CS4860-2020-Lecture 2

Robert L. Constable

September 6, 2020

Abstract

How do we convey the remarkable truth that the ancient study of logic, dating to Aristotle over two thousand four hundred years ago, is now more relevant than ever? It is especially important in one of the youngest academic disciplines, computer science, roughly only fifty years old. Our goal is to understand this relevance to computer science and its potential. Aristotle gave us a dozen books to build on – *Prior Analytics* (2 books), *Posterior Analytics* (two more books), and *Topics* (eight books). He undertook *the earliest known formal study of logic* and introduced the concept of *variables.* He studied for twenty years at Plato's Academy. His wife Pythias taught Alexander the Great.

Both constructive and intuitionistic mathematics are *fundamentally about computation*, as is computer science. Intuitionistic mathematics originates in the work of L.E.J. Brouwer, circa 1907 to 1966, and is primarily about understanding *the continuum of real numbers* and the logical principles that govern it. Oddly one of Brouwer's earliest contributions was to logic, and remains controversial in some logic courses. He denied the *law of excluded middle* $P \vee \neg P$ as he was working on his doctoral dissertation. It was a bold move for a young PhD student to challenge Aristotle. That was part of Brouwer's character. We will begin to examine his reasoning and the responses to it in this lecture.

1 The law of excluded middle

The law of excluded middle says that any proposition P is either true or false, written symbolically as $(P \vee \neg P.)$ How could this possibly be false? How could a young PhD student be so bold as to challenge this fundamental law of logic? Indeed it is impossible to prove its negation, $\neg(P \vee \neg P)$, because that would mean showing that P can't be true, hence we would know $\neg P$. So this simple law cannot be false. Brouwer agreed, saying that what we really know is only that, $\neg\neg(P \vee \neg P)$. This is indisputable. But $(P \vee \neg P)$ says something stronger constructively. It says we can decide which option is the case. Is it P or is it $(\neg P)$? Nowadays we have more experience with *unsolvable problems.* A computer scientist might believe that either the polynomial time solvable problems, P are the same type as the non-deterministic polynomial time solvable problems, NP, giving $P = NP$ or that these classes are not the same,$\neg(P = NP)$. But no one knows which is the case, and even more important, we are not sure that we will ever know.

Hence, to claim $(P = NP) \vee \neg(P = NP)$ seems unjustified. We might never know. Moreover, it does not make sense to say $\neg((P = NP) \vee \neg(P = NP))$ for the reasons we gave in the previous paragraph. Resolving this could be beyond our problem solving abilities. Moreover, if this question is resolved, there are a large number of other open problems, say $OP_1, OP_2, OP_3, ...,$ and it would be a bold claim that we can eventually resolve all of them, i.e. claiming $(OP_i) \vee \neg(OP_i)$. That claim seems indefensible, given the number of well known open problems in mathematics. So what does $P \vee \neg P$ really mean as a logical principle? It is more likely that we will always have some impossibly difficult open mathematical problems. But as Brouwer pointed out, there is something relevant that we can already say, namely $\neg\neg(P \vee \neg P)$! Why is that?

The simple answer is this. If we could prove $\neg(P \vee \neg P)$, then we would know $\neg P$. Brouwer knew this, and it is worth pondering why he knew it. Brouwer said that the correct version of this logical law is simply $\neg\neg(P \vee \neg P)$. We know why this is the case from the above discussion.

In previous articles we have quoted Errett Bishop from his book [8]. He said:"Every mathematical statement ultimately expresses the fact that if we perform certain computations within the set of positive integers, we should get certain results. Thus even the most abstract mathematical statement has a computational basis." This is an interesting informal observation, but it seems difficult to make it precise and even more difficult to prove.

This and subsequent lectures will focus on the treatment of logic made possible by computer science. We examine connections to core computational concepts from mathematics as well as to those from computer science. Computer science has provided concepts and methods to formalize, implement, and strengthen some of Euclid's and Brouwer's ideas as we will show. Brouwer claimed that it is not possible to formalize **all** of his intuitionistic mathematics. We will discuss such limitations. Our discussion will highlight differences between *constructive* and *intuitionistic* mathematics, thus between Euclid and Brouwer.

In due course we will present several results in intuitionistic mathematics that we have proved formally using the Nuprl proof assistant. As far as we know, as of 2020, Nuprl remains the only proof assistant that implements a significant part of intuitionistic mathematics. We organize the exposition in this course by following the progression of formal logics starting with the propositional calculus, then first-order logic, then higher-order logic, then type theory. We discuss the impact of intuitionism on formal logics. Type theory is a much more suitable formalism for expressing intuitionistic mathematics than set theory. Moreover we have considerable experience using Nuprl in creating, formalizing, and implementing *constructive real analysis.* This allows us to contrast the theorems of *classical* and *constructive* mathematics with those of *intuitionistic* mathematics. We look for ways to and results across these different approaches where possible. We will not cover Brouwer's *creating subject results.*

2 Intuitionistic Logic

Brouwer created intuitionistic mathematics to account for the *continuum* and computable operations on it. For Brouwer a continuum of great interest is *time.* As we will see, his research had a profound impact on our understanding of continua, both time and space. The real numbers are the basic mathematical abstraction associated with continua. They were were not identified as a distinct type until Cantor's work in 1872 building on Weierstrass' treatment of the irrational numbers in 1859. So the subject is at most 148 years old. Brouwer's life long engagement with this concept can be traced to as early as 1907, the year of his PhD thesis.

Brouwer was almost surely unaware of the coming impact of computers on mathematics and on the technology of creating proofs with help from computers. *Computer assisted mathematics is almost the exact opposite of intuitionistic mathematics.* Current machines have no "intuitive knowledge" about what they are doing. They work with a finite number of specific formal rules. We describe the way we work with computers as doing *formal mathematics*, and the commitment to enable this mode is *formalism.*

Our research enables an *integration* of formal computational mathematics with elements of intuitionistic mathematics. This style of work is illustrated by our Nuprl formal proof of the *Continuity Principle.* We could call this mode of research *formal intuitionistic mathematics.* This methodology combines two rich themes in the long and exciting history of mathematics. On the other hand, we are now in a unique new era that is distinguished by the role of *machine intelligence.* We will give examples of how the machine intelligence of the Nuprl system has led to new mathematical results. We will speculate on how we can nurture this new capability that many mathematicians and computer scientists find fascinating. It is destined to significantly influence the development of mathematics and computer science going forward.

Brouwer believed that to understand natural numbers, we should first think of 2, and then derive 1. He also believed that the explanation of $A \Rightarrow B$ is *impredicative.* A type definition is impredicative if it quantifies over the type being defined. This is not the case for the Nuprl type theory because we use a hierarchy of universes, and every type is in some universe U_i. Universe U_i belongs to the universe U_{i+1} to avoid being impredicative. This was an early decision for our type theory, following Per Martin-Lof by using universes. Godel claimed that logic cannot do anything decisive in mathematics. Intuition takes precedence over formalization. The Fan Theorem is the contrapositive of Konig's lemma, which is not constructively true. We know that Weak Konig's Lemma, WKL, implies the Fan Theorem, $WKL \Rightarrow Fan$. The Bar Theorem depends on the analysis of mental acts. At first Godel thought that choice sequences were not mathematical objects, but he changed his mind. Intuitionism also suggests a revision of classical mathematics. We will take up these topics here in due course.

Readers might wonder why the title of this document does not include implementing with *classical mathematics* as well as constructive and intuitionistic mathematics. The answer is that we want our proofs to provide computational information so that we can execute their computational content. A classical approach does not provide this. We look at a first rate classical account of a key theorem in intuitionistic analysis to make this point. We see that Courant in his very well known classical textbook, *Differential and Integral Calculus Vo1 1 and 2* [17] proves a very strong result from intuitionistic analysis. Such books about classical calculus have been used to teach real analysis to millions. Why don't we cover this approach? We give a reply in the section on the Real Number System where we show how classical reasoning about Brouwer's reals leads us astray.

We plan to include a brief account of *computational complexity.* For constructive logic, we know a significant amount and have published in this area. However, for intuitionistic mathematics the topic has not been robustly developed for reasons we will discuss. One immediate issue whether computational complexity measures make sense for free choice sequences. We know of no other commentary on this topic beyond what we will sketch here.

Course Plan Here are the topics we plan to cover.

1. **Introduction and Background to Formalization**
2. **Classical Propositional Logic from Smullyan's book [32]**
3. **Intuitionistic Propositional Logic, iPC**
4. **Evidence Based Mathematical Knowledge**
5. **Classical First-Order Logic from Smullyan's book, Chapter III, Godel Completeness Theorem [12]**
6. **Intuitionistic First-Order Logic, iFOL [13]**
7. **Intuitionistic First-Order Logic, iFOL and Bickford, Constable Completeness Theorem [6]**
8. **Elementary Number Theory, ENT**
9. **Godel's Incompleteness Theorem for ENT [33]**
10. **Intuitionistic Calculus**
11. **The Future of Machine Intelligence in Advancing Mathematics**

3 Proofs as Programs

Since 1985 we have been working to formalize proofs in mathematics and whenever possible implement the computational content in these proofs as computer programs. This is a way of turning *proofs into programs* [1]. This is particularly useful in real analysis where we can provide *infinite precision* real numbers. Lately we have been formalizing elements of L.E.J.Brouwer's intuitionistic mathematics and implementing them using the *Nuprl proof assistant*. Results and new insights are encouraging. For instance, *intuitionistic calculus* turns out to be technically "cleaner" than classical calculus on many topics and results, as we show. Brouwer's *Continuity Principle* is very expressive and useful, yet it contradicts classical mathematics. Brouwer's *Bar Induction* is an elegant form of induction that is classically true. We present both informal and formal versions of these key principles and theorems. The fact that every continuous function from closed intervals of the reals into the reals is *uniformly continuous* is a result of intuitionistic analysis that illustrates the expressive power and *practical value* of this theory. That one result saves the work of proving uniform continuity case by case in applications. Many more examples of the *technical advantages* of formal intuitionistic mathematics are included in these lectures. As more researchers become aware of Brouwer's deep insights, they will be more widely applied. A key goal of this course is to help students learn and apply these elegant and useful ideas. We will integrate them with our extensive work on constructive mathematics, as in formalizing elements of the Bishop and Bridges book *Constructive Analysis* [9].

The task of *implementing* mathematics brings together themes at the interface of mathematics and computer science. Computer science (CS) is a young and vibrant branch of science with roots in mathematics [38, 39]. There are at least four *research threads* that bring these two disciplines together. One is the **foundations of mathematics** thread, tracing its origins as far back as Euclid, at least 300 BCE. Another is the thread of **checking proofs using computers**, active since at least 1970 in the work of N.G. de Bruijn [18]. This theme led naturally to **formal proof creation with computer assistance** driven by the creation of proof assistants in computer science. We focus on two proof assistants, Coq [2] and Nuprl [14], because they are closely related and *they work together.* There are other excellent proof assistants mentioned later.

The fourth theme is **theorem and proof discovery** using proof assistants. This is part of the thriving area of computer science called *Artificial Intelligence* (AI). It is deployed to make modern proof assistants "smarter". There have been new discoveries in mathematics as a result of this synergistic combination of research themes and computer engineering advances. We see no end in sight to the steadily advancing synergy between technology, mathematical discovery, and human imagination. We will explore all four themes.

By 1985 the Nuprl proof assistant was used to implement *elements of constructive mathematics* including constructive *real analysis* presented in Bishop's book *Foundations of Constructive Analysis* [8] and in its enrichment in the Bishop and Bridges version *Constructive Analysis* [9]. From that implementation, we began to verify textbook material for real analysis *using Nuprl.* In 1986 the PRL research group wrote the book *Implementing Mathematics with the Nuprl proof development system* [14]. For various reasons, that book has been cited almost every week since it was first published by *Prentice-Hall.* It now has 2,120 citations and remains in print in both hardcover and paper back. A large team of doctoral students contributed to the proof assistant and to the book. Our current effort to implement intuitionistic mathematics was undertaken with a smaller group of experienced researchers. They are the co-authors on the five articles that document our implementation of the core of intuitionistic mathematics at Cornell University from 2014 until 2019 and the 2020 articles that continue this line of research. These articles are all cited as we introduce the technical ideas including new results in 2020 *Cubical type theory with several universes in Nuprl* [3] and *Open Bar – A Reconciliation between Intuitionistic and Classical Logic* [4]. We expect further articles on this topic in 2020.

In 2020 we are now using this implementation to explore elements of intuitionistic mathematics including calculus, topology, homotopy theory, and Cubical Type Theory (CTT). This challenging and exciting effort motivates a search for new ways of making Nuprl "smarter". Human intuition has always been critical in creating new mathematics, and *we can now explore more deeply connections between human intuition and machine intelligence.*

We have formalized the Nuprl type theory proof rules using the *Coq proof assistant* [16]. Nowadays we do not alter the Nuprl logical rules or add new ones without verifying them in Coq. We call this a *dual prover technology.* The Coq and Nuprl teams have worked with each other in a variety of ways since 1985 until today. We have collaborated on projects some of which we mention in this book. One outcome is that Coq will have access to elements of intuitionistic mathematics that we formalize. This could broaden the impact of formal intuitionistic mathematics.

We began writing these notes to determine the feasibility of creating a textbook for logic in computer science with precisely the focus of our title. Here is some relevant background. We are able to draw on Brouwer's extensive writings, also on the book *The Foundations of Intuitionistic Mathematics* [25] by Stephen Cole Kleene and Richard E. Vesley, the books by Heyting [22, 20, 21, 11] and the books by A.S. Troelstra [34, 35, 37, 36] and by Dirk van Dalen [43, 42, 44] and by Mark van Atten [40, 41]. We also depend on these resources [24, 23] as well as [10] and [26]. Moreover, Kleene's former PhD student, Professor Joan Rand Moschovakis, has written incisively on intuitionism [28, 27]. Her twenty eight page 1999 article with co-author Garyfallia Vafeiadou entitled *Intuitionistic Mathematics and Logic* [29] is in our view one of the *very best* insightful and compact accounts of intuitionistic mathematics written in English. We draw on this material and explicitly reference several of their insights and explanations.

Before we focus on intuitionism, we want to point out the advantages of creating detailed course notes. The book on the Nuprl proof assistant was *Implementing Mathematics with the Nuprl Proof Development System* [14], remains frequently cited after 34 years with 2,120 citations as of June 2020. In that book and in further work, we implemented several theorems in Errett Bishop's book *Foundations of Constructive Analysis* [8] including some new material added by Douglas Bridges in the updated book entitled *Constructive Analysis* [9]. These implementations allowed us to accomplish tasks in real analysis by *executing these formal proofs.* Our implementation *brought the book to life.* We built something useful which we are now applying to intuitionistic mathematics.

Moreover, we received permission from the publisher, *Springer-Verlag*, to display pages of the

Bishop and Bridges book *Constructive Analysis* on the PRL group home page, we give the url just below. Dr.Bickford has created links between definitions and theorems in Chapter 1 to their Nuprl formalization. An interesting task for future work is to formalize this chapter using our implementation of the relevant concepts from intuitionistic analysis, and compare the results. A similar artifact was created by Ariel Kellison to link her Nuprl formalization of theorems from Euclidean geometry to their presentation with diagrams available in Heath's well known book on Euclid's *Elements* [19].

3.1 A preview of topics

We have written extensively on constructive mathematics [14]. We started to use Nuprl to implement intuitionistic mathematics in 2017 [7], and we wrote two articles in 2018 [30, 5], another in 2019 [31] and one as yet unpublished in 2020 entitled *Open Bar: A Reconciliation between Intuitionistic and Classical Logic* available on the PRL group web page: http://nuprl-web.cs.cornell.edu/. The main work was in implementing the foundational intuitionistic principles such as the Continuity Principle and Bar Induction. We also needed an account of free choice sequences. Once we built the foundations, we were able to produce *verified, formal intuitionistic mathematics.* We also began to *experience the truth* of Brouwer's insight that it is not possible to formalize (or implement using digital computers) the full extent of intuitionistic truth. This situation brings to mind the comment by Godel that "logic cannot do anything decisive in mathematics." There is much more work to do in finding out just what this entails. What we already know is that many intuitionistic concepts and principles can be implemented. We are beginning to see in detail that intuitionistic calculus is both simpler and more advanced than the calculus that is now so widely taught.

3.2 Open Problems

For various topics in the subsequent sections, we will want to mention open problems in mathematics and computer science. To that end, we list some of these problems here.

1. Let P stand for the type of *polynomial time* computable problems and NP for the type of problems solvable in *non-deterministic polynomial time.* A famous open problem in computer science posed by Steve Cook [15] in 1971 is whether $P = NP$. It remains unsolved.

2. The *Hodge Conjecture* (HC) is that every Hodge Class on a projective complex manifold is algebraic. *A solution of this problem is worth one million dollars.*

3. The *Riemann Hypothesis* (RH) is that the Riemann zeta function has zeros only at the negative even integers and at complex numbers with real part 1/2. The problem has been

open since it was proposed by Riemann in 1859.

4. Hilbert's 16th problem, to determine the maximum number of closed separated branches of a plane algebraic curve of order n.

5. *Goldbach's Conjecture*: Every even integer greater than 2 is the sum of two primes. This is known to be the case for $n < 4 \times 10^{18}$.

References

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

[2] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development; Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. Springer-Verlag, 2004.

[3] Mark Bickford. Brouwer's fixed point theorem in intuitionistic mathematics. Technical report, 2020.

[4] Mark Bickford, Liron Cohen, Robert Constable, and Vincent Rahli. Open bar – a reconciliation between intuitionistic and classical logic. *Electronic Notes in Theoretical Computer Science*, pages 1–14, 2020.

[5] Mark Bickford, Liron Cohen, Robert L. Constable, and Vincent Rahli. Computability beyond Church-Turing via choice sequences. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 245–254, New York, NY, USA, 2018. ACM.

[6] Mark Bickford and Robert Constable. Formalizing the Intuitionistic Completeness proof for First-Order Logic in Nuprl. Department of Computer Science, Cornell University, Unpublished, 2013.

[7] Mark Bickford, Vincent Rahli, and Robert Constable. The Good, the Bad and the Ugly. In *Thirty-Second Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 266–279, 2017.

[8] E. Bishop. *Foundations of Constructive Analysis.* McGraw Hill, NY, 1967.

[9] E. Bishop and D Bridges. *Constructive Analysis.* Springer, New York, 1985.

[10] L.E.J Brouwer. *Brouwer's Cambridge Lectures on Intuitionism.* Cambridge University Press, 1981.

[11] L.E.J. Brouwer and A. Heyting. *Collected Works: Philosophy and foundations of mathematics, edited by A. Heyting.* Collected Works. North-Holland Pub. Co., 1976.

[12] Robert Constable and Mark Bickford. Intuitionistic Completeness of First-Order Logic. Technical Report arXiv reference 1110.1614, Computer Science Department, Cornell University, 2011.

[13] Robert Constable and Mark Bickford. Intuitionistic Completeness of First-Order Logic. *Annals of Pure and Applied Logic*, 165(1):164–198, January 2014.

[14] 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.

[15] S. A. Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing*, pages 151–158. ACM, New York, 1971.

[16] Thierry Coquand and G. P. Huet. Constructions: A higher order proof system for mechanizing mathematics. In *EUROCAL '85*, volume 203 of *Lecture Notes in Computer Science*. Springer-Verlag, 1985.

[17] R. Courant. *Differential and Integral Calculus Vo1 1 and 2*. Interscience Publishers, New York, 1959.

[18] N. G. de Bruijn. The mathematical language Automath: its usage and some of its extensions. In J. P. Seldin and J. R. Hindley, editors, *Symposium on Automatic Demonstration*, volume 125 of *Lecture Notes in Mathematics*, pages 29–61. Springer-Verlag, 1970.

[19] Euclid. *Elements*. Dover, approx 300 BCE. Translated by Sir Thomas L. Heath.

[20] A. Heyting. *Intuitionism, An Introduction*. North-Holland, Amsterdam, 1966.

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

[22] Arend Heyting. *Mathematische Grundlagenforschung. Intuitionismus.Beweistheorie.* Springer, Berlin, 1934.

[23] S. C. Kleene. *Introduction to Metamathematics*. D. Van Nostrand, Princeton, 1952.

[24] S.C. Kleene. Recursive functions and intuitionistic mathematics. *Proceeding of the International Congress of Mathematicians*, pages 679–685, 1950.

[25] Stephen Cole Kleene and Richard Eugene Vesley. *Foundations of Intuitionistic Mathematics*. North-Holland, 1965.

[26] Grigori Mints. *A Short Introduction to Intuitionistic Logic*. Springer US, University Series in Mathematics, New York,NY, 2000.

[27] Joan Rand Moschovakis. A classical view of the intuitionistic continuum. *Annals of Pure and Applied Logic*, 81:9–24, 1996.

[28] Joan Rand Moschovakis. The logic of Brouwer and Heyting. In *Logic from Russell to Church*, pages 77–125. 2009.

[29] Joan Rand Moschovakis and Garyfallia Vafeiadou. Intuitionistic mathematics and logic. Technical report, EU Common Fund, 1999.

[30] Vincent Rahli and Mark Bickford. Validating Brouwer's Continuity Principle for numbers using named exceptions. *Mathematical Structures in Computer Science*, 28:942–990, 2018.

[31] Vincent Rahli, Liron Cohen, Mark Bickford, and Robert Constable. Bar induction is compatible with constructive type theory. *Journal of the ACM (JACM)*, 66(2):13:1–13:35, April 2019.

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

[33] Raymond M. Smullyan. *Gödel's Incompleteness Theorems.* Oxford University Press, New York, 1992.

[34] Anne Sjerp Troelstra. *Metamathematical Investigation of Intuitionistic Mathematics*, volume 344 of *Lecture Notes in Mathematics.* Springer-Verlag, 1973.

[35] A.S. Troelstra. *Choice Sequences.* Oxford University Press, Oxford, 1977.

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

[37] A.S. Troelstra and D. van Dalen. *Constructivism in Mathematics, An Introduction*, volume I, II. North-Holland, Amsterdam, 1988.

[38] A. M. Turing. On computable numbers, with an application to the Entscheidungs problem. In *Proceedings London Math Society*, volume 42, pages 116–154, 1936.

[39] A. M. Turing. Computing Machinery and Intelligence. *Mind*, 59:433–60, 1950.

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

[41] Mark van Atten. Luitzen Egbertus Jan Brouwer. *Stanford Encyclopedia of Philosophy*, pages 1 – 17, 2020.

[42] Dirk van Dalen. *Brouwer's Cambridge Lectures on intuitionism.* Cambridge University Press, Cambridge, 1981.

[43] Dirk van Dalen. Intuitionistic logic. In *The Blackwell Guide to Philosophical Logic*, pages 224–257. Blackwell, Oxford, 2001.

[44] Dirk van Dalen. *L.E.J. Brouwer, Topologist, Intuitionist, Philosopher.* Springer-Verlag, 2013.