Formal Proofs in FOL

Robert Y. Lewis

CS 0220 2024

February 7, 2024

Overview

First order logic: a refresher

From Monday: *first order logic* extends the language of *propositional logic*.

Our atoms become predicates: propositions “about things.”

We can quantify over *domains* of things.

Every prime number greater than 2 is odd.
$\forall n : \mathbb{N}, \mathit{Prime}(n) \wedge (n > 2) \to \mathit{Odd}(n)$
We say $\mathbb{N}$ is the *domain of quantification* for the universal quantifier here.

Why should computer scientists care?

Well, how do you specify the behavior of the programs you write?

- $\forall L : \texttt{List}\mathbb{N}, \texttt{reverse}(\texttt{reverse}(L)) = L$
- $\forall L : \texttt{List}\mathbb{N}, \texttt{isSorted}(\texttt{sort}(\texttt{L}))$
- $\forall L : \texttt{List}\mathbb{N}, \neg(L = []) \to \exists x : \mathbb{N}, \texttt{head}(L) = x$

Test cases can only describe behavior in concrete instances. *Specifications*, written as first order formulas, are much richer!

forall proof rules

Introduction: To **prove** a forall goal $\forall x : T, G(x)$:
Suppose you have a (new, freshly named) $x : T$ in your context, and prove $G(x)$ for that new x.

I want to show that every number is either prime or the product of two other numbers. Suppose n is a number. Show that n is prime or n is the product of two other numbers.

Elimination: To **use** a forall hypothesis $\forall x : T, H(x)$:
If $t : T$ is any term of the right type, then you can add a hypothesis $H(t)$.

I know that every number is either prime or the product of two other numbers. Therefore, I know that either 2 prime or 2 is the product of two other numbers. I know that either 5 is prime or 5 is the product of two other numbers...

Exists proof rules

To **prove** an existential goal $\exists x : T, G(x)$:
Provide a witness.

I want to show that there is a perfect square whose final digit is 4. I claim that my witness is 64. Then, I must show that 64 is a perfect square and the final digit of 64 is 4.

Elimination: To **use** an existential hypothesis $\exists x : T, H(x)$:
you can create a (new, freshly named) $t : T$, and add a hypothesis $H(t)$. "Give a name to the witness."

I know that there is a perfect square whose final digit is 4. Let's call this perfect square *ps*. I know that *ps* is a perfect square and the final digit of *ps* is 4.

Quantifiers in Lean

We'll see some new tactics in Lean:

- forall introduction: `fix x`
- forall elimination: `have hx := hall x`
- exists introduction: `existsi w`
- exists elimination: `eliminate hex with x hx`

Remember the CS22 Lean reference manual: `https://docs.cs22.io/`