Introduction to Mathematical Logic

For CS Students

CS104

Yida TAO (陶伊达)

2025 年 4 月 21 日

Table of Contents

Limitations of Propositional Logic

Can we express the following sentences using propositional logic?

- Alice is married to Jay and Alice is not married to Leon.
- Every student is younger than some instructor.
- Every even integer greater than 2 is the sum of two primes.

Limitations of Propositional Logic

Can we express the following sentences using propositional logic?

- Alice is married to Jay and Alice is not married to Leon. (relations among individuals)
- Every student is younger than some instructor.(generalizing patterns)
- Every even integer greater than 2 is the sum of two primes. (infinite domains)

Limitations of Propositional Logic

Can we express the following sentences using propositional logic?

- Every man is mortal.
- Socrates is a man.
- Socrates is mortal.

Can we prove the 3rd sentence using the first 2 as premises?

Limitations of Propositional Logic

The smallest unit of propositional logic is the proposition.

We cannot delve into individual propositions for more detailed analysis, such as analyzing objects, properties, relations, quantity, etc.

First-order logic (also known as Predicate Logic) is can overcome this limitation and is much more expressive than propositional logic. FOL can be used to express (most) scientific theories.

Table of Contents

2 Basic Concepts of FOL

Domain

A domain (论域) is a *non-empty* set of objects.
It is a world that our statement is situated within.

Examples of domains: natural numbers, people, animals, etc.

Why is it important to specify a domain?

Domain

Consider the statement:

"There exists a number whose square is 2."

- If our domain is the set of *natural numbers*, is this statement true or false?
- If our domain is the set of *real numbers*, is this statement true or false?

The same statement can have different truth values in different domains.

How to represent objects in a domain?

Constants

Constants: concrete objects in the language (i.e., domain elements)

- Example 1: Constants in "Alice is married to Jay and Alice is not married to Leon": Alice, Jay, Leon
- Example 2: Constants in the domain of natural numbers: $0, 1, 5, 1000, \ldots\ldots$
- Example 3: Constants in the domain of animals (in animation): Winnie the Pooh, Mickey Mouse, Simba, ...

How to represent objects in "Every student is younger than some instructor"?

Variables

"Every student is younger than some instructor."

Variables: "place holders" for concrete values.

- Variables are written $u, v, w, x, y, z, \ldots$ or $x_1, y_3, u_5, \ldots$
- A variable lets us refer to an object without specifying which particular object it is (e.g., a student).

How to describe properties of the object ("being a student", "being an instructor") or relations between objects ("younger than")?

Predicates

"Every student is younger than some instructor."

- A predicate (谓词) represents:
  - A property of an individual object in the domain, or
  - a relationship among multiple individuals
- Example: S, I and Y are predicates:
  - $S(andy)$: Andy is a student
  - $I(x)$: x is an instructor
  - $Y(andy, y)$: Andy is younger than y.
- A predicate can have a different number of arguments. S and I have just one (*unary predicates*), Y has two (*binary predicate*).

"Every student is younger than some instructor."

How do we describe "every" and "some"?

More generally, how do we describe:

For how many objects in the domain is the statement true?

Quantifiers

"Every student is younger than some instructor."

Quantifiers (量词): the quantity of objects

- The universal quantifier $\forall$ (全称量词): the statement is true for every object in the domain.
- The existential quantifier $\exists$ (存在量词): the statement is true for one or more objects in the domain.

Read as:

- $\forall x$: "for all x", "every x"
- $\exists x$: "there exists x" or "for some x"

Quantifiers

Let P be a property, and $P(x)$ denote that x has property P:

- Universal proposition (全称命题): $\forall x P(x)$, denotes that every individual in the domain has property P.
- Existential proposition (存在命题): $\exists x P(x)$, denotes that there exists an individual x in the domain with property P.

Universal and existential quantifiers can be interpreted as generalizations of conjunction and disjunction, respectively. In the case where the domain D is a finite set, let $D = \{a_1, a_2, \ldots, a_n\}$, the following equivalence hold:

$$\forall x P(x) \Leftrightarrow P(a_1) \wedge \ldots \wedge P(a_n)$$

$$\exists x P(x) \Leftrightarrow P(a_1) \vee \ldots \vee P(a_n)$$

For statements involving an infinite domain (recall the warm up), quantifiers are naturally required.

Functions

"Every child is younger than its mother."

- In addition to writing $M(x, y)$ to mean that x is y's mother, we can also write $m(y)$ to mean y's mother.
- The symbol m is a function symbol: a function has arity n and sometimes denoted as $f^{(n)}$.
- In the example, m is a unary function: it takes one argument and returns the mother of that argument.

Every scientific theory has its objects of study, which form a non-empty set called the domain.

The elements in the domain, i.e., the objects under study, are individuals (constants or variables).

A scientific theory also studies the relations among individuals, including properties of individuals, which are predicates.

A scientific theory also studies the functions acting on individuals.

Table of Contents

Alphabet

The alphabet of a first-order language $\mathscr{L}$ consists of the set of non-logical symbols and logical symbols:

Non-logical symbols (非逻辑符号):

1. Constant symbols (个体常元): usually $c_1, c_2, c_3, \ldots$
2. Predicates (谓词、关系符号): denoted by uppercase letters (or with subscripts); superscript indicates arity, such as $P, Q, P_1, P_2, \ldots, Q_1^1, Q_1^2, \ldots$ (n-ary predicate)
3. Function symbols (函词): denoted by lowercase letters (or with subscripts); superscript indicates arity, such as $f, g, h, f_1, f_2^1, \ldots, g_1^2, \ldots$ (n-ary function)

Alphabet

Logical symbols (逻辑符号):

4. Quantifiers: $\forall$, $\exists$
5. Variables (个体变元): usually $x, y, z, x_1, x_2, ...y_1, y_2, ...$
6. Connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
7. Punctuation: $($, $)$, and ,
8. Equality (A special binary relation): $=$

All first-order languages have the same logical symbols, the meaning of which are fixed by the syntax and semantics.
Their differences lie only in non-logical symbols (constants, predicates and functions).
They may be assigned any meaning, consistent with their kind and arity.

Given the alphabet of $\mathscr{L}$, an expression (string, symbol string) of $\mathscr{L}$ is an ordered n-tuple composed of symbols from the alphabet of $\mathscr{L}$.

Given the expressions of $\mathscr{L}$, we define the set of terms, atomic formulas, and formulas of $\mathscr{L}$, denoted as $Term(\mathscr{L})$, $Atom(\mathscr{L})$, and $Form(\mathscr{L})$, respectively.

Definition

Terms (项): defined inductively as

- Constant symbols and variables are (atomic) terms.
- If f^n is a function symbol of arity n, and $t_1, t_2, \ldots, t_n$ are terms, then $f^n(t_1, t_2, \ldots, t_n)$ is a term.
- Nothing else is a term.

Intuitively, *terms* are expressions referring to "objects".

Examples of Terms

Let's suppose that

- $0, 1$ are constant symbols
- s^1 is a unary function.
- f^2, g^2 are binary functions.

Then,

- $0, 1, x, y, s(1), s(x)$ are all terms
- $f(x, s(y))$ is a term
- $g(f(0, f(s(x), y)), 1)$ is a term.

Examples of Terms

Let's suppose that

- $0, 1$ are constant symbols
- s^1 is a unary function.
- f^2, g^2 are binary functions.

But, the following expressions are NOT terms for **violating the arity of functions**.

- $s(x, y)$
- $f(s, 1)$
- $g(1, 0, x)$

Let's define atomic formula (atom, 原子公式): predicates applied on terms.

Definition

An expression of $\mathscr{L}$ is an element of $Atom(\mathscr{L})$ if and only if it has one of the following two forms:

(i) $P(t_1, \ldots, t_n)$, where P is an n-ary predicate symbol, and $t_1, \ldots, t_n \in Term(\mathscr{L})$

(ii) $= (t_1, t_2)$ (also denoted as $t_1 = t_2$), where $t_1, t_2 \in Term(\mathscr{L})$

Intuitively, *atomic formulas* refer to **properties** or **relations** of objects.

Atomic Formulas

3 FOL as a Formal Language

Let's suppose:

- $0, 1$ are constant symbols
- s^1 is a unary function.
- f^2, g^2 are binary functions.
- R^2 is a binary predicate.

Atomic formulas or not?

- $R(x)$
- $R(0, y)$
- $R(R(0, y), 1)$
- $f(0, y) = g(x, f(1, 1))$

Formulas

Definition (FOL 的公式)

$\alpha \in Form(\mathscr{L})$ if and only if it can be generated (by finite use of) the following (i)∼(iv):

(i) $Atom(\mathscr{L}) \subseteq Form(\mathscr{L})$.

(ii) If $\alpha \in Form(\mathscr{L})$, then $(\neg\alpha) \in Form(\mathscr{L})$.

(iii) If $\alpha, \beta \in Form(\mathscr{L})$, then $(\alpha * \beta) \in Form(\mathscr{L})$, where $*$ is any one of $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$.

(iv) If $\alpha \in Form(\mathscr{L})$ and x is a variable, then $(\forall x\ \alpha) \in Form(\mathscr{L})$, $(\exists x\ \alpha) \in Form(\mathscr{L})$.

Conventions

- Parentheses can be omitted as in propositional logic.
- Parentheses around quantifiers can be omitted.

The Mathematical Structure of Natural Numbers

In the domain $\mathbb{N}$ (the set of natural numbers):

- the individual (constant) 0
- the unary functions s (successor)
- the binary functions f (addition) and g (multiplication)

Whether the following expressions are terms, atomic formulas, formulas?

- $f(x, 0)$
- $f(x, 1) = s(x)$
- $\forall x f(x, 0)$
- $\exists x(g(x, y) = s(x))$

Precedence

Precedence

- Parentheses dictate the order of operations in any formula.
- $\forall x$ and $\exists x$ have the same precedence level as $\neg$, which are higher than all binary connectives.
- Between $\neg$, $\exists$, and $\forall$, they are typically associate right to left.

Precedence and Conventions

Examples: add brackets to the following formulas.

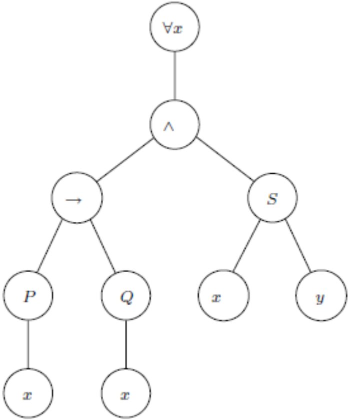
- $\exists x P(x, y) \lor Q(x, y)$
- $\neg \exists x \forall y \forall z R(x, y, z)$

We draw FOL parse tree in the same way as for propositional formulas, but with 3 additional sorts of nodes:

- The quantifiers $\forall x$ and $\exists y$ form nodes and have, like $\neg$, just one subtree.
- A predicate symbol $P(t_1, t_2, \ldots, t_n)$ has a node labelled P, which has n many subtrees, namely the parse trees of the terms $t_1, t_2, \ldots, t_n$.
- A function symbol $f(t_1, t_2, \ldots, t_n)$ has a node labelled f with n many subtrees for each of the terms $t_1, t_2, \ldots, t_n$

Parse Trees

The parse tree for the formula $\forall x((P(x) \to Q(x)) \land S(x, y))$.



Parse Trees

Example: draw the parse tree of $\forall x(\neg\exists y P(x, y) \vee \neg\exists y Q(y, x))$

Table of Contents

Use FOL to formalize the sentence

"Every student knows Math."
"Some student knows Math."

Let's define two predicates:

- $S(x)$: x is a student.
- $K(x, y)$: x knows y.

The sentence is formalized as:

$$\forall x(S(x) \to K(x, Math))$$

$$\exists x(S(x) \wedge K(x, Math))$$

Formalization

Use FOL to formalize the sentence

"Every student is younger than some instructor."

Let's define the predicates S, I, Y:

- $S(x)$: x is a student.
- $I(x)$: x is an instructor.
- $Y(x, y)$: x is younger than y.

The sentence is formalized as:

$$\forall x(S(x) \rightarrow (\exists y(I(y) \wedge Y(x, y))))$$

Use FOL to formalize the sentence

"Andy and Paul have the same maternal grandmother."

Let's define a function m:

- $m(x)$: the mother of x.

The sentence is formalized as:

$$m(m(andy)) = m(m(paul))$$

What if we use a binary predicate $M(x, y)$ to represent x is the mother of y?

Use FOL to formalize the sentence

"Every son of my father is my brother."

Design choice 1: represent "father" as a predicate.

- $S(x, y)$: x is a son of y
- $F(x, y)$: x is a father of y
- $B(x, y)$: x is a brother of y
- m: me

The symbolic encoding of the sentence: $\forall x \forall y (F(x, m) \wedge S(y, x) \rightarrow B(y, m))$
(But it's weird to say "every father")

Use FOL to formalize the sentence

"Every son of my father is my brother."

Design choice 2: represent "father" as a function.

- $S(x, y)$: x is a son of y
- $B(x, y)$: x is a brother of y
- $f(x)$: father of x
- m: me

The symbolic encoding of the sentence:

$$\forall x(S(x, f(m)) \to B(x, m))$$

Let's define in the domain $\mathbb{N}$:

- The constant 0
- The binary predicate R: $<$
- The unary function s: successor
- The binary functions f (addition) and g (multiplication)

Use FOL to formalize

- 0 is not the successor of any natural number.
- Two numbers are equal if and only if there successors are equal.
- x is an even number.

Exercises

Let's define in the domain $\mathbb{N}$:

- The constant 0
- The binary predicate R: $<$
- The unary function s: successor
- The binary functions f (addition) and g (multiplication)

Use FOL to formalize

- x is a prime number.
- There are infinitely many prime numbers.
- Describe "proof by induction"

Readings

- TextB: Chapter 2.1, 2.2
- TextF: Chapter 7.1, 7.2
- TextI: Chapter 2.1, 2.2

Coursework

Assignment 5 on FOL basics.

Introduction to Mathematical Logic

Thank you for listening!
Any questions?