Logic: Interdisciplinary Adventures in *Mathematics*, *Philosophy*, *Computer Science*, and *Theology*

Harvey Friedman

Logic is *everywhere!*

Logic as the science of reasoning

We do have a relatively deep scientific understanding of at least some key aspects of mathematical reasoning

THE FIVE LOGICAL CONNECTIVES

these support general reasoning
traced back to the great ancient philosopher Aristotle

not	$\neg$
and	$\wedge$
or	$\vee$
if then	$\longrightarrow$
if and only if	$\longleftrightarrow$

- Fred lives in Chicago, or Fred lives in New York $\quad p \vee q$
- If Fred lives in Chicago then Fred lives in a big city. $\quad p \longrightarrow r$
- If Fred lives in New York then Fred lives in a big city. $\quad q \longrightarrow r$
- *Therefore,* Fred lives in a big city. $\quad \therefore r$

Valid inference

In this very limited context, we can mathematically define what we mean by a valid inference, and we can determine whether such an inference is valid by the method of

Truth tables

The two quantifiers

In addition to the sentential connectives, we must work with the two quantifiers

for all $\forall$

there exists $\exists$

Aristotelean Logic makes very limited use of $\forall$. It took another 2200 years to incorporate the five logical connectives and both quantifiers $\forall, \exists$.

Predicate Calculus

There is a grammar for this, resulting in a formal language called Predicate Calculus. Rather than present this grammar, I will just present a group of familiar examples.

— Gottlob Frege
1879

Examples

1. Everybody loves everybody.
2. Somebody loves somebody.
3. Everybody loves somebody.
4. Somebody loves everybody.
5. Everybody is loved by everybody.
6. Somebody is loved by somebody.
7. Everybody is loved by somebody.
8. Somebody is loved by everybody.

Examples

Logically equivalent

1.	Everybody loves everybody.	1.	$(\forall x)\ (\forall y)\ (L(x,y))$.
2.	Somebody loves somebody.	2.	$(\exists x)\ (\exists y)\ (L(x,y))$.
3.	Everybody loves somebody.	3.	$(\forall x)\ (\exists y)\ (L(x,y))$.
4.	Somebody loves everybody.	4.	$(\exists x)\ (\forall y)\ (L(x,y))$.
5.	Everybody is loved by everybody.	5.	$(\forall x)\ (\forall y)\ (L(y,x))$.
6.	Somebody is loved by somebody.	6.	$(\exists x)\ (\exists y)\ (L(y,x))$.
7.	Everybody is loved by somebody.	7.	$(\forall x)\ (\exists y)\ (L(y,x))$.
8.	Somebody is loved by everybody.	8.	$(\exists x)\ (\forall y)\ (L(y,x))$.

Examples

1. Everybody loves everybody.
2. Somebody loves somebody.
3. Everybody loves somebody.
4. Somebody loves everybody.

1. $(\forall x)\ (\forall y)\ (L(x,y))$.
2. $(\exists x)\ (\exists y)\ (L(x,y))$.
3. $(\forall x)\ (\exists y)\ (L(x,y))$.
4. $(\exists x)\ (\forall y)\ (L(x,y))$.

1, 2, 3, 4, 7, 8 are all logically *inequivalent*

7. Everybody is loved by somebody.
8. Somebody is loved by everybody.

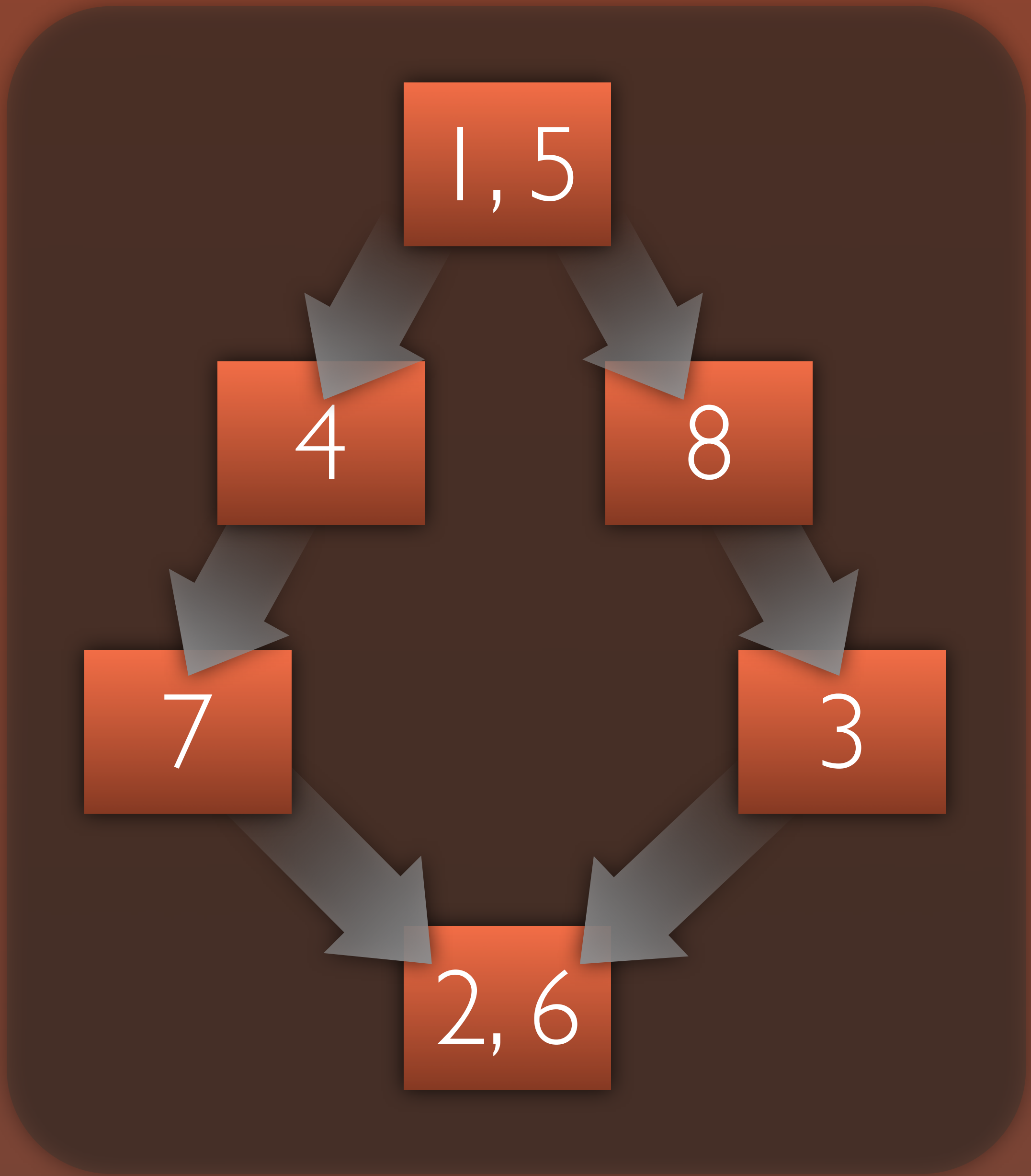
7. $(\forall x)\ (\exists y)\ (L(y,x))$.
8. $(\exists x)\ (\forall y)\ (L(y,x))$.

Logically equivalent

Under any intepretation of the concepts involved, the statements are both true or both false.

The modern treatment of logical equivalence is credited to Alfred Tarski, 1933, 1936.

Logical implications



1. $(\forall x)\ (\forall y)\ (L(x,y))$.
2. $(\exists x)\ (\exists y)\ (L(x,y))$.
3. $(\forall x)\ (\exists y)\ (L(x,y))$.
4. $(\exists x)\ (\forall y)\ (L(x,y))$.
5. $(\forall x)\ (\forall y)\ (L(y,x))$.
6. $(\exists x)\ (\exists y)\ (L(y,x))$.
7. $(\forall x)\ (\exists y)\ (L(y,x))$.
8. $(\exists x)\ (\forall y)\ (L(y,x))$.

An inference in predicate calculus is considered valid if:

> under any interpretation, if all of the premises are true, then the conclusion is true.

Completeness Theorem

— Kurt Gödel, 1930

Every valid inference in predicate calculus can be backed up by a proof using only a fixed finite set of basic axioms and rules of inference.

No Algorithm

— Alonzo Church, 1936

There is no general method for determining whether an inference in the predicate calculus is valid.

Integers

…, -2, -1, 0, 1, 2, …

- Add (+)
- Subtract (-)
- Compare (<, ≤, >, ≥, =, ≠)

Examples

$(\forall \text{ integers } x)(\exists \text{ integer } y)(x < y+y).$

$(\forall \text{ integers } x)(\exists \text{ integer } y)(x = y+y).$

$(\forall \text{ integers } x,y)(x < y \rightarrow (\exists \text{ integer } z)(x < z \wedge z < y)).$

The first is true, whereas the second and third are false.

Linear arithmetic over integers algorithm

— M. Presburger, 1929

There is a general method for determining whether a sentence in the predicate calculus based on the integers, $+,-,<,\leq,>,\geq,=,\neq$, is true.

Fractions

3/4, -10/3, 0/1, …

- Add (+)
- Subtract (-)
- Compare ($<$, $\leq$, $>$, $\geq$, $=$, $\neq$)

Examples

$(\forall \text{ rationals } x)(\exists \text{ rational } y)(x < y+y).$

$(\forall \text{ rationals } x)(\exists \text{ rational } y)(x = y+y).$

$(\forall \text{ rationals } x,y)(x < y \rightarrow (\exists \text{ rational } z)(x < z \wedge z < y)).$

This time, all three sentences are true.

Linear arithmetic over rationals algorithm
(modified *Presburger*)

There is a general method for determining whether a sentence in the predicate calculus based on the rationals, $+,-,<,\leq,>,\geq,=,\neq$, is true.

Real numbers

$\sqrt{2}, \pi, \ldots$

- Add (+)
- Subtract (-)
- Compare ($<, \leq, >, \geq, =, \neq$)

Linear arithmetic over reals

The true sentences of linear arithmetic over the reals are the same as the true sentences of linear arithmetic over the rationals.

Linear arithmetic over integers, rationals, reals together

There is a general method for determining whether a sentence in the predicate calculus based on integers, rationals, reals, $+,-,<,\leq,>,\geq,=,\neq$, is true.

So what happens if we also use multiplication?

Algorithm with $+, -, \bullet, <, \leq, >, \geq, =, \neq$ over *Integers*? **No!!** Kurt Gödel 1931.

Algorithm with $+, -, \bullet, <, \leq, >, \geq, =, \neq$ over *Rationals*? **No!!** Julia Robinson 1949.

Algorithm with $+, -, \bullet, <, \leq, >, \geq, =, \neq$ over *Reals*? **Yes!!** Alfred Tarski 1931, 1948.

What are the standards for a mathematical proof?

1. Axioms and rules of inference for Predicate Calculus.

2. Axioms for constructing mathematical objects.

Unification needed. **Set theory!**

Set theory

Unifies various constructions of mathematical objects.
Two sets are considered equal if and only if they have the same elements.

Founded by Georg Cantor 1874.
Gottlob Frege formulated associated axioms 1893/1903.
Bertrand Russell found a contradiction in Frege's axioms (Russell's Paradox) 1901.
Russell fixed this with his theory of types, 1908, 1910.
Type theory supplanted by formalized set theory:

Zermelo Set Theory (1908)

Extensionality: If two sets have the same elements, then they are equal.

Pairing: Given two sets x,y, there is a set z whose elements are exactly x,y.

Union: Given a set x, there is a set y whose elements are exactly the elements of the elements of x.

Power Set: Given a set x, there is a set y whose elements are exactly the subsets of x.

Separation: Given a set x and a description of a property of sets, there is a set y whose elements are exactly those elements of x which obey that property.

Infinity: There is an infinite set. Many alternative technical formulations can be used here.

Choice: Given a set x whose elements are each nonempty and have no elements in common with each other, there is a set y which has exactly one element in common with these elements of x.

Zermelo set theory axioms easily suffice to develop the main basic infrastructure of mathematics: natural numbers, integers, rationals, reals, complexes, addition, subtraction, multiplication, division, order, ordered pairs, functions, metric spaces, topological spaces, measure spaces, continuity, differentiability, analyticity, etcetera.

Mathematicians sit on top of this, taking it for granted.

ZFC

Zermelo Fraenkel Set Theory,
with the Axiom of Choice

ZFC = Zermelo Set Theory, plus two more axioms:

Foundation: Given a nonempty set x, there is an element y of x which has no elements in common with x.

Replacement: Given a set x, and a description of a unique assignment of a set to all elements of x, there is a set whose elements are exactly these assignments.

Why believe?

Not yet entirely clear.

Observation

All except INFINITY are easily seen to hold in the world of finite sets - by merely using induction.

From Finite to Infinite

Our axioms of set theory arise from transferring the "simple" principles that "obviously hold in the finite sets" to infinite set theory, and adding the axiom of Infinity.

ZFC enough to prove or refute all statements in mathematics?

First Incompleteness Theorem

— Kurt Gödel 1931, B. Rosser 1936

ZFC is not sufficiently strong to prove or refute all statements in integer arithmetic (+,•) – *unless(!)* the axioms of ZFC are contradictory. 

ZFC axioms contradictory? **No!** – we think!!

Second Incompleteness Theorem

— Kurt Gödel, 1931

ZFC is not sufficiently strong to prove that the axioms of ZFC are not contradictory – *unless(!)* the axioms of ZFC are contradictory.

Large programs are expensive and buggy. Many reasons. 

Need a major overhaul in the programming environments.

Logic will play a central role.

SOME SOFTWARE DISASTERS

THERAC-25

A radiation therapy machine, involved in six accidents 1985-87 . Patients given 100 times intended dose. Factors include bad software design and development.

ARIANE 5

A launcher for rockets. Blew up June, 1996, from a failure of the Inertial Reference Systems, Traced to a piece of software.

INTEL PENTIUM BUG

Intel's Pentium computer chip was recalled in 1994 at a cost of 500M USD. The floating point arithmetic software was flawed.

Formal specifications

Need formal specifications.

Written in applied Predicate Calculus.

Needed to support

- Reusability of software
- Verification of software

Software verification

- A mathematical proof that the software obeys the formal specification.
- Far beyond the usual software testing in common practice in industry.
- We want a programming environment where a team of software developers writes code *AND* verifies the code.

Software verification

- Programming languages need an overhaul
- Heavy duty tools need development
- Tools include algorithms from logic, refined by computer scientists, as discussed previously.

Work highly exploratory

Not clear how to design a working system for cost effectively creating verified code for real world programming situations

Potential applications of logic in education.

Develop systems that automatically generate multistep homework and exam problems satisfying Instructor controlled parameters.

Automatically graded. Student enters multi step solution in special format, obtains feedback after each step.

Full history of student responses on all problems stored and used to adjust future problems.

Data used to fine tune Instructor's Lectures.

Ideas have been around

Challenge

Make automatically generated multi step problems truly reflect course goals, and the interaction engage the students appropriately.

Need sophisticated problem templates, reflecting many degrees of difficulty.

Need formal structure to support real time computer grading.

Need formally structured modes through which students enter multistep solutions, and receive feedback in real time.

Logic needed everywhere!

Discrete Math for Computer Science Majors

- Generally speaking, computer science majors are required to take *a* course in basic discrete mathematics and basic proof structure.

- Not enough interaction time under normal teaching methods for students to absorb these difficult and subtle skills.

SYRUS Project

—run by me and Bruce Weide of CSE.

Template 1:

($\forall$ integers n, m, r)($x \; \alpha \; y \wedge z \; \beta \; w \rightarrow u \; \gamma \; v$).

True or false? If true, prove. If false, give counterexample.

Template 2:

($\forall$ integers n, m)($\exists$ integer r)($x \; \alpha \; y \wedge z \; \beta \; w$).

True or false? If true, prove. If false, give counterexample.

Here x, y, z, w, u, v are among letters n, m, r, and α, β, γ are among symbols $<$, $>$, $\leq$, $\geq$, $=$, $\neq$.

True/False is checked in real time. Proofs are entered interactively, with many steps, and real time feedback.

Use of *Logic* for teaching *Logic*? No surprise.

Ideas strongly applicable to elementary mathematics courses, including K–12.

Future targets

ϵ-δ proofs in calculus.

Courses in the mathematical sciences.

Courses in engineering.

And beyond...

THEOLOGY: POSITIVE PROPERTIES

There have been attempts to prove the existence of God, using the concept of positive properties. Most notably by Leibniz (co invemtor with Newton of the Calculus) and Gödel (the great logician we have encountered before).

These attempts go under the name of "the ontological argument" and are very controversial.

We use the concept of positive properties for quite a different purpose.

THEOLOGY: POSITIVE PROPERTIES

Instead, we use the concept of positive properties to prove that the usual ZFC axioms for mathematics (discussed earlier) are free of contradiction. I.e., that ZFC is consistent.

In other words, a concept from classical Theology is used to prove that mathematics is free from contradiction.

POSITIVE PROPERTIES

In this theological framework, we consider properties or attributes of things. Here are three interesting examples:

"Possessing an intellect"
"Knowing at least as much as any other thing"
"Knowing more than any other thing"

Certainly these properties are positive in the sense that it is "better" to have them (if possible) than not to have them.

But there is the question of whether these properties are possible to have.

The first one obviously is. The seond and third are problematic.

POSITIVE PROPERTIES

SOME GENERALPRINCIPLES

For any given poperty, it is either positive to possess it or it is positive to not possess it.

If it is positive to possess P and positive to possess Q, then it is positive to possess (P and Q).

A perfect being possesses all of the positive properties.

There is a perfect being.

There is exactly one perfect being.

POSITIVE PROPERTIES

Returning to the three examples:

"Possessing an intellect"
"Knowing at least as much as any other thing"
"Knowing more than any other thing"

To apply the general principles, we need to know that it is possible for a being to have these properties. The first one is not problematic.

We can argue for the second: For any given x, it is positive to know at least as much as x. Therefore a perfect being knows as much as any other thing. So the second example is possible.

More argument is needed for the third example.

THIS IS JUST AN INDICATION OF HOW ONE CAN AT LEAST ATTEMPT TO INTELLIBLY NAVIGATE SUCH MURKY WATERS.

POSITIVE PROPERTIES

THE SUPERNATURAL
THE NATURAL

The perfect being is clearly part of the supernatural world, and not part of the natural world.

In the natural world, beings may have a lot of positive properties, but not all of them.

We expect that for every natural being, there is another natural being which possesses all of the positive properties of the first, and more.

POSITIVE PROPERTIES

THE NATURAL (NOT THE SUPERNATURAL)

We start with the universe of natural objects.
We use the standard concept of "properties of objects" in mathematics.
We use the theologically based oncept of "positive properties of objects".
We assume the general principles we have discussed, which correspond to the standard mathematical idea of a "nonprincipal ultrafilter".

We do not have a natural object that possesses all positive properties. That is only true of a (the) perfect being, which is only exists in the supernatural.

An ANGEL is an object which possesses all DEFINABLE positive properties.

POSITIVE PROPERTIES

ANGELS

An ANGEL is an object which possesses all DEFINABLE positive properties.

POWERFUL AXIOM: THERE EXISTS AN ANGEL!

We have been able to prove, using an Angel, that mathematics is free of contradiction (I.e., ZFC is consistent).

I hope you've found these Interdisciplinary Adventures of some interest.

Thank you very much.