



数理逻辑

OI - 命题逻辑简介

(Press `?` for help, `n` and `p` for next and previous slide)

戴望州

南京大学智能科学与技术学院

2025年 - 春季

<https://daiwz.net>





语法





前情提要

- › “我可以怀疑物理世界是否存在。我甚至可以怀疑我的身体是否真的存在。但我不能怀疑我自己是否存在。所以，我不是我的身体。”
- › 原文：‘I can doubt that the physical world exists. I can even doubt whether my body really exists. I cannot doubt that I myself exist. So I am not my body.’
- › 这种推理似乎得到了笛卡尔的《沉思录》的支持（尽管我们可以争论归属）。这个论证有效吗？

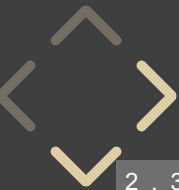




前情提要

回顾：根据形式化的一些基本假设，若能谈论命题 A 和命题 B 的真假，我们应该也能谈论命题“并非 A ”和“ A 并且 B ”，甚至“ A 蕴涵 B ”的真假。

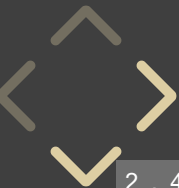
- › “命题”是陈述句，非真即假
 - » 分析命题 vs 综合命题
- › 命题联词可以连接较简单的命题形成较复杂的命题，如：“并非”、“并且”、“或者”、“蕴涵”……
- › 命题逻辑只考虑有关命题联词的真之规律
- › 命题逻辑是一阶谓词逻辑的片段（热身）





命题逻辑中的符号

符号	中文名	意义
(左括号	标点
)	右括号	标点
\neg	非运算符	并非
\wedge	合取运算符	与
\vee	析取运算符	或
\rightarrow	蕴涵符	若...则...
\leftrightarrow	双蕴涵符	当且仅当
A_1	第1个命题符	
A_2	第2个命题符	
...		
A_n	第n个命题符	
...	(可数无穷个)	



命题逻辑中的符号





命题逻辑中的符号

- I. 我们使用一些不属于“命题逻辑符号”中的符号来帮助我们进行书写和理解，这些符号成为元语言（meta-language）





命题逻辑中的符号

1. 我们使用一些不属于“命题逻辑符号”中的符号来帮助我们进行书写和理解，这些符号成为元语言（meta-language）
2. “符号”可构成序列
 - » 简写： $\langle (, \neg, \mathbf{A}_1,) \rangle = (\neg \mathbf{A}_1)$
 - » 表达式的长度是其中符号出现的数目
 - » 长度为0的表达式称为空表达式，用记号 ϕ 表示
 - » 这里的“ \langle ”、“ \rangle ”和“ ϕ ”都是元语言中的符号（vs “对象语言”）





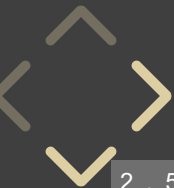
命题逻辑中的符号

1. 我们使用一些不属于“命题逻辑符号”中的符号来帮助我们进行书写和理解，这些符号成为元语言（meta-language）
2. “符号”可构成序列
 - » 简写： $\langle (, \neg, \mathbf{A}_1,) \rangle = (\neg \mathbf{A}_1)$
 - » 表达式的长度是其中符号出现的数目
 - » 长度为0的表达式称为空表达式，用记号 ϕ 表示
 - » 这里的“ \langle ”、“ \rangle ”和“ ϕ ”都是元语言中的符号（vs “对象语言”）
3. 命题逻辑语言中：任意一个符号都不是由其他符号构成的有穷序列
 - » 例如： $\mathbf{A}_3 \neq \leftrightarrow$ ， $\mathbf{A}_3 \neq \langle \neg, \mathbf{A}_4, () \rangle$
 - » 这意味着若

$$\langle a_1, \dots, a_m \rangle = \langle b_1, \dots, b_n \rangle$$

那么必有 $m = n$ 且 $a_i = b_i$ [Enderton, pp.4]

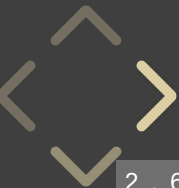
- » 这里的变元符号 a_i 、 b_j 同样属于元语言





表达式 (EXPRESSION)

定义 1.1: 表达式 (*expression*) 是有穷个符号构成的序列。





表达式 (EXPRESSION)

定义 I.I: 表达式 (*expression*) 是有穷个符号构成的序列。

例如:

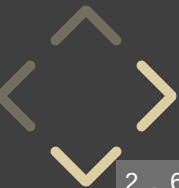
- > $(\neg \mathbf{A}_1)$, $((\mathbf{A}_2$
- > 若 α 和 β 是表达式, 则它们的拼接也是表达式, 例如:

$$\alpha = (\neg \mathbf{A}_1)$$

$$\beta = (\mathbf{A}_2)$$

则 $(\alpha \rightarrow \beta)$ 是

$$((\neg \mathbf{A}_1) \rightarrow \mathbf{A}_2)$$





合式公式 (WELL-FORMED FORMULAS)

定义 1.2 (wff) :

1. 每个命题符号 A_i 都是合式公式 (原子公式, atomic formula)
2. 若 α , β 是合式公式, 那么 $(\neg\alpha)$ 和 $(\alpha\Box\beta)$ 也是合式公式。其中 \Box 是 \wedge 、 \vee 、 \rightarrow 和 \leftrightarrow 中的一个
3. 除此以外都不是合式公式。





合式公式 (WELL-FORMED FORMULAS)

定义 1.2 (wff) :

1. 每个命题符号 \mathbf{A}_i 都是合式公式 (原子公式, atomic formula)
2. 若 α, β 是合式公式, 那么 $(\neg\alpha)$ 和 $(\alpha\Box\beta)$ 也是合式公式。其中 \Box 是 \wedge 、 \vee 、 \rightarrow 和 \leftrightarrow 中的一个
3. 除此以外都不是合式公式。

也可以用 Bacus-Naur Form 定义命题为

$$\varphi ::= \mathbf{A}_i \mid (\neg\varphi) \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid (\varphi_1 \rightarrow \varphi_2) \mid (\varphi_1 \leftrightarrow \varphi_2)$$





定义 1.3 (公式构造运算, *formula-building operations*) :

$$\mathcal{E}_{\neg}(\alpha) = (\neg\alpha)$$

$$\mathcal{E}_{\wedge}(\alpha, \beta) = (\alpha \wedge \beta)$$

$$\mathcal{E}_{\vee}(\alpha, \beta) = (\alpha \vee \beta)$$

$$\mathcal{E}_{\rightarrow}(\alpha, \beta) = (\alpha \rightarrow \beta)$$

$$\mathcal{E}_{\leftrightarrow}(\alpha, \beta) = (\alpha \leftrightarrow \beta)$$





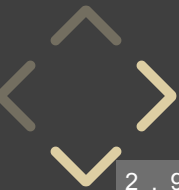
WFF 的构造序列

定义 I.4 (构造序列, *construction sequence*) :

有穷序列 $\langle \epsilon_1, \dots, \epsilon_n \rangle$ 被称为一个命题公式的构造序列, 当且仅当对任意 $i \leq n$ 满足以下条件之一:

1. ϵ_i 是一个命题符号
2. $\epsilon_i = \mathcal{E}_{\neg}(\epsilon_j)$, $j < i$
3. $\epsilon_i = \mathcal{E}_{\square}(\epsilon_j, \epsilon_k)$, $j < i$ 且 $k < i$

其中 \square 是 \wedge 、 \vee 、 \rightarrow 和 \leftrightarrow 中的一个





$$((\mathbf{A}_1 \wedge \mathbf{A}_{10}) \rightarrow ((\neg \mathbf{A}_3) \vee (\mathbf{A}_8 \leftrightarrow \mathbf{A}_3)))$$





结构递归





递归定义

> 自上而下:

$$S^* = \bigcap \{ S \mid S \text{ 包含所有命题符号且关于五元公式构造封闭} \}$$

> 自下而上:

$$\gg S_0 = \{ A_1, \dots \}$$

$$\gg S_i = \{ \neg \alpha \mid \alpha \in S_{i-1} \} \cup \{ \alpha \square \beta \mid \alpha \in S_{i-1}, \beta \in S_{i-1}, \square = \wedge, \vee, \rightarrow, \leftrightarrow \} \cup S_{i-1}$$

$$\gg S_* = \bigcup_n S_n$$





定理 1.5 [Enderton, pp.37]:

$$S^* = S_*$$





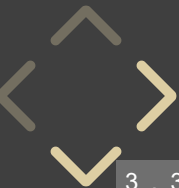
递归定义

定理 1.5 [Enderton, pp.37]:

$$S^* = S_*$$

记法: 令 $S = S^* = S_*$

- › 自下而上定义的好处在于展现了每个 wff 的构造过程
- › 自上而下的定义告诉我们 S 是一个闭包 (closure), 因此可以运用归纳原理 (Induction Principle) 进行证明和理解 [Enderton, pp.37]





归纳原理

定理 I.6（关于命题逻辑 wff 的归纳原理） [Hao, Yang and Yang, pp.28]:

令 $P(\alpha)$ 为一个关于 wff 的性质。假设：

1. 对所有命题符号 A_i ，性质 $P(A_i)$ 成立
2. 对所有的 wff α 和 β ，若 $P(\alpha)$ 和 $P(\beta)$ 成立，则 $P(\neg\alpha)$ 和 $P((\alpha \square \beta))$ 也成立，其中 \square 是 \wedge 、 \vee 、 \rightarrow 和 \leftrightarrow 中的任意一个

那么对于所有的 wff α 有性质 $P(\alpha)$ 成立。

证明：依结构归纳/强数学归纳法，显然。





归纳原理

例1.7: 证明 S 是无穷可数集, 即 $|S| = |\mathbb{N}|$





归纳原理

例1.7: 证明 S 是无穷可数集, 即 $|S| = |\mathbb{N}|$

证明概要:

1. 命题符号的个数为 $|\mathbb{N}|$
2. 任意wff均由有限步生成
3. $|\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$





括号引理

引理 1.8:

1. 每个 wff 中左右括号的数目相同
2. 每个 wff 的非空真初始段中左括号多于右括号。因此，wff 的真前段一定不是 wff





括号引理

引理 1.8:

1. 每个 wff 中左右括号的数目相同
2. 每个 wff 的非空真初始段中左括号多于右括号。因此，wff 的真前段一定不是 wff

1.8.a 证明概要 [Enderton, pp.19]:

记“括号平衡”为性质 P

1. Base: 对于原子公式 $\alpha = A_i$ 显然有 $P(\alpha)$ 成立
2. Induction step:
 - » Hypothesis: 设 $P(\alpha)$ 和 $P(\beta)$ 均成立
 - » $(\neg\alpha)$ 和 $(\alpha \square \beta)$ 仅添加最外侧括号，所以 $P((\neg\alpha))$ 与 $P((\alpha \square \beta))$ 成立 Q.E.D.





括号引理

1.8.b 证明概要 [Enderton, pp.30]:

记“左括号比右括号多”为性质 P

1. Base: 对于原子公式 $\alpha = \mathbf{A}_i$ 显然有 $P(\alpha)$ 成立 (无真初始段)

2. Induction step:

- » Hypothesis: 设 $P(\alpha)$ 和 $P(\beta)$ 均成立
- » 以 \mathcal{E}_\wedge 为例, 列举其所有可能真初始段, 易知均满足性质 P
- » 类似可证明其他构造运算下的情况。Q.E.D.





唯一可读性



魯哀公問於孔子曰：「樂正夔一足，信乎？」

孔子曰：「昔者舜欲以樂傳教於天下，乃令重黎舉夔於草莽之中而進之，舜以為樂正。夔於是正六律，和五聲，以通八風，而天下大服。重黎又欲益求人，舜曰：『夫樂，天地之精也，得失之節也，故唯聖人為能和。樂之本也。夔能和之，以平天下。若夔者一而足矣。』故曰夔一足，非一足也。」
（《呂氏春秋》，慎行論·察傳）

能否省略 WFF 中的括号?



$$A_1 \vee A_2 \wedge A_3$$





公式的结构

定理 1.9:

wff恰好具有以下六种形式之一:

1. 原子公式, 即命题词 A_i
2. $(\neg\alpha)$
3. $(\alpha \wedge \beta)$
4. $(\alpha \vee \beta)$
5. $(\alpha \rightarrow \beta)$
6. $(\alpha \leftrightarrow \beta)$

不仅如此, 在情形 2~6 中, 公式 α 和 β 还有二元联词 \square 都是**唯一**的。



公式的结构

例1.10: 公式

$$\gamma = (((\neg \mathbf{A}_1) \vee (\mathbf{A}_8 \rightarrow \mathbf{A}_{10})) \rightarrow (\mathbf{A}_1 \wedge \mathbf{A}_{10}))$$

具有哪种形式?

1. $\gamma = \alpha$
2. $\gamma = (\neg \alpha)$
3. $\gamma = (\alpha \wedge \beta)$
4. $\gamma = (\alpha \vee \beta)$
5. $\gamma = (\alpha \rightarrow \beta)$
6. $\gamma = (\alpha \leftrightarrow \beta)$





算法 I.IO: 解析算法 [ENDERTON, PP.31]

- > 输入: 命题逻辑 wff φ
- > 输出: φ 的解析树 T



算法 I.IO: 解析算法 [ENDERTON, PP.31]

- > 输入: 命题逻辑 wff φ
- > 输出: φ 的解析树 T

I. 令 $T = \varphi$ (根节点)



算法 I.IO: 解析算法 [ENDERTON, PP.31]

> 输入: 命题逻辑 wff φ

> 输出: φ 的解析树 T

1. 令 $T = \varphi$ (根节点)
2. 若所有的叶子均为原子公式, 则终止并输出 T ; 否则对 T 进行遍历, 直到找到一个非原子公式的叶子节点 N , 跳转至 3



算法 I.IO: 解析算法 [ENDERTON, PP.31]

> 输入: 命题逻辑 wff φ

> 输出: φ 的解析树 T

1. 令 $T = \varphi$ (根节点)
2. 若所有的叶子均为原子公式, 则终止并输出 T ; 否则对 T 进行遍历, 直到找到一个非原子公式的叶子节点 N , 跳转至 3
3. 表达式 N 的第一个符号必须为 $($ 。若第二个符号是 \neg , 跳转至 5; 否则跳转至 4



算法 I.IO: 解析算法 [ENDERTON, PP.31]

> 输入: 命题逻辑 wff φ

> 输出: φ 的解析树 T

1. 令 $T = \varphi$ (根节点)
2. 若所有的叶子均为原子公式, 则终止并输出 T ; 否则对 T 进行遍历, 直到找到一个非原子公式的叶子节点 N , 跳转至 3
3. 表达式 N 的第一个符号必须为 $($ 。若第二个符号是 \neg , 跳转至 5; 否则跳转至 4
4. 从 $($ 开始, 从左至右扫描得到一个括号平衡的 α
 - » α 的下一个符号必须为 \wedge 、 \vee 、 \rightarrow 和 \leftrightarrow 之中的一个
 - » 符号序列的剩余段为 β , 它必须包含一个括号平衡的 β 与 $)$
 - » 在 N 上分别创建左右子树 α 与 β , 跳转至 2



算法 I.10: 解析算法 [ENDERTON, PP.31]

› 输入: 命题逻辑 wff φ

› 输出: φ 的解析树 T

1. 令 $T = \varphi$ (根节点)
2. 若所有的叶子均为原子公式, 则终止并输出 T ; 否则对 T 进行遍历, 直到找到一个非原子公式的叶子节点 N , 跳转至 3
3. 表达式 N 的第一个符号必须为 $($ 。若第二个符号是 \neg , 跳转至 5; 否则跳转至 4
4. 从 $($ 开始, 从左至右扫描得到一个括号平衡的 α
 - › α 的下一个符号必须为 \wedge 、 \vee 、 \rightarrow 和 \leftrightarrow 之中的一个
 - › 符号序列的剩余段为 β), 它必须包含一个括号平衡的 β 与 $)$
 - › 在 N 上分别创建左右子树 α 与 β , 跳转至 2
5. 表达式 N 的前两个符号为 $(\neg$
 - › 符号序列的剩余段为 β), 它必须包含一个括号平衡的 β 与 $)$
 - › 在 N 上分别创建一颗子树 β , 跳转至 2



唯一可读性定理 [ENDERTON, PP.40]

定理 I.II (Unique Readability Theorem) :

对于五种公式构造运算在 S (即所有 wff) 上的限制 $\mathcal{E}_{\neg}|_S$ 和 $\mathcal{E}_{\Box}|_S$:

1. 值域 (range) 互不相交
2. 是一一映射

这意味着 S 是由命题集合 $\{A_1, \dots\}$ 通过五种构造算子自由生成的。



唯一可读性定理 [ENDERTON, PP.40]

定理 I.II (Unique Readability Theorem) :

对于五种公式构造运算在 S (即所有 wff) 上的限制 $\mathcal{E}_{\neg}|_S$ 和 $\mathcal{E}_{\Box}|_S$:

1. 值域 (range) 互不相交
2. 是一一映射

这意味着 S 是由命题集合 $\{A_1, \dots\}$ 通过五种构造算子自由生成的。

I.II 证明概要 [Enderton, pp.40]:

I. 先证明算子是一一的:

» 例如: 若 $\mathcal{E}_{\wedge}(\alpha, \beta) = (\alpha \wedge \beta) = (\gamma \wedge \delta) = \mathcal{E}_{\wedge}(\gamma, \delta)$, 则必有 $\alpha = \gamma$ 且 $\beta = \delta$, 否则与定理 1.8. b 矛盾

2. 再证明值域不相交 (算子定义)

» 例如: 若 $\alpha \wedge \beta = \gamma \rightarrow \delta$, 则有 $\alpha = \gamma$ 、 $\beta = \delta$ 且 $\wedge = \rightarrow$,



WFF 的简写

括号太多，不方便书写和阅读





WFF 的简写

括号太多，不方便书写和阅读

- 通常省略最外层的括号

$$(((\neg \mathbf{A}_1) \vee (\mathbf{A}_2 \rightarrow \mathbf{A}_3)) \rightarrow (\mathbf{A}_1 \wedge \mathbf{A}_3))$$

可写成

$$((\neg \mathbf{A}_1) \vee (\mathbf{A}_2 \rightarrow \mathbf{A}_3)) \rightarrow (\mathbf{A}_1 \wedge \mathbf{A}_3)$$





WFF 的简写

括号太多，不方便书写和阅读

- 通常省略最外层的括号

$$(((\neg \mathbf{A}_1) \vee (\mathbf{A}_2 \rightarrow \mathbf{A}_3)) \rightarrow (\mathbf{A}_1 \wedge \mathbf{A}_3))$$

可写成

$$((\neg \mathbf{A}_1) \vee (\mathbf{A}_2 \rightarrow \mathbf{A}_3)) \rightarrow (\mathbf{A}_1 \wedge \mathbf{A}_3)$$

- 辅助符号引入方括号、大括号

$$((\neg \mathbf{A}_1) \vee (\mathbf{A}_2 \rightarrow \mathbf{A}_3)) \rightarrow (\mathbf{A}_1 \wedge \mathbf{A}_3)$$

可写成

$$[(\neg \mathbf{A}_1) \vee (\mathbf{A}_2 \rightarrow \mathbf{A}_3)] \rightarrow (\mathbf{A}_1 \wedge \mathbf{A}_3)$$





WFF 的简写

› 约定联结符号的优先级，例如

$$\langle \neg, \wedge, \vee, \rightarrow, \leftrightarrow \rangle$$

›

$$((\neg \mathbf{A}_1) \vee (\mathbf{A}_2 \rightarrow \mathbf{A}_3)) \rightarrow (\mathbf{A}_1 \wedge \mathbf{A}_3)$$

可写成

$$\neg \mathbf{A}_1 \vee (\mathbf{A}_2 \rightarrow \mathbf{A}_3) \rightarrow (\mathbf{A}_1 \wedge \mathbf{A}_3)$$

›

$$\mathbf{A}_1 \vee (((\neg \mathbf{A}_1) \wedge \mathbf{A}_3) \vee (\neg(\mathbf{A}_2 \wedge \mathbf{A}_3)))$$

可写成

$$\mathbf{A}_1 \vee (\neg \mathbf{A}_1 \wedge \mathbf{A}_3 \vee \neg(\mathbf{A}_2 \wedge \mathbf{A}_3))$$

