《软件分析与验证》

一阶逻辑

贺飞
清华大学软件学院

2024 年 3 月 22 日

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
- 证明树 $\leftrightarrow$ 可证明
- 可靠、完备、可判定

命题逻辑的限制

命题逻辑是可判定的，但其表达能力有限。

下列陈述在命题逻辑中只能被当做不可分的整体对待：

- 小明是清华大学的学生
- 小红是清华大学的学生
- 教室里的同学都是清华大学的学生

事实上，它们之间是有关联的：

- 以 m 代表小明，h 代表小红；
- 以 $thu(x)$ 代表 “x 是清华大学的学生”；
- 上面的陈述可以分别被表述为：
  $thu(m), thu(h), \forall x.classroom(x) \to thu(x)$

为了表达这些概念，需要用到一阶逻辑。

语法

逻辑符号（*logic symbols*）：

- 真值符号 $\top$（代表 *true*）和 $\bot$（代表 *false*）；
- 变元符号；
- 逻辑联结词符号 $\neg$，$\wedge$，$\vee$，$\rightarrow$ 和 $\leftrightarrow$；
- 量词符号 $\forall$ 和 $\exists$。

非逻辑符号（*non-logic symbols*）：

- 常元符号；
- 函数符号，每个函数符号都联系一个正整数，称为它的元数（*arity*）；
- 谓词符号，每个谓词符号都联系一个正整数，称为它的元数。

构造规则：项和原子公式

定义

一阶逻辑的项（*term*）递归定义如下：

- 变元和常元是项；
- 对每一个 n 元函数 f，如果 $t_1, \cdots, t_n$ 都是项，则 $f(t_1, \cdots, t_n)$ 也是项。

定义

一阶逻辑的原子公式（*atomic formula*）定义如下：

- $\top$，$\bot$ 是原子公式；
- 对每一个 n 元谓词 p，如果 $t_1, \cdots, t_n$ 都是项，则 $p(t_1, \cdots, t_n)$ 是原子公式。

构造规则：合式公式

定义

一阶逻辑的合式公式（*well-formed formula*）（简称公式）递归定义如下：

- 原子公式是合式公式；
- 如果 φ 是合式公式，则 $\neg\varphi$ 也是合式公式；
- 如果 φ_1, φ_2 是合式公式，则 $\varphi_1 \wedge \varphi_2$ 也是合式公式；
- 如果 φ 是合式公式，x 是变元，则 $\exists x.\varphi$ 是合式公式。

其中，原子公式和原子公式的非统称为文字（*literal*）。

其他符号的处理：

- 同命题逻辑一样，$\top, \vee, \rightarrow, \leftrightarrow$ 可转换为只含 $\bot, \neg, \wedge$ 的公式；
- 对于量词 $\forall$，引入一条新规则：$\forall x.\varphi := \neg\exists x.\neg\varphi$。

符号上的约定：

- 变元：$x, y, z;$
- 常元：$a, b, c;$
- 函数：$f, g, h;$
- 项：$t;$
- 谓词：$p, q, r;$
- 公式：$\varphi, \psi;$

优先级上的约定：

- 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

对于公式 $\forall x.\varphi(x)$ 和 $\exists x.\varphi(x)$，我们称

- x 为约束变元（*bounded variable*）；
- x 在 $\varphi(x)$ 中的出现为约束出现 [1]；
- $\varphi(x)$ 是量词 $\forall x$ 或 $\exists x$ 的辖域（*scope*）。

如果变元 x 在公式 φ 中的某次出现不是约束出现，就称其为自由出现，同时称 x 为 φ 的自由变元（*free variable*）。

没有自由变元的公式称为闭公式（*closed formula*），也称语句（*sentence*）。

有自由变元的公式称为开公式（*open formula*）。

[1] 约定：x 在量词 $\forall x$ 和 $\exists x$ 中的出现也是约束出现。

量词辖域的确定：按匹配到的最大公式为它的辖域。

例

公式 $\exists x.p(f(x), y) \to \forall y.p(f(x), y)$

- $\exists x$ 的辖域是 $p(f(x), y) \to \forall y.p(f(x), y)$
- $\forall y$ 的辖域是 $p(f(x), y)$
- x 在公式中出现 3 次，均为约束出现
- y 在公式中出现 3 次，第 1 次是自由出现，后 2 次是约束出现

例

用一阶逻辑刻画下列陈述：

- 猫都比狗长寿

$$\forall x, y.\ dog(x) \land cat(y) \rightarrow ndays(y) > ndays(x)$$

- 三角形任何一条边的长度小于另两条边长度之和

$$\begin{aligned}\forall v_1, v_2, v_3.\ triangle(v_1, v_2, v_3) \rightarrow \\ dis(v_1, v_2) < (dis(v_2, v_3) + dis(v_1, v_3)) \\ \land\ dis(v_1, v_3) < (dis(v_1, v_2) + dis(v_2, v_3)) \\ \land\ dis(v_2, v_3) < (dis(v_1, v_2) + dis(v_1, v_3))\end{aligned}$$

- 数组 a 中的所有元素都是正数

$$\forall i.\ 0 \leq i < |a| \rightarrow a[i] > 0$$

例

用一阶逻辑刻画下列陈述:

- 至少有两只猫正在吃东西

$$\exists x, y.\ x \neq y \wedge cat(x) \wedge cat(y) \wedge eating(x) \wedge eating(y)$$

- 猫咪喜欢的只有鱼

$$\forall x. cat(x) \rightarrow \forall y. (love(x, y) \rightarrow fish(y))$$

- 每个有头驴的农民都会打它（`Every farmer who owns a donkey beats it`）。

$$\forall x, y.\ farmer(x) \wedge donkey(y) \wedge owns(x, y) \rightarrow beat(x, y)$$

语义

定义

一阶逻辑的一个**解释**（*interpretation*）$\mathcal{M} = (\mathcal{D}, \mathcal{I})$ 由两部分构成，其中 $\mathcal{D}$ 是一个非空集合，包含了所有希望讨论的元素，称为**论域**（*domain*）；$\mathcal{I}$ 是一个满足下列要求的**解释函数**（*interpretation function*）：

- 为每个常元指定 $\mathcal{D}$ 中的一个元素；
- 为每个 n 元函数符号 f 指定 $\mathcal{D}$ 上的一个 n 元函数

$$f_I : \mathcal{D}^n \mapsto \mathcal{D}$$

- 为每个 n 元谓词符号 p 指定 $\mathcal{D}$ 上的一个 n 元关系

$$p_I \subseteq \mathcal{D}^n$$

以 $FVar(\varphi)$ 表示公式 φ 中自由变元的集合，以 $\rho : FVar(\varphi) \to \mathcal{D}$ 表示从 $FVar(\varphi)$ 到 $\mathcal{D}$ 的一个映射函数，称为**赋值**（*assignment*）。

例

$$x + y > z \quad \to \quad y > z - x$$

这个公式中出现了算术运算符和比较操作符。根据常识，人们对这些符号的预期解释（*intended interpretation*）是：

- 论域 $\mathcal{D} = \mathbb{Z} = \{\ldots, -1, 0, 1, \ldots\}$
- 解释 $\mathcal{I} = \{+ \mapsto +_{\mathbb{Z}}, - \mapsto -_{\mathbb{Z}}, > \mapsto >_{\mathbb{Z}}\}$

我们完全可以给一个跟预期解释不一样的解释。

定义

项 t 在解释 $\mathcal{M}$ 和赋值 ρ 下的取值（*evaluation*）$[\![t]\!]_{\mathcal{M},\rho}$ 递归定义如下：

- 若 t 为常元 c，则 $[\![c]\!]_{\mathcal{M},\rho} = \mathcal{I}(c)$；
- 若 t 为变元 v，则 $[\![v]\!]_{\mathcal{M},\rho} = \rho(v)$；
- 若 t 为函数项 $f(t_1, \ldots, t_n)$，则 $[\![f(t_1, \ldots, t_n)]\!]_{\mathcal{M},\rho} = \mathcal{I}(f)([\![t_1]\!]_{\mathcal{M},\rho}, \ldots, [\![t_n]\!]_{\mathcal{M},\rho})$。

设 ρ 为赋值，x 为变元，c 为论域中的一个值，$\rho[x \mapsto c]$ 是 ρ 的一个变体，满足

- x 的赋值为 c，
- 除 x 以外其他变量的赋值与 ρ 一致。

定义

公式 φ 在解释 $\mathcal{M}$ 和赋值 ρ 下的取值 $[\![\varphi]\!]_{\mathcal{M},\rho}$ 递归定义如下：

$$
\begin{aligned}
[\![\bot]\!]_{\mathcal{M},\rho} &= \mathit{false} \\
[\![p(t_1,\ldots,t_n)]\!]_{\mathcal{M},\rho} &= \begin{cases} \mathit{true}, \text{ 如果}([\![t_1]\!]_{\mathcal{M},\rho},\ldots,[\![t_n]\!]_{\mathcal{M},\rho}) \in \mathcal{I}(p) \\ \mathit{false}, \text{ 否则} \end{cases} \\
[\![\neg\varphi]\!]_{\mathcal{M},\rho} &= \begin{cases} \mathit{true}, \text{ 如果}[\![\varphi]\!]_{\mathcal{M},\rho} = \mathit{false} \\ \mathit{false}, \text{ 如果}[\![\varphi]\!]_{\mathcal{M},\rho} = \mathit{true} \end{cases} \\
[\![\varphi_1 \wedge \varphi_2]\!]_{\mathcal{M},\rho} &= \begin{cases} \mathit{true}, \text{ 如果}[\![\varphi_1]\!]_{\mathcal{M},\rho} = \mathit{true}, [\![\varphi_2]\!]_{\mathcal{M},\rho} = \mathit{true} \\ \mathit{false}, \text{ 否则} \end{cases} \\
[\![\exists x.\varphi]\!]_{\mathcal{M},\rho} &= \begin{cases} \mathit{true}, \text{ 如果存在 } c \in \mathcal{D}.\ [\![\varphi]\!]_{\mathcal{M},\rho[x\mapsto c]} = \mathit{true} \\ \mathit{false}, \text{ 否则} \end{cases}
\end{aligned}
$$

例

考虑论域 $\mathcal{D} = \{\circ, \bullet\}$，下面的解释函数

- $\mathcal{I}(a) = \circ$
- $\mathcal{I}(f) = \{(\circ, \circ) \mapsto \circ, (\circ, \bullet) \mapsto \bullet, (\bullet, \circ) \mapsto \bullet, (\bullet, \bullet) \mapsto \circ\}$
- $\mathcal{I}(g) = \{\circ \mapsto \bullet, \bullet \mapsto \circ\}$
- $\mathcal{I}(p) = \{(\bullet, \circ), (\bullet, \bullet)\}$

和赋值 $\rho = \{x \mapsto \bullet, y \mapsto \circ\}$，求 $p(x, f(g(x), a)) \to p(y, g(x))$ 的取值。

解

- $[\![x]\!]_{\mathcal{M},\rho} = \rho(x) = \bullet$，$[\![y]\!]_{\mathcal{M},\rho} = \rho(y) = \circ$，$[\![a]\!]_{\mathcal{M},\rho} = \mathcal{I}(a) = \circ$
- $[\![g(x)]\!]_{\mathcal{M},\rho} = \mathcal{I}(g)([\![x]\!]_{\mathcal{M},\rho}) = \mathcal{I}(g)(\bullet) = \circ$
- $[\![f(g(x), a)]\!]_{\mathcal{M},\rho} = \mathcal{I}(f)([\![g(x)]\!]_{\mathcal{M},\rho}, [\![a]\!]_{\mathcal{M},\rho}) = \mathcal{I}(f)(\circ, \circ) = \circ$

由 $(\circ, \circ) \notin \mathcal{I}(p)$ 得 $[\![p(y, g(x))]\!]_{\mathcal{M},\rho} = \mathit{false}$；由 $(\bullet, \circ) \in \mathcal{I}(p)$ 得 $[\![p(x, f(g(x), a))]\!]_{\mathcal{M},\rho} = \mathit{true}$；所以原式取值为 false。

例

考虑论域 $\mathcal{D} = \{\circ, \bullet\}$，下面的解释函数

- $\mathcal{I}(a) = \circ$
- $\mathcal{I}(f) = \{(\circ, \circ) \mapsto \circ, (\circ, \bullet) \mapsto \bullet, (\bullet, \circ) \mapsto \bullet, (\bullet, \bullet) \mapsto \circ\}$
- $\mathcal{I}(g) = \{\circ \mapsto \bullet, \bullet \mapsto \circ\}$
- $\mathcal{I}(p) = \{(\bullet, \circ), (\bullet, \bullet)\}$

和赋值 $\rho = \{x \mapsto \bullet, y \mapsto \circ\}$，求公式 $\exists x.\neg p(x, g(a))$ 的取值。

解

首先

- $[\![a]\!]_{\mathcal{M},\rho} = \mathcal{I}(a) = \circ$
- $[\![g(a)]\!]_{\mathcal{M},\rho} = \mathcal{I}(g)([\![a]\!]_{\mathcal{M},\rho}) = \mathcal{I}(g)(\circ) = \bullet$

考察 $x \mapsto \circ$ 的情况：由于 $(\circ, \bullet) \notin \mathcal{I}(p)$，所以 $[\![p(\circ, g(a))]\!]_{\mathcal{M},\rho} = \mathit{false}$，于是 $[\![\neg p(\circ, g(a))]\!]_{\mathcal{M},\rho} = \mathit{true}$，故 $[\![\exists x.\neg p(x, g(a))]\!]_{\mathcal{M},\rho} = \mathit{true}$。

变元重命名

考虑公式 $\forall x.p(x)$，其中 x 为约束变元，其含义为：对 x 在论域 $\mathcal{D}$ 中的任何取值 c，$p(c)$ 都成立。
显然，公式 $\forall x.p(x)$ 的语义与约束变元采用哪个符号无关。

定义 (变元重命名)

设 x 是公式 φ 的一个约束变元，以一个在 φ 中未出现过的新变元 y 同时替换 x 在 φ 中的所有约束出现，称为变元重命名（Renaming）。

注意：

- 在变元重命名时，一定要采用新变元。
  - 例如：考虑将公式 $\forall x.x = y \to f(x) = f(y)$ 中的约束变元 x 重命名为 y，得到 $\forall y.y = y \to f(y) = f(y)$，跟原公式相比，新公式的语义发生了很大改变！
- 只对约束变元进行重命名，不对自由变元重命名。

定义 (代换)

设 φ 是一个合式公式，x 是一个变元，t 是一个项，记 $\varphi[x \mapsto t]$ 为将 φ 中 x 的所有**自由出现**同时替换为 t 得到的结果，称为 φ 的一个代换。

例如：考虑合式公式 $\varphi : \forall x.\, x = y \to f(x) = f(y)$，

- $\varphi[y \mapsto z]$ 为 $\forall x.\, x = z \to f(x) = f(z)$
- $\varphi[y \mapsto f(y)]$ 为 $\forall x.\, x = f(y) \to f(x) = f(f(y))$
- $\varphi[y \mapsto f(x)]$ 为 $\forall x.\, x = f(x) \to f(x) = f(f(x))$

注意上面的第三次代换：原公式中的 y 是自由变元，可以任意取值；将 y 代换为 $f(x)$ 之后，$f(x)$ 中的 x 被量词 $\forall x$ 所约束，$f(x)$ 的值也将受到约束，公式的意义也就发生了改变。称这类情况为变元捕获。

如何避免变元捕获?

——在代换之前先重命名公式中可能导致捕获的约束变元。

更一般的代换形式是：$\varphi[x_1 \mapsto t_1, \cdots, x_n \mapsto t_n]$, 其中 x_i 是变元，t_i 是项，表示将 φ 中 $x_1, \cdots, x_n$ 的所有自由出现**分别同时**替换为 $t_1, \cdots, t_n$ 的结果。

为避免变元捕获

- 考察 φ 中的每一个约束变元 x，若 x 在任何 t_i 中出现，则将 x 重命名为一个新变元 x'；
- 以 φ' 表示变元重命名后的公式，对 φ' 应用代换。

例

给定

$$\varphi: \quad (\forall x.p(x,y)) \to q(f(y),x)$$

计算 $\varphi\,[x \mapsto g(x,y),\ y \mapsto f(x)]$

解

- φ 中的约束变元 x 在代换项中出现，重命名为 x'，得到（注意最后的 x 不是约束出现，不需要重命名）：

$$\varphi': \quad (\forall x'.p(x',y)) \to q(f(y),x)$$

- 同时应用代换 $x \mapsto g(x,y)$ 和 $y \mapsto f(x)$，得到（注意是同时代换，而非顺序代换）：

$$\varphi'': (\forall x'.p(x',f(x))) \to q(f(f(x)),g(x,y))$$

可满足性

定义

一阶逻辑公式 φ 是

- 可满足式（*satisfiable*），当且仅当存在一个解释 $\mathcal{M}$ 和一个赋值 ρ，使得 $[\![\varphi]\!]_{\mathcal{M},\rho}$ 为真；
- 有效式（或永真式）（*valid*），当且仅当对任意解释 $\mathcal{M}$ 和任意赋值 ρ，$[\![\varphi]\!]_{\mathcal{M},\rho}$ 都为真。

φ 是永真式常常记作 $\models \varphi$。

定理

φ 是永真式当且仅当 $\neg\varphi$ 是永假式。

例（公式 $\exists x.f(x)=g(x)$ 可满足吗？）

原式在下面的解释下为 $true$，所以是可满足的：

- $D=\{0,1\}$
- $I(f)=\{0\mapsto 1, 1\mapsto 1\}$
- $I(g)=\{0\mapsto 0, 1\mapsto 1\}$

例（公式 $\exists x.f(x)=g(x)$ 是有效式吗？）

原式在下面的解释下为 $false$，所以不是有效式：

- $D=\{0,1\}$
- $I(f)=\{0\mapsto 1, 1\mapsto 1\}$
- $I(g)=\{0\mapsto 0, 1\mapsto 0\}$

定义 (语义蕴涵)

给定两个一阶逻辑公式 φ 和 ψ，如果对任意解释 $\mathcal{M}$ 和任意赋值 ρ，只要 $[\![\varphi]\!]_{\mathcal{M},\rho}$ 为真，$[\![\psi]\!]_{\mathcal{M},\rho}$ 就必为真，就称 φ 语义蕴涵（*implies*）ψ，或称 ψ 是 φ 的有效推论（*consequence*），记为 $\varphi \Rightarrow \psi$。

例如：$p(a)$ 是 $\forall x.p(x)$ 的有效推论。

证明系统

类似于命题逻辑，我们也采用相继式演算作为一阶逻辑公式的证明系统，记为 $\mathcal{S}_{FOL}$。

- 基本思想也是从待证相继式出发，通过应用推理规则逐步消去公式中的逻辑联结词和量词。
- 相继式 S 可证明当且仅当存在一棵以 S 为根节点的证明树。
- 对应于每一个逻辑联结词，$\mathcal{S}_{FOL}$ 有与 $\mathcal{S}_{PL}$ 类似的推理规则。
- $\mathcal{S}_{FOL}$ 有与 $\mathcal{S}_{PL}$ 类似的包含规则和且规则。
- 除此之外，$\mathcal{S}_{FOL}$ 还需要推理规则来处理量词。

推理规则

（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） $\dfrac{\Gamma \vdash P, \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \rightarrow Q \vdash \Delta}$

（右蕴涵） $\dfrac{\Gamma, P \vdash Q, \Delta}{\Gamma \vdash P \rightarrow Q, \Delta}$

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

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

从结论到前提，每条规则减少一个量词。

$$\text{（左全称）}\ \frac{\Gamma, \varphi[x \mapsto t] \vdash \Delta}{\Gamma, \forall x.\varphi(x) \vdash \Delta}$$

$$\text{（右全称）}\ \frac{\Gamma \vdash \varphi(c), \Delta}{\Gamma \vdash \forall x.\varphi(x), \Delta}\ \text{（c 在 $\Gamma, \varphi(x), \Delta$ 中不出现）}$$

$$\text{（左存在）}\ \frac{\Gamma, \varphi(c) \vdash \Delta}{\Gamma, \exists x.\varphi(x) \vdash \Delta}\ \text{（c 在 $\Gamma, \varphi(x), \Delta$ 中不出现）}$$

$$\text{（右存在）}\ \frac{\Gamma \vdash \varphi[x \mapsto t], \Delta}{\Gamma \vdash \exists x.\varphi(x), \Delta}$$

例 (证明 $\vdash (\forall x.p(x)) \to (\forall y.p(y))$)

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{p(c) \vdash p(c)}\text{包含}}{\forall x.p(x) \vdash p(c)}\text{左全称}}{\forall x.p(x) \vdash \forall y.p(y)}\text{右全称}}{\vdash (\forall x.p(x)) \to (\forall y.p(y))}\text{右蕴涵}
$$

例 (证明 $(\forall x.p(x)) \leftrightarrow (\neg\exists x.\neg p(x))$)

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p(c) \vdash p(c)}\,\text{id}}{\forall x.p(x) \vdash p(c)}\,\forall\text{L}}{\forall x.p(x), \neg p(c) \vdash \bot}\,\neg\text{L}}{\forall x.p(x), \exists x.\neg p(x) \vdash \bot}\,\exists\text{L}}{\forall x.p(x) \vdash \neg\exists x.\neg p(x)}\,\neg\text{R} \qquad \text{略}}{\vdash (\forall x.p(x)) \leftrightarrow (\neg\exists x.\neg p(x))}\,\leftrightarrow\text{R}
$$

定理 ($\mathcal{S}_{FOL}$ 的可靠性)

$\mathcal{S}_{FOL}$ 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 ($\mathcal{S}_{FOL}$ 的完备性)

$\mathcal{S}_{FOL}$ 是完备的（*complete*），即所有有效的一阶逻辑相继式都可以通过该演算系统推导出来。

定理 (一阶逻辑的可靠性与完备性)

设 φ 为任意一阶逻辑公式，如果存在一棵以 $\vdash \varphi$ 为根节点的证明树，则 φ 必是有效式，即 $\models \varphi$。如果 φ 是有效式，即 $\models \varphi$，则必定存在一棵以 $\vdash \varphi$ 为根节点的证明树。

定理 (一阶逻辑半可判定性)

一阶逻辑是半可判定的（*semi-decidable*），即判定一阶逻辑公式是否有效的算法

- 只有在该公式是有效式的前提下，才能保证在有限时间内终止并给出正确结果；
- 否则，可能永远不终止。

总结：一阶逻辑

- **语法**：一阶逻辑公式的构成
  - 符号集：逻辑符号、非逻辑符号
  - 构造规则：项、原子公式、文字、合式公式
- **语义**：一阶逻辑公式的含义
  - 解释 + 变量赋值：项求值和公式求值
  - 可满足式、有效式
  - 语义蕴涵
- **相继式演算系统** $\mathcal{S}_{FOL}$：证明一阶逻辑有效式
  - 推理规则
  - 可靠、完备、半可判定

下节课内容

- 一阶理论

谢谢!