

《软件分析与验证》

命题逻辑



贺飞

清华大学软件学院

2023年3月16日

1. 语法

2. 语义

3. 证明系统

语法

命题逻辑的符号集 (*alphabet*) 包括:

- 真值符号 \top (代表 *true*) 和 \perp (代表 *false*);
- 命题变元;
- 逻辑联结词 \neg , \wedge , \vee , \rightarrow 和 \leftrightarrow 。

其中 \top, \perp 和命题变元统称为原子命题。

定义 (合式公式)

命题逻辑的合式公式 (*well-formed formula*, 简称公式) 递归定义如下:

- 原子命题是合式公式;
- 如果 F 是合式公式, 那么 $\neg F$ 也是合式公式;
- 如果 F, G 是合式公式, 那么 $F \wedge G, F \vee G, F \rightarrow G$ 和 $F \leftrightarrow G$ 也是合式公式。

其中

- 原子命题也称原子公式 (*atom*)
- 原子公式和原子公式的非称为文字 (*literal*)

一些约定:

- 以小写字母 p, q, r 表示命题变元
- 以大写字母 F, G 表示命题逻辑公式
- 逻辑联结词的优先级: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- \wedge 和 \vee 是左结合的, \rightarrow 和 \leftrightarrow 是右结合的

可将公式转换成只包含 \perp, \neg, \wedge 及命题变元:

$$\top := \neg \perp$$

$$F_1 \vee F_2 := \neg(\neg F_1 \wedge \neg F_2)$$

$$F_1 \rightarrow F_2 := \neg F_1 \vee F_2$$

$$F_1 \leftrightarrow F_2 := (F_1 \rightarrow F_2) \wedge (F_2 \rightarrow F_1)$$

语义

变元赋值 (*assignment*, 简称赋值, 又称真值指派) 是从变元集到真值集 $\{true, false\}$ 的函数。设 ρ 为赋值, $\rho(p)$ 返回变元 p 在 ρ 中被指派的值。

定义 (命题逻辑公式的语义)

公式 F 在赋值 ρ 下的取值 (evaluation) $\llbracket F \rrbracket_\rho$ 递归定义如下:

$$\llbracket \perp \rrbracket_\rho = false$$

$$\llbracket p \rrbracket_\rho = \rho(p), \text{ 其中 } p \text{ 是 } F \text{ 中的一个变元}$$

$$\llbracket \neg F \rrbracket_\rho = \begin{cases} true, & \text{如果 } \llbracket F \rrbracket_\rho = false \\ false, & \text{如果 } \llbracket F \rrbracket_\rho = true \end{cases}$$

$$\llbracket F \wedge G \rrbracket_\rho = \begin{cases} true, & \text{如果 } \llbracket F \rrbracket_\rho = true \text{ 且 } \llbracket G \rrbracket_\rho = true \\ false, & \text{否则} \end{cases}$$

例

请确定公式 $F: p \wedge q \rightarrow p \vee \neg q$ 在赋值

$$\rho = \{p \mapsto true, q \mapsto false\}$$

下的取值。

解

因为 $\llbracket p \rrbracket_\rho = true$ 且 $\llbracket q \rrbracket_\rho = false$, 故 $\llbracket p \wedge q \rrbracket_\rho = false$ 。

又有 $\llbracket p \vee \neg q \rrbracket_\rho = \llbracket \neg(\neg p \wedge \neg\neg q) \rrbracket_\rho = true$, 故

$$\llbracket F \rrbracket_\rho = \llbracket \neg(p \wedge q) \vee (p \vee \neg q) \rrbracket_\rho = \llbracket \neg(\neg\neg(p \wedge q) \wedge \neg(p \vee \neg q)) \rrbracket_\rho = true$$

一个命题逻辑公式 F 是

- **可满足的** (*satisfiable*), 当且仅当存在一个赋值 ρ 使得 $\llbracket F \rrbracket_\rho$ 为真;
- **有效的** (或**永真的**) (*valid*), 当且仅当对任意赋值 ρ , $\llbracket F \rrbracket_\rho$ 都为真;
- **不可满足的** (或**永假的**) (*unsatisfiable*), 当且仅当对任意赋值 ρ , $\llbracket F \rrbracket_\rho$ 都为假。

F 是永真式常常记作 $\models F$ 。

例

- p 是可满足式
- $p \vee \neg p$ 是有效式
- $p \wedge \neg p$ 是不可满足式

例

证明公式 $p \wedge q \rightarrow p \vee \neg q$ 是永真式。

证明.

p	q	$\neg q$	$p \wedge q$	$p \vee \neg q$	原式
1	1	0	1	1	1
1	0	1	0	1	1
0	1	0	0	0	1
0	0	1	0	1	1

表: 真值表

真值表最后一列全为 1, 说明原公式在任意赋值下都为真, 因此为永真式。证毕。 □

定理

F 是永真式当且仅当 $\neg F$ 是永假式。

证明.

F 是永真式,

当且仅当 $\llbracket F \rrbracket_\rho$ 对任何赋值 ρ 都为真 (根据永真式定义),

当且仅当 $\llbracket \neg F \rrbracket_\rho$ 对任何赋值 ρ 都为假 (根据 \neg 的语义),

当且仅当 $\neg F$ 是永假式 (根据永假式定义)。



定义 (语义蕴涵)

给定两个公式 F 和 G , 如果对任意赋值 ρ , 只要 $\llbracket F \rrbracket_\rho$ 为真, $\llbracket G \rrbracket_\rho$ 就必为真, 就称 F 语义蕴涵 (*implies*) G , 或称 G 是 F 的有效推论 (*consequence*), 记为 $F \Rightarrow G$ 。

例如: $p \vee \neg q$ 是 $p \wedge q$ 的有效推论; \top 是 \perp 的有效推论。

证明 $F \Rightarrow G$ 的方法:

- 证明 $F \rightarrow G$ 是永真式 (比如真值表法)
- 基于一个演绎系统进行推理 (即将讨论)
- 证明 $\neg(F \rightarrow G)$ 是不可满足式 (可借助 SAT/SMT 求解器进行)

定义 (语义等价)

给定两个公式 F 和 G , 如果 $F \Rightarrow G$ 且 $G \Rightarrow F$, 就称 F 和 G **语义等价** (*semantically equivalent*), 记作 $F \Leftrightarrow G$ 。

例如: $p \wedge (q \vee r)$ 与 $(p \wedge q) \vee (p \wedge r)$ 语义等价。

F 与 G 语义等价的充要条件是在任意赋值 ρ 下它们的取值都相同, 即 $\llbracket F \rrbracket_{\rho} = \llbracket G \rrbracket_{\rho}$ 。

证明系统

常见的命题逻辑证明系统有¹:

- 公理系统
- 自然演绎系统
- 相继式演算系统

下面介绍命题逻辑的相继式演算系统 \mathcal{S}_{PL}

¹https://en.wikipedia.org/wiki/Proof_calculus

定义 (相继式)

一个相继式 (sequent) 是形如

$$F_1, \dots, F_m \vdash G_1, \dots, G_n$$

的式子, 其中 F_1, \dots, F_m 称前件, G_1, \dots, G_n 称后件。

例如: $p, q \vdash p \vee \neg q$ 是相继式, 其中 \vdash 左边的 p 和 q 是该相继式的前件, \vdash 右边的 $p \vee \neg q$ 是该相继式的后件。

从逻辑推理的角度看, 相继式前件的合取可看作假设, 后件的析取是我们希望证明的推论, 因此相继式的语义等价于

$$F_1 \wedge \dots \wedge F_m \rightarrow G_1 \vee \dots \vee G_n$$

一个相继式是有效的当且仅当上式在命题逻辑中是有效式。

以 Γ 和 Δ 表示公式集合，**推理规则** (*inference rule*) 的一般形式为：

$$\text{(规则名)} \frac{\Gamma_1 \vdash \Delta_1 \quad \dots \quad \Gamma_n \vdash \Delta_n}{\Gamma_{n+1} \vdash \Delta_{n+1}} \text{ 条件}$$

- 每条规则有若干个前提和一个结论。称有 0 个前提的规则为**公理**。
- 由下而上的考察推理规则——从结论出发，确认需要证明的前提有哪些。
- 需要强调的是，推理过程只涉及语法，不涉及公式的具体语义（语义正确性由推理系统的可靠性保证，后面讨论）。

从结论到前提，每条规则减少一个逻辑联结词。

$$\text{(左合取)} \quad \frac{\Gamma, P, Q \vdash \Delta}{\Gamma, P \wedge Q \vdash \Delta}$$

$$\text{(左析取)} \quad \frac{\Gamma, P \vdash \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \vee Q \vdash \Delta}$$

$$\text{(左否定)} \quad \frac{\Gamma \vdash P, \Delta}{\Gamma, \neg P \vdash \Delta}$$

$$\text{(左蕴涵)} \quad \frac{\Gamma \vdash P, \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \rightarrow Q \vdash \Delta}$$

从结论到前提，每条规则减少一个逻辑联结词。

$$\text{(右合取)} \quad \frac{\Gamma \vdash P, \Delta \quad \Gamma \vdash Q, \Delta}{\Gamma \vdash P \wedge Q, \Delta}$$

$$\text{(右析取)} \quad \frac{\Gamma \vdash P, Q, \Delta}{\Gamma \vdash P \vee Q, \Delta}$$

$$\text{(右否定)} \quad \frac{\Gamma, P \vdash \Delta}{\Gamma \vdash \neg P, \Delta}$$

$$\text{(右蕴涵)} \quad \frac{\Gamma, P \vdash Q, \Delta}{\Gamma \vdash P \rightarrow Q, \Delta}$$

$$\text{(包含)} \frac{}{\Gamma, P \vdash P, \Delta}$$

$$\text{(切)} \frac{\Gamma \vdash C, \Delta \quad \Gamma, C \vdash \Delta}{\Gamma \vdash \Delta}$$

- 其中包含规则是一条没有前提的公理，我们使用它来终结一个推导过程（或推导过程的一个分支）。
- 甘岑（G. Gentzen）证明了切规则是一条冗余规则，即所有使用了切规则的证明，都可以用一个不使用切规则的证明来替代。虽然切规则是一条冗余规则，加上切规则会大大减少证明的步骤，简化证明过程。

定义 (推导树)

推导树 (*derivation tree*) 是一棵满足下列条件的树:

- 每个节点对应一个相继式, 树叶对应空;
- 每个中间节点是某规则的结论, 而它的子节点是该规则的前提。

定义 (可推导)

如果存在一棵以相继式 F 为根节点的推导树, 就说 F **可推导**。

推导树构造过程:

1. 从根节点 F 出发, 反复应用推理规则, 扩展相应的边和节点, 直至当前树的叶子节点都不含联结词。
2. 如果不含联结词的叶子节点都能被公理所推导, 则 F 是有效式, 否则 F 不是有效式。

例

证明 $\vdash p \wedge q \rightarrow p \vee \neg q$ 。

证明.

$$\begin{array}{c} \text{右析取} \frac{\text{包含} \frac{\overline{p, q \vdash p, \neg q}}{p, q \vdash p \vee \neg q}}{p \wedge q \vdash p \vee \neg q}}{\vdash p \wedge q \rightarrow p \vee \neg q} \\ \text{左合取} \\ \text{右蕴涵} \end{array}$$

□

例

证明 $\vdash (q \rightarrow r) \rightarrow (p \vee q \rightarrow p \vee r)$ 。

证明.

$$\frac{\frac{q \rightarrow r, p \vdash p, r}{q \rightarrow r, p \vee q \vdash p, r} \quad \frac{\frac{q \vdash q, p, r}{q \rightarrow r, q \vdash p, r} \quad \frac{r, q \vdash p, r}{q \rightarrow r, q \vdash p, r}}{q \rightarrow r, p \vee q \vdash p, r}}$$

\Downarrow

$$\frac{\frac{\text{右蕴涵} \quad \frac{\text{右析取} \quad \frac{q \rightarrow r, p \vee q \vdash p, r}{q \rightarrow r, p \vee q \vdash p \vee r}}{q \rightarrow r \vdash p \vee q \rightarrow p \vee r}}{\vdash (q \rightarrow r) \rightarrow (p \vee q \rightarrow p \vee r)} \text{右蕴涵}}$$

□

给定一条推理规则

$$\frac{\Gamma_1 \vdash \Delta_1 \quad \dots \quad \Gamma_n \vdash \Delta_n}{\Gamma_{n+1} \vdash \Delta_{n+1}}$$

只要它的所有前提都是有效式，则它的结论一定也是有效式，就称该规则是**可靠的** (*sound*)。

引理

\mathcal{S}_{PL} 的所有推理规则都是**可靠的**。

例

证明 \mathcal{S}_{PL} 的左蕴涵规则是可靠的：

$$\text{(左蕴涵)} \quad \frac{\Gamma \vdash P, \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \rightarrow Q \vdash \Delta}$$

证.

以 (1)、(2)、(3) 分别指代规则的两个前提（先左后右）和结论。设 (1)、(2) 都是有效式，需要证明 (3) 也是有效式。令 ρ 为 (3) 的任何一个赋值，

- 如果 $\llbracket \wedge \Gamma \rrbracket_{\rho} = false$ 或者 $\llbracket P \rightarrow Q \rrbracket_{\rho} = false$ ，则 $\llbracket (3) \rrbracket_{\rho} = true$ （根据相继式的语义）；
- 否则， $\llbracket \wedge \Gamma \rrbracket_{\rho} = true$ 且 $\llbracket P \rightarrow Q \rrbracket_{\rho} = true$ ；根据 (1) 是有效式，要么 $\llbracket P \rrbracket_{\rho} = true$ ，要么 $\llbracket \Delta \rrbracket_{\rho} = true$ 。如果 $\llbracket P \rrbracket_{\rho} = true$ ，则 $\llbracket Q \rrbracket_{\rho} = true$ （因为 $\llbracket P \rightarrow Q \rrbracket_{\rho} = true$ ），所以 $\llbracket \Delta \rrbracket_{\rho} = true$ （因为 (2) 是有效式）。因此，无论哪种情况， $\llbracket (3) \rrbracket_{\rho} = true$ （根据相继式的语义）。

综上，(3) 是有效式，结论成立。



定理 (\mathcal{S}_{PL} 的可靠性)

\mathcal{S}_{PL} 是**可靠的** (*sound*), 即通过该演算系统推导出的所有结论都是有效式。

定理 (\mathcal{S}_{PL} 的完备性)

\mathcal{S}_{PL} 是**完备的** (*complete*), 即所有有效的相继式都可以通过该演算系统推导出来。

定理 (命题逻辑的可靠性与完备性)

设 F 为任意命题逻辑公式, 如果存在一棵以 $\vdash F$ 为根节点的推导树, 则 F 必是永真式, 即 $\models F$ 。如果 F 是永真式, 即 $\models F$, 则必定存在一棵以 $\vdash F$ 为根节点的推导树。

定理 (命题逻辑的可判定性)

命题逻辑是**可判定的** (*decidable*), 即存在一个算法, 对任意给定的命题逻辑公式, 能够在有限时间内正确地判定出该公式是否是有效式。

证明.

回忆一下真值表法, 虽然不是最优算法, 但对任何命题逻辑公式, 总能在有限时间 (指数时间) 内完成是否是有效式的判定。 □

定理: F 是永真式当且仅当 $\neg F$ 是不可满足式。

- 可以应用可满足性判定算法间接判定是否永真
- 目前存在许多非常有效的可满足性判定算法

语法：命题逻辑公式的构成

- 符号集： \top, \perp ，命题变元，逻辑联结词
- 构造规则：原子公式、文字、合式公式

语义：命题逻辑公式的含义

- 真值，变量赋值，公式取值
- 可满足式、永真式、不可满足式
- 语义蕴涵、语义等价

相继式演算系统 \mathcal{S}_{PL} ：证明永真式

- 推理规则：前件规则、后件规则、包含规则、切规则
- 推导树 \leftrightarrow 可推导
- 可靠性、完备性、可判定性

- 一阶逻辑

谢谢!