《软件分析与验证》

命题逻辑

贺飞

清华大学软件学院

2024 年 3 月 17 日

目录

语法

符号集

命题逻辑的符号集（$alphabet$）包括：

- 命题常元 $\top$（代表 $true$）和 $\bot$（代表 $false$）；
- 命题变元；
- 逻辑联结词 $\neg$，$\wedge$，$\vee$，$\rightarrow$ 和 $\leftrightarrow$ 。

其中 $\top, \bot$ 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：

- $\top$，$\bot$ 和命题变元是合式公式；
- 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 $F \wedge G$, $F \vee G$, $F \to G$ 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

语义

令变元赋值（*assignment*，简称赋值，又称真值指派）为一个从变元集到真值集 $\{true, false\}$ 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)，记为 $[\![F]\!]_\rho$，递归定义如下：

$$
\begin{aligned}
[\![\top]\!]_\rho &= true \\
[\![\bot]\!]_\rho &= false \\
[\![p]\!]_\rho &= \rho(p)，\text{其中 } p \text{ 是 } F \text{ 中的一个变元} \\
[\![\neg F]\!]_\rho &= \begin{cases} true，\text{如果} [\![F]\!]_\rho = false \\ false，\text{如果} [\![F]\!]_\rho = true \end{cases}
\end{aligned}
$$

定义 (命题逻辑公式的语义（续）)

$$[\![F \wedge G]\!]_\rho = \begin{cases} true, & \text{如果} [\![F]\!]_\rho = true \text{ 且 } [\![G]\!]_\rho = true \\ false, & \text{否则} \end{cases}$$

$$[\![F \vee G]\!]_\rho = \begin{cases} true, & \text{如果} [\![F]\!]_\rho = true \text{ 或 } [\![G]\!]_\rho = true \\ false, & \text{否则} \end{cases}$$

$$[\![F \to G]\!]_\rho = \begin{cases} false, & \text{如果} [\![F]\!]_\rho = true \text{ 且 } [\![G]\!]_\rho = false \\ true, & \text{否则} \end{cases}$$

$$[\![F \leftrightarrow G]\!]_\rho = \begin{cases} true, & \text{如果} [\![F]\!]_\rho = [\![G]\!]_\rho \\ false, & \text{否则} \end{cases}$$

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 $[\![p]\!]_\rho = true$ 且 $[\![q]\!]_\rho = false$，故 $[\![p \wedge q]\!]_\rho = false$。
又有 $[\![\neg q]\!]_\rho = true$，故 $[\![p \vee \neg q]\!]_\rho = true$。
所以 $[\![F]\!]_\rho = true$。

可满足性

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 $[\![F]\!]_\rho$ 为真；
- 有效的（或永真的）（*valid*），当且仅当对任意赋值 ρ，$[\![F]\!]_\rho$ 都为真；
- 不可满足的（或永假的）（*unsatisfiable*），当且仅当对任意赋值 ρ, $[\![F]\!]_\rho$ 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 $p \wedge q \to p \vee \neg q$ 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 $[\![F]\!]_\rho$ 对任何赋值 ρ 都为真（根据永真式定义），
当且仅当 $[\![\neg F]\!]_\rho$ 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 $[\![F]\!]_\rho$ 为真，$[\![G]\!]_\rho$ 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 $\bot$ 的有效推论。

证明 $F \Rightarrow G$ 的方法：

- 证明 $F \to G$ 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 $\neg(F \to G)$ 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 $[\![F]\!]_\rho = [\![G]\!]_\rho$。

根据命题逻辑的语义，下面公式成立：

$$\begin{aligned}
\top &\Leftrightarrow \neg\bot \\
F_1 \vee F_2 &\Leftrightarrow \neg(\neg F_1 \wedge \neg F_2) \\
F_1 \to F_2 &\Leftrightarrow \neg F_1 \vee F_2 \\
F_1 \leftrightarrow F_2 &\Leftrightarrow (F_1 \to F_2) \wedge (F_2 \to F_1)
\end{aligned}$$

可将任意命题逻辑公式语义等价地转换成只包含 $\bot, \neg, \wedge$ 及命题变元的公式。

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 $\mathcal{S}_{PL}$

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

定义 (相继式)

一个相继式（*sequent*）是形如

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

的式子，其中 $\vdash$ 称相继符，$F_1, \ldots, F_m$ 称前件，$G_1, \ldots, G_n$ 称后件。

例如：$p, q \vdash p \vee \neg q$ 是一个相继式，包含两个前件，分别是 p 和 q，以及一个后件，为 $p \vee \neg q$。

相继式的语义要求在所有前件都成立的情况下，至少有一个后件成立，即：

$$F_1 \wedge \cdots \wedge F_m \to G_1 \vee \cdots \vee G_n$$

如果上式在命题逻辑下是有效式，则称对应的相继式为有效式。

推理规则

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

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

推理规则的中央是一条横线。横线上方有若干个相继式，代表规则的前提。横线下方有一个相继式，代表规则的结论。

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。

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

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

（左否定）$\dfrac{\Gamma \vdash P, \Delta}{\Gamma, \neg P \vdash \Delta}$　　（右否定）$\dfrac{\Gamma, P \vdash \Delta}{\Gamma \vdash \neg P, \Delta}$

（左蕴涵）$\dfrac{\Gamma \vdash P, \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \to Q \vdash \Delta}$　　（右蕴涵）$\dfrac{\Gamma, P \vdash Q, \Delta}{\Gamma \vdash P \to Q, \Delta}$

（包含）$\dfrac{}{\Gamma, P \vdash P, \Delta}$　　（切）$\dfrac{\Gamma \vdash C, \Delta \quad \Gamma, C \vdash \Delta}{\Gamma \vdash \Delta}$

应用推理规则进行推导的过程：

- 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）。

推理规则（前件规则）

从结论到前提，每条规则减少一个逻辑联结词。

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

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

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

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

推理规则（后件规则）

从结论到前提，每条规则减少一个逻辑联结词。

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

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

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

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

推理规则

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

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

- 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，但加上切规则会大大减少证明的步骤，简化证明的过程。

例

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

证明.

$$\text{右蕴涵}\ \dfrac{\text{左合取}\ \dfrac{\text{右析取}\ \dfrac{\text{包含}\ \dfrac{}{p, q \vdash p, \neg q}}{p, q \vdash p \vee \neg q}}{p \wedge q \vdash p \vee \neg q}}{\vdash p \wedge q \to p \vee \neg q}$$

□

定义 (推导树)

推导树（*derivation tree*）是一棵满足下列条件的树：

- 每一个中间节点对应一个相继式；
- 每一个叶子结点要么为空，要么也对应一个相继式；
- 每一个由父节点和子节点构成的片段，形如

$$\frac{F_1 \quad \dots \quad F_n}{G},$$

都对应某个推理规则的实例。

定义 (证明树)

所有叶子结点都是空的推导树称为证明树（*proof tree*）。如果存在一棵以相继式 S 为根节点的证明树，就说 S 是可证明的。

证明树搜索

证明树搜索过程：

1. 从给定的相继式 S_0 出发，构造一棵只包含 S_0 节点的推导树；
2. 在上一步得到的推导树中，挑选一个不为空的叶子结点 S 以及一个可应用的推理规则

$$\frac{S_1 \quad \ldots \quad S_n}{S}$$

扩展推导树以增加新的叶子节点 $S_1, \ldots, S_n$，并连接它们与 S 的边；

3. 反复执行上一步，直至推导树无法再扩展为止，称此时的推导树为最大推导树。

最大推导树：

- 如果其叶子结点都为空，那么它就是一棵证明树，搜索成功；
- 反之，如果存在不为空的叶子结点，搜索失败。

例

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

证明.

$$\dfrac{\dfrac{}{q \to r, p \vdash p, r} \qquad \dfrac{\dfrac{}{q \vdash q, p, r} \qquad \dfrac{}{r, q \vdash p, r}}{q \to r, q \vdash p, r}}{q \to r, p \vee q \vdash p, r}$$

$$\Downarrow$$

$$\text{右蕴涵}\dfrac{\text{右蕴涵}\dfrac{\text{右析取}\dfrac{q \to r, p \vee q \vdash p, r}{q \to r, p \vee q \vdash p \vee r}}{q \to r \vdash p \vee q \to p \vee r}}{\vdash (q \to r) \to (p \vee q \to p \vee r)}$$

□

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 $\mathcal{S}_{PL}$ 的包含公理是可靠的：

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

证.

只需证明 $\Gamma, P \vdash P, \Delta$ 是有效式，即

$$\bigwedge \Gamma \wedge P \to P \vee \bigvee \Delta \tag{1}$$

是有效式。在任何一个赋值下，无论 P 被指派的值为真还是假，显然公式（1）都为真。 □

例

证明 $\mathcal{S}_{PL}$ 的左蕴涵规则是可靠的：

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

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 $[\![\bigwedge \Gamma]\!]_\rho = \mathit{false}$ 或者 $[\![P \to Q]\!]_\rho = \mathit{false}$，则 $[\![(3)]\!]_\rho = \mathit{true}$（根据相继式的语义）；
- 否则，$[\![\bigwedge \Gamma]\!]_\rho = \mathit{true}$ 且 $[\![P \to Q]\!]_\rho = \mathit{true}$；根据（1）是有效式，要么 $[\![P]\!]_\rho = \mathit{true}$，要么 $[\![\Delta]\!]_\rho = \mathit{true}$。如果 $[\![P]\!]_\rho = \mathit{true}$，则 $[\![Q]\!]_\rho = \mathit{true}$（因为 $[\![P \to Q]\!]_\rho = \mathit{true}$），所以 $[\![\Delta]\!]_\rho = \mathit{true}$（因为（2）是有效式）。因此，无论哪种情况，$[\![(3)]\!]_\rho = \mathit{true}$（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 $\vdash F$ 为根节点的证明树，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 $\vdash F$ 为根节点的证明树。

定理 (命题逻辑的可判定性)

命题逻辑是可判定的 (*decidable*)，即存在一个通用算法，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

小结

总结：命题逻辑

语法：命题逻辑公式的构成

- 符号集：$\top, \bot$，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

下节课内容

- 一阶逻辑

谢谢!