Propositional Logic in Lean

Heshan Du
University of Nottingham Ningbo China

October 2024

Aims and Learning Objectives

- To be able to apply the tactics *constructor*, *cases*, *left*, *right* in Lean.
- To be able to construct proofs for tautologies in propositional logic using Lean.

Reading

- Thorsten Altenkirch, *Introduction to Formal Reasoning*, 2023.
  - Chapter 2. Propositional Logic
- Kenneth H. Rosen, *Discrete Mathematics and its Applications*, 6th edition, 2007.
  - Chapter 1. The Foundations: Logic and Proofs

Conjunction: Example 1

- To prove a conjunction $P \wedge Q$, we need to prove both P and Q.
- This is achieved via the *constructor* tactic. *constructor* turns the goal $P \wedge Q$ into two goals P and Q.
- How to prove $P \rightarrow Q \rightarrow P \wedge Q$ in Lean?

```
variables P Q : Prop

example : P → Q → P ∧ Q :=
begin

end
```

Conjunction: Exercise 2

- Now we show that $\wedge$ is *commutative*.
- Prove $P \wedge Q \to Q \wedge P$ in Lean.

```
variables P Q : Prop

theorem comAnd : P ∧ Q → Q ∧ P :=
begin

end
```

Use a Conjunction in an assumption

- Assume $P \wedge Q$ is the same as assuming both P and Q.
- This is facilitated via the *cases* tactic which needs to know which assumption we are going to use and how we want to name the assumptions which replaces it.
- E.g., *cases pq with p q*.
- The name *cases* seems to be a bit misleading, since there is only one case to consider here.

Conjunction: Exercise 3

Prove $P \wedge Q \leftrightarrow Q \wedge P$ in Lean.

```
variables P Q : Prop

theorem comAndIff : P ∧ Q ↔ Q ∧ P :=
begin

end
```

Use a Theorem to Prove a Theorem

- Lean always abstracts the propositional variables we have declared.
- We can actually use *comAnd* with different instantiation to prove $P \wedge Q \leftrightarrow Q \wedge P$.

Answer to Exercise 3

```
theorem comAndIff : P ∧ Q ↔ Q ∧ P :=
begin
  constructor,
  apply comAnd,
  apply comAnd,
end
```

In the second use of *comAnd*, Q is instantiated with P, and P is instantiated with Q.

The Currying Equivalence

- A statement like $P \to Q \to R$ means that R can be proved from assuming both P and Q.
- Indeed, it is equivalent to $P \wedge Q \to R$.
- We can show this formally by using $\leftrightarrow$. Remember that $P \leftrightarrow Q$ is the same as $(P \to Q) \wedge (Q \to P)$.
- How to prove $P \wedge Q \to R \leftrightarrow P \to Q \to R$ in Lean?

Proving The Currying Equivalence in Lean

```
variables P Q R: Prop

theorem curry : P ∧ Q → R ↔ P → Q → R :=
begin

end
```

Disjunction

- To prove a disjunction $P \vee Q$, we can either prove P or we can prove Q.
- This is achieved via the appropriately named tactics *left* and *right*.
- How to prove $P \to P \vee Q$ in Lean?

```
variables P Q R: Prop

example : P → P ∨ Q :=
begin

end
```

- How to prove $Q \to P \vee Q$ in Lean?

Disjunction: Exercise

Prove $(P \to R) \to (Q \to R) \to P \lor Q \to R$ in Lean.

```
variables P Q R: Prop

theorem case_lem: (P → R) → (Q → R) → P ∨ Q → R :=
begin

end
```

Use a Disjunction

- To use a disjunction $P \vee Q$, we have to show that the current goal follows both from assuming P and from assuming Q.
- To achieve this, we use *cases*. This time the name actually makes sense.

Tactics: Cases

- We use *cases pq with p q*, which means that we are going to use $P \vee Q$.
- There are two cases, resulting in two subgoals.

```
2 goals
case or.inl
P Q R : Prop,
pr : P → R,
qr : Q → R,
p : P
⊢ R

case or.inr
P Q R : Prop,
pr : P → R,
qr : Q → R,
q : Q
⊢ R
```

Logic and Algebra 1

- As an example which involves both conjunction and disjunction, we prove *distributivity*. How to prove it in Lean?

```
variables P Q R: Prop

example : P ∧ (Q ∨ R) ↔ (P ∧ Q) ∨ (P ∧ R) :=
begin
  
end
```

- In algebra, we know a similar law $x(y+z) = xy + xz$.

Logic and Algebra 2

- What is the counterpart to implication?
- The translation of the law $x^{yz} = (x^y)^z$ corresponds to the currying equivalence $P \wedge Q \to R \leftrightarrow P \to Q \to R$.
- The translation of the law $x^{y+z} = x^y x^z$ is also a law of logic: $P \vee Q \to R \leftrightarrow (P \to R) \wedge (Q \to R)$.
- How to prove it in Lean?

```
variables P Q R: Prop

theorem case_thm: P ∨ Q → R ↔ (P → R) ∧ (Q → R) :=
begin

end
```

True, False and Negation 1

- There are two logical constants *true* and *false*.
- E.g., *It sometimes rains in Ningbo*. *Pigs can fly*.
- As far is logic and proof concerned, *true* is like an empty conjunction, and *false* is an empty disjunction.
- How to prove *true*?

```
variables P Q R: Prop

example : true :=
begin

end
```

True, False and Negation 2

- There is no way to prove *false*.

- Doing cases on *false* as an assumption makes the current goal go away and leaves no goals to be proven.

- How to prove *false* $\to$ *P* in Lean?

```
variables P Q R: Prop

theorem efq : false → P :=
begin

end
```

- *efq* is short for the latin phrase *Ex falso quod libet*, which means *from false follows everything*.

True, False and Negation 3

- We define $\neg P$ as $P \rightarrow \mathit{false}$, which means that P is impossible.
- Dear gentlemen, if you ask a girl to marry you and she replies *If we get married then pigs can fly*, this means *no*.

The Law of Contradiction

- The law of contradiction states that *it cannot be that both P and ¬P hold*.
- How to prove it in Lean?

```
variables P Q R: Prop

theorem contr: ¬ (P ∧ ¬ P) :=
begin

end
```

Summary of Tactics 1

Below is a table summarising the tactics we have seen so far:

	How to prove?	How to use?
$\rightarrow$	*assume h*	*apply h*
$\wedge$	*constructor*	*cases h with p q*
$\vee$	*left right*	*cases h with p q*
true	*constructor*	
false		*cases h*

They are related to introduction and elimination rules in natural deduction, a system devised by the German logician Gerhard Gentzen (1909-1945).

Summary of Tactics 2

- The syntax for using conjunction and disjunction is the same, *cases h with p q*, but the effect is different.
- In the case of conjunction, both assumptions are added to the context.
- In the case of disjunction, each assumption is added to one subproof.

Summary of Tactics 3

- In addition, we also introduced *exact h* which is a structural tactic.
- There is an alternative. The tactic *assumption* checks whether any assumption matches the current goal.
- How to prove $P \rightarrow P$ using *assumption* in Lean?

```
variables P : Prop

theorem I: P → P :=
begin
  
end
```

Some Important Notes

- There are many more tactics available in Lean some with a higher degree of automatisation.
- Some of the tactics we have introduced are applicable in ways we have not explained.
- **When solving exercises, please use only the tactics we have introduced and only in the way we have introduced them.**