CS4860-2020-Full-Lecture 6: PC and iPC are decidable

Robert L. Constable

September 23, 2020

Abstract

In this lecture we present important results by Godel, Kleene, and others about the intuitionistic propositional calculus, iPC. Godel shows that iPC is only apparently narrower than classical PC. Kleene notes that the consistency of iPC secures the consistency of PC. Godel showed that there is no truth table decision procedure for iPC. Gentzen showed that there is a decision procedure for both PC and iPC *Investigations into logical deduction* by Gerhard Gentzen, 1969 [11]. Basically we *we can prove that iPC is decidable* using the same method that proves PC is decidable. These are important results by some of the best logicians. We draw heavily on Kleene's book *Introduction to Meta-Mathematics*, ISHI Press International, 2009 [14].

1 Intuitionistic Propositional Logic Axioms

We recall the axioms given in Joan Moschovakis' 2018 article in the *Stanford Encyclopedia of Mathematics*, SEP [19]. These axioms are listed below.

1. $A \Rightarrow (B \Rightarrow A)$.
2. $(A \Rightarrow B) \Rightarrow ((A \Rightarrow (B \Rightarrow C)) \Rightarrow (A \Rightarrow C)$
3. $(A \Rightarrow (B \Rightarrow (A \ \& \ B))$
4. $(A \ \& \ B) \Rightarrow A$
5. $(A \ \& \ B) \Rightarrow B$
6. $A \Rightarrow (A \vee B)$
7. $B \Rightarrow (A \vee B)$
8. $(A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \vee B) \Rightarrow C))$
9. $(A \Rightarrow B) \Rightarrow ((A \Rightarrow \neg B) \Rightarrow \neg A)$
10. $\neg A \Rightarrow (A \Rightarrow B)$

Next we give Heyting's eleven axioms for iPC in his North-Holland book *Intuitionism, An Introduction* [12].

1. $P \Rightarrow (P \vee Q)$.
2. $(P \vee Q) \Rightarrow (Q \vee P)$
3. $((P \Rightarrow R)\ \&\ (Q \Rightarrow R)) \Rightarrow ((P \vee Q) \Rightarrow R)$
4. $(P \Rightarrow (P\ \&\ P)$
5. $((P \Rightarrow Q)\ \&\ (Q \Rightarrow R)) \Rightarrow ((P \vee Q) \Rightarrow R)$
6. $Q \Rightarrow (P \Rightarrow Q)$
7. $(P\ \&\ (P \Rightarrow Q)) \Rightarrow Q$
8. $(P\ \&\ Q) \Rightarrow (Q\ \&\ P)$
9. $(P \Rightarrow Q) \Rightarrow ((P\ \&\ R) \Rightarrow (Q\ \&\ R)$
10. $\neg P \Rightarrow (P \Rightarrow Q)$
11. $((P \Rightarrow Q)\ \&\ (P \Rightarrow \neg Q)) \Rightarrow \neg P$
12. $((P \Rightarrow R)\ \&\ (Q \Rightarrow R)) \Rightarrow ((P \vee Q) \Rightarrow R)$

Finally we present the axioms given Stephen Kuznetsov, a student at the University of Pennsylvania interested in intuitionistic logic.

1. $A \Rightarrow (B \Rightarrow A)$.
2. $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$
3. $(A\ \&\ B) \Rightarrow A$
4. $(A\ \&\ B) \Rightarrow B$
5. $(A \Rightarrow (B \Rightarrow (A\ \&\ B))$
6. $(A \Rightarrow (A \vee B)$
7. $(B \Rightarrow (A \vee B)$
8. $(A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \vee B) \Rightarrow C)$
9. $\bot \Rightarrow A$
10. There is one *inference rule*: $A\ \ (A \Rightarrow B)/B$

We now present the functional programs that are said to *realize* these axioms.

1. $A \Rightarrow (B \Rightarrow A)$ by $\lambda(a.\lambda(b.a))$
2. $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$
   by $\lambda(f.\lambda(ab.\lambda a.f(a)(ab(a))))$
3. $(A \ \& \ B) \Rightarrow A$ by $\lambda(ab.first(ab))$
4. $(A \ \& \ B) \Rightarrow B$ by $\lambda(ab.second(ab))$
5. $(A \Rightarrow (B \Rightarrow (A \ \& \ B))$ by $\lambda(a.\lambda(b. < a, b >))$
6. $(A \Rightarrow (A \vee B)$ by $\lambda(a.inl(a))$
7. $(B \Rightarrow (A \vee B)$ by $\lambda(b.inr(b))$
8. $(A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \vee B) \Rightarrow C)$ by $\lambda(f.\lambda(g.\lambda(ab.decide(ab.a.f(a); b.g(b))))))$
9. $\bot \Rightarrow A$ by $\lambda(x.any(x))$
10. There is one *inference rule*: $A \ \ (A \Rightarrow B)/B$

2 Proof Assistants

Proof assistants are used to formalize and implement results in mathematics, to help researchers precisely formulate open problems and solve them, and to discover interesting new problems and conjectures. Results with computational content can be *executed*. Most results can be executed on laptops, but some rely on a cluster of processors to help find the proof. As a very simple example, if we formally prove that for any natural number n there is a least prime number p greater than n, then we can execute the proof on a number and see exactly what the prime number is. As another example, if we constructively prove that an equation has a real number solution, then we can compute that real number to arbitrary precision by executing the proof. This method of using *constructive proofs as programs* [1] can be provided for number theory, for significant parts of real and complex analysis, for geometry, for constructive and intuitionistic topology and for results in other branches of mathematics. There are now several excellent proof assistants and informative web pages that describe them. We have an especially close relationship with the *Coq proof assistant* because we use it to prove the Nuprl rules correct. We call this a *dual prover technology*.

We were not expecting to create a proof assistant for full *intuitionistic analysis* in the style of Luitzen Egbertus Jan Brouwer, *L.E.J. Brouwer* for short.[1] One reason is that Brouwer proved that there is no way to implement *all* of intuitionistic mathematics nor even *formalize* all of it. Moreover without using diagonalization techniques, Brouwer proved that the intuitionistic reals are not enumerable. We present that elegant argument later. There are many other results in his intuitionistic mathematics that *have not been formalized* and almost surely can be. Those are results of special interest.

[1] His friends called him "Bertus."

Next we mention other topics of this course. Our approach is informed by a modern reaction to the quotation from Leopold Kronecker ("Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk" ("The loving God made the integers, all else is people's work."). In the coming age of *intelligent machines*, we will say something more like this: "In the 20th century humans built *intelligent machines*, all else is our work with our colleagues, students, and these machines."

3 Introduction and Background

Since 2014 we have been formalizing elements of L.E.J.Brouwer's intuitionistic mathematics and implementing them with the *Nuprl proof assistant*. Results are encouraging. For instance, *intuitionistic calculus* turns out to be technically simpler than classical calculus. The *Continuity Principle* is very expressive and useful, yet it contradicts classical mathematics. Brouwer's *Bar Induction* is a clean and elegant form of induction that is classically valid as well. We present formal versions of these key principles and theorems. The fact that continuous functions from closed intervals of the reals into the reals are uniformly continuous 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 this course.

The task of implementing mathematics brings together themes at the interface between mathematics and computer science. Computer science (CS) is a young and vibrant branch of science with roots in mathematics. 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 [9]. 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 [6], because they are closely related and they work together. There are other excellent proof assistants, some 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, human imagination. We will explore all four themes.

Starting circa 1985 the Nuprl proof assistant was used to implement *elements of constructive mathematics* including constructive *real analysis* as presented in Bishop's book *Foundations of Constructive Analysis* [3] and in its enrichment in the Bishop and Bridges version *Constructive Analysis* [4]. 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* [6]. For various reasons, that book has been cited almost every week since it was first published by *Prentice-Hall*. A large team of doctoral students contributed to the proof assistant and the book. Our current effort to implement intuitionistic mathematics is undertaken with a smaller group of very experienced computer science researchers. They are the co-authors on the five articles that document our implementation of core intuitionistic

mathematics at Cornell University from 2014 until 2019.

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* [8]. 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.

We began writing these notes to determine the plausibility of creating a textbook for logic in computer science with precisely the focus of our title. We are able to draw on Brouwer's extensive writings, also on the book *The Foundations of Intuitionistic Mathematics* [15] by Stephen Cole Kleene and Richard E. Vesley, the book by Heyting [12, 13] and the books by A.S. Troelstra [21, 22, 23] and by Dirk van Dalen [24]. We also depend on these resources [16, 14] as well as [5]. Moreover, Kleene's former PhD student, Professor Joan Rand Moschovakis, has written incisively on intuitionism [18, 17]. Her twenty eight page 1999 article with co-author Garyfallia Vafeiadou entitled *Intuitionistic Mathematics and Logic* [20] 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.

The first author of this book was also a PhD student of S. C. Kleene. One of our goals is to relate several intrinsically computer science ideas to uniquely intuitionistic concepts such as *free choice sequences*, realizers for *Bar Induction*, and for the *Continuity Principle.* We learned these ideas from the indispensable book *The Foundations of Intuitionistic Mathematics* by Kleene and Vesley [15]. We subsequently studied many publications of Brouwer, Heyting, van Dalen, van Atten as well as several other modern intuitionists cited in the text. We have explored Brouwer's *Creating Subject* arguments, but do not cover them here.

Before we focus on intuitionism, we want to point out the advantages of creating a book. Our previous book, *Implementing Mathematics with the Nuprl Proof Development System* [6], is still frequently cited after 34 years. In that book and in further work, we implemented several theorems in Errett Bishop's book *Foundations of Constructive Analysis* [3] including some new material added by Douglas Bridges in the updated book entitled *Constructive Analysis* [4]. These implementations allowed us to accomplish tasks in real analysis by *executing these formal proofs.* Our implementation *brought the book to life.* By now that book has well over 1,200 citations. We wrote something useful which we can now apply to intuitionistic mathematics.

We have 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* [10].

http://www.nuprl.org/MathLibrary/ConstructiveAnalysis/
http://nuprl-web.cs.cornell.edu/MathLibrary/TheElements/TheElements.html

3.1 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 [7] 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] E. Bishop. *Foundations of Constructive Analysis*. McGraw Hill, NY, 1967.

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

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

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

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

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

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

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

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

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

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

[19] Joan Rand Moschovakis. Intuitionistic logic. *Stanford Encyclopedia of Philosophy*, 2018.

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

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

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

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

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