Introduction to Lean

Jesse Vogel

Leiden University

19 September, 2022

What is Lean?

- Interactive Theorem Prover (ITP)
- Mathlib
- Alternatives: Coq, Agda, Isabelle, HOL, Mizar, Metamath, ...

Type theory

Type theory	Set theory
types	sets
terms	elements
$x : T$	$x \in T$

Axiom (Product types)

Given types A and B, we can construct the *product type*

$$\prod_{a:A} B \qquad (\textbf{Lean} \quad \texttt{Π (a : A), B})$$

also written as

$$A \to B \qquad (\textbf{Lean} \quad \texttt{A → B})$$

Axiom (Function application)

Given a term $f : A \to B$ and a term $a : A$, there exists a term

$$f(a) : B \qquad (\textbf{Lean} \quad \texttt{f a})$$

Note: $f : A \to B \to C$ means $f : A \to (B \to C)$,
so $f(a)(b)$ is denoted '`f a b`'

Axiom (Function abstraction)

Given types A and B, we can construct terms of $A \to B$ using λ-*abstraction*.

Example `λ (x : Nat), x * x + 3` is a term of `Nat → Nat`

Axiom (Dependent product types)

Given a type A and term $t : A \to \texttt{Type}$,
we can construct the *dependent product*

$$\prod_{a:A} t(a) \qquad (\textbf{Lean} \quad \texttt{Π (a : A), t a})$$

Axiom (Conversion rules)

- renaming variables

  Example $`\lambda$ (x : Nat), x + 1$'$ $\rightsquigarrow$ $`\lambda$ (y : Nat), y + 1$'$

- unfold lambdas

  Example $`(\lambda$ (x : Nat), x + 1) 7$'$ $\rightsquigarrow$ $`$7 + 1$'$

- notion of extensionality

  Example $`\lambda$ (x : Nat), f x$'$ $\rightsquigarrow$ $`$f$'$

- etc.

Logic in type theory

Curry–Howard correspondence:

logical propositions are types,

whose terms are proofs of that proposition

a proposition is true if and only if it is non-empty

Logic	Type theory
proposition P	type P
proof of P	term $p : P$
$P \wedge Q$	$P \times Q$
$P \vee Q$	$P + Q$
$P \Rightarrow Q$	$P \to Q$
$\forall$	Π-type
$\exists$	Σ-type
true	unit type
false	empty type
not P	$P \to \texttt{false}$

Definition. A *monoid* consists of the following data:

- a type M
- a map $m : M \to M \to M$
- a term $e : M$
- a *proof* (i.e. term) of $\forall\ (a\, b\, c : M),\ m(m(a,b),c) = m(a, m(b,c))$
- a *proof* (i.e. term) of $\forall\ (a : M),\ m(a,e) = a \wedge m(e,a) = a$

Axiom (Propositional extensionality)

Any two proofs of the same proposition are equal

$$\forall\ (P : \texttt{Prop})\ (p\, q : P),\ p = q$$

Inductive types

```
inductive Weekday
| monday    : Weekday
| tuesday   : Weekday
| wednesday : Weekday
| thursday  : Weekday
| friday    : Weekday
| saturday  : Weekday
| sunday    : Weekday
```

Inductive types

```
inductive Int
| of_nat          : Nat → Int    -- of_nat n
| neg_succ_of_nat : Nat → Int    -- neg_succ_of_nat n

inductive Sum (A B : Type)
| inl : A → Sum                  -- inl a
| inr : B → Sum                  -- inr b

inductive Product (A B : Type)
| mk : A → B → Product           -- mk a b

inductive Sigma (A : Type) (t : A → Type)
| mk : Π (a : A), (t a → Sigma)
```

Inductive types

```
inductive Nat
| zero : Nat          -- zero
| succ : Nat → Nat    -- succ n
```

Axiom (Recursion)

```
Nat.rec :
  Π (t : Nat → Sort*), t(0) → ( Π (n : Nat), t(n) → t(succ(n) ) →
  Π (n : Nat), t(n)
```

Inductive types

```
inductive Nat
| zero : Nat
| succ : Nat → Nat
```

Axiom (Recursion) (for `t` equal to `P : Nat → Prop`)

```
P(0) → (∀ (n : Nat), P(n) → P(succ(n)) → ∀ (n : Nat), P(n)
```

Principle of induction

Let $P(n)$ be a proposition for every natural number n. If $P(0)$ holds, and for all $n \in \mathbb{N}$ we have $P(n) \Rightarrow P(\text{succ}(n))$, then $P(n)$ holds for all $n \in \mathbb{N}$.

Lean demo