

产生式系统 习题

第 1 题

1、对 $N=5$ 、 $k \leq 3$ 时，求解传教士和野人问题的产生式系统各组成部分进行描述（给出综合数据库、规则集合的形式化描述，给出初始状态和目标条件的描述），并画出状态空间图。

答： 1，综合数据库

定义三元组： (m, c, b)

其中： $0 \leq m \leq 5$ ，表示传教士在河左岸的人数。

$0 \leq c \leq 5$ ，表示野人在河左岸的认输。

$b \in \{0,1\}$ ， $b=1$ ，表示船在左岸， $b=0$ ，表示船在右岸。

2，规则集

规则集可以用两种方式表示，两种方法均可。

第一种方法： 按每次渡河的人数分别写出每一个规则，共(3 0)、(0 3)、(2 1)、(1 1)、(1 0)、(0 1)、(2 0)、(0 2)八种渡河的可能（其中(x y)表示 x 个传教士和 y 个野人上船渡河），因此共有 16 个规则（从左岸到右岸、右岸到左岸各八个）。注意：这里没有(1 2)，因为该组合在船上的传教士人数少于野人人数。

规则集如下：

r1: IF (m, c, 1) THEN (m-3, c, 0) r2: IF (m, c, 1) THEN (m, c-3, 0) r3: IF (m, c, 1) THEN (m-2, c-1, 0)
r4: IF (m, c, 1) THEN (m-1, c-1, 0) r5: IF (m, c, 1) THEN (m-1, c, 0) r6: IF (m, c, 1) THEN (m, c-1, 0)
r7: IF (m, c, 1) THEN (m-2, c, 0) r8: IF (m, c, 1) THEN (m, c-2, 0) r9 : IF (m, c, 0) THEN (m+3, c, 1)
r10: IF (m, c, 0) THEN (m, c+3, 1) r11: IF (m, c, 0) THEN (m+2, c+1, 1) r12: IF (m, c, 0) THEN (m+1, c+1, 1)
r13: IF (m, c, 0) THEN (m+1, c, 1) r14: IF (m, c, 0) THEN (m, c+1, 1) r15: IF (m, c, 0) THEN (m+2, c, 1)
r16: IF (m, c, 0) THEN (m, c+2, 1)

第二种方法： 将规则集综合在一起，简化表示。规则集如下：

r1: IF (m, c, 1) and $0 < i+j \leq 3$ and ($i \geq j$ or $i=0$) THEN (m-i, c-j, 0)
r2: IF (m, c, 0) and $0 < i+j \leq 3$ and ($i \geq j$ or $i=0$) THEN (m+i, c+j, 1)

3，初始状态：(5, 5, 1)

4，结束状态：(0, 0, 0)

第 2 题

2、对量水问题给出产生式系统描述，并画出状态空间图。

有两个无刻度标志的水壶，分别可装 5 升和 2 升的水。设另有一水缸，可用来向水壶灌水或倒出水，两个水壶之间，水也可以相互倾灌。已知 5 升壶为满壶，2 升壶为空壶，问如何通过倒水或灌水操作，使能在 2 升的壶中量出一升的水来。

答： 1，综合数据库

定义二元组： $(L5, L2)$

其中： $0 \leq L5 \leq 5$ ，表示容量为 5 升的壶的当前水量。

$0 \leq L2 \leq 2$ ，表示容量为 2 升的壶的当前水量。

2，规则集

r1: IF (L5, L2) THEN (5, L2) /* 将 L5 灌满水 */ r2: IF (L5, L2) THEN (L5, 2) /* 将 L2 灌满水 */
r3: IF (L5, L2) THEN (0, L2) /* 将 L5 水到光 */ r4: IF (L5, L2) THEN (L5, 0) /* 将 L2 水到光 */
r5: IF (L5, L2) and $L5+L2 \leq 5$ THEN (L5+L2, 0) /* L2 到入 L5 中 */
r6: IF (L5, L2) and $L5+L2 > 5$ THEN (5, L5+L2-5) /* L2 到入 L5 中 */

r7: IF (L5, L2) and $L5+L2 \leq 2$ THEN (0, L5+L2) /* L5 到入 L2 中 */

r8: IF (L5, L2) and $L5+L2 > 5$ THEN (L5+L2-2, 2) /* L5 到入 L2 中 */

3, 初始状态: (5, 0)

4, 结束条件: (x, 1), 其中 x 表示不定。当然结束条件也可以写成: (0, 1)

第 3 题

3、对梵塔问题给出产生式系统描述, 并讨论 N 为任意时状态空间的规模。

相传古代某处一庙宇中, 有三根立柱, 柱子上可套放直径不等的 N 个圆盘, 开始时所有圆盘都放在第一根柱子上, 且小盘处在大盘之上, 即从下向上直径是递减的。和尚们的任务是把所有圆盘一次一个地搬到另一个柱子上去 (不许暂搁地上等), 且小盘只许在大盘之上。问和尚们如何搬法最后能完成将所有的盘子都移到第三根柱子上 (其余两根柱子, 有一根可作过渡盘子使用)。

求 $N=2$ 时, 求解该问题的产生式系统描述, 给出其状态空间图。讨论 N 为任意时, 状态空间的规模。

答: 1, 综合数据库

定义三元组: (A, B, C)

其中 A, B, C 分别表示三根立柱, 均为表, 表的元素为 $1 \sim N$ 之间的整数, 表示 N 个不同大小的盘子, 数值小的数表示小盘子, 数值大的数表示大盘子。表的第一个元素表示立柱最上面的柱子, 其余类推。

2, 规则集

为了方便表示规则集, 引入以下几个函数:

first(L): 取表的第一个元素, 对于空表, first 得到一个很大的大于 N 的数值。

tail(L): 取表除了第一个元素以外, 其余元素组成的表。

cons(x, L): 将 x 加入到表 L 的最前面。

规则集:

r1: IF (A, B, C) and (first(A) < first(B)) THEN (tail(A), cons(first(A), B), C)

r2: IF (A, B, C) and (first(A) < first(C)) THEN (tail(A), B, cons(first(A), C))

r3: IF (A, B, C) and (first(B) < first(C)) THEN (A, tail(B), cons(first(B), C))

r4: IF (A, B, C) and (first(B) < first(A)) THEN (cons(first(B), A), tail(B), C)

r5: IF (A, B, C) and (first(C) < first(A)) THEN (cons(first(C), A), B, tail(C))

r6: IF (A, B, C) and (first(C) < first(B)) THEN (A, cons(first(C), B), tail(C))

3, 初始状态: ((1, 2, ..., N), (), ())

4, 结束状态: ((), (), (1, 2, ..., N))

问题的状态规模: 每一个盘子都有三中选择: 在 A 上、或者在 B 上、或者在 C 上, 共 N 个盘子, 所以共有 3^N 种可能。即问题的状态规模为 3^N 。

第 4 题

4、对猴子摘香蕉问题, 给出产生式系统描述。

一个房间里, 天花板上挂有一串香蕉, 有一只猴子可在房间里任意活动 (到处走动, 推移箱子, 攀登箱子等)。设房间里还有一只可被猴子移动的箱子, 且猴子登上箱子时才能摘到香蕉, 问猴子在某一状态下 (设猴子位置为 a, 箱子位置为 b, 香蕉位置为 c), 如何行动可摘取到香蕉。

答: 1, 综合数据库

定义 5 元组: (M, B, Box, On, H)

其中:

M: 猴子的位置 B: 香蕉的位置 Box: 箱子的位置 On=0: 猴子在地板上 On=1: 猴子在箱子上

H=0: 猴子没有抓到香蕉 H=1: 猴子抓到了香蕉

2, 规则集

r1: IF (x, y, z, 0, 0) THEN (w, y, z, 0, 0) 猴子从 x 处走到 w 处

r2: IF (x, y, x, 0, 0) THEN (z, y, z, 0, 0) 如果猴子和箱子在一起, 猴子将箱子推到 z 处

r3: IF (x, y, x, 0, 0) THEN (x, y, x, 1, 0) 如果猴子和箱子在一起, 猴子爬到箱子上

r4: IF (x, y, x, 1, 0) THEN (x, y, x, 0, 0) 如果猴子在箱子上, 猴子从箱子上下来

r5: IF (x, x, x, 1, 0) THEN (x, x, x, 1, 1) 如果箱子在香蕉处, 猴子在箱子上, 猴子摘到香蕉

其中 x, y, z, w 为变量

3, 初始状态 (c, a, b, 0, 0)

4, 结束状态 (x1, x2, x3, x4, 1) 其中 x1~x4 为变量。

第 5 题

5、对三枚钱币问题给出产生式系统描述及状态空间图。

设有三枚钱币, 其排列处在"正、正、反"状态, 现允许每次可翻动其中任意一个钱币, 问只许操作三次, 如何翻动钱币使其变成"正、正、正"或"反、反、反"状态。

答: 1, 综合数据库

定义四元组: (x, y, z, n)

其中 x,y,z ∈ [0,1], 1 表示钱币为正面, 0 表示钱币为反面。n=0,1,2,3, 表示当前状态是经过 n 次翻钱币得到的。

2, 规则库

r1: IF (x, y, z, n) THEN (~x, y, z, n+1) r2: IF (x, y, z, n) THEN (x, ~y, z, n+1) r3: IF (x, y, z, n) THEN (x, y, ~z, n+1)

其中 ~x 表示对 x 取反。

3, 初始状态 (1, 1, 0, 0)

4, 结束状态 (1, 1, 1, 3) 或者 (0, 0, 0, 3)

第 6 题

6、说明怎样才能用一个产生式系统把十进制数转换为二进制数, 并通过转换 141.125 这个数为二进制数, 阐明其运行过程。

提示: 将十进制数分为整数部分和小数部分两部分。用四元组(a, b, c, d)表示综合数据库, 其中 a, b 表示到目前为止还没有转换的十进制数的整数部分和小数部分, c, d 表示已经转换得到的二进制数的整数部分和小数部分。然后根据十进制数转换二进制数的原理, 分别定义整数的转换规则和小数的转换规则, 一次规则的执行, 转换得到二进制数的一位。

第 7 题

7、设可交换产生式系统的一条规则 R 可应用于综合数据库 D 来生成 D', 试证明若 R 存在逆, 则可应用于 D' 的规则集等同于可应用于 D 的规则集。

答: 设规则 R 的逆用 R' 表示。由题意有 R 应用于 D 后, 得到数据库 D', 由可交换系统的性质,

有: $\text{rule}(D) \subseteq \text{rule}(D')$

其中 $\text{rule}(D)$ 表示可应用于 D 的规则集合。

由于 R' 是 R 的逆, 所以 R' 应用于 D' 后, 得到数据库 D。同样由可交换系统的性质,

有: $\text{rule}(D') \subseteq \text{rule}(D)$

综合上述两个式子, 有 $\text{rule}(D') = \text{rule}(D)$ 。

第 8 题

8、一个产生式系统是以整数的集合作为综合数据库, 新的数据库可通过把其中任意一对元素的乘积添加到原数据库的操作来产生。设以某一个整数子集的出现作为目标条件, 试说明该产生式系统是可交换的。

答: 说明一个产生式系统是可交换的, 就是要证明该产生式系统满足可交换产生式系统的三条性质。

(1) 该产生式系统以整数的集合为综合数据库，其规则是将集合中的两个整数相乘后加入到数据库中。由于原来数据库是新数据库的子集，所以原来的规则在新数据库中均可以使用。所以满足可交换产生式系统的第一条性质。

(2) 该产生式系统以某个整数的子集的出现为目标条件，由于规则执行的结果只是向数据库中添加数据，如果原数据库中已经满足目标了，即出现了所需要的整数子集，规则的执行结果不会破坏该整数子集的出现，因此新的数据库仍然会满足目标条件。满足可交换产生式系统的第二性质。

(3) 设 D 是该产生式系统的一个综合数据库。对 D 施以一个规则序列后，得到一个新的数据库 D' 。该规则序列中的有些规则有些是可以应用于 D 的，这些规则用 R_1 表示。有些规则是不能应用于 D 的，这些规则用 R_2 表示。由于 R_1 中的规则可以直接应用与 D ，所以 R_1 中规则的应用与 R_2 中规则的执行结果无关，也与 R_1 中其他的规则的执行无关。所以可以认为，先将 R_1 中所有的规则对 D 应用，然后再按照原来的次序应用 R_2 中的规则。因此对于本题的情况，这样得到的综合数据库与 D' 是相同的。而由于 R_1 中一条规则的执行与其他的规则无关，所以 R_1 中规则的执行顺序不会影响到最终的结果。因此满足可交换产生式系统的第三个条件。

因此这样一个产生式系统是一个可交换的产生式系统。

产生式系统的搜索策略习题

第 1 题

1、用回溯策略求解如下所示二阶梵塔问题，画出搜索过程的状态变化示意图。

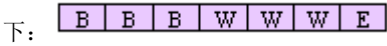


对每个状态规定的操作顺序为：先搬 1 柱的盘，放的顺序是先 2 柱后 3 柱；再搬 2 柱的盘，放的顺序是先 3 柱后 1 柱；最后搬 3 柱的盘，放的顺序是先 1 柱后 2 柱。

答： 为了方便起见，我们用((AB))(())这样的表表示一个状态。
这样得到搜索图如右图：

第 2 题

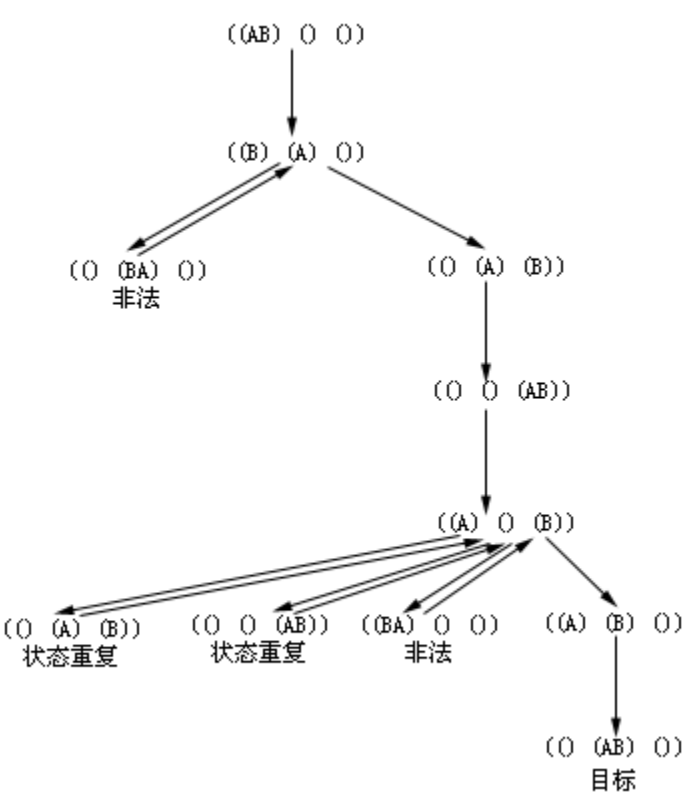
2、滑动积木块游戏的棋盘结构及某一种将牌的初始排列结构如



其中 B 表示黑色将牌，W 表示白色将牌，E 表示空格。游戏的规定走法是：

- (1) 任意一个将牌可以移入相邻的空格，规定其耗散值为 1；
- (2) 任意一个将牌可相隔 1 个或 2 个其他的将牌跳入空格，规定其耗散值等于跳过将牌的数目；游戏要达到的目标是使所有白将牌都处在黑将牌的左边（左边有无空格均可）。对这个问题，定义一个启发函数 $h(n)$ ，并给出利用这个启发函数用算法 A 求解时所产生的搜索树。你能否辨别这个 $h(n)$ 是否满足下界范围？在你的搜索树中，对所有的节点满足不满足单调限制？

提示：可定义 h 为：
 $h = \text{B 右边的 W 的数目}$



设 j 节点是 i 节点的子节点，则根据走法不同， $h(i)-h(j)$ 的值和 $C(i,j)$ 分为如下几种情况：

- (1) B 或 W 走到了相邻的一个空格位置，此时： $h(i)-h(j)=0, C(i,j)=1$ ；
- (2) W 跳过了 1 或 2 个 W，此时 $h(i)-h(j)=0, C(i,j)=1$ 或 2；
- (3) W 向右跳过了一个 B（可能同时包含一个 W），此时： $h(i)-h(j)=-1, C(i,j)=1$ 或 2；
- (4) W 向右跳过了两个 B，此时： $h(i)-h(j)=-2, C(i,j)=2$ ；
- (5) W 向左跳过了一个 B（可能同时包含一个 W），此时： $h(i)-h(j)=1, C(i,j)=1$ 或 2；
- (6) W 向左跳过了两个 B，此时： $h(i)-h(j)=2, C(i,j)=2$ ；
- (7) B 跳过了 1 或 2 个 B，此时 $h(i)-h(j)=0, C(i,j)=1$ 或 2；
- (8) B 向右跳过了一个 W（可能同时包含一个 B），此时： $h(i)-h(j)=1, C(i,j)=1$ 或 2；
- (9) B 向右跳过了两个 W，此时： $h(i)-h(j)=2, C(i,j)=2$ ；
- (10) B 向左跳过了一个 W（可能同时包含一个 B），此时： $h(i)-h(j)=-1, C(i,j)=1$ 或 2；
- (11) B 向左跳过了两个 W，此时： $h(i)-h(j)=-2, C(i,j)=2$ ；

纵上所述，无论是哪一种情况，具有：

$$h(i)-h(j) \leq C(i,j)$$

且容易验证 $h(t)=0$ ，所以该 h 是单调的。由于 h 满足单调条件，所以也一定有 $h(n) \leq h^*(n)$ ，即满足 A* 条件。

第 3 题

3、对 1.4 节中的旅行商问题，定义两个 h 函数（非零），并给出利用这两个启发函数用算法 A 求解 1.4 节中的五城市问题。讨论这两个函数是否都在 h^* 的下界范围及求解结果。

$$\sum_{i=1}^n k_i$$

答：定义 $h_1=n*k$ ，其中 n 是还未走过的城市数， k 是还未走过的城市间距离的最小值。 $h_2=\sum_{i=1}^n k_i$ ，其中 n 是还未走过的城市数， k_i 是还未走过的城市间距离中 n 个最小的距离。显然这两个 h 函数均满足 A* 条件。

第 4 题

4、2.1 节四皇后问题表述中，设应用每一条规则的耗散值均为 1，试描述这个问题 h^* 函数的一般特征。你是否认为任何 h 函数对引导搜索都是有用的？

提示：对于四皇后问题，如果放一个皇后的耗散值为 1 的话，则任何一个解的耗散值都是 4。因此如果 h 是对该耗散值的估计，是没有意义的。对于像四皇后这样的问题，启发函数应该是对找到解的可能性的评价。比如像课上讲到的，利用一个位置放皇后后，消去的对角线的长度来进行评价。

第 5 题

5、对 $N=5, k \leq 3$ 的 M-C 问题，定义两个 h 函数（非零），并给出用这两个启发函数的 A 算法搜索图。讨论用这两个启发函数求解该问题时是否得到最佳解。

答：定义 $h_1=M+C-2B$ ，其中 M, C 分别是在河的左岸的传教士人数和野人人数。 $B=1$ 表示船在左岸， $B=0$ 表示船在右岸。

也可以定义 $h_2=M+C$ 。

h_1 是满足 A* 条件的，而 h_2 不满足。

要说明 $h(n)=M+C$ 不满足 A* 条件是很容易的，只需要给出一个反例就可以了。比如状态 (1, 1, 1)， $h(n)=M+C=1+1=2$ ，而实际上只要一次摆渡就可以达到目标状态，其最优路径的耗散值为 1。所以不满足 A* 的条件。

下面我们来证明 $h(n)=M+C-2B$ 是满足 A* 条件的。

我们分两种情况考虑。先考虑船在左岸的情况。如果不考虑限制条件，也就是说，船一次可以将三人从左岸运到右岸，然后再有一个人将船送回来。这样，船一个来回可以运过河 2 人，而船仍然在左岸。而最后剩下的三个人，则可以一次将他们全部从左岸运到右岸。

所以，在不考虑限制条件的情况下，也至少需要摆渡 $\left\lceil \frac{M+C-3}{2} \right\rceil \times 2 + 1$ 次。其中分子上的“-3”表示剩下三个留待最后一次运过

$$\frac{M+C-3}{2}$$

去。除以"2"是因为一个来回可以运过去 2 人，需要 $\frac{M+C-3}{2}$ 个来回，而"来回"数不能是小数，需要向上取整，这个用符号 $\lceil \quad \rceil$ 表示。而乘以"2"是因为一个来回相当于两次摆渡，所以要乘以 2。而最后的"+1"，则表示将剩下的 3 个运过去，需要一次摆渡。化简有：

$$\left\lceil \frac{M+C-3}{2} \right\rceil \times 2 + 1 \geq \frac{M+C-3}{2} \times 2 + 1 = M+C-3+1 = M+C-2$$

再考虑船在右岸的情况。同样不考虑限制条件。船在右岸，需要一个人将船运到左岸。因此对于状态(M, C, 0)来说，其所需的最少摆渡数，相当于船在左岸时状态(M+1, C, 1)或(M, C+1, 1)所需的最少摆渡数，再加上第一次将船从右岸送到左岸的一次摆渡数。因此所需的最少摆渡数为：(M+C+1)-2+1。其中(M+C+1)的"+1"表示送船回到左岸的那个人，而最后边的"+1"，表示送船到左岸时的一次摆渡。

化简有：(M+C+1)-2+1=M+C。

综合船在左岸和船在右岸两种情况下，所需的最少摆渡次数用一个式子表示为：M+C-2B。其中 B=1 表示船在左岸，B=0 表示船在右岸。由于该摆渡次数是在不考虑限制条件下，推出的最少所需要的摆渡次数。因此，当有限制条件时，最优的摆渡次数只能大于等于该摆渡次数。所以该启发函数 h 是满足 A*条件的。

第 6 题

6、证明 OPEN 表上具有 $f(n) < f^*(s)$ 的任何节点 n，最终都将被 A*选择去扩展。

答：题目的另一个说法是：当 A*结束时，OPEN 表中任何一个具有 $f(n) < f^*(s)$ 的节点都被扩展了。

用反证法证明。

假设在 A*结束的时候，OPEN 表中有一个节点 n 没有被扩展，且 $f(n) < f^*(s)$ 。A*算法每次从 OPEN 表中取出 f 值最小的节点扩展，当该节点是目标节点时，算法结束。并且由可采纳性定理，知道这时 A*找到了从初始节点到目标节点的最佳路径，即 $f(t) = f^*(s)$ 。如果这时 OPEN 中存在 $f(n) < f^*(s)$ 的节点 n，由于 $f(n) < f(t)$ ，则这时 A*算法应选择 n 扩展，而不是目标 t，与 A*已经结束矛盾。

第 7 题

7、如果算法 A*从 OPEN 表中去掉任一节点 n，对 n 有 $f(n) > F$ ($F > f^*(s)$)，试说明为什么算法 A*仍然是可采纳的。

答：因为 A*选作扩展的任何一个节点 n，均有 $f(n) \leq f^*(s)$ ，因此 $f(n) > f^*(s)$ 的节点，不会被 A*所扩展。所以如果从 OPEN 表中去掉 $f(n) > f^*(s)$ 的节点，不会影响 A*的可采纳性。而 F 是 $f^*(s)$ 的上界范围，因此去掉 $f(n) > F$ 的节点也同样不会影响 A*的可采纳性。

第 8 题

8、用算法 A 逆向求解图 2.7 中的八数码问题，评价函数仍定义为 $f(n) = d(n) + w(n)$ 。逆向搜索在什么地方和正向搜索相会。

提示：对于 8 数码问题，逆向搜索和正向搜索是完全一样的，只是把目标状态和初始状态对调就可以了。

第 9 题

9、讨论一个 h 函数在搜索期间可以得到改善的几种方法。

提示：在搜索期间改善 h 函数，是一种动态改变 h 函数的方法。像改进的 A*算法中，对 NEST 中的节点按 g 值的大小选择待扩展的节点，相当于令这些节点的 $h=0$ ，就是动态修改 h 函数的一种方法。

由定理 6，当 h 满足单调条件时，A*所扩展的节点序列，其 f 是非递减的。对于任何节点 i, j，如果 j 是 i 的子节点，则有 $f(i) \leq f(j)$ 。利用该性质，我们可以提出另一种动态修改 h 函数的方法：

$$f(j) = \max(f(i), f(j))$$

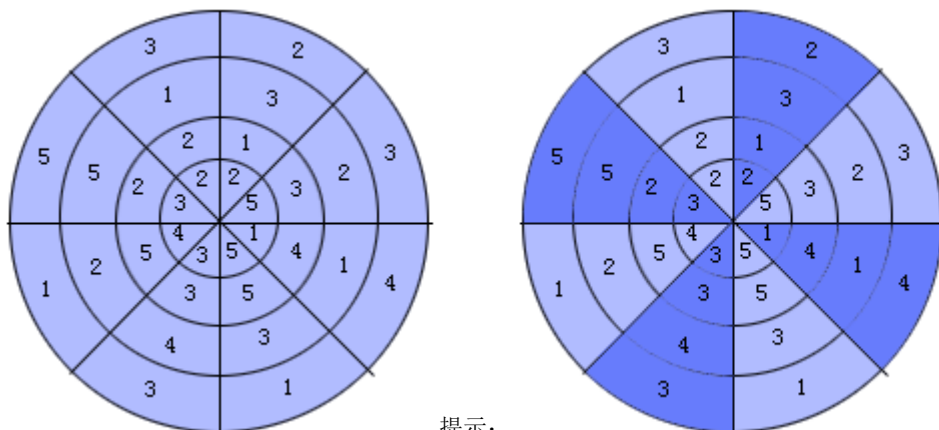
以 $f(j)$ 作为节点 j 的 f 值。f 值的改变，隐含了 h 值的改变。

当 h 不满足单调条件时，经过这样修正后的 h 具有一定的单调性质，可以减少重复节点的可能性。

第 10 题

10、四个同心圆盘的扇区数字如图所示，每个圆盘可单独转动。问如何转动圆盘使得八个径向的 4 个数字和均为 12。

提示：很多知识对求解问题有好处，这些知识并不一定要写成启发函数的形式，很多情况下，也不一定能清晰的写成一个函数的形式。



题目原图：

提示：

为了叙述方便，我们将两个相对的扇区称为相对扇区，图中阴影部分的扇区称为阴影扇区，非阴影部分的扇区称为非阴影扇区。由题意，在目标状态下，一个扇区的数字之和等于 12，一个相对扇区的数字之和等于 24，而一个阴影扇区或者非阴影扇区的数字之和为 48。为此，我们可以将目标进行分解，首先满足阴影扇区的数字之和为 48（这时非阴影部分的数字和也一定为 48）。为了这个目标我们可以通过每次转动圆盘 45° 实现。在第一个目标被满足的情况下，我们再考虑第二个目标：每一个相对扇区的数字和为 24。在实现这个目标的过程中，我们希望不破坏第一个目标。为此我们采用转动 90° 的方式实现，这样即可以调整相对扇区的数字和，又不破坏第一个目标。在第二个目标实现之后，我们就可以实现最终目标：扇区内的数字和为 12。同样我们希望在实现这个目标的时候，不破坏前两个目标。为此我们采用转动 180° 的方式实现。这样同样是即可以保证前两个目标不被破坏，又可以实现第三个目标。经过这样的分析以后，我们发现该问题就清晰多了。当然，是否每一个第一、第二个目标的实现，都能够实现第三个目标呢？有可能不一定。在这种情况下，就需要在发现第三个目标不能实现时，重新试探其他的第一、第二个目标。

与或图搜索 习题

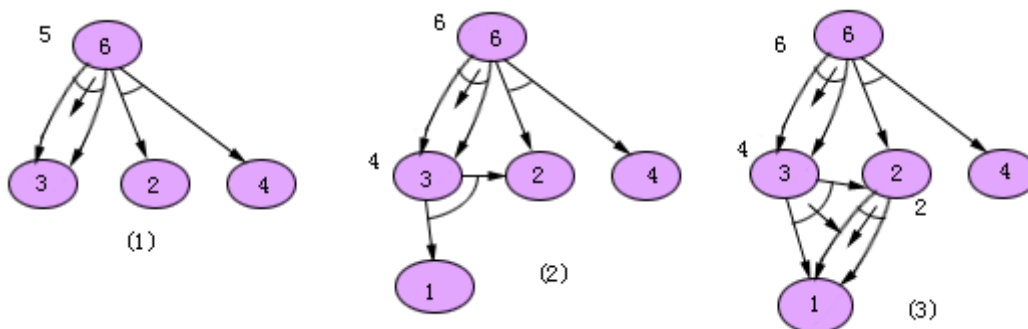
第 1 题

1、数字重写问题的变换规则如下：

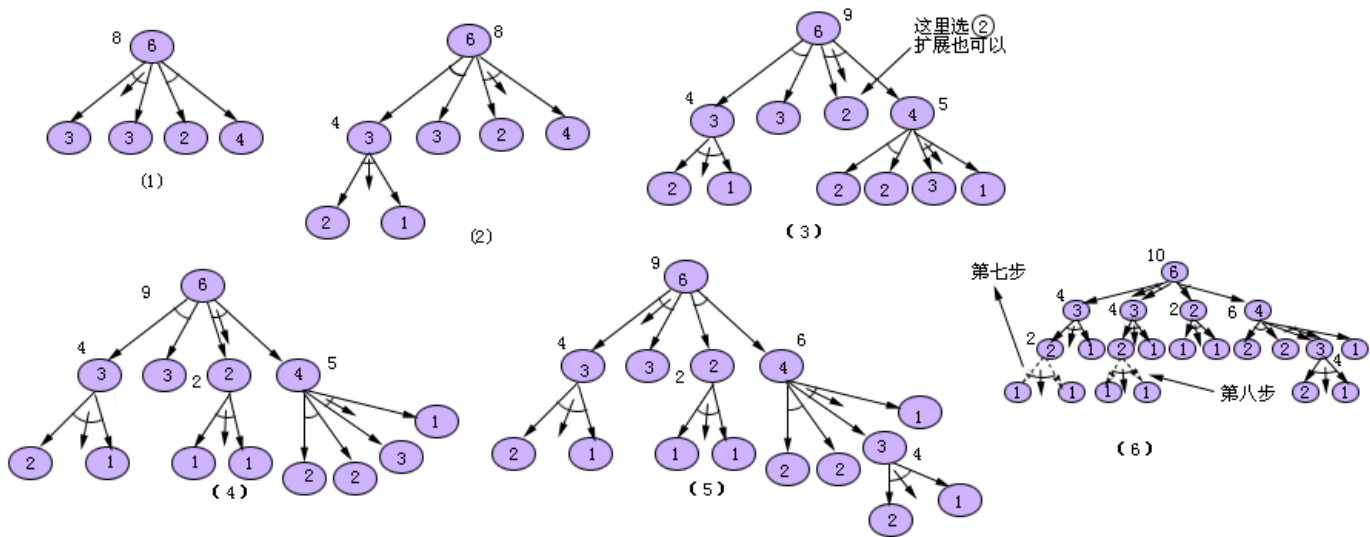
$6 \rightarrow 3, 3$ $4 \rightarrow 3, 1$
 $6 \rightarrow 4, 2$ $3 \rightarrow 2, 1$
 $4 \rightarrow 2, 2$ $2 \rightarrow 1, 1$

问如何用这些规则把数字 6 变换成一个由若干个 1 组成的数字串。试用算法 AO* 进行求解，并给出搜索图。求解时设 k-连接符的耗散值是 k 个单位，h 函数值规定为： $h(1) = 0$ ， $h(n) = n$ ($n \neq 1$)。

答：此题要求按照课中例题的方式，给出算法，以下是每个循环结束时的搜索图。



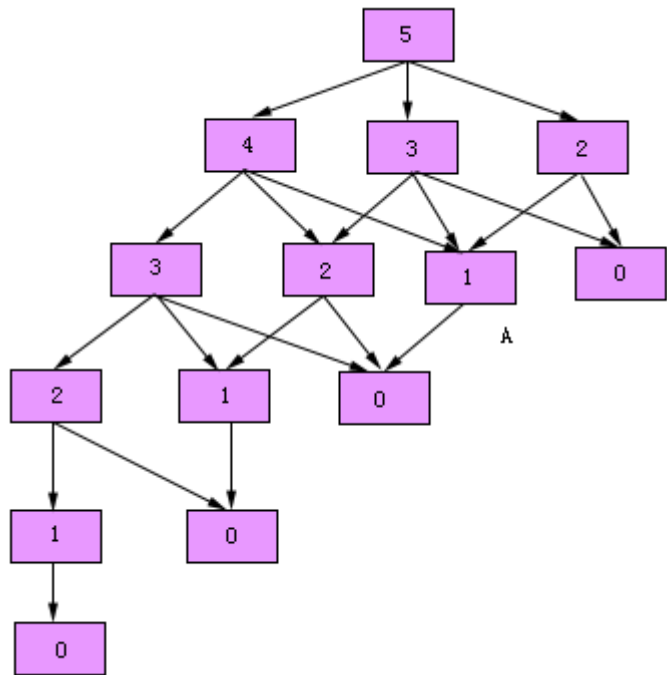
上面这种做法比较简单，也可以如下做：



第 2 题

2、余一棋的弈法如下：两棋手可以从 5 个钱币堆中轮流拿走一个、两个或三个钱币，拣起最后一个钱币者算输。试通过博弈证明，后走的选手必胜，并给出一个简单的特征标记来表示取胜策略。

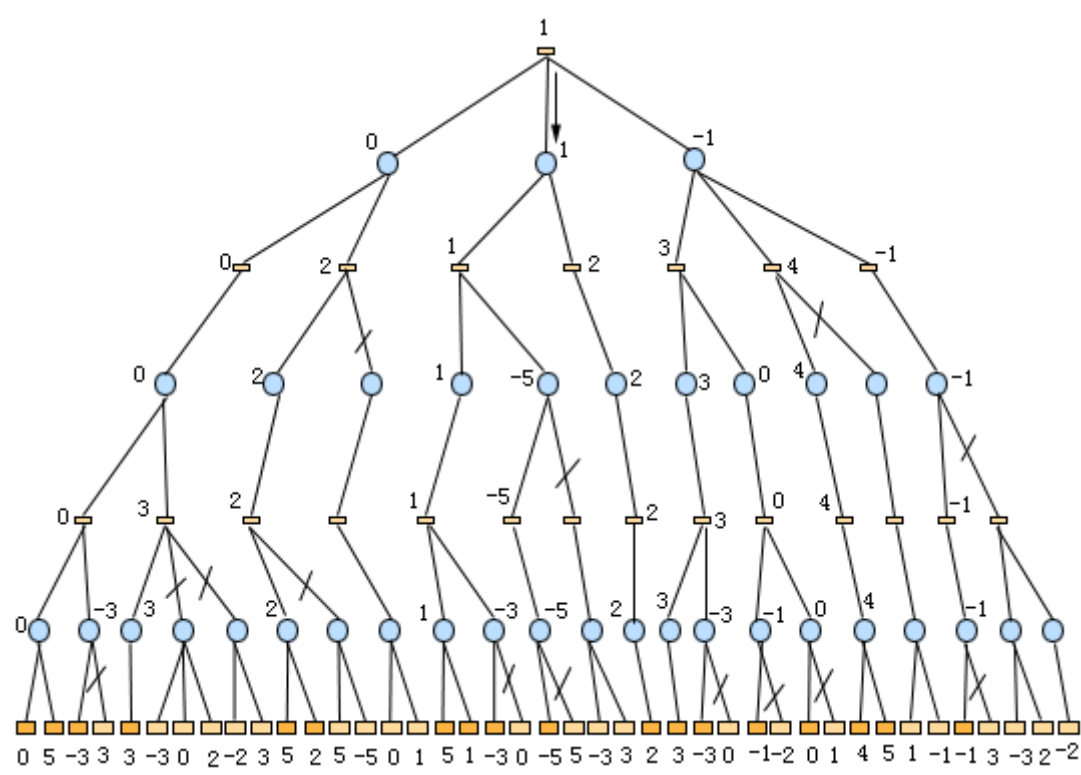
答：



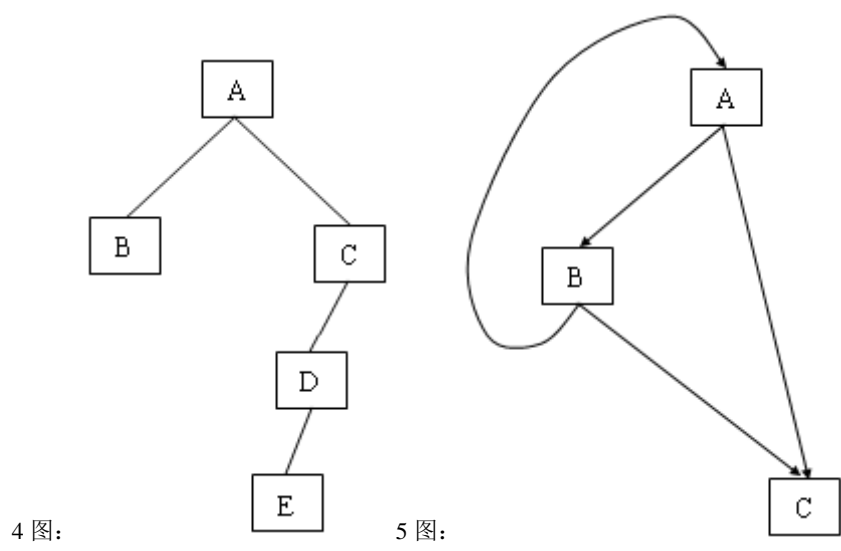
从该搜索图可以看出，无论先走者选择哪个走步，后走者都可以走到标记为 A 的节点，该节点只剩下一枚钱币，所以先走者必输。对于一般的具有 n 个钱币的情况，当 $n=4 \times m+1$ 时，后走者存在取胜策略。因为后走者可以根据先走者的走法，选择自己的走法，使得双方拿走的气币数为 4，这样经过 m 个轮回后，共拿走了 $4 \times m$ 个钱币，只剩下了一枚钱币，而此时轮到先走者走棋。所以在这种情况下，后走者存在取胜的策略。对于钱币数不等于 $4 \times m+1$ 的情况，先走者可以根据实际的气币数选择取走的气币数，使得剩下的钱币数为 $4 \times m+1$ 个，此时先走者相当于 $4 \times m+1$ 个钱币时的后走者了。因此在这种情况下，先走者存在获胜的策略。

第 3 题

3、对下图所示的博弈树，以优先生成左边节点顺序来进行 α - β 搜索，试在博弈树上给出何处发生剪枝的标记，并标明属于 α 剪枝还是 β 剪枝。



4、AO*算法中，第 7 步从 S 中选一个节点，要求其子孙不在 S 中出现，讨论应如何实现对 S 的控制使得能有效地选出这个节点。如下图所示，若 E 的耗散值发生变化时，所提出的对 S 的处理方法应能正确工作。



5、如何修改 AO*算法使之能处理出现回路的情况。如下图所示，若节点 C 的耗散值发生变化时，所修改的算法能正确处理这种情况。

6、对 3×3 的一字棋，设用+1 和-1 分别表示两选手棋子的标记，用 0 表示空格，试给出一字棋产生式系统的描述。

7、写一个 α - β 搜索的算法。

8、用一个 9 维向量 C 来表示一字棋棋盘的格局，其分量根据相应格内的×，空或○的标记分别用+1， 0， 或-1 来表示。试规定另一个 9 维向量 W，使得点积 C W 可作为 MAX 选手（棋子标记为×）估计非终端位置的一个有效的评价函数。用这个评价函数来完成几步极小-极大搜索，并分析该评价函数的效果。

逻辑与归结 习题

第 1 题

1、化下列公式成子句形式:

$$(1) (\forall x) [P(x) \rightarrow P(x)]$$

$$(2) \{\sim(\forall x) P(x)\} \rightarrow (\exists x) [\sim P(x)]$$

$$(3) \sim(\forall x) \{P(x) \rightarrow (\forall y) [P(y) \rightarrow P(f(x, y))] \wedge \sim(\forall y) [Q(x, y) \rightarrow P(y)]\}$$

$$(4) (\forall x)(\exists y) \{[P(x, y) \rightarrow Q(y, x)] \wedge [Q(y, x) \rightarrow S(x, y)]\} \rightarrow (\exists x)(\forall y) [P(x, y) \rightarrow S(x, y)]$$

答: (1) $(\forall x)[P(x) \rightarrow P(x)]$ $(\forall x)[\sim P(x) \vee P(x)]$ $\{\sim P(x) \vee P(x)\}$

$$(2) \{\sim(\forall x)P(x)\} \rightarrow (\exists x)[\sim P(x)] \quad \{(\forall x)P(x)\} \vee (\exists x)[\sim P(x)] \quad \{(\forall x)P(x)\} \vee (\exists y)[\sim P(y)]$$
$$(\forall x)(\exists y)[P(x) \vee \sim P(y)] \quad \{P(x) \vee \sim P(f(a))\}$$

$$(3) \sim(\forall x)\{P(x) \rightarrow (\forall y)[P(y) \rightarrow P(f(x, y))] \wedge \sim(\forall y)[Q(x, y) \rightarrow P(y)]\}$$

$$\sim(\forall x)\{P(x) \rightarrow \{(\forall y)[\sim P(y) \vee P(f(x, y))] \wedge \sim(\forall y)[\sim Q(x, y) \vee P(y)]\}\}$$

$$\sim(\forall x)\{P(x) \rightarrow \{(\forall y)[\sim P(y) \vee P(f(x, y))] \wedge (\exists y)[Q(x, y) \wedge \sim P(y)]\}\}$$

$$\sim(\forall x)\{P(x) \rightarrow \{(\forall y)[\sim P(y) \vee P(f(x, y))] \wedge (\exists z)[Q(x, z) \wedge \sim P(z)]\}\}$$

$$\sim(\forall x)\{\sim P(x) \vee \{(\forall y)[\sim P(y) \vee P(f(x, y))] \wedge (\exists z)[Q(x, z) \wedge \sim P(z)]\}\}$$

$$(\exists x)\{P(x) \wedge \{(\exists y)[P(y) \wedge \sim P(f(x, y))] \vee (\forall z)[\sim Q(x, z) \vee P(z)]\}\}$$

$$(\exists x)(\exists y)(\forall z)\{P(x) \wedge [P(y) \wedge \sim P(f(x, y))] \vee [\sim Q(x, z) \vee P(z)]\}$$

$$(\exists x)(\exists y)(\forall z)\{P(x) \wedge [P(y) \vee \sim Q(x, z) \vee P(z)] \wedge [\sim P(f(x, y)) \vee \sim Q(x, z) \vee P(z)]\}$$

$$\{P(a) \wedge [P(b) \vee \sim Q(a, z) \vee P(z)] \wedge [\sim P(f(a, b)) \vee \sim Q(a, z) \vee P(z)]\}$$

$$\{P(a), P(b) \vee \sim Q(a, z1) \vee P(z1), \sim P(f(a, b)) \vee \sim Q(a, z2) \vee P(z2)\}$$

$$(4) (\forall x)(\exists y)\{[P(x, y) \rightarrow Q(y, x)] \wedge [Q(y, x) \rightarrow S(x, y)]\} \rightarrow (\exists x)(\forall y)[P(x, y) \rightarrow S(x, y)]$$

$$(\forall x)(\exists y)\{[P(x, y) \rightarrow Q(y, x)] \wedge [Q(y, x) \rightarrow S(x, y)]\} \rightarrow (\exists x)(\forall y)[P(x, y) \rightarrow S(x, y)]$$

$$(\forall x)(\exists y)\{[\sim P(x, y) \vee Q(y, x)] \wedge [\sim Q(y, x) \vee S(x, y)]\} \rightarrow (\exists u)(\forall v)[\sim P(u, v) \vee S(u, v)]$$

$$\sim\{(\forall x)(\exists y)\{[\sim P(x, y) \vee Q(y, x)] \wedge [\sim Q(y, x) \vee S(x, y)]\}\} \vee (\exists u)(\forall v)[\sim P(u, v) \vee S(u, v)]$$

$$(\exists x)(\forall y)\{[P(x, y) \wedge \sim Q(y, x)] \vee [Q(y, x) \wedge \sim S(x, y)]\} \vee (\exists u)(\forall v)[\sim P(u, v) \vee S(u, v)]$$

$$(\exists x)(\forall y)(\exists u)(\forall v)\{[P(x, y) \wedge \sim Q(y, x)] \vee [Q(y, x) \wedge \sim S(x, y)]\} \vee [\sim P(u, v) \vee S(u, v)]$$

$$(\exists x)(\forall y)(\exists u)(\forall v)\{[P(x, y) \vee Q(y, x)] \wedge [P(x, y) \vee \sim S(x, y)] \wedge [\sim Q(y, x) \vee \sim S(x, y)]\} \vee [\sim P(u, v) \vee S(u, v)]$$

$$(\exists x)(\forall y)(\exists u)(\forall v)[P(x, y) \vee Q(y, x) \vee \sim P(u, v) \vee S(u, v)] \wedge [P(x, y) \vee \sim S(x, y) \vee \sim P(u, v) \vee S(u, v)] \wedge [\sim Q(y, x) \vee \sim S(x, y) \vee \sim P(u, v) \vee S(u, v)]$$

$$[P(a, y) \vee Q(y, a) \vee \sim P(f(y), v) \vee S(f(y), v)] \wedge [P(a, y) \vee \sim S(a, y) \vee \sim P(f(y), v) \vee S(f(y), v)] \wedge [\sim Q(y, a) \vee \sim S(a, y) \vee \sim P(f(y), v) \vee S(f(y), v)]$$

$$\{P(a, y1) \vee Q(y1, a) \vee \sim P(f(y1), v) \vee S(f(y1), v), P(a, y2) \vee \sim S(a, y2) \vee \sim P(f(y2), v2) \vee S(f(y2), v2), \sim Q(y3, a) \vee \sim S(a, y3) \vee \sim P(f(y3), v3) \vee S(f(y3), v3)\}$$

第 2 题

2、以一个例子证明置换的合成是不可交换的。

答: 设有两个置换 $s1=\{a/x\}$ 和 $s2=\{x/y\}$, 合适公式 $P(x, y)$ 。则:

$$P(x, y)s1s2=P(a, x)$$

$$P(x, y)s2s1=P(a, a)$$

二者不相等。所以说, 置换的合成是不可交换的。

第 3 题

3、找出集 $\{P(x, z, y), P(w, u, w), P(A, u, u)\}$ 的mgu。

答: $\{A/x, A/y, A/z, A/w, A/u\}$

第4题

4、说明下列文字集不能合一的理由:

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

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

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

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

在合一时, $f(x, x)$ 要与 $f(y, f(y, a))$ 进行合一, x 替换成 y 后, y 要与 $f(y, a)$ 进行合一, 出现了嵌套的情况, 所以不能进行合一。

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

一个是谓词 P , 一个是 P 的反, 不能合一。

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

在合一的过程中, x 替换为 $f(A)$, 而 $f(A)$ 与 A 不能合一。

第5题

5、已知两个子句为

$Loves(father(a), a)$

$\sim Loves(y, x) \vee Loves(x, y)$

试用合一算法求第一个子句和第二个子句的第一个文字合一时的结果。

答: 略

第6题

6、用归结反演法证明下列公式的永真性:

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

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

(3) $(\forall x) (\forall y) \{[P(f(x)) \wedge Q(f(B))] \rightarrow [P(f(A)) \wedge P(y) \wedge Q(y)]\}$

(4) $(\forall x) (\forall y) P(x, y) \rightarrow (\forall y) (\forall x) P(x, y)$

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

答: (1) $(\forall x) \{[P(x) \rightarrow P(A)] \wedge [P(x) \rightarrow P(B)]\}$

目标取反化子句集:

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

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

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

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

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

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

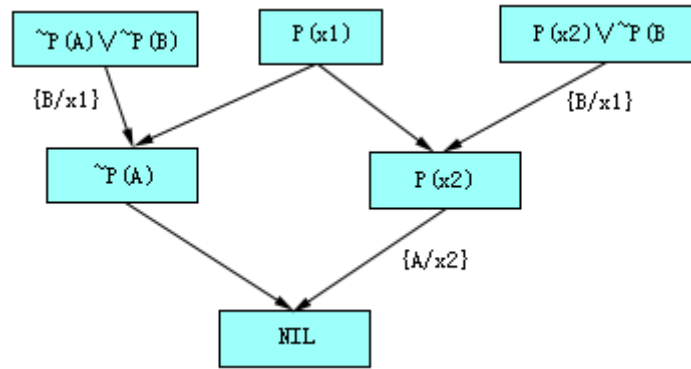
得子句集:

1, $P(x_1)$

2, $\sim P(A) \vee P(x_2)$

3, $P(x_3) \vee \sim P(B)$

4, $\sim P(A) \vee \sim P(B)$



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

目标取反化子句集:

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

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

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

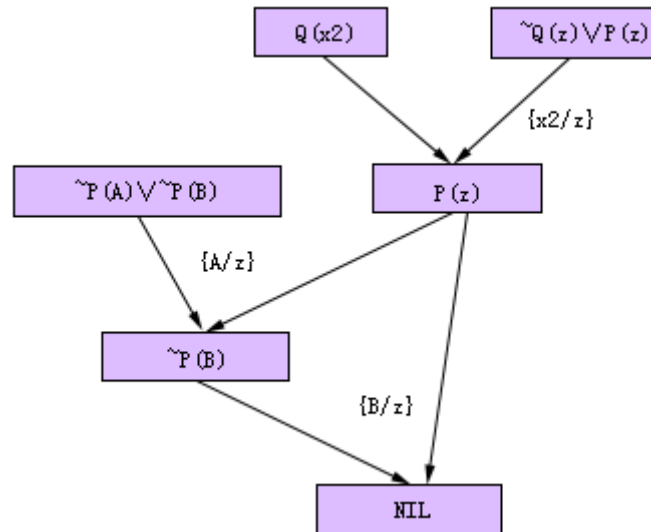
$(\forall z)(\forall x)[\sim Q(z) \vee P(z)] \wedge \{[Q(x) \wedge \sim P(A)] \vee [Q(x) \wedge \sim P(B)]\}$

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

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

得子句集:

- 1, $\sim Q(z) \vee P(z)$
- 2, $Q(x2)$
- 3, $Q(x3) \vee \sim P(B)$
- 4, $\sim P(A) \vee Q(x4)$
- 5, $\sim P(A) \vee \sim P(B)$



(3) $(\forall x) (\forall y) \{ [P(f(x)) \wedge Q(f(B))] \rightarrow [P(f(A)) \wedge P(y) \wedge Q(y)] \}$

目标取反化子句集:

$\sim(\forall x)(\forall y)\{[P(f(x)) \wedge Q(f(B))] \rightarrow [P(f(A)) \wedge P(y) \wedge Q(y)]\}$

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

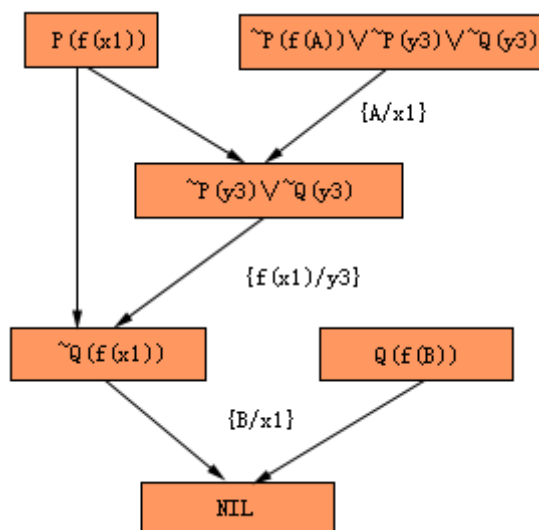
$(x)(y)\{[P(f(x)) \wedge Q(f(B))] \wedge [\sim P(f(A)) \vee \sim P(y) \vee \sim Q(y)]\}$

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

得子句集:

- 1, $P(f(x1))$

- 2, $Q(f(B))$
 3, $\sim P(f(A)) \vee \sim P(y_3) \vee \sim Q(y_3)$



(4) $(\forall x)(\forall y)P(x, y) \rightarrow (\forall y)(\exists x)P(x, y)$

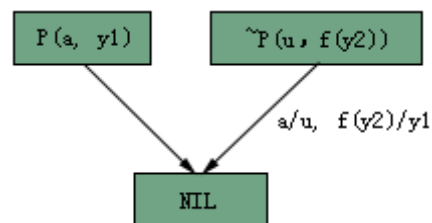
目标取反化子句集:

$\sim\{(\forall x)(\forall y)P(x, y) \rightarrow (\forall y)(\exists x)P(x, y)\}$
 $\sim\{[(\forall x)(\forall y)P(x, y)] \vee (\forall y)(\exists x)\sim P(x, y)\}$
 $\sim\{[(\forall x)(\forall y)P(x, y)] \vee (\forall v)(\exists u)\sim P(u, v)\}$
 $[(\forall x)(\forall y)P(x, y)] \wedge (\forall v)(\exists u)\sim P(u, v)$
 $(\forall x)(\forall y)(\forall v)(\exists u)P(x, y) \wedge \sim P(u, v)$

$P(a, y) \wedge \sim P(u, f(y))$

得子句集:

- 1, $P(a, y_1)$
 2, $\sim P(u, f(y_2))$



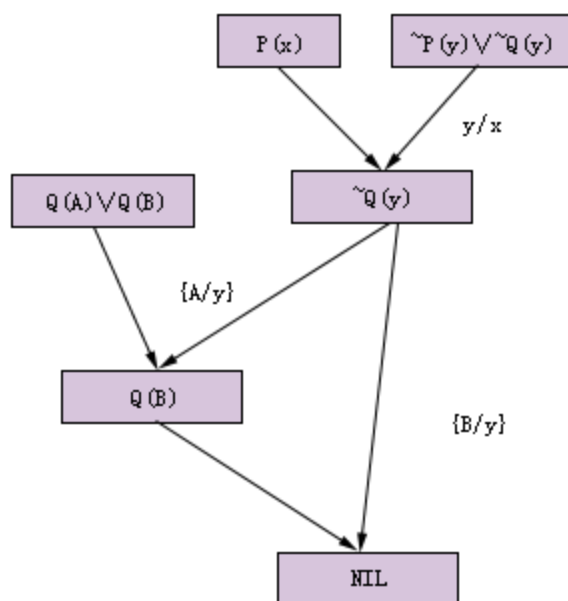
(5) $(\forall x)\{P(x) \wedge [Q(A) \vee Q(B)]\} \rightarrow (\exists x)[P(x) \wedge Q(x)]$

目标取反化子句集:

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

得子句集:

- 1, $P(x)$
 2, $Q(A) \vee Q(B)$
 3, $\sim P(y) \vee \sim Q(y)$



第 7 题

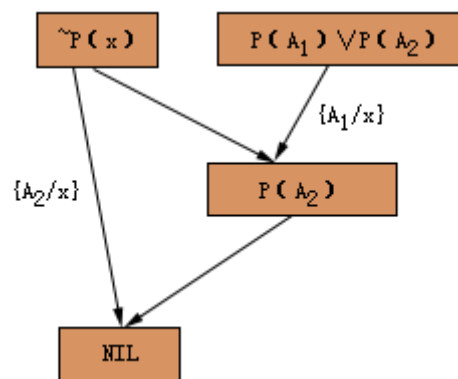
7、以归结反演法证明公式 $(\forall x)P(x)$ 是 $[P(A_1) \vee P(A_2)]$ 的逻辑推论，然而， $(\forall x)P(x)$ 的 Skolem 形即 $P(A)$ 并非 $[P(A_1) \vee P(A_2)]$ 的逻辑推论，请加以证明。

答：（1）将 $(\forall x)P(x)$ 取反化为子句：

$$\sim (\forall x) P(x) = (\exists x) \sim P(x)$$

与条件 $[P(A_1) \vee P(A_2)]$ 合在一起得子句集:

$$\{\sim P(x), P(A_1) \vee P(A_2)\}$$



所以, 公式 $(\forall x) P(x)$ 是 $[P(A_1) \vee P(A_2)]$ 的逻辑推论。

(2) 对于 $(\forall x) P(x)$ 的Skolem形, 即 $P(A)$, 取反后为 $\sim P(A)$, 与条件 $[P(A_1) \vee P(A_2)]$ 合在一起得子句集:

$$\{\sim P(A), P(A_1) \vee P(A_2)\}$$

该子句集不能进行归结, 故 $P(A)$ 不是 $[P(A_1) \vee P(A_2)]$ 的逻辑推论。

第8题

8、给定下述语句:

John likes 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) 用归结法证明"John likes peanuts。"

(2) 用归结法提取回答"What food does Sue eat?"

答: 该问题用谓词公式描述如下:

已知:

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

$$(2) Food(Apple)$$

$$(3) (\forall x)(\forall y)\{[Eat(y, x) \wedge \sim Kill(x, y)] \rightarrow Food(x)\}$$

$$(4) Eat(Bill, Peanut) \wedge \sim Kill(Penut, Bill)$$

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

目标 1: $Like(John, Peanut)$

目标 2: $(\forall x)Food(x) \wedge Eat(Sue, x)$

已知条件化子句集:

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

$$= (\forall x)\{\sim Food(x) \vee Like(John, x)\}$$

$$\Rightarrow \{\sim Food(x) \vee Like(John, x)\}$$

$$(2) Food(Apple)$$

$$(3) (\forall x)(\forall y)\{[Eat(y, x) \wedge \sim Kill(x, y)] \rightarrow Food(x)\}$$

$$= (\forall x)(\forall y)\{\sim [Eat(y, x) \wedge \sim Kill(x, y)] \vee Food(x)\}$$

$$= (\forall x)(\forall y)\{\sim[\text{Eat}(y, x) \vee \text{Kill}(x, y)] \vee \text{Food}(x)\}$$

$$\Rightarrow \{\sim\text{Eat}(y, x) \vee \text{Kill}(x, y) \vee \text{Food}(x)\}$$

$$(4) \text{Eat}(\text{Bill}, \text{Peanut}) \wedge \sim\text{Kill}(\text{Penut}, \text{Bill})$$

$$\Rightarrow \{\text{Eat}(\text{Bill}, \text{Peanut}), \sim\text{Kill}(\text{Penut}, \text{Bill})\}$$

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

$$= (\forall x)\{\sim\text{Eat}(\text{Bill}, x) \vee \text{Eat}(\text{Sue}, x)\}$$

$$\Rightarrow \sim\text{Eat}(\text{Bill}, x) \vee \text{Eat}(\text{Sue}, x)$$

目标 1 取反化子句集:

$\sim\text{Like}(\text{John}, \text{Peanut})$

目标 2 取反化子句集:

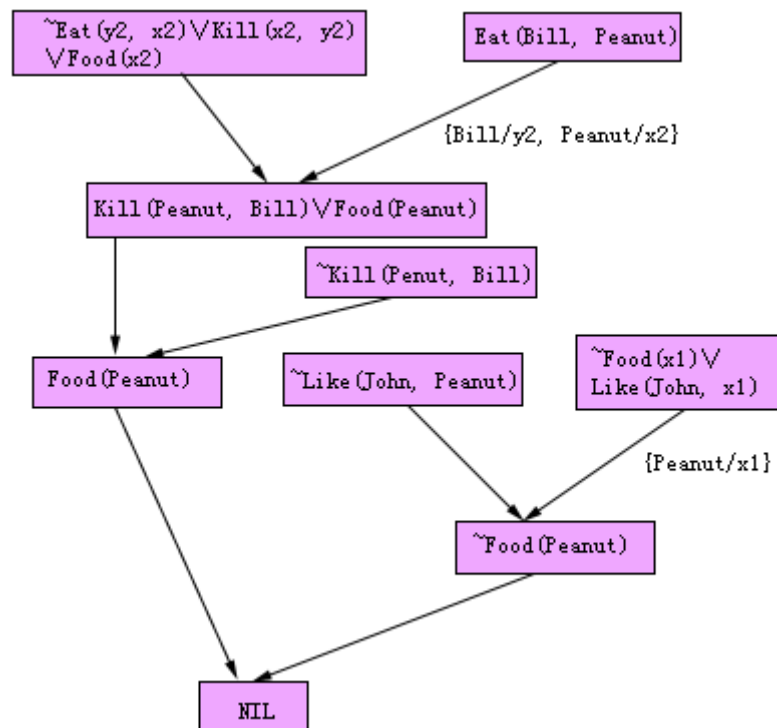
$\sim\{(\exists x)\text{Food}(x) \wedge \text{Eat}(\text{Sue}, x)\}$

$$= (\forall x)\sim\text{Food}(x) \vee \sim\text{Eat}(\text{Sue}, x)$$

$$\Rightarrow \sim\text{Food}(x) \vee \sim\text{Eat}(\text{Sue}, x)$$

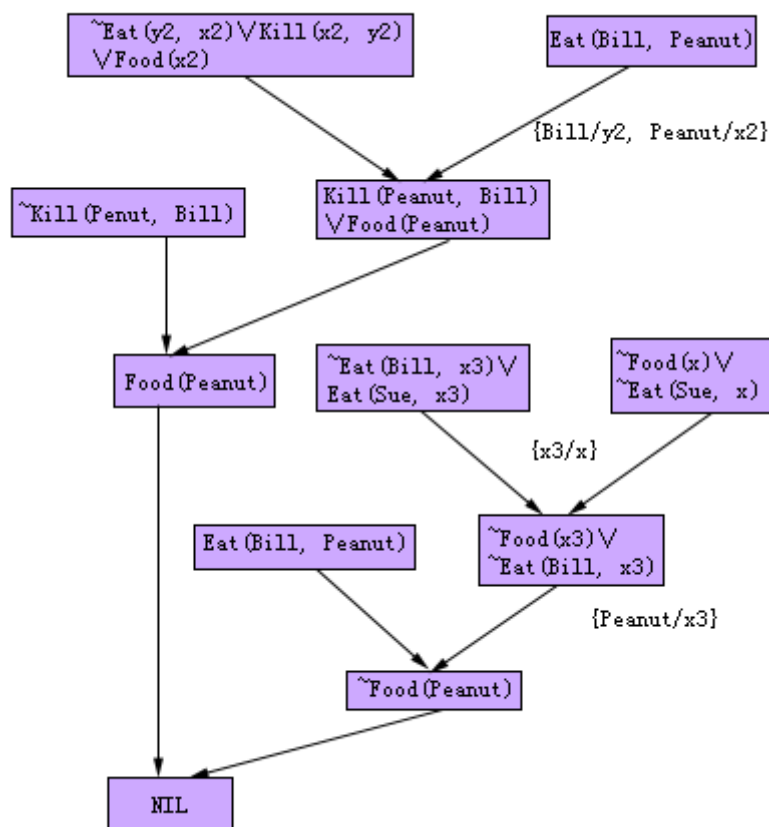
对于目标 1, 经变量换名后, 得子句集:

$\{\sim\text{Food}(x_1) \vee \text{Like}(\text{John}, x_1), \text{Food}(\text{Apple}), \sim\text{Eat}(y_2, x_2) \vee \text{Kill}(x_2, y_2) \vee \text{Food}(x_2), \text{Eat}(\text{Bill}, \text{Peanut}), \sim\text{Kill}(\text{Penut}, \text{Bill}), \sim\text{Eat}(\text{Bill}, x_3) \vee \text{Eat}(\text{Sue}, x_3), \sim\text{Like}(\text{John}, \text{Peanut})\}$ 归结树如下:

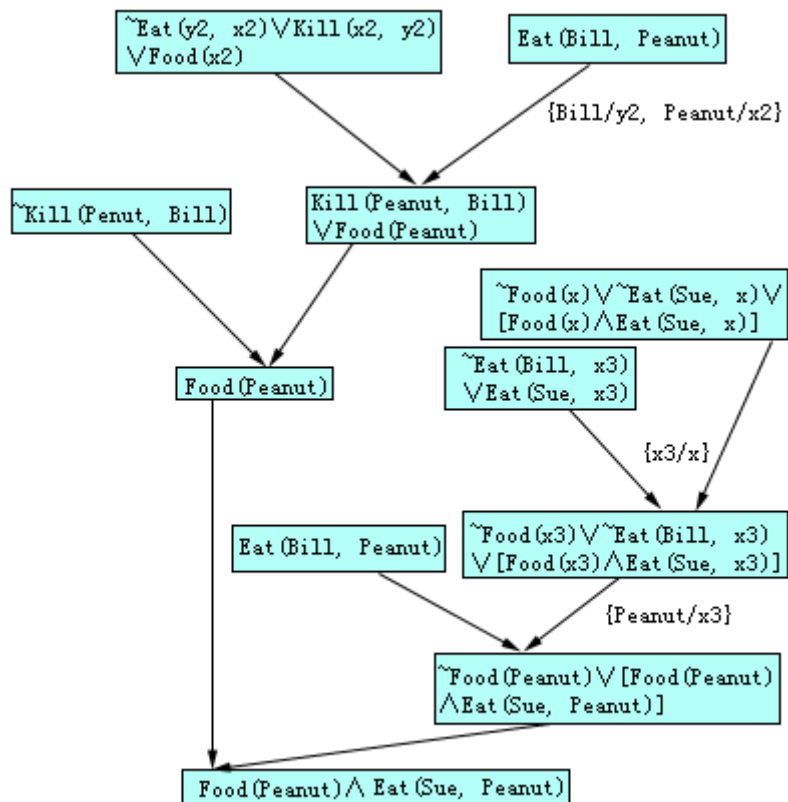


对于目标 2, 经变量换名后, 得子句集:

$\{\sim\text{Food}(x_1) \vee \text{Like}(\text{John}, x_1), \text{Food}(\text{Apple}), \sim\text{Eat}(y_2, x_2) \vee \text{Kill}(x_2, y_2) \vee \text{Food}(x_2), \text{Eat}(\text{Bill}, \text{Peanut}), \sim\text{Kill}(\text{Penut}, \text{Bill}), \sim\text{Eat}(\text{Bill}, x_3) \vee \text{Eat}(\text{Sue}, x_3), \sim\text{Food}(x) \vee \sim\text{Eat}(\text{Sue}, x)\}$ 归结树如下:



修改证明树如下:



得到解答为：Food(Peanut) ∧ Eat(Sue, Peanut)

第 9 题

9、已知事实公式为

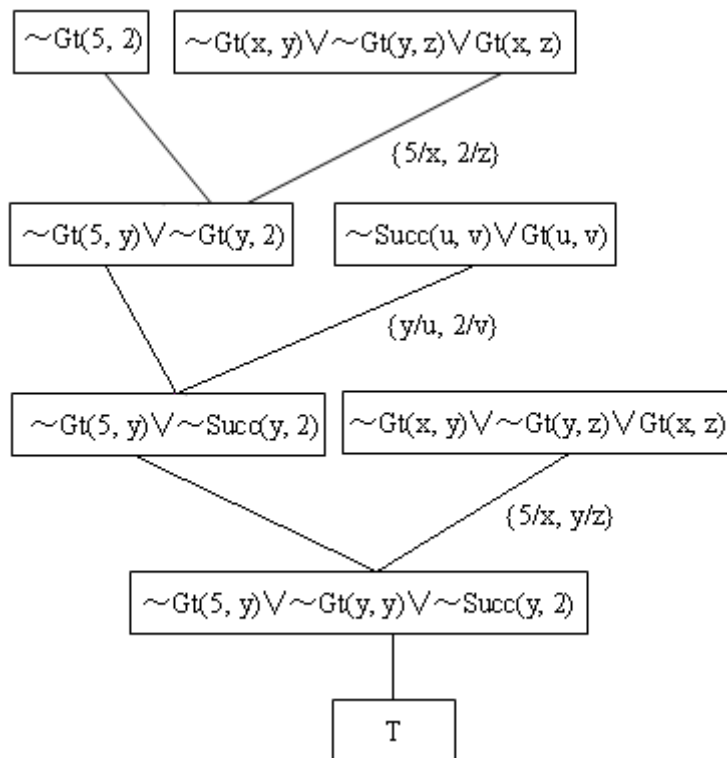
$(\forall x)(\forall y)(\forall z)(Gt(x, y) \wedge Gt(y, z) \rightarrow Gt(x, z))$

$(\forall u)(\forall v)(Succ(u, v) \rightarrow Gt(u, v))$

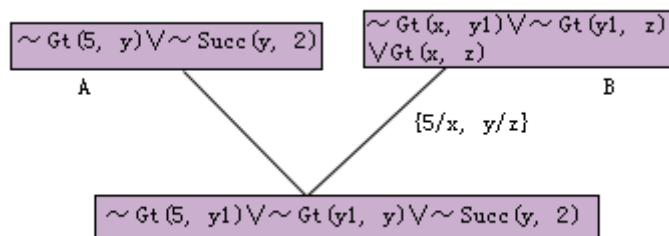
$(\forall x)(\neg Gt(x, x))$

求证 $Gt(5, 2)$

试判断下面的归结过程是否正确？若有错误应如何改进：



答：该归结过程存在错误。其原因是由于不同的子句用了相同的变量名引起的。如上图中的 A、B 两个子句的归结，两个子句中的 y 应该是不同的变量，在归结时，如果用不同的变量分别表示，就不会出现这样的问题了。比如 B 中的 y 用 y1 代替，则归结结果如下：



第 10 题

10、设公理集为

$(\forall u) \text{LAST}(\text{cons}(u, \text{NIL}), u)$ (cons 是表构造函数)

$(\forall x)(\forall y)(\forall z)(\text{LAST}(y, z) \rightarrow \text{LAST}(\text{cons}(x, y), z))$ (LAST(x, y) 代表 y 是表 x 的最末元素)

(1) 用归结反演法证明如下定理: $(\forall v) \text{LAST}(\text{cons}(2, \text{cons}(1, \text{NIL})), v)$

(2) 用回答提取过程求表 (2, 1) 的最末元素 v。

(3) 简要描述如何使用这个方法求长表的最末元素。

答: 化子句集:

$$(\forall u) \text{LAST}(\text{cons}(u, \text{NIL}), u)$$

$$\Rightarrow \text{LAST}(\text{cons}(u, \text{NIL}), u)$$

$$(\forall x) (\forall y) (\forall z) (\text{LAST}(y, z) \rightarrow \text{LAST}(\text{cons}(x, y), z))$$

$$= (\forall x) (\forall y) (\forall z) (\sim \text{LAST}(y, z) \vee \text{LAST}(\text{cons}(x, y), z))$$

$$\Rightarrow \sim \text{LAST}(y, z) \vee \text{LAST}(\text{cons}(x, y), z)$$

目标取反:

$$\sim (\forall v) \text{LAST}(\text{cons}(2, \text{cons}(1, \text{NIL})), v)$$

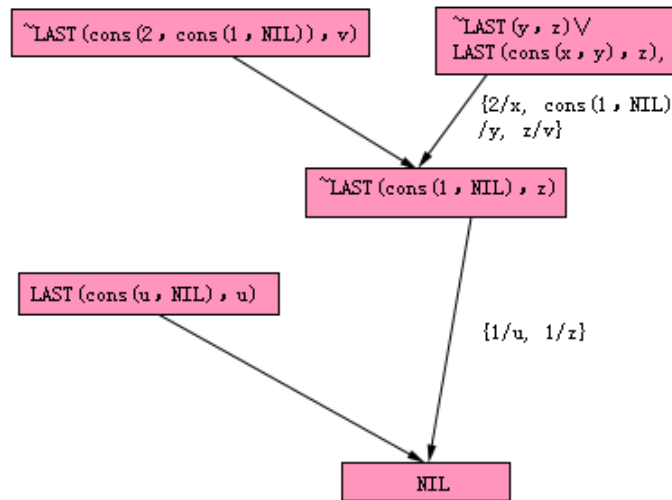
$$= (\exists v) \sim \text{LAST}(\text{cons}(2, \text{cons}(1, \text{NIL})), v)$$

$$\Rightarrow \sim \text{LAST}(\text{cons}(2, \text{cons}(1, \text{NIL})), v)$$

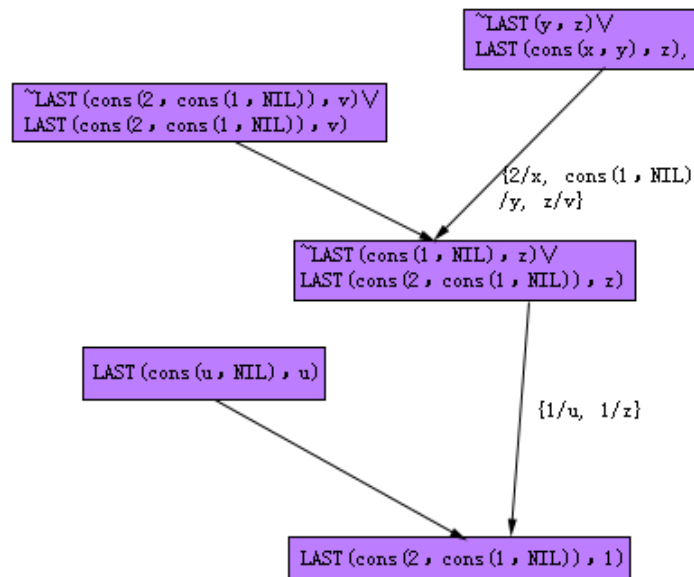
经变量换名后, 得子句集:

$$\{\text{LAST}(\text{cons}(u, \text{NIL}), u), \sim \text{LAST}(y, z) \vee \text{LAST}(\text{cons}(x, y), z), \sim \text{LAST}(\text{cons}(2, \text{cons}(1, \text{NIL})), v)\}$$

归结树如下:



修改证明树:



得到解答：LAST(cons(2, cons(1, NIL)), 1), 表 cons(2, cons(1, NIL))的最后一个元素为 1。

通过以上归结过程，我们可以看出，该方法求解长表的最后一个元素的方法是，每次将长表去掉第一个元素，直到最后得到了只有一个元素的表，该元素就是长表的最后一个元素。

第 11 题

11、对一个基于规则的几何定理证明系统，把下列语句表示成产生式规则：

- (1) 两个全等的三角形的对应角相等。
- (2) 两个全等的三角形的对应边相等。
- (3) 如果两个三角形对应边是相等的，则这两个三角形全等。
- (4) 一个等腰三角形的底角是相等的。

答：略

第 12 题

12、我们来考虑下列一段知识：Tony、Mike 和 John 属于 Alpine 俱乐部，Alpine 俱乐部的每个成员不是滑雪运动员就是一个登山运动员，登山运动员不喜欢雨而且任一不喜欢雪的人不是滑雪运动员，Mike 讨厌 Tony 所喜欢的一切东西，而喜欢 Tony 所讨厌的一切东西，Tony 喜欢雨和雪。以谓词演算语句的集合表示这段知识，这些语句适合一个逆向的基于规则的演绎系统。试说明这样一个系统怎样才能回答问题"有没有 Alpine 俱乐部的一个成员，他是一个登山运动员但不是一个滑雪运动员呢？"

答：我们用 Skier(x)表示 x 是滑雪运动员，Alpinist(x)表示 x 是登山运动员，Alpine(x)表示 x 是 Alpine 俱乐部的成员。

问题用谓词公式表示如下：

已知：

- (1) Alpine(Tony)
- (2) Alpine(Mike)
- (3) Alpine(John)
- (4) $(x)\{Alpine(x) \rightarrow [Skier(x) \vee Alpinist(x)]\}$
- (5) $(x)\{Alpinist(x) \rightarrow \sim Like(x, Rain)\}$
- (6) $(x)\{\sim Like(x, Snow) \rightarrow \sim Skier(x)\}$
- (7) $(x)\{Like(Tony, x) \rightarrow \sim Like(Mike, x)\}$
- (8) $(x)\{\sim Like(Tony, x) \rightarrow Like(Mike, x)\}$
- (9) Like(Tony, Snow)
- (10) Like(Tony, Rain)

目标： $(\forall x)\{Alpine(x) \wedge Alpinist(x) \wedge \sim Skier(x)\}$

化子句集：

- (1) Alpine(Tony)
- (2) Alpine(Mike)
- (3) Alpine(John)
- (4) $(x)\{Alpine(x) \rightarrow [Skier(x) \vee Alpinist(x)]\} = (x)\{\sim Alpine(x) \vee [Skier(x) \vee Alpinist(x)]\} \Rightarrow \sim Alpine(x) \vee Skier(x) \vee Alpinist(x)$
- (5) $(x)\{Alpinist(x) \rightarrow \sim Like(x, Rain)\} = (x)\{\sim Alpinist(x) \vee \sim Like(x, Rain)\} \Rightarrow \sim Alpinist(x) \vee \sim Like(x, Rain)$
- (6) $(x)\{\sim Like(x, Snow) \rightarrow \sim Skier(x)\} = (x)\{Like(x, Snow) \vee \sim Skier(x)\} \Rightarrow Like(x, Snow) \vee \sim Skier(x)$
- (7) $(x)\{Like(Tony, x) \rightarrow \sim Like(Mike, x)\} = (x)\{\sim Like(Tony, x) \vee \sim Like(Mike, x)\} \Rightarrow \sim Like(Tony, x) \vee \sim Like(Mike, x)$
- (8) $(x)\{\sim Like(Tony, x) \rightarrow Like(Mike, x)\} = (x)\{Like(Tony, x) \vee Like(Mike, x)\} \Rightarrow Like(Tony, x) \vee Like(Mike, x)$
- (9) Like(Tony, Snow) (10) Like(Tony, Rain)

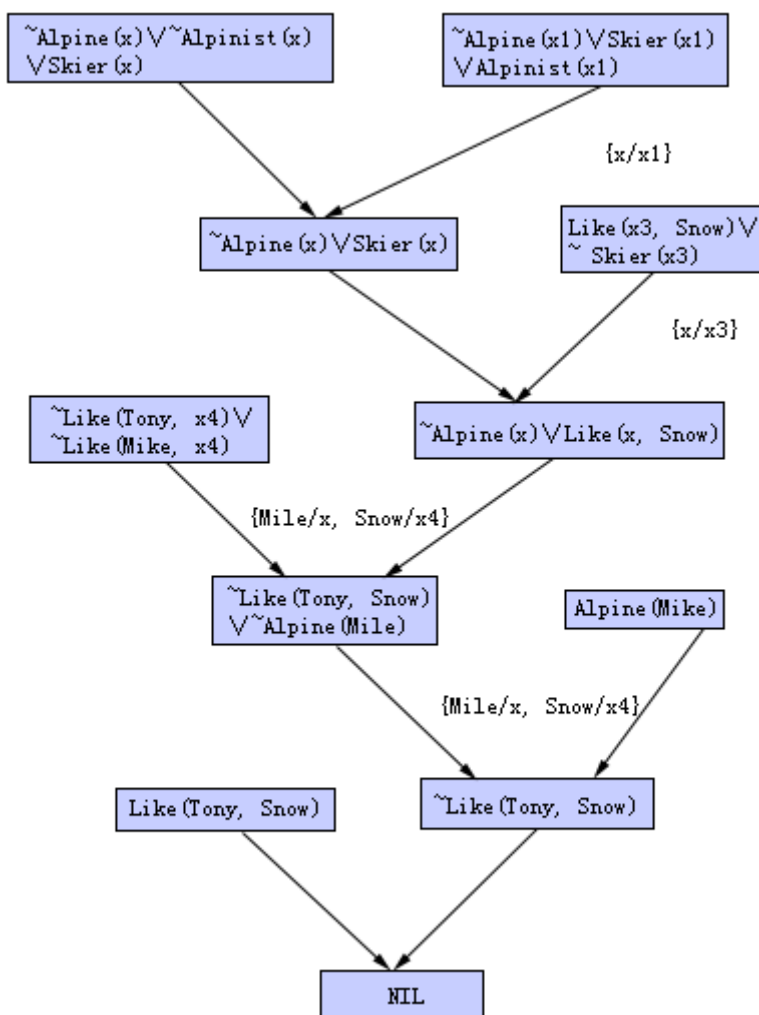
目标取反：

$$\begin{aligned} & \sim(\forall x)\{Alpine(x) \wedge Alpinist(x) \wedge \sim Skier(x)\} \\ & = (x)\{\sim Alpine(x) \vee \sim Alpinist(x) \vee Skier(x)\} \end{aligned}$$

$\Rightarrow \sim \text{Alpine}(x) \vee \sim \text{Alpinist}(x) \vee \text{Skier}(x)$

经变量换名后，得到子句集：

$\{\text{Alpine}(\text{Tony}), \text{Alpine}(\text{Mike}), \text{Alpine}(\text{John}), \sim \text{Alpine}(x1) \vee \text{Skier}(x1) \vee \text{Alpinist}(x1), \sim \text{Alpinist}(x2) \vee \sim \text{Like}(x2, \text{Rain}), \text{Like}(x3, \text{Snow}) \vee \sim \text{Skier}(x3), \sim \text{Like}(\text{Tony}, x4) \vee \sim \text{Like}(\text{Mike}, x4), \text{Like}(\text{Tony}, x5) \vee \text{Like}(\text{Mike}, x5), \text{Like}(\text{Tony}, \text{Snow}), \text{Like}(\text{Tony}, \text{Rain}), \sim \text{Alpine}(x) \vee \sim \text{Alpinist}(x) \vee \text{Skier}(x)\}$ 归结树如下：



第 13 题

13、一个积木世界的状态由下列公式集描述：

ONTABLE (A) CLEAR (E)
 ONTABLE (C) CLEAR (D)
 ON (D, C) HEAVY (D)
 ON (B, A) WOODEN (B)
 HEAVY (B) ON (E, B)

绘出这些公式所描述的状态的草图。

下列语句提供了有关这个积木世界的一般知识：

每个大的蓝色积木块是在一个绿色积木块上。

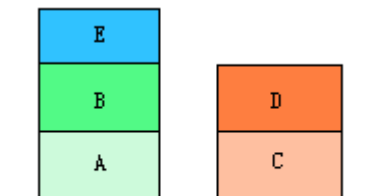
每个重的木制积木块是大的。

所有顶上没有东西的积木块都是蓝色的。

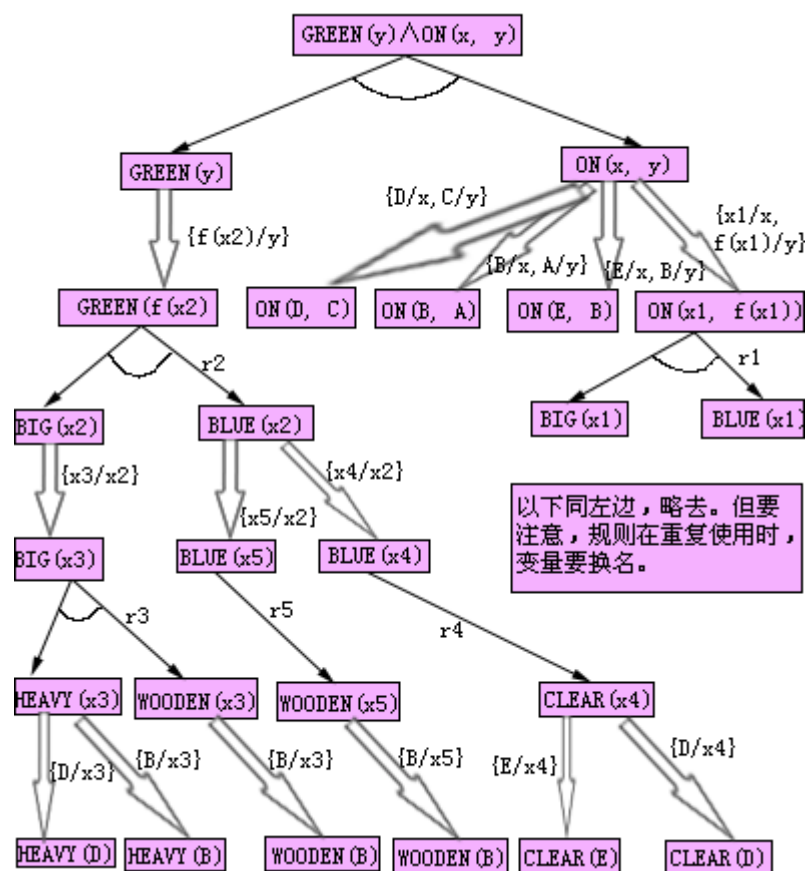
所有木制积木块是蓝色的。

以具有单文字后项的蕴涵式的集合表示这些语句。绘出能求解"哪个积木块是在绿积木上"这个问题的一致解图（用 B 规则）。

答： 状态草图：



状态草图：



知识的谓词表示：

$(\forall x)(\exists y)\{[BIG(x) \wedge BLUE(x)] \rightarrow ON(x, y) \wedge GREEN(y)\}$

$(\forall x)\{[HEAVY(x) \wedge WOODEN(x)] \rightarrow BIG(x)\}$

$(\forall x)\{CLEAR(x) \rightarrow BLUE(x)\}$

$(\forall x)\{WOODEN(x) \rightarrow BLUE(x)\}$

目标： $(\exists x)(\exists y)GREEN(y) \wedge ON(x, y)$

对规则 Skolem 化, 对目标用对偶形式 Skolem 化后,

整理得：

事实：

ONTABLE (A) CLEAR (E)

ONTABLE (C) CLEAR (D)

ON (D, C) HEAVY (D)

ON (B, A) WOODEN (B)

HEAVY (B) ON (E, B)

规则：

r1:[BIG(x1) ∧ BLUE(x1)] → ON(x1, f(x1))

r2:[BIG(x2) ∧ BLUE(x2)] → GREEN(f(x2))

r3:[HEAVY(x3) ∧ WOODEN(x3)] → BIG(x3)

r4:CLEAR(x4) → BLUE(x4)

r5:WOODEN(x5) → BLUE(x5)

目标： GREEN(y) ∧ ON(x, y)

容易验证，只有一个解图是一致的，其合一复合为：

$\{B/x, f(B)/y\}$

带入目标公式，得到解答： $GREEN(f(B)) \wedge ON(B, f(B))$

其含义是，积木 B 在绿色积木上边。这里的 f(B)可以理解为 B 下面那个积木。

综合练习 1

一、判断

1、比起极小 -- 极大法来， α - β 剪枝法增大了找不到最佳走步的危险性，但其效率较高。（错）

2、在 A 算法中，满足单调条件的 h 必然满足 A*算法的条件。（对）

3、设有机器人走迷宫问题，其入口坐标为 (x_0, y_0) ，出口坐标为 (x_t, y_t) ，当前机器人位置为 (x, y) ，若定义 $h = \sqrt{(x_t - x)^2 + (y_t - y)^2}$ ，当从入口到出口存在通路时，用 A 算法求解该问题，定能找到从入口到出口的最佳路径。（对）

二、 填空题

1、在修正的 A 算法中，fm 的含义是_____

解答：1、在修正的 A 算法中，fm 的含义是 到当前为止，扩展的节点中，f 的最大值

2、对任意节点 n，设 m 是 n 的子节点，当 h 满足条件时，称 h 是单调的。_____

解答：2、对任意节点 n，设 m 是 n 的子节点，当 h 满足条件 $h(n) - h(m) \leq C(n, m)$, $h(t) = 0$ 时，称 h 是单调的。

三、 问答题

第 1 题

1、简述用 A*算法求解问题时为什么会出现重复扩展节点问题，解决的方法有哪些？

答：当问题有解时，A*算法总是找到问题的最优解结束。如果 h 函数定义的不合理，则当扩展一个节点时，不一定就找到了从初始节点到该节点的最优路径，对于这样的节点，就有可能被多次扩展。特别是如果这样的节点处于问题的最优解路径上时，则一定会被多次扩展。解决的方法一是对 h 函数的定义给出限制，使得 h 满足单调性。对于满足单调性条件的 h，则一定不会出现重复扩展节点问题。二是对 A*算法加以改进，使用修正的 A*算法进行搜索，则可以减少重复扩展节点问题。

第 2 题

2、简述回溯策略与深度优先策略的不同点。

答：回溯搜索策略与深度有限搜索策略最大的不同是深度有限搜索策略属于图搜索，而回溯搜索则不是图搜索。在回溯搜索中，只保留了从初始节点到当前节点的搜索路径。而深度优先搜索，则保留了所有的已经搜索过的路径。

第 3 题

3、某问题由下列公式描述：

(1) $\neg P(a)$

(2) $(\forall x)(\forall y)[\neg P(y) \rightarrow Q(b, x, g(x, y))]$

(3) $(\forall x)[P(f(x))]$

(4) $(\forall x)[P(x) \wedge Q(b, c, x) \rightarrow R(h(x))]$

(5) $(\forall x)(\forall y)[Q(b, x, y) \rightarrow Q(b, x, f(y))]$

(1)、试用归结法证明 $(\exists x)R(x)$ ；

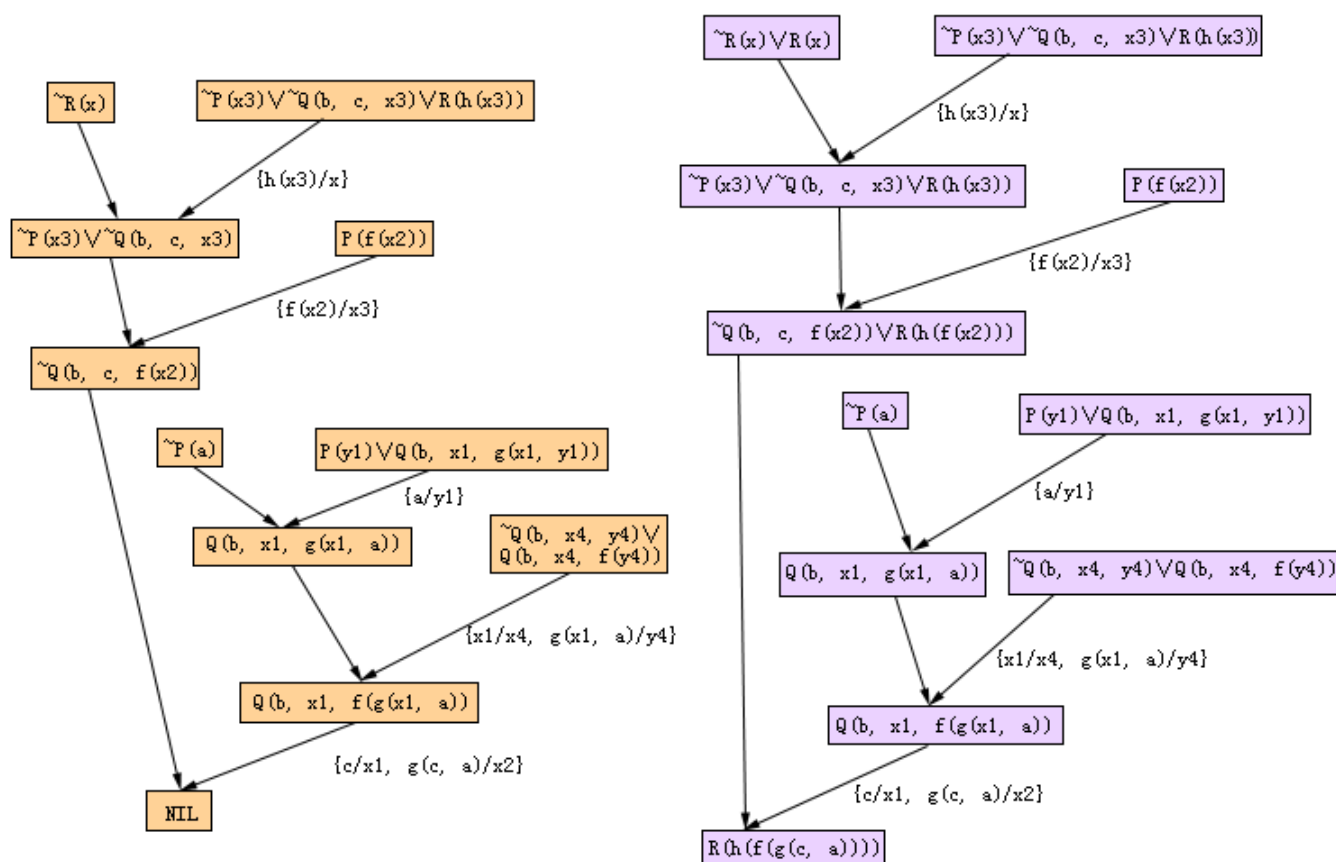
(2)、做出修改证明树，提取问题的回答。

答： 化子句集如下：

(1) $\neg P(a)$
 (2) $(\forall x)(\forall y)[\neg P(y) \rightarrow Q(b, x, g(x, y))]$
 $\Rightarrow (\forall x)(\forall y)[P(y) \vee Q(b, x, g(x, y))]$
 $\Rightarrow P(y) \vee Q(b, x, g(x, y))$
 (3) $(\forall x)[P(f(x))]$
 $\Rightarrow P(f(x))$
 (4) $(\forall x)[P(x) \wedge Q(b, c, x) \rightarrow R(h(x))]$
 $\Rightarrow (\forall x)[\neg(P(x) \wedge Q(b, c, x)) \vee R(h(x))]$
 $\Rightarrow (\forall x)[\neg P(x) \vee \neg Q(b, c, x) \vee R(h(x))]$
 $\Rightarrow \neg P(x) \vee \neg Q(b, c, x) \vee R(h(x))$

(5) $(\forall x)(\forall y)[Q(b, x, y) \rightarrow Q(b, x, f(y))]$
 $\Rightarrow (\forall x)(\forall y)[\neg Q(b, x, y) \vee Q(b, x, f(y))]$
 $\Rightarrow \neg Q(b, x, y) \vee Q(b, x, f(y))$
 目标求反:
 $\neg(\exists x)R(x)$
 $\Rightarrow (\forall x)\neg R(x)$
 $\Rightarrow \neg R(x)$
 变量换名后, 得子句集:
 $\{\neg P(a), P(y_1) \vee Q(b, x_1, g(x_1, y_1)),$
 $P(f(x_2)), \neg P(x_3) \vee \neg Q(b, c, x_3) \vee R(h(x_3)),$
 $\neg Q(b, x_4, y_4) \vee Q(b, x_4, f(y_4)), \neg R(x)\}$

归结树 和 修改证明树 如下:



得到问题的解答: $R(h(f(g(c, a))))$

第 4 题

4、下图所示博弈树, 按从左到右的顺序进行 α - β 剪枝搜索, 试标明各生成节点的到推值, 何处发生剪枝, 及应选择的走步。

r2: IF $\text{len}(N)$ 为偶数 and $n=\text{In}(1, 3, 7, 9)$ THEN $\{x\} \cup \{n\}$

r3: IF $\text{len}(N)$ 为奇数 and $n=\text{In}(2, 4, 6, 8)$ THEN $\{x\} \cup \{n\}$

其中 $\text{len}(N)$ 为求序列的长度， $\text{In}(a, b, c, d)$ 为取 a 、 b 、 c 、 d 之一。

初始状态: $\{\}$

结束条件: 得到的序列 N 前 i 个数组成的整数能被 i 整除。

综合练习 2

一、判断

1、只有在单位耗散值的情况下，当问题有解时，宽度优先算法才能保证找到最优解。（对）

2、在 A*算法结束之前，OPEN 表中任何满足 $f(n) < f^*(s)$ 的节点 n ，一定被扩展。（对）

二、填空题

1、基于规则的正向演绎系统使用的条件是（1）事实表达式是____（2）规则形式为____，其中____（3）目标公式为____

解答：1、基于规则的正向演绎系统使用的条件是（1）事实表达式是 **任意形式**（2）规则形式为 **$L \rightarrow W$ 或 $L_1 \vee L_2 \rightarrow W$** ，其中 **$L$ 为单文字， W 为任意形**（3）目标公式为 **文字析取形**

2、基于规则的逆向演绎系统使用的条件是（1）事实表达式是____（2）规则形式为____，其中____（3）目标公式为____

解答：2、基于规则的逆向演绎系统使用的条件是（1）事实表达式是 **文字合取形**（2）规则形式为 **$W \rightarrow L$ 或 $W \rightarrow L_1 \wedge L_2$** ，其中 **$L$ 为单文字， W 为任意形**（3）目标公式为 **任意形式**

3、归结法中，可以通过_____的方法得到问题的解答。

解答：3、归结法中，可以通过 **修改证明树的** 方法得到问题的解答。

三、问答题

第 1 题

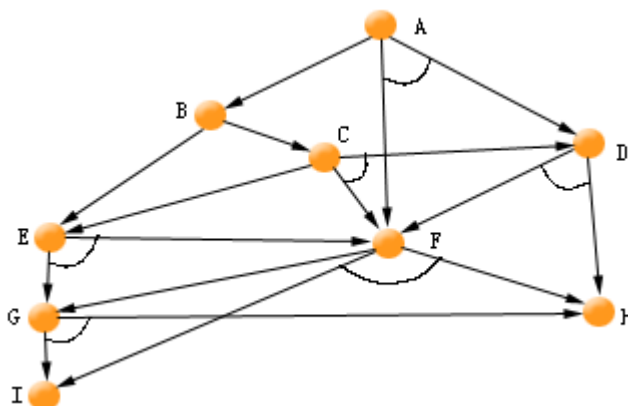
1、某问题状态图如右图所示。假定 k 连接符的耗散值为 k 。

各节点的 h 值假定为：

$h(A)=3, h(B)=2, h(C)=6, h(D)=3, h(E)=4, h(F)=2, h(G)=3,$

$h(H)=h(I)=0$ (目标节点)

用 AO*算法求解该问题，给出每次循环后的搜索图，并给出求得的解图。



答：（见下页）

第3题

3、有四人过河，只有一条船，最多可乘坐两人。若单个过，各需 1，1，5，9 分钟，若两人一起过，则需要的时间以多的为准（如需要 5 分和 9 分的两人同时乘坐，则需要 9 分）。问最少需要多少分钟。

- (1)、用产生式系统描述该问题，要求给出综合数据库的定义，规则集，初始状态和结束状态。
- (2)、定义一个 h 函数，并说明是否满足 A*条件。
- (3)、用 A 算法求解该问题，给出状态搜索图，标出扩展次序、各节点的 f 值、解路径及解路径的耗散值。

答： 综合数据库：

(m1, m5, m9, b) 设从河的左岸到右岸，其中 m1, m5, m9 分别表示过河时间需要 1 分钟，5 分钟和 9 分钟的人，在河左岸的人数。b = 1 表示船在左岸，b = 0 表示船在右岸。 规则集：

```
IF (m1, m5, m9, 1) THEN (m1-1, m5, m9, 0)
IF (m1, m5, m9, 1) THEN (m1-2, m5, m9, 0)
IF (m1, m5, m9, 1) THEN (m1, m5-1, m9, 0)
IF (m1, m5, m9, 1) THEN (m1, m5, m9-1, 0)
IF (m1, m5, m9, 1) THEN (m1-1, m5-1, m9, 0)
IF (m1, m5, m9, 1) THEN (m1-1, m5, m9-1, 0)
IF (m1, m5, m9, 1) THEN (m1, m5-1, m9-1, 0)
IF (m1, m5, m9, 0) THEN (m1+1, m5, m9, 1)
IF (m1, m5, m9, 0) THEN (m1+2, m5, m9, 1)
IF (m1, m5, m9, 0) THEN (m1, m5+1, m9, 1)
IF (m1, m5, m9, 0) THEN (m1, m5, m9+1, 1)
IF (m1, m5, m9, 0) THEN (m1+1, m5+1, m9, 1)
IF (m1, m5, m9, 0) THEN (m1+1, m5, m9+1, 1)
IF (m1, m5, m9, 0) THEN (m1, m5+1, m9+1, 1)
```

初始状态：

(2, 1, 1, 1)

结束状态

(0, 0, 0, 0)

h 函数：

$h(n) = m - b$ ，其中 m 为在左岸的人数，b 为船是否在左岸。

对于任意两个节点 n_i 和 n_j ，其中 n_j 是 n_i 的子节点。

当 n_i 中 $b=1$ 时，则 n_j 中 $b=0$ ，因此： $\max(h(n_i)-h(n_j))=(m-1)-(m-1)=0$ ，而 $C(n_i, n_j)$ 最小为 1，因此 $h(n_i)-h(n_j) < C(n_i, n_j)$ ；

当 n_i 中 $b=0$ 时，则 n_j 中 $b=1$ ，因此： $\max(h(n_i)-h(n_j))=m-m=0$ ，而 $C(n_i, n_j)$ 最小为 1，因此 $h(n_i)-h(n_j) < C(n_i, n_j)$ 。

而对于目标节点 t， $h(t)=0$ 。

因此该 h 函数满足单调性条件。所以 h 满足 A*条件。

第4题

4、某问题由下列公式描述：

- (1)、 $(\exists s) \sim P(s)$
- (2)、 $(\forall s) (P(g(s)))$
- (3)、 $(\exists x) (\forall s) (\exists y) ((P(s) \wedge Q(b, x, s)) \rightarrow H(y))$
- (4)、 $(\forall x) (\forall s) (Q(b, x, s) \rightarrow Q(b, x, g(s)))$
- (5)、 $(\forall x) (\forall s) (\exists y) (\sim P(s) \rightarrow Q(b, x, y))$

求证: $(\exists x)H(x)$ 请用基于规则的逆向演绎系统求解 $(\exists x)H(x)$ 成立。要求给出一个求得的一致解图,并说明为什么它是一致的;给出目标的解答。

答: 对事实和规则进行 skolem 化:

- (1) $(\exists s) \sim P(a)$
- (2) $(\forall s)(P(g(s)) \rightarrow P(s))$
- (3) $(\exists x)(\forall s)(\exists y)((P(s) \wedge Q(b, x, s)) \rightarrow H(y))$
 $(P(s) \wedge Q(b, c, s)) \rightarrow H(f(s))$
- (4) $(\forall x)(\forall s)(Q(b, x, s) \rightarrow Q(b, x, g(s)))$
 $Q(b, x, s) \rightarrow Q(b, x, g(s))$
- (5) $(\forall x)(\forall s)(\exists y)(\sim P(s) \rightarrow Q(b, x, y))$
 $\sim P(s) \rightarrow Q(b, x, h(x, s))$

经变量换名后, 有事实和规则如下:

$\sim P(a)$

$P(g(s1))$

r1: $(P(s2) \wedge Q(b, c, s2)) \rightarrow H(f(s2))$

r2: $Q(b, x3, s3) \rightarrow Q(b, x3, g(s3))$

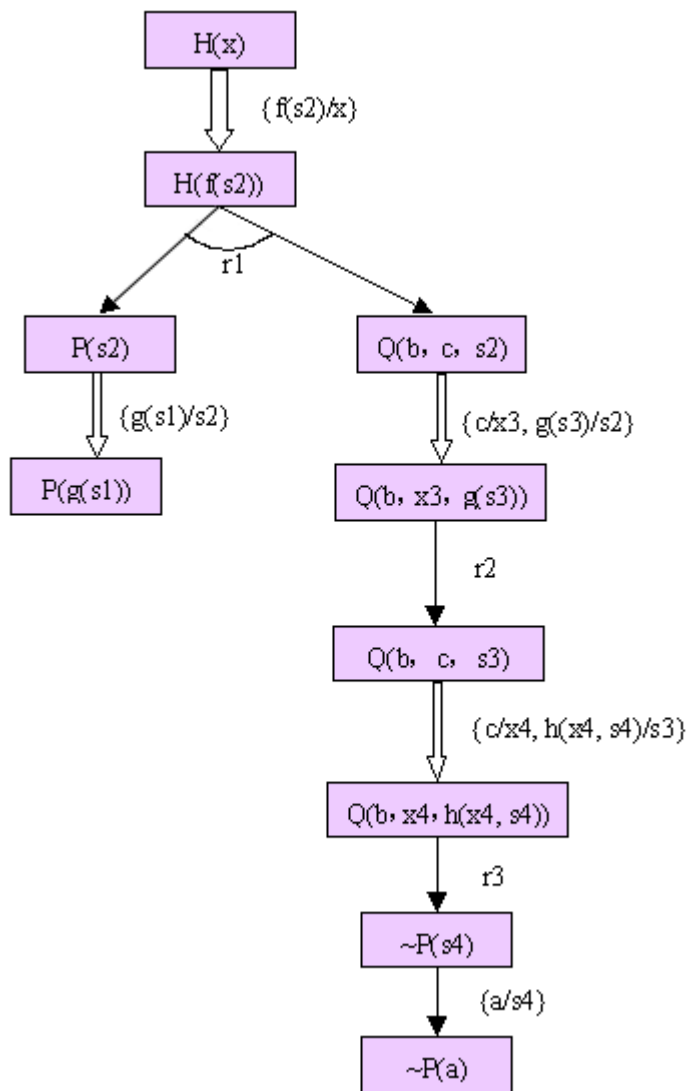
r3: $\sim P(s4) \rightarrow Q(b, x4, h(x4, s4))$

用对偶形式对目标 skolem 化:

$(\exists x)H(x)$

$H(x)$

演绎图如右图(这里只给出了一个一致解图)。



由置换集构造 U1 和 U2:

$U1 = (x, s2, x3, s2, x4, s3, s4)$

$U2 = (f(s2), g(s1), c, g(s3), c, h(x4, s4), a)$

由于 U1 和 U2 是可合一的, 因此该解图是一致解图。合一复合为:

$\{f(g(h(c, a)))/x, g(h(c, a))/s2, c/x3, h(c, a)/s3, c/x4, h(c, a)/s1, a/s4\}$ 将该合一复合带入目标中, 得到解答:

$x = f(g(h(c, a)))$

第 5 题

5、简述为什么修正的 A*算法有可能会减少重复节点的扩展, 而又不会比 A*多扩展节点。

答:

1、由于修正的 A*算法对与 OPEN 表中 f 值小于 f_m 的节点按照 g 值的大小排序, 优先扩展 g 值小的节点。对于这部分节点来说, 相当于 $h=0$ 。而 $h=0$ 满足单调条件, 因此至少对于这部分节点来说, 减少了重复扩展节点的可能性。因此修正的 A*算法有可能减少重复节点的扩展。

2、由于 A*扩展的节点, 其 f 值总是小于等于 $f^*(s)$, 而 f_m 是到目标为止扩展过的节点的最大 f 值, 因此必有 f_m 小于等于 $f^*(s)$ 。又由于 OPEN 表中 f 值小于 $f^*(s)$ 的节点, 总是被 A*扩展的, 因此从 OPEN 表中选择那些 f 值小于 f_m 的节点 (这些节点的 f 值也必然小于 $f^*(s)$), 按照 g 值的大小优先扩展, 只是改变了部分节点的扩展次序, 不会多扩展节点。