



数理逻辑

02 - 一阶逻辑

(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



什么是一阶（**first-order**）？



一阶语言

逻辑符号

- I. 标点符号：“(”, “)”, “,”
2. 联结词： $\rightarrow$, $\neg$
 - » 其它联结词： $\wedge$, $\vee$, $\leftrightarrow$
3. 变元（可数无穷个）： $v_0, v_1, \dots$
4. 量词： $\forall$
 - » 其它量词： $\exists$
5. 等词（可选）： $=$

非逻辑符号 (signature)

- I. n 元谓词（可数无穷个）， $n \geq 1$ ：
 $A_0^n, A_1^n, \dots$
2. 常元（可数无穷个）： $c_0, c_1, \dots$
 - » 特殊常元： $\top, \perp, T, F$
3. n 元函数（可数无穷个）：
 $f_0^n, f_1^n, \dots$



一阶语言的例子

纯一阶逻辑 (*pure First-Order Logic, FOL*)

常元	$c_0, c_1, \dots$
n 元谓词	$A_0^n, A_1^n, \dots$
n 元函数	$f_0^n, f_1^n, \dots$
等词	无

集合论

常元	$\emptyset$
2元谓词	$\in$
n 元函数	无
等词	有



一阶语言的例子

初等数论

常元	0
谓词	$<$
1元函数	S
2元函数	$+, \times, \mathbf{E}$
等词	有

› 注: **S**, **E**分别为后继和指数函数

上面的例子里, 我们只是按照习惯来定义各个 language signature 中谓词与函数的**记号**, 例如 $<, \in, \leq$ 只是 A_0^2 ; **0**, $\emptyset$ 为 c_0 ; $+, \times, \mathbf{E}$ 为 f_0^2, f_1^2, f_3^2



FOL之于数学

因为：

- › 一阶语言包含集合论语言
- › 一切数学都可被嵌入在集合论中

所以：

1. 集合论的**一阶语言**能够描述任意数学
2. 一切数学定理都来自对集合论公理的**逻辑推导**

一阶逻辑语言足以描述一切数学结构吗？



FOL的例子 [ENDERTON, PP.73]

1. “所有苹果都坏了”

$$\forall v_1(A(v_1) \rightarrow B(v_1))$$

2. “有些苹果坏了”

$$\exists v_1(A(v_1) \wedge B(v_1))$$

$$\neg \forall v_1(\neg(\neg(A(v_1) \rightarrow (\neg B(v_1))))))$$

3. 所有 X 都属于 Y

$$\gg \quad \forall v_1(X(v_1) \rightarrow Y(v_2))$$

$$\gg \quad \forall v_1(X(v_1) \wedge Y(v_2)) \text{ (语气太强: 所有东西都是 } X \text{ 而且也是 } Y \text{)}$$

4. 存在 X 属于 Y

$$\gg \quad \exists v_1(X(v_1) \wedge Y(v_1))$$

$$\gg \quad \exists v_1(X(v_1) \rightarrow Y(v_1)) \text{ (语气太弱: 存在一些东西, 只有当它是 } X \text{ 时, 它才是 } Y \text{)}$$

定义 2.1 (项, *term*) [Enderton, pp.74]:

每个 n 元函数符号 f 对应一个 n 元项构造算子 $\mathcal{F}_f$:

$$\mathcal{F}_f(\epsilon_1, \dots, \epsilon_n) = f\epsilon_1 \dots \epsilon_n$$

一阶逻辑的项的集合是常元和变元符号经过 (0 次或多次) $\mathcal{F}_f$ 运算得到的表达式集合

例如:

$+ v_2 \mathbf{S}0,$
 $\mathbf{SSSS}0,$
 $+ \mathbf{E}v_1 \mathbf{SS}0 \mathbf{E}v_2 \mathbf{SS}0.$



原子公式

定义 2.2 (原子公式, *atomic formula*, *atom*) [Enderton, pp.74]:

一阶逻辑的原子公式是如下形式的表达式

$$Pt_1 \cdots t_n$$

其中 P 是 n 元谓词, $t_1, \dots, t_n$ 是项

例如:

$$\begin{aligned} &= v_1 v_2, \\ &< SS0 SSS0, \\ &= Ev_1 SS0 Ev_2 SS0. \end{aligned}$$



合式公式

定义 2.3 (合式公式, *well-formed formula*, wff) [Enderton, pp.75]:

一阶逻辑的合式公式集合是原子公式通过运用0次或多次公式构造算子 $\mathcal{E}_{\neg}, \mathcal{E}_{\leftarrow}, Q_i (i = 1, 2, \dots)$ 形成的表达式集合。其中

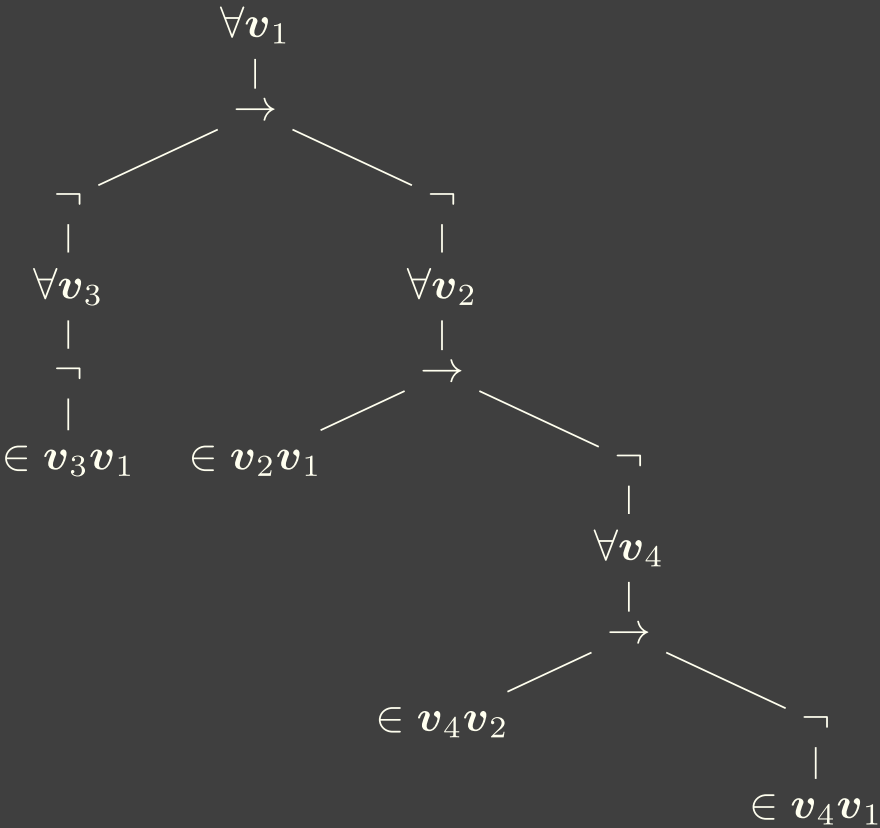
$$\begin{aligned}\mathcal{E}_{\neg}(\gamma) &= (\neg\gamma) \\ \mathcal{E}_{\leftarrow}(\gamma, \delta) &= (\gamma \leftarrow \delta) \\ Q_i(\gamma) &= \forall v_i \gamma\end{aligned}$$

例如:

- > $\neg v_i$ 不是 wff
- > $\forall v_1((\neg \forall v_3(\neg \in v_3 v_1)) \rightarrow (\neg \forall v_2(\in v_2 v_1 \rightarrow (\neg \forall v_4(\in v_4 v_2 \rightarrow (\neg \in v_4 v_1)))))$ 是 wff (ZFC 的正则公理)



$$\forall v_1((\neg \forall v_3(\neg \in v_3 v_1)) \rightarrow (\neg \forall v_2(\in v_2 v_1 \rightarrow (\neg \forall v_4(\in v_4 v_2 \rightarrow (\neg \in v_4 v_1))))))$$





- > $\forall v_2 \in v_2 v_1$
 - » “所有集合都是_____₁的元素”
- > $(\neg \forall v_1 (\neg \forall v_2 \in v_2 v_1))$
 - » “存在一个集合，任意集合都是它的元素”
 - » 记 $\exists x \alpha \equiv \neg \forall x (\neg \alpha)$



$$\varphi \equiv \sum_{j=0}^k a_j$$



自由变元

定义 2.4 (自由出现, *occur free*) [Enderton, pp.76]:

考虑一个变元 x , 我们递归地定义:

1. 对于原子公式 α , x 在 α 中自由出现当且仅当 x 在 α 中出现
2. x 在 $(\neg\alpha)$ 中自由出现当且仅当 x 在 α 中自由出现
3. x 在 $(\alpha \rightarrow \beta)$ 中自由出现当且仅当 x 在 α 中自由出现或在 β 中自由出现
4. x 在 $(\forall v_i \alpha)$ 中自由出现当且仅当 x 在 α 中出现且 $x \neq v_i$

若一个变元如果不是自由 (*not free*) 的, 则我们称它为 (受) 约束的 (*bounded*)。若一个 wff 没有自由出现的变元, 则称它是闭公式 (*closed wff*) 或语句 (*sentence*)



$$(\varphi)_2^k = \left(\sum_{j=0}^k a_j \right)_2^k = \sum_{j=0}^2 a_j$$



$$(\varphi)_i^j = \left(\sum_{j=0}^k a_j \right)_i^j = \sum_{j=0}^k a_j$$



变元替换

定义 2.5a (替换, *substitution*) [Enderton, pp.112]

wff中的变元替换 (substitution) 可递归地定义如下:

1. 对任意原子公式 α , α_t^x 是用项 t 代替 α 中出现的所有 x 后所得的表达式
2. $(\neg \alpha)_t^x = (\neg \alpha_t^x)$
3. $(\alpha \rightarrow \beta)_t^x = (\alpha_t^x \rightarrow \beta_t^x)$
4. $(\forall y \alpha)_t^x = \begin{cases} \forall y \alpha, & \text{if } x = y \\ \forall y (\alpha)_t^x, & \text{if } x \neq y \end{cases}$

我们记这个替换算子为 $\theta = [t/x]$, 则 $\alpha_t^x = \alpha \circ \theta = \alpha\theta$



$$\left[\sum_{j=0}^n (k \cdot a_j) \right]_{f(j)}^k \stackrel{?}{=} \sum_{j=0}^n (f(j) \cdot a_j)$$



可替换

定义 2.5b (可替换, *substitutable*) [Enderton, pp.II2]

我们定义项 t 对于 α 中的变元 x 是**可替换的** (substitutable) 如下:

- I. 对任意原子公式 α , 项 t 对于 α 中出现的所有 x 都是可替换的 (原子公式无量词)
2. 项 t 对于 $(\neg\alpha)$ 中的 x 是可替换的, 当且仅当它对 α 中出现的 x 是可替换的
3. 项 t 对于 $(\alpha \rightarrow \beta)$ 中的 x 是可替换的, 当且仅当它对 α 和 β 中出现的 x 均是可替换的
4. 项 t 对于 $\forall y \alpha$ 中的 x 是可替换的当且仅当:
 - » x 在 $\forall y \alpha$ 中是**约束出现的** (为保证替换后不影响原公式意义), 或者
 - » y 在 t 中**未出现** (为保证 t 中的变元不在替换后被 $\forall y$ 量化) 且 t 对于 α 中的 x 是可替换的



替换的例子

一阶逻辑的 Hilbert 系统里有一条公理模式：

$$\forall x \alpha \rightarrow \alpha_t^x$$

其中 t 对于 α 中的 x 是可替换的，那么：

› 以下 wff 是该公理模式的一个实例

$$\forall v_3 (\forall v_1 (Av_1 \rightarrow \forall v_2 Av_2) \rightarrow (Av_2 \rightarrow \forall v_2 Av_2))$$

» 其中 α 是 $(Av_1 \rightarrow \forall v_2 Av_2)$ ， x 是 v_1 ， t 是 v_2

› 以下 wff 则不是该公理模式的实例

$$\forall v_1 \neg \forall v_2 Bv_1v_2 \rightarrow \neg \forall v_2 Bv_2v_2$$

» 因为 v_2 在 $\neg \forall v_2 Bv_1v_2$ 中受约束，因此它不能替换这里的 v_1



证明



证明的定义

定义 2.6 (定义 1.19-21) (*proof, deduction, or derivation*) [Enderton, pp.III]:

从 Γ 到 φ 的**证明** (推导) 是一个有穷的 wff 序列 $\langle \alpha_0, \dots, \alpha_n \rangle$, 其中 α_n 就是 φ 且对任意 $k \leq n$ 有

1. α_k 属于 $\Gamma \cup \Lambda$ (Λ 为公理集), 或者
2. α_k 是由序列中位于它前面的两个 wff 经过 MP 规则推导而得; 即存在 $i, j \leq k$ 有 α_j 为 $\alpha_i \rightarrow \alpha_k$

若以上证明存在, 我们就说 φ 是从 Γ 出发**可证的** (*provable, deducible or derivable*), 或称 φ 是 Γ 中的**定理** (*theorem of Γ*), 记为 $\Gamma \vdash \varphi$



一阶逻辑中的公理系统

定义 2.7 (一阶逻辑的公理系统, *Axiomatic system for FOL*) [Enderton, pp.110]:

令 α 与 β 为 FOL wff, x 和 y 为变元。FOL 的公理集 Λ 是具有以下形式的 wff 的所有概括:

1. 重言式 (例如定义 1.22 中的公理)
2. $\forall x \alpha \rightarrow \alpha_t^x$, 其中 t 对于 α 中的 x 是可替换的
 - » 它的逆否命题也是公理: $\beta \rightarrow \exists x \beta_x^t$, 其中 x 不在 t 中出现
3. $\forall x (\alpha \rightarrow \beta) \rightarrow (\forall x \alpha \rightarrow \forall x \beta)$
4. $\alpha \rightarrow \forall x \alpha$, 其中 x 不在 α 中自由出现
5. (有等词时) $x = x$
6. (有等词时) $(x = y) \rightarrow (\alpha \rightarrow \alpha')$, 其中 α 为原子公式, α' 为对 α 中的 x 进行 0 个或多个位置的替换后得到的 wff



一阶逻辑中的公理系统

(续) 定义 2.7 (一阶逻辑的公理系统, *Axiomatic system for FOL*) :

该公理系统中只有一条 MP 推理规则, 即若推导中已有 $\beta \rightarrow \alpha$ 和 β , 那么 α 也是正确的, 即:

$$\beta, (\beta \rightarrow \alpha) \vdash \alpha$$



一阶逻辑推导的例子

试证：公理 2.7.2 的推论

$$\vdash Px \rightarrow \exists y Py$$

证明：

Lemma 1: $\vdash (\alpha \rightarrow \beta) \rightarrow (\neg\beta \rightarrow \neg\alpha)$, 根据演绎定理, 即证 $\{\alpha \rightarrow \beta, \neg\beta\} \vdash \neg\alpha$

1. $(\alpha \rightarrow \beta) \rightarrow ((\alpha \rightarrow \neg\beta) \rightarrow \neg\alpha)$ AX 1.22.9
2. $\alpha \rightarrow \beta$ Hyp
3. $(\alpha \rightarrow \neg\beta) \rightarrow \neg\alpha$ 1, 2 MP
4. $\neg\beta \rightarrow (\alpha \rightarrow \neg\beta)$ Ax, 1.22.7
5. $\neg\beta$ Hyp
6. $\alpha \rightarrow \neg\beta$ 4, 5 MP
7. $\neg\alpha$ 3, 6 MP



一阶逻辑推导的例子

试证：定义 2.7.2 的推论

$$\vdash Px \rightarrow \exists y Py$$

证明 (Cont'd) :

Lemma 2 (*syllogism*) : $\vdash (\alpha \rightarrow \beta) \rightarrow ((\beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \gamma))$, 用演绎定理易证

Lemma 3: $\vdash \alpha \rightarrow \neg\neg\alpha$, 根据演绎定理, 即证 $\{\alpha\} \vdash \neg\neg\alpha$

- | | | |
|----|--|-------------|
| 1. | α | Hyp |
| 2. | $\alpha \rightarrow (\neg\alpha \rightarrow (\alpha \wedge \neg\alpha))$ | Ax 1.22.3 |
| 3. | $\alpha \rightarrow (\neg\alpha \rightarrow \perp)$ | Rewrite/Def |
| 4. | $\neg\alpha \rightarrow \perp$ | 1, 3 MP |
| 5. | $(\neg\alpha \rightarrow \perp) \rightarrow \neg\neg\alpha$ | Ax 1.22.13 |
| 6. | $\neg\neg\alpha$ | 4, 5 MP |



一阶逻辑推导的例子

试证：定义 2.7.2 的推论

$$\vdash Px \rightarrow \exists y Py$$

证明 (Cont'd) :

- | | | |
|----|---|-----------|
| 1. | $\forall y \neg Py \rightarrow \neg Px$ | AX 2.7.2 |
| 2. | $(\forall y \neg Py \rightarrow \neg Px) \rightarrow (\neg \neg Px \rightarrow \neg \forall y \neg Py)$ | Lemma 1 |
| 3. | $\neg \neg Px \rightarrow \neg \forall y \neg Py$ | 1, 2 MP |
| 4. | $\neg \neg Px \rightarrow \exists y Py$ | Rewrite 3 |
| 5. | $(Px \rightarrow \neg \neg Px) \rightarrow [(\neg \neg Px \rightarrow \exists y Py) \rightarrow (Px \rightarrow \exists y Py)]$ | Lemma 2 |
| 6. | $Px \rightarrow \neg \neg Px$ | Lemma 3 |
| 7. | $(\neg \neg Px \rightarrow \exists y Py) \rightarrow (Px \rightarrow \exists y Py)$ | 5, 6 MP |
| 8. | $Px \rightarrow \exists y Py$ | 4, 7 MP |

Q.E.D.



一阶逻辑推导的例子2

试证：

$$\vdash (\forall x Px \wedge \forall y Qy) \rightarrow \forall x (Px \wedge Qx)$$

证明：

Lemma: $\{\alpha \rightarrow \beta, \alpha \rightarrow \gamma\} \vdash \alpha \rightarrow (\beta \wedge \gamma)$, 根据演绎定理和公理1.22.3 易证

1. $(\forall x Px \wedge \forall y Qy) \rightarrow \forall x Px$ Ax 1.22.1
2. $\forall x Px \rightarrow Pt$ AX 2.7.2
3. $(\forall x Px \wedge \forall y Qy) \rightarrow Pt$ 1, 2 Syl.
4. $(\forall x Px \wedge \forall y Qy) \rightarrow \forall y Qy$ Ax 1.22.1
5. $\forall y Qy \rightarrow Qt$ AX 2.7.2
6. $(\forall x Px \wedge \forall y Qy) \rightarrow Qt$ 4, 5 Syl.
7. $(\forall x Px \wedge \forall y Qy) \rightarrow (Pt \wedge Qt)$ 3, 6 Lemma
8. $(\forall x Px \wedge \forall y Qy) \rightarrow \forall x (Px \wedge Qx)$ Ded. and Gen.

Q.E.D.



一阶逻辑的元定理



演绎定理

引理 2.8 (T 规则) [Enderton, pp.118]:

若 $\Gamma \vdash \alpha_1, \dots, \Gamma \vdash \alpha_n$ 且 $\{\alpha_1, \dots, \alpha_n\}$ 重言蕴涵 β (即 $(\alpha_1 \rightarrow (\alpha_2 \rightarrow \dots \rightarrow (\alpha_n \rightarrow \beta)))$ 为重言式), 则 $\Gamma \vdash \beta$

› 通过 n 次 MP 推理可证

定理 2.9 (演绎定理) [Enderton, pp.118]:

$\Gamma \cup \{\alpha\} \vdash \beta$ 当且仅当 $\Gamma \vdash \alpha \rightarrow \beta$

› **证明:** 见命题逻辑的演绎定理 (1.25)



一些其它的元定理

推论 2.10 (逆否命题, *Contraposition*) [Enderton, pp.119]:

$\Gamma \cup \{\varphi\} \vdash \neg\psi$ 当且仅当 $\Gamma \cup \{\psi\} \vdash \neg\varphi$

› 前面的例子中已证

推论 2.11 (归谬, *Reductio ad Absurdum*, *RAA*) [Enderton, pp.119]:

若 $\Gamma \cup \{\varphi\}$ 不一致, 则 $\Gamma \vdash \neg\varphi$

› 证明过程与命题逻辑的定理 1.30 类似



概括定理

定理 2.9 (概括定理) [Enderton, pp.117]:

若 $\Gamma \vdash \varphi$ 且 x 不在 Γ 中自由出现, 则 $\Gamma \vdash \forall x \varphi$

证明:

令 φ 存在一个证明 $\langle \alpha_0, \dots, \alpha_n = \varphi \rangle$, 利用对证明长度施归纳即可证明该定理。

- **奠基:** 当证明长度为 1 时, $\varphi \in \Gamma$, 根据公理 2.7.4 可知当 x 不在 φ 中自由出现时有 $\varphi \rightarrow \forall x \varphi$, 根据演绎定理可得 $\Gamma \vdash \forall x \varphi$



概括定理

定理 2.9 (概括定理) [Enderton, pp.117]:

若 $\Gamma \vdash \varphi$ 且 x 不在 Γ 中自由出现, 则 $\Gamma \vdash \forall x \varphi$

证明 (续):

- > **归纳:** 假设 $j < i$ 时对任意 j 有该结论成立, 下面对于 α_i 分情况讨论:
 - » $\alpha_i \in \Lambda$, 根据定义有 $\forall x \alpha_i$ 依然是逻辑公理, 显然有 $\Gamma \vdash \forall x \alpha_i$ (尽管 x 可能在 α_i 中出现, 但不影响结论)
 - » $\alpha_i \in \Gamma$, 与奠基情况一致
 - » α_i 由 α_j 与 $\alpha_j \rightarrow \alpha_i$ 通过 MP 规则得到。由归纳假设有 $\Gamma \vdash \forall x \alpha_j$ 且 $\Gamma \vdash \forall x (\alpha_j \rightarrow \alpha_i)$ 。对公理 2.7.3

$$\forall x (\alpha_j \rightarrow \alpha_i) \rightarrow (\forall x \alpha_j \rightarrow \forall x \alpha_i)$$

运用两次 MP 规则即可得 $\Gamma \vdash \forall x \alpha_i$

Q.E.D.



概括定理

定理 2.9 (概括定理) [Enderton, pp.117]:

若 $\Gamma \vdash \varphi$ 且 x 不在 Γ 中自由出现, 则 $\Gamma \vdash \forall x \varphi$

- › 可见, 公理 2.7.3 与 2.7.4 存在的作用就是为了证明概括定理
 - » **2.7.3** $\forall x (\alpha \rightarrow \beta) \rightarrow (\forall x \alpha \rightarrow \forall x \beta)$
 - » **2.7.4** $\alpha \rightarrow \forall x \alpha$, 其中 x 不在 α 中自由出现
- › 概括定理在有些逻辑系统中被作为推理规则 (如 Gentzen 的 LK 自然演绎系统等)
 - » 若能对 x 不进行假设 (约束) 即可证明命题 $__x__$, 那么可以说“由于 x 的任意性, 我们有 $\forall x __x__$ 成立”
- › 常见的用法: “*without loss of generality*” (不失一般性, WLOG)
 - » 用特例代替一般性推理, 最后概括 (同时使用公理 2.7.2 与概括定理)



另一套公理系统

定义 2.7' (一阶逻辑的公理系统, *Axiomatic system for FOL*) [Open Logic, section 10.1]:

公理 (Λ):

1. 命题逻辑公理 (定义 1.22) 的 FOL 概括
2. 关于量词的公理: 对闭项 (*closed term*, 即不含变元的项) 有
 - » $\forall x \beta \rightarrow \beta_t^x$
 - » $\beta(t) \rightarrow \exists x \beta$ [Enderton, pp. 124] (*Rule EI*)

推理规则:

1. MP 规则
2. 关于量词的推理规则 (QR):
 - » 若 $\beta \rightarrow \alpha(a)$ 已经出现在证明序列中, 且 a 不在 $\Gamma \cup \{\beta\}$ 中出现, 那么 $\beta \rightarrow \forall x \beta(x)$ 是正确的
 - » 若 $\alpha(a) \rightarrow \beta$ 已经出现在证明序列中, 且 a 不在 $\Gamma \cup \{\beta\}$ 中出现, 那么 $\exists x \beta(x) \rightarrow \beta$ 是正确的



概括定理

例:

1. $\{\forall x (Px \rightarrow Qx), \forall z Pz\} \vdash Qc$, 显然
2. $\{\forall x (Px \rightarrow Qx), \forall z Pz\} \vdash Qy$, 和1类似
3. $\{\forall x (Px \rightarrow Qx), \forall z Pz\} \vdash \forall y Qy$, 不失一般性
4. $\{\forall x (Px \rightarrow Qx), \forall z Pz\} \vdash \forall x Qx$, 约束变元换名



常数概括定理

推论 2.13 (常数概括, *Generalisation on Constants*) [Enderton, pp.123]:

假设 $\Gamma \vdash \varphi$ 且 c 是一个不在 Γ 中出现的常数符号。则存在变元 y , y 不在 φ 中出现, 使得 $\Gamma \vdash \varphi_y^c$ 成立。更进一步, 存在一个从 Γ 到 $\forall y \varphi_y^c$ 不含 c 的推演

证明:

令 $\langle \alpha_0, \dots, \alpha_n = \varphi \rangle$ 是从 Γ 到 φ 的一个证明, 令 y 为第一个不在任意 $\alpha_i, i \in \{0, \dots, n\}$ 中出现的变元。那么可以断言

$$\langle (\alpha_0)_y^c, \dots, (\alpha_n)_y^c \rangle$$

是一个从 Γ 到 φ_y^c 的一个推演。为证明这个断言, 需验证每个 $(\alpha_k)_y^c$ 要么属于 $\Gamma \cup \Lambda$, 要么根据 MP 规则从 $\{(\alpha_0)_y^c, \dots, (\alpha_{k-1})_y^c\}$ 中推导而来。



常数概括定理

推论 2.13 (常数概括, *Generalisation on Constants*) [Enderton, pp.123]:

假设 $\Gamma \vdash \varphi$ 且 c 是一个不在 Γ 中出现的常数符号。则存在变元 y , y 不在 φ 中出现, 使得 $\Gamma \vdash \varphi_y^c$ 成立。更进一步, 存在一个从 Γ 到 $\forall y \varphi_y^c$ 不含 c 的推演

证明 (续):

下面对未作替换的 α_k 分情况讨论:

1. $\alpha_k \in \Gamma$: 因此 c 不在 α_k 中出现, 可知 $(\alpha_k)_y^c = \alpha_k \in \Gamma$
2. $\alpha_k \in \Lambda$: 即 α_k 为定义 2.7 中的逻辑公理, 那么在 $(\alpha_k)_y^c$ 依然是一条逻辑公理 (Why?), 因此 $(\alpha_k)_y^c \in \Lambda$
 - » 若 α_k 是命题逻辑中的重言式, 替换其中常量得到的仍然是重言式。例如 $Pc \rightarrow \neg\neg Pc$, 显然有 $(Pc \rightarrow \neg\neg Pc)_y^c = Py \rightarrow \neg\neg Py$, 它仍是逻辑公理
 - » 若 α_k 是 $\forall x \psi \rightarrow \psi_t^x$, 那么 $(\alpha_k)_y^c$ 就是 $\forall x \psi_y^c \rightarrow (\psi_t^x)_y^c$ 。注意到 $(\psi_t^x)_y^c$ 正是 $(\psi_y^c)_{t_y^c}^x$, 所以它也是逻辑公理。其它组的公理也容易验证
3. α_k 是由 α_i 与 $\alpha_j = (\alpha_i \rightarrow \alpha_k)$ ($i, j < k$) 用 MP 规则推出: 那么有 $(\alpha_j)_y^c = ((\alpha_i)_y^c \rightarrow (\alpha_k)_y^c)$, 因此可知 $(\alpha_k)_y^c$ 可由 $(\alpha_i)_y^c$ 与 $(\alpha_j)_y^c$ 经过 MP 规则导出



常数概括定理

推论 2.13 (常数概括, *Generalisation on Constants*) [Enderton, pp.123]:

假设 $\Gamma \vdash \varphi$ 且 c 是一个不在 Γ 中出现的常数符号。则存在变元 y , y 不在 φ 中出现, 使得 $\Gamma \vdash \varphi_y^c$ 成立。更进一步, 存在一个从 Γ 到 $\forall y \varphi_y^c$ 不含 c 的推演

证明 (续) :

由上面的推导可知, $\langle (\alpha_0)_y^c, \dots, (\alpha_n)_y^c \rangle$ 的确是 φ_y^c 的一个推演。

令 Φ 为 $\langle (\alpha_0)_y^c, \dots, (\alpha_n)_y^c \rangle$ 中所有 $\alpha_i \in \Gamma$ 构成的集合。

- › 显然 Φ 是一个有穷集合, 有 y 不在 Φ 中出现且 $\Phi \vdash \varphi_y^c$
- › 根据概括定理, $\Phi \vdash \forall y \varphi_y^c$, 所以 $\Gamma \vdash \forall y \varphi_y^c$

注意到概括定理的证明中不会引入新的常元, 这样就可以得到一个从 Γ 到 $\forall y \varphi_y^c$ 的证明, 且 c 不出现在其中。
Q.E.D.



常数概括定理应用的例子

任意柯西序列 $\langle a_n \rangle$ 有界。

$$\exists M \geq 0 \forall n \in \mathbb{N}, |a_n| \leq M$$

证明：

1. 柯西列的定义：对任意 $\epsilon > 0$ ，存在 $N \in \mathbb{N}$ 令 $\forall m, n \geq N, |a_n - a_m| < \epsilon$
2. 令 $\epsilon = 1$ ，由于 $\langle a_n \rangle$ 是柯西列，那么存在 $N \in \mathbb{N}$ 令 $\forall m, n \geq N$ 有 $|a_n - a_m| < 1$
3. **常数引入**：不失一般性，我们令 $n = c$ ，且令 $m = N$ ，那么有 $|a_c - a_N| < 1$
4. **消去全称量词后**，根据三角不等式有 $|a_c| < |a_N| + 1$
5. **概括**：因为 c 是任意选取的，故定理对任意 $n \geq N$ 均成立，即序列 $\langle a_n \rangle$ 有界



约束变元替换

我们想证：

$$\vdash \forall x \forall y Pxy \rightarrow \forall y Pyy$$

由于 $\forall x \forall y Pxy$ 中的 x 不能用 y 进行替换，所以它无法套用公理 2.7.2。

但如果我们有

$$\vdash \forall x \forall z Pxz \rightarrow \forall y Pyy$$

显然就会好证很多。

那么只需要有

$$\vdash \forall x \forall y Pxy \rightarrow \forall x \forall z Pxz$$

就能证明最开始的结论。



ALPHABETIC VARIANTS

定理 2.14 (约束变元替换定理, *Alphabetic Variants*) [Enderton, pp.126]:

令 φ 是一个 wff, t 是一个项, x 是一个变元。总可以找到一个 wff φ' , 它和 φ 的差别仅在于约束变元, 使得

1. $\varphi \vdash \varphi'$ 且 $\varphi' \vdash \varphi$
2. t 可以在 φ' 中无冲突地替换 x

该定理的目的是当 t 无法被用来替换 x 时, 可以通过对约束变元换名来实现合法替换。

证明概要:

1. 不失一般性, 可固定 t 和 x 并递归地从 φ 构造 φ' , 通过结构归纳进行证明
 - » 归纳假设为: 应用构造算子前有 $\varphi \vdash \varphi'$ (或 $\varphi' \vdash \varphi$)
2. 无量词情况较简单; 对于有量词的情况构造 $(\forall y \varphi)' = \forall z (\varphi')_z^y$, 其中 z 不在 φ' , x 和 t 中出现, 这时 t 便可用来替换其中的 x (定理结论 2 成立)
3. 运用概括定理易证定理结论 1 成立



与等词有关的元定理

1. $\vdash \forall x (x = x)$
2. $\vdash \forall x \forall y (x = y \rightarrow y = x)$
3. $\vdash \forall x \forall y \forall z (x = y \rightarrow (y = z \rightarrow x = z))$
4. $\vdash \forall x_1 \forall x_2 \forall y_1 \forall y_2 (x_1 = y_1 \rightarrow (x_2 = y_2 \rightarrow (Px_1x_2 = Py_1y_2)))$, P 为一二元谓词
5. $\vdash \forall x_1 \forall x_2 \forall y_1 \forall y_2 (x_1 = y_1 \rightarrow (x_2 = y_2 \rightarrow (fx_1x_2 = fy_1y_2)))$, f 为一二元函数



语义



回顾：命题逻辑中的赋值

定义 1.12 (truth assignment) :

对于命题符号集合 S ，一个真值指派 v 是一个函数

$$v : S \rightarrow \{F, T\}$$

定义 1.12' (truth assignment, extended) :

1. v 是一个真值指派 (赋值) 指它是一个函数 $v : S \rightarrow \{F, T\}$ ，从而对于任何命题符号 $A_i \in S$ ， $v(A_i)$ 为 T 或 F
2. 对于任何真值指派 v ，定义 $\bar{v} : \bar{S} \rightarrow \{F, T\}$ 如下
 - » $\bar{v}(A_i) = v(A_i), i \in \mathbb{N}$
 - » $\bar{v}(\neg\alpha) = B_{\neg}(\bar{v}(\alpha))$
 - » $\bar{v}(\alpha \square \beta) = B_{\square}(\bar{v}(\alpha), \bar{v}(\beta)), \square \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$

目标：先为原子公式赋值，再为每个 wff 生成唯一的真值指派



FOL中的赋值

与“天生自带真值”的命题不同，在FOL中我们需要一些**来自外部的信息**才能达成以上目标：

1. 确定一些**对象**（*objects*），至少要有 I 个（因为有些公理对论域为空的语言不成立），作为被量词所量化的定义**论域**（*domain of discourse*）
2. 为论域中的每个对象指派一个FOL语言中的**正式名字**（即常元符号），作为对它的解释和索引（它与论域中的元素一一对应，自然也要非空）
3. 为论域中的每个 k -元谓词/函数也指派一个FOL语言中的**正式名字**（谓词符号/函数符号）。这些符号对应于论域中一些 k -元组（对于函数则是 $k + 1$ -元组）构成的集合（可能是空集），本质上是谓词与函数的**外延**（*extension*），它们为这个**外部结构**提供基础的**真值**

» “*extension*”的定义详见ZFC中的“外延公理”



定义 2.15 (结构, *Structure*) [Enderton, pp.80]:

FOL的一个结构 $\mathfrak{A}$ 是一个函数, 它的定义域是FOL的量词符号及非逻辑符号 (*signature*), 并且满足下列条件:

1. $\mathfrak{A}$ 为量词 $\forall$ 指派一个非空集合 $|\mathfrak{A}|$ 作为论域 (*domain* 或 *universe*)
2. $\mathfrak{A}$ 为每个 n -元谓词符号 P 指派一个 n -元关系 $P^{\mathfrak{A}} \subseteq |\mathfrak{A}|^n$
3. $\mathfrak{A}$ 为每个常元符号 c 指派一个 $|\mathfrak{A}|$ 中的元素 $c^{\mathfrak{A}}$
4. $\mathfrak{A}$ 为每个 n -元函数符号 f 指派一个 n -元函数 $f^{\mathfrak{A}} : |\mathfrak{A}|^n \rightarrow |\mathfrak{A}|$



结构的例子

在集合论的FOL语言中，定义一个结构 $\mathfrak{A}$ 如下：

- › $|\mathfrak{A}|$ 为自然数集合
- › $\in^{\mathfrak{A}}$ 为二元组集合 $\{\langle m, n \rangle \mid m < n\}$

一个该语言中语句为

$$\exists x \forall y \neg y \in x$$

由于它在 $\mathfrak{A}$ 中为真，我们称 $\mathfrak{A}$ 是它的一个模型（*model*），记为

$$\models_{\mathfrak{A}} \exists x \forall y \neg y \in x$$

下面的语句在 $\mathfrak{A}$ 下应该如何解释？ $\mathfrak{A}$ 是它的模型吗？

$$\forall x \forall y \exists z \forall t (t \in z \leftrightarrow t = x \vee t = y)$$



我们在何种情况下才能说一个一般的 FOL wff 为**真**？

- › 显然我们在自由变元没有赋值时无法下定论，因为它**不是命题**。例如 $\varphi \equiv x \in 10$

因此，我们希望在给定一个变元赋值 $s : V \rightarrow |\mathfrak{A}|$ 时，

$$\models_{\mathfrak{A}} \varphi[s]$$

当且仅当将 φ 中的自由变元 x 替换为 $s(x) = c^{\mathfrak{A}} \in |\mathfrak{A}|$ 后，再按照 $\mathfrak{A}$ 的规定进行“翻译”，所得的语句为**真**

- › 例如 $\varphi_5^x = 5 \in 10$

定义 2.16 (解释, *Interpretation*) [Enderton, pp.83; Hao et.al., 90]:

令 $\mathfrak{A}$ 是 FOL 语言的结构, $s : V \rightarrow |\mathfrak{A}|$ 是一个将变元映射到论域的函数。我们定义 $\mathfrak{A}$ 和 s 满足 (*satisfy*) wff φ , 记为 $\models_{\mathfrak{A}} \varphi[s]$ (或者 $(\mathfrak{A}, s) \models \varphi$), 递归定义如下:

› **项的解释**: 把对变元的赋值 s 扩展为对所有项的赋值。令 T 表示所有项构成的集合, 递归定义项的赋值函数 $\bar{s} : T \rightarrow |\mathfrak{A}|$ 如下

1. 对每一个变元符号 x , $\bar{s}(x) = s(x)$
2. 对每一个常元符号 c , $\bar{s}(c) = c^{\mathfrak{A}}$
3. 若 $t_1, \dots, t_n$ 是项, 且 f 是一个 n -元函数符号, 那么

$$\bar{s}(ft_1, \dots, t_n) = f^{\mathfrak{A}}(\bar{s}(t_1), \dots, \bar{s}(t_n))$$

定义 2.16 (解释, *Interpretation*) [Enderton, pp.83; Hao et.al., 90]:

令 $\mathfrak{A}$ 是 FOL 语言的结构, $s : V \rightarrow |\mathfrak{A}|$ 是一个将变元映射到论域的函数。我们定义 $\mathfrak{A}$ 和 s 满足 (*satisfy*) wff φ , 记为 $\models_{\mathfrak{A}} \varphi[s]$ (或者 $(\mathfrak{A}, s) \models \varphi$), 递归定义如下:

› **原子公式的解释:** 由于原子公式的定义是非递归的, 因此它的解释的定义也是非递归的

1. 对于特殊的谓词“=”, 我们有 $\models_{\mathfrak{A}} t_1 t_2$ 当且仅当 $\bar{s}(t_1) = \bar{s}(t_2)$ (注意, 这两个 = 分别处在 FOL 和元语言中)
2. 对其他每个 n -元谓词符号 P 有

$$\models_{\mathfrak{A}} P t_1 \dots t_n \quad \text{iff} \quad (\bar{s}(t_1), \dots, \bar{s}(t_n)) \in P^{\mathfrak{A}}$$

定义 2.16 (解释, *Interpretation*) [Enderton, pp.83; Hao et.al., 90]:

令 $\mathfrak{A}$ 是 FOL 语言的结构, $s : V \rightarrow |\mathfrak{A}|$ 是一个将变元映射到论域的函数。我们定义 $\mathfrak{A}$ 和 s 满足 (*satisfy*) wff φ , 记为 $\models_{\mathfrak{A}} \varphi[s]$ (或者 $(\mathfrak{A}, s) \models \varphi$), 递归定义如下:

‣ **其它 wff 的解释:** 根据 wff 的结构, 递归定义如下:

1. 若为原子公式, 其定义如上
2. $\models_{\mathfrak{A}} \neg \varphi[s]$ 当且仅当 $\not\models_{\mathfrak{A}} \varphi[s]$
3. $\models_{\mathfrak{A}} (\varphi \rightarrow \psi)[s]$ 当且仅当 $\not\models_{\mathfrak{A}} \varphi[s]$ 或 $\models_{\mathfrak{A}} \psi[s]$; 或者以上两者均成立
4. $\models_{\mathfrak{A}} \forall x \varphi[s]$ 当且仅当对任意 $d \in |\mathfrak{A}|$, 我们有 $\models_{\mathfrak{A}} \varphi[s(x \mid d)]$, 其中 $s(x \mid d)$ 定义如下:

$$s(x \mid d)(y) = \begin{cases} s(y), & \text{if } y \neq x, \\ d, & \text{if } y = x. \end{cases}$$



例子 [SMITH, PP.345]

定义一个FOL语言如下:

- › 常元: m, n
- › 一元谓词: F, G
- › 二元谓词: L

考察下列语句的解释与真值:

1. $(\exists x Lmx \rightarrow Lmn)$
2. $\forall x (Gx \rightarrow (Lxm \vee \neg Lmx))$
3. $\forall x (Gx \rightarrow \exists y Lxy)$

对应的结构 $\mathfrak{A}$ 如下:

- › $|\mathfrak{A}|$: {Romeo, Juliet, Benedick, Beatrice}
- › 常元指派:
 - » m : Romeo
 - » n : Juliet
- › 谓词指派:
 - » F : {Romeo, Benedick}
 - » G : {Juliet, Beatrice}
 - » L : { $\langle \text{Romeo, Juliet} \rangle$, $\langle \text{Juliet, Romeo} \rangle$,
 $\langle \text{Benedick, Beatrice} \rangle$, $\langle \text{Beatrice, Benedick} \rangle$,
 $\langle \text{Benedick, Benedick} \rangle$ }



例子 [SMITH, PP.345]

定义一个FOL语言如下:

- 常元: m, n
- 一元谓词: F, G
- 二元谓词: L

考察下列语句的解释与真值:

1. $(\exists x Lmx \rightarrow Lmn)$
2. $\forall x (Gx \rightarrow (Lxm \vee \neg Lmx))$
3. $\forall x (Gx \rightarrow \exists y Lxy)$

对应的结构 $\mathfrak{A}$ 如下:

- $|\mathfrak{A}|: \{4, 7, 8, 11, 12\}$
- 常元指派:
 - » $m: 7$
 - » $n: 12$
- 谓词指派:
 - » $F: |\mathfrak{A}|$ 中的全部偶数
 - » $G: |\mathfrak{A}|$ 中的全部奇数
 - » $L: |\mathfrak{A}|$ 中所有满足 $m < n$ 的有序对 $\langle m, n \rangle$



当 $\mathcal{WFF}$ 中拥有自由变元时

定理 2.17 [Enderton, pp.86]:

若 s_1 与 s_2 是两个从 V 到 $|\mathfrak{A}|$ 的变量赋值函数，它们在 $\mathcal{WFF}$ φ 所有自由变元（如果有的话）上的取值相等，那么

$$\models_{\mathfrak{A}} \varphi[s_1] \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi[s_2]$$

推论 2.18 [Enderton, pp.86]: 对于一个语句 σ ，下列结论中必有一条成立：

1. $\mathfrak{A}$ 对于任意变元赋值函数 s 均满足 σ ，这时我们称 $\mathfrak{A}$ 是 σ 的**模型**（*model*）
2. $\mathfrak{A}$ 对于任意变元赋值函数 s 均不满足 σ

证明思路： 由于 $\mathcal{WFF}$ 的可满足性是归纳定义的，直接使用结构归纳证明即可。



对于语句

$$\exists x (x \cdot x = 1 + 1)$$

- › 实数域 $\mathfrak{R} = (\mathbb{R}; 0, 1, +, \cdot)^\dagger$ 是它的模型
- › 有理数域 $\mathfrak{Q} = (\mathbb{Q}; 0, 1, +, \cdot)$ 不是它的模型
- › $\exists, \wedge, \vee, \leftrightarrow$ 的解释与它们在命题逻辑中的语义一样 [Enderton, pp.87]

$^\dagger \mathfrak{A} = (|\mathfrak{A}|; c_1^{\mathfrak{A}}, \dots, P_1^{\mathfrak{A}}, \dots, f_1^{\mathfrak{A}}, \dots)$ 是一种非正式记法，只有在不引起歧义的时候才能不加标注地使用



FOL 的逻辑有效性/逻辑蕴涵

定义 2.19 (逻辑蕴涵) [Enderton, pp.88]:

令 Γ 为 wff 集合, φ 是一个 wff。那么我们说 Γ **逻辑蕴涵** (*logically implies, entails*) φ 当且仅当对任意结构 $\mathfrak{A}$ 和任意赋值 $s : V \rightarrow |\mathfrak{A}|$ 均有 $\Gamma \models_{\mathfrak{A}} \varphi[s]$ 。这时我们记为

$$\Gamma \models \varphi$$

1. 这里的**逻辑蕴涵**使用的 (元语言) 符号 $\models$ 与第二章《命题逻辑》里一模一样。由于一阶逻辑包含 (*subsumes*) 命题逻辑 (只需允许 $\circ$ 元谓词存在), 因此逻辑蕴涵包含重言蕴涵, 也更接近我们在第一章《非形式化逻辑》里的定义

定义: 我们称一个推演步骤是**有效的**当且仅当不存在任何一种可能, 令该推演的前提为真且结论为假。同样地, 在这种情况下我们称这些前提**蕴涵** (*entails*) 其结论。

2. 与之前的定义一样:
 - » 记**逻辑等价** (*logically equivalent*) 为 $\gamma \models \varphi$ 或者 $\gamma \simeq \varphi$
 - » 若 $\emptyset \models \varphi$ 则称 φ 是 (逻辑) 有效的 (*valid*), 简记为 $\models \varphi$



FOL 的逻辑有效性/逻辑蕴涵

定义 2.19 (逻辑蕴涵) [Enderton, pp.88]:

令 Γ 为 wff 集合, φ 是一个 wff。那么我们说 Γ **逻辑蕴涵** (*logically implies, entails*) φ 当且仅当对任意结构 $\mathfrak{A}$ 和任意赋值 $s : V \rightarrow |\mathfrak{A}|$ 均有 $\Gamma \models_{\mathfrak{A}} \varphi[s]$ 。这时我们记为

$$\Gamma \models \varphi$$

推论 2.20 (语句的逻辑蕴涵) [Enderton, pp.88]:

对一个语句集合 Σ 和语句 σ , $\Sigma \models \sigma$ 当且仅当 Σ 的任意一个模型都是 σ 的模型。我们称语句 σ 是**有效的**当且仅当在任意结构下 σ 均为真。



例子

1. $\forall v_1 Qv_1 \models Qv_2$
2. $\models \neg\neg\sigma \rightarrow \sigma$
3. $\models \exists x (Qx \rightarrow \forall x Qx)$
4. [Enderton, pp.99] $\Gamma \cup \{\alpha\} \models \varphi$ 当且仅当 $\Gamma \models (\alpha \rightarrow \varphi)$

» 证明:

$$\begin{aligned} & \Gamma \cup \{\alpha\} \models \varphi \\ \Leftrightarrow & \text{对任意 } \mathfrak{A} \text{ 和赋值 } s \text{ 有 } \models_{\mathfrak{A}} \Gamma \cup \{\alpha\}[s] \text{ 蕴涵 } \models_{\mathfrak{A}} \varphi[s] \\ \Leftrightarrow & \text{对任意 } \mathfrak{A} \text{ 和赋值 } s \text{ 有 } \models_{\mathfrak{A}} \Gamma[s] \text{ 蕴涵“若 } \models_{\mathfrak{A}} \alpha[s] \text{ 那么 } \models_{\mathfrak{A}} \varphi[s]” \\ \Leftrightarrow & \text{对任意 } \mathfrak{A} \text{ 和赋值 } s \text{ 有 } \models_{\mathfrak{A}} \Gamma[s] \text{ 蕴涵 } \models_{\mathfrak{A}} (\alpha \rightarrow \varphi)[s] \\ \Leftrightarrow & \Gamma \models (\alpha \rightarrow \varphi) \end{aligned}$$

Q.E.D.



更多例子

证明以下结论[Enderton, pp.99]:

$$\{\forall x (\alpha \rightarrow \beta), \forall x \alpha\} \vdash \forall x \beta$$

证明:

1. 根据定义, 对任意使得 $\models_{\mathfrak{A}} (\alpha \rightarrow \beta)[s]$ 和 $\models_{\mathfrak{A}} \alpha[s]$ 成立的 $\mathfrak{A}$ 和赋值 s , 我们有:
 - » 对任意 $d \in |\mathfrak{A}|$ 有 $\models_{\mathfrak{A}} (\alpha \rightarrow \beta)[s(x \mid d)]$ 且 $\models_{\mathfrak{A}} \alpha[s(x \mid d)]$
2. 根据 $\rightarrow$ 符的解释, 对任意 $d \in |\mathfrak{A}|$ 有 $\models_{\mathfrak{A}} (\alpha \rightarrow \beta)[s(x \mid d)]$ 当且仅当:
 - » $\not\models_{\mathfrak{A}} \alpha[s(x \mid d)]$ 或者 $\models_{\mathfrak{A}} \beta[s(x \mid d)]$; 要么这二者均成立。因此一共有3种情况:
 - » $\models_{\mathfrak{A}} \alpha[s(x \mid d)]$ 且 $\models_{\mathfrak{A}} \beta[s(x \mid d)]$
 - » $\not\models_{\mathfrak{A}} \alpha[s(x \mid d)]$ 且 $\models_{\mathfrak{A}} \beta[s(x \mid d)]$
 - » $\not\models_{\mathfrak{A}} \alpha[s(x \mid d)]$ 且 $\not\models_{\mathfrak{A}} \beta[s(x \mid d)]$
3. 我们在第一步中得知, 对任意 $d \in |\mathfrak{A}|$ 有 $\models_{\mathfrak{A}} \alpha[s(x \mid d)]$
 - » 那么必然有 $\models_{\mathfrak{A}} \beta[s(x \mid d)]$, 即 $\models_{\mathfrak{A}} \forall x \beta[s]$

Q.E.D.



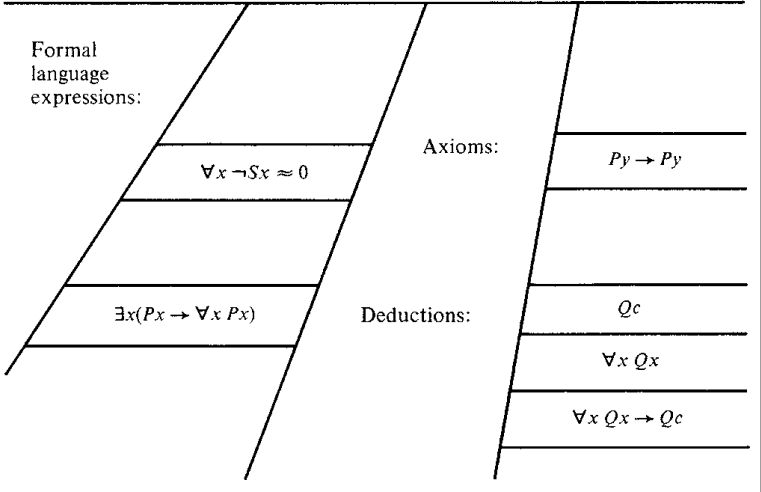
The study, in English, of the view below:

If $\Gamma \models \varphi$, then $\Gamma \vdash \varphi$.

$\Gamma; \alpha \vdash \beta \wedge \neg \beta \Rightarrow \Gamma \vdash \neg \alpha$

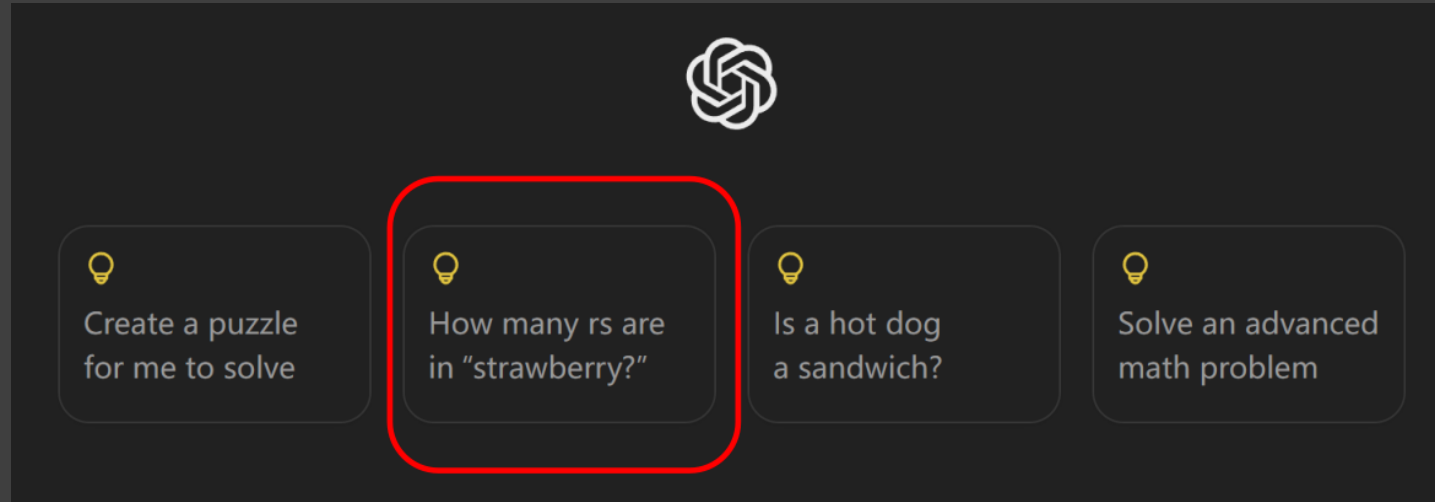
$\models_{\mathfrak{R}} \forall x \neg Sx \approx 0$

$\models \exists x(Px \rightarrow \forall x Px)$





问题求解中的一阶推理



How many vowels are there in the word
“Gooooooooooooooooooooooooooooooooal”?



How many vowels are there in the word
"Gooooooooooooooooooooooooooooooooal"?

您正在提供关于 ChatGPT 新版本的反馈。
您更偏向于哪个回复？回复可能需要一点时间来加载。

 回复 1

Thought, 持续 15 秒 

To determine the number of vowels in the word
"Gooooooooooooooooooooooooooooooooal," let's break it down:

- Total Letters in the Word:** There are 34 letters in total.
 - G (1 letter)
 - 31 O's (from counting each 'o' between 'G' and 'a')
 - A (1 letter)
 - L (1 letter)
- Identifying the Vowels:** Vowels are A, E, I, O, U.
 - O's: There are 31 O's.
 - A: There is 1 A.
 - Total Vowels: $31 O's + 1 A = 32 \text{ vowels}$
- Final Answer:** 32 vowels in the word.

Answer: 32

我更喜欢这个回复

 回复 2

Thought, 持续 8 秒 

To determine the number of vowels in the word
"Gooooooooooooooooooooooooooooooooal", let's break it down:

- Identify the Vowels:**
 - The standard vowels in the English alphabet are A, E, I, O, U.
- Analyze the Word:**
 - The word is composed of:
 - G (a consonant)
 - A long string of "o" characters
 - A (a vowel)
 - L (a consonant)
- Count the Vowels:**
 - "o" appears 30 times in the word.
 - "A" appears once.
- Total Vowels:**
 - $30 \text{ (from "o")} + 1 \text{ (from "a")} = 31 \text{ vowels}$

Therefore, the word "Gooooooooooooooooooooooooooooooooal" contains 31 vowels.

我更喜欢这个回复



#trial	answer	#trial	answer
1	30 ✓	6	31 ✗
2	31 ✗	7	31 ✗
3	31 ✗	8	29 ✗
4	33 ✗	9	Choose between 32 ✗ and 31 ✗
5	24 ✗	10	29 ✗



O3-MINI 与 O3-MINI-HIGH

ChatGPT o3-mini

共享

How many vowels are there in the word "Gooooooooooooooooooooooooooooal"?

已推理, 持续 11 秒

I'm trying to count the vowels in the word "Gooooooooooooooooooooooooooooal." I start by identifying the letters: "G", then many "o"s, and finally "a" and "l."

I know "G" isn't a vowel, but "o" is, and "a" is also a vowel. So I focus mainly on the "o"s. Let's see — we have one "G," followed by 30 "o"s, then "a" and "l." That gives me 31 vowels in total.

The word "Gooooooooooooooooooooooooooooal" consists of a starting "G", followed by 25 "o"s, then the letters "a" and "l". The vowels in this word are the 25 "o"s and the "a", which gives a total of **26 vowels**.

询问任何问题

+

搜索

深入研究

?

ChatGPT 也可能犯错。请核查重要信息。

How many vowels are there in the word "Gooooooooooooooooooooooooooooal"?

已推理, 持续 1m 55s

Okay, the question asks to count vowels in the word "Gooooooooooooooooooooooooooooal." Let's break it down: It starts with a G (not a vowel), then many o's (which are vowels), and ends with 'al', where 'a' is a vowel, and 'l' is not.

So, I'll count each "o" and the "a". That gives me the vowels: "o" repeated a lot, plus the "a" at the end. I'll make sure to count them all!

Okay, so the word "Gooooooooooooooooooooooooooooal" has vowels: 'o' and 'a'. The letter "G" isn't a vowel, and "l" isn't either. Let's see how many "o"s there are.

I will count carefully: The word starts with "G", followed by 31 "o"s, then ends with "al".

So, after the "G", I count 31 "o"s, plus the "a" at the end—that's 32 vowels in total.

Alright, the word is "Gooooooooooooooooooooooooooooal", and I need to count the 'o's carefully. Starting with "G", then it's mostly 'o's followed by "a" and "l" at the end. If I break up the 'o's into groups, I get 5 groups of 5 'o's and one final group of 3 'o's.

So, $5 + 5 + 5 + 5 + 3 = 23$ 'o's. Don't forget the 'a' and 'l', but the vowels so far are exactly 23 'o's. Keep counting to stay accurate!

Let's carefully count the vowels in the word "Gooooooooooooooooooooooooooooal". First, "G" is not a vowel; then there are a series of "o"s followed by "a" and "l" at the end.

I can break it up into groups: "ooooo" (5), "ooooo" (10), "ooooo" (15), "ooooo" (20) and "ooo" (23). Therefore, those 23 "o"s plus the "a" make 24 vowels in total. So, the answer seems to be 24 vowels.

Let's count the vowels in the string "Gooooooooooooooooooooooooooooal". By grouping, I see 10 "o"s in the first block, 10 in the second block, and 5 in the third, giving 25 "o"s total. Plus the "a" at the end, that makes 26 vowels.

So, the total number of vowels in this string is 26. It fits with the total length of the string if I also account for the other letters.

Let's break down the word:

- The word is G + ooooooooooooooooooooooooooooo + al.
- The only vowels here are the "o" and "a".
- Counting the letters:
 - There are 25 "o"s.
 - There is 1 "a".
- Total vowels = 25 + 1 = 26 vowels.

?

Okay, let's see. The user is asking how many vowels are in the word "Gooooooooooooooooooooooooooooal". Hmm, vowels in English are A, E, I, O, U, and sometimes Y. But in this case, the word is "Goal" with a bunch of O's in the middle. Let me break it down.

First, let's write out the word properly. It starts with a G, then a lot of O's, and ends with an L. So the structure is G followed by multiple O's and then AL. Wait, no, the original word is "Goal", right? But here it's written as "Gooooo...al". So maybe it's G, then a bunch of O's, then A and L. Wait, but in the given word, after the O's, is there an A? Let me check again. The user wrote "Gooooooooooooooooooooooooooooal". Let me count the letters after G. Let's see: G followed by a lot of O's and ending with "al". So maybe the word is structured as G + multiple O's + A + L. So the letters would be G, O, O, ..., O, A, L.

So the vowels here would be all the O's and the A. The Y isn't present here, so we don't have to consider that. Let's confirm the

SUBTIZING (基于直觉的数量感知)



“ **Subitizing** is the rapid, accurate, and effortless ability to perceive small quantities of items in a set, *typically when there are four or fewer items*, **without relying on linguistic or arithmetic processes**. The term refers to the sensation of instantly knowing how many objects are in the visual scene when their number falls within the subitizing range. ”

[Wikipedia: Subitizing]





How many vowels are there in the word
"Goooooooooooooooooooooooooooooooooal"?

已思考 23 秒 ✓

- 这道题要求我数出"Goooooooooooooooooooooooooooooooooal"中的元音字母。这个单词以字母G开头，接着是22个o，再加上"al"。我得核实一下o的数量，确认每个部分的字符数。要准确数出o的数量并确保没漏掉才能回答用户的问题，所以我会先认真确认每个字母的数量，避免犯错。
- 用户给出的是单词"Goooooooooooooooooooooooooooooooooal"。首先，我需要确认有多少个o作为元音，又有一个a。通过计算，o的数量是从第二个字符到倒数第二个字符之间的o。总共有23个o，再加一个a，这样就得到了元音的总数。为了确保准确，我会逐个数出这些字符，避免遗漏。
- 我在检查元音数量时尝试了两次计数。"Goooooooooooooooooooooooooooooooooal" 中，o的位置是从2到24，总共有23个o，再加上一个a和l。这意味着24个元音：23个o加1个a。通过仔细计算，我确认可以将o的位置准确定位，并避免遗漏任何字符。最终的结果是24个o，再加上一个a，总共25个元音。



一般问题（亚符号表征）→ 形式化（符号表征）→ 推理 / 计算



Count the apples and oranges in this image

已思考 1m 24s ▾

已分析

```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
from skimage import measure
# Load image
img = Image.open('/mnt/data/69374645-db78-4fa2-ba64-137ce108addf.png')
```



一般问题（亚符号表征）→ **形式化（符号表征）**[†]→ 推理 / 计算

[†]往往需要人类参与，且必须要有预定义的符号语言（Domain Specific Language）

- › 符号主义AI、概率图模型、因果结构学习、程序合成.....
- › 神经符号AI、统计关系学习、概率逻辑程序.....



“... these approaches miss an important consequence of uncertainty in a world of things: *uncertainty about what things are in the world*. Real objects seldom wear unique identifiers or preannounce their existence like the cast of a play. In areas such as vision, language understanding, ..., *the existence of objects must be inferred from raw data* (pixels, strings, and so on) that contain no explicit object references.”

[Russell, 2015]

review articles



DOI:10.1145/2699411

Open-universe probability models show merit in unifying efforts.

BY STUART RUSSELL

Unifying Logic and Probability

PERHAPS THE MOST enduring idea from the early days of AI is that of a *declarative* system reasoning over explicitly represented knowledge with a general inference engine. Such systems require a formal language to describe the real world; and *the real world has things in it*. For this reason, classical AI adopted first-order logic—the mathematics of objects and relations—as its foundation.

The key benefit of first-order logic is its expressive power, which leads to concise—and hence learnable—models. For example, the rules of chess occupy 10^9 pages in first-order logic, 10^2 pages in propositional logic, and 10^{18} pages in the language of finite automata. The power comes from separating predicates from their arguments and quantifying over

those arguments: so one can write rules about $\text{On}(p, c, x, y, t)$ (piece p of color c is on square x, y at move t) without filling in each specific value for c, p, x, y , and t .

Modern AI research has addressed another important property of the real world—*pervasive uncertainty*—about both its state and its dynamics—using probability theory. A key step was Pearl’s development of *Bayesian networks*, which provided the beginnings of a formal language for probability models and enabled rapid progress in reasoning, learning, vision, and language understanding. The expressive power of Bayes nets is, however, limited. They assume a fixed set of *variables*, each taking a value from a fixed range; thus, they are a *propositional* formalism, like Boolean circuits. The rules of chess and of many other domains are beyond them.

What happened next, of course, is that classical AI researchers noticed the pervasive uncertainty, while modern AI researchers noticed, or remembered, that the world has things in it. Both traditions arrived at the same place: *the world is uncertain and it has things in it*. To deal with this, we have to *unify logic and probability*.

But how? Even the meaning of such a goal is unclear. Early attempts by Leibniz, Bernoulli, De Morgan, Boole, Peirce, Keynes, and Carnap (surveyed by Hailperin⁴² and Howson⁴³) involved attaching probabilities to logical sentences. This line of work influenced AI

» key insights

- First-order logic and probability theory have addressed complementary aspects of knowledge representation and reasoning: the ability to describe complex domains concisely in terms of objects and relations and the ability to handle uncertain information. Their unification holds enormous promise for AI.
- New languages for defining open-universe probability models appear to provide the desired unification in a natural way. As a bonus, they support probabilistic reasoning about the existence and identity of objects, which is important for any system trying to understand the world through perceptual or textual inputs.



可靠性



FOL的可靠性

定理 2.27 (可靠性定理, *Soundness Theorem*) [Enderton, pp.131]:

若 $\Gamma \vdash \varphi$, 则 $\Gamma \models \varphi$

若有下面这条引理, 根据蕴含连词的解释与逻辑蕴含的定义, 可靠性的证明将非常简单

引理 2.28[Enderton, pp.131]:

所有逻辑公理均是有效的, 即对任意 $\varphi \in \Lambda$ 均有 $\models \varphi$



FOL的可靠性

可靠性的证明（假设引理成立）：

根据证明的定义进行归纳证明：

I. **奠基**：证明长度为1时有以下两种情况：

- » $\varphi \in \Lambda$ 是逻辑公理，那么根据引理 2.28 有 $\vdash \varphi$ ，因此 $\Gamma \vdash \varphi$
- » $\varphi \in \Gamma$ ，显然有 $\Gamma \vdash \varphi$

2. **归纳**：假设证明长度小于 n 时可靠性定理成立，当证明长度为 n 时：

- » 若 $\varphi \in \Lambda \cup \Gamma$ ，则与奠基情况相同，否则
- » φ 从 ψ 与 $\psi \rightarrow \varphi$ 通过 MP 规则推导而来：根据归纳假设有 $\Gamma \vdash \psi$ 且 $\Gamma \vdash (\psi \rightarrow \varphi)$
 - » **注意**：这里我们使用的是定义 2.7 中定义的第一套公理系统，其中只有一条 MP 推理规则。对于有多条推理规则的系统的可靠性证明请见 [\[Open Logic; Hao et.al.\]](#) 对应章节
- » 又根据蕴含连词的解释可知 $\Gamma \vdash (\psi \rightarrow \varphi)$ 当且仅当 $\not\vdash \psi$ 或 $\vdash \varphi$
- » 前者与归纳假设中的 $\Gamma \vdash \psi$ 矛盾，因此有 $\vdash \varphi$

Q.E.D.



FOL公理的有效性

公理的概括：任意有效公式的概括仍然是有效的，即

$$\models \varphi \Rightarrow \models \forall x \varphi$$

证明：

φ 是有效的，当且仅当

- › 对任意结构 $\mathfrak{A}$ 和任意赋值函数 s 有 $\models_{\mathfrak{A}} \varphi[s]$ ，当且仅当
- › 对任意结构 $\mathfrak{A}$ 和任意赋值函数 s 均有：对任意 $d \in |\mathfrak{A}|$ 有 $\models_{\mathfrak{A}} \varphi[s(x \mid d)]$
 - » $(\Rightarrow)$ $s(x \mid d)$ 是一种赋值函数
 - » $(\Leftarrow)$ 由 d 的任意性，只需令 $d = s(x)$ 就有 $s(x \mid d) = s(x \mid s(x)) = s$

当且仅当

- › 对任意结构 $\mathfrak{A}$ 和任意赋值函数 s 有 $\models_{\mathfrak{A}} \forall x, \varphi[s]$ ，即 $\models \forall x \varphi$ 成立

Q.E.D.



FOL公理的有效性

第一组公理：命题逻辑中的重言式在FOL中是有效的。

证明：

若以下引理成立，那么只需取 $\Gamma = \emptyset$ 即可令以上结论成立。

引理[Enderton, pp.129]:

令 $\mathfrak{A}$ 是一个结构， $s : V \rightarrow |\mathfrak{A}|$ 是赋值函数。定义一个在所有素公式[†] (*prime formulas*) α 上的真值指派 v 如下

$$v(\alpha) = T \quad \text{iff} \quad \models_{\mathfrak{A}} \alpha[s]$$

那么对任意wff φ (不管是否为素公式) 均有

$$\bar{v}(\varphi) = T \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi[s]$$

进一步，若 Γ 重言蕴涵 φ ，那么 Γ 也逻辑蕴涵 φ

[†] 素公式指所有原子公式以及形如 $\forall x \alpha$ 的公式[Enderton, pp.114]



素公式：命题逻辑与一阶逻辑的桥梁

一阶逻辑的语法其实是对命题逻辑的细化，这可以体现在wff的分类上：

定义[Enderton, pp.114]:

1. 原子公式与形如 $\forall x \varphi$ 的wff被称为素公式 (prime formula)
 - » *Formulas with no deeper propositional structure*
2. 其他形式的wff为非素公式 (nonprime formula) 或复杂公式 (compound formula)

- › 复杂公式均可以基于素公式通过逻辑连词构造
- › 素公式本身可看作命题逻辑中的命题符号 (注意区分[Enderton]中的闭语句“*sentence*”和命题符“*sentence symbol*”)
- › 形式不同的素公式只能看作不同的命题符号
 - » 形式语言的公式在语法上只是一个字符串



重言式

我们不再将重言式简单地归类进命题逻辑或一阶逻辑，这仅仅取决于wff本身的形式。回顾以下定义：

定义 I.14 (tautologically implies, entails) :

Σ 重言蕴涵 τ (记为 $\Sigma \models \tau$) 当且仅当所有满足 Σ 的真值指派均满足 τ

> 当 $\Sigma = \emptyset$, 记为 $\emptyset \models \tau$ 或 $\models \tau$

» 意味着任意真值指派均满足 τ , 我们称 τ 为重言式或永真式 (tautology)

显然，只有真值指派才能判定一个wff是否是重言式。例如下面的例子 [Enderton, pp.115]

$$(\forall y \neg Py \rightarrow \neg Px) \rightarrow (Px \rightarrow \neg \forall y \neg Py)$$

就是一个重言式，其中有两个命题符（素公式）： $\forall y \neg Py$ 和 Px 。 $\forall x (Px \rightarrow Px)$ 和 $\forall x Px \rightarrow Pc$ 则不是重言式（尽管它们是有效的，其逻辑有效性源自于它们是公理而不是真值表上的验证）。



“重言蕴涵”与“可证”

既然 FOL wff 可以被看作命题逻辑公式，我们能够很直观地得到下面的结论

定理 2.29 [Enderton, pp.115]:

(FOL 中) $\Gamma \vdash \varphi$ 当且仅当 $\Gamma \cup \Lambda$ 重言蕴涵 φ

- › 该定理显然成立。只需要使用命题逻辑的紧致性（完全性的推论）即可
- › 注意重言蕴涵 φ 的前提（前件）是 $\Gamma \cup \Lambda$ ，它包含所有的 FOL 公理
 - » 这样的话，例如 $\varphi \equiv \forall x Px \rightarrow Px$ 虽然是非重言式，但由于 $\varphi \in \Lambda$ ，仍然有 $\Gamma \cup \Lambda$ 重言蕴涵 φ
- › 正因如此，我们必须证明 Λ 的有效性。而这个有效性不能再通过“重言蕴涵”（即真值指派）这样的命题逻辑工具验证，必须回到结构中去证明它们的“真”



FOL公理的有效性

第一组公理：命题逻辑中的重言式在FOL中是有效的。

证明：

若以下引理成立，那么只需取 $\Gamma = \emptyset$ 即可令以上结论成立。

引理[Enderton, pp.129]:

令 $\mathfrak{A}$ 是一个结构， $s : V \rightarrow |\mathfrak{A}|$ 是赋值函数。定义一个在所有素公式[†] (*prime formulas*) α 上的真值指派 v （注：下面这句话即意味着对于FOL， v 由 $\mathfrak{A}$ 和 s 决定）如下

$$v(\alpha) = T \quad \text{iff} \quad \models_{\mathfrak{A}} \alpha[s]$$

那么对任意wff φ （不管是否为素公式）均有

$$\bar{v}(\varphi) = T \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi[s]$$

进一步，若 Γ 重言蕴涵 φ ，那么 Γ 也逻辑蕴涵 φ

[†] 素公式指所有原子公式以及形如 $\forall x \alpha$ 的公式[Enderton, pp.114]



FOL公理的有效性

第一组公理：命题逻辑中的重言式在FOL中是有效的。

证明 (Cont'd) :

I. 对wff的结构进行归纳:

- » 对于每个可以看作命题的素公式, $\bar{v}(\alpha) = v(\alpha) = T$ 当且仅当 $\models_{\mathfrak{A}} \alpha[s]$
- » 对于 $\neg\varphi$ 形式的wff, $\bar{v}(\neg\varphi) = T$ 当且仅当 $v(\varphi) = F$
 - » 根据 $\neg$ 的解释以及归纳假设得 $\models_{\mathfrak{A}} \neg\varphi[s]$ 当且仅当 $\not\models_{\mathfrak{A}} \varphi[s]$
- » 对于形如 $\psi \rightarrow \varphi$ 的wff, $\bar{v}(\psi \rightarrow \varphi) = T$ 当且仅当 $v(\psi) = F$ 或者 $v(\varphi) = T$
 - » 根据 $\rightarrow$ 的解释以及归纳假设得 $\models_{\mathfrak{A}} (\psi \rightarrow \varphi)[s]$ 当且仅当 $\not\models_{\mathfrak{A}} \psi[s]$ 或者 $\models_{\mathfrak{A}} \varphi[s]$

2. 根据重言蕴含的定义, $\Gamma \vdash \varphi$ 意味着:

- » 对任意结构 $\mathfrak{A}$ 和赋值 s , 任意 $\psi \in \Gamma$ 均有若 $\bar{v}(\psi[s]) = T$ 那么必有 $\bar{v}(\varphi[s]) = T$.
- » 根据上面的结论, 即: 对任意结构 $\mathfrak{A}$ 和赋值 s , 任意 $\psi \in \Gamma$ 均有若 $\models_{\mathfrak{A}} \psi[s]$ 那么必有 $\models_{\mathfrak{A}} \varphi[s]$, 也就是 Γ 逻辑蕴含 φ Q.E.D.



FOL公理的有效性

以下公理均是有效的（证明留作习题）

- 第三组公理： $\models \forall(\alpha \rightarrow \beta) \rightarrow (\forall x \alpha \rightarrow \forall x \beta)$
- 第四组公理： $\models \alpha \rightarrow \forall x \alpha$ ，其中 x 不在 α 中自由出现
- 第五组公理： $\models x = x$
- 第六组公理： $\models (x = y) \rightarrow (\alpha \rightarrow \alpha')$ ，其中 α 为原子公式， α' 为对 α 中的 x 进行0次或多次替换后得到的wff



第二组公理的有效性

第二组公理: $\models \forall x \alpha \rightarrow \alpha_t^x$, 其中 t 对于 α 中的 x 是可替换的

先看一个简单的例子: $\forall x Px \rightarrow Pt$ 是有效的。假设

$$\models_{\mathfrak{A}} \forall x Px[s]$$

根据定义, 对任意 $d \in |\mathfrak{A}|$ 有

$$\models_{\mathfrak{A}} Px[s(x \mid d)]$$

我们可以令 $d = \bar{s}(t)$, 那么就有

$$\models_{\mathfrak{A}} Px[s(x \mid \bar{s}(t))]$$

根据谓词的解释, 也就是说

$$\bar{s}(t) \in P^{\mathfrak{A}}$$

也即 $\models_{\mathfrak{A}} Pt[s]$



第二组公理的有效性

我们希望将这种“先赋值，再替换”与“先替换，再赋值”的语义等价性扩张到更复杂的 wff 中，即

$$\models_{\mathfrak{A}} \varphi[s(x \mid \bar{s}(t))] \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi_t^x[s]$$

其中 t 可以替换 φ 中的 x 。这个性质可以被总结为下面的引理

引理 2.30 [Enderton, pp.133]:

对于一个结构 $\mathfrak{A}$ 和赋值 s ，对任意项 u ，令 u_t^x 为将 u 中的 x 替换为 t 后的结果，那么

$$\bar{s}(u_t^x) = \overline{s(x \mid \bar{s}(t))}(u)$$

换言之，一个替换既可以发生在项（语法）中，也可以发生在赋值（语义）里，且二者是等价的



替换引理

引理 2.30 [Enderton, pp.133]: $\bar{s}(u_t^x) = \overline{s(x \mid \bar{s}(t))}(u)$

证明：直接对项 u 的结构进行归纳。

I. 奠基：

- » 若 u 是一个常元或非 x 变元，那么上式退化为 $\bar{s}(u) = \bar{s}(u)$
- » 若 $u = x$ 则上式退化为 $\bar{s}(t) = \bar{s}(t)$

2. 归纳：假设结论对项构造算子施加次数小于 n 时成立

- » 那么 $u = fu_1 \cdots u_n = f(u_1, \dots, u_n)$ 时， $u_t^x = f((u_1)_t^x, \dots, (u_n)_t^x)$
- » 根据项的解释，有 $\bar{s}(u_t^x) = \bar{s}(f((u_1)_t^x, \dots, (u_n)_t^x))$
- » 在 $\mathfrak{A}$ 中即 $f^{\mathfrak{A}}(\bar{s}((u_1)_t^x), \dots, \bar{s}((u_n)_t^x))$
- » 根据归纳假设，即 $f^{\mathfrak{A}}(\overline{s(x \mid \bar{s}(t))}(u_1), \dots, \overline{s(x \mid \bar{s}(t))}(u_n))$
- » 根据项的解释，即 $\overline{s(x \mid \bar{s}(t))}(f(u_1, \dots, u_n))$



引理 2.3I (替换引理, *Substitution Lemma*) [Enderton, pp.133]:

若项 t 可以替换 wff φ 中的变元 x , 那么

$$\models_{\mathfrak{A}} \varphi_t^x[s] \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi[s(x \mid \bar{s}(t))]$$

证明: 对 wff φ 的结构进行归纳。

› **奠基:** φ 为原子公式。那么根据引理 2.30 可知该结论成立。例如 $\varphi \equiv Pu_1 \cdots u_n$ 则

$$\begin{aligned} & \models_{\mathfrak{A}} (Pu_1 \cdots u_n)_t^x[s] \\ \iff & (\bar{s}((u_1)_t^x), \dots, \bar{s}((u_n)_t^x)) \in P^{\mathfrak{A}} && \text{Def.} \\ \iff & \overline{(s(x \mid \bar{s}(t))(u_1), \dots, s(x \mid \bar{s}(t))(u_n))} \in P^{\mathfrak{A}} && \text{Lemma 2.30} \\ \iff & \models_{\mathfrak{A}} (Pu_1 \cdots u_n)[s(x \mid \bar{s}(t))] \end{aligned}$$



替换引理

证明 (Cont'd) :

› **归纳:** 假设替换引理对构造次数小于 n 的 wff 成立

» 若 φ 为形如 $\neg\psi$ 或 $\psi \rightarrow \theta$ 的 wff, 根据 $\neg$ 和 $\rightarrow$ 的解释, 替换引理显然对它们成立

» 若 φ 为形如 $\forall y \psi$, 需要根据可替换的定义分情况讨论:

1. x 在 φ 中未自由出现, 那么 s 与 $s(x \mid \bar{s}(t))$ 在 φ 中所有自由变元的取值相同, 所以 $\varphi_t^x = \varphi$, 结论显然成立

2. x 在 φ 中自由出现。由于 t 可以替换 φ 中的 x , 因此 y 不在 t 中出现且 t 可替换 ψ 中的 x

» 因为 y 不在 t 中出现, 所以对任意 $d \in |\mathfrak{A}|$ 有 $\bar{s}(t) = \overline{s(y \mid d)}(t)$ (*)

» 由于 $x \neq y$, $\varphi_t^x = \forall y \psi_t^x$, 那么有

$$\begin{aligned}
\vdash_{\mathfrak{A}} \varphi_t^x[s] & \text{ iff 对任意 } d, \vdash_{\mathfrak{A}} \psi_t^x[s(y \mid d)] && \text{定义} \\
& \text{ iff 对任意 } d, \vdash_{\mathfrak{A}} \psi[s(y \mid d)(x \mid \bar{s}(t))] && \text{归纳假设 \& (*)} \\
& \text{ iff } && \vdash_{\mathfrak{A}} \varphi[s(x \mid \bar{s}(t))]
\end{aligned}$$

因此, 原命题对任意 wff 均成立

Q.E.D.



第二组公理的有效性

第二组公理是逻辑有效的, 即 $\models \forall x \varphi \rightarrow \varphi_t^x$, 其中 t 对于 φ 中的 x 是可替换的

证明:

假设 t 可以在 φ 中替换 x , 且 $\models_{\mathfrak{A}} (\forall x \varphi)[s]$, 即需要证 $\models_{\mathfrak{A}} \varphi[s(x \mid \bar{s}(t))]$

已知对任意 $d \in |\mathfrak{A}|$

$$\models_{\mathfrak{A}} \varphi[s(x \mid d)]$$

那么就可以令 $d = \bar{s}(t)$, 有

$$\models_{\mathfrak{A}} \varphi[s(x \mid \bar{s}(t))]$$

根据替换引理可知

$$\models_{\mathfrak{A}} \varphi_t^x[s]$$

由于 $\mathfrak{A}$ 和 s 的任意性, 有 $\models \forall x \varphi \rightarrow \varphi_t^x$ 成立。进而, FOL 的公理均是逻辑有效的, 这也完成了可靠性定理的证明 Q.E.D.



FOL可靠性定理的推论

推论 2.32 [Enderton, pp.134]:

若 $\vdash (\varphi \leftrightarrow \psi)$, 那么 φ 与 ψ 逻辑等价

推论 2.33 [Enderton, pp.134]:

若 φ' 是 φ 的一个换名 (**alphabetic variant**), 那么它们是逻辑等价的

推论 2.34 [Enderton, pp.134]:

若 Γ 是可满足的, 那么它是**一致的**



Gödel 完全性定理



一阶逻辑的完全性（完备性）定理是数理逻辑的基本定理之一，**非常重要**。它由 Kurt Gödel 于 20 世纪 30 年代证明。

本小节中我们给出带等词的一阶谓词逻辑的完全性定理，主要采用 Henkin 在 20 世纪 50 年代给出的证明方法，即极大一致集方法。



哥德尔的博士论文:

On the completeness of the calculus of logic (1929)

1. Introduction

The main object of the following investigations is the proof of the completeness of the axiom system for what is called the restricted functional calculus, namely the system given in *Whitehead and Russell 1910*, Part I, *1 and *10, and, in a similar way, in *Hilbert and Ackermann 1928* (hereafter cited as H. A.), III, §5. Here ‘completeness’ is to mean that every valid formula expressible in the restricted functional calculus (a valid *Zählaussage*, as Löwenheim would say) can be derived from the axioms by means of a finite sequence of formal inferences. This assertion can easily be seen to be equivalent to the following: Every consistent axiom system¹ consisting of only *Zählaussagen* has a realization. (Here ‘consistent’ means that no contradiction can be derived by means of finitely many formal inferences.)



定理 2.35 (完全性定理, *Completeness Theorem*) [Enderton, pp.135]:

1. 若 $\Gamma \models \varphi$, 则 $\Gamma \vdash \varphi$
2. 任意一致的 wff 集合是可满足的

› 以上两个命题的等价性可从命题逻辑部分的定理 1.31 简单推广而来



完全性定理

考虑可数无穷的一阶语言[†]：与命题逻辑完全性的证明类似，我们将从一致集 Γ 开始，一步步扩张为极大一致集 Δ 并构造它的模型

1. $\Gamma \subseteq \Delta$

2. Δ 是一致的，且是极大的。即对任意wff α 都有：要么 $\alpha \in \Delta$ ，要么 $(\neg\alpha) \in \Delta$

3. 对任意wff φ 和任意的变元 x ，存在一个常量 c 使得

$$(\neg\forall x \varphi \rightarrow \neg\varphi_c^x) \in \Delta$$

» 即 $(\exists x \neg\varphi \rightarrow \neg\varphi_c^x) \in \Delta$ ，也叫Henkin's Expansion

4. 从 Δ 中构造 $\mathfrak{A}$ 令 Γ 中所有不含等词的wff都被满足。其中 $|\mathfrak{A}|$ 是所有项的集合，即

$$\langle t_1, \dots, t_n \rangle \in P^{\mathfrak{A}} \quad \text{iff} \quad Pt_1 \cdots t_n \in \Delta$$

5. 在 $\mathfrak{A}$ 中增加对等词的支持

[†]完全性对于不可数的FOL依然成立，但需要更复杂的集合论工具



第一步：增加FOL中的常元

令 Γ 为可数FOL语言的一个一致集，那么在该语言中增加可数个常元后（公理集被扩充）， Γ 仍然是一致的

证明[Enderton, pp. 135]:

（注意：一致性是语法中的概念，只与FOL的语言本身有关）

1. 向FOL $\mathcal{L}$ 中增加可数多个常元 $C = \{c_1, c_2, \dots\}$ 后记作 $\mathcal{L}_C$ 。假设 $\mathcal{L}$ 一致，但 $\mathcal{L}_C$ 不一致
2. 根据定义， $\mathcal{L}_C$ 中有 $\Gamma \vdash (\beta \wedge \neg\beta)$ 。显然，该证明（包括 β 与 $\neg\beta$ ）中至多只包含有穷个属于 C 的新常元符号
3. 这些新常元不在 Γ 中出现（ Γ 在 $\mathcal{L}$ 中定义）。那么根据概括定理，我们可将以上证明以及 β 中出现的新常元概括为变元
4. 如此，便得到 $\mathcal{L}$ 中的一段证明： $\Gamma \vdash (\beta' \wedge \neg\beta')$ 。这与 $\mathcal{L}$ 一致的假设矛盾 Q.E.D.



增加FOL常元为什么不会带来矛盾?

例子：我们将Peano算术公理系统PA作为 Γ

1. $\forall x (S(x) \neq 0)$
 2. $\forall x \forall y (S(x) = S(y) \rightarrow x = y)$
 3. $\forall x (x + 0 = x)$
 4. $\forall x \forall y (x + S(y) = S(x + y))$
 1. $\forall x (x \times 0 = 0)$
 2. $\forall x \forall y (x \times S(y) = (x \times y) + x)$
 3. $[\varphi(0) \wedge \forall x (\varphi(x) \rightarrow \varphi(S(x)))] \rightarrow \forall x \varphi(x)$, 其中 φ 是一阶wff
- » 归纳原理是一个公理模式



增加 FOL 新的 $\mathcal{W}_{FF}$ 为什么不带来矛盾?

仍然是在 PA 中:

1. 按照极大一致集的扩张方式, 显然不会带来矛盾 (Lindenbaum's lemma)
2. 增加 Henkin 公理 $\neg \forall x \varphi \rightarrow \neg \varphi_c^x$ 呢?

比如:

- > $\neg \forall x (x + x = 4) \rightarrow \neg (c + c = 4)$
- > $\neg \forall x (s(x) \neq 0) \rightarrow \neg (s(c) \neq 0)$



第二步：HENKIN EXPANSION

对FOL $\mathcal{L}_C$ 中的所有wff φ 以及变元 x ，在 Γ 中增加如下wff

$$\neg \forall x \varphi \rightarrow \neg \varphi_c^x$$

其中 $c \in C$ 是新引入的常元符号。我们记这些wff的集合为 Θ （也被称为“Henkin公理”）

Henkin公理的构造过程：

1. 枚举二元组 $\langle \varphi, x \rangle$ ，其中 φ 和 x 分别为 $\mathcal{L}_C$ 中的wff和变元。由于 $\mathcal{L}_C$ 可数，所以这种枚举是可行的
2. 令 $\theta_1 \equiv \neg \forall x_1 \varphi_1 \rightarrow (\varphi_1)_{c_1}^{x_1}$ ，其中 $c_1 \in C$ 是 φ_1 中未出现的第一个新常元（本质上可看作一个关于反例——如果存在的话——的参数变元，因为我们不指定它在原论域中的取值）。一般地，我们定义

$$\theta_n \equiv \neg \forall x_n \varphi_n \rightarrow (\varphi_n)_{c_n}^{x_n}$$

其中 $c_n \in C$ 是第一个未出现在 φ_n 和 θ_k 中的新常元，其中 $k < n$

3. 令 $\Theta = \{\theta_1, \theta_2, \dots\}$ ，我们断言 $\Gamma \cup \Theta$ 是一致的



第二步：HENKIN'S EXPANSION

证明[Enderton, pp. 136]:

- I. 假设 $\Gamma \cup \Theta$ 不一致, 那么必然对于某个 $m \geq 0$ 有 $\Gamma \cup \{\theta_1, \dots, \theta_m\}$ 一致, 但 $\Gamma \cup \{\theta_1, \dots, \theta_m, \theta_{m+1}\}$ 不一致, 根据归谬法 (RAA):

$$\Gamma \cup \{\theta_1, \dots, \theta_m\} \vdash \neg \theta_{m+1}$$

其中 $\theta_{m+1} \equiv \neg \forall x \varphi \rightarrow \neg \varphi_c^x$, 而 φ, x, c 则是 $\mathcal{L}_C$ 中的某个 wff、变元和常元

2. 因为 $\neg(\alpha \rightarrow \beta) \vdash \neg(\alpha \wedge \neg \beta)$, 根据规则 T 可知

$$\Gamma \cup \{\theta_1, \dots, \theta_m\} \vdash \neg \forall x \varphi \quad \text{并且} \quad \Gamma \cup \{\theta_1, \dots, \theta_m\} \vdash \varphi_c^x$$

3. 对于第二个表达式, 因为 c 是在 $\Gamma \cup \{\theta_1, \dots, \theta_m\}$ 中从未出现的常元, 根据常数概括定理, 我们有

$$\Gamma \cup \{\theta_1, \dots, \theta_m\} \vdash \forall x \varphi$$

那么与 $\Gamma \cup \{\theta_1, \dots, \theta_m\} \vdash \neg \forall x \varphi$ 可推出

$$\Gamma \cup \{\theta_1, \dots, \theta_m\} \vdash \perp$$

4. 这与 m 是最小性假设矛盾

Q.E.D.



第三步：极大一致集

至此，可将 $\Gamma \cup \Theta$ 进一步扩张为一个完全且一致的极大一致集 Δ ，即对 $\mathcal{L}_C$ 中的任意wff φ 都有：要么 $\varphi \in \Delta$ ，要么 $(\neg\varphi) \in \Delta$

证明概要[Enderton, pp. 137]:

- › 与命题逻辑部分的引理1.35类似
 - » 令 Λ 为 $\mathcal{L}_C$ 中的公理，若 $\Gamma \cup \Theta$ 一致，根据定理2.27有 $\Gamma \cup \Theta \cup \Lambda$ 不可能同时重言蕴涵 β 和 $\neg\beta$ 且 $\Lambda \subseteq \Delta$ 。那么便存在一个素公式上的真值指派函数 v 满足 $\Gamma \cup \Theta \cup \Lambda$ ，只需令

$$\Delta = \{\varphi \mid \bar{v}(\varphi) = T\}$$

即可，显然 φ 和 $\neg\varphi$ 不可能同时属于 Δ ，且 Δ 是一致的

- › 由于 Δ 的极大性，它也是一个关于 $\vdash$ 的传递闭包

$$\begin{aligned} \Delta \vdash \varphi &\Rightarrow \Delta \not\vdash \neg\varphi && \text{一致性} \\ &\Rightarrow (\neg\varphi) \notin \Delta \\ &\Rightarrow \varphi \in \Delta && \text{极大性} \end{aligned}$$



第四步：从 Δ 中读出结构 $\mathfrak{A}$

接下来，我们可以从 Δ 中为 $\mathcal{L}_C$ 构造一个结构 $\mathfrak{A}$ ，其中暂时用一个特殊的二元谓词 E 代替 $\mathcal{L}_C$ 中的等词（如果有的话）：

1. $|\mathfrak{A}|$ 为 $\mathcal{L}_C$ 中的所有项
2. 定义二元关系 $E^{\mathfrak{A}}$ ，令

$$\langle u, t \rangle \in E^{\mathfrak{A}} \quad \text{当且仅当 wff } u = t \text{ 属于 } \Delta$$

3. 对所有 $\mathcal{L}_C$ 中的 n -元谓词 P ，定义 n -元关系 $P^{\mathfrak{A}}$ ，令

$$\langle t_1, \dots, t_n \rangle \in P^{\mathfrak{A}} \quad \text{当且仅当 wff } Pt_1 \cdots t_n \in \Delta$$

4. 对所有 $\mathcal{L}_C$ 中的 n -元函词 f ，定义 n -元函数 $f^{\mathfrak{A}}$ ，令

$$f^{\mathfrak{A}}(t_1, \dots, t_n) = ft_1 \cdots t_n$$

以上均包含 $n = 0$ 的情况。对所有常元 c 定义 $c^{\mathfrak{A}} = c$ ；定义变元赋值函数 $s : V \rightarrow |\mathfrak{A}|$ 为恒等函数 $s(x) = x$



第四步：从 Δ 中读出结构 $\mathfrak{A}$

那么，对任意 $\mathcal{L}_C$ 中的公式 φ ，令 φ^* 是将其中的等词替换为 E 后的形式，我们有

$$\models_{\mathfrak{A}} \varphi^*[s] \quad \text{iff} \quad \varphi \in \Delta$$

证明[Enderton, pp.138]:

- › 对 wff 进行结构归纳，即对 wff 构造算子运用次数施归纳
- › 其中关于原子公式和 $(\neg\varphi)^*$ 形式，该结论不言自明
- › 接下来只展示关于 $(\varphi \rightarrow \psi)^*$ 与 $(\forall x \varphi)^*$ 的证明



第四步：从 Δ 中读出结构 $\mathfrak{A}$

证明 (Cont'd) [Enderton, pp.138]:

$$\begin{aligned}\models_{\mathfrak{A}} (\varphi \rightarrow \psi)^*[s] &\Leftrightarrow \not\models_{\mathfrak{A}} \varphi^*[s] \text{ or } \models_{\mathfrak{A}} \psi^*[s] \\ &\Leftrightarrow \varphi \notin \Delta \text{ or } \psi \in \Delta && \text{IH} \\ &\Leftrightarrow (\neg\varphi) \in \Delta \text{ or } \psi \in \Delta \\ &\Leftrightarrow \Delta \vdash (\varphi \rightarrow \psi) && \text{tautologically} \\ &\Leftrightarrow (\varphi \rightarrow \psi) \in \Delta\end{aligned}$$



第四步：从 Δ 中读出结构 $\mathfrak{A}$

证明 (Cont'd) :

显然有 $(\forall x \varphi)^* = \forall x \varphi^*$, 因此我们只需证明: $\models_{\mathfrak{A}} \forall x \varphi^*[s] \iff \forall x \varphi \in \Delta$

1. 如果 $\forall x \varphi \in \Delta$, 那么根据公理 2.7.2 和 MP 规则可知 $\Delta \vdash \varphi_c^x$, 故而有 $\varphi_c^x \in \Delta$, 其中 $c \in |\mathfrak{A}|$
2. 根据归纳假设, 对所有 $c \in |\mathfrak{A}|$ 有 $\models_{\mathfrak{A}} \varphi^*[s(x \mid c)]$, 即 $\models_{\mathfrak{A}} \forall x \varphi^*[s]$ 。故而有

$$\forall x \varphi \in \Delta \Rightarrow \models_{\mathfrak{A}} \forall x \varphi^*[s]$$

3. 反过来, 若 $\forall x \varphi \notin \Delta$, 那么根据 Δ 的极大性有 $\neg \forall x \varphi \in \Delta$
4. 注意到 Henkin 公理 $(\neg \forall x \varphi \rightarrow \neg \varphi_c^x) \in \Delta$, 其中 $c \in C$ 是特意引入的“反例常元”
5. 结合 MP 规则有 $\Delta \vdash \neg(\varphi_c^x)^*$, 因此 $(\neg \varphi_c^x) \in \Delta$, 根据归纳假设和替换引理有 $\not\models_{\mathfrak{A}} (\varphi^*)_c^x[s]$
6. 那么根据全称量词 $\forall$ 的解释可知 $\not\models_{\mathfrak{A}} \forall x \varphi^*[s]$, 因此

$$\forall x \varphi \notin \Delta \Rightarrow \not\models_{\mathfrak{A}} \forall x \varphi^*[s]$$

Q.E.D.



第五步：处理等词

在前面的证明里，我们最初构造出来的结构 $\mathfrak{A}$ 是一个通用的结构

- › 即“等号”也只被当作一个普通的二元关系符号处理，它的解释（interpretation）记作 $E^{\mathfrak{A}}$ ，只是 $\mathfrak{A}$ 上的某个二元关系而已



第五步：处理等词

于是，就会出现这样的情况：

- › FOL里有两个常元 c 和 d
- › wff集 Δ 里有 $c = d$
- › 但是，在前面构造出来的结构 $\mathfrak{A}$ 里可能有 $c^{\mathfrak{A}} \neq d^{\mathfrak{A}}$ ，即 c 和 d 分别被解释为 $\mathfrak{A}$ 中的不同元素
- › 不过由于我们对“=”进行的解释——二元关系 $E^{\mathfrak{A}}$ ，在构造过程中包含了 $(c^{\mathfrak{A}}, d^{\mathfrak{A}})$ 这个元素对
 - » 注意，这只是说明：“我在这个关系的定义里放了 $(c^{\mathfrak{A}}, d^{\mathfrak{A}})$ ”
- › 这种做法没法保证满足我们对等号的**真正要求**：
 - » 当 $c = d$ 在 Δ 中出现时，它应该强制 $c^{\mathfrak{A}} = d^{\mathfrak{A}}$ （它们真的是同一个对象）



第五步：处理等词

换句话说：

- › 在一般的结构中，“=”可以像普通关系一样，随便给出一个集合里的二元关系当解释
- › 但在有等号的语言里，所谓“等号”必须解释为 $\mathfrak{A}$ 上的恒等关系

所以，如果我们只是构造了 $\mathfrak{A}$ 而不做进一步处理，那么可能会出现：

- › $c^{\mathfrak{A}} \neq d^{\mathfrak{A}}$ 但是 $(c^{\mathfrak{A}}, d^{\mathfrak{A}}) \in =^{\mathfrak{A}}$
- › **这在逻辑上是不允许的**

第五步：处理等词



怎么办？



商 (quotient) 结构



第五步：处理等词

我们定义 $E_{\mathfrak{A}}$ 为 $\mathfrak{A}$ 里的等号关系 $=^{\mathfrak{A}}$ ，然后构造商结构 $\mathfrak{A}/E_{\mathfrak{A}}$ ，把所有被 $E_{\mathfrak{A}}$ 认为相等的元素归成一个等价类。这样得到的新结构里：

- › 等号就是恒等关系
- › 所有模型中的等式约束都自动满足



解决等词语义： 可定义性、同态与同构



可定义性：自由变量的意义

考虑一个结构： $\mathfrak{N} = (\mathbb{N}; 0, S, +, \cdot)$

1. 该结构中能够定义单点集，如 $\{2\}$

$$v_1 = SS0$$

2. 尽管没有关系 $<$ ，即集合 $\{\langle v_1, v_2 \rangle \mid v_1 < v_2\}$ ，但这并不妨碍我们用以下公式定义它

$$\exists v_3 (v_1 + Sv_3 = v_2)$$

3. 素数（由于1和 $<$ 均可定义，以下wff是合法的）

$$(1 < v_1) \wedge \forall v_2 \forall v_3 (v_1 = v_2 \cdot v_3 \rightarrow v_2 = 1 \vee v_3 = 1)$$

4. 负数和有理数，比如 $-5_{\mathbb{Z}} \equiv \langle v_1, v_2 \rangle$ 和 $\frac{1}{2}_{\mathbb{Q}} \equiv \langle (v_1)_{\mathbb{Z}}, (v_2)_{\mathbb{Z}} \rangle$

$$v_1 + 5 = v_2 + 0; \quad (v_1 \cdot 2)_{\mathbb{Z}} = (v_2 \cdot 1)_{\mathbb{Z}}$$

5. 无理数（戴德金分割）



可定义性

定义 2.2I (可定义性, *definability*) [Enderton, pp.90]:

考虑一个结构 $\mathfrak{A}$ 与一个 wff φ , 若 φ 的自由变元为 $v_1, \dots, v_k$, 那么我们便能够在 $\mathfrak{A}$ 中定义一个 k -元关系

$$\{\langle v_1, \dots, v_k \rangle \mid \models_{\mathfrak{A}} \varphi[v_1, \dots, v_k]\}$$

我们称该 k -元关系在 $\mathfrak{A}$ 中由 φ **定义** (*defines*)

- 一般地, 一个 $|\mathfrak{A}|$ 上的 k -元关系在 $\mathfrak{A}$ 中**可定义** (*definable*) 当且仅当存在一个自由变元为 $v_1, \dots, v_k$ 的 wff 恰好能够描述它

定义 2.22 (类, *Class*) [Enderton, pp.92]:

对于一个语句集 Σ , 用 $\mathcal{K} = \text{Mod } \Sigma$ 来表示由 Σ 的模型所组成的类。如果 Σ 是单个语句的集合 $\{\sigma\}$, 则用 $\text{Mod } \sigma$ 而不用 $\text{Mod } \sigma$ 表示

- › 由一条 FOL 语句定义类被称为初等类 (*elementary class*, EC), 即存在一条语句 σ 使得 $\mathcal{K} = \text{Mod } \sigma$
- › 由 FOL 语句集合 Σ 定义类被称为广义初等类 (*elementary class in a wider sense*, EC_{Δ})



类的例子

- I. [Hao et.al., pp.96] 一阶语言 $\mathcal{L} = \{=, P\}$, 其中 P 为二元谓词符号, 令 σ 为下列 3 个语句的合取

$$\begin{aligned}\forall x \forall y \forall z (Pxy \rightarrow (Pyz \rightarrow Pxz)) \\ \forall x \forall y (Pxy \vee x = y \vee Pyx) \\ \forall x \forall y (Pxy \rightarrow \neg Pyx)\end{aligned}$$

2. [Hao et.al., pp.97] 一阶语言 $\mathcal{L} = \{=, \circ, ^{-1}, e\}$, 其中 $\circ$ 和 $^{-1}$ 分别为二元和一元函数符号, e 为常数符号, 定义下面一组语句集 Σ

$$\begin{aligned}(1) \quad & \forall x \forall y \forall z (x \circ (y \circ z) = (x \circ y) \circ z) \\ (2) \quad & \forall x (x \circ e = e \circ x) \\ (3) \quad & \forall x (x \circ x^{-1} = x^{-1} \circ x = e)\end{aligned}$$



类的例子

```
import re

n = 13
print(not re.match(r'^.?$|^(..+?)\1+$', '1'*n))
```



类的例子

朴素集合论中的**概括原则**：若 φ 是一个公式， v_2 是其中的自由变元，那么：

$$\exists v_1 \forall v_2 (v_2 \in v_1 \leftrightarrow \varphi(v_2))$$

- › 最早来自于 Frege 的 *The Foundations of Arithmetic* (1884)
- › 当定义 $\varphi \equiv \neg(v_2 \in v_2)$ ，那么它定义的就只是一个真类而不是集合（罗素悖论）
- › 为了回避该问题，ZFC 选取了分离公理模式（Axiom schema of specification）代替它

$$\forall v_3 \exists v_1 \forall v_2 (v_2 \in v_1 \leftrightarrow (v_2 \in v_3 \wedge \varphi(v_2)))$$



同态

定义 2.23 (同态, *Homomorphisms*) [Enderton, pp.94; Hao et.al., pp.100]:

令 $\mathfrak{A}$ 和 $\mathfrak{B}$ 是FOL语言的两个结构, 称函数 $h: |\mathfrak{A}| \rightarrow |\mathfrak{B}|$ 是从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 的一个同态, 如果它满足下列条件:

1. 对每个 (不是等词的) 谓词 P 以及每组 $|\mathfrak{A}|$ 中的 n -元组 $\langle a_1, \dots, a_n \rangle$, 都有

$$\langle a_1, \dots, a_n \rangle \in P^{\mathfrak{A}} \text{ iff } \langle h(a_1), \dots, h(a_n) \rangle \in P^{\mathfrak{B}}$$

2. 对每个函数符号 f 以及每组 $|\mathfrak{A}|$ 中的 n -元组 $\langle a_1, \dots, a_n \rangle$, 都有

$$h(f^{\mathfrak{A}}(a_1, \dots, a_n)) = f^{\mathfrak{B}}(h(a_1), \dots, h(a_n))$$

3. 对于每个常数符号 c 都有

$$h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$$

在上述定义中如果 h 是一个双射, 则称 h 是从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 的一个同构 (*isomorphism*), 并称 $\mathfrak{A}$ 和 $\mathfrak{B}$ 同构[†], 记为 $\mathfrak{A} \cong \mathfrak{B}$

[†][Enderton]根据映射的大小区分了“onto”和“into”的同构。由于不影响可靠性和完全性的证明, 我们采取[Hao et.al.]的术语, 即“同构”指的是onto的同构, 也即双射的同构。



同态的例子

定义自然数集合上的结构 $\mathfrak{A} = (\mathbb{N}; +, \cdot)$ ，并定义一个映射 $h : \mathbb{N} \rightarrow \{e, o\}$

$$h(n) = \begin{cases} e & \text{if } n \text{ is even} \\ o & \text{if } n \text{ is odd} \end{cases}$$

那么 h 是一个 $\mathfrak{A}$ 到 $\mathfrak{B}$ 的（满）同态，其中 $\mathfrak{B} = (\{e, o\}; +^{\mathfrak{B}}, \cdot^{\mathfrak{B}})$ ，运算定义如下：

$+^{\mathfrak{B}}$	e	o
e	e	o
o	o	e

$\cdot^{\mathfrak{B}}$	e	o
e	e	e
o	e	o

容易验证，它满足定义 2.23 中的条件 2 和 3

既然同态定义了结构之间的联系，
那么 FOL wff 的语义（真值）能够通过同态从一个结构传到另一个结构吗？



同态定理

定理 2.24 (同态定理, *Homomorphism Theorem*) [Enderton, pp.96; Hao et.al., pp.100]:

假定 h 为从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 的一个同态, 并且 $s : V \rightarrow |\mathfrak{A}|$ 的一个赋值:

1. 对任意项 t , $h(\overline{s(t)}) = \overline{h \circ s(t)}$, 其中 $\overline{s(t)}$ 在 $\mathfrak{A}$ 中计算且 $\overline{h \circ s}$ 在 $\mathfrak{B}$ 中计算
2. 对任何不含量词且不含等词的公式 α

$$\models_{\mathfrak{A}} \alpha[s] \quad \text{iff} \quad \models_{\mathfrak{A}} \alpha[h \circ s]$$

3. 如果 h 是单射, 则 2 中的 wff α 可以包含等词
4. 如果 h 是 $|\mathfrak{A}|$ 到 $|\mathfrak{B}|$ 的满射, 则 2 中的 wff α 也可以包含量词



同态定理的证明概要 [ENDERTON, PP.97]

1. 根据 s 和 h 的定义, 对项的结构做归纳证明
2. 对 (无等词和量词结构的) wff 结构做归纳证明
3. 显然
4. 同样可以使用归纳法。根据归纳假设已经有 $\models_{\mathfrak{A}} \varphi[s] \Leftrightarrow \models_{\mathfrak{B}} \varphi[h \circ s]$
 - » 首先, 不管 h 是否是满射都有

$$\begin{aligned}\models_{\mathfrak{B}} \forall x \varphi(h \circ s) &\Leftrightarrow \text{对任意 } e \in |\mathfrak{B}|, \models_{\mathfrak{A}} \varphi[(h \circ s)(x \mid e)] \\ &\Rightarrow \text{对任意 } d \in |\mathfrak{A}|, \models_{\mathfrak{B}} \varphi[(h \circ s)(x \mid h(d))] \\ &\Leftrightarrow \text{对任意 } d \in |\mathfrak{A}|, \models_{\mathfrak{A}} \varphi[s(x \mid d)]\end{aligned}$$

- » 若是满射, 上面推理步骤的第 2 步显然也有 $\Leftarrow$ 成立



初等等价与自同构

定理 2.25 (初等等价, *Elementarily equivalent*) [Enderton, pp.97; Hao et.al., pp.102]:

我们称一个 FOL 语言的两个结构 $\mathfrak{A}$ 和 $\mathfrak{B}$ 是初等等价的, 当且仅当对任意语句 σ 都有

$$\models_{\mathfrak{A}} \sigma \quad \text{iff} \quad \models_{\mathfrak{B}} \sigma$$

记为 $\mathfrak{A} \equiv \mathfrak{B}$ 。同构的两个结构初等等价, 即 $\mathfrak{A} \cong \mathfrak{B} \Rightarrow \mathfrak{A} \equiv \mathfrak{B}$

定理 2.26 (自同构, *Automorphism*) [Enderton, pp.98; Hao et.al., pp.102]:

一个由结构 $\mathfrak{A}$ 到自身的同构被称为自同构。令 h 是 $\mathfrak{A}$ 的一个自同构, 且 R 是一个在 $|\mathfrak{A}|$ 上的 n -元关系, 它在 $\mathfrak{A}$ 中可定义。则对任意 $|\mathfrak{A}|$ 中的 n -元组 $\langle a_1, \dots, a_n \rangle$ 有

$$\langle a_1, \dots, a_n \rangle \in R \Leftrightarrow \langle h(a_1), \dots, h(a_n) \rangle \in R$$



例子 [ENDERTON, PP.IOI]

试证明： 加法关系 $A = \{\langle m, n, p \rangle \mid p = m + n\}$ 在 $(\mathbb{N}; \cdot)$ 中无法定义。

证明：

根据算术基本定理， $(\mathbb{N}; \cdot)$ 是一个由全体素数与乘法生成的自由代数（交换半群），且包含一个0元。因此只要将生成元（素数）打乱顺序就可以构造自同构：

- 若 $n = \prod_p p^{n_p}$ ，那么我们定义 $h(n) = 2^{n_3} \cdot 3^{n_2} \prod_{p \geq 5} p^{n_p}$ ，显然它是一个一一映射
- 易证 h 关于乘法是一个自同构：对于 $n = \prod_p p^{n_p}$ 和 $m = \prod_p p^{m_p}$ 有

$$h(n \cdot m) = h\left(\prod_p p^{n_p+m_p}\right) = 2^{n_3+m_3} \cdot 3^{n_2+m_2} \prod_{p \geq 5} p^{n_p+m_p} = h(n) \cdot h(m)$$

- 然而 $\langle 1, 1, 2 \rangle \in A$ 但 $\langle h(1), h(1), h(2) \rangle = \langle 1, 1, 3 \rangle \notin A$ ，根据定理 2.24 可知 A 在该结构中无法定义

Q.E.D.



Gödel 完全性定理（续）



第五步：处理等词（续）

若 $\mathcal{L}$ 中有等词，我们定义 $E^{\mathfrak{A}}$ 为 $|\mathfrak{A}|$ 上的一个等价关系作为对 $=$ 的解释。对任意项 $t \in |\mathfrak{A}|$ ，定义 $[t]$ 为 t 关于 $E^{\mathfrak{A}}$ 的等价类。那么，我们可以获得一个商结构（*quotient structure*）：

1. $|\mathfrak{A}/E|$ 是 $|\mathfrak{A}|$ 关于 $E^{\mathfrak{A}}$ 的商集
2. 对每个 n -元谓词 P

$$\langle [t_1], \dots, [t_n] \rangle \in P^{\mathfrak{A}/E} \quad \text{iff} \quad \langle t_1, \dots, t_n \rangle \in P^{\mathfrak{A}}$$

3. 对每个 n -元函词 f

$$f^{\mathfrak{A}/E}([t_1], \dots, [t_n]) = [f^{\mathfrak{A}}(t_1, \dots, t_n)]$$

4. 对于每个常元

$$c^{\mathfrak{A}/E} = [c^{\mathfrak{A}}]$$

5. 令 $h : |\mathfrak{A}| \rightarrow |\mathfrak{A}/E|$ 为一个自然映射 $h(t) = [t]$



第五步：处理等词

那么， h 就是一个 $\mathfrak{A}$ 到 $\mathfrak{A}/E$ 的同态。进一步地， $E^{\mathfrak{A}/E}$ 是 $|\mathfrak{A}/E|$ 中的等价关系。因此，对任意 φ 有

$$\begin{aligned}\varphi \in \Delta &\Leftrightarrow \models_{\mathfrak{A}} \varphi^*[s] && \text{第四步} \\ &\Leftrightarrow \models_{\mathfrak{A}/E} \varphi^*[h \circ s] && \text{同态定理 2.24} \\ &\Leftrightarrow \models_{\mathfrak{A}/E} \varphi[h \circ s] && E^{\mathfrak{A}/E} \text{ 是等价关系}\end{aligned}$$

因此 $\mathfrak{A}/E$ 和赋值函数 $h \circ s$ 满足 Δ 中的任意wff

证明概要[Enderton, pp.141]:

- I. 首先需证明关于 $=$ 的解释 $E^{\mathfrak{A}}$ 的确是 $\mathfrak{A}$ 中的等价关系，且满足FOL等词的几个性质（显然成立）
2. 验证 h 是一个同态映射（显然成立）



第六步：从 $\mathcal{L}_C$ 返回 $\mathcal{L}$

将结构 $\mathfrak{A}/E$ 限制 (restrict) 在 $\mathcal{L}$ 上，它与 $h \circ s$ 满足 Γ 中的任意一个 wff
Q.E.D.



小结



一阶逻辑

- I. 一阶语言的语法
 - » 量词和等词
 - » 自由变元与替换
2. 一阶逻辑的 Hilbert System
 - » 概括定理
3. 语义
 - » 结构与真
 - » 解释
4. 一阶逻辑与命题逻辑的关系
5. 一阶逻辑的可靠性
 - » 替换引理
6. 哥德尔完全性定理