221900180田永铭 数理逻辑作业8

Problem1 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$F1, ..., Fn$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 $g_1 \wedge ... \wedge g_m$，逻辑程序的任务是判断是否有下面的式子成立：

$F1, ..., Fn \models g1 \wedge ... \wedge gm.$

它的推理系统非常简单：

1. 子句 $F1, ..., Fn$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）：
   $A1 \vee A2 \vee \cdots \vee c B1 \vee B2 \vee \cdots \vee \neg c$
   ————————————————————
   $A1 \vee A2 \vee \cdots \vee B1 \vee B2 \vee \cdots.$

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。

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。

证明：

具有可靠性，由于推理规则只有一个归结原理，所以只需要证明归结原理的可靠性，证明如下：

在命题逻辑程序的证明系统中，可靠性可以定义为：

$$(l_1 \vee \cdots \vee l_k) \wedge (m_1 \vee \cdots \vee m_n) \models$$
$$\quad (l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots m_n).(*)$$

其中:l_i与m_j为互补的文字.

假设$\models$之前的语句都为真,即$(l_1 \vee \cdots \vee l_k)$ 和 $(m_1 \vee \cdots \vee m_n)$都为真：

(1)假设l_i为真，则m_j必假，又由$(m_1 \vee \cdots \vee m_n)$为真，所以$m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots m_n$必须为真;

(2)假设l_i为假，同理知，$l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots l_k$ 为真.

这两种情况都能得到$\models$之后的语句为真.

所以(*)式成立，所以可靠性成立.

证毕！

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。

证明：

我选择试着证一证：

具有完备性，证明如下：

与上一问的证明同理，我们只需要证明每一步归结都是完备的.

首先定义KB，通常指的是 "Knowledge Base"，即知识库。知识库是一个存储系统，其中包含了大量的知识或信息，通常以逻辑命题的形式表示。在给定的上下文中，KB 包含了一组原子命题（基本命题）及其之间的逻辑关系或规则。

所以我们接下来只需要证明：

只要$KB \models \alpha$，我们一定可以使用resolution得到这个结果.

而$KB \models \alpha$ 等价于 $(KB \wedge \neg\alpha)$ 是不可满足的.

现在定义一个归结闭集$RC(S)$,它是所有由S归结出来的子句的集合.

我们将$(KB \wedge \neg\alpha)$ 记作S，则我们接下来的目标就是证明**若S不可满足，则可以使用resolution得到**α.

由"ground resolution theorem"可以得到：

若S不可满足，则$RC(S)$中包含空子句.

这可定理可以通过它的逆否命题来证明，即：

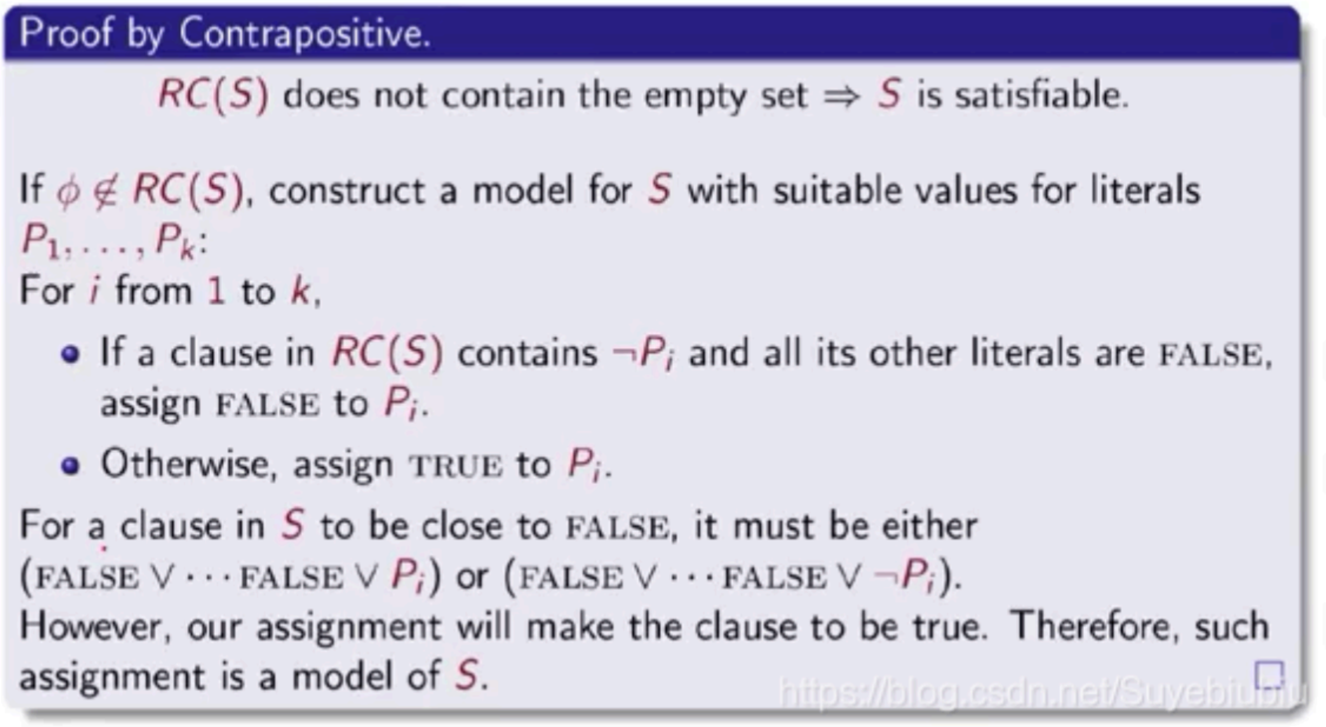
若RC(S)中不含有空子句，则S是可满足的.（*）（证明略去）

因此RC（S）中必定包含空子句，从而$RC(S)$是有限的，所以归结总会中止.

而我们以及证明了归结的可靠性，所以每一步归结必为真，又RC(S)的归结必然包含α，归结又是有穷步，所以最终必能通过归结得到α。

证毕!

附：略去（*）式的证明如下：



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。

问题3

(1)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。

解：

模仿一阶逻辑，一个解释结构需要包含逻辑程序中的所有符号，并且定义它们的定义。其实，我们正可以利用Herbrand结构来进行定义，定义如下：

由逻辑程序语言P生成的字母表包含以下符号：

- P的个体常项符号a,b,c· · ·
- P中的变元$x_1, x_2 \cdots x_s$
- P中的逻辑文字$L_1, L_2 \cdots L_m$
- 原子公式$A_i, B_j \cdots$
- 函数词$f, h \cdots$
- 谓词$p, q, r \cdots$

逻辑程序语言项的定义：

与一阶逻辑不同，这里定义的叫做接地项，这在作业的前置知识中给出了定义，这里不再赘述。

定义它的域、基和解释

1. **Herbrand 域**:
   - 定义：Herbrand 域 U_L 是逻辑程序 P 中所有接地项（ground term）的集合。
   - 例如，若逻辑程序中有常元 a，函数符号 f 和 g，那么 Herbrand 域 U_L 为 $\{a, f(a), g(a), f(f(a)), g(g(a)), f(g(a)), g(f(a)), \ldots\}$。
2. **Herbrand 基（B_p）**:
   - 定义：Herbrand 基 B_L 是逻辑程序 P 中所有接地原子的集合。
   - 例如，若逻辑程序中有谓词 p, q 和 r，那么 Herbrand 基 B_L 为 $\{p(a), q(a, a), r(a), p(f(a)), q(f(a), a), \ldots\}$。
3. **Herbrand 解释**:
   - 定义：Herbrand 解释：若P是一个逻辑程序，那么P的埃尔布朗解释I_p是对B_p的一个真值指派，它也可以表示为2^{B_p}的一个子集，出现在该子集中的具体原子赋值为true.
   - 解释：I 指定了在解释 I 下哪些接地原子为真。

定义完毕!

(2)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）

证明:

事实上，我们能构造出P的一个模型，我们把它叫做最小埃尔布朗模型（Least Herbrand Model）：

若P是一个逻辑程序，且$\{M_i\}$是P的一个非空的埃尔布朗模型集合，那么$\cap_i M_i$也是P的一个埃尔布朗模型。

当$\{M_i\}$是P所有的埃尔布朗模型时，它们的交集即是最小埃尔布朗模型，记为M_p。

同时，我们断言：
若P是一个逻辑程序，那么P的埃尔布朗模型是满足（satisfies）P的一个埃尔布朗解释，即满足P中所有子句的一个真值指派，且这样的模型必然存在，证明如下：

为了构造逻辑程序 P 的模型，我们利用极大一致集的方法。具体步骤如下：

1. **极大一致集**:
   - 定义：一个逻辑程序 P 的 Herbrand 模型可以通过构造一个包含 P 的极大一致集（maximal consistent set）来得到。
   - 设 P 为逻辑程序中的所有子句（闭公式）组成的集合。
2. **构造极大一致集**:
   - 通过包含 P 的极大一致集 M 构造模型 I_M。 M 包含 P 中的所有子句，并且对于 Herbrand 基中的每一个接地原子 A，要么 A 在 M 中，要么 $\neg A$ 在 M 中。并且总保持加进去还是一致的。我们记这个构造出的极大一致集为Δ.
3. **Herbrand 模型**:
   - 定义：Herbrand 模型 I_M 是极大一致集 M 中所有接地原子的集合。
   - 解释：I_M 中包含所有在 M 中为真的接地原子。
4. **验证模型**:
   - 检查 I_M 是否满足逻辑程序 P 中的每一个子句。具体来说，对于每一个子句 C（例如 $A_1 \vee \cdots \vee A_k \vee \neg B_1 \vee \cdots \vee \neg B_n$），需要确保在 I_M 中 A_i（对于某个 i）为真或 B_j（对于某个 j）为假。

验证模型的证明:

要证明 I_M 是逻辑程序 P 的一个模型，我们必须展示对于 P 中的每个子句 C，I_M 满足 C。我们模仿教材，定义一个由所有的命题符号组成的集合上的真值指派v：对任意的符号A，$v(A) = T$ iff $A \in \Delta$. 那么对于任意的子句C,$v(C) = T$ iff $C \in \Delta$.这一点可以对C进行归纳法证明，具体来讲：

（1）归纳基础是：对于原子公式成立。

（2）归纳假设：对于两个子句C_1, C_2都成立。

（3）归纳结论：对于两个子句C_1, C_2的唯二两种构造$\neg$与$\vee$产生的子句也成立。

这里证明与书上习题完全同理，略去.

所以这个极大一致集 I_M 确实满足逻辑程序 P 中的所有子句，因此 I_M 是逻辑程序 P 的一个模型。

证毕!

附:另一种证明方法（简略）

我还在网上找到了戴望洲老师的PPT，提供了一种思路：

具体来说：

在定义最小埃尔布朗模型之后，我们有定理：

若P是一个逻辑程序，那么$M_p = \{A \in B_p | P \vdash A\}$。

再定义：

直接推论算子（Immediate consequence operator）：若P是一个逻辑程序，直接推论算子是一个映射$T_p : 2^{B_p}映射到2^{B_p}$，它有如下定义：令I是一个埃尔布朗解释，那么：

$T_p(I) = \{A \in B_p | A \leftarrow A1 \wedge \cdots \wedge A_n$是中一条规则的具体实例且$\{A_1, \cdots, A_n\} \subset I\}$.

由有定理：

由于T_p是单调的，根据不动点定理，存在不动点令$T_p(I) = I$。而P的最小埃尔布朗模型M_p则是T_p的最小不动点（least fixed point）。

所以只要找到最小不动点，就能找到逻辑程序的模型.

尽管推论可能走向无穷，但是这是可以做到的.

证毕!

参考文献

戴望洲老师PPT

知识表示与推理公开课

命题逻辑形式推演（resolution归结原理