MAU11602

Lecture 18

2026-03-11

Peano arithmetic, language

There are a number of formal theories of arithmetic, generically called Peano arithmetic, after Giuseppe Peano, even though Peano didn't fully formalise his system and no one now uses it.

Before we do anything we need a language. The language elements are:

- 0. This is the only literal number used.
- The successor function S. With this and 0 we can represent any natural number, e.g. $SSSSSSSSSSSSSSSSS0$ is what we normally write as 17.

As you may have guessed Peano arithemetic is designed make it easy to state and prove theorems *about* it, not easy to state and prove theorem *in* it!

- The arithmetic operators $+$ and $\cdot$.

There is no $-$ or $/$ because those are not total functions and allowing partial functions would complicate the logic.

- The Boolean operators $\neg$, $\wedge$, $\vee$, and $\rightarrow$.

The Boolean constants $\top$ and $\bot$ are usually not used, so we need $\neg$ for descriptive completeness.

Language, continued

Equality, $=$, and the other arithmetic relations $<$, $\leq$, $\geq$, and $>$.
- Variables. In principle there should be infinitely many of them, but I'll just write them with single letters.

The domain is meant to be natural numbers, so all variables should be thought of as representing natural numbers.
- The quantifiers $\forall$ and $\exists$.
- Parentheses, for overriding the precedence and associativity rules

Importantly, some things are left out of the language:
- The partial functions $-$ and $/$ and the Boolean constants $\top$ and $\bot$.
- Conditionals or let expressions.
- Any functions other than the ones provided.
- Data structures, even simple, non-recursive ones like ordered pairs.
- Sets

I won't fully describe the grammar of this language, with all of its rules of precedence and associativity. Hopefully you can mentally parse statements anyway.

Types, interpretation, descriptive completeness

The type theory of this language is very simple. Expressions can only have two types: natural or boolean.

It's so simple that most accounts of Peano arithmetic don't even mention its existence.

If you believe the natural numbers exist then every expression of Boolean type with no free variables is either true or false, so there is essentially only one interpretation.

It therefore makes sense to ask about syntactic and semantic completeness, once we have some axioms and rules of inference.

What about descriptive completeness? Without sets and functions this doesn't look good.

Things are not as bad as they seem, if you're sufficiently devious.

A test case

There are infinitely many primes. We can't formally prove this yet because we don't have any axioms or rules of inference, but can we even state it?

This looks like a statement about cardinality of a set and our language doesn't have sets, but it's equivalent to saying there is no upper bound on the sizes of primes.

We should look for a statement of the form

$$\neg\exists a.\forall b.p(b) \rightarrow b \leq a,$$

where $p(b)$ expresses the fact that b is prime.

A natural number is prime if and only if its only divisors are 1 and itself, so $p(b)$ should have the form

$$\forall c.q(b,c) \rightarrow c = S0 \vee c = b$$

where $q(b,c)$ expresses the fact that b is divisible by c.

We don't have division, but we can still make statements about is using multiplication.

We can express $q(b,c)$ by

$$\exists d.b = c \cdot d$$

.

A test case, continued

Putting all the pieces together, the statement that there are infinitely many primes is equivalent to

$$\neg \exists a. \forall b. (\forall c. (\exists d. b = c \cdot d) \to c = S0 \lor c = b) \to b \leq a.$$

We had to use some properties of the natural numbers to create this "equivalent" statement. Some are really just definitions, e.g. of primes and divisibilty.
Some should arguably be theorems though, like the fact that finiteness is the same as the existence of an upper bound.

Another test case

Okay, we can express the fact that a number is prime, but what about prime powers? We don't have exponentiation in our language.
Suppose a is prime. Then b is a power of a if and only if every divisor of b either is 1 or is divisible by a.
These are all things we already know how to express, so we just need to put them together:

$$\forall c. ((\exists d. b = c \cdot d) \to c = S0 \vee (\exists d. c = a \cdot d)) .$$

This is a bit unsatisifying for several reasons:
- It doesn't give us a way to identify which power of a b is.
- It only works for prime a, and the first point means we can't extend it easily to composite a. For example, powers of 6 are powers of 2 and of 3, but they have to be the same power, and it's not obvious how to express this.
- We're starting to need lots of theorems about arithmetic to state theorems about arithmetic.

Yet another test case

It is an open question whether there are infinitely many prime Fibonacci numbers.
Forget proving this, case we even state it?
Our first test case gives all the necessary ingredients except Fibonacci numbers, so we're looking for a statement of the form

$$\neg\exists a.\forall b.\,(\forall c.\,(\exists d.b = c \cdot d) \to c = S0 \vee c = b) \wedge r(b) \to b \leq a.$$

where $r(b)$ somehow expresses the fact that b is a Fibonacci number.
The usual (recursive) definition of Fibonacci numbers can't be used here because we have no way to define a sequence, even without recursion.
We could use the fact that b is a Fibonacci number if and only if

$$\exists e.\exists f.\,(c \cdot e = S(e \cdot (e + f)) \wedge (b = e \vee b = f))\,.$$

so we can therefore substitute this for $r(b)$ above to get the statement we want.
All of these substitution steps are capture avoiding substitution, but I chose variables to make sure I wouldn't have to perform any α conversions.
Proving that this expression works requires a fair amount of number theory.

Arithmetic sets

We say that a set of natural numbers is *arithmetic* if there is a Boolean expression in Peano arithmetic with with one free variable such that a natural number is in the set if and only if substituting that natural number for all free occurences of the variable gives a true value.

The word arithmetic has two different pronunciations in that sentence.

So far we've seen that the set of divisors of a given number is arithmetic, that the set of primes is arithmetic, that the set of powers of a given prime is arithmetic, and that the set of Fibonacci numbers is arithmetic.

We haven't seen evidence that any set is not arithmetic. The fact that we can't think of an appropriate expression doesn't mean there isn't one!

There are non-arithmetic sets though. We can see this with a counting argument, if we assume some facts about cardinality of sets from later in the module.

The set of arithmetic sets is countable and the set of all subsets of the natural numbers is uncountable, so they can't be the same set.

This argument doesn't identify any particular non-arithmetic set, but later I will give you actual examples.

Arithmetic functions

Similarly we can define arithmetic functions. A unary arithmetic function is a function from the naturals to the naturals such that is a Boolean expression in Peano arithmetic with with two free variables such that an ordered pair of natural numbers is in the graph of the function if and only if substituting the left element of the pair for all free occurences of the first variable and the right element for all free occurences of the second the gives a true value.
For example, the square function is arithmetic. The corresponding expression is just

$$x \cdot x = y$$

We can extend this definition from unary functions to *n*-ary functions in the obvious way.

Properties of arithmetic sets and functions

Arithmetic sets give us a way of measuring just how close Peano arithmetic comes to descriptive completeness.

The various ways of combining expressions give us properties of arithmetic sets and functions.

For example, if P and Q are boolean expressions with x with x as its only free variable then $P \wedge Q$ is a boolean expression with x as its only free variable, so if A and B are arithmetic sets then $A \cap B$ is an arithmetic set.

If P is a boolean expression whose free variables are x and y then $\exists x.P$ is a boolean expresion whose only free variable is y, so the range of a unary arithmetic function is an arithmetic set.

If we knew that the Fibonacci sequence, as a function, was arithmetic this would tell us that the set of Fibonacci numbers is arithmetic.

We could really us a more systematic way of proving sets, and especially functions, are arithmetic.