Proof Assistants

MATH230

Te Kura Pāngarau | School of Mathematics and Statistics
Te Whare Wānanga o Waitaha | University of Canterbury

Outline

1. Examples
2. Proof Assistants

Example

$$PA \vdash 0 + 1 = 1$$

Induction Example

$$PA \vdash \forall x\ (0 + x = x)$$

Example

$$PA \vdash 7 \times 1 = 7$$

Is This Feasible?

We appear to have a precise formal language in which we can prove number theoretic statements - great!

Unfortunately, it takes slides and slides to prove basic statements about arithmetic - let alone something non-trivial like the fundamental theorem of arithmetic.

The idea of developing new mathematics using natural deductions is unthinkable - no one would advocate for that.

Proof Assistants

Happily we now have computers! These computers can store sub-proofs to make writing new proofs much easier. While still maintaining the easy verification of our natural deductions.

Software can be written to tuck a lot of the steps away and verify them for us.

Agda, Coq, Isabelle, HOL, and L$\exists\forall$N are a few examples of programming languages that have been developed. They have all, in some way, absorbed the ideas we've developed throughout the course.

These languages give mathematicians tools that they can use to genuinely prove things at the frontier.

Proof Assistants

These come in two primary flavours (i) automatic theorem provers (ii) interactive theorem provers.

If a first-order theory is decidable, then there is (in-theory) an algorithm that can be written to decide whether a particular wff is a theorem. For example, Presburger's paper shows that questions about the addition on natural numbers are decidable.

However these are often impractically slow.

L∃∀N

L∃∀N is an interactive theorem prover which has the user enter the proofs. The structure of the language allows the program to verify the proof. Some automatic tactics are available for certain problems.

The **Xena Project** is building a database of undergraduate math with proofs verified by the proof checker L∃∀N.

Mathematicians at the frontier are verifying their work in L∃∀N!

Tutorial 6 will have you work in L∃∀N. This way you get an idea for how these ideas have had an effect today, on modern mathematics. Plus, proving basic properties like commutativity of addition is much nicer in L∃∀N, than writing it on paper.

So how does this work?