

11.3 习题解析

1. 化为子句形式有哪些步骤？请结合例子说明。

参考答案：

将公式化成标准子句形式有 9 个步骤。

- (1) 用 $\neg A \vee B$ 取代 $A \rightarrow B$, 消去“ \rightarrow ”符号。
 - (2) 降低 \neg 符号的辖域, 直到原子公式之前或消去 \neg 符号。
 - (3) 变量标准化。重新命名哑变量 (Dummy Variable), 以保证每个量词有自己唯一的变量名。
 - (4) 将公式变为前束范式 (Prefix)。
 - (5) 消去存在量词, 用 Skolem 常数或 Skolem 函数代替存在量词所量化变量的每个出现。
 - (6) 消去全称量词。这样一个公式变成无量词公式了。
 - (7) 重复利用分配律, 变公式为析取式的合取式。
 - (8) 消去“ \wedge ”连词, 使公式成为若干子句。
 - (9) 将变量换名, 使同一个变量符不会出现在两个和两个以上的子句中。
- 其实例见教材中的举例。

2. 确定下列子句间的归类关系。

$$C_1 = P(x, y) \vee Q(z)$$

$$C_2 = Q(A) \vee P(B, B) \vee R(u)$$

$$C_3 = P(x, y) \vee R(z)$$

参考答案:

由归类定义, 设有两个子句 L, M , 若存在一个代换 s , 使得 Ls 所包含的文字为 M 包含的文字的子集, 则说 L 把 M 归类。在上述子句中, 将 C_1 中 $P(x, y)$ 的 x, y 都代换为 B , 则为 $P(B, B)$, z 替换为 A 即:

存在代换 $s_1 = \{ A/z, B/x, B/y \}$

使得 $C_1 s_1 = P(B, B) \vee Q(A)$ 的文字成为 C_2 中的文字的一个子集

所以 C_1 归类 C_2

因此, C_1 内含 C_2 。

同理, C_3 内含 C_2 , 此时, 要将 C_3 中寻找代换。

即, $\exists s_2 = \{ u/z, B/x, B/y \}$

使得 $C_3 s_2 = P(B, B) \vee R(u)$ 成为 C_2 的一个子集。

所以 C_3 归类 C_2

C_1 与 C_3 无归类关系。

3. 设子句集 $S = \{ P(x), \neg P(x) \vee Q(x, A), \neg Q(A, y) \}$

试求:

(1) S 的 $H(S)$ 。

(2) S 的 $HB(S)$ 。

(3) S 的一棵完全语义树。

(4) S 的一棵封闭语义树。

参考答案:

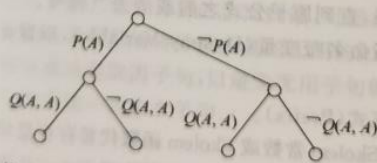
(1) S 中有唯一的常量 A , 且无函数, 因此

$$H(S) = \{ A \}$$

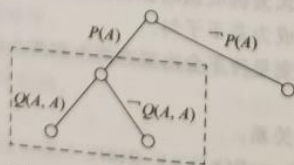
(2) 将 $H(S)$ 中的元素代入 S 中的原子公式中, 得到

$$HB(S) = \{ P(A), Q(A, A) \} \quad \text{—— Herbrand Base}$$

(3) 将 $HB(S)$ 中的元素的正负文字分别作为完全语义树的边, 得到完全语义树为



(4) 在完全语义树上找失效点后, 得到封闭语义树为



可以看出, 虚框内的两个文字归结后, 根节点成为失效点, 即能够使得 S 为假。

4. 此题省略。

5. 给定下述语句:

John like all kinds of food.

Apples are food.

Anything anyone eats and isn't killed by is food.

Bill eats peanuts and is still alive.

Sue eats everything Bill eats.

(1) 把这些句子翻译成逻辑谓词中的公式。

(2) 把(1)的公式转换成子句。

(3) 用归结证明 John likes peanuts.

(4) 回答 Sue 吃什么。

参考答案:

(1) 先将对应的命题表示为逻辑公式

① $\forall x(\text{Food}(x) \rightarrow \text{Like}(\text{John}, x))$

② $\text{Food}(\text{Apple})$

③ $\forall x \forall y(\text{Eat}(x, y) \wedge \neg \text{Killedby}(x, y) \rightarrow \text{Food}(y))$

④ $\text{Eat}(\text{Bill}, \text{peanut}) \wedge \neg \text{Killedby}(\text{Bill}, \text{peanut})$

⑤ $\forall x(\text{Eat}(\text{Bill}, x) \rightarrow \text{Eat}(\text{Sue}, x))$

(2) 将逻辑公式改写成子句

① $\neg \text{Food}(x) \vee \text{Like}(\text{John}, x)$

② $\text{Food}(\text{Apple})$

③ $\neg \text{Eat}(x, y) \vee \text{Killedby}(x, y) \vee \text{Food}(y)$

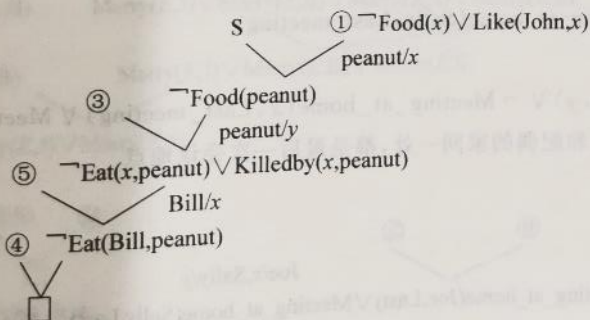
④ $\text{Eat}(\text{Bill}, \text{peanut})$

⑤ $\neg \text{Killedby}(\text{Bill}, \text{peanut})$

⑥ $\neg \text{Eat}(\text{Bill}, x) \vee \text{Eat}(\text{Sue}, x)$

(3) 本题要证: $\text{Like}(\text{John}, \text{peanut})$

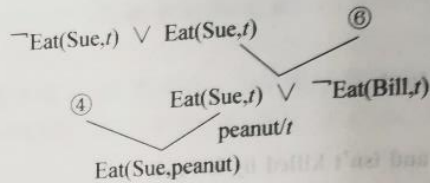
假设 S 为 $\neg \text{Like}(\text{John}, \text{peanut})$



(4) 用回答问题的方式, 求 Sue 吃什么。

引入子句: $\neg \text{Eat}(\text{Sue}, t) \vee \text{Eat}(\text{Sue}, t)$

其中 $\text{Eat}(\text{Sue}, t)$ 只进行代换, 不参加归结



6. 给定下述事实:

Joe, Sally, Bill 和 Ellen 都是桥牌俱乐部的成员。

Joe 同 Sally 结婚。

Bill 是 Ellen 的兄弟。

俱乐部中每位已婚者的配偶也在俱乐部中。

俱乐部的最后一次会议在 Joe 家举行。

(1) 用谓词逻辑表达这些事实。

(2) 根据上面给出的事实, 确定下述语句的真假。

① 俱乐部的最后一次会议在 Sally 家举行。

② Ellen 未婚。

(3) 借助上面给出的 5 个事实, 能用归结证明这两个语句的真假吗? 若能, 请给出归结过程; 否则, 增加需要的事实, 然后在构造证明。

参考答案:

用谓词逻辑表达以上知识。

① $\text{Member}(\text{Joe}) \wedge \text{Member}(\text{Sally}) \wedge \text{member}(\text{Billy}) \wedge \text{Member}(\text{Ellen})$

② $\text{Marry}(\text{Joe}, \text{Sally})$

③ $\text{Brother}(\text{Bill}, \text{Ellen})$

④ $\neg \text{Member}(x) \vee \neg \text{Marry}(x, y) \vee \text{Marry}(x, \text{Joe}) \vee \text{Marry}(x, \text{Ellen}) \vee \text{Marry}(x, \text{Bill})$

⑤ $\text{Meeting_at_home}(\text{Joe}, \text{Last_meeting})$

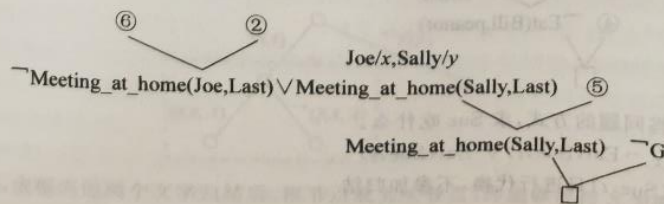
目标的否定

$\neg G: \neg \text{Meeting_at_home}(\text{Sally}, \text{Last_meeting})$

增加知识

⑥ $\neg \text{Marry}(x, y) \vee \neg \text{Meeting_at_home}(x, \text{Last_meeting}) \vee \text{Meeting_at_home}(y, \text{Last_meeting})$ // 和配偶的家同一处, 都是最后一次会议地点

归结过程:

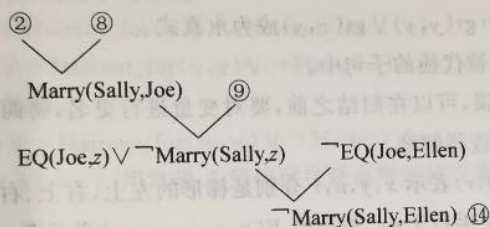
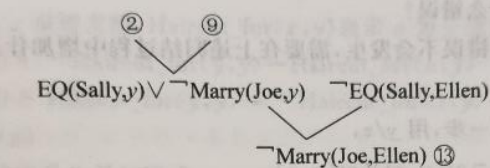


$S = \text{Marry}(\text{Ellen}, \text{sb})$

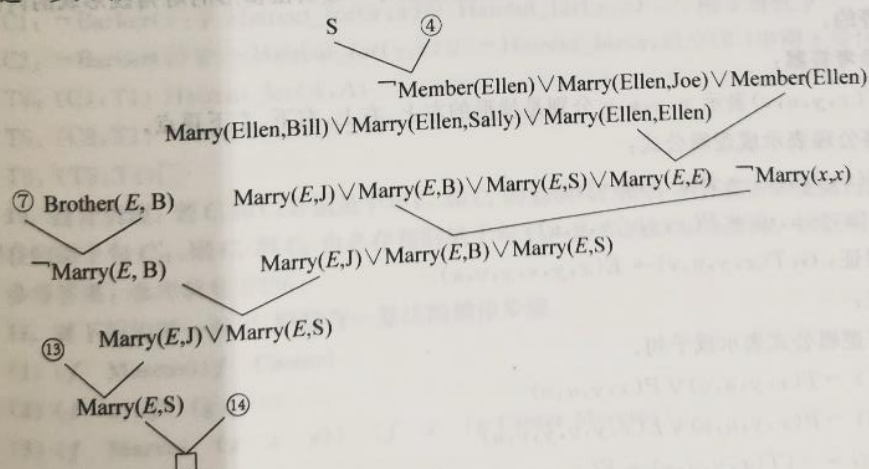
增加知识

- ⑦ $\neg \text{Brother}(x, y) \vee \neg \text{Marry}(x, y)$ // 不能与其兄弟结婚
- ⑧ $\neg \text{Marry}(x, y) \vee \text{Marry}(y, x)$ // 对称性
- ⑨ $\neg \text{Marry}(x, y) \vee \text{EQ}(y, z) \vee \neg \text{Marry}(x, z)$ // 一个人最多只能和一名配偶结婚
- ⑩ $\neg \text{Marry}(x, x)$ // 不能和自己结婚
- ⑪ $\neg \text{EQ}(\text{Joe}, \text{Ellen})$
- ⑫ $\neg \text{EQ}(\text{Sally}, \text{Ellen})$

推理步骤如下。



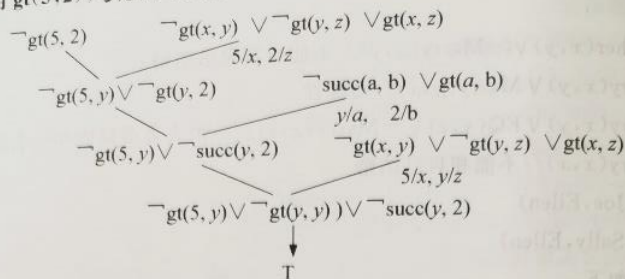
进一步归结证明：



7. 假设给定下述事实。

- $\forall x \forall y \forall z \text{gt}(x, y) \wedge \text{gt}(y, z) \rightarrow \text{gt}(x, z)$
- $\forall a \forall b \text{succ}(a, b) \rightarrow \text{gt}(a, b)$
- $\forall x \text{gt}(x, x)$

若要证明 $gt(5, 2)$, 考虑如下归结证明尝试。



(1) 以上证明有什么错误?

(2) 为了保证这些错误不会发生, 需要在上述归结过程中增加什么?

参考答案:

(1) 在归结的最后一步, 用 y/z ,

使得 $\neg gt(x, y) \vee \neg gt(y, z) \vee gt(x, z)$

成为 $\neg gt(x, y) \vee \neg gt(y, y) \vee gt(x, y)$ 成为永真式

原因是 y 已出现在被代换的子句中。

(2) 要改进这种错误, 可以在归结之前, 要对变量进行更名, 将两个子句中相同的变量名, 换成不同的名称, 再进行归结。

8. 用谓词 $T(x, y, u, v)$ 表示 x, y, u, v 分别是梯形的左上、右上、右下、左下顶点, 用谓词 $P(x, y, u, v)$ 表示线段 xy 平行于 uv , 用谓词 $E(x, y, z, u, v, w)$ 表示角 xyz 等于角 uvw ; 用公理“梯形的上边与下边平行”及“平行线的内错角相等”, 证明由梯形的对角线形成的内错角是相等的。

参考答案:

$T(x, y, u, v)$ 表示 x, y, u, v 分别是梯形的左上、右上、右下、左下顶点。

将公理表示成逻辑公式:

$T(x, y, u, v) \rightarrow P(x, y, u, v);$

$P(x, y, u, v) \rightarrow E(x, y, v, y, v, u)$

要证: $G: T(x, y, u, v) \rightarrow E(x, y, v, y, v, u)$

证:

将逻辑公式表示成子句。

(1) $\neg T(x, y, u, v) \vee P(x, y, u, v)$

(2) $\neg P(x, y, u, v) \vee E(x, y, v, y, v, u)$

$\neg G = \neg (T(x, y, u, v) \rightarrow E(x, y, v, y, v, u))$
 $= \neg (\neg T(x, y, u, v) \vee E(x, y, v, y, v, u))$

$= T(x, y, u, v) \wedge \neg E(x, y, v, y, v, u)$

化成子句为 (拆成了两个子句):

(3) $T(x, y, u, v)$

(4) $\neg E(x, y, v, y, v, u)$

归结过程:

(5) $P(x, y, u, v)$ 由(1)、(3)归结

(6) $E(x, y, v, y, v, u)$ 由(5)、(2)归结

(7) \square 由(6)、(4)归结

注意: 以上谓词的变量由于都是全称变量, 因此省略了进行变量换名和代换的步骤。

9. 证明理发师悖论: 若每个理发师都为不能给自己理发的人理发, 且每个理发师都不能给自己理发的人理发, 则不存在任何理发师。

参考答案:

用 $\text{Barber}(x)$ 表示 x 是理发师, $\text{Haircut_for}(x, y)$ 表示 x 为 y 理发

$\forall x \forall y \text{ Barber}(x) \wedge \neg \text{Haircut_for}(y, y) \rightarrow \text{Haircut_for}(x, y)$

$\forall x \forall y \text{ Barber}(x) \wedge \text{Haircut_for}(y, y) \rightarrow \neg \text{Haircut_for}(x, y)$

$G: \neg \exists x \text{ Barber}(x)$

将逻辑谓词表达式转化为子句形式

(1) $\neg \text{Barber}(x) \vee \text{Haircut_for}(y, y) \vee \text{Haircut_for}(x, y)$

(2) $\neg \text{Barber}(x) \vee \neg \text{Haircut_for}(y, y) \vee \neg \text{Haircut_for}(x, y)$

将子句(2)中变量换名:

(2') $\neg \text{Barber}(z) \vee \neg \text{Haircut_for}(w, w) \vee \neg \text{Haircut_for}(z, w)$

(3) $\neg G: \text{Barber}(A)$ //用常量 A 替代存在量词限定的变量

归结过程:

$T1: \text{Barber}(A)$

$C1: \neg \text{Barber}(x) \vee \text{Haircut_for}(x, x) \vee \text{Haircut_for}(x, x)$ //用 x 替代 y

$C2: \neg \text{Barber}(z) \vee \neg \text{Haircut_for}(z, z) \vee \neg \text{Haircut_for}(z, z)$ // (2') 中用 z 替代 w

$T4: (C1, T1) \text{Haircut_for}(A, A)$

$T5: (C2, T1) \neg \text{Haircut_for}(A, A)$

$T6: (T5, T4) \square$

10. 提升引理: 若 C'_1 和 C'_2 分别是子句 C_1 和 C_2 的基例, C_1 和 C_2 不存在公共变量, 且 C'_1 和 C'_2 存在归结子句 C'_{12} , 则 C_1 和 C_2 也必存在归结子句 C_{12} , 且 C'_{12} 是 C_{12} 的基例。试证明。

参考答案: 参考教材 P236。

11. 对下面的每一对项, 跟踪合一算法的操作步骤。

(1) $(f \text{ Marcus})(f \text{ Caesar})$

(2) $(f \text{ } x)(f \text{ } (g \text{ } y))$

(3) $(f \text{ Marcus } (g \text{ } x \text{ } y)) (f \text{ } x \text{ } (g \text{ Caesar Marcus}))$

参考答案:

(1) 设 $L_1 = (f \text{ Marcus}), L_2 = (f \text{ Caesar})$

① L_1, L_2 都不为原子。

② $\text{Length}(L_1) = \text{Length}(L_2)$ 。

③ 设 SubST 为 NIL 。

(3) $T(x, y, u, v)$

(4) $\neg E(x, y, v, y, v, u)$

归结过程:

(5) $P(x, y, u, v)$ 由(1)、(3)归结

(6) $E(x, y, v, y, v, u)$ 由(5)、(2)归结

(7) \square 由(6)、(4)归结

注意: 以上谓词的变量由于都是全称变量, 因此省略了进行变量换名和代换的步骤。

9. 证明理发师悖论: 若每个理发师都为不能给自己理发的人理发, 且每个理发师都不能给自己理发的人理发, 则不存在任何理发师。

参考答案:

用 $\text{Barber}(x)$ 表示 x 是理发师, $\text{Haircut_for}(x, y)$ 表示 x 为 y 理发

$\forall x \forall y \text{ Barber}(x) \wedge \neg \text{Haircut_for}(y, y) \rightarrow \text{Haircut_for}(x, y)$

$\forall x \forall y \text{ Barber}(x) \wedge \text{Haircut_for}(y, y) \rightarrow \neg \text{Haircut_for}(x, y)$

$G: \neg \exists x \text{ Barber}(x)$

将逻辑谓词表达式转化为子句形式

(1) $\neg \text{Barber}(x) \vee \text{Haircut_for}(y, y) \vee \text{Haircut_for}(x, y)$

(2) $\neg \text{Barber}(x) \vee \neg \text{Haircut_for}(y, y) \vee \neg \text{Haircut_for}(x, y)$

将子句(2)中变量换名:

(2') $\neg \text{Barber}(z) \vee \neg \text{Haircut_for}(w, w) \vee \neg \text{Haircut_for}(z, w)$

(3) $\neg G; \text{Barber}(A)$ //用常量 A 替代存在量词限定的变量

归结过程:

$T1: \text{Barber}(A)$

$C1: \neg \text{Barber}(x) \vee \text{Haircut_for}(x, x) \vee \text{Haircut_for}(x, x)$ //用 x 替代 y

$C2: \neg \text{Barber}(z) \vee \neg \text{Haircut_for}(z, z) \vee \neg \text{Haircut_for}(z, z)$ // (2') 中用 z 替代 w

$T4: (C1, T1) \text{Haircut_for}(A, A)$

$T5: (C2, T1) \neg \text{Haircut_for}(A, A)$

$T6: (T5, T4) \square$

10. 提升引理: 若 C'_1 和 C'_2 分别是子句 C_1 和 C_2 的基例, C_1 和 C_2 不存在公共变量, 且 C'_1 和 C'_2 存在归结子句 C'_{12} , 则 C_1 和 C_2 也必存在归结子句 C_{12} , 且 C'_{12} 是 C_{12} 的基例。试证明。

参考答案: 参考教材 P236。

11. 对下面的每一对项, 跟踪合一算法的操作步骤。

(1) $(f \text{ Marcus})(f \text{ Caesar})$

(2) $(f \text{ } x)(f \text{ } (g \text{ } y))$

(3) $(f \text{ Marcus } (g \text{ } x \text{ } y)) (f \text{ } x \text{ } (g \text{ Caesar Marcus}))$

参考答案:

(1) 设 $L_1 = (f \text{ Marcus}), L_2 = (f \text{ Caesar})$

① L_1, L_2 都不为原子。

② $\text{Length}(L_1) = \text{Length}(L_2)$ 。

③ 设 SubST 为 NIL 。

④ 分别对 $(f \ f)$ 、 (Marcus Caesar) 进行合一。

⑤ f 与 f 进行合一判断, 返回 NIL (不需要代换)。

⑥ Marcus 与 Caesar 进行合一判断, 返回 F (不可合一)。

⑦ $\text{SubST} = \{F\}$ 。

不能合一。

(2) (步骤省略) 分别对 $(f \ f)$ 、 $(x \ (g \ y))$ 进行合一。

后者返回 $\text{SubST} = \{(g \ y)/x\}$, 可以合一。

(3) 先应将 $(f \ x \ (g \ \text{Caesar} \ \text{Marcus}))$ 中的变量 x 换名为 z , 再跟踪算法, 判断 $(f \ \text{Marcus} \ (g \ x \ y))$ 与 $(f \ z \ (g \ \text{Caesar} \ \text{Marcus}))$ 是否合一; 算法递归调研, 分别判断 $(f \ f)$ 、 $(\text{Marcus} \ z)$ 、 $((g \ x \ y) \ (g \ \text{Caesar} \ \text{Marcus}))$ 是否合一。算法结果返回

$\text{SubST} = \{\text{Marcus}/z, \text{Caesar}/x, \text{Marcus}/y\}$

补充:

判断两个公式是否合一, 可以通过寻找差异集来判断, 差异集: 设 S 是一个非空的具有相同谓词名的原子公式集, 从 S 中各公式的左边第一项开始, 同时向右比较, 直到发现第一个不都相同的项为止, 用这些项的差异部分组成一个集合, 这个集合就是原子公式子句集的一个差异集。例如:

$S = \{P(x, y, z), P(x, f(a), h(b))\}$, 则 S 的差异集 $D_1 = \{y, f(a)\}$, $D_2 = \{z, h(b)\}$

求子句集 S 的最一般合一 (MGU) 的算法, 即合一算法 (Unification Algorithm):

① 置 $k=0, S_k=S, \delta_k=\epsilon$ 。

② 若 S_k 是单元素集, 则算法终止, $\text{MGU}=\delta_k$ 。

③ 求 S_k 的差异集 D_k 。

④ 若 D_k 中存在元素 x_k 和 t_k , 其中 x_k 是变元, t_k 是项且 x_k 不在 t_k 中出现, 则置 $S_{k+1} = S_k \cdot \{t_k/x_k\}$, $\delta_{k+1} = \delta_k \cdot \{t_k/x_k\}$, $k=k+1$, 转步骤②。

⑤ 算法终止, S 的 MGU 不存在 (若 S 是可合一的, 算法总是在步骤②终止)。

12. 说明下列文字能合一和不能合一的理由。

(1) $\{P(f(x, x), A), P(f(y, f(y, A)), A)\}$

(2) $\{P(A), P(f(x))\}$

(3) $\{P(f(A), x), P(x, A)\}$

(4) $\{P(x, f(y, z)), P(x, f(g(a), h(b)))\}$

(5) $\{P(x, f(y, z)), P(x, a), P(x, g(h(k(x))))\}$

参考答案:

(1) 用观察法: $P(f(x, x), A)$ 与 $P(f(y, f(y, A)), A)$ 比较

$f(x, x)$ 与 $f(y, f(y, A))$ 比较, $y/x, f(y, y)$ 与 $f(y, f(y, A))$ 比较, y 与 $f(y, A)$ 不能进行代换。因此本题不能合一。

(2) $P(A)$ 与 $P(f(x))$ 比较后, A 与 $f(x)$ 比较, 函数与常量不能进行代换。

(3) $\{P(f(A), x), P(x, A)\}$ 进行合一之前, 将 $P(x, A)$ 中的变量 x 换名为 y , 再对 $P(f(A), x), P(y, A)$ 进行比较, 得到代换 $\{f(A)/y, A/x\}$, 代换后两个公式完全一样, 因此可以合一。

(4) 同(3)一样, 先将 $P(x, f(g(a), h(b)))$ 中的 x 换名为 u , 再对 $P(x, f(y, z)), P(u,$

$f(g(a), h(b))$ 进行比较, 得到代换 $\{u/x, g(a)/y, h(b)/z\}$, 代换后两个公式完全一样, 因此可以合一。

(5) 本题是考虑 $P(x, f(y, z)), P(x, a), P(x, g(h(k(x))))$ 三项之间的合一, 可以先考虑 $P(x, f(y, z)), P(x, a)$ 两项之间的合一。显然 $f(y, z)$ 与 a 之间找不到代换 (a 是常量), 因此无法合一, 所以三项之间也无法合一。

13. 某村民王某被害, 有 4 个嫌疑犯 A、B、C、D。公安局派出 5 个侦察员, 他们的侦察结果分别是: A、B 中至少有一人作案, B、C 中至少有一人作案, C、D 中至少有一人作案, A、C 中至少有一人与此案无关, B、D 中至少有一人与此案无关。所有侦察结果都是可靠的。试用归结原理求出谁是罪犯。

此题的另一种说法是: 某人被盗, 公安派出所派出 5 个侦察员去调查, 研究案情时, 侦察员 A 说: “赵与钱中至少有一人作案”; 侦察员 B 说: “钱与孙中至少有一人作案”; 侦察员 C 说: “孙与李中至少有一人作案”; 侦察员 D 说: “赵与孙中至少有一人与此案无关”; 侦察员 E 说: “钱与李中至少有一人与此案无关”。如果这 5 个侦察员的话都是可信的, 试问谁是盗窃犯?

参考答案:

定义谓词 $P(x)$: x 是作案者; 则:

A: $P(\text{Zhao}) \vee P(\text{Qian})$

B: $P(\text{Qian}) \vee P(\text{Sun})$

C: $P(\text{Sun}) \vee P(\text{Li})$

D: $\sim P(\text{Zhao}) \vee \sim P(\text{Sun})$

E: $\sim P(\text{Qian}) \vee \sim P(\text{Li})$

用回答问题的方式求解此题, 其目标子句为:

G: $P(y) \wedge \text{Answer}(y)$

其否定为: $\sim P(y) \vee \text{Answer}(y)$

归结过程:

C1: $P(\text{Zhao}) \vee P(\text{Qian})$

C2: $P(\text{Qian}) \vee P(\text{Sun})$

C3: $P(\text{Sun}) \vee P(\text{Li})$

C4: $\sim P(\text{Zhao}) \vee \sim P(\text{Sun})$

C5: $\sim P(\text{Qian}) \vee \sim P(\text{Li})$

T2: $(C1, C4) P(\text{Qian}) \vee \sim P(\text{Sun})$

T8: $(C2, T2) P(\text{Qian})$

T1($\sim G$): $\sim P(y) \vee \text{Answer}(y)$

T23: $(\sim G, T8) \text{Answer}(\text{Qian})$

另一个推导:

C5: $\sim P(\text{Qian}) \vee \sim P(\text{Li})$

C2: $P(\text{Qian}) \vee P(\text{Sun})$

T6: $(C2, C5) P(\text{Sun}) \vee \sim P(\text{Li})$

C3: $P(\text{Sun}) \vee P(\text{Li})$
 T13: $(C3, T6) P(\text{Sun})$
 T1($\sim G$): $\sim P(y) \vee \text{Answer}(y)$
 T25: $(T1, T13) \text{Answer}(\text{Sun})$
 所以, 钱和孙是盗窃犯。

14. 已知某些病人喜欢所有的医生, 没有一个病人喜欢任意一个骗子。证明任意一个医生都不是骗子。

$P(x)$: x 是病人

$D(x)$: x 是医生

$Q(x)$: x 是骗子

$\text{Like}(x, y)$: x 喜欢 y

(1) 试将已知命题和要证命题用谓词逻辑公式表达出来。

(2) 将谓词逻辑公式转化成子句形式。

(3) 给出子句形式的证明步骤。

参考答案:

(1) 试将已知命题和要证命题用谓词逻辑公式表达出来。

① $\exists x(P(x) \wedge (\forall y D(y) \rightarrow \text{Like}(x, y)))$

② $\neg(\exists x \exists y(P(x) \wedge Q(y) \wedge \text{Like}(x, y)))$

也即 $\forall x \forall y(P(x) \wedge Q(y) \rightarrow \neg \text{Like}(x, y))$

要证的目标: $\forall x(D(x) \rightarrow \neg Q(x))$

(2) 将谓词逻辑公式转化成子句形式。

将公式①转换成子句:

$\exists x(P(x) \wedge (\forall y D(y) \rightarrow \text{Like}(x, y)))$

改为:

$\exists x \forall y(P(x) \wedge (D(y) \rightarrow \text{Like}(x, y)))$ //量词前移

$\forall y(P(A) \wedge (D(y) \rightarrow \text{Like}(A, y)))$ //去掉存在量词

$P(A) \wedge (D(y) \rightarrow \text{Like}(A, y))$ //去掉全称量词

$P(A) \wedge (\neg D(y) \vee \text{Like}(A, y))$ //改蕴含式为子句形式

(1) $P(A)$

(2) $\neg D(y) \vee \text{Like}(A, y)$

将公式②转换成子句:

$\forall x \forall y(P(x) \wedge Q(y) \rightarrow \neg \text{Like}(x, y))$ //去掉 \wedge 连接词, 改为两个子句

$\forall x \forall y(\neg P(x) \vee \neg Q(y) \vee \neg \text{Like}(x, y))$ //改蕴含式为子句形式

(3) $\neg P(x) \vee \neg Q(y) \vee \neg \text{Like}(x, y)$ //去掉全称量词

对目标进行否定:

$\neg \forall x(D(x) \rightarrow \neg Q(x))$

$\exists x \neg(D(x) \rightarrow \neg Q(x))$

$\exists x \neg(\neg D(x) \vee \neg Q(x))$

$\exists x(D(x) \wedge Q(x))$

//改蕴含式为子句形式

//德摩根定理

(4)
(5)
(3)
归结
(6)
(7)
(8)
(9)

1. 什
答:

过程如下

(1)
(2)
(3)
(4)

S中。

如此

2. 影

F_1 :

F_2 :

F_3 :

求证

参考

首先

$N(x)$

$I(x)$

$E(x)$

$O(x)$

$GZ(x)$

$s(x)$

将上

F_1 :

F_2 :

F_3 :

G.

$D(B) \wedge Q(B)$

(4) $D(B)$

(5) $Q(B)$

(3) 给出子句形式的证明步骤。

归结证明过程:

(6) $\neg P(A) \vee \neg Q(y) \vee \neg D(y)$ 由(2)(3)

(7) $\neg P(A) \vee \neg Q(B)$ 由(4)(6)

(8) $\neg P(A)$ 由(5)(7)

(9) \square 由(8)(1)

//去掉存在量词

11.4 补充习题

1. 什么是归结推理? 它推理的基本方法是什么?

答: 归结推理就是利用归结和反演实现定理的证明。它的基本方法就是反证法, 具体程如下。

(1) 将定理证明的前提谓词公式转化为子句集 F 。

(2) 将求证的目标表示成合适的谓词公式 G (目标公式)。

(3) 将目标公式的否定式 $\neg G$ 转化成子句的形式, 并加入到子句集 F 中, 得到子句集 S 。

(4) 应用归结原理对子句集 S 中的子句进行归结, 并把每次归结得到的归结式都并入中。

如此反复进行, 若归结得到一个空子句 NIL , 则停止归结, 证明了 G 为真。

2. 设有下列知识。

F_1 : 自然数都是大于等于零的整数。

F_2 : 所有整数不是偶数就是奇数。

F_3 : 偶数除以 2 是整数。

求证: 所有自然数不是奇数就是其一半为整数的数。

参考答案:

首先定义如下谓词。

$N(x)$: x 是自然数。

$I(x)$: x 是整数。

$E(x)$: x 是偶数。

$O(x)$: x 是奇数。

$GZ(x)$: x 大于零。

$s(x)$: x 除以 2。

将上述各语句翻译成谓词公式为:

$F_1: \forall x(N(x) \rightarrow GZ(x) \vee I(x))$

$F_2: \forall x(I(x) \rightarrow (E(x) \vee O(x)))$

$F_3: \forall x(E(x) \rightarrow I(s(x)))$

$G: \forall x(N(x) \rightarrow (I(s(x)) \vee O(x)))$

将上述各谓词公式翻译成子句为：

$F_1 \wedge F_2 \wedge F_3 \wedge \neg G$ 的子句集为

- (1) $\neg N(x) \vee GZ(x)$
- (2) $\neg N(y) \vee I(y)$
- (3) $\neg I(z) \vee E(z) \vee O(z)$
- (4) $\neg E(u) \vee I(s(u))$
- (5) $N(a)$
- (6) $\neg O(a)$
- (7) $\neg I(s(a))$

子句(5)、(6)、(7)的意思很明显,有一个自然数 a 不是奇数,它的一半也不为整数的数。

归结得:

- | | |
|----------------------|------------------------|
| (8) $I(a)$ | $[(2), (5), \{a/y\}]$ |
| (9) $E(a) \vee O(a)$ | $[(8), (3), \{a/z\}]$ |
| (10) $E(a)$ | $[(9), (6)]$ |
| (11) $I(s(a))$ | $[(10), (4), \{a/u\}]$ |
| (12) \square | $[(11), (7)]$ |

即证明了所有自然数不是奇数就是其一半为整数的数。

3. 已知子句集 $S = \{P(x) \vee Q(y), \neg P(A), \neg Q(B)\}$ 。

请为其构造语义树并判断该子句集是否存在一棵封闭的语义树。若存在,请指出每一个失效节点分别使得哪一个子句的基例示为假;若不存在,请说明原因。

参考答案:

失效点: 若 $I(N)$ 使 S 中某个子句的基例示为假,但 $I(N')$ 不使 S 中任何子句的基例示为假,其中 N' 是 N 的任何先辈节点,则节点 N 称为失效点。

封闭语义树: 若语义树 Tr 的每一分支都终止在失效点上,则称其为封闭语义树。子句集 S 语义树如图 11.1 所示。

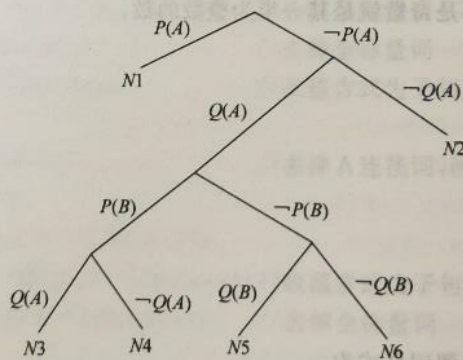


图 11.1 子句集 S 对应的语义树

4. 跟踪合一算法,判断下列公式之间是否合一。

- (1) $\{P(a, f(a)), P(x, f(b))\}$

参考答案:

首先明确概念:

① a, b 为常量, x, y 为变量, P, Q 为谓词。

② f, g, h 为函数。

③ 在两个谓词中名称一样的变量并不是同一个变量, 因此要先将变量名称修改为不同的名称。

设 $L1 = (P a(f a))$, $L2 = (P(x, (f, b)))$, 跟踪合一算法的步骤如下。

① $L1, L2$ 都不为原子。

② 置 SUBST 为 NIL。

③ 分别对 $(P, P), (a, x)(f, f)(a, b)$ 进行合一。

④ P 与 P 合一返回 Nil, a 与 x 合一返回 a/x , f 与 f 合一返回 Nil, a 与 b 合一返回 F。

⑤ SUBST = {F}

得出不能合一。

(2) $\{Q(f(y)), Q(f(g(a, b)))\}$

参考答案:

设 $L1 = (Q(f(y)))$, $L2 = (Q(f(g(a, b))))$, 跟踪合一算法的步骤如下。

① $L1, L2$ 都不为原子。

② 置 SUBST 为 NIL。

③ 分别对 $(Q, Q), (f, f)(y, (g(a, b)))$ 进行合一。

④ Q 与 Q 合一返回 Nil, f 与 f 合一返回 Nil, y 与 $(g(a, b))$ 合一返回 $y/g(a, b)$ 。

SUBST = $\{y/g(a, b)\}$

得出能合一。

(3) $\{f(\text{Marcus}, g(x, y)), f(x, g(\text{Caesar}, \text{Marcus}))\}$

参考答案:

设 $L1 = (f(\text{Marcus}, g(x, y)))$, $L2 = (f(z, g(\text{Caesar}, \text{Marcus})))$, 跟踪合一算法的步骤如下。

① $L1, L2$ 都不为原子。

② 置 SUBST 为 NIL。

③ 分别对 $(f, f), (g, g)$ 进行合一。

④ f 与 f 合一返回 Nil, f 与 f 合一返回 Nil, x 与 Caesar 合一返回 Caesar / x , y 与 Marcus 合一返回 Marcus / y , Marcus 与 z 合一返回 Marcus / z 。

得出能合一。

5. 设有如下语句, 请用相应的谓词公式分别表示。

(1) 有的人喜欢梅花, 有的人喜欢菊花, 有的人既喜欢梅花又喜欢菊花。

参考答案:

定义谓词:

$P(x)$: x 是人

$L(x, y)$: x 喜欢 y

其中, y 的个体域是 {梅花, 菊花}。

将知识用谓词表示为:

$$(\exists x)(P(x) \rightarrow L(x, \text{梅花}) \vee L(x, \text{菊花}) \vee L(x, \text{梅花}) \wedge L(x, \text{菊花}))$$

(2) 有人每天下午都去打篮球。

参考答案:

定义谓词:

$P(x)$: x 是人

$B(x)$: x 打篮球

$A(y)$: 时间 y 是下午

将知识用谓词表示为:

$$(\exists x)(\forall y)(P(x) \wedge A(y) \rightarrow B(x))$$

(3) 新型计算机速度又快, 存储容量又大。

参考答案:

定义谓词:

$NC(x)$: x 是新型计算机

$F(x)$: x 速度快

$B(x)$: x 容量大

将知识用谓词表示为:

$$(\forall x)(NC(x) \rightarrow F(x) \wedge B(x))$$

(4) 不是每个计算机系的学生都喜欢在计算机上编程序。

参考答案:

定义谓词:

$S(x)$: x 是计算机系学生

$L(x, \text{programming})$: x 喜欢编程序

$U(x, \text{computer})$: x 使用计算机

将知识用谓词表示为:

$$\neg (\forall x)(S(x) \rightarrow L(x, \text{programming}) \wedge U(x, \text{computer}))$$

(5) 凡是喜欢编程序的人都喜欢计算机。

参考答案:

定义谓词:

$P(x)$: x 是人

$L(x, y)$: x 喜欢 y

将知识用谓词表示为:

$$(\forall x)(P(x) \wedge L(x, \text{programming}) \rightarrow L(x, \text{computer}))$$

6. 用谓词表示法求解机器人摆积木问题。设机器人有一只机械手, 要处理的事件有一张桌子, 桌面上可堆放若干相同的方积木块。机械手有 4 个操作积木的典型动作: 从桌面上拣起一块积木; 将手中的积木放到桌面上; 在积木上再摆上一块积木; 从积木上面拣起一块积木。积木事件的布局如图 11.2 所示。

参考答案:

(1) 先定义描述状态的谓词。

CLEAR(x): 积木 x 上面是空的。
 ON(x, y): 积木 x 在积木 y 的上面。
 ONTABLE(x): 积木 x 在桌子上。
 HOLDING(x): 机械手抓住 x 。
 HANDEEMPTY(): 机械手是空的。
 其中, x 和 y 的个体域都是 $\{A, B, C\}$ 。

问题的初始状态是:

ONTABLE(A)

ONTABLE(B)

ON(C, A)

CLEAR(B)

CLEAR(C)

HANDEEMPTY()

问题的目标状态是:

ONTABLE(C)

ON(B, C)

ON(A, B)

CLEAR(A)

HANDEEMPTY()

(2) 再定义描述操作的谓词。

在本问题中, 机械手的操作需要定义以下 4 个谓词。

Pickup(x): 从桌面上拣起一块积木 x 。

Putdown(x): 将手中的积木放到桌面上。

Stack(x, y): 在积木 x 上面再摞上一块积木 y 。

Upstack(x, y): 从积木 x 上面拣起一块积木 y 。

其中, 每一个操作都可分为条件和动作两部分, 具体描述如下。

Pickup(x)

条件: ONTABLE(x), HANDEEMPTY, CLEAR(x)

动作: 删除表: ONTABLE(x), HANDEEMPTY()

添加表: HANDEEMPTY()

Putdown(x)

条件: HANDEEMPTY()

动作: 删除表: HANDEEMPTY()

添加表: ONTABLE(x), CLEAR(x), HANDEEMPTY()

Stack(x, y)

条件: HANDEEMPTY(), CLEAR(y)

动作: 删除表: HANDEEMPTY(x), CLEAR(y)

添加表: HANDEEMPTY, ON(x, y), CLEAR(x)

Upstack(x, y)

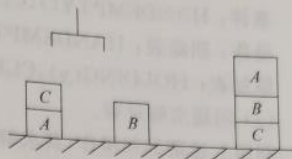


图 11.2 机器人摞积木问题

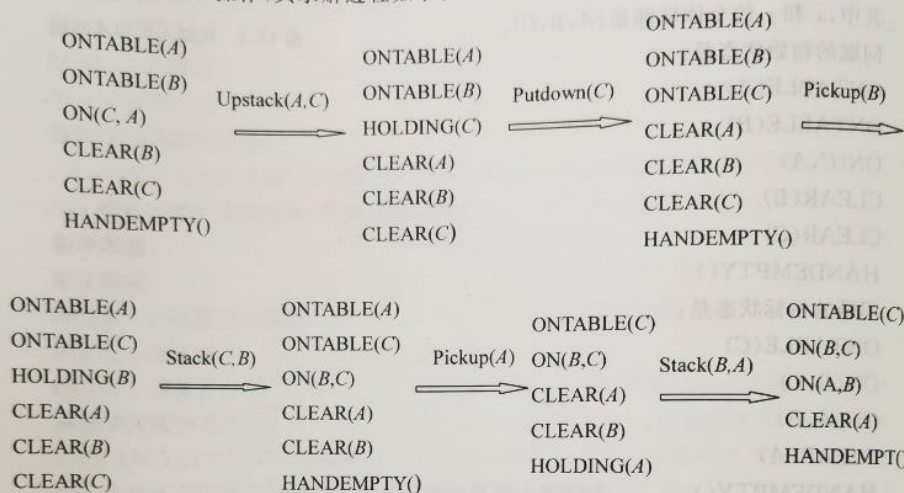
条件: $\text{HANDEEMPTY}(), \text{CLEAR}(y), \text{ON}(y, x)$

动作: 删除表: $\text{HANDEEMPTY}(), \text{ON}(y, x)$

添加表: $\text{HOLDING}(y), \text{CLEAR}(x)$

(3) 问题求解过程。

利用上述谓词和操作, 其求解过程如下。



7. 用谓词表示法求解农夫、狼、山羊、白菜过河问题。农夫、狼、山羊、白菜全部放在一条河的左岸, 现在要把他们全部送到河的右岸去。农夫有一条船, 过河时, 除农夫外船上至多能载狼、山羊、白菜中的一种。如果农夫不在时, 狼要吃山羊, 山羊要吃白菜。试规划出一个确保全部安全过河的计划。请写出所用谓词的定義, 并给出每个谓词的功能及变量的个体域。

参考答案:

(1) 先定义描述状态的谓词。

要描述这个问题, 需要能够说明农夫、狼、山羊、白菜和船在什么位置。为简化问题表示, 只描述左岸和右岸的状态。并且, 由于左岸和右岸的状态互补, 因此可进一步简化为仅对左岸或右岸的状态做直接描述。本题选择对左岸进行直接描述的方法, 即定义谓词如下。

$\text{At_LB}(x)$: x 在左岸 (At left bank)

其中, x 的个体域是 {农夫, 船, 狼, 山羊, 白菜}。对应地, $\neg \text{At_LB}(x)$ 表示 x 在右岸。

问题的初始状态为:

$\text{At_LB}(\text{农夫})$

$\text{At_LB}(\text{船})$

$\text{At_LB}(\text{狼})$

$\text{At_LB}(\text{山羊})$

$\text{At_LB}(\text{白菜})$

问题的目标状态:

$\neg \text{At_LB}(\text{农夫})$

$\neg \text{At_LB}(\text{船})$

$\neg \text{At_LB}(\text{狼})$

$\neg \text{At_LB}(\text{山羊})$

$\neg \text{At_LB}(\text{白菜})$

(2) 再定义描述操作的谓词。

本题需要以下 4 个描述操作的谓词。

L-to-R(): 农夫自己划船从左岸到右岸。

L-to-R(x): 农夫带着 x 划船从左岸到右岸。

R-to-L(): 农夫自己划船从右岸到左岸。

R-to-L(x): 农夫带着 x 划船从右岸到左岸。

其中, x 的个体域是{狼, 山羊, 白菜}。

对上述每个操作, 都包括条件和动作两部分。它们对应的条件和动作如下。

L-to-R(): 农夫划船从左岸到右岸

条件: $\text{At_LB}(\text{船}), \text{At_LB}(\text{农夫}), \neg \text{At_LB}(\text{狼}) \vee \neg \text{At_LB}(\text{山羊}), \neg \text{At_LB}(\text{山羊}) \vee \neg \text{At_LB}(\text{白菜})$

动作: 删除表: $\text{At_LB}(\text{船}), \text{At_LB}(\text{农夫})$

添加表: $\neg \text{At_LB}(\text{船}), \neg \text{At_LB}(\text{农夫})$

L-to-R(狼): 农夫带着狼划船从左岸到右岸

条件: $\text{At_LB}(\text{船}), \text{At_LB}(\text{农夫}), \text{At_LB}(\text{狼}), \neg \text{At_LB}(\text{山羊})$

动作: 删除表: $\text{At_LB}(\text{船}), \text{At_LB}(\text{农夫}), \text{At_LB}(\text{狼})$

添加表: $\neg \text{At_LB}(\text{船}), \neg \text{At_LB}(\text{农夫}), \neg \text{At_LB}(\text{狼})$

L-to-R(山羊): 农夫带着山羊划船从左岸到右岸

条件: $\text{At_LB}(\text{船}), \text{At_LB}(\text{农夫}), \text{At_LB}(\text{山羊}), \text{At_LB}(\text{狼}), \text{At_LB}(\text{白菜})$

或: $\text{At_LB}(\text{船}), \text{At_LB}(\text{农夫}), \text{At_LB}(\text{山羊}), \neg \text{At_LB}(\text{狼}), \neg \text{At_LB}(\text{白菜})$

动作: 删除表: $\text{At_LB}(\text{船}), \text{At_LB}(\text{农夫}), \text{At_LB}(\text{山羊})$

添加表: $\neg \text{At_LB}(\text{船}), \neg \text{At_LB}(\text{农夫}), \neg \text{At_LB}(\text{山羊})$

L-to-R(白菜): 农夫带着白菜划船从左岸到右岸

条件: $\text{At_LB}(\text{船}), \text{At_LB}(\text{农夫}), \text{At_LB}(\text{白菜}), \neg \text{At_LB}(\text{狼})$

动作: 删除表: $\text{At_LB}(\text{船}), \text{At_LB}(\text{农夫}), \text{At_LB}(\text{白菜})$

添加表: $\neg \text{At_LB}(\text{船}), \neg \text{At_LB}(\text{农夫}), \neg \text{At_LB}(\text{白菜})$

R-to-L: 农夫划船从右岸到左岸

条件: $\neg \text{At_LB}(\text{船}), \neg \text{At_LB}(\text{农夫}), \text{At_LB}(\text{狼}) \vee \text{At_LB}(\text{山羊}), \text{At_LB}(\text{山羊}) \vee$

$\text{At_LB}(\text{白菜})$

或: $\neg \text{At_LB}(\text{船}), \neg \text{At_LB}(\text{农夫}), \neg \text{At_LB}(\text{狼}), \neg \text{At_LB}(\text{白菜}), \text{At_LB}(\text{山羊})$

动作: 删除表: $\neg \text{At_LB}(\text{船}), \neg \text{At_LB}(\text{农夫})$

添加表: $\text{At_LB}(\text{船}), \text{At_LB}(\text{农夫})$

R-to-L(山羊): 农夫带着山羊划船从右岸到左岸

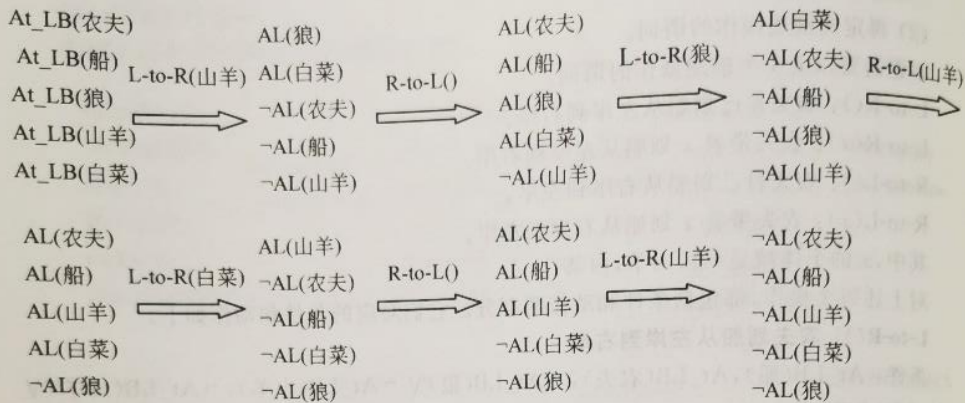
条件: $\neg \text{At_LB}(\text{船}), \neg \text{At_LB}(\text{农夫}), \neg \text{At_LB}(\text{山羊}), \neg \text{At_LB}(\text{狼}), \neg \text{At_LB}(\text{山}$

羊), $\text{At_LB}(\text{白菜})$

动作: 删除表: $\neg At_LB(\text{船}), \neg At_LB(\text{农夫}), \neg At_LB(\text{山羊})$

添加表: $At_LB(\text{船}), At_LB(\text{农夫}), At_LB(\text{山羊})$

(3) 问题求解过程。



8. 用谓词表示法求解传教士和野人问题。在河的左岸有 3 个传教士、3 个野人和一条船,传教士想用这条船将所有的人都运过河去,但要受到以下条件限制。

(1) 传教士和野人都会划船,但船一次只能装运两个人。

(2) 在任何一岸,野人数不能超过传教士,否则传教士会被野人吃掉。

假定野人愿意服从任何一种过河安排,请规划出一种确保传教士安全的过河方案。要求写出所用谓词的定義、功能及变量的个体域。

参考答案:

(1) 定义谓词。

先定义传教士和野人人数的谓词。

$G(x, y, S)$: 在状态 S 下 x 大于 y

$GE(x, y, S)$: 在状态 S 下 x 大于或等于 y

其中, x, y 分别代表传教士人数和野人数,他们的个体域均为 $\{0, 1, 2, 3\}$ 。

再定义船所在岸的谓词和传教士不在该岸上的谓词。

$Boat(z, S)$: 状态 S 下船在 z 岸

$EZ(x, S)$: 状态 S 下 x 等于 0, 即传教士不在该岸上

其中, z 的个体域是 $\{L, R\}$, L 表示左岸, R 表示右岸。

再定义安全性谓词。

$Safety(z, x, y, S) \equiv (G(x, 0, S) \wedge GE(x, y, S)) \vee (EZ(x, S))$

其中, z, x, y 的含义同上。该谓词的含义是: 状态 S 下, 在 z 岸, 保证传教士安全, 当且仅当传教士不在该岸上, 或者传教士在该岸上, 但人数超过野人数。该谓词同时也描述了相应的状态。

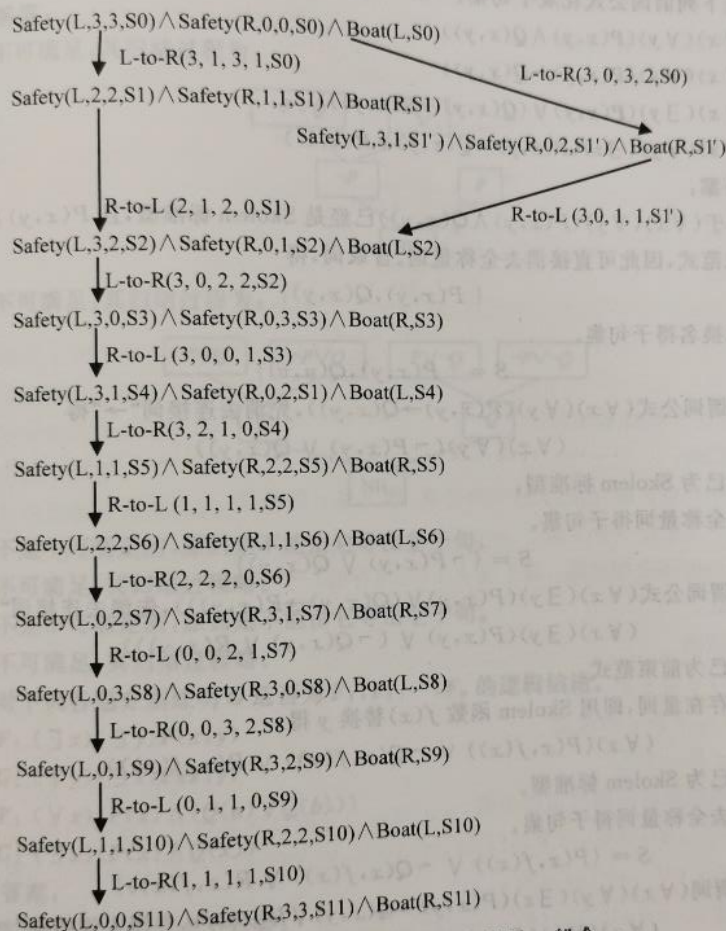
再定义描述过河方案的谓词。

$L\text{-to-}R(x, x1, y, y1, S)$: $x1$ 个传教士和 $y1$ 个野人渡船从河的左岸到河的右岸

条件: $Safety(L, x-x1, y-y1, S') \wedge Safety(R, 3-x+x1, 3-y+y1, S') \wedge Boat(L, S)$

动作: $\text{Safety}(L, x - x1, y - y1, S') \wedge \text{Safety}(R, 3 - x + x1, 3 - y + y1, S') \wedge \text{Boat}(R, S')$
 $R\text{-to-}L(x, x1, y, y1, S)$: $x2$ 个传教士和 $y2$ 个野人渡船从河的左岸到河的右岸
 条件: $\text{Safety}(R, 3 - x - x2, 3 - y - y2, S') \wedge \text{Safety}(L, x + x2, y + y2, S') \wedge \text{Boat}(R, S)$
 动作: $\text{Safety}(R, 3 - x - x2, 3 - y - y2, S') \wedge \text{Safety}(L, x + x2, y + y2, S') \wedge \text{Boat}(L, S')$

(2) 过河方案。



9. 判断下列公式是否为可合一, 若可合一, 则求出其最一般合一。

- (1) $P(a, b), P(x, y)$
- (2) $P(f(x), b), P(y, z)$
- (3) $P(f(x), y), P(y, f(b))$
- (4) $P(f(y), y, x), P(x, f(a), f(b))$
- (5) $P(x, y), P(y, x)$

参考答案:

- (1) 可合一, 其最一般合一为: $\sigma = \{a/x, b/y\}$ 。
 (2) 可合一, 其最一般合一为: $\sigma = \{y/f(x), b/z\}$ 。
 (3) 可合一, 其最一般合一为: $\sigma = \{f(b)/y, b/x\}$ 。
 (4) 不可合一。
 (5) 可合一, 其最一般合一为: $\sigma = \{y/x\}$ 。

10. 把下列谓词公式化成子句集。

- (1) $(\forall x)(\forall y)(P(x, y) \wedge Q(x, y))$
 (2) $(\forall x)(\forall y)(P(x, y) \rightarrow Q(x, y))$
 (3) $(\forall x)(\exists y)(P(x, y) \vee (Q(x, y) \rightarrow R(x, y)))$
 (4) $(\forall x)(\forall y)(\exists z)(P(x, y) \rightarrow Q(x, y) \vee R(x, z))$

参考答案:

(1) 由于 $(\forall x)(\forall y)(P(x, y) \wedge Q(x, y))$ 已经是 Skolem 标准型, 且 $P(x, y) \wedge Q(x, y)$ 已经是合取范式, 因此可直接消去全称量词、合取词, 得

$$\{P(x, y), Q(x, y)\}$$

再进行变元换名得子句集。

$$S = \{P(x, y), Q(u, v)\}$$

(2) 对谓词公式 $(\forall x)(\forall y)(P(x, y) \rightarrow Q(x, y))$, 先消去连接词“ \rightarrow ”得

$$(\forall x)(\forall y)(\neg P(x, y) \vee Q(x, y))$$

此公式已为 Skolem 标准型。

再消去全称量词得子句集。

$$S = \{\neg P(x, y) \vee Q(x, y)\}$$

(3) 对谓词公式 $(\forall x)(\exists y)(P(x, y) \vee (Q(x, y) \rightarrow R(x, y)))$, 先消去连接词“ \rightarrow ”得

$$(\forall x)(\exists y)(P(x, y) \vee (\neg Q(x, y) \vee R(x, y)))$$

此公式已为前束范式。

再消去存在量词, 即用 Skolem 函数 $f(x)$ 替换 y 得

$$(\forall x)(P(x, f(x)) \vee \neg Q(x, f(x)) \vee R(x, f(x)))$$

此公式已为 Skolem 标准型。

最后消去全称量词得子句集。

$$S = \{P(x, f(x)) \vee \neg Q(x, f(x)) \vee R(x, f(x))\}$$

(4) 对谓词 $(\forall x)(\forall y)(\exists z)(P(x, y) \rightarrow Q(x, y) \vee R(x, z))$, 先消去连接词“ \rightarrow ”得

$$(\forall x)(\forall y)(\exists z)(\neg P(x, y) \vee Q(x, y) \vee R(x, z))$$

再消去存在量词, 即用 Skolem 函数 $f(x)$ 替换 y 得

$$(\forall x)(\forall y)(\neg P(x, y) \vee Q(x, y) \vee R(x, f(x, y)))$$

此公式已为 Skolem 标准型。

最后消去全称量词得子句集。

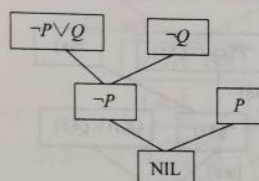
$$S = \{\neg P(x, y) \vee Q(x, y) \vee R(x, f(x, y))\}$$

11. 判断下列子句集中哪些是不可满足的。

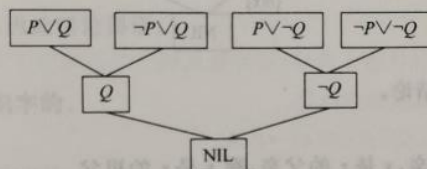
- (1) $\{\neg P \vee Q, \neg Q, P, \neg P\}$
- (2) $\{P \vee Q, \neg P \vee Q, P \vee \neg Q, \neg P \vee \neg Q\}$
- (3) $\{P(y) \vee Q(y), \neg P(f(x)) \vee R(a)\}$
- (4) $\{\neg P(x) \vee Q(x), \neg P(y) \vee R(y), P(a), S(a), \neg S(z) \vee \neg R(z)\}$
- (5) $\{\neg P(x) \vee Q(f(x), a), \neg P(h(y)) \vee Q(f(h(y)), a) \vee \neg P(z)\}$
- (6) $\{P(x) \vee Q(x) \vee R(x), \neg P(y) \vee R(y), \neg Q(a), \neg R(b)\}$

参考答案:

(1) 不可满足, 其归结过程为:



(2) 不可满足, 其归结过程为:



(3) 不是不可满足的, 原因是不能由它导出空子句。

(4) 不可满足, 其归结过程略。

(5) 不是不可满足的, 原因是不能由它导出空子句。

(6) 不可满足, 其归结过程略。

12. 对下列各题分别证明 G 是否为 F_1, F_2, \dots, F_n 的逻辑结论。

(1) $F_1: (\exists x)(\exists y)P(x, y)$

$G: (\forall y)(\exists x)P(x, y)$

(2) $F_1: (\forall x)(P(x) \wedge (Q(a) \vee Q(b)))$

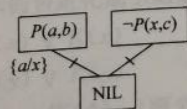
$G: (\exists x)(P(x) \wedge Q(x))$

参考答案:

(1) 先将 F 和 $\neg G$ 化成子句集:

$S = \{P(a, b), \neg P(x, c)\}$

再对 S 进行归结:



b 与 c 不一定存在合一, 所以, G 不是 F 的逻辑结论。

(2) 先将 F 和 $\neg G$ 化成子句集

由 F 得: $S_1 = \{P(x), (Q(a) \vee Q(b))\}$

由于 $\neg G$ 为: $\neg(\exists x)(P(x) \wedge Q(x))$, 即

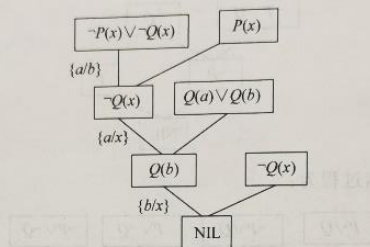
$(\forall x)(\neg P(x) \vee \neg Q(x))$

可得: $S_2 = \{\neg P(x) \vee \neg Q(x)\}$

因此, 扩充的子句集为

$S = \{P(x), (Q(a) \vee Q(b)), \neg P(x) \vee \neg Q(x)\}$

再对 S 进行归结:



所以, G 是 F 的逻辑结论。

13. 设已知:

(1) 如果 x 是 y 的父亲, y 是 z 的父亲, 则 x 是 z 的祖父。

(2) 每个人都有一个人。

使用归结演绎推理证明: 对于某人 u , 一定存在一个人 v , v 是 u 的祖父。

参考答案:

先定义谓词

$F(x, y)$: x 是 y 的父亲

$GF(x, z)$: x 是 z 的祖父

$P(x)$: x 是一个人

再用谓词把问题的断言描述出来。

已知 $F_1: (\forall x)(\forall y)(\forall z)(F(x, y) \wedge F(y, z)) \rightarrow GF(x, z)$

$F_2: (\forall y)(P(y) \rightarrow F(x, y))$

求证结论 $G: (\exists u)(\exists v)(P(u) \rightarrow GF(v, u))$

然后再将 F_1, F_2 和 $\neg G$ 化成子句集。

① $\neg F(x, y) \vee \neg F(y, z) \vee GF(x, z)$

② $\neg P(r) \vee F(s, r)$

③ $P(u)$

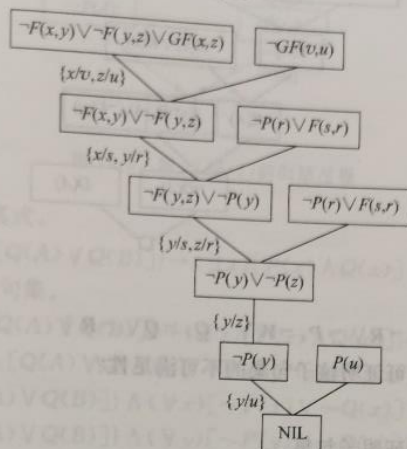
④ $\neg GF(v, u)$

注意: 后面两个子句是由

$\neg G = (\forall u)(\forall v) \neg (P(u) \rightarrow GF(v, u)) = (\forall u)(\forall v)(P(u) \wedge \neg GF(v, u))$

拆开成两个子句。

对上述扩充的子句集,其归结推理过程如下。



由于导出了空子句,因此结论得证。

14. 设已知:

(1) 能阅读的人是识字的。

(2) 海豚不识字。

(3) 有些海豚是很聪明的。

请用归结演绎推理证明: 有些很聪明的人并不能阅读。

参考答案:

第一步,先定义谓词。

设: $R(x)$ 表示 x 是能阅读的

$K(y)$ 表示 y 是识字的

$D(x)$ 表示 x 是海豚

$W(z)$ 表示 z 是很聪明的

第二步,将已知事实和目标用谓词公式表示出来。

能阅读的人是识字的: $(\forall x)(R(x) \rightarrow K(x))$

海豚不识字: $(\forall y)(D(y) \rightarrow \neg K(y))$

有些海豚是很聪明的: $(\exists z)(D(z) \wedge W(z))$

有些很聪明的人并不能阅读: $(\exists x)(W(x) \wedge \neg R(x))$

第三步,将上述已知事实和目标否定化成子句集。

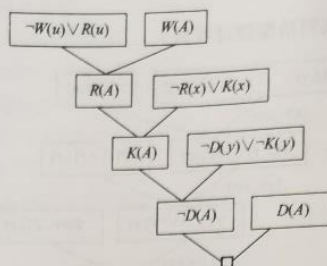
$$\neg R(x) \vee K(x)$$

$$\neg D(y) \vee K(y)$$

$$D(A), W(A) \quad (\text{两个子句})$$

$$\neg W(u) \vee R(u)$$

第四步,用归结演绎推理进行证明:



15. 对于子集:

$$\{P \vee Q, Q \vee R, R \vee W, \neg R \vee \neg P, \neg W \vee \neg Q, \neg Q \vee \neg R\}$$

用线性输入策略是否可证明该子集的不可满足性?

参考答案:

用线性输入策略不能证明子集

$$\{P \vee Q, Q \vee R, R \vee W, \neg R \vee \neg P, \neg W \vee \neg Q, \neg Q \vee \neg R\}$$

的不可满足性。原因是按线性输入策略,不存在从该子集到空子句的归结过程。

16. 用归结证明下列公式的永真性。

$$(1) (\exists x) \{ [P(x) \rightarrow P(A)] \wedge [P(x) \rightarrow P(B)] \}$$

$$(2) (\forall x) \{ [P(x) \wedge [Q(A) \vee Q(B)]] \rightarrow (\exists x) [P(x) \wedge Q(x)] \}$$

参考答案:

$$(1) (\exists x) \{ [P(x) \rightarrow P(A)] \wedge [P(x) \rightarrow P(B)] \}$$

将目标取反,化子句集:

$$\sim (\exists x) \{ [P(x) \rightarrow P(A)] \wedge [P(x) \rightarrow P(B)] \}$$

$$\sim (\exists x) \{ [\sim P(x) \vee P(A)] \wedge [\sim P(x) \vee P(B)] \}$$

$$(\forall x) \{ [P(x) \wedge \sim P(A)] \vee [P(x) \wedge \sim P(B)] \}$$

$$(\forall x) \{ [P(x) \wedge \sim P(A)] \vee P(x) \} \wedge \{ [P(x) \wedge \sim P(A)] \vee \sim P(B) \}$$

$$(\forall x) \{ P(x) \wedge [\sim P(A) \vee P(x)] \wedge [P(x) \vee \sim P(B)] \wedge [\sim P(A) \vee \sim P(B)] \}$$

$$P(x) \wedge [\sim P(A) \vee P(x)] \wedge [P(x) \vee \sim P(B)] \wedge [\sim P(A) \vee \sim P(B)]$$

将该公式的 \wedge 连接部分拆开,并进行变量换名,得子句集。

$$① P(x_1)$$

$$② \sim P(A) \vee P(x_2)$$

$$③ P(x_3) \vee \sim P(B)$$

$$④ \sim P(A) \vee \sim P(B)$$

再进行如图 11.3 所示的归结。

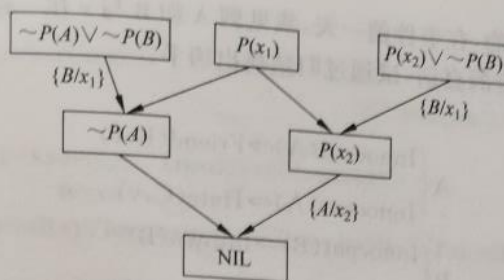


图 11.3 第 17 题(1)的归结过程

所以,公式(1)是永真式。

(2) $(\forall x)(P(x) \wedge [Q(A) \vee Q(B)]) \rightarrow (\exists x)[P(x) \wedge Q(x)]$

将目标取反,化成子句集。

$\sim\{(\forall x)(P(x) \wedge [Q(A) \vee Q(B)]) \rightarrow (\exists x)[P(x) \wedge Q(x)]\}$ //去掉蕴含符号

$\sim\{\sim\{(\forall x)P(x) \wedge [Q(A) \vee Q(B)]\} \vee (\exists x)[P(x) \wedge Q(x)]\}$

$\{(\forall x)P(x) \wedge [Q(A) \vee Q(B)]\} \wedge (\exists x)[\sim P(x) \vee \sim Q(x)]$

$\{(\forall x)P(x) \wedge [Q(A) \vee Q(B)]\} \wedge (\forall y)[\sim P(y) \vee \sim Q(y)]$ (变量换名)

$(\forall x)(\forall y)\{P(x) \wedge [Q(A) \vee Q(B)] \wedge [\sim P(y) \vee \sim Q(y)]\}$ //去掉全称量词

$P(x) \wedge [Q(A) \vee Q(B)] \wedge [\sim P(y) \vee \sim Q(y)]$

得子句集。

① $P(x)$

② $Q(A) \vee Q(B)$

③ $\sim P(y) \vee \sim Q(y)$

再进行如图 11.4 所示的归结。

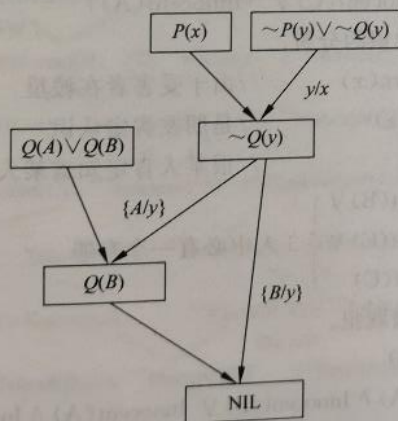


图 11.4 第 17 题(2)的归结过程

17. 已知: Victor 在城里,被杀了。取证时,3 个人的回答是:

Anlter: 我没杀人,B 是 v 的朋友,C 恨 v。

Bertain: 事情发生那天,我已出城,且不认识 v。

Cirlition: 我是无辜的, 在事件前一天, 我见到 A 和 B 与 v 在一起。
假设除凶手外都是讲的真话, 试通过归结找出凶手。

参考答案:

$$\begin{aligned} A & \begin{cases} \text{Innocent}(A) \Rightarrow \text{Friend}(B, v) \\ \text{Innocent}(A) \Rightarrow \text{Hate}(C, v) \end{cases} \\ B & \begin{cases} \text{Innocent}(B) \Rightarrow \text{Intown}(B) \\ \text{Innocent}(B) \Rightarrow \neg \text{Know}(B, v) \end{cases} \\ C & \begin{cases} \text{Innocent}(C) \Rightarrow \text{With}(A, v) \\ \text{Innocent}(C) \Rightarrow \text{With}(B, v) \end{cases} \end{aligned}$$

$$(1) \neg \text{Innocent}(A) \vee \text{Friend}(B, v)$$

$$(2) \neg \text{Innocent}(A) \vee \text{Hate}(C, v)$$

$$(3) \neg \text{Innocent}(B) \vee \neg \text{Intown}(B)$$

$$(4) \neg \text{Innocent}(B) \vee \neg \text{Know}(B, v)$$

$$(5) \neg \text{Innocent}(C) \vee \text{With}(A, v)$$

$$(6) \neg \text{Innocent}(C) \vee \text{With}(B, v)$$

附加常识:

$$(7) \neg \text{With}(x, v) \vee \text{Intown}(x)$$

$$(8) \neg \text{Friend}(y, x) \vee \text{Know}(y, z)$$

$$(9) \neg \text{Hate}(s, t) \vee \text{Know}(s, t)$$

$$(10) \text{Innocent}(A) \vee \text{Innocent}(B) \vee \neg \text{Innocent}(C) \quad \left. \begin{array}{l} \text{常识} \end{array} \right\}$$

$$\text{Innocent}(A) \vee \text{Innocent}(C) \vee \neg \text{Innocent}(B)$$

$$\text{Innocent}(B) \vee \text{Innocent}(C) \vee \neg \text{Innocent}(A)$$

补充关系很重要, 附加常识的分析。

$$\text{With}(x, \text{Victor}) \Rightarrow \text{Intown}(x) \quad // \text{由于受害者在城里}$$

$$\text{Friend}(y, z) \Rightarrow \text{know}(y, z) \quad // \text{是朋友肯定认识}$$

$$\text{Hate}(s, t) \Rightarrow \text{know}(s, t) \quad // \text{恨某人肯定知道某人}$$

$$\left. \begin{array}{l} \text{Innocent}(A) \wedge \text{Innocent}(B) \vee \\ \text{Innocent}(A) \wedge \text{Innocent}(C) \vee \\ \text{Innocent}(B) \wedge \text{Innocent}(C) \end{array} \right\} 3 \text{人中必有一个有罪}$$

证: 求证目标 G: B 是嫌疑犯。

$$\text{设 } \neg G = S = \text{Innocent}(B)$$

$$\text{首先将公式: } \text{Innocent}(A) \wedge \text{Innocent}(B) \vee \text{Innocent}(A) \wedge \text{Innocent}(C) \vee \text{Innocent}(B) \wedge \text{Innocent}(C)$$

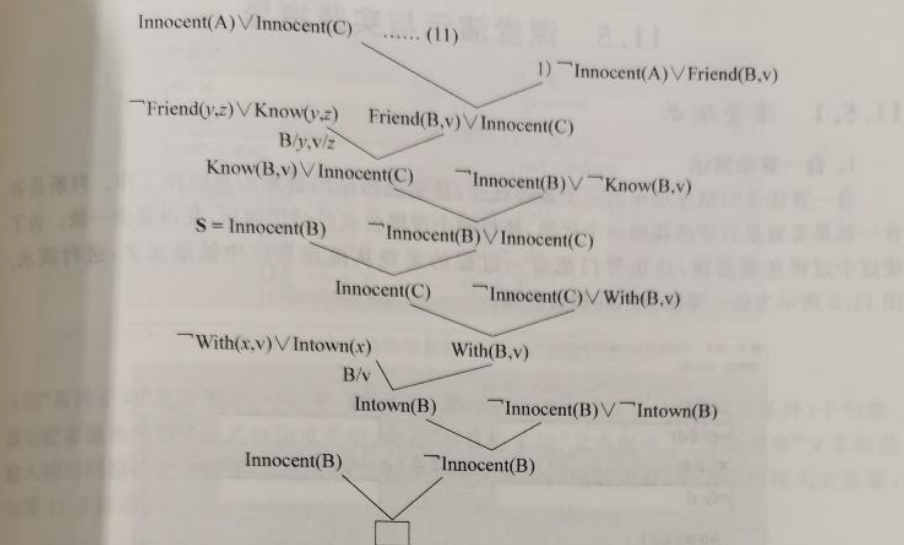
化成子句为

$$^{(11)} (\text{Innocent}(A) \vee \text{Innocent}(C)) \wedge ^{(12)} (\text{Innocent}(B) \vee \text{Innocent}(C)) \wedge ^{(13)} (\text{Innocent}(A) \vee$$

$$\text{Innocent}(B))$$

拆开后是 3 个子句, 再进行归结。

方法 1:



方法 2:

$G: \exists w \neg \text{Innocent}(w) \quad \neg G: \forall w \text{Innocent}(w)$

将 $\text{Innocent}(w) \vee \neg \text{Innocent}(w)$ 作为子句, 其中 $\neg \text{Innocent}(w)$ 只参与代换, 不参与归结(这是归结推理的一种方式, 即回答问题的方式, 看只参与代换的 $\neg \text{Innocent}(w)$ 中, w 被代换成什么, 即为答案)。

