数理逻辑 2026 年度课程作业参考答案

Jun 4

1 第一次作业

1. (a) 设 P 为“今天下雨”，Q 为“今天刮风”，R 为“秦老师去打篮球”。
逻辑公式为：$(P \vee Q) \to \neg R$

(b) 设 P 为“代码没有 Bug”，Q 为“服务器未宕机”，R 为“系统在 5 分钟内运行结束”。
逻辑公式为：$R \to (P \wedge Q)$

(c) 设 P 为“秦老师上数理逻辑课”，Q 为“秦老师上辩证法课”，R 为“今天是周末”。
逻辑公式为：$\neg R \to (P \wedge Q)$ （等价形式 $(P \wedge Q) \vee R$ 亦算正确）

2. 公式 $A = ((P \to Q) \wedge \neg Q) \to \neg P$。

(a) **完整真值表：**

P	Q	$P \to Q$	$\neg Q$	$(P \to Q) \wedge \neg Q$	$\neg P$	**A**
T	T	T	F	F	F	**T**
T	F	F	T	F	F	**T**
F	T	T	F	F	T	**T**
F	F	T	T	T	T	**T**

(b) **判断与理由：**
该公式是**永真式 (Tautology)**。
理由：由上述真值表可知，在命题变量 P, Q 的所有可能的真值赋值组合下，公式 A 的最终计算结果均为 T (True)。

3. 证法一：基于有限长度字符串的证明（字母表法）

1. 构造一个包含所有可能符号的字母表 $\Sigma = PV \cup \{\neg, \vee, \wedge, \to, \leftrightarrow, (,)\}$。
2. 已知 PV 是可数无穷的，逻辑连接词和辅助括号构成的集合是有限的（共 7 个符号）。两个至多可数集的并集仍然是可数集，因此字母表 Σ 是可数无穷的。

3. 设 Σ^* 为基于字母表 Σ 生成的所有有限长度字符串的集合。因为有限长字符串的集合也是可数的，所以 $|\Sigma^*| = \aleph_0$。
4. 命题公式集合 PROP 仅仅是符合特定语法的 Σ^* 的一个子集（即 $\text{PROP} \subseteq \Sigma^*$），因此 $|\text{PROP}| \leq \aleph_0$。
5. 同时，单独的命题变量本身也是合式公式（即 $PV \subseteq \text{PROP}$），故 $|\text{PROP}| \geq |PV| = \aleph_0$。
6. 根据 Cantor-Bernstein-Schroeder (CBS) 定理，可得 $|\text{PROP}| = \aleph_0$。

证法二：基于命题结构的归纳证明（层次构造法）

1. 按照合式公式的归纳定义，按构造层次（深度）定义集合序列 S_n：
   - $S_0 = PV$
   - $S_{n+1} = S_n \cup \{\neg A \mid A \in S_n\} \cup \{(A \circ B) \mid A, B \in S_n, \circ \in \{\vee, \wedge, \to, \leftrightarrow\}\}$
2. 显然，所有命题公式构成的集合 $\text{PROP} = \bigcup_{n=0}^{\infty} S_n$。
3. 使用数学归纳法证明：对任意 $n \geq 0$，S_n 都是可数集。
   - **基础情况：** $S_0 = PV$，已知是可数无穷的，结论成立。
   - **归纳假设：**假设 S_n 是可数集。
   - **归纳步骤：**在构造 S_{n+1} 时，一元运算生成的集合 $\{\neg A \mid A \in S_n\}$ 与 S_n 势相等，是可数的；二元运算生成的集合是笛卡尔积 $S_n \times S_n$ 在 4 种连接词下的有限次并集。由于可数集的笛卡尔积依然可数，故二元组合也是可数的。因此，S_{n+1} 作为有限个可数集的并集，依然是可数的。
4. 因为对所有 n，S_n 均可数，所以可数无穷多个可数集的并集 $\bigcup_{n=0}^{\infty} S_n$ 必定也是可数的。即 $|\text{PROP}| \leq \aleph_0$。
5. 由于 $PV = S_0 \subseteq \text{PROP}$，有 $|\text{PROP}| \geq \aleph_0$。
6. 根据 CBS 定理，得 $|\text{PROP}| = \aleph_0$。

4. 证明：对命题公式 A 的结构进行归纳。

- **基础情况：**当 $A \in PV$（原子命题）时，公式中没有括号。此时 $L(A) = 0$，$R(A) = 0$，显然 $L(A) = R(A)$ 成立。
- **归纳假设：**假设对于命题公式 B 和 C，$L(B) = R(B)$ 且 $L(C) = R(C)$ 均成立。
- **归纳步骤：**
  (a) 当 $A = (\neg B)$ 时：$L(A) = L(B)+1, R(A) = R(B)+1$。代入归纳假设可知，$L(A) = R(A)$。
  (b) 当 $A = (B \circ C)$ 时，其中 $\circ \in \{\vee, \wedge, \to, \leftrightarrow\}$：$L(A) = L(B) + L(C) + 1$，$R(A) = R(B) + R(C) + 1$。代入归纳假设可知，$L(A) = R(A)$。
- **结论：**由结构归纳法得，对于任意 $A \in \text{PROP}$，都有 $L(A) = R(A)$。证明完毕。

P	Q	$Q \to P$	$\mathbf{P} \to (\mathbf{Q} \to \mathbf{P})$
T	T	T	**T**
T	F	T	**T**
F	T	F	**T**
F	F	T	**T**

P	Q	$P \to Q$	$\neg P$	$\neg Q$	$\neg Q \to \neg P$	$(\mathbf{P} \to \mathbf{Q}) \leftrightarrow (\neg\mathbf{Q} \to \neg\mathbf{P})$
T	T	T	F	F	T	**T**
T	F	F	F	T	F	**T**
F	T	T	T	F	T	**T**
F	F	T	T	T	T	**T**

5. (a) $P \to (Q \to P)$

(b) $(P \to Q) \leftrightarrow (\neg Q \to \neg P)$

(c) $(P \vee Q) \to (\neg P \to Q)$

P	Q	$P \vee Q$	$\neg P$	$\neg P \to Q$	$(\mathbf{P} \vee \mathbf{Q}) \to (\neg\mathbf{P} \to \mathbf{Q})$
T	T	T	F	T	**T**
T	F	T	F	T	**T**
F	T	T	T	T	**T**
F	F	F	T	F	**T**

由于上述三个真值表的最终结果均为全 T，三个公式均为永真式。

6. (a) **证明** $(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$

假设公式不为永真式，即存在一种真值赋值使得前件为 T 且后件为 F。

- 由后件 $((P \to Q) \to (P \to R))$ 为 F，必然有：$(P \to Q)$ 为 T，且 $(P \to R)$ 为 F。
- 由 $(P \to R)$ 为 F 推出：$P = \mathrm{T}$，且 $R = \mathrm{F}$。
- 将 $P = \mathrm{T}$ 代入 $(P \to Q) = \mathrm{T}$ 中，必有 $Q = \mathrm{T}$。
- 将 $P = \mathrm{T}, Q = \mathrm{T}, R = \mathrm{F}$ 代入前件 $P \to (Q \to R)$，得到 $\mathrm{T} \to (\mathrm{T} \to \mathrm{F}) = \mathrm{T} \to \mathrm{F} = \mathrm{F}$。
- 前件计算结果为 F，与前件必须为 T 的假设产生矛盾。因此原公式必定是永真式。

(b) **证明** $(\neg P \to \neg Q) \to (Q \to P)$

假设存在赋值使得前件 $(\neg P \to \neg Q)$ 为 T，且后件 $(Q \to P)$ 为 F。

- 由 $(Q \to P)$ 为 F 推出：$Q = \mathrm{T}$，$P = \mathrm{F}$。
- 此时 $\neg Q = \mathrm{F}$，$\neg P = \mathrm{T}$。
- 将其代入前件 $\neg P \to \neg Q$，得到 $\mathrm{T} \to \mathrm{F} = \mathrm{F}$。

- 这与前件必须为 T 矛盾。因此公式为永真式。

(c) **证明** $((P \to Q) \land (Q \to R)) \to (P \to R)$
假设存在赋值使得前件 $((P \to Q) \land (Q \to R))$ 为 T，且后件 $(P \to R)$ 为 F。

- 由后件为 F 推出：$P = \mathrm{T}$，且 $R = \mathrm{F}$。
- 由前件为 T 推出：$(P \to Q)$ 为 T，且 $(Q \to R)$ 为 T。
- 因为 $P = \mathrm{T}$ 且 $P \to Q = \mathrm{T}$，必然得出 $Q = \mathrm{T}$。
- 将 $Q = \mathrm{T}, R = \mathrm{F}$ 代入 $(Q \to R)$，得到 $\mathrm{T} \to \mathrm{F} = \mathrm{F}$。
- 这与 $(Q \to R)$ 必须为 T 产生矛盾。因此公式为永真式。

7. 对于公式 $F = (P \lor Q) \land (\neg P \lor R) \land (\neg Q \lor \neg R)$。

(a) F 是**可满足的 (Satisfiable)**。因为存在至少一种变量赋值使 F 结果为 T。

(b) 为了使得 F 为 T，必须满足所有合取项均为 T：

- 条件 1：$P \lor Q = \mathrm{T}$ （P 和 Q 至少有一个为 T）
- 条件 2：$\neg P \lor R = \mathrm{T}$ （即蕴含式 $P \to R = \mathrm{T}$）
- 条件 3：$\neg Q \lor \neg R = \mathrm{T}$ （即蕴含式 $Q \to \neg R = \mathrm{T}$）

分析推导：

- 假设 $P = \mathrm{T}$，由条件 2 必有 $R = \mathrm{T}$。代入条件 3 得 $\neg Q \lor \mathrm{F} = \mathrm{T}$，故 $Q = \mathrm{F}$。检验赋值 $(P = \mathrm{T}, Q = \mathrm{F}, R = \mathrm{T})$，满足条件 1，均成立。
- 假设 $P = \mathrm{F}$，由条件 1 必有 $Q = \mathrm{T}$。代入条件 3 得 $\mathrm{F} \lor \neg R = \mathrm{T}$，故 $R = \mathrm{F}$。检验赋值 $(P = \mathrm{F}, Q = \mathrm{T}, R = \mathrm{F})$，满足条件 2，均成立。
- 如果 $P = \mathrm{T}$ 且 $Q = \mathrm{T}$，由条件 2 得 $R = \mathrm{T}$，由条件 3 得 $\neg R = \mathrm{T}$，产生矛盾。

结论：使得该命题公式真值为 True 的赋值共有如下两种：

- **赋值 1：** $P = \mathrm{T}, Q = \mathrm{F}, R = \mathrm{T}$
- **赋值 2：** $P = \mathrm{F}, Q = \mathrm{T}, R = \mathrm{F}$

8. (a) **仅使用 NAND (↑) 的等价变形：**
已知公式：$F = (P \lor Q) \land \neg(P \land Q)$
利用分配律展开：

$$F = (P \land \neg(P \land Q)) \lor (Q \land \neg(P \land Q))$$

根据德 · 摩根定律，将最外层的 $\lor$ 转化为 $\land$ 与否定：

$$F = \neg\Big(\neg\big(P \land \neg(P \land Q)\big) \land \neg\big(Q \land \neg(P \land Q)\big)\Big)$$

观察到 $\neg(A \land B)$ 的形式等价于 $A \uparrow B$，我们直接进行替换：

$$F = \Big(P \uparrow (P \uparrow Q)\Big) \uparrow \Big(Q \uparrow (P \uparrow Q)\Big)$$

(b) **(选做) 完全组证明：**

证明： 已知集合 $\{\neg, \vee, \wedge\}$ 是功能完全组，我们只需证明仅用 NAND ($\uparrow$) 或仅用 NOR ($\downarrow$) 即可表达出这三个基本连接词。

对 {NAND} 的证明：

- 非门 (NOT)：$\neg P \equiv \neg(P \wedge P) \equiv P \uparrow P$
- 与门 (AND)：$P \wedge Q \equiv \neg(P \uparrow Q) \equiv (P \uparrow Q) \uparrow (P \uparrow Q)$
- 或门 (OR)：$P \vee Q \equiv \neg(\neg P \wedge \neg Q) \equiv (\neg P) \uparrow (\neg Q) \equiv (P \uparrow P) \uparrow (Q \uparrow Q)$

由于用 NAND 能等价表示出全部三种基本连接词，故 {NAND} 是完全组。

对 {NOR} 的证明：

- 非门 (NOT)：$\neg P \equiv \neg(P \vee P) \equiv P \downarrow P$
- 或门 (OR)：$P \vee Q \equiv \neg(P \downarrow Q) \equiv (P \downarrow Q) \downarrow (P \downarrow Q)$
- 与门 (AND)：$P \wedge Q \equiv \neg(\neg P \vee \neg Q) \equiv (\neg P) \downarrow (\neg Q) \equiv (P \downarrow P) \downarrow (Q \downarrow Q)$

同理可知 {NOR} 也是完全组。证明完毕。

2 第二次作业

1. (a) 在刚执行完 `intros P Q H.` 这一行后：

- **Context (上下文) 包含**：命题变量 `P : Prop` 和 `Q : Prop`，以及假设 `H : P ∨ Q`。
- **当前的 Goal (目标) 是**：`Q ∨ P`。

(b) 关于 `destruct` 与 `(*Step A*)`：

- `destruct H as [HP | HQ]` 这一步会对析取（或）假设 `H` 进行分类讨论，将产生 **2 个**子目标 (Sub-goals)。
- 在代码标注的 `(*Step A*)` 处，Context 中除了 `P, Q` 之外，还包含左分支产生的具体假设 `HP : P`。
- 此时当前的 Goal 依然是：`Q ∨ P`。

2. (a) **`mult 2 2` 的逐步求值 (归约) 过程**：

已知 2 在底层表示为 `S (S O)`（或 `S (S O)`）。

- `mult (S (S O)) (S (S O))` 匹配到分支 `| S n' => plus m (mult n' m)`，其中 `n'` 为 `S O`，`m` 为 `S (S O)`。
- 归约为：`plus (S (S O)) (mult (S O) (S (S O)))`。
- 针对内层的 `mult (S O) (S (S O))`，继续匹配 `S n'` 分支，此时 `n'` 为 `O`。
- 归约为：`plus (S (S O)) (mult O (S (S O)))`。
- 对于 `mult O (S (S O))`，匹配到基本分支 `| O => O`，归约为 `O`。

- 代回原式，得到：`plus (S (S 0)) (plus (S (S 0)) 0)`。
- 根据标准加法 `plus` 的定义，最终归约求值结果为 4，即 `S (S (S (S 0)))`。

(b)
- 在 Coq 中，逻辑上的“命题 (Proposition)”对应编程语言中的**类型 (Type)**。
- 逻辑上的“蕴含关系” $A \to B$ 对应编程语言中的**函数类型 (Function Type)**，即接收一个 A 类型的值作为输入，返回一个 B 类型的值作为输出的函数。

3. (a) 所有计算机系的学生都喜欢数理逻辑课。
公式：$\forall x(C(x) \to L(x, s))$

(b) 存在一个数学系的学生，他既喜欢数理逻辑课，又喜欢编程课。
公式：$\exists x(M(x) \land L(x, s) \land L(x, p))$

(c) 任何喜欢编程课的学生，都不是数学系的学生。
公式：$\forall x(L(x, p) \to \neg M(x))$

4. (a) **可满足性 (Satisfiability)**：

- Γ_1 是**可满足的**。
- **模型构造**：假设模型 $M = (D, I)$，其中论域定义为包含一个员工的集合 $D = \{e_1\}$。根据一阶语言中一元谓词解释为论域子集的定义，我们给出如下语义分配：

$$U^M = \{e_1\}, \quad A^M = \emptyset, \quad R^M = \{e_1\}, \quad W^M = \{e_1\}$$

- **验证**：
  对于规则 (1) $\forall x(A(x) \to U(x))$：因为 $A^M = \emptyset \subseteq U^M$，故成立。
  对于规则 (2) $\forall x(U(x) \to R(x))$：因为 $U^M = \{e_1\} \subseteq R^M = \{e_1\}$，故成立。
  对于规则 (3) $\exists x(U(x) \land \neg A(x) \land W(x))$：论域中存在元素 e_1，满足 $e_1 \in U^M$、$e_1 \notin A^M$ 且 $e_1 \in W^M$，故成立。
  这三条规则在模型 M 下均被解释为真。
- **实际工作对应**：“寻找模型使其为真”的过程对应了软件工程中的**系统状态模拟**或**构造有效测试用例 (Test Case)**，即证明存在一种实际的系统状态可以完美遵守当前设计的所有规范。

(b) **协调性 (Consistency)**：

- 追加规则 (4) 后，新的规范集合 Γ_2 **不再是协调的**。
- **逻辑矛盾**：由规则 (3) 可知，存在一个特定员工（设为 x），他满足 $\neg A(x)$ 为真且 $W(x)$ 为真。但是，由于 $\neg A(x)$ 为真，代入规则 (4) $\forall x(\neg A(x) \to \neg W(x))$ 中，必然推导出 $\neg W(x)$ 为真。这样对于同一个员工 x，推导出了 $W(x) \land \neg W(x)$，构成逻辑矛盾。
- **开发后果**：把内部不协调的规范文档交给程序员，会导致**无法实现的需求冲突**。代码在处理非管理员用户的写请求时会产生二义性（既被规则 3 要求赋予权限，又被规则 4 拦截），最终必然引发越权漏洞或系统崩溃等严重的业务逻辑错误。

5. 公式 $A = \forall x(P(x,y) \to \exists y(Q(x,y) \wedge R(z)))$。

(a)

$$\begin{aligned}
FV(A) &= FV(\forall x(P(x,y) \to \exists y(Q(x,y) \wedge R(z)))) \\
&= FV(P(x,y) \to \exists y(Q(x,y) \wedge R(z))) \setminus \{x\} \\
&= \Big(FV(P(x,y)) \cup FV(\exists y(Q(x,y) \wedge R(z)))\Big) \setminus \{x\} \\
&= \Big(\{x,y\} \cup \big(FV(Q(x,y) \wedge R(z)) \setminus \{y\}\big)\Big) \setminus \{x\} \\
&= \Big(\{x,y\} \cup \big((\{x,y\} \cup \{z\}) \setminus \{y\}\big)\Big) \setminus \{x\} \\
&= \Big(\{x,y\} \cup \{x,z\}\Big) \setminus \{x\} \\
&= \{x,y,z\} \setminus \{x\} = \{y,z\}
\end{aligned}$$

(b) **计算公式 A 的替换结果 $A[t/y]$，其中 $t = f(x,z)$**：

- 我们要将公式 A 中自由出现的 y 替换为 $f(x,z)$。
- **注意名称冲突 (Variable Capture)**：最外层受全称量词 $\forall x$ 约束，而我们要替换进去的项 t 中恰好包含变元 x。直接替换会导致 t 中的 x 被外层量词错误约束。
- **第一步 (换名/α-等价)**：先将最外层的约束变元 x 换为一个新的变元（例如 w）。得到等价公式：
  $A' = \forall w(P(w,y) \to \exists y(Q(w,y) \wedge R(z)))$
- **第二步 (执行替换)**：在 A' 中将自由出现的 y（仅出现在 $P(w,y)$ 中，后方的 y 被 $\exists y$ 约束不是自由变元）替换为 $f(x,z)$。
- **最终结果**：$\forall w(P(w, f(x,z)) \to \exists y(Q(w,y) \wedge R(z)))$。

6. 公式 $C = \big(\forall x \exists y P(x,y) \wedge \forall x \forall y(P(x,y) \to Q(x,y)) \wedge \forall x \forall y(Q(x,y) \to R(x,y))\big) \to \forall x \exists y R(x,y)$。

(b) **永真性 (Validity)**：

- C 是**永真的**。
- **语义证明**：要证明 C 是永真式，即证明对于任意模型 $M = (D, I)$ 和任意赋值 σ，均有 $M \models_\sigma C$。
- **第一步：语义展开**
  根据蕴含式 $\to$ 的语义定义，这等价于：假设 $M \models_\sigma$ 前件，我们需要证明 $M \models_\sigma$ 后件。
  根据合取式 $\wedge$ 的语义定义，前提成立意味着以下三个条件同时成立：
  1. $M \models_\sigma \forall x \exists y P(x,y)$：对任意 $d \in D$，存在 $d' \in D$ 使得 $M \models_{\sigma[x:=d, y:=d']} P(x,y)$，即有序对 $(d,d') \in P^M$。
  2. $M \models_\sigma \forall x \forall y(P(x,y) \to Q(x,y))$：对任意 $d_1, d_2 \in D$，若 $M \models_{\sigma[x:=d_1, y:=d_2]} P(x,y)$，则 $M \models_{\sigma[x:=d_1, y:=d_2]} Q(x,y)$。即若 $(d_1,d_2) \in P^M$，则 $(d_1,d_2) \in Q^M$（也即 $P^M \subseteq Q^M$）。
  3. $M \models_\sigma \forall x \forall y(Q(x,y) \to R(x,y))$：同理展开，对任意 $d_1, d_2 \in D$，若 $M \models_{\sigma[x:=d_1, y:=d_2]} Q(x,y)$，则 $M \models_{\sigma[x:=d_1, y:=d_2]} R(x,y)$（也即 $Q^M \subseteq R^M$）。

对于目标结论 $\forall x \exists y R(x,y)$，根据量词语义展开，我们需要证明：对任意 $d \in D$，存在 $d'' \in D$ 使得 $M \models_{\sigma[x:=d,y:=d'']} R(x,y)$，即证明有序对 $(d,d'') \in R^M$。

- **第二步：谓词取值讨论**
  由展开后的条件 (1) 可知，任取论域中的元素 $d \in D$，必然存在一个元素 $d' \in D$，使得有序对 $(d,d') \in P^M$。
  将该有序对 (d,d') 代入条件 (2)，由于 $P^M \subseteq Q^M$，必然有 $(d,d') \in Q^M$。
  再将 (d,d') 代入条件 (3)，由于 $Q^M \subseteq R^M$，必然得出 $(d,d') \in R^M$。
  因此，对于任意给定的 $d \in D$，我们确实找到了一个元素（即 $d'' = d'$），使得 $(d,d'') \in R^M$ 成立。目标结论的语义要求得到满足，故 C 在任意模型下均为真，是永真式。

(a) **可满足性 (Satisfiability)**：

- C 是**可满足的**。因为根据 (b) 的证明 C 是永真的，那么它必然是可满足的。
- **模型构造**：我们构造一个模型 $M = (D, I)$。取论域 $D = \{1\}$。对于二元谓词 P, Q, R，我们使用有序对集合的方法给出其语义解释：

$$P^M = \{(1,1)\}, \quad Q^M = \{(1,1)\}, \quad R^M = \{(1,1)\}$$

- 在此模型 M 中，前提与结论的语义显然均成立，因此公式 C 在该模型下为真，证明了其可满足性。

7. 在标准算术模型 $M = (\mathbb{N}, I)$ 中：

(a) $\forall x \exists y (x < y)$：

- **真 (True)**。
- **解释**：自然数集是无穷大的，对于任意一个给定的自然数 x，我们总能找到一个比它大的自然数 y（例如 $y = x + 1$）。

(b) $\exists x \forall y (x + y = y)$：

- **真 (True)**。
- **解释**：在自然数中存在加法的单位元 0，使得对于任意自然数 y，等式 $0 + y = y$ 均成立。

(c) $\exists x \forall y (x \cdot y = y)$：

- **真 (True)**。
- **解释**：在自然数中存在乘法的单位元 1，使得对于任意自然数 y，等式 $1 \cdot y = y$ 均成立。

8. (选做) 语义定义与替换引理推导

- **已知**：模型 M 下公式 $\forall x A$ 为真，即 $M \models \forall x A$。
- **证明**：对于任意项 t，$M \models A[t/x]$。

1. 根据语义定义，$M \models \forall x A$ 意味着：对于模型 M 的论域 D 中的**任意**元素 d，都有 $M \models_{\sigma[x:=d]} A$ 成立（其中 σ 为任意赋值）。
2. 对于任意合法的项 t，在给定的模型 M 和赋值 σ 下，它总是能够被解释并映射为论域 D 中的某个具体元素，记为 $d' = I_\sigma(t)$。
3. 因为 $M \models_{\sigma[x:=d]} A$ 对全体论域元素均成立，它特别地对元素 d' 也必定成立。因此我们有 $M \models_{\sigma[x:=d']} A$，即 $M \models_{\sigma[x:=I_\sigma(t)]} A$。
4. 根据**替换引理 (Substitution Lemma)**：语法层面的变元替换等价于语义层面的环境赋值更新，即 $M \models_\sigma A[t/x]$ 当且仅当 $M \models_{\sigma[x:=I_\sigma(t)]} A$。
5. 由此直接推导出 $M \models_\sigma A[t/x]$，即 $M \models A[t/x]$ 成立。证明完毕。

3 期中作业

1. (a) 设定命题变量：T 表示 “机房温度超过 30 度”，A 表示 “空调开启”，R 表示 “触发警报”。逻辑公式为：$F = (T \land \neg A) \to R$。

(b) 真值表如下：

T	A	R	$\neg A$	$T \land \neg A$	$\mathbf{F} : (\mathbf{T} \land \neg \mathbf{A}) \to \mathbf{R}$
T	T	T	F	F	**T**
T	T	F	F	F	**T**
T	F	T	T	T	**T**
T	F	F	T	T	**F**
F	T	T	F	F	**T**
F	T	F	F	F	**T**
F	F	T	T	F	**T**
F	F	F	T	F	**T**

判断：公式 F 是**可满足式 (Satisfiable)**。它在多数赋值下为真，但在 $T = \mathrm{T}, A = \mathrm{F}, R = \mathrm{F}$ 时为假，因此既不是重言式也不是矛盾式。

(c) 根据真值表，当 $R = \mathrm{F}$ 且公式 $F = \mathrm{T}$ 时，机房 (T, A) 的状态可能有以下三种：

1. $T = \mathrm{F}, A = \mathrm{F}$ （温度正常，空调未开）
2. $T = \mathrm{F}, A = \mathrm{T}$ （温度正常，空调开启）
3. $T = \mathrm{T}, A = \mathrm{T}$ （温度超标，但空调已开启）

自然语言解释：在警报器损坏无法拉响的情况下，系统没有违反安全规范的原因是——此时机房温度本身并没有超过 30 度（处于安全状态）；或者即使温度超过了 30 度，空调也已经处于开启状态（正在积极降温），从而破坏了违规的前置条件 “高温且无空调”。

2. (a) 我们构造一个解释模型 $M = (D, I)$，设定论域 $D = \{1\}$（代表一条特定的代码提交记录）。对一元谓词给出如下解释：$P^M = \{1\}$, $C^M = \{1\}$, $E^M = \{1\}$。

验证：

- 规则 (1) $\forall x(P(x) \to C(x))$：因为 $P^M \subseteq C^M$，公式在 M 下为真。
- 规则 (2) $\exists x(P(x) \wedge E(x))$：论域中存在元素 1，满足 $1 \in P^M$ 且 $1 \in E^M$，公式在 M 下为真。

因此，规范集合 Γ_1 在该模型 M 下可被同时满足。

(b) 新的规范集合 Γ_2 **是不协调的（存在矛盾）**。
推演过程：由规则 (2) 可知，存在一条具体的提交记录（设为 a），满足 $P(a)$ 且 $E(a)$ 为真。
根据合取消除，我们有 $P(a)$ 为真，且 $E(a)$ 为真。
将 $P(a)$ 代入规则 (1) 的特化实例 $P(a) \to C(a)$ 中，通过肯定前件，推导出 $C(a)$ 为真。
将 $E(a)$ 代入规则 (3) 的特化实例 $E(a) \to \neg C(a)$ 中，同样通过肯定前件，推导出 $\neg C(a)$ 为真。
对同一份提交记录 a，系统同时推导出了 $C(a) \wedge \neg C(a)$，构成自相矛盾。

3. 证明：对命题公式 $A \in \mathrm{PROP}^*$ 的结构进行归纳。

- **基础情况**：当 $A \in PV$（原子命题符）时，命题符出现 1 次，没有二元连接词。即 $v(A) = 1$，$c(A) = 0$。显然 $1 = 0 + 1$ 成立。
- **归纳假设**：假设对于任意合式的命题公式 B 和 C，等式均成立，即 $v(B) = c(B) + 1$ 且 $v(C) = c(C) + 1$。
- **归纳步骤**：当 A 是通过二元连接词 $\circ \in \{\wedge, \vee, \to, \leftrightarrow\}$ 由 B 和 C 构造而成，即 $A = (B \circ C)$ 时：
  - 命题符总数 $v(A) = v(B) + v(C)$。
  - 二元连接词总数 $c(A) = c(B) + c(C) + 1$ （其中“+1”为新增的连接词 $\circ$）。

  将归纳假设代入 $v(A)$ 中：

$$v(A) = (c(B) + 1) + (c(C) + 1) = (c(B) + c(C) + 1) + 1$$

  将 $c(A)$ 代入上式，得到 $v(A) = c(A) + 1$。
- **结论**：由结构归纳法得，对于任意 $A \in \mathrm{PROP}^*$，都有 $v(A) = c(A) + 1$。证明完毕。

4. 公式 $D = \forall x(P(x) \vee Q(x)) \to (\forall x P(x) \vee \forall x Q(x))$

(a) **构造使公式为假的模型** M_1：
要使蕴含式为假，必须让前件为真且后件为假。
设定论域 $D_1 = \{1, 2\}$。解释为：$P^{M_1} = \{1\}$，$Q^{M_1} = \{2\}$。
验证：论域中所有元素（1 和 2），要么属于 P 要么属于 Q，故前件 $\forall x(P(x) \vee Q(x))$ 为真。
但在后件中，$\forall x P(x)$ 为假（元素 2 不满足），$\forall x Q(x)$ 也为假（元素 1 不满足）。因此后件为假，公式整体在 M_1 下为假。

(b) **构造使公式为真的模型** M_2：
设定论域 $D_2 = \{1\}$。解释为：$P^{M_2} = \{1\}$，$Q^{M_2} = \emptyset$。

验证：因为论域只有 1 个元素，且它满足 P，故 $\forall x P(x)$ 为真。那么后件 $\forall x P(x) \lor \forall x Q(x)$ 必然为真。由于蕴含式的后件为真，无论前件如何，整体公式在 M_2 下必定为真。

5. 使用 G 系统构建证明树如下：

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Ax}}{Q \to R, P \vdash P, R} \quad \dfrac{\text{Ax}}{R, Q \to R, P \vdash R}}{P \to R, Q \to R, P \vdash R} \to L \quad \dfrac{\dfrac{\text{Ax}}{P \to R, Q \vdash Q, R} \quad \dfrac{\text{Ax}}{P \to R, R, Q \vdash R}}{P \to R, Q \to R, Q \vdash R} \to L}{P \to R, Q \to R, P \lor Q \vdash R} \lor L}{(P \to R) \land (Q \to R), P \lor Q \vdash R} \land L}{(P \to R) \land (Q \to R) \vdash P \lor Q \to R} \to R}{\vdash (P \to R) \land (Q \to R) \to (P \lor Q \to R)} \to R$$

6. 使用 G 系统构建证明树如下（a 为引入的新自由变元）：

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Ax}}{P(a) \vdash P(a)}}{P(a) \vdash \exists x P(x)} \exists R}{P(a) \vdash \exists x P(x) \lor \exists x Q(x)} \lor R_1 \quad \dfrac{\dfrac{\dfrac{\text{Ax}}{Q(a) \vdash Q(a)}}{Q(a) \vdash \exists x Q(x)} \exists R}{Q(a) \vdash \exists x P(x) \lor \exists x Q(x)} \lor R_2}{P(a) \lor Q(a) \vdash \exists x P(x) \lor \exists x Q(x)} \lor L}{\exists x (P(x) \lor Q(x)) \vdash \exists x P(x) \lor \exists x Q(x)} \exists L}{\vdash \exists x (P(x) \lor Q(x)) \to \exists x P(x) \lor \exists x Q(x)} \to R$$

7. $\Gamma = \{(P \lor Q) \to R, P, \neg R\}$

(a) **判断**：该公式集合是**不协调的 (Inconsistent)**。

(b) 证明一个前提集合 Γ 是不协调的，等价于在 G 系统中证明序贯 $\Gamma \vdash \emptyset$ 成立。

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{Ax}}{P \vdash P, R}}{P \vdash P \lor Q, R} \lor R_1 \quad \dfrac{\text{Ax}}{R, P \vdash R}}{(P \lor Q) \to R, P \vdash R} \to L}{(P \lor Q) \to R, P, \neg R \vdash} \neg L$$

(c) 由于需求存在内在矛盾，没有任何一种合法状态可以满足所有的规范。在代码实现中，这会导致处理该逻辑的代码分支成为**永远无法执行的死代码 (Dead Code)**；或者在系统接收到特定合法输入时，触发断言失败、死锁、无限循环等**崩溃行为**，因为程序无法同时兼顾互相抵触的约束条件。

8. (a) 模型 $M = (\mathbb{N}, I)$，$P^M = \{(u, v) \mid u > v\}$，$I(f)(u) = u + 2$。
赋值 $\sigma(y) = 3$。公式 $A = \exists z P(x, z)$。项 $t = f(y)$。

- **计算项的语义值**：$t_{M[\sigma]} = I_\sigma(f(y)) = I(f)(\sigma(y)) = 3 + 2 = 5$。

- **计算** $M \models_{\sigma[x:=5]} \exists z P(x,z)$：
  判断是否存在 $d \in \mathbb{N}$ 使得有序对 $(5,d) \in P^M$（即 $5 > d$）。由于存在这样的自然数（如 $d=4$），故该语义值为 **True**。
- **计算替换后公式** $M \models_\sigma A[t/x]$：
  $A[t/x] = \exists z P(f(y), z)$。
  判断是否存在 $d \in \mathbb{N}$ 使得有序对 $(I_\sigma(f(y)), d) \in P^M$。因为 $I_\sigma(f(y)) = 5$，即判断是否存在 d 使得 $5 > d$。同理存在（如 $d=4$），故该语义值为 **True**。

两者真值一致，替换引理在该实例下得到验证。

(b)
- **已知**：$M \models_\sigma \forall x A$。
- **证明**：对于任意合法的项 t，$M \models_\sigma A[t/x]$。
  1. 根据语义定义，$M \models_\sigma \forall x A$ 意味着：对于模型 M 的论域 D 中的**任意**元素 d，都有 $M \models_{\sigma[x:=d]} A$ 成立。
  2. 对于任意合法的项 t，在给定的模型 M 和赋值 σ 下，它总是能够被解释并映射为论域 D 中的某个具体元素，记为 $d' = I_\sigma(t)$。
  3. 因为 $M \models_{\sigma[x:=d]} A$ 对全体论域元素均成立，它特别地对元素 d' 也必定成立。因此我们有 $M \models_{\sigma[x:=d']} A$，即 $M \models_{\sigma[x:=I_\sigma(t)]} A$。
  4. 根据**替换引理 (Substitution Lemma)**：语法层面的变元替换等价于语义层面的环境赋值更新，即 $M \models_\sigma A[t/x]$ 当且仅当 $M \models_{\sigma[x:=I_\sigma(t)]} A$。
  5. 由此直接推导出 $M \models_\sigma A[t/x]$ 成立。证明完毕。

9. (a) 当执行到 `(* 状态分支 2*)` 这一行刚结束时：

- Context 中包含：命题变量 `P : Prop`, `Q : Prop`, `R : Prop`，以及假设 `HP : P` 和 `HR : R`。

(b) 在 `(* 状态分支 2*)` 处，Goal 最初是：`(P ∧ Q) ∨ (P ∧ R)`。

- 执行了 `right.` 之后，Goal 变为只需证明右侧的析取项：`P ∧ R`。
- 执行了 `split.` 之后，Goal 被进一步拆分为**两个子目标**：一个是证明 `P`，另一个是证明 `R`。

(c) 代码填空：

- `(* 填空处 A*)`：`apply HP.`
- `(* 填空处 B*)`：`apply HR.`

10. (a) 定义有理数（分数）类型 `frac`：

```
Inductive frac : Type :=
  | mk_frac : nat -> nat -> frac.
```

(b) 编写分数除法函数 `div_frac`：

```
Definition div_frac (f1 f2 : frac) : frac :=
  match f1, f2 with
  | mk_frac a b, mk_frac c d => mk_frac (mult a d) (mult b c)
  end.
```

(c) **Bonus：防范分母为 0 的解决思路**

在 Coq 严格的类型系统中，为了防止分母为零，常见的工程与理论手段有两种：

1. **依赖类型 (Dependent Types)**：在定义 `frac` 构造器时，强制要求传入一个证明项（如 `b <> 0` 的逻辑证明），将数据与证明捆绑在一起，从根本上杜绝非法数据。
2. **Option 错误处理机制**：将除法函数的返回值类型从 `frac` 更改为 `option frac`。在 `match` 计算结束后检查分母 `mult b c`，若为 0 则返回 `None`，否则返回 `Some (mk_frac ...)`。通过偏函数机制显式处理异常。

4 第三次作业

1. (a) 已知程序的安全规范可以表示为包含无限多条一阶逻辑公式的集合 $\Sigma = \{S_0, S_1, S_2, \dots\}$。由于代码分析工具已经证明：对于任意给定的有限正整数 k，程序在前 k 次迭代内是绝对安全的。这意味着，对于公式集 Σ 的**任意有限子集** $\Gamma \subseteq \Sigma$（比如 $\Gamma = \{S_0, S_1, \dots, S_k\}$），都存在一个系统状态轨迹模型使其为真，即 Γ 是**可满足的 (Satisfiable)**。

根据一阶逻辑的**紧性定理 (Compactness Theorem)**：如果一个公式集合的任意有限子集都是可满足的，那么该无限公式集整体也必然是可满足的。

因此，无限公式集 Σ 整体是可满足的，这意味着存在一个无限的执行轨迹模型，在该模型下程序在未来的无限循环中也绝对不会发生数组越界或内存泄漏。

(b) Dijkstra 的断言反映了物理世界中“穷举测试”的局限性：我们无法在有限的物理时间内跑完无限的循环。

紧性定理在数理逻辑中提供了一种“从局部向全局跃迁”的数学信仰：在理想的理论前提下，只要我们有能力（例如通过符号执行或数学归纳法）泛化地证明程序的**任意有限步长度的切片**都是正确的，我们就能在数学上严丝合缝地推断出它在**无限的生命周期**中也是安全的。紧性定理在理论上跨越了“有限步穷举”与“无穷无尽”之间的鸿沟。

2. (a) 已知一阶公理集合 Γ 拥有一个无限模型，即标准自然数模型 $M_{std} = (\mathbb{N}, +, \cdot, <, 0, 1)$。该模型的论域基数为 $\aleph_0$（可数无穷）。

根据**向上勒文海姆-斯科伦定理 (Upward Löwenheim-Skolem Theorem)**：如果一个具有至多可数词汇表的一阶理论拥有一个无穷模型，那么对于任意大于等于 $\aleph_0$ 的无限基数 κ，该理论都必然拥有一个基数为 κ 的模型。

因此，我们取不可数基数 $\kappa = \aleph_1$（或实数集的基数 $\beth_1$），必然存在一个论域大小为不可数无穷的模型 $M_{non-std}$，它同样完美满足理论 Γ。由于真实的自然数集是可数的，这个具有不可数论域的模型 $M_{non-std}$ 绝不可能是我们心目中的标准自然数模型（它必然包含了无穷多个无法通过 $+1$ 抵达的“非标准超自然数”）。

(b) 一阶逻辑在刻画无限结构时存在内在的“表达力缺陷”（无法做到范畴性 Categoricity）。无论我们写出多么复杂、多么完备的一阶公理，都无法在同构意义上唯一地锁定（界定）像 $\mathbb{N}$ 或 $\mathbb{R}$ 这样的无限数据结构。必然会有无数个长得奇形怪状的“非标准模型”混进来，它们虽然满足我们写下的所有一阶语法规则，但在物理或数学直觉上却与我们的目标数据结构大相径庭。

3. 给定公式 $A = \forall x \exists y (P(x,y) \to \forall z \exists w (Q(x,y,z) \land \neg R(w)))$。

(a) 1. 消除蕴含词 $\to$：
$\forall x \exists y (\neg P(x,y) \lor \forall z \exists w (Q(x,y,z) \land \neg R(w)))$

2. 提取量词至最外层（由于前后量词变元无冲突，直接提取）：
$\forall x \exists y \forall z \exists w (\neg P(x,y) \lor (Q(x,y,z) \land \neg R(w)))$

(b) 在前束范式中消除存在量词 $\exists y$ 和 $\exists w$。

1. 存在量词 $\exists y$ 受前面的全称量词 $\forall x$ 约束，因此引入一个一元斯科伦函数 $f(x)$ 替代 y。
2. 存在量词 $\exists w$ 受前面的全称量词 $\forall x$ 和 $\forall z$ 约束，因此引入一个二元斯科伦函数 $g(x,z)$ 替代 w。

斯科伦化后的最终公式（仅含全称量词且通常省略写出）：
$\forall x \forall z \big(\neg P(x, f(x)) \lor (Q(x, f(x), z) \land \neg R(g(x,z)))\big)$

注：函数 f 依赖于变量 x；函数 g 依赖于变量 x, z。

4. 给定常元符 c，一元函数符 $f(x)$，二元函数符 $g(x,y)$。

(a)
- $H_0 = \{c\}$
- $H_1 = H_0 \cup \{f(t) \mid t \in H_0\} \cup \{g(t_1,t_2) \mid t_1, t_2 \in H_0\}$
  $= \{c, f(c), g(c,c)\}$
- H_2 包含 H_1 的所有元素，以及对 H_1 中元素施加一层函数调用的所有组合。其代表性元素如：
  $H_2 = \{c, f(c), g(c,c), f(f(c)), f(g(c,c)), g(c, f(c)), g(f(c), c), g(f(c), f(c)), \dots\}$

(b) 埃尔布朗域的伟大之处在于它实现了从“无限且抽象的语义世界”向“纯语法符号世界”的降维。在判断一组子句的可满足性时，我们原本需要在一阶逻辑的任意数学解释模型（实数域、树结构、图论网络等）中去寻找答案，犹如大海捞针。埃尔布朗定理告诉我们，如果公式在某个神秘的数学模型中成立，那么它在由它**自身的语法符号“拼积木”一样搭出来的词语世界（埃尔布朗模型）**中也必然成立。这就把一个极其抽象的语义问题，转化成了计算机可以执行的“语法符号模式匹配”问题。

5. 给定子句集 $S = \{P(a), \neg P(x) \lor P(f(x)), \neg P(f(f(a)))\}$。

(a) 初始常元为 a，一元函数为 f。
$H = \{a, f(a), f(f(a)), f(f(f(a))), \dots\}$

(b) 我们从子句 $\neg P(x) \vee P(f(x))$ 中，通过从埃尔布朗域中依次提取变元 x 的值，生成以下有限个基实例（Ground Instances）子集：

(1) $P(a)$ （原样提取第一个子句）

(2) 取 $x = a$ 代入：$\neg P(a) \vee P(f(a))$

(3) 取 $x = f(a)$ 代入：$\neg P(f(a)) \vee P(f(f(a)))$

(4) $\neg P(f(f(a)))$ （原样提取第三个子句）

命题逻辑推理证明：

- 由 (1) 知 $P(a)$ 的真值为 T。
- 结合 (1) 和 (2)，通过分离规则 (Modus Ponens，或析取三段论)，推导出 $P(f(a))$ 真值为 T。
- 结合推导出的 $P(f(a))$ 和 (3)，同理推导出 $P(f(f(a)))$ 真值为 T。
- 但由 (4) 知 $\neg P(f(f(a)))$ 真值为 T，即要求 $P(f(f(a)))$ 真值为 F。
- **得出矛盾！**

因此，根据命题逻辑法则，这组由有限个基实例构成的子集是不一致的。由埃尔布朗定理可知，由于找到了一个不可满足的有限基实例子集，原一阶子句集 S 也是不可满足的。