模态逻辑

杨睿之

复旦大学哲学学院

2024 年秋季

前情提要

- （一阶逻辑模型论的）饱和模型

定义

给定一阶逻辑语言 $\mathcal{L}$ 和 $\mathcal{L}$-结构 $\mathfrak{M}$。假设 A 是 $\mathfrak{M}$ 论域的一个子集

- 定义语言 $\mathcal{L}_A = \mathcal{L} \cup \{c_a\}_{a\in A}$，即添加了 $|A|$ 那么多个新的常数符号。$\mathfrak{M}$ 到 $\mathcal{L}_A$ 的膨胀 $\mathfrak{M}_A$ 把每个 c_a 解释为 a
- 定义 $\mathcal{L}_A$ 公式集 $p(x)$（其中公式至多含有自由变元 x）是 $\mathfrak{M}$ 的一个 A 上的 1-型 （1-type），当且仅当 $p(x)$ 在 $\mathfrak{M}_A$ 中 有穷可实现 。即，对 $p(x)$ 每个有穷子集 $p_0(x)$ 存在 $\mathfrak{M}$ 论域中元素 b 有 $\mathfrak{M}_A \models p_0(b)$

前情提要

事实

在前一页定义的假设下。$p(x)$ 是 $\mathfrak{M}$ 的 A 上的 1-型，即 $p(x)$ 在 $\mathfrak{M}_A$ 中有穷可实现，当且仅当 $p(x) \cup \mathrm{Th}\,\mathfrak{M}_A$ 是一致的

前情提要

定义

- 令 κ 是一个（有穷或无穷的）基数。$\mathfrak{M}$ 是一个 $\mathcal{L}$-结构。我们称 $\mathfrak{M}$ 是 κ-饱和的 （κ-saturated），当且仅当对任意 $\mathfrak{M}$ 论域的子集 A 有 $|A| < \kappa$，任意 $\mathfrak{M}$ 的 A 上的 1-型 $p(x)$ 在 $\mathfrak{M}_A$ 中可实现，即存在 $\mathfrak{M}$ 中 b 有 $\mathfrak{M}_A \vDash p(b)$

显然，若 $\kappa < \lambda$，λ-饱和蕴含 κ-饱和。模态逻辑中主要关注 ω-饱和性 ，即所有只含有有穷参数的 1-型都可实现

例

- 有穷结构都是 ω-饱和的
- $(\mathbb{Q},<)$ 是 ω-饱和的，是 $\aleph_1$-饱和的吗?
- $(\mathbb{N},<)$ 不是 ω-饱和的

前情提要

例

- 有穷结构都是 ω-饱和的
- $(\mathbb{Q}, <)$ 是 ω-饱和的，是 $\aleph_1$-饱和的吗?
- $(\mathbb{N}, <)$ 不是 ω-饱和的

饱和模型

例

- 对每个模态逻辑公式 ϕ，$ST_x(\phi)$ 是一个只含有 x 为自由变元的一阶逻辑公式
- 考虑 $\mathfrak{M} = (W, R, V)$，$w \in W$。假设模态逻辑公式集 Σ 在 $R[w]$ 上有穷可满足。则

$$\{Rwx\} \cup \{ST_x(\phi) \mid \phi \in \Sigma\}$$

是一个 $\mathfrak{M}$ 的 $\{w\}$ 上的 1-型

饱和模型

定理

给定模态逻辑语言类型 τ。任何 ω-饱和的 τ-模型都是模态饱和的。因此，ω-饱和的 τ-模型类具有 Hennessy-Milner 性质

定义

给定一阶逻辑语言 $\mathcal{L}$，$\mathcal{L}$-结构序列 $\langle\mathfrak{A}_i\rangle_{i\in I}$ 和 I 上超滤 U，我们定义 $\langle\mathfrak{A}_i\rangle_{i\in I}$ 模 U 的超积 $\prod_U \mathfrak{A}_i$ 为下述 $\mathcal{L}$-结构

- $\prod_U \mathfrak{A}_i$ 的 论域 是集合 $\prod_U A_i$，其中每个 A_i 是 $\mathfrak{A}_i$ 的论域
- 对 $\mathcal{L}$ 中 n 元谓词符号 R，$\prod_U \mathfrak{A}_i$ 对 R 的解释是 $\prod_U A_i$ 上 n 元关系 R_U，满足（其中，R_i 是 $\mathfrak{A}_i$ 对 R 的解释）：

$$R_U[f_1]_U \ldots [f_n]_U \;\Leftrightarrow\; \{i \in I \mid R_i f_1(i) \ldots f_n(i)\} \in U$$

定理 (超积基本定理（Łoś）)

给定语言 $\mathcal{L}$。令 U 是 I 上的超滤，对每个 $i \in I$ 有 $\mathcal{L}$-结构 $\mathfrak{A}_i$

- 对任意 $\mathcal{L}$ 公式 $\alpha(x_1, \ldots, x_n)$ 和 $\mathfrak{A}$ 中元素 $[f_1]_U, \ldots, [f_n]_U$ 有

$$\mathfrak{A} \vDash \alpha([f_1]_U, \ldots, [f_n]_U) \iff \{i \in I \mid \mathfrak{A}_i \vDash \alpha(f_1(i), \ldots, f_n(i))\} \in U$$

饱和模型

定义

我们称 I 上的超滤 U 是 可数不完全的 （countably incomplete），当且仅当存在 $\langle X_n\rangle_{n<\omega}$ 使得，每个 $X_n \in U$ 但 $\bigcap_n X_n \notin U$

例

- 任何 $\mathbb{N}$ 上的非主超滤都是可数不完全的：$\bigcap_n(\mathbb{N} \setminus \{n\}) = \emptyset$

饱和模型

定理

令 $\mathcal{L}$ 是一个可数语言。I 是非空集合。对每个 $i \in I$，$\mathfrak{M}_i$ 是 $\mathcal{L}$-结构。U 是 I 上的一个可数不完全的超滤。那么超积 $\prod_U \mathfrak{M}_i$ 是 $\aleph_1$-饱和的

引理 (迂回引理)

给定模态语言类型 τ 和 τ-模型 $\mathfrak{M}$ 和 $\mathfrak{N}$ 以及分别在其中的状态 w, v。下列命题等价

- $\mathfrak{M}, w \leftrightsquigarrow \mathfrak{N}, v$
- $\mathfrak{ue}\mathfrak{M}, u_w \leftrightharpoons \mathfrak{ue}\mathfrak{N}, u_v$
- 存在超幂 $\prod_U \mathfrak{M}$ 和 $\prod_U \mathfrak{N}$，使得

  $\prod_U \mathfrak{M}, [c_w]_U \leftrightharpoons \prod_U \mathfrak{N}, [c_v]_U$

引理 (迂回引理)

给定模态语言类型 τ 和 τ-模型 $\mathfrak{M}$ 和 $\mathfrak{N}$ 以及分别在其中的状态 w, v。下列命题等价

- $\mathfrak{M}, w \leftrightsquigarrow \mathfrak{N}, v$
- $\mathfrak{ue}\mathfrak{M}, u_w \leftrightharpoons \mathfrak{ue}\mathfrak{N}, u_v$
- 存在 ω-饱和模型 $\mathfrak{M}^*, w^*$ 和 $\mathfrak{N}^*, v^*$ 以及初等嵌入 $f : \mathfrak{M}, w \preccurlyeq \mathfrak{M}^*, w^*$ 和 $g : \mathfrak{N}, v \preccurlyeq \mathfrak{N}^*, v^*$ 有，$\mathfrak{M}^*, w^* \leftrightharpoons \mathfrak{N}^*, v^*$

Van Benthem 刻画定理

定义

给定模态语言类型 τ，令 $\mathcal{L}_\tau$ 为对应的一阶逻辑语言。我们称一个 $\mathcal{L}_\tau$ 公式 $\alpha(x)$ 是 互模拟不变的 （invariant for bisimulations)，当且仅当对任意 τ-模型 $\mathfrak{M}$ 和 $\mathfrak{N}$ 以及分别在其中的状态 w, v，如果 $\mathfrak{M}, w \leftrightarrows \mathfrak{N}, v$，那么

$$\mathfrak{M} \models \alpha(w) \quad \Leftrightarrow \quad \mathfrak{N} \models \alpha(v)$$

Van Benthem 刻画定理

下述定理将一阶逻辑公式的一个语义性质与一个句法性质联系起来

定理 (Van Benthem 刻画定理)

给定模态语言类型 τ 和对应的一阶逻辑语言 $\mathcal{L}_\tau$。$\mathcal{L}_\tau$ 公式 $\alpha(x)$ 是互模拟不变的，当且仅当它逻辑等价于一个 τ-公式的标准翻译