The Story of Logic – CS/Math 4860 Class Notes Fall 2019

Robert L. Constable

September 3, 2019

1 Classical Period

These brief notes on the history of logic are meant to be an easily understood basis for stating briefly and informally some of the great ideas in logic. Appreciating how the basic concepts were discovered and why they were needed is an accessible basis for a technical understanding of these concepts. For many students, this historical approach has been a route by which they have come to deeply understand and appreciate logic.

Logic is one of the oldest continuously studied academic subjects, and while investigating its central questions, Alan M. Turing gave birth to computer science in 1936 as you will see toward the end of these notes. He and his thesis advisor at Princeton University, Alonzo Church, laid theoretical foundations for computer science that continue to frame some of the deepest questions and provide the conceptual tools for important applications. Indeed, their ideas underlie one of the most enduring contributions of computer science to intellectual history, the discovery that we can program computers to assist us in solving mathematical and scientific problems that we could not solve without them. It is no longer in doubt that computers can be programmed to perform high level mental operations that were once regarded as the pinnacle of human mental activity. Let us start as far back as is historically sensible as far as we know.

1.1 Epimenides of Crete circa 500 BCE

Epimenides is reputed to have posed a version of the Liar's Paradox, "All Cretans are liars" said a known Cretan. This paradox is mentioned in the Bible (Paul's epistle to Titus). A more modern related paradox asks about the truth or falsity of the statement:

This sentence is false.

Clearly this apparently declarative sentence cannot have a truth value as normally understood, i.e. to be either true or false. To this day logicians do not have a definitive accounting of the meaning of this paradoxical statement and others like it. In the precise logical languages we study in this course, no such sentence is allowed to be "well formed". This suggests that natural language cannot be treated as a precise logical language.

A related question that was only considered much later is what to make of this sentence:

This sentence is meaningless.

1.2 Aristotle

Aristotle wrote one of the most influential books on logic, called by his followers the *Organon*, circa 330 BCE (the date is my approximation of the date when it might have been written, probably at the Lyceum academy in Athens during the time of Ptolemy I). Up until the time of Kant it was understood to be a definitive treatment of logic for about two thousand one hundred years (2,100 years). According to Aristotle (see `http://plato.stanford.edu/entries/aristotle-logic/`):

Prior Analytics I.2, 24b 18-20 Deduction is speech (logos) in which certain things having been supposed, something different from these supposed results follows of necessity from their being so.

Speech consists of sentences, each has the form of a *subject* and a *predicate*. We continue to this day to discuss grammar in these terms. When we say Socrates is mortal, the subject is Socrates and the predicate is is-mortal. For Aristotle, subjects and predicates are terms in his logic, either individual terms like Socrates or universal terms such as is mortal.

Aristotle's logic is about deduction from premises to conclusion. The notion

that the conclusion "follows of necessity" is the modern idea of logical consequence which we study in this course. His deductions are presented in a very specific form called a *syllogism*. Here is a famous example.

All Men are Mortal **premise one**
Socrates is a Man **premise two**
==========
Socrates is Mortal **conclusion**

Aristotle worked out a detailed account of syllogisms including rules for creating an argument from a sequence of them. There are good examples in a systematic modern notation available on-line, as in `http://plato.stanford.edu/entries/aristotle-logic/` There is something very modern in his approach in that he proves properties of his logical system using a more informal style. We call such a study of a logical system, *metalogic*.
Aristotle used his logic for deducing scientific results, and he famously said that:

1. Whatever is scientifically known must be demonstrated.
2. The premises of a demonstration must be scientifically known.

This account allowed "agnostics" to argue that scientific knowledge is impossible because the premises must be demonstrated as well, so there is an *infinite regress* in trying to find even one statement which is scientifically known. Aristotle claimed that the demonstration process ends when we reach certain premises which we accept based on a state of mind called *nous* (intuition, insight, intellect, intelligence, understanding). He claimed that the human mind was capable of certain intuitions needed for science. We will see this idea emerge with great force in the philosophy of mathematics and to have a significant impact on the practice of computer science.

1.3 Euclid

Euclid's *Elements* (circa 306 BC) are not always considered in the history of logic, but they are critical in understanding the use of logic in mathematics and computer science. I trust that readers will be familiar with Euclidean

geometry as presented in secondary school. As with Euclid, school textbooks and teachings begin with definitions of the objects such as points, lines, line segments, straight lines, figures, circles, plane angles, and so forth for 23 definitions in the *Elements*. We can think of these definitions as also establishing types or classes, e.g. the type of Points, the type of Figures, Circles (a subtype of Figures), and so forth. Programmers use such types or classes when they write geometric algorithms.
Euclid stated five basic axioms (postulates) about geometry, but the first three don't look like the subject predicate declarative sentences of Aristotle. Here are simple versions of the first four axioms.

1. To draw a straight line from any point to any point.
2. To extend a line to a straight line.
3. To draw a circle with given center and radius.
4. All right angles are equal.

The first three axioms are *statements about what can be constructed*. Euclid is saying that it is possible to perform certain basic constructions, and to understand the axioms is to know how to perform the constructions.
Euclid's famous fifth axiom is paraphrased this way.

5. If a straight line falling on two straight lines makes interior angles on the same side less than two right angles, then the two straight lines meet at a point on the side where the angles are less than right angles.

It is interesting that various theorems of Euclid such as *To bisect an angle*, can easily be understood by programmers as an algorithmic task. The result is a construction which "solves the problem."

Ever since Descartes introduced analytic geometry and translated geometric questions into analysis, mathematicians have realized that there are other rigorous ways to study geometry beyond Euclid's. Nevertheless Euclidean geometry has been the subject of sustained axiomatic and logical analysis right up to the present. New proofs in the style of Euclid are being published even now such as the Steiner-Lehmus theorem that if two angle bisectors of a

triangle are equal in length, then the triangles must be isosceles see [GM63]. Two of the most cited works are Hilbert's book *Foundations of Geometry* and Tarski's decision procedure for geometry based on the first-order theory of real closed fields, see [TG99] and von Plato's axiomatization in type theory [von95].

2 Transition

2.1 Leibniz

The mathematician Leibniz co-invented the calculus with Newton, and he also introduced a conceptual advance in our understanding of the scope and potential of logic, a conception going beyond that of Aristotle in some important ways. He believed that logic was a science that contained concepts essential to all other sciences.

Leibniz also is a major node in the genealogical tree of mathematicians and logicians. Many logicians since the 1660s can trace their academic ancestry back to Leibniz, including me.

Here is a quotation by Leibniz from his *De Arte Combinatoria*, 1666. It is taken from an article by J. Bates, T. Knoblock and me entitled Writing Programs that Construct Proofs, *Journal of Automated Reasoning*, vol. 1, no. 3, pp. 285-326, 1984.

"A term is the subject or predicate of a categorical proposition .... Let there be assigned to any term its symbolic number, to be used in calculation as the term itself is used in reasoning. I chose number whilst writing; in due course I will adapt other signs ... for the moment, however, numbers are the of the greatest use ... because everything is certain and determinate in the case of concepts, as it is in the case of numbers. The one rule for discovering suitable symbolic numbers is this; that when the concept of a given term is composed directly of the concept of two or more other terms, then the symbolic number of the given term should be produced by multiplying together the symbolic numbers of the terms which compose the concept of the given term. In this way we shall be able to discover and prove by our calculus at

any rate all the propositions which can be proved without the analysis of what has temporarily been assumed to be prime by means of numbers. We can judge immediately whether propositions presented to us are proved, and that which others could hardly do with the greatest mental labor and good fortune, we can produce with guidance of symbols alone .... As a result of this, we shall be able to show within a century what many thousands of years would hardly have granted to mortals otherwise."

2.2 Boole

Boole wrote *The Laws of Thought* in 1854. On page 27 of the Dover 1958 reprinting we see the propositional calculus similar to how it is taught today but using the symbols from algebra, +, -, $\times$, and =. The sign + corresponds to what we now call exclusive or. In this major advance over Aristotle, Boole studies formulas built out of logical operators whose meaning is defined in terms of the truth values, say 0 and 1 for false and true. Boole's presentation is inspired by the laws of algebra, and he hopes to present methods for solving logical equations. The propositional calculus is also called *Boolean algebra* when presented in Boole's original style. The laws of this algebra are especially important in analyzing binary computer circuits (with values 0 and 1). In honor of Boole we also call the truth values *Booleans* and denote them by $\mathbb{B}$.

3 The Nineteenth Century Crisis in Analysis

During the 1800's, analysis (advanced calculus) rapidly developed, despite the fact that many of the basic concepts such as function, sequence, set, continuity, convergence, infinite series, infinitesimal, etc. were not clear. For example, Cauchy believed that every continuous function was differentiable and that the infinite sum of a sequence of continuous functions was continuous both of which are false. Yet he was the person who gave a precise definition of continuity and was one of the leading figures in analysis. Many mathematicians believed that an infinite series made sense as a mathematical object even if it did not converge, *but they were not sure what these series were.* Even Euler believed that every infinite series was the result of expanding a unique finite formula. Gradually these concepts were clarified by many gifted

mathematicians such as Abel, Bolzano, Borel, Bernoulli, Dedekind, Dirichlet, Fourier, Heine, Jordan, Lebesgue, Poincare, Riemann, Weierstrass, and others for whom many of the theorems of analysis are named.

The Arithmetization of Analysis During this period there were significant advances in understanding the concept of a *real number*, the very foundation of analysis. One approach to real numbers since Descartes was to ground them in geometry and consider real numbers as points on the line. But with the discovery of non-Euclidean geometries, people began to think that geometric truths were not absolute and perhaps not a secure basis for understanding the reals. *Gauss expressed the opinion that only the natural numbers were a source of absolute mathematical truth.*

Cauchy was able to reduce the concept of a real number to that of a sequence of rational numbers that converges, so called *Cauchy sequences*. Thus he reduced the idea of real number to the idea of functions and rational numbers.

3.1 Frege

The most significant achievement in logic after Aristotle is Frege's publication in 1879 of his *Begriffsschrift* (concept-script, concept-writing, idea-writing, ideography). It is a booklet of 88 pages that changed logic forever. `http://plato.stanford.edu/entries/frege/`

In this work he invented what we now call *first-order logic*, FOL, a major topic of this course.[1] He started by making a case for advancing Leibniz's conception of logic and arguing that the Aristotelian conception was too narrowly tied to natural language. He said: "These derivations (in notation) from what is traditional find their justification in the fact that logic has hitherto always followed ordinary language and grammar too closely. In particular, I believe that the replacement of the concepts subject and predicate by argument and function, respectively, will stand the test of time."

[1]The textbook by Raymond M. Smullyan that we have used for years in teaching logic is entitled *First-Order Logic*. It is only 158 pages of text and is extremely clear.

Frege was attempting to understand the notion of a sequence very precisely. The gradual arithmetization of analysis and the calculus (see below), had reduced the idea of a real number to the notion of sequences of rational numbers, and rational numbers could be reduced to pairs of integers, and integers could be reduced to natural numbers $\mathbb{N}$. This reduction focused attention on defining natural numbers and the notion of a sequence. There were questions about how to prove properties of sequences.

The most reliable way of carrying out a proof, obviously, is to follow pure logic, a way that, "disregarding the particular characteristics of objects, depends solely on those laws upon which all knowledge rests. He goes on to say how hard it is to do this using ordinary language. "To prevent anything intuitive from penetrating here unnoticed, I had to bend every effort to keep the chain of inferences free of gaps. In attempting to comply with this requirement in the strictest possible way I found the inadequacy of language to be an obstacle. This deficiency led me to the idea of the present ideography." Here is how he elaborates on Leibniz:

"Leibniz, too, recognized and perhaps overrated the advantages of an adequate system of notation. His idea of a universal characteristic, of a calculus philosophicus or ratiocinator, was go gigantic that the attempt to realize it could not go beyond the bare preliminaries. The enthusiasm that seized its originator when he contemplated the immense increase in the intellectual power of mankind that a system of notation directly appropriate to objects themselves would bring about led him to underestimate the difficulties that stand in the way of such an enterprise. But, even if this worthy goal cannot be reached in one leap, we need not despair of a slow, step-by-step approximation. When a problem appears to be unsolvable in its full generality, one should temporarily restrict it; perhaps in can then be conquered by a gradual advance."
One of the important innovations in *Begriffsschrift* is the notion of judgements. Frege distinguished between judging that a formula is true and indicating that it has content and could be judged. The content indicator is written $-A$ for a purported formula A and the truth judgement is written with the turnstile $\vdash$ to claim that A is true or at least worth trying to prove. We see $\vdash$ A in the above document (we sometimes write $-A$ and $\vdash A$ respectively for A having propositional content and A being judged for truth, thus wanting proof).

3.2 Peano

In 1889 Peano published his small book *The Principles of Arithmetic* in which he showed how to use symbolic logic to axiomatize a theory of natural numbers, now known as Peano Arithmetic. He used Boole's ideas for propositional logic and Schröder's notation for quantifiers such as "for all" and "there exists".

3.3 Cantor

By 1899 Cantor knew there was a paradox in his conception of set theory under the assumption that there is a set M of all sets (Menge is the German word for set). Cantor had defined the cardinality of any set A, denoted $|A|$. He proved his famous theorem that the cardinality of the set of all subsets of any set A, call it Pow(A), has a larger cardinality than A, that is $|Pow(A)| > |A|$. But if M is the set of all sets, then $Pow(M)$ is a subset of M, so $|Pow(M)| < |M|$, which contradicts his theorem. His conclusion is that there cant be a set M of all sets, such a set is in some ways "too big." This is Cantor's Paradox.

3.4 Russell

In 1902, Russell discovered a paradox along similar lines as Cantor's, but much simpler to explain. He considered the set of all sets that do not contain themselves, call it R and define it as $R = \{x \mid not(x \in x)\}$. Now Russell asks whether $R \in R$. We can see immediately that if $R \in R$, then by definition of R, we know $not(R \in R)$. If $not(R \in R)$, then by definition of R, $R \in R$. Since according to classical logic either R belongs to R or it does not, and since each possibility leads to a contradiction, we have a contradiction in classical mathematics if we assume there is such a set as R. This is known as *Russell's Paradox*.

In trying to sort out the reasons behind this paradox, Russell was led to formulate his famous Theory of Types in the 1908 book *Mathematical Logic Based on a Theory of Types* [16]. Then in the period from 1910 to 1925

Whitehead and Russell wrote and published a three volume book based on this type theory and designed to be a secure logical foundation for all of mathematics entitled *Principia Mathematica* (PM) [20]. We will examine aspects of this type theory during the course, and we will use type theory from the beginning in our informal mathematics because the notion of type is now central in computer science, owing largely to early research in Britain, France, Sweden and the US as well as to the increasing role of type systems in the design of programming languages.

3.5 Zermelo

Meanwhile in 1908 the German mathematician Zermelo published an influential paper Investigations in the foundations of set theory in which he organized a set theory, now called Z, around a collection of methods for forming sets that he thought were safe from the paradoxes of Cantor, Russell and others. He did not allow the set of all sets to be a set, and he restricted the methods of building sets starting from a single infinite set used to build the natural numbers (axiom of infinity). Other sets could be built using the *separation axiom*: if S is a set and P a definite property on sets, then $\{x : S \mid P(x)\}$ is a set. In 1922 this notion of "definite property" was made more precise by the logicians Fraenkel and Skolem, and by Weyl. It is now standard practice to require that P be a formula of first-order logic using the primitive concept of set membership normally denoted by epsilon, $\in$. Fraenkel and Skölem also suggested a powerful new axiom deemed by many to be safe, called the axiom of replacement. This set theory is now called ZF set theory and is considered by many mathematicians to be the "standard foundation" for modern mathematics; however, it is not able to express ideas from category theory and type theory without extension.

3.6 Knönecker

Krönecker was a distinguished German mathematician known for his work in algebraic and analytic number theory. He is also known for being opposed to using the concept of a completed infinite set in mathematical reasoning as Cantor and others were starting to do. He agreed with Gauss that the idea of an infinite set was just a manner of speaking about collections such as the natural numbers, $0, 1, 2, 3, \ldots$ which can be continued without end because

given any number n, we can construct a larger number by adding one to it. He was in particular opposed to Cantor's work on set theory, but he also disagreed with methods of proof that did not produce concrete answers. He strongly favored computational methods and explicit constructions.

Krönecker is mentioned in Bell's *Men of Mathematics* as being "viciously opposed" to Cantor and others who proved results about infinite sets as if they were completed totalities, but this appears to be a considerable exaggeration according to Edwards, *Essays in Constructive Mathematics* [7]. What is true is that in 1886 he made an after dinner speech in which he said "God made the integers, all else is the work of man" as a way of expressing his interest in constructions. Here is the German for what he said:

Krönecker: *Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk.*

We will look next at a mathematician who did openly attack non-computational methods and who was in contention for being the leading mathematician of his day.

3.7 Brouwer

In his 1907 doctoral dissertation, *On the Foundations of Mathematics*, Brouwer provided a meaning for mathematical statements based on mental constructions. These constructions are intuitively known to be effective for basic mathematical tasks. This philosophical stance is known as *intuitionism*, and Brouwer was interested in building mathematics according to this philosophy. Brouwer believed that the crisis in the foundations of analysis was due to mathematicians not understanding the full extent of constructive methods. In particular he believed that our mental constructions are the proper justification for logic and that they do not justify the full logic of Aristotle which is based on affirmation and denial of sentences. Restating this in terms of propositional or Boolean logic, one of the fundamental laws that is not justified according to Brouwer is the law of excluded middle, P or *not* P, for any proposition P. For Brouwer, to assert P is to know how to prove it, and he could imagine propositions which we could never prove or disprove.

Brouwer believed that logic was the study of a particular subset of abstract constructions but that it played no special role in the foundations of mathematics, it was simply a form of mathematics.

Brouwer believed that our mathematical intuitions concern two basic aspects of mathematics, the discrete and the continuous. Our intuitions about discreteness and counting discrete objects give rise to number theory, and our intuitions about time give rise to the notion of continuity and real numbers. He believed that mathematicians had not recognized the rich constructions need for understanding real numbers, including for computing with real numbers. He was determined to study these constructions and thus settle the disputed issues in analysis and the theory or real numbers. But first he decided he should establish himself as the best mathematician in the world. He proceeded to develop point set topology and proved his famous and widely used fixed point theorem. He attracted several concerts and fellow travelers, and when one of the most promising young mathematicians, Weyl, became a follower of Brouwer, another contender for "world's best mathematician, Hilbert, entered the fray in a "frog and mouse war" with Brouwer. He is famous for saying that "we can know and must know" the truth or falsity of any mathematical statement.

3.8 Heyting

Brouwer was dismissive of formal logic in his early years (but later became and editor with Beth and Heyting of the very influential series *Studies in Logic*), so he did not write down a logic that could be used for intuitionistic reasoning. However in 1930 Heyting wrote The formal rules of intuitionistic logic. This was the creation of another branch of logical systems that respected intuitionistic principles. Heyting also formalized a theory of arithmetic similar to *Peano Arithmetic* (PA) but using intuitionistic logic; it is called *Heyting Arithmetic* (HA). We will study these logics in this course and relate them. In 1925 Kolmogorov wrote a precursor work entitled On the principle of excluded middle, in which he axiomatized a fragment of intuitionistic logic.

A key feature of these new accounts of logic is that the notion of truth is replaced by the notion of knowledge. The logic is centered on explaining how we can know propositions. This semantics is known as BHK semantics and we will study it in depth. It leads to the major new principle of logic called the propositions-as-types principle which we will also study in depth.

3.9 Hilbert

Hilbert had been thinking about the foundations of mathematics since 1899 when he wrote *The Foundations of Geometry*. In 1904 he began thinking more broadly about the foundations of logic and arithmetic, but it was not until 1925 that he began serious work on these topics, sketched in his article "On the Infinite". He responded to *Principia Mathematica* by showing how to create a completely formal logical theory. Nowadays we mean by this a theory that can be implemented in software, but Hilbert captured this concept before we knew about computers. He treated a formal theory as the basis for a meaningless game played according to strict rules; the goal of the game was to prove theorems. The important property of the game was that it was consistent; that is, it would not be possible to prove a theorem P and also prove not P.

Hilbert was willing to admit that the idea of actual infinity might be meaningless, merely a convenient way of thinking, an idealization. He imagined that by his method of formalizing theories and analyzing them in a constructive metatheory, he could justify the most free-ranging imagination, the creation of ideal worlds in which there were actual infinities. He called such a world a paradise and said that people like Brouwer could not "drive us out of this paradise" because we could use strict methods of proof, constructive methods, to show that the game of set theory or the game of type theory was consistent. He even believed that the game could settle all questions; it could be complete in the sense that all true theorems of the theory could be proved. Hilbert's program for saving mathematics was bold and inspiring. His program was also a way to save mathematics as it was being practiced. He saw himself as a savior of mathematics from critics like Kronecker, Brouwer, and his brilliant former student Weyl.

In the period from 1925 until 1940 mathematics was at a tipping point. It could go with Brouwer and his many followers and become constructive mathematics or it could remain the classical mathematics that Hilbert treasured and advocated, open to very non-constructive methods of the kind Hilbert had used in his ascendancy into the group of the very best mathematicians of his day, a group that included at least Brouwer and Poincare as well. //

4 Modern Period

4.1 Gödel

In 1931 Gödel published his article "On formally undecidable propositions of Principia Mathematica" and related systems. He proved that *Principia Mathematica* (PM) could not be complete if it was consistent; that is, unless PM could prove everything, there was a true sentence of that theory which was not provable.

Gödel also proved in his second major theorem that PM could not prove its own consistency and that his results held for any system strong enough to axiomatize elementary arithmetic, such as *Peano Arithmetic* (PA). This result was a blow to Hilbert's idea that the proof of consistency of systems like PM and PA, an issue for what he called *metamathemtics* [11], could be done constructively, even in a finitary manner – much more strictly computational than intuitionistic mathematics. Gödel showed that this part of the Hilbert program was not possible, and that Hilbert's view of a paradise of imaginary thought could not be supported by metamathematical results of the kind he imagined.

These surprising discoveries seemed to put an end to Hilbert's program for saving mathematics. They were stunning results that made Gödel the most famous logician of the age and got him a life time job at the Institute for Advanced Study in Princeton. These results also showed that perhaps Brouwer was right and there would be statements in mathematics that we could never prove even though they might be true in some sense, say in the sense that we could never disprove them either. Gödel became fascinated by the mathematics of Brouwer and made substantial contributions to the metamathematical study of these systems, results that we will teach in this course.

4.2 Church

By 1940 Gödel had moved from Austria to the Institute for Advanced Study in Princeton. Alonzo Church was a professor of mathematics at Princeton University where for the year 1937 Alan Turing had also come to study logic with Church. In 1936 Church had discovered an unsolvable prob-

lem of elementary number theory and published the short note "A note on the Entscheidungsproblem". The Entscheidungsproblem, or decision problem, was one of the famous open problems that Hilbert had posed in 1928. Turing had also solved this problem as we will discuss below. `http://en.wikipedia.org/wiki/Hilbert`
Church's solution was in terms of his lambda calculus, a formalism later shown to be equivalent to Turing machines and all other known formalisms for expressing the computable functions. In a sense the lambda calculus was the first programming language, and indeed it led John McCarthy and his students to create the programming language Lisp [15] circa 1960.

Church was the doctoral thesis advisor for a very large number of logicians. According to the Mathematics Genealogy Project `http://genealogy.math.ndsu.nodak.edu/` Church has produced 2,697 descendants, a number that keeps growing. Among them are some of the most famous logicians in the world such as Alan Turing (1938), Dana Scott (1958), a Turing Award winner, Michael Rabin (1957) a Turing Award winner, Simon Kochen (1959), Stephen Kleene (1934), J. B. Rosser (1934), Martin Davis (1950), and Raymond Smullyan (1959), the author of our textbook. Kleene went on to write on of the most influential textbooks in logic, *Introduction to Metamathematics* [11] and to develop a *realizability semantics* for Brouwer's notion of constructive truth [13].

Church is also known for developing the Simple Theory of Types [4] which is the logical system implemented in the HOL theorem prover and widely used in computer science applications, starting with hardware verification.

4.3 Turing

In 1936 Turing published his ground breaking paper "On computable numbers with an application to the Entscheidungsproblem". He discovered this result before he came to Princeton to work with Church. In this paper he defines computable real numbers using what has become known as Turing Machines. Turing's model of computation is still widely taught, and his unsolvable problem is taught to computer science freshmen around the world, the *halting problem*.

Turing showed that no Turing Machine can solve the problem of whether a given such machine will halt on a given input or "on blank tape". This model of computation is very natural and convincing as a general model of how people and machines compute in principle. Gödel was extremely impressed by this model, even though he imagined a competing idea, the so called Herbrand-Gödel equations for general recursive functions. He said that Turing's model showed that the idea of computability was absolute, that is independent of any particular logic.

At the same time, Gödel did not appreciate Church's lambda calculus which also defined the class of all computable functions. Indeed the famous Church-Turing Thesis is the claim that these formalisms define what mathematicians mean by effectively computable functions. The difference is that Turing machines are a simple machine model and the lambda calculus is a very high-level functional programming language. Church put to Turing the task of proving that the lambda calculus could be translated into Turing Machines. Thus Turing was assigned the task of building the first compiler for a high level language. Perhaps from that moment on he became a computer scientist. In any case, he went on to contribute broadly to what we now consider to be computer science.

During World War II, Turing worked at Bletchley Park in England as a code breaker. His role during the war was secret until the 1970's. Since then much has been written about his heroic role. One of the most accessible books is *The Man Who Knew Too Much* by David Leavitt.

4.4 Kleene

Stephen Cole Kleene was a student of Alonzo Church who made fundamental connections between Brouwer's intuitionistic mathematics and the notion of Church-Turing computable functions [13] by formalizing Brouwer's notion of computability. He used these ideas to give a semantics for intuitionistic logic. Kleene visited Brouwer to learn first hand and in depth about intuitionistic mathematics. His book with his student Vesley laid rigorous foundations for a formal theory of intuitionistic mathematics in their book *Foundations of Intuitionistic Mathematics* [12]. Using this book, we were able to enrich our constructive type theory [5] to become a fully intuitionistic type theory

[1]. This was accomplished in a series of articles accompanied by a large implementation effort. As far as we know this is the first implementation of intuitionistic logic and mathematics using computers. It has led to many new research questions and results in both mathematics and computer science. The use of nondeterministic computation in intuitionistic mathematics establishes links to one of the most famous open problems in computer science and mathematics, is the class P of polynomial time deterministic computations equal to the class of NP of non-deterministic polynomial time computations, symbolically does P = NP?

4.5 Bishop

In 1967 Bishop published his extremely influential book on constructive mathematics called *Foundations of Constructive Analysis* [2]. He wrote more about the underlying logic in his article *Mathematics as a Numerical Language* [3] In his book on real and complex analysis he showed that it is possible to develop most of modern analysis without adopting ideas from Brouwer's intuitionism that contradict classical mathematics. For example, in Brouwer's analysis, all functions from the reals to the reals are continuous. Bishop's analysis is entirely consistent with classical analysis, yet all the functions are computable and all the results are proved constructively. This remarkable achievement seems to me a good example of American ingenuity. Bishop enjoys many positive benefits of intuitionism without attacking classical mathematics. His is a philosophy of co-existence. Indeed he was not happy with some of Brouwer's ideas such as free choice sequences.

4.6 Martin-Löf

We now turn to the work of a living logician, Per Martin-Löf who created *Intuitionistic Type Theory (ITT)* in 1982, and wrote about its connection to computer science in the 1982 paper "Constructive Mathematics and Computer Programming" [14]. Per is a colleague who has influenced our work, so I might be biased in my assessment. We will talk about his theory ITT in this course and its impact on Constructive Type Theory (CTT) [5] which was developed at Cornell as an extension of ITT more suitable as a foundation for computing. The goal of Per's work was to provide a firm logical founda-

tion to Bishop's analysis. Later as he came to know more computer science, he broadened his goals to providing a semantics for programming languages and logics [14]. but he did not investigate intuitionistic logic beyond making a few critical (and memorably stated) remarks about Brouwer's approach. We will examine these remarks as written in Bishop's book and make some judgements about them.

4.7 Hoare

Another living figure who contributed to the logic of programming was the computer scientist Tony Hoare who in 1969 published his paper "An Axiomatic Basis for Computer Programming" [9]. This paper created what is now called *Hoare Logic*, a logic for reasoning about standard programming constructs such as assignment statements, while loops, and recursive procedures. He also wrote a seminal paper on data types, "Notes on Data Structuring" [10] that could be considered the first applied fragment of type theory and help promote the idea that types are a major organizing concept in programming languages and programming methodology. Hoare had to face the fact that types in programming languages do not mean quite the same thing as they do in mathematics. The functions of type *nat* to *nat* in a programming language are partial functions; they may not terminate. In mathematics the functions from $\mathbb{N}$ to $\mathbb{N}$ all terminate.

4.8 Milner

In September 2009, Robin Milner was also alive and was one of my closest colleagues. He is the inventor of the ML programming language and its type inference algorithm. He published the very influential book *Edinburgh LCF: A Mechanized Logic of Computation* [8] with his colleagues Michael Gordon and Christopher Wadsworth. The theory LCF is a formalization of Dana Scott's logic of partial functions. A long lasting contribution of this book is that most of the major interactive theorem provers in use today (Coq, HOL, HOL-Light, Isabelle, Nuprl, and MetaPRL) all use the LCF architecture and Milners idea of tactics.

4.9 Scott

Dana Scott is a well known logician who was also a major contributor to computer science. His article "Constructive Validity" [17, 18] provided a computational semantics for programming languages. The book by Stoy, *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory* [19] was widely cited and influential. His article "Data types as lattices" [18] is another contribution of logic to computer science.

4.10 Cornell PRL Group

Since 1984 this Cornell research group has been a leader in the subject of Computational Type Theory and its automation. In 1986 we published the book *Implementing Mathematics with the Nuprl Proof Development System* [5] which we will mention in the course. The Nuprl prover described in that book (Nuprl-1) is still actively used in program verification and in formal methods research (Nuprl-5), especially software security and protocol verification. In its current form it validates Brouwer's notion that the space of constructions (computations) is extraordinarily rich and extends beyond the normal computable functions to include concurrent processes and other forms of interaction. It even seems likely that we can justify Brouwer's idea of free choice sequences in applications. `http://www.scholarpedia.org/article/Computational_type_theory`

5 The Future

Thomas Hobbes in 1651 like Leibniz in 1666 already glimpsed the future. Hobbes published *Leviathan* in 1651 and expressed an opinion about artificial life saying "For seeing life is but a motion of Limbs ... why may we not say that all Automata ... have an artificial life." He believed that "life and mind were natural consequences of matter when suitably arranged" said George Dyson in his book *Darwin Among the Machines* [6].

During the course we will examine the role of *proof assistants* for type theory – such as Coq, Isabelle, and Nuprl – and speculate about their capabilities in the next decade. These proof assistants already know logic and are indis-

pensable partners in scientific research.

References

[1] Abhishek Anand and Vincent Rahli. Towards a formally verified proof assistant. In *ITP*, pages 27–44, 2014.

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

[3] E. Bishop. Mathematics as a numerical language. In *Intuitionism and Proof Theory*, pages 53–71. North-Holland, NY, 1970.

[4] Alonzo Church. A formulation of the simple theory of types. *The Journal of Symbolic Logic*, 5:55–68, 1940.

[5] 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.

[6] George B. Dyson. *Darwin Among the Machines: The Evolution of Global Intelligence*. Perseus Books, 1997.

[7] Harold M. Edwards. Introduction to my book "essays in constructive mathematics". In Thierry Coquand, Henri Lombardi, and Marie-Françoise Roy, editors, *Mathematics, Algorithms, Proofs*, number 05021 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2006. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[8] 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.

[9] C. A. R. Hoare. An axiom basis for computer programming. *Communications of the ACM*, 12(10):576–580,583, 1969.

[10] C. A. R. Hoare. Notes on data structuring. In *Structured Programming*. Academic Press, New York, 1972.

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

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

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

[14] Per Martin-Löf. Constructive mathematics and computer programming. In *Proceedings of the Sixth International Congress for Logic, Methodology, and Philosophy of Science*, pages 153–175, Amsterdam, 1982. North Holland.

[15] J. McCarthy et al. *Lisp 1.5 Users Manual*. MIT Press, Cambridge, MA, 1962.

[16] Bertrand Russell. Mathematical logic as based on a theory of types. *Am. J. Math.*, 30:222–62, 1908.

[17] D. Scott. Constructive validity. In D. Lacombe M. Laudelt, editor, *Symposium on Automatic Demonstration*, volume 5(3) of *Lecture Notes in Mathematics*, pages 237–275. Springer-Verlag, NY, 1970.

[18] D. Scott. Data types as lattices. *SIAM J. Comput.*, 5:522–87, 1976.

[19] J.E. Stoy. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory.* MIT Press, Cambridge, MA, 1977.

[20] A.N. Whitehead and B. Russell. *Principia Mathematica*, volume One, Two, Three. Cambridge University Press, 2nd edition, 1925–27.