数理逻辑（2024 春）作业 - 08

1 预备知识

下面我们定义一个一阶逻辑语言的子集：逻辑程序。

1.1 子句

定义 1：一个字句（*clause*）是具有以下形式的 wff

$$\forall x_1 \cdots \forall x_S (L_1 \vee \cdots \vee L_m)$$

其中每个 L_i 都是一个原子公式或原子公式的否定，它们也被称为逻辑文字（*literal*），$x_1, \ldots, x_S$ 是 $L_1 \vee \cdots \vee L_m$ 中所有出现过的变元。

例如，$\forall x \forall y \forall z (p(x,z) \vee \neg q(x,y) \vee \neg r(y,z))$ 就是一个字句。一般情况下一个字句可以表示为

$$\forall x_1 \cdots \forall x_S (A_1 \vee \cdots \vee A_k \vee \neg B_1 \vee \cdots \vee \neg B_n)$$

其中的 A_i 和 B_j 都是原子公式，$x_1, \ldots, x_S$ 是 $A_1 \vee \cdots \vee A_k \vee \neg B_1 \vee \cdots \vee B_n$ 中出现的全部变元。它也能表示成

$$A_1, \cdots, A_k \leftarrow B_1, \cdots B_n$$

这里省略了所有的量词，且用逗号代替了 A_i 间的析取符号（“或”）和 B_j 间的合取符号（“与”）。蕴涵词的箭头所指部分 $A_1, \cdots, A_k$ 被称为规则头（*head*）或后件（consequence），另一部分 $B_1, \cdots, B_n$ 被称为规则体（*body*）或前件（*antecedent*）。

1.2 逻辑程序

定义 2：一个定子句（*definite clause*）指只有一个后件的子句

$$A \leftarrow B_1, \cdots B_n$$

由有穷个定子句构成的集合被称为**定程序**（*definite program*）或**逻辑程序**（*logic program*）。若定程序中所有子句的规则头都是同一个谓词 p，我们称这个定程序为 p 的定义（*definition*）。

1.3 事实和目标

定义 3：单位子句（*unit clause*）被用来在定程序中声明一个事实，它指后件为空的定子句

$$A \leftarrow$$

目标子句（*goal clause*）指前件为空的子句

$$\leftarrow B_1, \cdots B_n$$

其中每个 B_i 被称为该目标下的一个子目标（*sub-goal*）。空子句（*empty clause*）指前后件均为空的子句，记为 □。

1.4 Horn 子句

定义 5：一个 $Horn$ 子句是一个定子句或一个目标子句。

1.5 接地

定义 6：一个接地项（*ground term*，或译为具体项）是一个 FOL 项，但不包含任意变元。类似的，一个接地原子（*ground atom*，或具体原子）是不包含任意变元的原子公式。

1.6 Herbrand 域

定义 7：令 L 为一阶语言，它的 Herbrand 域（*Herbrand Universe*）U_L 是 L 中的全部接地项。

例如对于一个逻辑程序，其中 x, y 为变元：

$$\begin{aligned} p(x) &\leftarrow q(f(x), g(x)) \\ r(y) &\leftarrow \end{aligned}$$

那么它的 FOL 语言中有函数词 f, g，谓词 p, q, r。若该语言中只有一个常元 a，那么它的 $Herbrand$ 域为

$$\{a, f(a), g(a), f(f(a)), f(g(a)), g(f(a)), g(g(a)), \ldots\}$$

1.7 Herbrand 基

定义 8：令 L 为一阶语言，它的 Herbrand 基（*Herbrand Base*）B_L 是 L 中的全部接地原子。

例如对于上面的例子，它的 $Herbrand$ 基为：

$$\{p(a), r(a), q(a, a), p(f(a)), r(f(a)), p(g(a)), r(g(a)), q(a, f(a)), q(f(a), a), q(f(a), f(a)), q(f(a), g(a)), \ldots\}$$

2 作业（推荐大家通过自行查资料完成）

2.1 命题逻辑程序

考虑命题逻辑程序，即不含谓词和函数词的逻辑程序，那么每条子句中的原子公式均为一个命题符。

给定一个包含定子句 $F_1, \ldots, F_n$ 的命题逻辑程序以及一个系列待证明子目标，它可以表示为一个合取式 $ g_1 ∧ ... ∧ g_m$，逻辑程序的任务是判断是否有下面的式子成立：

$$\{F_1, \ldots, F_n\} \models g_1 \wedge \ldots \wedge g_m$$

它的推理系统非常简单：

1. 子句 $F_1, \ldots, F_n$ 被当作该程序中的公理（or 前提）
2. 推理规则只有一条，被称为“归结”或“消去”（*Resolution*）：

$$\frac{A_1 \vee A_2 \vee \cdots \vee c \quad B_1 \vee B_2 \vee \cdots \vee \neg c}{A_1 \vee A_2 \vee \cdots \vee B_1 \vee B_2 \vee \cdots}$$

   这里采用的是 Gentzen 式的记法，横线表示“逻辑蕴涵”。可见，两个子句中相反的逻辑文字 c 和 $\neg c$ 被消掉了。

2.1.1 问题 1

命题逻辑程序的证明系统拥有可靠性吗？请证明你的结论。

2.1.2 问题 2（可选）

命题逻辑程序的证明系统拥有完备性吗？请证明你的结论。

2.2 一阶逻辑程序的语义

和一般的 FOL 一样，逻辑程序中可以定义一个语义结构来判断其真假。

2.2.1 问题 3

- 仿照教材上的方式，为**逻辑程序**定义一个结构作为它的**解释**。
- 若一个逻辑程序 P 是一致的，那么它就拥有一个模型（因为它只包含闭公式）。仿照极大一致集的方式，在你定义的结构中构造出 P 的模型。（提示：Herbrand Model）