



1/40

华东师范大学

软硬件协同设计技术与应用教育部工程研究中心

URL—<http://faculty.ecnu.edu.cn/chenyixiang>

{yxchen,hywu}@sei.ecnu.edu.cn,

命题逻辑系统

陈仪香 吴恒洋

MoE Engineering Center for Software/Hardware Co-Design Technology and Application
Software Engineering Institute
East China Normal University(ECNU)
Shanghai, China

2018级研究生软件工程理论课程，2018年2月



为什么要学习逻辑



2/40

数理逻辑是一门以数学方法为基础，用符号系统的形式来研究思维结构及规律的学科。数理逻辑能力是计算机工作者所要具备的基本素质。数学思维和推理逻辑与计算机有着紧密相连的关系，在三者关系中，如果将计算机比喻成一座大厦，那么数学思维和推理逻辑就是建成这座大厦所必不可少的基础材料和重要支柱。

1. 数理逻辑在计算机硬件设计中的应用

计算机硬件的设计过程中，特别是面对解决选择的问题，通常用数理逻辑进行推理。开关电路——布尔代数，本质是逻辑。在搜索引擎（索引）上也有巨大作用

2. 数理逻辑为计算机程序语言提供思想和方法

(1) PROLOG，应用于人工智能，是基于逻辑的程序设计语言

(2) 子程序调用中的参数代换、赋值等都出自数理逻辑的方法

(3) 关系数据库——谓词逻辑



Back

Close

3. 数理逻辑是规范语言、定理证明器、模型检测器的基础

$$\mathcal{M} \models \phi$$

其中， \mathcal{M} -模型； ϕ -规范语言（用户需求），以逻辑的形式表示。

中心：设计好的算法计算 \models



3/40



Back

Close



学得东西都是有用的——以数学为例

1. Google 2002年推出了自己的“新闻”服务。和传统媒体的做法不同, 这些新闻不是记者写的, 也不是人工编辑的, 而是由计算机整理、分类和聚合各个新闻网站的内容, 一切都是自动生成的。这里面关键的技术就是新闻的自动分类——背后的数学**余弦定理**

2. Google的网页排名技术。其“PageRank”网页排名算法是革命性的发明—公认的文献检索中最大的贡献之一, 有人甚至认为整个公司的成功都是基于这个算法。创始人: 拉里·佩奇、谢尔盖·布林。佩奇也是因为这个算法在30岁时当选为美国工程院院士, 是继乔布斯, 盖茨之后有一位当院士的辍学生。其背后的数学——**线性代数**。

中国的大部分软件工程师在一个未知领域都是从直观感觉出发, 用“凑”的方法来解决问题, 说的不好听, 就是山寨。Google招揽理论基础优异的工程师。——吴军《数学之美》



Back

Close

命题逻辑系统的语法结构



5/40

- 如何构造命题逻辑公式？

定义 命题是陈述句

陈述句可以分为

- 简单句
- 复合句：简单句+连接词

陈述句有真假之分：陈述句表达的含义是真或假。

- 命题：简单命题，复合命题
- 真命题和假命题
- 命题变元：表示命题的变元，用字母 $p, q, \dots, p_1, p_2, \dots$ 表示。
- 复合命题：命题变元+连接符组合



Back

Close

命题逻辑公式



6/40

- 字母表

1. 命题变元: p_1, p_2, \dots , 用 p 表示
2. 连接符: $\wedge, \vee, \rightarrow, \neg$
3. 辅助符: $), ($

- 命题逻辑公式用符号 A 表示:

$$A ::= p \mid A \wedge A \mid A \vee A \mid \neg A \mid A \rightarrow A$$

定义 设 $S = \{p_1, p_2, \dots, p_n, \dots\}$ 是命题变元集, $F(S)$ 表示 S 上的命题公式集, 其元素用 A 表示, 其归纳定义如下:

1. $S \subseteq F(S)$, 即, 每个命题变元都是命题逻辑公式, 称为原子命题公式
2. 若 $A, B \in F(S)$, 则 $\neg A, A \wedge B, A \vee B, A \rightarrow B \in F(S)$ 。

注 $F(S)$ 是 S 生成的 $(\neg, \wedge, \vee, \rightarrow)$ 型的自由代数。



例子：下列都是命题公式：

$p_1 \rightarrow p_2, (p_1 \rightarrow p_2) \vee (p_2 \wedge \neg p_3), \neg p_1 \rightarrow p_2.$

但这些都不是命题逻辑公式： $p_1 \rightarrow, (p_2 \wedge) \vee p_3)$



Back

Close

命题逻辑系统的语义结构

命题有真假含义之分，即有真命题和假命题之分。用1(T)表示真值真，而0(F)表示真值假。



8/40



Back

Close

真值

- 原子命题公式的真值可以指定为真或假

定义 真值指派是指对命题变元指定真值，即 S 到 $\{0, 1\}$ 的一个映射.

例子：考虑命题变元 p_1, p_2, p_3, p_4 . 真值组 $(0, 1, 1, 0)$ 是 p_1, p_2, p_3, p_4 的一个真值指派。

- 命题逻辑的连接符构成了一个真值 $\{0, 1\}$ 上的一元或二元运算：

1. 一元运算 \neg :

$$\begin{array}{ccc} \neg : \{0, 1\} & \longrightarrow & \{0, 1\} \\ 0 & \mapsto & 1 \\ 1 & \mapsto & 0 \end{array}$$

2. 二元运算 \wedge :

$$\begin{array}{ccc} \wedge : \{0, 1\}^2 & \longrightarrow & \{0, 1\} \\ (0, 0) & \mapsto & 0 \\ (0, 1) & \mapsto & 0 \\ (1, 0) & \mapsto & 0 \\ (1, 1) & \mapsto & 1 \end{array}$$





3. 二元运算 \vee :

$$\begin{array}{lll} \vee : \{0, 1\}^2 & \longrightarrow & \{0, 1\} \\ (0, 0) & \mapsto & 0 \\ (0, 1) & \mapsto & 1 \\ (1, 0) & \mapsto & 1 \\ (1, 1) & \mapsto & 1 \end{array}$$

4. 二元运算 \rightarrow :

$$\begin{array}{lll} \rightarrow : \{0, 1\}^2 & \longrightarrow & \{0, 1\} \\ (0, 0) & \mapsto & 1 \\ (0, 1) & \mapsto & 1 \\ (1, 0) & \mapsto & 0 \\ (1, 1) & \mapsto & 1 \end{array}$$

- 由命题变元的真值指派以及连接符所确定的真值运算可得到命题逻辑公式的真值.

如: 给定命题变元 p_1, p_2, p_3, p_4 .



Back

Close

公 式	(0,1,1,0)	(1,1,0,1)	(1,1,1,1)
$(p_1 \rightarrow p_2) \vee p_3$	1	1	1
$\neg(p_1 \wedge p_3) \rightarrow p_4$	0	1	1
$p_1 \rightarrow p_1$			
$(p_5 \wedge p_2) \rightarrow p_1$	\times	\times	\times



11/40



Back

Close

赋值



12/40

定义 $F(S)$ 的一个赋值 v 是一个映射 $F(S) \rightarrow \{0, 1\}$, 并满足下面条件:

1. $v(\neg A) = \neg v(A)$
2. $v(A \vee B) = v(A) \vee v(B)$
3. $v(A \wedge B) = v(A) \wedge v(B)$
4. $v(A \rightarrow B) = v(A) \rightarrow v(B)$.

用符号 Ω 表示所有赋值集.

注1: 同态映射—

注2: 定义中等式中左边符号 $\wedge, \vee, \neg, \rightarrow$ 是命题逻辑连接符, 而右边符号 $\wedge, \vee, \neg, \rightarrow$ 是真值集 $\{0, 1\}$ 上的一元或二元运算。

注3: 这是归纳定义: 定义复杂公式的赋值是通过复杂公式中的简单公式的赋值经过运算得到。因此, 复杂公式的赋值完全命题变元的赋值确定。

如: 计算公式 $(p_1 \rightarrow p_2) \rightarrow p_3$ 的赋值: $v((p_1 \rightarrow p_2) \rightarrow p_3) = (v(p_1) \rightarrow v(p_2)) \rightarrow v(p_3)$.



Back

Close



注4: 假设 A 含有 n 个命题变元, 则 A 的赋值有 2^n 个。

定理 赋值可以诱导一个真值指派, 而真值指派也可以诱导一个赋值。

证明: 设 v 是 $F(S)$ 上的一个赋值, 即 v 是 $F(S)$ 到 $\{0, 1\}$ 的一个函数。由于命题变元集 S 是 $F(S)$ 的一个子集, 因此, v 在 S 上的限制 $v|_S$ 是命题变元的真值指派。反之, 设 V 是 S 的一个真值指派, 则 V 可以诱导一个 $F(S)$ 的赋值 v_V :

设 $A \in F(S)$, 若 A 是命题变元 p , 则定义 $v_V(p) = V(p)$. 若 A 是一个复合公式 $A_1 \vee A_2$, 则定义 $v_V(A) = v_V(A_1) \vee v_V(A_2)$. 类似地, 定义其他的三种复合公式 A 是 $A_1 \wedge A_2$, $A_1 \rightarrow A_2$ 以及 $\neg A_1$ 情形。这样定义的映射 v_V 是 $F(S)$ 的一个赋值。 \square

真度



14/40

定义 设逻辑公式 A 含有 n 个命题变元, 设 $T(A) = \{v \in \Omega \mid v(A) = 1\}$. 公式 A 的真度 $\tau(A)$ 定义为:

$$\tau(A) = \frac{|T(A)|}{2^n}.$$

例子: 计算 $\tau(p_1 \rightarrow p_2)$, $\tau(p_1 \vee p_2)$, $\tau(p_1 \wedge p_2)$, $\tau(\neg p)$.
 $\tau((p_1 \rightarrow p_2) \rightarrow p_1)$.

注: 说明 $F(S)$ 中每个公式的真度都是一个以2的方幂为分母的非负数。反过来, 是不是每个这种分数都是 $F(S)$ 中的某个公式呢? 回答是肯定的, 见下面定理。

定理 设 $H = \{\tau(A) \mid A \in F(S)\}$, 则

$$H = \left\{ \frac{k}{2^n} \mid k = 0, 1, \dots, 2^n; n = 1, 2, \dots \right\}.$$



Back

Close

重言式与矛盾式



15/40

定义 设 $A \in F(S)$, 若 $\tau(A) = 1$, 则称 A 为重言式 (永真式), 若 $\tau(A) = 0$ 则称 A 为矛盾式 (永假式). 用符号 $\models A$ 表示 A 是重言式.

例子: 永真式 $p \rightarrow p$, 矛盾式 $p \wedge \neg p$.

注: (1) 上述定义等价于: 设 $A \in F(S)$, 若对每个赋值 $v \in \Omega$ 都有 $v(A) = 1$, 则称 A 为重言式 (永真式), 若对每个赋值 $v \in \Omega$ 都有 $v(A) = 0$, 则称 A 为矛盾式 (永假式)。

(2) 在计算机科学中, 若 $\models A$, 则称 A 是有效的; 若存在一个赋值 $v \in \Omega$ 使得 $v(A) = 1$, 则称 A 是可满足的。显然有效蕴含可满足。

(3) A 是可满足的当且仅当 $\neg A$ 是无效的。

例子: $p \vee q \rightarrow p$ 的有效性和可满足性。



Back

Close

逻辑等价、真值等价



16/40

定义 设 $A, B \in F(S)$, 若对于 $v \in \Omega$ 都有 $v(A) = v(B)$, 则称 A 与 B 是逻辑等价的, 记作 $A = B$.

例子: $A \vee B = \neg A \rightarrow B$, $A \rightarrow B = \neg(A \wedge \neg B)$.

定理 $\{\neg, \rightarrow\}, \{\neg, \vee\}, \{\neg, \wedge\}$ 是连接符集 $\{\neg, \vee, \wedge, \rightarrow\}$ 的充足集。

定理 性质:

交换律

$$A \vee B = B \vee A$$

$$A \wedge B = B \wedge A$$

结合律

$$A \vee (B \vee C) = (A \vee B) \vee C$$

$$A \wedge (B \wedge C) = (A \wedge B) \wedge C$$

分配律

$$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$$

$$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$$

de Morgan律

$$\neg(A \vee B) = \neg A \wedge \neg B$$

$$\neg(A \wedge B) = \neg A \vee \neg B$$

吸收律

$$A \vee (A \wedge B) = A$$

$$A \wedge (A \vee B) = A$$

幂等律

$$A \vee A = A$$

$$A \wedge A = A$$

对合律

$$\neg\neg A = A$$



Back

Close



定义

- (标准) 析取范式: $(Q_{11} \wedge Q_{12} \wedge \cdots Q_{1n}) \vee \cdots \vee (Q_{m1} \wedge Q_{m2} \wedge \cdots \wedge Q_{mn})$,
- (标准) 合取范式: $(Q_{11} \vee Q_{12} \vee \cdots Q_{1n}) \wedge \cdots \wedge (Q_{m1} \vee Q_{m2} \vee \cdots \vee Q_{mn})$,

其中 Q_{ij} 或是命题变元, 或是命题变元的非。

定理 任何非矛盾式的公式都逻辑等价于一个析取范式,
任何非重言式的公式都逻辑等价于一个合取范式.

注: 限制条件实际都可以去掉。因为矛盾式逻辑等价于一个析取范式 $\neg p \wedge p$; 重言式逻辑等价于一个合取范式 $\neg p \vee p$.

例子

$$(p \rightarrow \neg q) \rightarrow (q \vee \neg p)$$

的合取范式和吸取范式。



Back

Close



p	q	$\neg p$	$\neg q$	$p \rightarrow \neg q$	$q \vee \neg p$	$(p \rightarrow \neg q) \rightarrow (q \vee \neg p)$
1	1	0	0	0	1	1
1	0	0	1	1	0	0
0	1	1	0	1	1	1
0	0	1	1	1	1	1

合取范式: $\neg p \vee q$

析取范式: $(p \wedge q) \vee (\neg p \wedge q) \vee (\neg p \wedge \neg q)$

(1) 合取范式考虑计算结果为FALSE, 析取范式考虑计算结果TRUE.

(2) 合取范式作用是方便有效性检查: $\models \varphi$. 其中: $\varphi = (Q_{11} \vee Q_{12} \vee \cdots Q_{1n}) \wedge \cdots \wedge (Q_{m1} \vee Q_{m2} \vee \cdots \vee Q_{mn})$, Q_{ij} 或是命题变元, 或是命题变元的非。等价于 $\models Q_{i1} \vee Q_{m2} \vee \cdots \vee Q_{in} \ (i = 1, \cdots, m)$ 。

(3) $\models Q_{i1} \vee Q_{m2} \vee \cdots \vee Q_{in}$ 是有效的当且仅当存在 $1 \leq k, j \leq n$ 使得 Q_{ik} 是 $\neg Q_{ij}$.





1. 证明下列公式是重言式:

(a) $A \rightarrow (\neg A \rightarrow B)$

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

(c) $(\neg A \rightarrow \neg B) \rightarrow (A \rightarrow B)$

2. 证明下列各条成立

(a) $(A \vee B) \rightarrow C = (A \rightarrow C) \wedge (B \rightarrow C)$

(b) $(A \wedge B) \rightarrow C = (A \rightarrow C) \vee (B \rightarrow C)$

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

3. 求公式 $(\neg p_1 \rightarrow p_2) \rightarrow p_3$ 的析取范式和合取范式。

4. 计算下列逻辑公式的真度

(a) $(p_1 \vee p_2) \rightarrow p_3$

(b) $(p_1 \rightarrow p_2) \vee (p_3 \rightarrow p_4)$

(c) $(\neg p_1 \rightarrow p_2) \rightarrow p_3$



Back

Close



5. 设 $A \downarrow B$ 表示 $\neg(A \vee B)$, 证明连接符 $\{\downarrow\}$ 是命题逻辑连接符集的充足集.
6. 研究题: 任给分数 $\frac{n}{m}$ ($0 \leq n \leq m$), 是否存在一个命题公式 A 使得 A 的真度 $\tau(A) = \frac{n}{m}$?



命题逻辑系统的推理机制

建立命题逻辑系统的推理机制，由公理和推理规则组成，回答一个公式是否可以其它公式推出，即是否其它公式的结论？



21/40



Back

Close

形式系统L



22/40

定义 命题逻辑形式系统L: 命题逻辑公式集 $F(S)$ +三条公理L1,L2,L3+一条推理规则MP:

- 三条公理:

$$L1: A \rightarrow (B \rightarrow A)$$

$$L2: (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$$

$$L3: (\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$$

- 一条推理规则: Modus Ponens 规则 (MP规则), 分离规则

MP: 从 $A \rightarrow B$ 与 A 可以得到 B .

$$\frac{A, A \rightarrow B}{B}$$

注: L1, L2, L3是公理模式, 由此可以得到无穷多个公理, 如:
 $(p \rightarrow (q \rightarrow p))$, $(p \rightarrow (p \rightarrow p))$, $(p \rightarrow ((q \vee r) \rightarrow p))$ 都是公理。



Back

Close

L中的证明与定理



23/40

定义 L 中的一个证明是一有限公式序列 A_1, A_2, \dots, A_n , 其中每个 A_i 或是公理, 或存在 $j, k < i$ 使得 A_i 是 A_j 和 A_k 使用 MP 规则推导的结果。其中最后一项公式 A_n 称为 L 中的一个定理, 记作 $\vdash_L A_n$ 或 $\vdash A_n$. n 叫做证明的长度。

注: 定理可当公理使用与证明中:

若有限公式序列 A_1, A_2, \dots, A_n 中每个 A_i 或是公理, 或是定理, 或存在 $j, k < i$ 使得 A_i 是 A_j 和 A_k 使用 MP 规则推导的结果, 则 A_1, A_2, \dots, A_n 是一证明。

例子: 证明:

1. $\vdash (p_1 \rightarrow p_2) \rightarrow (p_1 \rightarrow p_1)$
2. $\vdash (A \rightarrow A)$



Back

Close

推演



24/40

定义 设 $\Gamma \subset F(S)$, 从 Γ 出发的 L 中一个推演是一有限公式序列 A_1, A_2, \dots, A_n , 其中每个 A_i 或是公理, 或是 Γ 中成员, 或存在 $j, k < i$ 使得 A_i 是 A_j 和 A_k 使用MP规则推导的结果。其中最后一项公式 A_n 称为 Γ -结论, 记作 $\Gamma \vdash_L A_n$ 或 $\Gamma \vdash A_n$. n 叫做推演的长度。

注: 当 $\Gamma = \emptyset$ 时, Γ -结论就是定理, 即 $\emptyset \vdash A$ 就是 $\vdash A$.

例子: 证明: $\{A, B \rightarrow (A \rightarrow C)\} \vdash B \rightarrow C$

定理 (演绎定理)

设 $\Gamma \subset F(S), A, B \in F(S)$. 则 $\Gamma \cup \{A\} \vdash B$ 当且仅当 $\Gamma \vdash A \rightarrow B$.

证明: " \Rightarrow " 设 $\Gamma \cup \{A\} \vdash B$, 则有一个从 $\Gamma \cup \{A\}$ 得到 B 的推演。

- (1) A_1
- (2) A_2
- \vdots
- (n) A_n

其中 A_n 就是 B .

对推演长度 n 使用数学归纳法, 构造 $\Gamma \vdash A \rightarrow B$ 的推演.



Back

Close



1. 奠基步: $n = 1$ 此时, B 或是公理, 或是 Γ 中成员, 或是 A 。

(a) B 是公理,

- | | | |
|-----|-----------------------------------|------------|
| (1) | B | 公理 |
| (2) | $B \rightarrow (A \rightarrow B)$ | $L1$ |
| (3) | $A \rightarrow B$ | $MP(1, 2)$ |

所以 $\vdash A \rightarrow B$, 进而 $\Gamma \vdash A \rightarrow B$.

(b) $B \in \Gamma$

- | | | |
|-----|-----------------------------------|------------|
| (1) | B | Γ |
| (2) | $B \rightarrow (A \rightarrow B)$ | $L1$ |
| (3) | $A \rightarrow B$ | $MP(1, 2)$ |

所以 $\Gamma \vdash A \rightarrow B$.

(c) B 就是 A .

因为 $\vdash A \rightarrow A$, 所以 $\vdash A \rightarrow B$.

2. 归纳步: 假定当推演长度小于 n 是结论成立, 即当 $\Gamma \cup \{A\} \vdash B$ 的推演长度小于 n 时, 都有 $\Gamma \vdash A \rightarrow B$ 成立。现假定 $\Gamma \cup \{A\} \vdash B$ 的推演长度等于 n . 从而有 $\Gamma \cup \{A\} \vdash B$ 的推演:



Back

Close



- (1) A_1
- (2) A_2
- \vdots
- (n) A_n (就是 B)

当 B 是公理, 或是 Γ 中成员, 或是 A 时, 从前面的证明可以得到 $\Gamma \vdash A \rightarrow B$. 因此, 我们直接假定 B 是两项 C 与 $C \rightarrow B$ 使用MP规则得到,即存在 $i, j < n$ 使得 A_i 是 C , 而 A_j 是 $C \rightarrow B$, 如下所示:

- (1) A_1
- (2) A_2
- \vdots
- (i) $A_i(C)$
- \vdots
- (j) $A_j(C \rightarrow B)$
- \vdots
- (n) $A_n(B)$

由此有 $\Gamma \cup \{A\} \vdash C$ 以及 $\Gamma \cup \{A\} \vdash C \rightarrow B$, 它们的推演长度分别是 i 和 j , 都小于 n 。故而, 由归纳假设得到 $\Gamma \vdash A \rightarrow C$





与 $\Gamma \vdash A \rightarrow (C \rightarrow B)$ 都成立。因而, 可得到推演:

$$\begin{array}{ll}
 (1) & \\
 \vdots & \\
 (k) & A \rightarrow C \\
 (k+1) & \\
 \vdots & \\
 (l) & A \rightarrow (C \rightarrow B) \\
 (l+1) & (A \rightarrow (C \rightarrow B)) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow B)) \quad L2 \\
 (l+2) & (A \rightarrow C) \rightarrow (A \rightarrow B) \quad MP(l, l+1) \\
 (l+3) & A \rightarrow B \quad MP(k, l+2)
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} (\Gamma \vdash A \rightarrow C) \\ \\ \\ (\Gamma \vdash A \rightarrow (C \rightarrow B)) \end{array}$$

所以, 我们得到了 $\Gamma \vdash A \rightarrow B$.

” \Leftarrow ” 设 $\Gamma \vdash A \rightarrow B$, 则存在一推演:

$$\begin{array}{ll}
 (1) & A_1 \\
 (2) & A_2 \\
 \vdots & \vdots \\
 (n) & A_n(A \rightarrow B)
 \end{array}$$

其中, 每个 A_i 或是公理, 或是 Γ 中成员, 或存在 $k, j < i$ 使得 A_i 是 A_j 和 A_k 使用MP规则得到。



在这个推演基础上构造一个序列如下：

$$\begin{array}{lll} (1) & A_1 & \\ (2) & A_2 & \\ \vdots & \vdots & \\ (n) & A_n(A \rightarrow B) & \\ (n+1) & A & (\text{假设}) \\ (n+2) & B & MP(n, n+1) \end{array}$$

这个序列是 $\Gamma \cup \{A\}$ 到 B 的一个推演,故有, $\Gamma \cup \{A\} \vdash B$. □

注：使用演绎定理可以简化证明。

例子：证明： $\vdash A \rightarrow A$.

例子：证明： $\vdash \neg A \rightarrow (A \rightarrow B)$.



Back

Close

HS规则

定理 (三段论规则-HS规则)

$\{A \rightarrow B, B \rightarrow C\} \vdash A \rightarrow C.$

证明：构造推演序列如下：

- (1) $A \rightarrow B$ Γ
- (2) $B \rightarrow C$ Γ
- (3) A 假设
- (4) B $MP(1, 3)$
- (5) C $MP(2, 4)$

所以有, $\{A, A \rightarrow B, B \rightarrow C\} \vdash C$. 这样, 由演绎定理得到, $\{A \rightarrow B, B \rightarrow C\} \vdash A \rightarrow C.$ \square

推论 设 $\vdash A \rightarrow B$ 且 $\vdash B \rightarrow C$, 则 $\vdash A \rightarrow C$.



可证等价关系



30/40

定义 设 $A, B \in F(S)$, 若 $\vdash A \rightarrow B$ 且 $\vdash B \rightarrow A$ 成立, 则称 A 与 B 可证等价, 记作 $A \approx B$.

例子: 设 $A \in F(S)$, 则 $\neg\neg A \approx A$.

例子: 设 $A, B \in F(S)$, 则 $(A \rightarrow B) \approx (\neg B \rightarrow \neg A)$.

证明: 由L3得到 $\vdash (\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$. 现在需要证明 $\vdash (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$. 由演绎定理只须证明 $\{A \rightarrow B\} \vdash (\neg B \rightarrow \neg A)$.

定理 可证等价关系 \approx 是 $F(S)$ 上的同余关系.

证明:

1. \approx 是 $F(S)$ 上的等价关系.
2. \approx 对 $F(S)$ 的逻辑运算非 \neg 是同余的, 即若 $A \approx B$ 则 $\neg A \approx \neg B$.
3. \approx 对 $F(S)$ 的逻辑运算 \rightarrow 是同余的, 即若 $A \approx B$ 且 $C \approx D$, 则 $(A \rightarrow C) \approx (B \rightarrow D)$.

□

子式替换定理

定理 设 A 中含有子式 A_1 , 且 $A_1 \approx B_1$, 若 A 中的一处或多处出现的 A_1 换成 B_1 所得的公式记为 B , 则有 $A \approx B$.



31/40



Back

Close



(1). 试证:

1. $\vdash A \rightarrow (B \rightarrow (A \rightarrow B)).$
2. $\vdash (B \rightarrow C) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
3. $\vdash (A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B).$

(2) 试证:

1. $(A \rightarrow (B \rightarrow C)) \approx (B \rightarrow (A \rightarrow C))$
2. $(A \rightarrow (A \rightarrow B)) \approx (A \rightarrow B).$



Back

Close

命题逻辑系统的可靠性与完备性



33/40

给定命题变元集 S ，使用逻辑连接词 $\neg, \vee, \wedge, \rightarrow$ 构造 S 上的命题逻辑公式集 $F(S)$ ，它是 S 上的 $(\neg, \vee, \wedge, \rightarrow)$ 型的自由代数，在赋值语义考虑下有逻辑公式真度以及重言式概念，符号 $\models A$ 表明命题逻辑公式 A 是重言式，而从逻辑推理考虑下有证明和定理概念，符号 $\vdash A$ 表明 A 是一个定理。本段解决 $\vdash A$ 与 $\models A$ 的关系：

$$\forall A \in F(S), \models A \text{ 当且仅当 } \vdash A.$$

(1) 反映了“可证”和“真”之间的关系。通常“真”大于“可证”，但在命题逻辑中二者一致。

(2) 希尔伯特的第二问题：算术公理系统的无矛盾性。希尔伯特提出用形式主义计划的证明论方法加以证明。

(3) 哥德尔不完全性定理：“20世纪最有意义的数学真理”

第一定理：任意一个包含一阶谓词逻辑与初等数论的形式系统，都存在一个命题，它在这个系统中既不能被证明为真，也不能被证明为否。说明了“真”大于“可证”。

第二定理：如果系统 S 含有初等数论，当 S 无矛盾时，它的无矛盾性不可能在 S 内证明。



Back

Close

命题逻辑L的可靠性



34/40

定理 (可靠性定理) 命题逻辑系统L中的定理都是重言式, 即

若 $\vdash A$ 则 $\models A$.

证明: 依赖于下面两个引理.

(1) L 中的公理都是重言式。

(2) MP规则保重言式, 即设 $A, B \in \mathcal{F}(S)$. 如果 $A \rightarrow B \in Tau$ 且 $A \in Tau$, 则 $B \in Tau$.

□



Back

Close

命题逻辑L的完备性



35/40

命题逻辑系统L的完备性证明需要一些准备工作.

引理 $\vdash (\neg A \rightarrow A) \rightarrow A$.

定义 (扩张)

设 L^* 是系统 L 的基础上改变 L 的公理或给 L 添加新的公理而得到的系统, 若 L 中的定理都是 L^* 的定理, 则称 L^* 是 L 的扩张.

例子: 在 L 中增加新公理 $A \rightarrow A$ 得到 L 的一个扩张 L^* . 但 L 的定理与 L^* 的定理相同.

例子: 在 L 中增加新公理 $A \rightarrow \neg A$ 得到 L 的一个扩张 L^* , L^* 是 L 的真扩张, 因为 $A \rightarrow \neg A$ 不是 L 的定理, 但是 L^* 的定理.

例子: 设 $A \in F(S)$, 若 A 不是 L 中的定理, 增加公理 $\neg A$ 则得到 L 的扩张 L^* .

注: L 是其自身的扩张.



Back

Close

相容性



36/40

定义 L 的扩张 L^* 称为相容的, 若 $F(S)$ 中不存在公式 A 使得 A 与 $\neg A$ 都是 L^* 的定理。

例子: L 是相容的。

引理 L 的扩张 L^* 是相容的当且仅当 $F(S)$ 中有一公式不是 L^* 中的定理。

证明: \Rightarrow 设 L^* 是相容的, 若再设 $F(S)$ 中的任何公式都是 L^* 的定理, 则对于任何的公式 A 都有 A 与 $\neg A$ 均是 L^* 的定理, 与 L^* 的相容性矛盾。

\Leftarrow 设 L^* 是不相容的, 则存在公式 $A \in F(S)$ 使得 A 和 $\neg A$ 均是 L^* 中的定理。下面证明 $F(S)$ 中的任何公式 B 都是 L^* 的定理。



Back

Close

完全性



37/40

定义 系统 L 的扩张 S 称为完全的, 若对 $F(S)$ 中的每个公式 A 都有 $\vdash_S A$ 或 $\vdash_S \neg A$ 成立.

注: 系统 L 是不完全的, 因为 $\vdash_L p$ 与 $\vdash_L \neg p$ 均不成立.

引理 相容系统 L^* 都有一个相容的完全扩张 S .

证明: 设相容系统 L^* 是不完全的, 则有 $A \in F(S)$ 使得 $\vdash_{L^*} A$ 不成立. 以 $\neg A$ 作为新公理加到 L^* 中得到 L^* 的一个扩张 L^{**} , 则 L^{**} 是相容的. 若 L^{**} 是不完全的, 则存在公式 $B \in F(S)$ 使得 $\vdash_{L^{**}} B$ 不成立. 以 $\neg B$ 作为新公理加到 L^{**} 中得到 L^{**} 的一个扩张 L^{***} . 这个过程可以一直下去. 一般地叙述如下.

由于 $F(S)$ 是可数集, 可设为 $A_0, A_1, \dots, A_n, \dots$

现在构造 L^* 的扩张序列 $J_0, J_1, \dots, J_n, \dots$

令 $J_0 = L^*$

若 $\vdash_{J_0} A_0$, 则令 $J_1 = J_0$,

若 $\vdash_{J_0} A_0$ 不成立, 则增加 $\neg A_0$ 为新公理从 J_0 得到 J_1 , 则 J_1 是 J_0 的相容扩张. 一般地, 对于 $n \geq 1$, 由 J_{n-1} 构造 J_n :



Back

Close



$$J_n = \begin{cases} J_{n-1} & \text{若 } \vdash_{J_{n-1}} A_{n-1} \text{ 成立} \\ J_{n-1} + \{\neg A_{n-1}\} & \text{若 } \vdash_{J_{n-1}} A_{n-1} \text{ 不成立} \end{cases}$$

其中 $J_{n-1} + \{\neg A_{n-1}\}$ 表示 J_{n-1} 增加新公理 $\neg A_{n-1}$ 所得到的 J_{n-1} 的公理.

定义 L^* 的扩张 S : S 的公理集是 J_n ($n = 0, 1, 2, \dots$) 的公理集之并, 则 S 是 L^* 的相容完全扩张. \square



L的完备性定理

定理 命题逻辑系统L中的重言式都是定理，即若 $\models A$ 则 $\vdash A$.

证明：假设 A 不是L中的定理，则把 $\neg A$ 添加为新公理得到相容系统 L^* 。取 L^* 的相容完全扩张 S ，定义映射 $v: F(S) \rightarrow \{0, 1\}$ 为 $v(B) = 1$ 当且仅当 $\vdash_S B$ ($B \in F(S)$)。则 v 是 $F(S)$ 上的一个赋值。由 v 的定义可以得到 $v(\neg A) = 1$ 与 A 是重言式矛盾。所以 A 是L的定理。

注：重言式是从形式系统的外部来描述公式的特点。而定理是从形式系统的内部来描述公式的特点。因而，可靠性定理说明了形式系统内部描述，从外部来看是合理的，是可靠的。完备性定理反映了外部的描述是恰当的，没有多余的信息。



作业



40/40

1. 利用L的完备性定理证明以下各式成立:

$$(a) \vdash (\neg A \rightarrow A) \rightarrow A$$

$$(b) \vdash \neg(A \rightarrow B) \rightarrow (B \rightarrow A)$$

$$(c) ((A \vee B) \rightarrow C) \approx (A \rightarrow C) \wedge (B \rightarrow C)$$

$$(d) ((A \wedge B) \rightarrow C) \approx (A \rightarrow C) \vee (B \rightarrow C)$$

2. 设 $\Gamma \subseteq F(S)$, Γ 是有限集, $A \in F(S)$. 证明:

$$\Gamma \vdash A \text{ 当且仅当 } \Gamma \models A.$$

其中 $\Gamma \models A$ 定义为对于任何赋值 v 若对于 Γ 中的每个成员 B 只要 $v(B) = 1$ 就有 $v(A) = 1$.



Back

Close