Introduction to Mathematical Logic

For CS Students

CS104/CS108

Yida TAO (陶伊达)

2024 年 3 月 5 日

南方科技大学

Table of Contents

A Logic Puzzle

1 Warm up

Alice, Alice's husband, their son, their daughter, and Alice's brother were involved in a murder. One of the five killed one of the other four. The following facts refer to the five people mentioned:

- A man and a woman were together in a bar at the time of the murder.
- The victim and the killer were together on a beach at the time of the murder.
- One of Alice's two children was alone at the time of the murder.
- Alice and her husband were not together at the time of the murder.
- The victim's twin was not the killer.
- The killer was younger than the victim.

Which one of the five was the victim? Write down your argument (论证).

What constitutes an argument?

1 Warm up

translate nature Language into formal language

→ 可以用 TOF 判断

- **Statements**: e.g., Alice was in a bar at the time of murder.
- **Logical indicators**: e.g., ...and..., ...or..., if...then..., ...because..., etc.
- **Common sense**: e.g., a father cannot be younger than his child; a parent and his or her child cannot be twins.
- **Inference rules**: e.g., if Alice was with her brother at the bar, then ..... we have a contradiction. Therefore, Alice was not with her brother in the bar.

推理规则

How to (mechanically) judge an argument is rigorous?

How to (mechanically) judge an argument is rigorous?

- To make arguments rigorous, we need to develop a language in which we can express sentences in such a way that brings out their logical structure.
- The language we begin with is the language of propositional logic.
- Propositional logic is based on **propositions**, or **declarative sentences**, which one can argue as being true or false.

Table of Contents

2 Propositions & Connectives

Definition

- **Proposition (命题)**: A proposition is a declarative sentence that can be judged as either true or false.
- **Atomic Proposition (原子命题)**: A proposition that does not contain any smaller part that is still a proposition is called an atomic proposition.
- **Compound Proposition (复合命题)**: A proposition that involves the assembly of multiple propositions is called a compound proposition.

Which are Propositions?

- 雪是白的。
- 雪是黑的。
- 好大的雪啊！
- $2 + 2 = 5$
- Where are you going?
- Ready? Go!
- $x + y < 0$
- This sentence is false.

Logical Connectives

连接多个命题

Words that connect multiple propositions to form a compound proposition are called **logical connectives**. For example:

- … and … (并且)
- not … (并非)
- … or … (或者)
- if … then … (如果… 那么…)
- … if and only if … (当且仅当)

Examples of Compound Propositions

- 2 is not an odd number.
- 2 is both an even number and a prime number.
- If a quadrilateral (四边形) has a pair of opposite sides that are parallel and equal, then it is a parallelogram (平行四边形).

Symbols

2 Propositions & Connectives

- To bring out the logical form of declarative sentences in natural language, we translate them into strings of symbols.
- This gives us a compressed but still complete encoding of declarative sentences and allows us to concentrate on the mere mechanics of our argumentation.

Symbols

- **Atomic proposition**:
  $p, q, r, \ldots\ldots$
- **Compound proposition**:
  $A, B, C, \ldots\ldots$

Connectives	Read as	名称	Term
$\wedge$	and	合取	conjunction
$\vee$	or	析取	disjunction
$\neg$	not	否定	negation
$\rightarrow$	if...then...	蕴含	implication
$\leftrightarrow$	if and only if	等价	equivalence

Symbolic Representation

2 Propositions & Connectives

p: Today is Friday.	Today is not Friday.	$\neg p$
p: 2 is a prime number, q: 2 is an even number.	2 is both a prime and an even number.	$p \wedge q$
p: Freshmen take Java classes, q: Freshmen take Python classes.	Freshmen take either Java or Python classes.	$p \vee q$
p: It will rain tomorrow, q: I will stay home and read.	If it rains tomorrow, then I will stay home and read.	$p \rightarrow q$
p: A triangle is an isosceles triangle, q: A triangle has two equal angles.	A triangle is an isosceles triangle if and only if it has two equal angles.	$p \leftrightarrow q$
p: 你是大一新生，q: 你能在寝室用电脑。	只有你不是大一新生，才能在寝室用电脑。	$p \rightarrow \neg q$

可能有其他限制！

Table of Contents

3 From Symbols to Language

How do we organize symbols (e.g., $p, q, r, ..., \wedge, \vee, ...$) to form a language?

Languages in General

3 From Symbols to Language

- All languages have a set of symbols, rules for constructing compound constructions out of atomic constructions, and meanings assigned to the significant units.
  - For example, the letter ”A” is part of English, but not part of Hindi.
- The syntax of a language is the grammar of the language.
  - For example, English is a Subject-Verb-Object language, while Arabic is Subject-Object-Verb.
- The semantics of a language is the meaning of the significant parts.
  - For example, ”snow” means snow in English, but ”schnee” means snow in German.

Formal Language

3 From Symbols to Language

A formal language consists of words whose letters are taken from an alphabet and are well-formed according to a specific set of rules called a formal grammar.

- Alphabet: A set of symbols.
- String: A finite sequence of symbols from the alphabet.
- A formal language is a set of strings all over the same alphabet.

Formal Language

3 From Symbols to Language

We use whichever alphabet is suitable to the task at hand when working with formal languages: the standard Roman alphabet for processing text, decimal digits for processing numbers, the alphabet A, T, C, G for processing genetic data, etc.

	symbols
binary	01 (or ab)
Roman	abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ
decimal	0123456789
special	~`!@#$%^&*()_-+={[}]|\:;"'<,>.?/
keyboard	*Roman + decimal + special*
genetic code	ATCG
protein code	ACDEFGHIKLMNPQRSTVWY

Formal Language

3 From Symbols to Language

	in the language	*not in the language*
palindromes	madamimadam amanaplanacanalpanama amoraroma	madamimbob madam, i'm adam not a palindrome
odd integers	3 101 583805233	2 100 2147483648
amino acid encodings	AAA ACA ATA AGA CAA CCA CTA CGA TCA TTA GAA GCA GTA GGA AAC ACC	CCC AAAAAAAA ABCDE
U.S. telephone number	(609) 258-3000 (800) 555-1212	(99) 12-12-12 2147483648
English words	and middle computability	abc niether misunderestimate
legal English sentences	This is a sentence. I think I can.	xya a b.c.e?? Cogito ergo sum.
legal Java identifiers	a class $xyz3_XYZ	12 123a a((BC))*
legal Java programs	public class Hi { public static void main(String[] args) { } }	int main(void) { return 0; }

Propositional Logic

3 From Symbols to Language

- The language of propositional logic, PL, is a formal language used in logic to study and analyze propositions and their logical relationships.
- It consists of a set of symbols and logical connectives that represent simple statements (propositions) and allow the construction of compound propositions.
- It is possible to translate sentences of most natural languages, such as Greek, English, German, French, etc... into PL.

Propositional Logic

3 From Symbols to Language

We will introduce the formal language of propositional logic $\mathscr{L}^P$ in terms of:

- Symbols and Alphabet
- Syntax: The rules governing the arrangement of symbols to form valid strings.
- Semantics: Truth values
- Inference rules and deduction systems

Table of Contents

4 Propositional Logic as a Formal Language

Alphabet of $\mathscr{L}^P$

$\mathscr{L}^P$, as the language of proposition logic, has 3 types of symbols.

- Atomic proposition (Atom): $p, q, r, \ldots$
- Logical connectives: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
- Punctuation: (and)

Expressions of $\mathscr{L}^P$

- Expressions (表达式): finite strings of symbols, e.g., p, $pqqr$, $(\neg r)$, $p \wedge q \rightarrow r$.
- The length of an expression is the number of occurrences of symbols in it.
- Empty expression: an expression of length 0, denoted by λ.
- Two expressions u and v are equal if they are of the same length and have the same symbols in the same order.
- An expression is read from left to right.

Formulas of $\mathscr{L}^P$

The well-formed formulas of propositional logic are expressions which we obtain by using the construction rules below, and only those, finitely many times:

- Atom: Every propositional atom $p, q, r, \ldots$ is a well-formed formula.
- $\neg$: If ϕ is a well-formed formula, then so is $(\neg\phi)$.
- $\wedge$: If ϕ and ψ are well-formed formulas, then so is $(\phi \wedge \psi)$.
- $\vee$: If ϕ and ψ are well-formed formulas, then so is $(\phi \vee \psi)$.
- $\rightarrow$: If ϕ and ψ are well-formed formulas, then so is $(\phi \rightarrow \psi)$.
- $\leftrightarrow$: If ϕ and ψ are well-formed formulas, then so is $(\phi \leftrightarrow \psi)$.

Formulas of $\mathscr{L}^P$

Definition 3.1 Atom($\mathscr{L}^P$)

Atom($\mathscr{L}^P$) is the set of expressions of $\mathscr{L}^P$ consisting of a proposition symbol only.

Definition 3.2 Form($\mathscr{L}^P$)

An expression $A \in Form(\mathscr{L}^P)$ if and only if its being so follows from (i)$\sim$(iii):

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

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

(iii) If A,B $\in$ Form($\mathscr{L}^P$), then (A $\wedge$ B), (A $\vee$ B), (A $\rightarrow$ B), (A $\leftrightarrow$ B) $\in$ Form($\mathscr{L}^P$).

Formulas of $\mathscr{L}^P$

Inductive definitions, like the one of well-formed propositional logic formulas above, are so frequent that they are often given by a defining grammar in Backus Naur form (BNF).

In that form, the above definition reads more compactly as:

$$\phi ::= p|(\neg\phi)|(\phi \wedge \phi)|(\phi \vee \phi)|(\phi \rightarrow \phi)|(\phi \leftrightarrow \phi)$$

Formulas of $\mathscr{L}^P$

Is this a well-formed formula?

$$(((\neg p) \leftrightarrow (q \vee r)) \rightarrow (r \wedge p))$$

Parse Tree

Definition 3.3

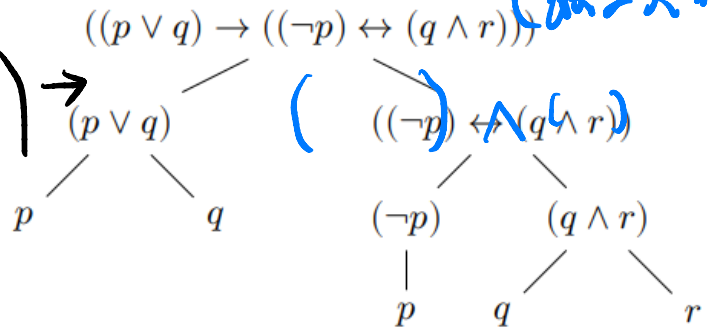
A parse tree of a formula in $\mathscr{L}^P$ is a tree such that

- The root is the formula.
- Leaves are atomic formulas, and
- Each internal node is formed by applying some formation rule on its children.

Parse Tree

4 Propositional Logic as a Formal Language

A parse tree of $((p \vee q) \rightarrow ((\neg p) \leftrightarrow (q \wedge r)))$

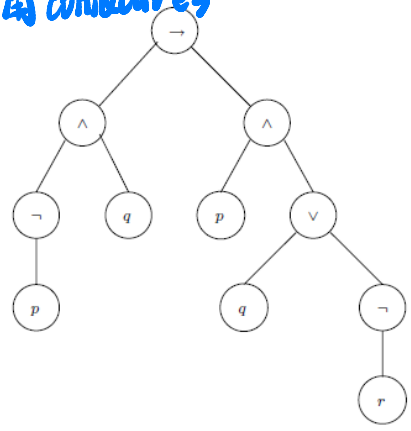


Parse Tree

4 Propositional Logic as a Formal Language

A simplified parse tree of $(((\neg p) \wedge q) \rightarrow (p \wedge (q \vee (\neg r))))$

node 是 connectives



Parse Tree

公式良好 ⇔ 有 parse tree

Theorem 3.1

An expression of $\mathscr{L}^P$ is a well-formed formula if and only if there is a parse tree of it.

Theorem 3.2

Each well-formed formula of $\mathscr{L}^P$ has a unique parsing tree.

Subformula

Definition 3.4

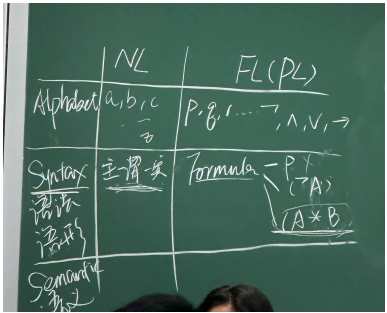
A formula G is a subformula of formula F if G occurs within F. G is a proper subformula of F if $G \neq F$.
The nodes of the parse tree of F form the set of subformulas of F.

Definition 3.5

Immediate subformulas are the children of a formula in its parse tree, and *leading connective* is the connective that is used to join the children.

Q: The subformulas and leading connective for $((\neg p) \leftrightarrow (q \wedge r))$?

Examples



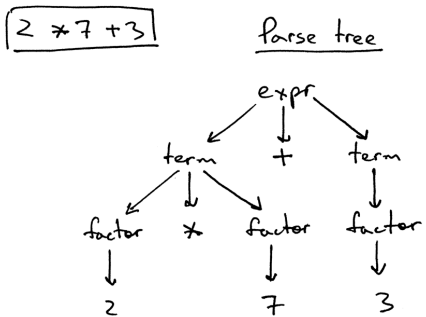
Which are well-formed formulas in $\mathscr{L}^P$?

- (p)
- $(\neg p)$
- $((q \vee$
- $((p \neg \wedge) q (r \neg$
- $(((\neg p) \leftrightarrow (q \vee r)) \rightarrow (r \wedge p))$

*只是p才是well-formed

Applications

A parse tree is created by a parser, which is a component of a compiler that processes the source code and checks it for syntactic correctness.

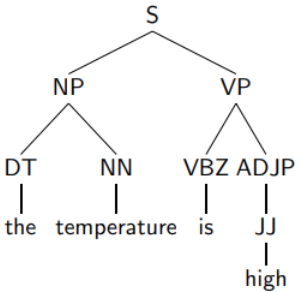


Applications

4 Propositional Logic as a Formal Language

自然语言处理

In NLP, a parse tree can be used to computationally reason about a natural-language sentence.



The Structure of Formula

4 Propositional Logic as a Formal Language

Let's consider important properties regarding the structure of formulas.

Lemma 3.1

Well-formed formulas of $\mathscr{L}^P$ are non-empty expressions.

Lemma 3.2

Every well-formed formula of $\mathscr{L}^P$ has an equal number of opening and closing brackets.

Base : P

Induction Hypothesis : P(A) P(B)

Inductive Step : $(\neg A)$ $(A * B)$

The Structure of Formula

4 Propositional Logic as a Formal Language

- uv denotes the result of concatenating two expressions u, v in this order. Note that $\lambda u = u\lambda = u$.
- v is a segment of u if $u = w_1 v w_2$ where u, v, w_1, w_2 are expressions.
- v is a proper segment of u is v is non-empty and $v \neq u$.
- If $u = vw$, where u, v, w are expressions, then v is an initial segment (prefix) of u, w is a terminal segment (suffix) of u.
- If $u = vw$, where u, v, w are expressions, and v, w are non-empty, then v is an proper prefix of u, w is a proper suffix of u.

Lemma 3.3

- Every proper prefix of a well-formed formula in $\mathscr{L}^P$ has more opening brackets than closing brackets.
- Similarly, every proper suffix of a well-formed formula in $\mathscr{L}^P$ has more closing brackets than opening brackets.
- Hence, proper prefix and proper suffix are not formulas in $\mathscr{L}^P$ (Lemma 3.2).

不是 well-formed formula

The Structure of Formula

Every proper prefix of a well-formed formula in $\mathscr{L}^P$ has more opening brackets than closing brackets.

Proof by (structural) induction:
Let $P(\varphi)$ be the property that every proper prefix of the well-formed formula φ has more opening brackets than closing brackets. We prove $P(\varphi)$ is true for all well-formed formulas φ by structural induction.

Assumption doesn't holds 无意义的

Base Case: If φ is an atom, then there are no proper prefixes and the claim is vacuously true.
Inductive Hypothesis: Assume that $P(\alpha)$ and $P(\beta)$ are true for some well-formed formulas α and β.

Complete the induction step by yourself.

Unique Readability Theorem

There is a unique way to construct every well-formed formula.

Proof by induction

Let $P(\varphi)$ be the property that there is a unique way to construct the well-formed formula φ. We prove this property for all well-formed formulas φ by structural induction.

Base case: There is only one way to construct an atom.

Inductive Hypothesis: Assume that $P(\alpha)$ and $P(\beta)$ are true for some well-formed formulas α and β.

Unique Readability Theorem

There is a unique way to construct every well-formed formula.

Proof by induction (continued)

Inductive Step: Consider two possibilities:

- $\varphi = (\neg\alpha)$ assume another formula can be used to represent φ
- $\varphi = (\alpha \star \beta)$ ① $(\neg\alpha') \Rightarrow \alpha = \alpha'$ lead to contradiction

(Proved in class) ② $\varphi = (\alpha' \star \beta')$ 去括号 $\alpha \star \beta) = \alpha$

Conventions

For readability, we employ the following convention, which states that we may drop the outermost parentheses.

If F or (F) is a formula, then we view F and (F) as the same formula.

But, how do we interpret formulas like: $p \wedge q \rightarrow \neg r \vee q$?

Precedence

Parentheses are used to resolve ambiguity. But they are hard to read.
If no parentheses are present, we could use **precedence** and **associativity** to disambiguate formulas.

- Each connective on the left has priority over those on the right. $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.
- Parentheses take the highest precedence.
- Connectives are assumed to associate to the right (right associative), i.e., first group the rightmost occurrence. For example, $p \rightarrow q \rightarrow r$ means $p \rightarrow (q \rightarrow r)$

Examples

Add back the brackets based on the precedence rules.

- $\neg p \rightarrow q$
- $p \wedge q \rightarrow r$
- $p \wedge q \rightarrow \neg r \vee q$
- $\neg p \rightarrow p \wedge \neg q \vee r \leftrightarrow q$

Readings

Optional

- TextA: Chapter 2
- TextB: Chapter 1.3
- Text1: 第二章 2.1, 2.2, 2.3
- Text3: 第二章 2.1, 2.2, 2.4

Assignments

Coursework

- Assignment 2

Introduction to Mathematical Logic

Thank you for listening!
Any questions?