Introduction to Mathematical Logic

For CS Students

CS104/CS108

Yida TAO (陶伊达)

2024 年 4 月 23 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.

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 subjects, predicates, quantifiers, etc.

First-order logic (also known as Predicate Logic) can overcome this limitation and is used to express (most) scientific theories.

Table of Contents

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: 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")?

"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"

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

3 FOL as a Formal Language

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): → 含义确定，而非逻辑符号未确定

4. Quantifiers: $\forall, \exists$
5. Variables (个体变元): usually $x, y, z, x_1, x_2, \ldots y_1, y_2, \ldots$
6. Connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
7. Punctuation: $($, $)$, and ,
8. Equality (A special binary relation): $=$ 不是函数，特殊二元谓词符号

是 binary perdicate (谓词)

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.

Syntax

3 FOL as a Formal Language

表示式是从字符表中符号组合成的

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.

Terms

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.
  得满函数所需个数
- Nothing else is a term.

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

Examples of Terms

Let's suppose that

- $0, 1, \ldots$ are constant symbols (nullary function)
- x and y are variables
- $s^{(1)}$ is a unary function.
- $+^{(2)}, -^{(2)}, *^{(2)}$ are binary functions.

Then,

- $0, x, y, s(2), s(x), +(x, s(y))$ are all terms
- We can also write $+(x, y)$ as $(x + y)$, which is also a term
- $*(-(2, +(s(x), y)), x)$ is a term.

Examples of Terms

Let's suppose that

- $0, 1, \ldots$ are constant symbols (nullary function)
- x and y are variables
- $s^{(1)}$ is a unary function.
- $+^{(2)}, -^{(2)}, *^{(2)}$ are binary functions.

But, the following expressions are not terms for violating the arity of s.

- $s(x, y)$ is not a term
- $(s + x)$ is not a term

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:

函数/"=" (个数合理)

(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.

Formulas

3 FOL as a Formal Language

Definition (FOL 的公式) 用 atom formula 形成 formula

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

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

(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 $(\forall x\ \alpha) \in Form(\mathscr{L})$, $(\exists x\ \alpha) \in Form(\mathscr{L})$.

$(\exists x\ \alpha)$, $(\forall x \alpha)$

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.

Conventions

() 在命题逻辑可省

- Parentheses can be omitted as in propositional logic.

() 在量词外可省

- Parentheses around quantifiers can be omitted.
- The order in which to consider $\neg$, $\exists$, and $\forall$ is determined by the order in which they are listed.

Precedence and Conventions

$((\neg(\forall x P(x))) \to Q(x,y))$

不同于 $(\forall x\, (P(x) \to Q(x)))$

Examples: add brackets to the following formulas.

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

$\neg\alpha$: $\neg$ — α $\forall x \alpha$: $\forall x$ — α $\exists x \alpha$: $\exists x$ — α

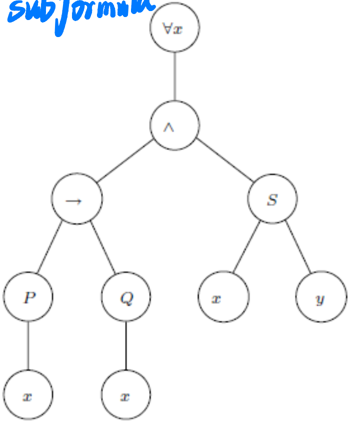
We draw FOL parse tree in the same way as for propositional formulas, but with 3 additional sorts of nodes:

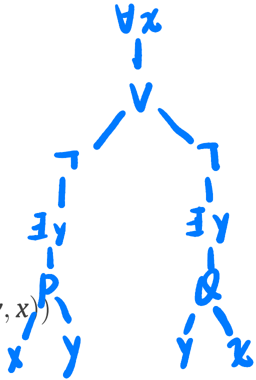
- The quantifiers $\forall x$ and $\exists y$ form nodes and have, like $\neg$, just one subtree.
- A predicate symbol $P(t_1, t_2, \ldots, t_n)$ has a node labelled P, which has n many subtrees, namely the parse trees of the terms $t_1, t_2, \ldots, t_n$.
- A function symbol $f(t_1, t_2, \ldots, t_n)$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

$\neg$ — $P^{(2)}$ — (x, y) $\forall x$ — $Q^{(2)}$ — ($f^{(1)}$ — x, y)

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

every subtree is a subformula





在树上更靠后

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

The Mathematical Structure of Natural Numbers

In the domain N (the set of natural numbers):

- the individual (constant) 0
- the relation $=$ (equality, a binary predicate)
- the unary functions $'$ (successor)
- the binary functions $+$ (addition) and $\cdot$ (multiplication)

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

(函数)

$\forall x\, \alpha$

α要是 formula ✗

- $x + 0$ — $+(x, 0)$ term
- $x + y = x'$ — $= (+(x,y), x')$ atomic formula (term, term)
- $\forall x(\underbrace{x + 0}_{\text{term}})$
- $\forall x(x + y = x')$

$= (+(x,y), x')$

Definition

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.

A Note on Quantifiers

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 $D = \{a_1, a_2, ..., a_n\}$, the following equivalence hold:

$$\forall x P(x) \Leftrightarrow P(a_1) \land ... \land P(a_n)$$

$$\exists x P(x) \Leftrightarrow P(a_1) \lor ... \lor P(a_n)$$

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

A Note on Quantifiers

The following equivalence hold:

- $\forall x P(x) \Leftrightarrow \neg \exists x \neg P(x)$
- $\exists x P(x) \Leftrightarrow \neg \forall x \neg P(x)$
- $\neg \forall x P(x) \Leftrightarrow \exists x \neg P(x)$
- $\neg \exists x P(x) \Leftrightarrow \forall x \neg P(x)$

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.

domain 不是所有学生

The sentence is formalized as:

if x is student, then x knows Math.

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

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

there exists such x that x is student and x knows Math

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. M

The sentence is formalized as:

$\forall x \forall y \forall u \forall w \, (M(x, andy) \land M(y, x) \land M(u, paul) \land M(w, u) \rightarrow =(y, w))$

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

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:

$$\forall x(S(x) \to (\exists y(I(y) \land Y(x, y))))$$

$x+1=0$ 不能判真值

$\forall x\ (x+1=0)$ $v=0$

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

$\forall x\ \forall y\ (F(x,m) \wedge S(y,x) \to B(y,m))$

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")

可交换

x/y的选择不依赖于x/y

反例

$\forall x(\forall y\ P(x,y))$

选出x后 y 有范围

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:

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

Use FOL to formalize the sentence

"For all natural numbers, there exists a prime number greater than it."

Design choice 1: Domain: All numbers including the set of natural numbers.

- $N(x)$: x is a natural number
- $P(x)$: x is a prime number
- $G(x, y)$: x is greater than y

The symbolic encoding of the sentence:

$$\forall x(N(x) \rightarrow \exists y(P(y) \land G(y, x)))$$

Use FOL to formalize the sentence

"For all natural numbers, there exists a prime number greater than it."

Design choice 2: Domain: Set of natural numbers.

- $P(x)$: x is a prime number
- $G(x, y)$: x is greater than y

The symbolic encoding of the sentence:

$$\forall x \exists y (P(y) \wedge G(y, x))$$

Use FOL to formalize the sentence

- "Not all birds can fly." $\exists x\,(B(x) \land \neg F(x))$
- "Every child is younger than its mother." $\forall x\,(C(x) \to Y(x, m(x)))$
- "Every even integer greater than 2 is the sum of two primes."

T

$$\forall x\,(I(x) \land T(x) \to \exists y \exists z\, E(x, y+z) \land P(y) \land P(z))$$

Let's define in the domain N (the set of natural numbers):

- The binary predicate R: $<$
- The binary functions f (addition) and g (multiplication)

Use FOL to formalize the sentence

- "x is an even number." $\exists y\,(g(2,y) = x)$ / $\exists y\,(f(y,y) = x)$
- "x is a prime number." $\exists y \exists z\,(g(y,z) = x \land \neg y = 1 \land \neg z = 1)$
- "There are infinitely many prime numbers."

$$(\forall x((\neg \exists y \exists z (g(y,z) = x \land \neg y = 1 \land \neg z = 1)$$
$$\rightarrow \exists w\,(<(x,w) \land \neg \exists a \exists b\,(g(a,b) = w \land \neg a = 1$$
$$\land \neg b = 1))))$$

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?