. $\sim\sim A \Rightarrow A$. **classical**

Predicate Calculus Axioms

Inference Rule9:

$$\frac{C \Rightarrow A(x)}{C \Rightarrow \forall x.A(x)}$$

10. $\forall x.A(x) \Rightarrow A(t)$.
11. $A(t) \Rightarrow \exists x.A(x)$.

Inference Rule 12:

$$\frac{A(x) \Rightarrow C}{\exists x.A(x) \Rightarrow C}$$

Number Theory Axioms

13. $A(0)\&\forall x.(A(x) \Rightarrow A(x')) \Rightarrow A(x)$.
14. $a' = b' \Rightarrow a = b$.
15. $\sim (a' = 0)$.
16. $a = b \Rightarrow (a = c) \Rightarrow b = c$.
17. $a = b \Rightarrow a' = b'$.
18. $a + 0 = a$.
19. $a + b' = (a + b)'$.
20. $a \times 0 = 0$.
21. $a \times b' = (a \times b) + a$.

How do we use these axioms to create proofs? Kleene gives this example on page 85. This is an example of what logicians call a Hilbert style axiom system in which there is only one proof rule in addition to the axioms. The is the rule of *modus ponens*, if we know A and $A \Rightarrow B$, then we can deduce B.

1.$A \Rightarrow (A \Rightarrow A)$. Axiom schema 1a.
2.$(A \Rightarrow (A \Rightarrow A)) \Rightarrow (A \Rightarrow ((A \Rightarrow A) \Rightarrow A)) \Rightarrow (A \Rightarrow A)$. Schema 1b.
3.$(A \Rightarrow ((A \Rightarrow A) \Rightarrow A))$. Rule 2,1,2
4. $A \Rightarrow ((A \Rightarrow A) \Rightarrow A)$ Axiom schema 1a.
5.$(A \Rightarrow A)$. Rule 2,4,3.

Kleene also shows on page 84 how to prove $a = a$ for numerical variables a in 17 lines of axioms and inference rules. This kind of proof makes formal logic and formal mathematics seem *impossibly tedious* and essentially incomprehensible. What computer science brought to this research early on is the notion that machines could conduct these tedious proofs in the background and achieve an exceptionally strong degree of formal correctness. In their classic paper, *Empirical Explorations with the Logic Theory Machine: A case study in heuristics* [18], Newell, Shaw, and Simon demonstrated that computers could execute the tediously long proofs and check all the details so that humans would not be required to function at these low levels. This was a revolutionary advance in automated reasoning that inspired decades of further research to create the AI tools that make proof assistants possible.

We will examine an example from Kleene showing how to structure proofs as trees using Frege's turnstile symbol, $\vdash$, separating hypotheses from the goal to be proved, $H \vdash G$ where the hypotheses H is a *list* of labled formulas and variable declarations, and the goal is a single formula. Such expressions are called *sequents*. In the Nuprl book they are written as $H >> G$. In these notes we use $x_1 : A_1, ..., x_n : A_n \vdash G$. Some formalisms, such as tableaux, allow multiple goals, say $G_1, ..., G_m$, but Nuprl does not.

2 Intuitionistic First-Order Logic, iFOL

In his 1908 PhD thesis [3], Brouwer proposed a new interpretation of first-order logic. His idea is that we come to know mathematical truths based on our intuitive experience of the *continuum of time* and how the mind breaks this continuum into the experience of now, before, and after. There is no ideal *Platonic world* in which mathematical statements are either true or false, and that we come to know these truths based on a conception of logic in which every meaningful mathematical statement is either true in the Platonic world or false there. This Platonic view was the prevailing understanding of truth before Brouwer proposed a compelling alternative. He justified his alternative understanding based on how we come to experience mathematical ideas and recognize those constructions which the human mind can grasp and explore. We are not able to experience this Platonic world, yet we can recognize simple logical truths as intuitively true and computationally meaningful in the sense of our mental constructions.

Brouwer gave the interpretation of mathematical truth that Per Martin-Löf writes about in several papers [17, 16]. These ideas are discussed well in our on-line textbook [22]. We will also study the presentation in *On the Meaning of the Logical Constants and the Justification of the Logical Laws* [17]. This semantics is also given in the Nuprl book [7] and in numerous other articles and books [6, 5].

References

[1] J. L. Bates. *A Logic for Correct Program Development*. PhD thesis, Cornell University, 1979.

[2] J. L. Bates and Robert L. Constable. Proofs as programs. *ACM Transactions of Programming Language Systems*, 7(1):53–71, 1985.

[3] L.E.J. Brouwer. *Over de grondslagen der wiskunde (On the foundations of mathematics)*. PhD thesis, Amsterdam and Leipzig, 1907.

[4] L.E.J. Brouwer. Intuitionism and formalism. In P. Benacerraf and H. Putnam, editors, *Philosophy of mathematics: selected writings*. Cambridge University Press, 1983.

[5] Robert L. Constable. Naïve computational type theory. In H. Schwichtenberg and R. Steinbrüggen, editors, *Proof and System-Reliability, Proceedings of International Summer School Marktoberdorf, July 24 to August 5, 2001*, volume 62 of *NATO Science Series III*, pages 213–260, Amsterdam, 2002. Kluwer Academic Publishers.

[6] Robert L. Constable. Computational type theory. *Scholarpedia*, 4(2):7618, 2009.

[7] Robert L. Constable, Stuart F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, Douglas J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, James T. Sasaki, and Scott F. Smith. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice-Hall, NJ, 1986.

[8] Robert L. Constable, T. Knoblock, and J. L. Bates. Writing programs that construct proofs. *Journal of Automated Reasoning*, 1(3):285–326, 1984.

[9] Gerhard Gentzen. Investigations into logical deduction (1934). In M. Szalo, editor, *The Collected Paers of Gerhard Gentzen*. North-Holland, Amsterdam, 1969.

[10] Michael Gordon, Robin Milner, and Christopher Wadsworth. *Edinburgh LCF: a mechanized logic of computation*, volume 78 of *Lecture Notes in Computer Science*. Springer-Verlag, NY, 1979.

[11] A. Heyting, editor. *L. E. J. Brouwer Collected Works*, volume 1. North-Holland, Amsterdam, 1975.

[12] S. C. Kleene and R. E. Vesley. *Foundations of Intuitionistic Mathematics*. North-Holland, 1965.

[13] S.C. Kleene. On the interpretation of intuitionistic number theory. *J. of Symbolic Logic*, 10:109–124, 1945.

[14] S.C. Kleene. Mathematical logic: Constructive and non-constructive operations. *Proceedings of the International Congress of Mathematics*, pages 137–153, 1960.

[15] H. Lauchli. An abstract notion of realizability for which intuitionistic predicate calculus is complete. In J. Myhill, A. Kino, and R.E. Vesley, editors, *Intuitionism and Proof Theory*, pages 227–34. North-Holland, Amsterdam, 1970.

[16] Per Martin-Löf. *Notes on Constructive Mathematics*. Almqvist & Wiksell, Uppsala, 1970.

[17] Per Martin-Löf. On the meaning of the logical constants and the justification of the logical laws. Lectures in Siena, 1983.

[18] A. Newell, J.C. Shaw, and H.A. Simon. Empirical explorations with the logic theory machine: A case study in heuristics. In *Proceedings West Joint Computer Conference*, pages 218–239, 1957.

[19] D. Prawitz. *Natural Deduction*. Dover Publications, New York, 1965.

[20] R. M. Smullyan. *First–Order Logic*. Springer-Verlag, New York, 1968.

[21] S. Stenlund. *Combinators, λ–Terms, and Proof Theory.* D. Reidel, Dordrechte, 1972.

[22] Simon Thompson. *Type Theory and Functional Programming*. Addison-Wesley, 1991.

[23] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, Amsterdam, 1996.

[24] A.S. Troelstra. Realizability. In S.R. Buss, editor, *Handbook of Proof Theory*, pages 407 – 473. Elsivier Science, 1998.

[25] Mark van Atten. *On Brouwer*. Wadsworth Philosophers Series. Thompson/Wadsworth, Toronto, Canada, 2004.

[26] Walter P. van Stigt. *Brouwer's Intuitionism*. North-Holland, Amsterdam, 1990.

[27] S. Wainer and L. Wallen. Basic proof theory. In Simmons & Wainer Aczell, editor, *Proof Theory*, pages 1–26. Cambridge University Press, 1993.