

Notes 2023-08-23

Evelyn Mitchell

August 23, 2023

1 Mathematical Truth

<https://plato.stanford.edu/entries/intuitionistic-logic-development/#ProofInterpretation>

Truth in intuitionistic logic is **by construction**. That is, you must be able to construct an instance of the asserted statement in order to assume it.

BHK-Interpretation (for "Brouwer, Heyting, Kolmogorov") or Proof Interpretation as given by Troelstra and van Dalen in *Constructivism in Mathematics* (Troelstra and van Dalen 1988: 9):

(H1) A proof of $A \wedge B$ is given by presenting a proof of A and a proof of B .

(H2) A proof of $A \vee B$ is given by presenting either a proof of A or a proof of B (plus the stipulation that we want to regard the proof presented as evidence for $A \vee B$).

(H3) A proof of $A \implies B$ is a construction which permits us to transform any proof of A into a proof of B .

(H4) Absurdity \perp (contradiction) has no proof; a proof of $\neg A$ is a construction which transforms any hypothetical proof of A into a proof of a contradiction.

(H5) A proof of $\forall x A(x)$ is a construction which transforms a proof of $d \in D$ (D the intended range of the variable x) into a proof of $A(d)$.

(H6) A proof of $\exists x A(x)$ is given by providing $d \in D$, and a proof of $A(d)$.

2 typesetting

For proofs: https://www.overleaf.com/learn/latex/Theorems_and_proofs

\wedge is logical and.

\vee is logical or.

\perp is contradiction.

\implies is logical implication. If the left side is true, the right side is true.

\in is an element of
 \forall for all
 \exists exists

3 Glossary

condition Example:

$$q(s, a) = \mathbb{E}[G_t | S_t = s, A_t = a]$$

The Expectation is conditioned on $S_t = s$ and $A_t = a$.

(The example is from Reinforcement Learning: <https://youtu.be/TCCjZe0y4Qc> ; Sutton and Barto 2018)

4 Bourbaki Theory of Sets

I'm taking a suggestion to follow the formal path into proof writing. Thus, Nicolas Bourbaki, Elements of Mathematics, Volume 1: Theory of Sets

Each chapter proceeds in a logical order, with later parts only requiring knowledge of what has come before, so no backtracking, or assuming knowledge of a later topic. The chapters start with **definitions**, the **axioms** and the **theorems** of the chapter.

Additional results are variously named as "propositions", "lemmas", "corollaries", "remarks" and so on. Commentaries on especially important theorems are called "scholium".

Dangerous bend passages are marked with a large Z in the margin. (So that is where Knuth got it!). These mark warnings to the reader of serious errors.

Exercises are designed to cement learning, and to prompt understanding in the reader of results which are interesting, but didn't fit in the text.

The authors are creating a self-contained text, as an explicit goal. There are Historical Notes with pointers to the rest of the literature, including unsolved problems.

This volume includes: 1. Description of formal mathematics.

2. Theory of sets.

3. Ordered sets; cardinals; natural numbers.

4. Structures.

4.1 Introduction

Mathematics, since the ancient Greeks, has had a standard of precise and rigorous explanation, known as *proof*.

Bourbaki is intending to discern the structure underlying both vocabulary and syntax. That is, they are concerned with precise definition of the words used in mathematical language, and clear description of the form of mathematical statements. Their goal is "formalized" mathematics.

When they were working on this project, in 1930s France, it was a novel and needed project. Now, in 2023, with the advances in mathematical theory, and computer science,

the idea of formalizing knowledge precisely is still rare in the sense of popularity, but not rare in the sense of daily importance. Our entire technology stack is founded on devices with precise, constructed formal definitions, and entire tool chains based on translation between languages of higher and lesser levels of abstraction. From loose product goals such as "Given a login screen, when a new user is signing up, then create an entry in the user database for them" to computer programs assembled out of library components to intermediate representations, to machine language, down to bits flowing through silicon, each step uses the idea of formalization to create actions in the world.

That doesn't mean that our technology is without error, for we do not yet have fully proof-checked programs at every layer. Nor are tools for proof repair, such as described in Ringer 2021 (<https://dependenttyp.es/pdf/thesis.pdf>) yet in common use.

Since Bourbaki, and especially since Voevodsky's work on Dependent Types, mathematics has become more formalized. There are projects such as MetaMath, and Lean, to express theorems in languages such as Coq, Isabelle/HOT and Lean, which allow for computers to assist in checking proofs.

The formulae of ordinary algebraic calculation would be another example, if the rules governing the use of brackets were to be completely codified and strictly adhered to; in practice, some of these rules are never made explicit, and certain derogations of them are allowed. p7 English edition 2004

Formalizing helps us protect against lapses of reasoning.

— Up to paragraph starting "The *axiomaticmethod*" on page 8.