Introduction to Mathematical Logic

For CS Students

CS104

Yida TAO (陶伊达)

2025 年 3 月 10 日

Table of Contents

What’s the meaning of well-formed formulas

The meaning of a natural language sentence depends on the meaning of words and connectives.

- ”The sky is blue and the grass is green.”
- ”Nothing right in my left brain, and nothing left in my right brain.”

Similarly, the meaning of well-formed formulas in a formal language (e.g., $p \wedge q$) depends on the truth values of the *atoms* and how the *logical connectives* manipulate these truth values.

Table of Contents

Truth Table

- Truth values of atomic propositions: True(1) or False(0)
- Truth values of compound propositions: depends on its atomic propositions and logical connectives
- Truth table for $\neg p$:

p	$\neg p$
1	0
0	1

Truth Table

- Truth values of atomic propositions: True(1) or False(0)
- Truth values of compound propositions: depends on its atomic propositions and logical connectives
- Truth table for $p \wedge q$:

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

Logical Connectives in NL & PL

Logical connectives in formal languages are not completely equivalent to words in natural language.

She became violently sick and she went to the doctor.
She went to the doctor and she became violently sick.

In natural language, ”and” indicates time progression; whereas logical connectives in formal language like PL are solely concerned with truth values.

- Truth values of atomic propositions: True(1) or False(0)
- Truth values of compound propositions: depends on its atomic propositions and logical connectives
- Truth table for $p \vee q$:

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

"He will come today or tomorrow"
"From $(a-1)(a-2)=0$, we get $a=1$ or $a=2$."

In natural language, sometimes "A or B" means "A is true or B is true but not both".

In PL, however, "A or B" means "A is true or B is true or both are true."

Truth table for $p \rightarrow q$ (logical implication): if p (is true), then q (is true).

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Also known as: It’s not the case that p is true and q is false.

Truth Table

Why $p \to q$ is true when p is false?

- If $x > 7$, then $x > 5$.
- “If you stick a fork in an electrical outlet, you will get hurt.”

p	q	$p \to q$
0	0	1
0	1	1
1	0	0
1	1	1

$p \leftrightarrow q$ (iff) is true when both p and q carry the same truth value, and is false otherwise.

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

To interpret a formula, we have to give meanings to the propositional variables and the connectives.
A propositional variable gets a meaning via a valuation (赋值/真值指派).

Definition

A truth valuation is a function $v : Atom(\mathscr{L}^P) \to \{0,1\}$.

For $p \in Atom(\mathscr{L}^P)$, we use $v(p)$ or p^v to denote the truth value of p under the truth valuation v, $p^v \in \{0,1\}$

Fix a truth valuation v. Every formula A has a value under v, denoted as A^v, can be recursively defined as:

Definition

1. $p^v \in \{0, 1\}$
2. $(\neg A)^v = \begin{cases} 1 & \text{if } A^v = 0 \\ 0 & \text{else} \end{cases}$
3. $(A \wedge B)^v = \begin{cases} 1 & \text{if } A^v = B^v = 1 \\ 0 & \text{else} \end{cases}$
4. $(A \vee B)^v = \begin{cases} 1 & \text{if } A^v = 1 \text{ or } B^v = 1 \\ 0 & \text{else} \end{cases}$
5. $(A \rightarrow B)^v = \begin{cases} 1 & \text{if } A^v = 0 \text{ or } B^v = 1 \\ 0 & \text{else} \end{cases}$
6. $(A \leftrightarrow B)^v = \begin{cases} 1 & \text{if } A^v = B^v \\ 0 & \text{else} \end{cases}$

Theorem

Fix a truth valuation v. Every formula $\alpha \in Form(\mathscr{L}^p)$ has a value α^v in $\{0, 1\}$.

Proof: By structural induction. Let $P(\alpha)$ be "α has a value α^v in $\{0, 1\}$".

1. If α is a propositional variable, then v assigns it a value of 1 or 0 (by the definition of a truth valuation).
2. If α has a value in $\{0, 1\}$, then $(\neg\alpha)$ also does (by the truth table of $(\neg\alpha)$).
3. If α and β each has a value in $\{0, 1\}$, then $(\alpha \star \beta)$ also does for every binary connective $\star$, as shown by the corresponding truth tables.

By the principle of structural induction, every formula has a value.
By the unique readability of formulas, we have proved that a formula has only one truth value under any truth valuation v. QED

Truth Valuation

What is the truth value of $A = p \land q \to (\neg q \lor r)$ given the following truth valuation?

- $p^v = q^v = r^v = 1, A^v = ?$
- $p^v = q^v = r^v = 0, A^v = ?$

Semantic Properties

Let $A \in Form(\mathscr{L}^p)$.

- If for every truth valuation v, $A^v = 1$, then A is tautology (永真式或重言式)
- If for every truth valuation v, $A^v = 0$, then A is contradiction (永假式或矛盾式)
- If there exists a truth value v such that $A^v = 1$, then A satisfiable (可满足的)

Semantic Properties

How to determine the semantic properties (tautology, contradiction, satisfiable) of a well-formed formula?

- Intuition
- Valuation tree
- Proof by contradiction
- Truth table

Valuation Tree

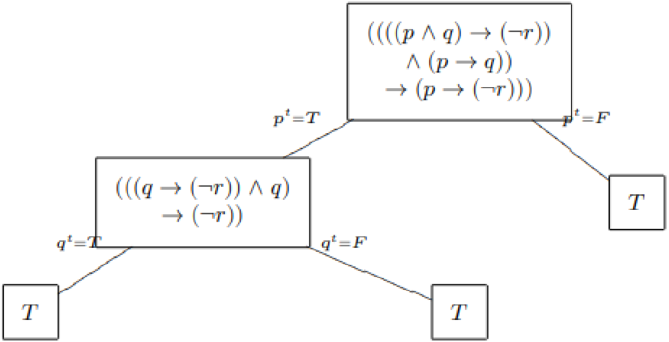
We can plug in a truth value for one variable, and keep simplifying the formula using the following rules:

$$
\begin{array}{c|c}
\neg \mathrm{T} & \mathrm{F} \\
\neg \mathrm{F} & \mathrm{T}
\end{array}
\qquad
\begin{array}{c|c}
(p \wedge \mathrm{T}) & p \\
(p \wedge \mathrm{F}) & \mathrm{F} \\
(p \wedge p) & p
\end{array}
\qquad
\begin{array}{c|c}
(p \vee \mathrm{T}) & \mathrm{T} \\
(p \vee \mathrm{F}) & p \\
(p \vee p) & p
\end{array}
\qquad
\begin{array}{c|c}
(p \rightarrow \mathrm{T}) & \mathrm{T} \\
(p \rightarrow \mathrm{F}) & (\neg p) \\
(\mathrm{T} \rightarrow p) & p \\
(\mathrm{F} \rightarrow p) & \mathrm{T} \\
(p \rightarrow p) & \mathrm{T}
\end{array}
$$

We can evaluate a formula by using these rules to construct a valuation tree.

Valuation Tree

Show that $((((p \wedge q) \to (\neg r)) \wedge (p \to q)) \to (p \to (\neg r)))$ is a tautology by using a valuation tree.



Proof by Contradiction

Prove that the formula below is tautology.

$$((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$$

We could use **truth table** to determine the properties of a formula

p	q	r	$(p \vee q)$	$(q \wedge r)$	$((p \vee q) \rightarrow (q \wedge r))$
F	F	F	F	F	T
F	F	T	F	F	T
F	T	F	T	F	F
F	T	T	T	T	T
T	F	F	T	F	F
T	F	T	T	F	F
T	T	F	T	F	F
T	T	T	T	T	T

Here's a question about playing Monopoly:

> *If you get more doubles than any other player then you will lose, or if you lose then you must have bought the most properties.*

True or false? We will answer this question, and won't need to know anything about Monopoly. Instead we will look at the logical form of the statement.

Finding live/dead code: can code block P_1, P_2, P_3, P_4 be executed?

```
if ( (input > 0) or (not output) ) {
    if ( not (output and (queuelength < 100) ) ) {
        P1
    } else if ( output and (not (queuelength < 100)) ) {
        P2
    } else { P3 }
} else { P4 }
```

Finding live/dead code: can code block P_1, P_2, P_3, P_4 be executed?

```
if ( (input > 0) or (not output) ) {
    if ( not (output and (queuelength < 100) ) ) {
        P1
    } else if ( output and (not (queuelength < 100)) ) {
        P2
    } else { P3 }
} else { P4 }
```

Let's define i: input> 0, u: output, and q: queuelength < 100

Finding live/dead code: can code block P_1, P_2, P_3, P_4 be executed?

i	u	q	$(i \vee (\neg u))$	$(\neg(u \wedge q))$	$(u \wedge (\neg q))$	
T	T	T	T			
T	T	F	T			
T	F	T	T			
T	F	F	T			
F	T	T	F			P_4
F	T	F	F			P_4
F	F	T	T			
F	F	F	T			

Applications

2 Semantics

Finding live/dead code: can code block P_1, P_2, P_3, P_4 be executed?

i	u	q	$(i \vee (\neg u))$	$(\neg(u \wedge q))$	$(u \wedge (\neg q))$	
T	T	T	T	F	F	P_3
T	T	F	T	T		P_1
T	F	T	T	T		P_1
T	F	F	T	T		P_1
F	T	T	F			P_4
F	T	F	F			P_4
F	F	T	T	T		P_1
F	F	F	T	T		P_1

Table of Contents

Two formulas A and B are logically equivalent if and only if they have the same value **under any valuation**.

- $A^v = B^v$ for every truth valuation v.
- A and B must have the same final column in their truth tables.
- $A \leftrightarrow B$ is a tautology.

Why do we care about logical equivalence?

- Will I lose marks if I provide a solution that is syntactically different but logically equivalent to the provided solution?
- Do these two circuits behave the same way?
- Do these two pieces of code fragments behave the same way?

Logical Identities

Equivalence	Name
$p \wedge T \equiv p$ $p \vee F \equiv p$	Identity laws 恒等率
$p \vee T \equiv T$ $p \wedge F \equiv F$	Domination laws 支配率
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws 幂等率
$\neg(\neg p) \equiv p$	Double negation laws 双非率
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws 交换率

Logical Identities

3 Logical Equivalence

Equivalence	Name
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws 结合率
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws 分配率
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws 德摩根率
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws 吸收率
$p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$	Negation laws 否定率

Logical Identities

Equivalence	Name
$p \to q \equiv \neg p \vee q$	Implication
$p \to q \equiv \neg q \to \neg p$	Contrapositive (逆否)
$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$	Equivalence

Example: Prove $((\neg p) \wedge q) \vee p \equiv p \vee q$

Exercise

"If it is sunny, I will play golf, provided that I am relaxed."

- s: it is sunny
- g: I will play golf.
- r: I am relaxed.

Prove that all three translations are logically equivalent.

- $s \to (r \to g)$
- $r \to (s \to g)$
- $(s \land r) \to g$

Substitution

A substitution is a *syntactic transformation* on formal expressions.

Definition

$A, B \in Form(\mathscr{L}^p)$. B is a substitution instance of A if and only if B may be obtained from A by substituting formulas for propositional variables in A, replacing each occurrence of the same variable by an occurrence of the same formula.

Example:

- $(r \to s) \land (t \to s)$ is a substitution instance of $p \land q$
- $(p \leftrightarrow p) \leftrightarrow (p \leftrightarrow p)$ is a substitution instance of $p \leftrightarrow p$

Theorem

$A, B \in Form(\mathscr{L}^p)$. If A is a tautology, and B is a substitution instance of A, then B is again a tautology.

Example: $(r \wedge s) \to (q \to (r \wedge s))$ is a tautology, since it is a substitution instance of $p \to (q \to p)$

Substitution theorem

$A \in Form(\mathscr{L}^p)$. A contain a subformula C (i.e., C is a *segment* of A and is itself a well-formed formula). If $C \equiv D$, then replacing some occurrences (not necessarily all) of the subformula C in A with D to obtain the formula B, then $A \equiv B$.

Example: $p \to q \equiv (\neg p \vee q)$. Then, $(p \to q) \wedge (r \to (p \to q)) \equiv ?$

Logical equivalence is an equivalence relation on $Form(\mathscr{L}^p)$

- Reflexive: for any $A \in Form(\mathscr{L}^p)$, $A \equiv A$.
- Symmetric: for any $A, B \in Form(\mathscr{L}^p)$, $A \equiv B$, then $B \equiv A$.
- Transitive: for any $A, B, C \in Form(\mathscr{L}^p)$, if $A \equiv B$, $B \equiv C$, then $A \equiv C$.

All formulas in the same equivalence class have the same truth table.

The two code have different syntax, but equivalent semantics.

Listing 1: Your code

```
if (i || !u) {
  if (!(u && q)) {
    P1
  } else if (u && !q) {
    P2
  } else { P3 }
} else { P4 }
```

Listing 2: Your friend's code

```
if ((i && u) && q) {
  P3
} else if (!i && u) {
  P4
} else {
  P1
}
```

To prove that the two code are semantically equivalent, show that each code block is executed under logically equivalent conditions.

Block	Fragment 1	Fragment 2
P_1	$(i \vee (\neg u)) \wedge (\neg(u \wedge q))$	$(\neg(i \wedge u \wedge q)) \wedge (\neg((\neg i) \wedge u))$
P_2	$(i \vee (\neg u)) \wedge (\neg(\neg(u \wedge q)))$ $\wedge (u \wedge (\neg q))$	$\mathbf{F}$
P_3	$(i \vee (\neg u)) \wedge (\neg(\neg(u \wedge q)))$ $\wedge (\neg(u \wedge (\neg q)))$	$(i \wedge u \wedge q)$
P_4	$(\neg(i \vee (\neg u)))$	$(\neg(i \wedge u \wedge q)) \wedge ((\neg i) \wedge u)$

Readings

Optional

- TextB: 1.4.1
- TextI: 1.1, 1.2
- Text1: 第二章 2.4
- Text3: 第二章 2.3

[0] Part of the slides is based on CS245 course notes from University of Waterloo

Introduction to Mathematical Logic

Thank you for listening!
Any questions?