Introduction to Formal Proof

Bernard Sufrin

Trinity Term 2018

0: Introduction and Overview

The design of the following treatise is to investigate the fundamental laws of those operations of the mind by which reasoning is performed; to give expression to them in the symbolical language of a Calculus, and upon this foundation to establish the science of Logic ... and, finally, to collect ... some probable intimations concerning the nature and constitution of the human mind.

George Boole: *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities* (Macmillan, 1854)[1]

[1] http://www.gutenberg.org/ebooks/15114

Course Overview

▷ A proof is a rigorous argument supporting a mathematical/computational/logical conjecture

▷ We are only prepared to accept proofs that obey certain rules of reasoning

▷ In this course we are going to show how to:

- codify some widely-accepted rules of reasoning about certain kinds of conjecture
- *systematically* test whether mathematical/computational/logical proofs are sound (*i.e.* obey the rules of reasoning)
- systematize the discovery of proofs that obey the rules

- Most working mathematicians (and computer scientists interested in proof)
  - choose (or accept) some rules of reasoning[2], and then repeatedly:
    - make some definitions and some assumptions/hypotheses/premisses (build a model)
    - and then repeatedly
      - use intuition to posit a consequence (make a conjecture about the model)
      - use the rules of reasoning to try to prove the conjecture
      - revise the conjecture if the proof stalls ... otherwise call it a theorem
    - revise the model when it gets frustrating, inelegant or unrealistic
  - revise the rules (or choose different ones) when forced to by exigency or curiosity

[2] The rules are sometimes left uncodified, implicit, or intuitive

▷ In this course we will give a brief account of the underpinning of formal proof, explaining:

- What a deductive system is
- What a formal proof in a deductive system "really" is
- The basis for our confidence in what we can prove formally in a deductive system
- What we do when we add definitions to a deductive system

▷ Eventually we will revisit the kind of proof you have already done in discrete mathematics, functional programming, *etc*.

▷ *en route* we will cover

- Propositional Logic
- Natural Deduction
- Notions of Soundness and Completeness
- First-order (Predicate) Logic
- Equational Theories

▷ We will puncutate the theory we present with many little case-studies

▷ The formal proofs we study will mostly be of *conceptually simple* results.

- because the foundations of fully formal reasoning we introduce will be unfamiliar
- and it will therefore be easier to see how they work when applied to familiar material

Case Study: three presentations of an equational proof

▷ Definitions:

$$\begin{array}{lcl} swap(x, y) & = & (y, x) \\ (f \cdot g)x & = & f(g\ x) \\ id\ x & = & x \end{array}$$

▷ Conjecture:

$$swap \cdot swap = id$$

Stylized presentation with compressed transitive rules

1:	(swap•swap)(x,y)	
2:	= swap(swap(x,y))	Unfold •
3:	= swap(y,x)	Unfold swap
4:	= (x,y)	Unfold swap
5:	= id(x,y)	Fold id
6:	swap•swap=id	*ext2* 1-5

▷ Expressions on lines 2-5 annnotated by *stylized* evidence linking them to their predecessor

- $Unfold\ f$: (sub-) expression rewritten using the definition of f read left-to-right
- $Fold\ f$: (sub-) expression rewritten using the definition of f read right-to-left
- Nested instances of the transitivity rule presented concisely

▷ Line 6 annotated with evidence linking it to the subproof in 1-5

$ext_2\ 1-5$:

- appeals to the rule of extensionality
- $1-5$: delineates the proof of the antecedent requirement for that rule

Inference tree presentation

$$\dfrac{\dfrac{}{(\mathsf{swap}\cdot\mathsf{swap})(x,y)=\mathsf{swap}(\mathsf{swap}(x,y))}\,\text{Unfold }\cdot \qquad \dfrac{\dfrac{}{\mathsf{swap}(\mathsf{swap}(x,y))=\mathsf{swap}(y,x)}\,\text{Unfold swap} \qquad \dfrac{\dfrac{\dfrac{}{\mathsf{swap}(y,x)=(x,y)}\,\text{Unfold swap} \qquad \dfrac{\dfrac{}{\mathsf{id}(x,y)=(x,y)}\,\text{id}}{(x,y)=\mathsf{id}(x,y)}\,=\text{symmetric}}{\mathsf{swap}(y,x)=\mathsf{id}(x,y)}\,=\text{transitive}}{\mathsf{swap}(\mathsf{swap}(x,y))=\mathsf{id}(x,y)}\,=\text{transitive}}{(\mathsf{swap}\cdot\mathsf{swap})(x,y)=\mathsf{id}(x,y)}\,=\text{transitive}$$

$$\dfrac{(\mathsf{swap}\cdot\mathsf{swap})(x,y)=\mathsf{id}(x,y)}{\mathsf{swap}\cdot\mathsf{swap}=\mathsf{id}}\,ext2$$

▷ Read the proof tree bottom-up

- The root invokes extensionality
- The branches invoke transitivity of =
- Each leaf equation is a substitution instance of one of the definitions
- Each interior equation is justified (using a named rule) by its ancestor equation(s)

Linearized presentation

▷ The proof tree can also be presented as a sequence of numbered equations

1:	(swap•swap)(x,y)=swap(swap(x,y))	Unfold •
2:	swap(swap(x,y))=swap(y,x)	Unfold swap
3:	swap(y,x)=(x,y)	Unfold swap
4:	id(x,y)=(x,y)	id
5:	(x,y)=id(x,y)	= symmetric 4
6:	swap(y,x)=id(x,y)	= transitive 3,5
7:	swap(swap(x,y))=id(x,y)	= transitive 2,6
8:	(swap•swap)(x,y)=id(x,y)	= transitive 1,7
9:	swap•swap=id	ext2 8

▷ Here the links in the tree structure are provided by the line numbers that follow the indications that the transitivity and extensionality rules have been used to justify a particuar line.

- ▷ The essential mathematical content of the proof is clearer in the stylized form
- ▷ The complete structure of the proof is made explicit in the tree and full linear forms
- ▷ **But they are all presentations of the same proof!**
- ▷ We will shortly see why this is so

Contents

Note 1: 1 ☞

Boole's goal was to do for Aristotelean logic what Descartes had done for Euclidean geometry: free it from the limits of human intuition by giving it a precise algebraic notation. To give a simple example, when Aristotle wrote: "All men are mortal."

Boole replaced the words "men" and "mortal" with variables, and the logical words "all" and "are" with arithmetical operators:

$$x = x * y$$

Which could be interpreted as "Everything in the set x is also in the set y."

The Laws of Thought created a new scholarly field—mathematical logic—which in the following years became one of the most active areas of research for mathematicians and philosophers. Bertrand Russell called the Laws of Thought "the work in which pure mathematics was discovered."

(Extracted from: https://www.theatlantic.com/technology/archive/2017/03/aristotle-computer/518697/)

Note 2: 2 ☞

Some writers use the epithet *valid* instead of *sound*. Whatever word we use we are engaged in the systematic effort to make as precise as possible the conditions under which an argument is acceptable.

Note 3: 3 ☞

We will use the words "assumption", "hypothesis", and "premiss"[3] almost interchangeably in these notes; though we have a mild preference for using "premiss" only for assumptions made in the outermost layer of an argument, since the word is derived from the Latin phrase *præmissa (propositio)* (proposition set in front).

As we shall see towards the end of the course, a *definition* can be treated as a particular kind of inference rule.

Note 4: 3 ☞

Frustrating means that we cannot prove things that are obviously true about the situation we are modelling. *Inelegant* means that the proofs we can do are complex, hard to discover, and hard to understand. *Unrealistic* means that things we can prove about the model turn out not to be true in the world we are modelling.

Note 5: Failure to find a formal proof 3 ☞

Failure to find a formal proof of a conjecture does not mean that the conjecture is wrong. It may mean that one is not clever enough, or that the assumptions are too weak, or that the rules of proof we are using are insufficient.

[3] The US-English spelling of premiss is "premise".

Note 6: Some familiar rules for reasoning about equalities 7 ☞

Extensionality Rule

from $F\ \overline{x} = G\ \overline{x}$ **(for a fresh variable $\overline{x}$ of the right type)**
infer $F = G$

Transitivity Rule

from $T_1 = T_2$
and $T_2 = T_3$
infer $T_1 = T_3$

Symmetry Rule

from $T_1 = T_2$
infer $T_2 = T_1$

Operand Rule

from $T_1 = T_2$
infer $F\ T_1 = F\ T_2$

Operator Rule

from $F_1 = F_2$
infer $F_1\ T = F_2\ T$

All but the extensionality rule can be derived from the rule of substitutivity of equal terms, which is (roughly)

from $T_1 = T_2$
and $\phi(T_1)$
infer $\phi(T_2)$