人工智能导论第一次作业

3.7 给出下列问题的初始状态、目标测试、后继函数和耗散函数

a. 只用四种颜色对平面地图染色，要求每两个相邻的地区不能染成相同的颜色

状态空间：所有染色情况集合

1. 初始状态：地图所有格子未染色
2. 目标测试：地图所有格子已经染色，且每两个相邻地区的颜色不同
3. 后继函数：选择一个地区染色
4. 耗散函数：染色次数

b.一间屋子里有一只3英尺高的猴子，屋子的房顶上挂着一串香蕉，离地面8公尺。屋子里有两个可叠放起来、可移动、可攀登的3英尺高的箱子。猴子很想得到香蕉。

状态空间：房间内猴子、箱子的位置以及猴子的状态

1. 初始状态：如题目描述
2. 目标测试：猴子得到香蕉
3. 后继函数：一系列动作，包括猴子攀爬上/下某个箱子，猴子搬运某个箱子到指定位置，猴子拿香蕉
4. 耗散函数：猴子采取的动作次数

c.有一个程序，当送入一个特定文件的输入记录时会输出“不合法的输入记录”。已知每个记录的处理独立于其它记录。要求找出哪个记录不合法。

状态空间：待测试的记录集合

1. 初始状态：所有的记录集合
2. 目标测试：记录集合中仅有一个记录
3. 后继函数：将记录集合分成最多数量相差1的两半，分别用程序测试，输出“不合法”的那一半作为新的状态
4. 耗散函数：使用程序的次数

d.有三个水壶，容量分别为12加仑、8加仑和3加仑，还有一个水龙头。可以把壶装满或者倒空，从一个壶倒进另一个壶或者倒在地上。要求量出刚好1加仑水。

状态空间：三元组，分别表征三个水壶当前的水量

1. 初始状态：三个水壶水量为0
2. 目标测试：三个水壶有一个水量为1
3. 后继函数：一些可选操作，将某壶装到满或者倒空，或者将A壶向B壶倒水直到B壶满或者A壶为空

4. 耗散函数：倒水次数

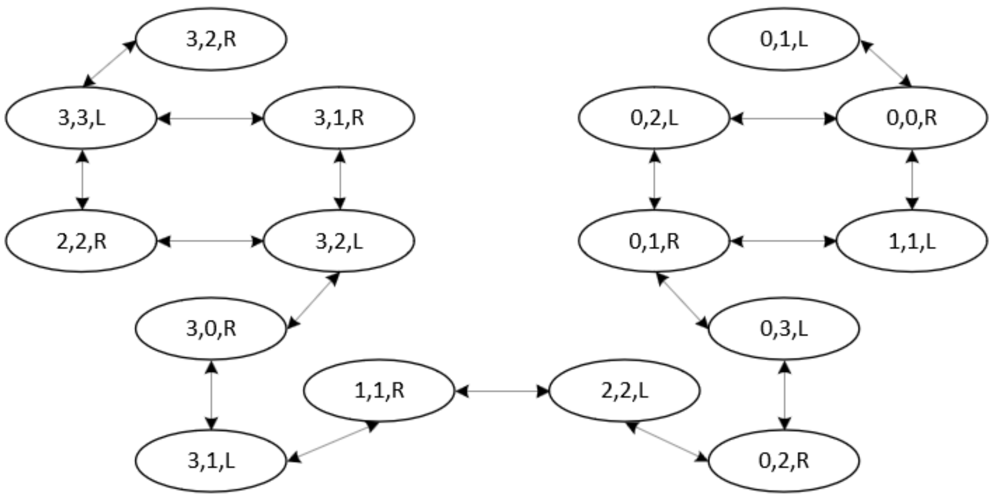
3.9 传教士与野人问题

传教士和野人问题通常描述如下：三个传教士和三个野人在河的一边，还有一条能载一个人或者两个人的船。找到一个办法让所有的人都渡到河的另一岸，要求在任何地方野人数都不能多于传教士的人数(可以只有野人没有传教士)。这个问题在AI领域中很著名,因为它是第一篇从分析的观点探讨问题形式化的论文的主题（Amarel, 1968）

a.精确地形式化该问题，只描述确保该问题有解所必需的特性。画出该问题的完全状态空间图。

状态空间：三元组，河左岸的传教士和野人人数，船在哪一边，并且满足任何地方野人数不多于传教士人数或者只有野人，船在一个可能的位置

1. 初始状态：河左岸3个传教士、3个野人，船在左岸
2. 目标测试：河左岸0个传教士、0个野人
3. 后继函数：选定船所在一侧的传教士/野人上/下船，船摆渡到另一岸
4. 耗散函数：船摆渡次数



状态空间如图，其中用双向箭头表示同时存在状态 i 指向状态 j 和状态 j 指向状态 i 的边

b.用一个合适的搜索算法实现和最优地求解该问题。检查重复状态是个好主意吗?

直接使用广度优先搜索，记录查找路径，第一个搜索到的目标状态的路径就对应着一个最少的摆渡次数的方案。由于状态空间小，因此事实上各种检测重复结点的最短路径搜索算法都可以。

有必要，可以防止在重复状态间陷入死循环。

c.这个问题的状态空间如此简单，你认为为什么人们求解它却很困难?

尽管状态空间简单，困难在于需要具体地去实现检查重复状态、遍历某一个状态下所有合法的操作及目标状态、回溯。

人工智能导论第二次作业

4.1 跟踪A*搜索算法用直线距离启发式求解从Lugoj到Bucharest问题的过程。按顺序列出算法扩展的节点和每个节点的f,g,h值。

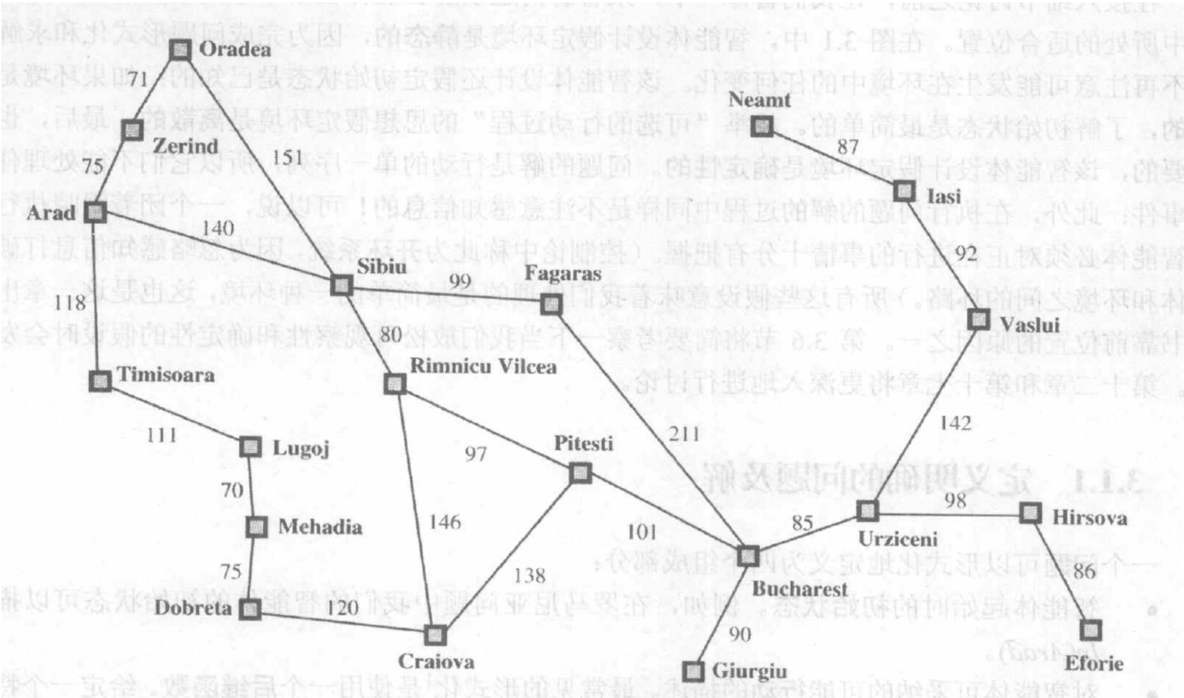


图 3.2 一个简化的部分罗马尼亚道路图

Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Dobreta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

图 4.1 h_{SLD} 的值——到 Bucharest 的直线距离

解：两点之间直线最短，故对于任意节点 a, b 和目标节点 c， $h(a,c)$ 作为 a 与 c 之间的直线距离必然 $\le a$ 与 b 之间的距离 + b 与 c 之间的直线距离，故可知其满足一致性，可以使用图算法拓展的A*搜索算法。此处用 $name(f,g,h)$ 表示每个节点及其 f,g,h 值

1. Lugoj(244,0,244)
2. Mehadia(311,70,241) , Timisoara(440,111,329)
3. Dobreta(387,145,242)
4. Craiova(425,265,160)
5. Pitesti(503,403,100) , Rimnicu Vilcea(604,411,193)

6. Arad(595,229,366)
7. Bucharest(504,504,0)

4.2 启发式路径算法是一个最佳优先搜索，它的目标函数是 $f(n)=(2-\omega)g(n)+\omega h(n)$ 。算法中 ω 取什么值能保证算法是最优的?当 $\omega=0$ 时，这个算法是什么搜索? $\omega=1$ 呢? $\omega=2$ 呢?

解：考虑

$$f(n)=(2-\omega)g(n)+\omega h(n)=(2-\omega)(g(n)+\frac{\omega}{2-\omega}h(n))$$

仅考虑$(2-\omega)$系数后面的部分，令$h'(n)=\frac{\omega}{2-\omega}h(n)$，则在$h(n)$作为一个单独函数本身满足其可采纳的情况下，若

$$h'(n)=\frac{\omega}{2-\omega}h(n)\leq h(n)$$

即 $\omega\leq 1$ ，那么

$$h'(n)\leq h(n)\leq h^*(n)$$

所以这种情况下 $h'(n)$ 是可采纳的，即可以保证算法是最优的

由定义得

当 $\omega=0$ 时，这个算法是一致代价搜索

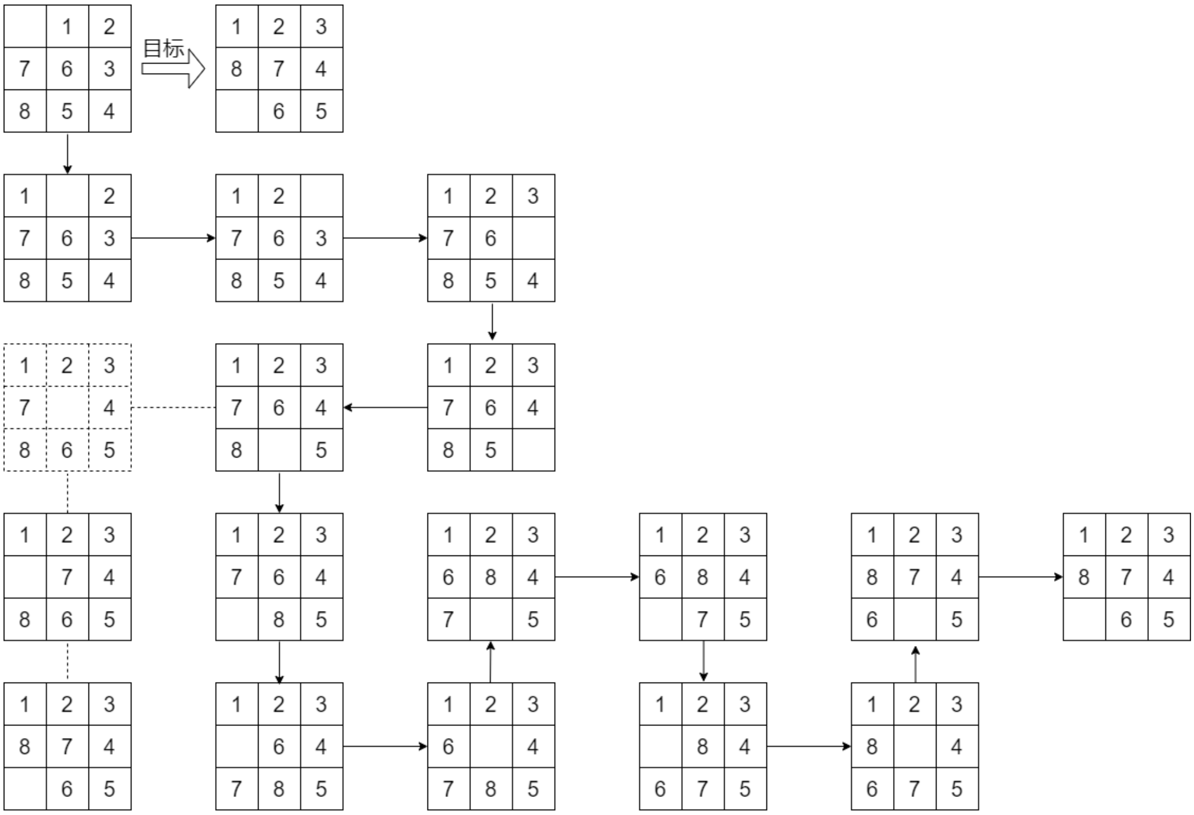
当 $\omega=1$ 时，这个算法是A*搜索

当 $\omega=2$ 时，这个算法是贪心搜索

4.6 设计一个启发函数，使它在八数码游戏中有时会估计过高，并说明它在什么样的特殊问题下会导致非最优解。(可以借助计算机的帮助。)证明：如果 h 被高估的部分从来不超过 c ， A^* 算法返回的解的耗散比最优解的耗散多出的部分也不超过 c 。

解：设计启发函数如下——

$h(n)$是如此一个函数，当它处在某个特定状态时，其值极大（显然可以做到足够大使得其远超出实际的耗散，并且下一步的拓展必然不选择该状态），其余状态均定义为不在位的棋子数，于是由图可见一特殊问题例子



下面对本图做一些说明，每个状态由一个九宫格标记，其中虚线九宫格所对应的就是该 $h(n)$ 定义中被赋予足够大值的”特殊状态“。其中”目标“箭头由初始状态指向目标状态，而实箭头表示按照上述 $h(n)$ 定义下所搜索到的解，在箭头的分支处，虚线箭头表示这一步之后本应当是最优解的各步骤。显然这个启发函数在箭头分支处导致了估计过高，并且最后找出来的解也不是最优解。

下面证明命题。

证明：对于任意的节点 s_0 与目标 s ，考虑其基于此处的启发式函数的扩展序列 $s_1, s_2, \ldots, s_n, s$,与此同时另有一条最优路径的序列 $s'_1, s'_2, \ldots, s'_m, s$ ，设 $s_i = s'_j, s_{i+1} \neq s'_{j+1}$，且 s_{i+1} 并不在基于启发式函数求出的解的路径上。那么有

$$g(s) = f(s) \leq f(s_{i+1}) \leq f(s'_{i+1}) \leq g(s'_{i+1}) + h^*(s_{i+1}) + c = h^*(s_0) + c$$

证毕

4.7 证明如果一个启发式是一致的，它肯定是可采纳的。构造一个非一致的可采纳启发式。

证明：首先，如果启发式 $h(n)$ 是一致的，即对于任何状态 i, j 以及目标状态 s ，设 j 为 i 的后继，则 $h(i) + c(i, j) \geq h(j)$ 。

那么对于任一节点 s_0 ，考虑其到目标状态 s 最短路径上的一系列节点 $s_1, s_2, \ldots, s_n$ ，那么有

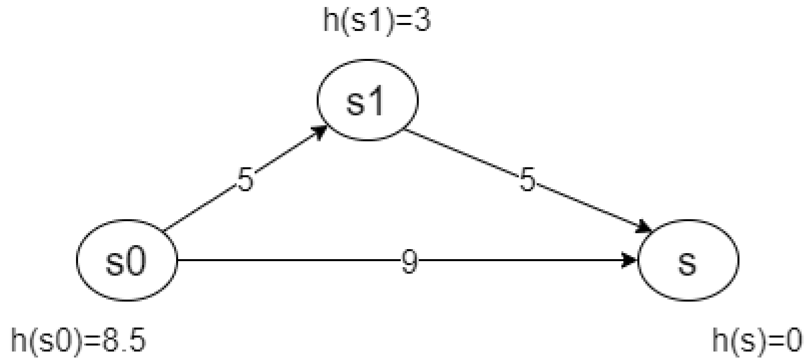
$$h^*(s_0) = c(s_0, s_1) + c(s_1 + s_2) + \cdots + c(s_n, s) \geq (h(s_n) - h(s)) + \cdots + (h(s_1) - h(s_2)) + (h(s_0) - h(s_1)) = h(s_0) - h(s) = h(s_0)$$

故得到结论

$$h(s_0) \leq h^*(s_0)$$

由于节点 s_0 的选取是任意的，得证其满足可采纳性

构造：如图



首先易知 $h(s0)=8.5 \le h^*(s0)=9, h(s1)=3 \le h^*(s1)=5, h(s)=0$ ，满足一致性要求，但是 $h(s0)=8.5 > c(s0,s1)+h(s1)=5+3=8$ 。并不具备一致性。

人工智能第三次作业

6.5 Solve the cryptarithmetic problem in Figure 6.2 by hand, using the strategy of backtracking with forward checking and the MRV and least-constraining-value heuristics.

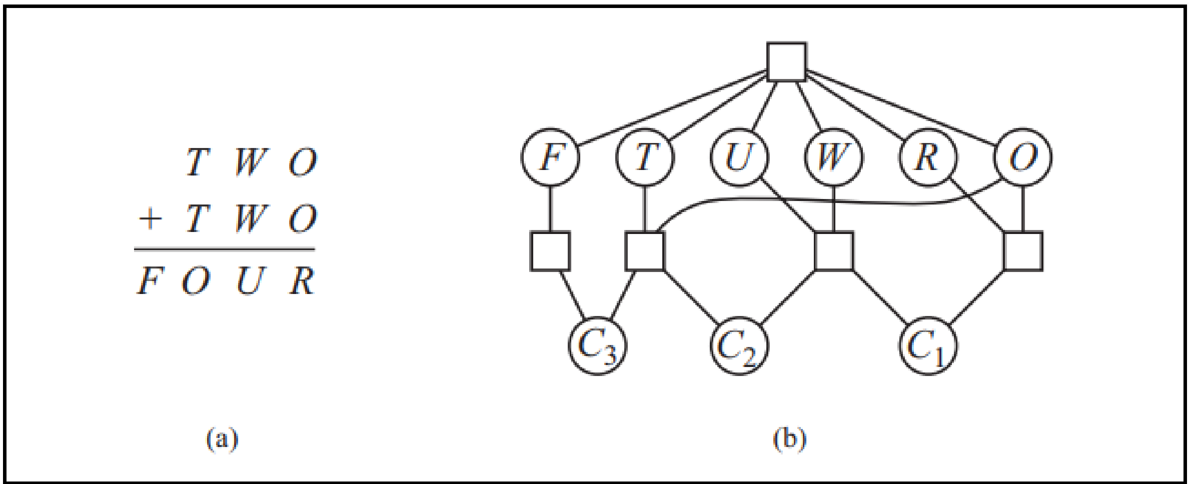


Figure 6.2 (a) A cryptarithmetic problem. Each letter stands for a distinct digit; the aim is to find a substitution of digits for letters such that the resulting sum is arithmetically correct, with the added restriction that no leading zeroes are allowed. (b) The constraint hypergraph for the cryptarithmetic problem, showing the *Alldiff* constraint (square box at the top) as well as the column addition constraints (four square boxes in the middle). The variables C_1, C_2, and C_3 represent the carry digits for the three columns.

解：从上往下，从左往右，每个多元约束分别为

- F、T、U、W、R、O互不相同
- F = C_3
- $C_3 \times$ 10 + O = 2 $\times$ T + C_2
- $C_2 \times$ 10 + U = 2 $\times$ W + C_1
- $C_1 \times$ 10 + R = 2 $\times$ O

由单元约束，T、U、W、R、O均为0-9之间的数字，F和C_1, C_2, C_3取值均为0或1。

罗列算法模拟执行步骤如下

- 选定T为初始变量，因为它是度数最高的节点之一
- T尝试赋值1,在当前信息下这是可能的给邻居变量留下最多选择的赋值之一
- 前向检验，F、U、W、R以及 C_3 删去1，O的取值范围删至 $\{2, 3\}$ 。
- 由MRV，选定F为下一个变量，因为它是取值范围最小的变量之一
- F尝试赋值0，因为其仅有一个赋值可能
- 前向检验，U,W,R删去0
- 由MRV，选定 C_3 为下一个变量，因为它是取值范围最小的变量之一
- C_3 尝试赋值0，因为其仅有一个赋值可能
- 前向检验，无变动
- 由MRV，选定O为下一个变量，因为它是取值范围最小的变量之一
- O尝试赋值3，在当前信息下这是可能的给邻居变量留下最多选择的赋值之一

- 前向检验，R的取值范围删至 $\{6\}$ ，C_1 的取值范围删至 $\{0\}$，C_2 的取值范围删至 $\{1\}$，U、W删去3
- 由MRV，选定R为下一个变量，因为它是取值范围最小的变量之一
- R尝试赋值6，因为其仅有一个赋值可能
- 前向检验，U、W删去6
- 由MRV，选定 C_1 为下一个变量，因为它是取值范围最小的变量之一
- C_1 尝试赋值0，因为其仅有一个赋值可能
- 前向检验，U的取值范围删至 $\{4,8\}$，W的取值范围删至 $\{7,9\}$ 。
- 由MRV，选定 C_2 为下一个变量，因为它是取值范围最小的变量之一
- C_2 尝试赋值1，因为其仅有一个赋值可能
- 前向检验，无变动
- 由MRV，选定U为下一个变量，因为它是取值范围最小的变量之一
- U尝试赋值4，在当前信息下这是可能的给邻居变量留下最多选择的赋值之一
- 前向检验，W删去9
- 由MRV，选定W为下一个变量，因为它是取值范围最小的变量之一
- W尝试赋值7，因为其仅有一个赋值可能
- 前向检验，无变动
- 完成，得到一个赋值方案

6.11 Use the AC-3 algorithm to show that arc consistency can detect the inconsistency of the partial assignment WA=green,V=red for the problem shown in Figure 6.1

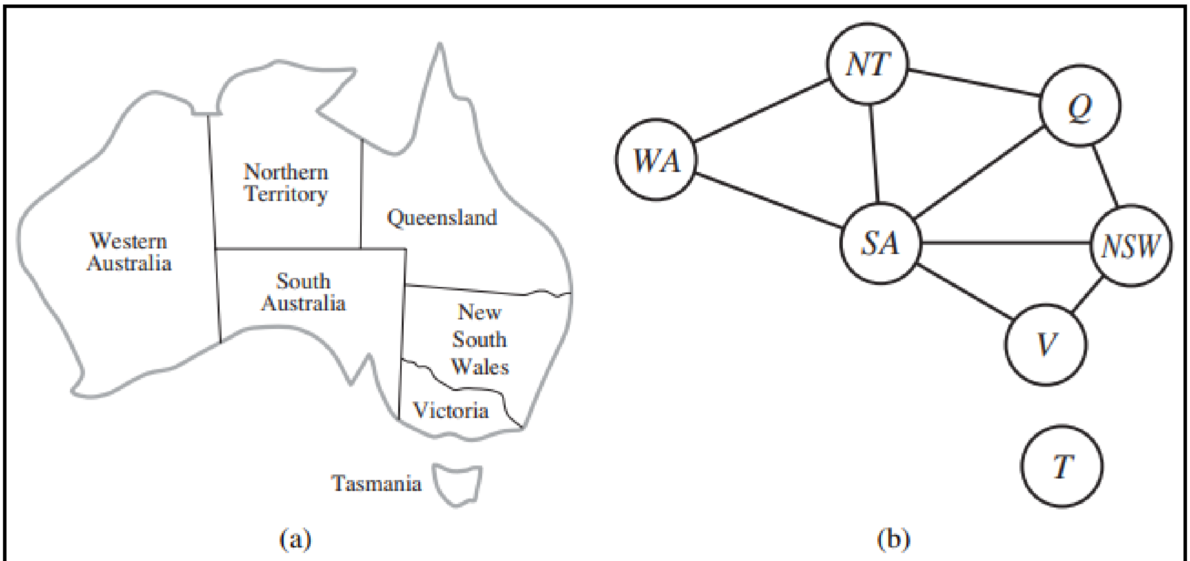


Figure 6.1 (a) The principal states and territories of Australia. Coloring this map can be viewed as a constraint satisfaction problem (CSP). The goal is to assign colors to each region so that no neighboring regions have the same color. (b) The map-coloring problem represented as a constraint graph.

解：罗列算法模拟执行步骤如下

- 队列中放入所有边（二元约束条件），除WA,V外，所有节点的可能属性值集合为 $\{red, yellow, green\}$ 。
- <SA，WA> 出队列，从SA中去掉 $green$ ，将<NT,SA>、<Q,SA>、<NSW,SA>、<V,SA>、<WA,SA> 加入队列（如果队列中没有的话）

- <SA，V> 出队列，从SA中去掉 red ，将<NT,SA>、<Q,SA>、<NSW,SA>、<V,SA>、<WA,SA>加入队列（如果队列中没有的话）
- <NT，WA> 出队列，从NT中去掉 $green$ ，将<WA,NT>、<SA,NT>、<Q,NT>加入队列（如果队列中没有的话）
- <NT，SA> 出队列，从NT中去掉 $yellow$ ，将<WA,NT>、<SA,NT>、<Q,NT>加入队列（如果队列中没有的话）
- <NSW，V> 出队列，从NSW中去掉 red ，将<V,NSW>、<SA,NSW>、<Q,NSW>加入队列（如果队列中没有的话）
- <NSW，SA> 出队列，从NSW中去掉 $yellow$ ，将<V,NSW>、<SA,NSW>、<Q,NSW>加入队列（如果队列中没有的话）
- <Q,NT>出队列，从Q中去掉 red ，将<NT,Q>、<SA,Q>、<NSW,Q>加入队列（如果队列中没有的话）
- <Q,SA>出队列，从Q中去掉 $yellow$ ，将<NT,Q>、<SA,Q>、<NSW,Q>加入队列（如果队列中没有的话）
- <Q,NSW>出队列，从Q中去掉 $green$ ，将<NT,Q>、<SA,Q>、<NSW,Q>加入队列（如果队列中没有的话）
- 此时Q的可能属性值集合已经为空集，返回错误

6.12 What is the worst-case complexity of running AC-3 on a tree-structured CSP?

解：假设有 n 个节点，那么由于其为树结构，所以有 $n-1$ 条边（二元约束条件），假设每个节点至多有 k 个取值可能，不妨将边加入队列的次数和删减取值的次数分开统计，那么在执行AC-3算法的过程中，每一条边（二元约束条件）上至多执行 k 次删减取值的操作，同时也至多 k 次加入队列， 所以在不考虑revise检查相容性的开销的情况下，总时间复杂度为 $O(nk)$ 。

但是如果考虑revise检查相容性的开销，每次相当于O(两个结点的取值可能数的乘积)。那么总时间复杂度直接为 $O(nk^3)$ 。

人工智能第四次作业

5.9 This problem exercises the basic concepts of game playing, using tic-tac-toe (noughts and crosses) as an example.

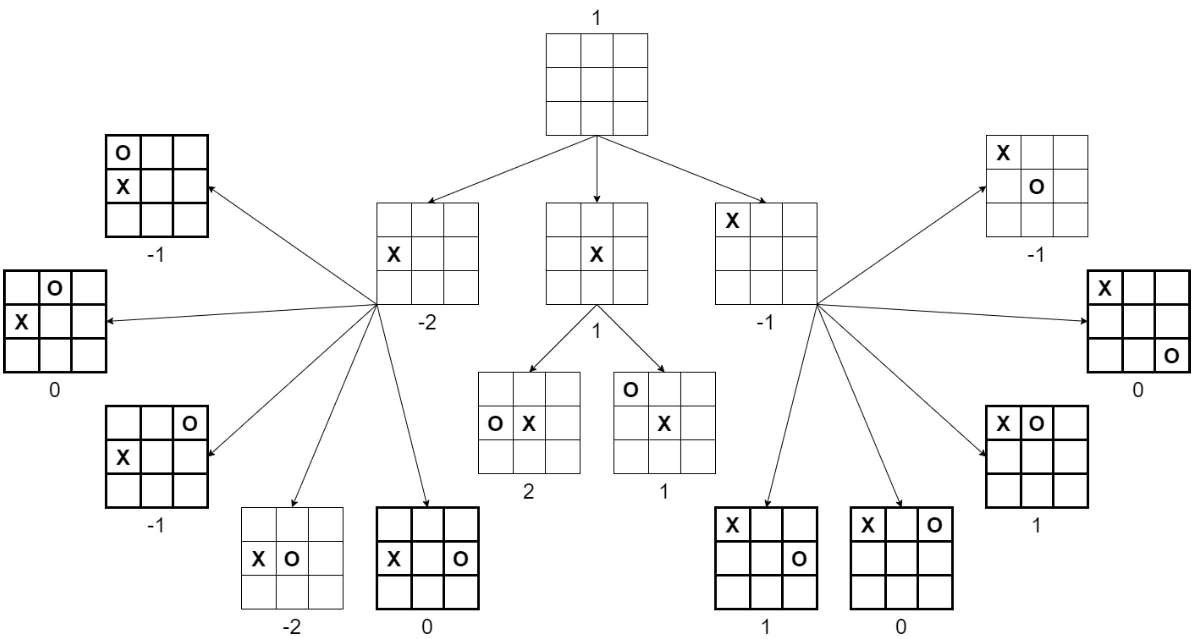
We define X_n as the number of rows, columns, or diagonals with exactly $nX's$ and no $O's$. Similarly, On is the number of rows, columns, or diagonals with just $nO's$. The utility function assigns$+1$ to any position with $X_3 = 1$ and -1 to any position with $O_3 = 1$. All other terminal positions have utility 0. For nonterminal positions, we use a linear evaluation function defined as $Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$.

a. Approximately how many possible games of tic-tac-toe are there?

解：考虑双方每轮落子，最多九轮，那么可能的棋局估计上限值为$9!$

b. Show the whole game tree starting from an empty board down to depth 2 (i.e., one X and one O on the board), taking symmetry into account.

解：绘制如图，本大题c、d、e三小题也均按照此图解答。



c. Mark on your tree the evaluations of all the positions at depth 2.

解：如b小题图，已经将评估函数值标记在所有对应着深度为2的棋局的结点的下方。

d. Using the minimax algorithm, mark on your tree the backed-up values for the positions at depths 1 and 0, and use those values to choose the best starting move.

解：如b小题图，已经将根据极小极大算法算出的倒推值标记在所有对应着深度为1或0的棋局的结点的附近。

所以最好的起始行棋是把X下在最中间。

e. Circle the nodes at depth 2 that would not be evaluated if alpha–beta pruning were applied, assuming the nodes are generated in the optimal order for alpha–beta pruning.

解：如b小题图，为了呈现方便，所有被剪掉的结点边框都采用了加粗显示（亦可看做是一种“circle”）。

假设先探索完了X先打在最中间的结点及其子树，随后另外两种X初次位置可能的两个结点所有的子节点里，都先探索完实际minmax值最小的结点，这样一来根据 $\alpha - \beta$ 剪枝，其它的子节点就没有被探索的需要。

5.8 Consider the two-player game described in Figure 5.17.

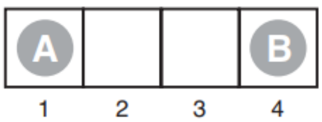
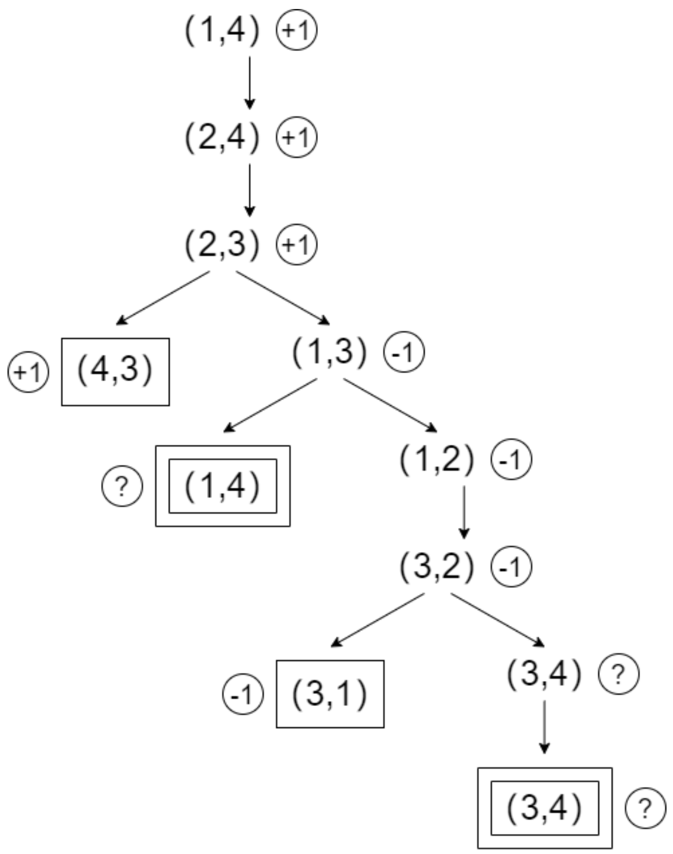


Figure 5.17 The starting position of a simple game. Player A moves first. The two players take turns moving, and each player must move his token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if A is on 3 and B is on 2, then A may move back to 1.) The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game to A is $+1$; if player B reaches space 1 first, then the value of the game to A is -1.

a. Draw the complete game tree, using the following conventions:

- Write each state as (s_A, s_B), where s_A and s_B denote the token locations.
- Put each terminal state in a square box and write its game value in a circle.
- Put *loop states* (states that already appear on the path to the root) in double square boxes. Since their value is unclear, annotate each with a “?” in a circle.

解：绘制图如下



b. Now mark each node with its backed-up minimax value (also in a circle). Explain how you handled the “?” values and why.

解：极大极小值如a小题图。

“?”做如下处理：

- 循环状态设置值为“?”
- 子结点中有“?”状态的情况，优先排除所有“?”值，对于剩下的按照原先的极大极小值算法进行倒推，若有的话则选择相应的最大/最小值对应结点的行动，如果排除“?”值后就没有值了，那么该结点倒推值为“?”，选择一个对应“?”的行动

原因：当一个结点面临两种选择，一者是选择眼下可预见的"更佳"的行动，一者是直接进入未知的循环状态。此处假定了agent总是选择“更佳”的行动，这也有助于防止进入或者尽早跳出循环，并减少了“?”状态的传播。

c. Explain why the standard minimax algorithm would fail on this game tree and briefly sketch how you might fix it, drawing on your answer to (b). Does your modified algorithm give optimal decisions for all games with loops?

解：在循环状态处会产生无限递归循环导致算法无法终止。

方法正如a小题图所示，修改后的算法维护每次的递归搜索栈，一旦发现新搜索到的结点和栈中已有的结点重复，那就说明这个新结点是循环结点，这时候对其终止递归调用，直接返回“?”。同时在极大极小值的计算中加入b小题所述的针对"?"的扩展。

并不一定对所有带循环的游戏都能给出最优选择。假如一个游戏中主动进入循环的一方每进行一轮循环都将得到更高的收益，显然此处应当总是选择进入未知的循环状态而不是选择眼下可预见的“更佳”的行动。

d. This 4-square game can be generalized to n squares for any $n > 2$. Prove that A wins if n is even and loses if n is odd.

解：当n为偶数，A先采取一直向右移动，不论B如何移动，A与B必然相遇，而且此时由于格子数为偶数，相遇之后必然是A先开始行动。令A跳过B前往下一格，B要么继续往左移动，如此A必然先B一步到达右端，B也可以往回跳过A，但如此一来又回到了A与B相遇且A先开始行动的情况，而且A相对于之前向右行进了2格，因此在有限的格子内这个情况并不会反复出现。采用如此策略后，必将终止于两种情况，其一是A直接一直向右移动到最右端，其二是B位于最右端，而A处于B的左边，而下一步是A行动，此时A只需向左走一格，那么B不得不向左走一格，之后A向右跳至最右端即可。

当n为奇数，情况与前者相反，A与B必然相遇，但此时相遇后必然是B先开始行动，后续同理可知B有必胜策略。

5.13 Develop a formal proof of correctness for alpha–beta pruning.

To do this, consider the situation shown in **Figure 5.18**. The question is whether to prune node n_j, which is a maxnode and a descendant of node n_1. The basic idea is to prune it if and only if the minimax value of n_1 can be shown to be independent of the value of n_j.

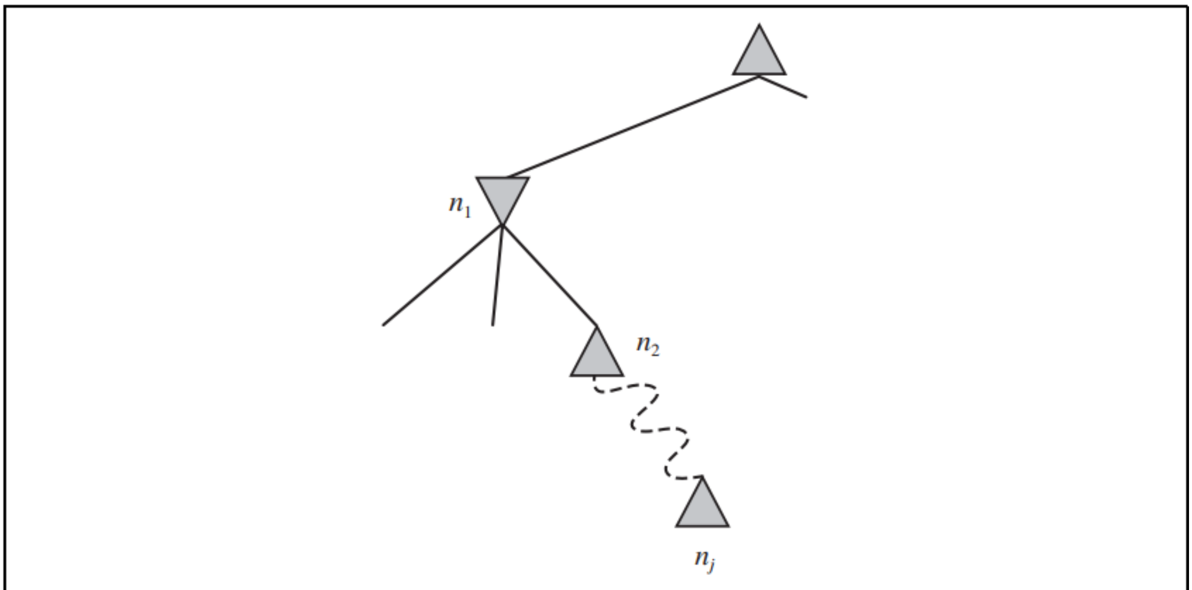


Figure 5.18 Situation when considering whether to prune node n_j.

a. Mode n_1 takes on the minimum value among its children: $n_1 = min(n_2, n_{21}, \ldots, n_{2b_2})$. Find a similar expression for n_2 and hence an expression for n_1 in terms of n_j.

解：设$n_2 = max(n_3, n_{31}, \ldots, n_{3b_3})$，于是 $n_1 = min(max(n_3, n_{31}, \ldots, n_{3b_3}), n_{21}, \ldots n_{2b_2})$，如此递归展开下去直到其中出现$n_j$ 。

b. Let l_i be the minimum (or maximum) value of the nodes to the left of node n_i at depth i, whose minimax value is already known. Similarly, let r_i be the minimum (or maximum) value of the unexplored nodes to the right of n_i at depth i. Rewrite your expression for n_1 in terms of the l_i and r_i values.

解：$n_1 = min(max(n_3, l_3, r_3), l_2, r_2)$，如此递归展开下去直到出现 $min(n_j, l_j, r_j)$。

c. Now reformulate the expression to show that in order to affect n_1, n_j must not exceed a certain bound derived from the l_i values.

解：若n_1的取值需要收到 n_j 的影响，那么首先 n_2 的估计值不应当被判定为$\geq l_2$，即 $max(n_3, l_3, r_3)$ 的估计值不应被判定为$\geq l_2$，且由于 n_3 也应当对结果产生影响，n_3的估计值不应被判定为 $\geq l_2$并且 n_3 不应被判定为 $\leq l_3$。如此递归下去得到 n_j 的值不应当超过之前 $min\{l_i | i = 2, 4, \ldots, j\}$ 的值。

d. Repeat the process for the case where n_j is a min-node.

解：同上，首先$n_1 = min(max(n_3, n_{31}, \ldots, n_{3b_3}), n_{21}, \ldots n_{2b_2})$，如此递归展开下去直到出现 n_j。于是有 $n_1 = min(max(n_3, l_3, r_3), l_2, r_2)$，如此递归展开下去直到出现 $max(n_j, l_j, r_j)$。最后得到 n_j 的值不应当小于 $max\{l_i | i = 3, 5, \ldots, j\}$。

人工智能第五次作业

7.13 :This exercise looks into the relationship between clauses and implication sentences.

Show that the clause $(\neg P_1 \vee \cdots \vee \neg P_m \vee Q)$ is logically equivalent to the implication sentence. $(P_1 \wedge \cdots \wedge P_m) \Rightarrow Q$

解：采用等价替换原则推理如下

$$\begin{aligned}(\neg P_1 \vee \cdots \vee \neg P_m \vee Q) &\Leftrightarrow (\neg P_1 \vee \cdots \vee \neg P_m) \vee Q \\ &\Leftrightarrow \neg(P_1 \wedge \cdots \wedge P_m) \vee Q \\ &\Leftrightarrow (P_1 \wedge \cdots \wedge P_m) \Rightarrow Q\end{aligned}$$

Show that every clause (regardless of the number of positive literals) can be written in the form $(P_1 \wedge \cdots \wedge P_m) \Rightarrow (Q_q \vee \cdots \vee Q_n)$,where the P_s and Q_s are proposition symbols. A knowledge base consisting of such sentences is in implicative normal form or Kowalski from(K0walski,1979).

解：采用等价替换原则，可按如下方式对任意给定的子句$(\neg P_1 \vee \cdots \vee \neg P_m \vee Q_q \vee \cdots \vee Q_n)$进行转化

$$\begin{aligned}(\neg P_1 \vee \cdots \vee \neg P_m \vee Q_q \vee \cdots \vee Q_n) &\Leftrightarrow (\neg P_1 \vee \cdots \vee \neg P_m) \vee (Q_q \vee \cdots \vee Q_n) \\ &\Leftrightarrow \neg(P_1 \wedge \cdots \wedge P_m) \vee (Q_q \vee \cdots \vee Q_n) \\ &\Leftrightarrow (P_1 \wedge \cdots \wedge P_m) \Rightarrow (Q_q \vee \cdots \vee Q_n)\end{aligned}$$

Write down the full resolution rule for sentences in implicative normal form.

每一个蕴含范式可看作一个析取式，那么，用于析取式的完整归结原则转换到蕴含范式中则如下：
如果$P_i = S_j$

$$\frac{\begin{array}{l}(P_1 \wedge \cdots \wedge P_m) \Rightarrow (Q_q \vee \cdots \vee Q_n) \\ (R_q \wedge \cdots \wedge R_k) \Rightarrow (S_s \vee \cdots \vee S_t)\end{array}}{(P_1 \wedge \cdots \wedge P_{i-1} \wedge P_{i+1} \wedge \cdots \wedge P_m \wedge R_q \wedge \cdots \wedge R_k) \Rightarrow (Q_q \vee \cdots \vee Q_n \vee S_s \vee \cdots \vee S_{j-1} \vee S_{j+1} \vee \cdots \vee S_t)}$$

Proof :Prove the completeness of the forward chaining algorithm.

证明：易证前向链接算法中，新加入的结论总是可靠的，而且由于子句和符号是有限的，那么前向链接算法必然在有限步内终结。下证明任何一个可靠的符合KB的单个命题词q总能被算法找到。

因为KB中的知识并不矛盾，所以假如q能由KB推出，那么q这个符号必然在KB的知识中出现过。同时，KB能推出q，等价于KB和$\neg q$不可满足。

假设q仅仅出现在子句的前提部分，将KB看作一系列析取式的集合，那么q将仅仅出现在包含$\neg q$的析取式里，显然$\neg q$是可满足的，矛盾。

所以q也会出现在子句的结论部分。考虑所有这样的子句，必然存在至少一条子句，其所有的前提都是可以被KB推出的，不然$\neg q$就可满足了。考虑这些子句的所有前提，要么是事实，要么出现在其它至少一条子句的结论部分，以此类推追溯，如果q确实能够被KB推出，那么可以排除在几条子句间反复循环追溯的情况，于是必然存在一棵以q为根的树，其每个非叶子结点都和其所有子节点一同表示一条子句的结论和前提，并且其叶子结点都是事实。

那么容易论证，按照前向链接算法，总会沿着这棵树反向推出其根结点q。

如果q不是KB能推出来的，那么由于算法的可靠性，它并不能被加入，而由于算法必然有限步内终结，所以总能输出q不符合。

综上，算法是完备的。

人工智能第六次作业

8.24 Represent the following sentences in first-order logic,using a consistent vocabulary(which you must define);

- **a.** Some students tool French in spring 2001.
- **b.** Every student who takes French passes it.
- **c.** Only one student took Greek in spring 2001.
- **d.** The best score in Greek is always higher than the best score in French.
- **e.** Every person who buys a policy is smart.
- **f.** No person buys an expensive policy.
- **g.** There is an agent who sells policies only to people who are not insured.
- **h.** There is a barber who shaves all men in town who do not shave themselves.
- **i.** A person born in the UK,each of whose parents is a UK citizen or a UK resident,is a UK citizen by birth.
- **j.** A person born outside the UK,one of whose parents is a UK citizen by birth,is a UK citizen by descent.
- **k.** Politicians can fool some of the people all of the time,and they can fool all of the people some of the time,but they can't fool all of the people all of the time.

首先，根据题意定义一系列函数和常量

- Take(student,course,time),真值函数,student在time的时候上了course
- Pass(student,course),真值函数,student通过了course的考试
- Score(student,course,time),返回student在time的时候上了course的成绩
- Gt(scores,scoreb),真值函数，分数scorea比分数scoreb高
- EQ(x,y),真值函数，x和y相同
- isSmart(person),真值函数，person是聪明的
- Buy(person,good),真值函数，person买了good
- Sell(x,y,z),真值函数，x卖z给y
- isStudent(x),真值函数，x是学生
- isPerson(x),真值函数，x是人
- isPolicy(x),真值函数，x是保险
- isAgent(x),真值函数，x是代理
- isBarber(x),真值函数，x是理发师
- isEngCitizen(x,t),真值函数，x在t程度上是英国公民
- isEngLiver(x,t),真值函数，x在t程度上是英国永久居住者
- isPolitician(x),真值函数，x是政治家
- isExpensive(x),真值函数，x是昂贵的
- isInsured(x),真值函数，x已经投保
- isIntown(x),真值函数，x在镇上
- isMan(x),真值函数，x是男人
- Shave(x,y),真值函数，x给y刮胡子
- Born(x,y),真值函数，x出生于y
- Parent(x,y),真值函数，x是y的双亲之一
- Fool(x,y,t),真值函数，x在t时候愚弄y
- 各类常量，比如French，Greek，2001Spring，England，Birth，Descent

于是将诸语句表示如下

- **a.** $\exists x(isStudent(x) \wedge Take(x, French, 2001Spring))$
- **b.** $\forall x((isStudent(x) \wedge \exists time(Take(x, French, time))) \Rightarrow Pass(x, French))$
- **c.** $\exists x(isStudent(x) \wedge Take(x, Greek, 2001Spring) \wedge (\forall y(\neg EQ(x, y) \Rightarrow \neg Take(x, Greek, 2001Spring))))$
- **d.** $\forall time \exists x \forall y GT(Score(x, Greek, time), Score(y, French, time))$
- **e.** $\forall x(isPerson(x) \wedge (\exists y(isPolicy(y) \wedge Buy(x, y))) \Rightarrow isSmart(x))$

- **f.** $\neg\exists x(isPerson(x) \wedge \exists y(isPolicy(y) \wedge Buy(x,y) \wedge isExpensive(y)))$
- **g.** $\exists x(isAgent(x) \wedge \forall y,z((Sell(x,y,z) \wedge isPerson(y) \wedge isPolicy(z)) \Rightarrow \neg isInsured(y)))$
- **h.** $\exists x(isBarber(x) \wedge \forall y((isIntown(y) \wedge isMan(y) \wedge \neg Shave(y,y)) \Rightarrow Shave(x,y)))$
- **i.** 此处将题意理解为，成为英国公民或永久居住者之后这身份始终有效，而且此处的父母可以不是人类

$$\begin{aligned}&\forall x(isPerson(x) \wedge Born(x, England) \wedge \\ &\qquad \forall y(Parent(y,x) \Rightarrow \exists t(EngCitizen(y,t) \vee EngLiver(y,t))) \\ &\quad \Rightarrow EngCitizen(x, Birth))\end{aligned}$$

- **j.** 此处理解同上

$$\begin{aligned}&\forall x(isPerson(x) \wedge \neg Born(x, England) \wedge \\ &\qquad \exists y(Parent(y,x) \wedge EngCitizen(y, Birth)) \\ &\quad \Rightarrow EngCitizen(x, Descent))\end{aligned}$$

- **k.** 同样，政治家也可以不是人，这取决于具体的语义

$$\begin{aligned}&\forall x(isPolitician(x) \Rightarrow \\ &\qquad (\exists y \forall t Fool(x,y,t)) \\ &\qquad \wedge (\exists t \forall y Fool(x,y,t)) \\ &\qquad \wedge \neg(\forall y \forall t Fool(x,y,t)))\end{aligned}$$

8.17 Explain what is wrong with the following proposed definition of adjacent squares in the wampum world:

$\forall x,y \quad Adjacent([x,y],[x+1,y]) \wedge Adjacent([x,y],[x,y+1])$

解：这个定义太过简陋且有谬误，它仅仅规定了所有格子与其右侧与上侧的格子相连（假设右和上分别是x和y的增长方向），那么当格子位于右侧与上侧的边缘时，显然其是错误的，因为这些边缘上的格子已经没有右侧或者上侧的格子与之对应。而且，处于非边缘处的格子理应有四个方向的格子与其相连，而这里仅仅提到了两个。最后，其并没有显式规定其它的格子是否相连，比如那些相距3个格子的格子。

9.3 Suppose a knowledge base contains just one sentence,∃ x AsHighAs(x,Everest).Which of the following arc legitimate results of applying Existential Instantiation?

- **a.** $AsHighAs(Everest, Everest)$
- **b.** $AsHighAs(Kilimanjaro, Everest)$
- **c.** $AsHighAs(Kilimanjaro, Everest) \wedge AsHighAs(BenNevis, Everest)$(after two applications)

解：只有**b**是合法的。**a**替换了同名符号，而**c**将存在量词实例化用了两次。

9.4 For each pair of atomic sentences, give the most general unifier if it exists:

- **a.** $P(A,B,B), P(x,y,z)$
- **b.** $Q(y,G(A,B)), Q(G(x,x),y)$
- **c.** $Older(Father(y),y), Older(Father(x),John)$
- **d.** $Knows(Father(y),y), Knows(x,x)$

解：由题意，分析如下

- **a.** $Unify(P(A,B,B),P(x,y,z)) = \{x/A, y/B, z/B\}$
- **b.** $Unify(Q(y,G(A,B)),Q(G(x,x),y)) = fail$
- **c.** $Unify(Older(Father(y),y),Older(Father(x),John)) = \{x/John, y/John\}$
- **d.** $Unify(Knows(Father(y),y),Knows(x,x)) = fail$

9.6 Write down logical representations for the following sentences, suitable for use with Generalized Modus Ponens:

- **a.** Horses, cows, and pigs are mammals.
- **b.** An offspring of a horse is a horse.
- **c.** Bluebeard is a horse.
- **d.** Bluebeard is Charlie’s parent.
- **e.** Offspring and parent are inverse relations.
- **f.** Every mammal has a parent.

解：词汇表如下

- $Horse(), Cow(), Pig()$：是马/牛/猪
- $Offspring(x, y)$：x是y的后代
- $Parent(x, y)$：x是y的双亲
- $Mammal(x)$：x是哺乳动物

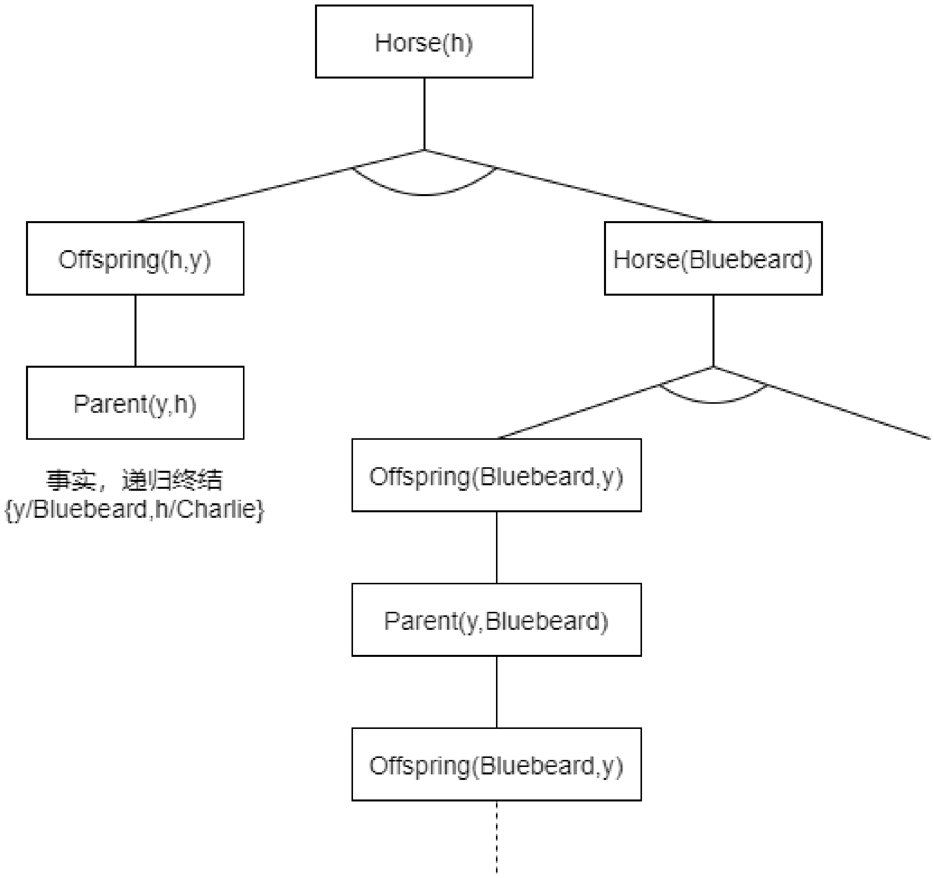
，由题意罗列如下

- **a.**
$$\begin{aligned} Horse(x) &\Rightarrow Mammal(x) \\ Cow(x) &\Rightarrow Mammal(x) \\ Pig(x) &\Rightarrow Mammal(x) \end{aligned}$$
- **b.** $Horse(y) \wedge Offspring(x, y) \Rightarrow Horse(x)$
- **c.** $Horse(Bluebeard)$
- **d.** $Parent(Bluebeard, Charlie)$
- **e.**
$$\begin{aligned} Offspring(x, y) &\Rightarrow Parent(y, x) \\ Parent(y, x) &\Rightarrow Offspring(x, y) \end{aligned}$$
- **f.** $Mammal(x) \Rightarrow Parent(M(x), x)$

9.13 In this exercise, use the sentences you wrote in Exercise 9.6 to answer a question by using a backward-chaining algorithm.

- **a.** Draw the proof tree generated by an exhaustive backward-chaining algorithm for the query $\exists h Horse(h)$, where clauses are matched in the order given.
- **b.** What do you notice about this domain?
- **c.** How many solutions for h actually follow from your sentences?

a. 解：由题意绘图如下



b. 解：由上图可见其陷入了无限循环。一方面这说明了后向链接算法并不完备，另一方面也可以看出来，match的顺序对于算法运行的过程和结果有着很大的影响，如果最先被match的是 $Horse(Bluebeard)$ 而非 $Horse(y) \land Offspring(x, y) \Rightarrow Horse(x)$,那么算法将直接终结而不是陷入无限循环。

c. 解：$Bluebeard$ ， $Charlie$都是满足 $Horse(h)$ 的。

人工智能第七次作业

13.15 Answer the Question

After your yearly checkup, the doctor has bad news and good news. The bad news is that you tested positive for a serious disease and that the test is 99% accurate (i.e., the probability of testing positive when you do have the disease is 0.99, as is the probability of testing negative when you don't have the disease). The good news is that this is a rare disease, striking only 1 in 10,000 people of your age. Why is it good news that the disease is rare? What are the chances that you actually have the disease?

解：设 A 为是否患有该病，B 为是否测试呈阳性，计算确实患有这种病的概率为

$$
\begin{aligned}
P(A=1|B=1) &= P(A=1 \wedge B=1)/P(B=1) \\
&= (P(B=1|A=1)P(A=1))/(P(B=1|A=1)P(A=1)+P(B=1|A=0)P(A=0)) \\
&= (99\% \times \frac{1}{10000})/(99\% \times \frac{1}{10000} + (1-99\%) \times (1-\frac{1}{10000})) \\
&= \frac{1}{102} \approx 0.98\%
\end{aligned}
$$

也就是说因为这个“好消息”，实际上患病概率只有0.98%，所以它的确是个好消息。

13.18 Answer the Question

Suppose you are given a bag containing n unbiased coins. You are told that $n-1$ of these coins are normal, with heads on one side and tails on the other, whereas one coin is a fake, with heads on both sides.

- **a** Suppose you reach into the bag, pick out a coin at random, flip it, and get a head. What is the (conditional) probability that the coin you chose is the fake coin?
- **b** Suppose you continue flipping the coin for a total of k times after picking it and see k heads. Now what is the conditional probability that you picked the fake coin?
- **c** Suppose you wanted to decide whether the chosen coin was fake by flipping it k times. The decision procedure returns $fake$ if all k flips come up heads; otherwise it returns $normal$. What is the (unconditional) probability that this procedure makes an error?

解：分小题解答如下

- **小题a**：设 A 为是否拿到假币，B 为是否为正面

$$
\begin{aligned}
P(A=1|B=1) &= P(A=1 \wedge B=1)/P(B=1) \\
&= (P(B=1|A=1)P(A=1))/(P(B=1|A=1)P(A=1)+P(B=1|A=0)P(A=0)) \\
&= (1 \times \frac{1}{n})/(1 \times \frac{1}{n} + 0.5 \times (1-\frac{1}{n})) \\
&= \frac{2}{n+1}
\end{aligned}
$$

- **小题b**：设 A 为是否拿到假币，B为是否连续 k 次正面

$$
\begin{aligned}
P(A=1|B=1) &= P(A=1 \wedge B=1)/P(B=1) \\
&= (P(B=1|A=1)P(A=1))/(P(B=1|A=1)P(A=1)+P(B=1|A=0)P(A=0)) \\
&= (1\times\frac{1}{n})/(1\times\frac{1}{n}+\frac{1}{2^k}\times(1-\frac{1}{n})) \\
&= \frac{2^k}{n+2^k-1}
\end{aligned}
$$

- **小题c**：设 A 为是否拿到假币，B为是否连续 k 次正面

$$
\begin{aligned}
P(A=1 \wedge B=0)+P(A=0 \wedge B=1) &= 0+P(A=0 \wedge B=1) \\
&= P(B=1|A=0)P(A=0) \\
&= \frac{1}{2^k}\times\frac{1}{n} \\
&= \frac{n-1}{n2^k}
\end{aligned}
$$

13.21 (Adapted from Pearl (1988).) Answer the Question

Suppose you are a witness to a nighttime hit-and-run accident involving a taxi in Athens. All taxis in Athens are blue or green. You swear, under oath, that the taxi was blue. Extensive testing shows that, under the dim lighting conditions, discrimination between blue and green is 75% reliable.

- **a.** Is it possible to calculate the most likely color for the taxi? (*Hint*: distinguish carefully between the proposition that the taxi is blue and the proposition that it appears blue.)
- **b.** What if you know 9 out of 10 Athenian taxis are green?

解：分小题解答如下

- **小题a**：不可能，除非结合生活常识。我们假设当地绿色出租车的比例为 p ，计算这个猜测下肇事车辆为绿色出租车的概率如下

$$
\begin{aligned}
P(\text{肇事为蓝车}|\text{目击为蓝车}) &= P(\text{肇事为蓝车} \wedge \text{目击为蓝车})/P(\text{目击为蓝车}) \\
&= \frac{P(\text{目击为蓝车}|\text{肇事为蓝车})\times P(\text{肇事为蓝车})}{P(\text{目击为蓝车}|\text{肇事为蓝车})\times P(\text{肇事为蓝车})+P(\text{目击为蓝车}|\text{肇事非蓝车})\times P((\text{肇事非蓝车}))} \\
&= \frac{3/4\times p}{3/4\times p+(1-3/4)\times(1-p)} \\
&= \frac{3p}{2p+1}
\end{aligned}
$$

随着 p 值在 $(0,1)$ 变动这个概率可以在 $(0,1)$间变动，所以无法判断

- **小题b**：代入小题 a 的结论，得到肇事车辆为蓝车的概率是 $0.3/1.2=25\%$，不是蓝车的概率是 75%。

13.22 Answer the Question

Text categorization is the task of assigning a given document to one of a fixed set of categories on the basis of the text it contains. Naive Bayes models are often used for this task. In these models, the query variable is the document category, and the “effect” variables are the presence or absence of each word in the language; the assumption is that words occur independently in documents, with frequencies determined by the document category.

- **a.** Explain precisely how such a model can be constructed, given as “training data” a set of documents that have been assigned to categories.
- **b.** Explain precisely how to categorize a new document.

- **c.** Is the conditional independence assumption reasonable? Discuss.

解：分小题解答如下

- **小题a：** 对每一个分好类的文档的类别，统计其各个单词出现的频率，可以构建出模型的先验概率 $P(Category=A)$： 描述各文档中类别 A 所占的比例；$P(Word_i|Category=A)$：描述类别A的文档里单词 $Word_i$ 出现的频率
- **小题b：** 对每一个给定的新文档，统计其中各单词的出现频率。根据贝叶斯公式算出其归属于各个 $Category$ 的概率，将其归类到概率最大的一类中
- **小题c：** 不合理，单词有可能具有前后的关联，导致其出现概率不能简单看作其组成部分的概率的乘积。

人工智能第八次作业

14.12

Two astronomers in different parts of the world make measurements M_1 and M_2 of the number of stars N in some small region of the sky, using their telescopes. Normally, there is a small possibility e of error by up to one star in each direction. Each telescope can also (with a much smaller probability f) be badly out of focus (events F_1 and F_2), in which case the scientist will undercount by three or more stars (or if N is less than 3 , fail to detect any stars at all). Consider the three networks shown in Figure 14.22.

- **a.** Which of these Bayesian networks are correct (but not necessarily efficient) representations of the preceding information?
- **b**. Which is the best network? Explain.
- **c**. Write out a conditional distribution for $P(M_1|N)$, for the case where $N \in 1,2,3$ and $M_1 \in 0,1,2,3,4$. Each entry in the conditional distribution should be expressed as a function of the parameters e and/or f .
- **d**. Suppose $M_1 = 1$ and $M_2 = 3$. What are the possible numbers of stars if you assume no prior constraint on the values of N ?
- **e**. What is the *most likely* number of stars, given these observations? Explain how to compute this, or if it is not possible to compute, explain what additional information is needed and how it would affect the result.

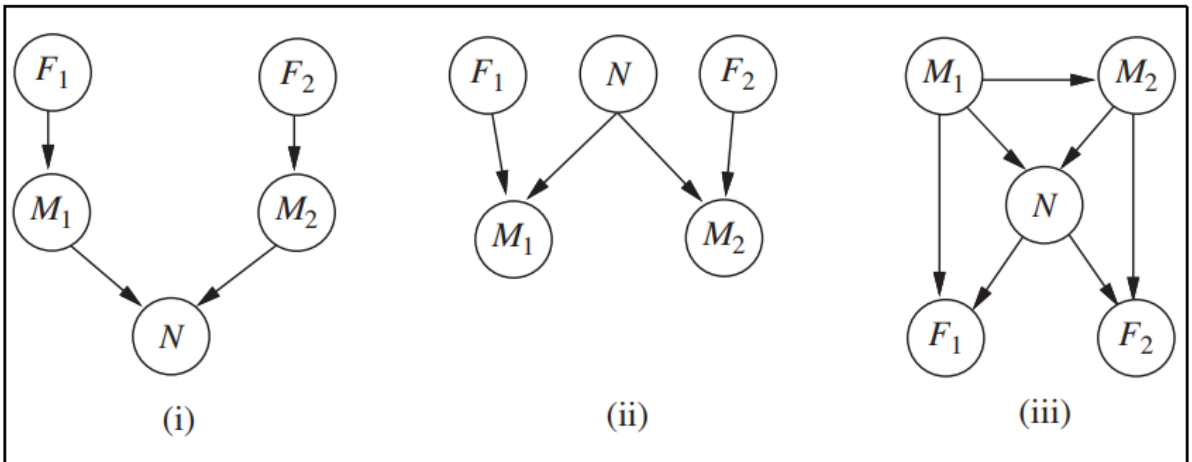


Figure 14.22 Three possible networks for the telescope problem.

解答：分小题解答如下

- **a.** （i）情况并不正确，星星的实际数目不应当对于每次望远镜对焦得到的数目是无关系的存在。（ii）正确，（iii）则是根据一定的考虑顺序，将隐性的关系也放了进来，它同样是正确的。
- **b.** 最好的情况应该是（ii），不考虑错误的（i），与（iii）相较，其更加精简，保留了最关键的关联。
- **c.** 条件概率表如下（此处将 e 理解为多/少数一颗星星的概率，即二者分开计算，同时若因对焦问题丢失星星，在 $N \leq 3$ 的情况下，数出来的必然是0）

$P(M_1\|N)$	$N=1$	$N=2$	$N=3$
$M_1=0$	$f+e(1-f)$	f	f
$M_1=1$	$(1-2e)(1-f)$	$e(1-f)$	0
$M_1=2$	$e(1-f)$	$(1-2e)(1-f)$	$e(1-f)$
$M_1=3$	0	$e(1-f)$	$(1-2e)(1-f)$
$M_1=4$	0	0	$e(1-f)$

- **d.** 可能是 $2, 4, \geq 6$。
- **e.** 首先，我们需要知道星星数目的概率分布，不然将无法衡量在 $M_1=1, M_2=3$ 的情况下 N 的取值的可能。此处对 N 的分布做出假设，设 $N=2$ 的概率为 p_2 ，$N=4$ 的概率为 p_4 ，$N\geq 6$ 的概率为 $p_{\geq 6}$ 。同时，由于对焦丢失星星的具体颗数不明，所以此处仅仅列举诸概率的上限如下：

$P(N=2, M_1=1, M_2=3)$	$P(N=4, M_1=1, M_2=3)$	$P(N\geq 6, M_1=1, M_2=3)$
$p_2e^2(1-f)^2$	p_4ef	$p_{\geq 6}f^2$

推断星星颗数最可能的数目时，需要结合以上概率进行判断。

14.13

Consider the network shown in Figure 14.22(ii), and assume that the two telescopes work identically. $N\in 1,2,3$ and $M_1, M_2 \in 0,1,2,3,4$, with the symbolic CPTs as described in Exercise 14.12. Using the enumeration algorithm (Figure 14.9 on page 525), calculate the probability distribution $P(N|M_1=2, M_2=2)$.

function ENUMERATION-ASK(X, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
      e, observed values for variables **E**
      bn, a Bayes net with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$ /* **Y** = *hidden variables* */

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL(bn.VARS, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is **e** extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-ALL($vars$, **e**) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $Y \leftarrow$ FIRST($vars$)
  if Y has value y in **e**
    then return $P(y \mid parents(Y)) \times$ ENUMERATE-ALL(REST($vars$), **e**)
    else return $\sum_y P(y \mid parents(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_y$)
      where $\mathbf{e}_y$ is **e** extended with $Y = y$

Figure 14.9 The enumeration algorithm for answering queries on Bayesian networks.

解答： 枚举所有会导致 $M_1=2, M_2=2$ 的情况，将其按照 N 的取值进行分类，分别求和计算其概率。由于此处仅需要考虑 $N=1,2,3$ 的情况，不难证明望远镜并没有出现对焦不准确的情况，设客观世界里星星数的概率分布满足 $P(N=1)=p_1$ ，$P(N=2)=p_2$ ，$P(N=3)=p_3$。为精简起见计算其相对概率如下

$$
\begin{aligned}
< P(N=1|M_1=2, M_2=2), P(N=2|\ldots), P(N=2|\ldots) > &= C < p_1e^2+0, p_2(1-2e)^2+0, p_3e^2+0 > \\
&= C < p_1e^2, p_2(1-2e)^2, p_3e^2 >
\end{aligned}
$$