

第三章 问题求解

1. 状态空间

上节的例子展示了人工智能问题的一般解决方案：

- (1) 缩小问题范围，减少功能，适当划分人机功能。
- (2) 机器利进行答案搜索
 - (2.1) 穷尽式地调查，尽量挖掘问题的约束条件
 - (2.2) 机器利用约束条件和约束条件的传播，缩小搜索范围

但上节的例子中求解过程是人在进行的，若要机器操作，必须要形式化。本节提出一种形式化的框架，即状态空间。

装空间有两个要素：状态和规则。

状态空间把求解过程中的问题分成若干状态，包括一个起始状态，一个或多个目标状态，若干中间状态。问题的求解过程就是由起始状态经由中间状态到达目标状态。

状态之间的转换依靠规则。规则有条件和动作两部分组成。条件是对于当前状态性质的一些要求，动作是转移到某一个状态。

状态和规则都必须形式化。

例 1. 农夫过河问题：农夫带一只狼、一只羊、一棵白菜，利用一条船过河。船必须由农夫划，船上除农夫外只能带一样东西。离开农夫则狼要吃羊，羊要吃白菜。

状态的形式化表示：

符号：每一个角色用一个字母表示。农夫 n，狼 l，羊 y，白菜 b。

符号串：左岸角色串 LeftBank 为在左岸的角色的字母串，右岸角色串 RightBank 为在右岸的角色的字母串。这两个串中的字母都取自 n、l、y、b 之一，每个字母在两个串中必出现一次且只出现一次。

状态：(LeftBank, RightBank)，

例如：起始状态为 ((n l y b) ()), 目标状态为 (() (n l y b))。

规则由两部分组成，形式为

条件→动作

本例中的规则为：

- (1) L(n), L(y) → LtoR(n,y)
- (2) L(n), L(l), ~[L(y), L(b)] → LtoR(n,l)
- (3) L(n), L(b), ~[L(l), L(y)] → LtoR(n,b)
- (4) L(n), ~[L(y), L(b)] , ~[L(l), L(y)] → LtoR(n)
- (5) R(n), R(y) → RtoL(n,y)
- (6) R(n), R(l), ~[R(y), R(b)] → RtoL(n,l)
- (7) R(n), R(b), ~[R(l), R(y)] → RtoL(n,b)
- (8) R(n), ~[R(y), R(b)] , ~[R(l), R(y)] → RtoL(n)

其中 L(x)表示角色 x 在左岸 (x 在字符串 LeftBank 中)，R(x)表示角色 x 在右岸 (x 在字符串 RightBank 中)，~[……]表示括号中的条件不满足，LtoR(x)或 LtoR(x,y)表示将角色 x 或 x 和 y 从左岸带到右岸(将字母 x 或 x 和 y 从字符串 LeftBank 移至 RightBank 中)，RtoL(x)或 RtoL(x,y)表示将角色 x 或 x 和 y 从右岸带到左岸(将字母 x 或 x 和 y 从字符串 RightBank 移至 LeftBank 中)。

状态空间为：

((n l y b) ())

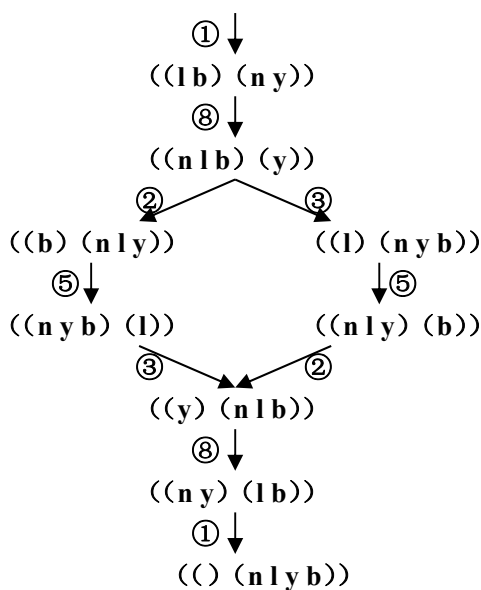


图 3-1

上面的规则可以写成更加贴近与计算机数据结构的形式。设长度为 4 的字符型数组组成结构 NLYB, 这个数组表示农夫、狼、羊、白菜在左岸的存在状况。数组的四个位置分别用变量 N (代表农夫)、L (代表狼)、Y (代表羊)、B (代表白菜) 表示, 变量取 1 表示存在, 取零表示不存在。于是, 规则可以这样写:

- (1) NLYB(1,L,1,B)→NLYB(0,L,0,B)
- (2) NLYB(1,1,Y,B), !(Y&&B)→NLYB(0,0,Y,B)
- (3) NLYB(1,L,Y,1), !(L&&Y)→NLYB(0,L,Y,0)
- (4) NLYB(1,L,Y,B), !(L&&Y), !(Y&&B)→NLYB(0,L,Y,B)
- (5) NLYB(0,L,0,B)→NLYB(1,L,1,B)
- (6) NLYB(0,0,Y,B), !(Y&&B)→NLYB(1,1,Y,B)
- (7) NLYB(0,L,Y,0), !(L&&Y)→NLYB(1,L,Y,1)
- (8) NLYB(0,L,Y,B), !(L&&Y), !(Y&&B)→NLYB(1,L,Y,B)

状态空间是一种由节点和有向边组成的图结构, 其中节点是状态, 有向边是状态到状态的变换, 这种变换是施用规则的结果, 因此有时把使用的规则号标注在有向边上。

状态空间由起始状态开始。起始状态节点和其他一些状态节点有其子节点, 即施用规则后得到的新状态。有些状态节点没有子节点, 原因是没有规则适用于这些节点, 或者它们满足了问题求解的条件, 无需再生成新的状态。后者称为目标节点。问题求解的任务同目标节点紧密相关, 下一个例子将进一步说明。

2. 规则的形式

例 2. 八数码问题。

1 到 8 这 8 个数字的数牌和一个空位排成 3*3 的阵列。要求通过逐次交换空位与相邻数牌, 把阵列排成标准形式。下面是一个一般 8 数码阵列向标准阵列转换的实例。

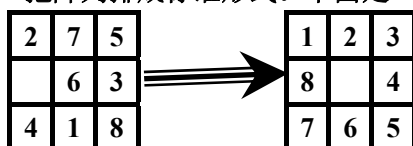


图 3-2

这个问题中状态就是 3*3 阵列。只要把空位用一个符号表示, 这样的阵列就能表示成数字、

括号和空格符号组成的符号串：

$$((275)(x63)(418)) \Rightarrow ((123)(8x4)(765))$$

图 3-3

为了直观易懂，本例中的状态仍用 3*3 的表格表示。

为了写出状态变换规则，我们用 S_{ij} 记录第 i 行、第 j 列的阵列位置，如图 3-4

S_{11}	S_{12}	S_{13}
S_{21}	S_{22}	S_{23}
S_{31}	S_{32}	S_{33}

图 3-4

规则列举如下：

- (1) $S_{11}=x \rightarrow \text{exchange}(S_{11}, S_{12})$
- (2) $S_{11}=x \rightarrow \text{exchange}(S_{11}, S_{21})$
- (3) $S_{12}=x \rightarrow \text{exchange}(S_{12}, S_{11})$
- (4) $S_{12}=x \rightarrow \text{exchange}(S_{12}, S_{13})$
- (5) $S_{12}=x \rightarrow \text{exchange}(S_{12}, S_{22})$
- (6) $S_{13}=x \rightarrow \text{exchange}(S_{13}, S_{12})$
- (7) $S_{13}=x \rightarrow \text{exchange}(S_{13}, S_{23})$
- (8) $S_{21}=x \rightarrow \text{exchange}(S_{21}, S_{11})$
- (9) $S_{21}=x \rightarrow \text{exchange}(S_{21}, S_{31})$
- (10) $S_{21}=x \rightarrow \text{exchange}(S_{21}, S_{22})$
- (11) $S_{22}=x \rightarrow \text{exchange}(S_{22}, S_{12})$
- (12) $S_{22}=x \rightarrow \text{exchange}(S_{22}, S_{32})$
- (13) $S_{22}=x \rightarrow \text{exchange}(S_{22}, S_{21})$
- (14) $S_{22}=x \rightarrow \text{exchange}(S_{22}, S_{23})$
- (15) $S_{23}=x \rightarrow \text{exchange}(S_{23}, S_{13})$
- (16) $S_{23}=x \rightarrow \text{exchange}(S_{23}, S_{22})$
- (17) $S_{23}=x \rightarrow \text{exchange}(S_{23}, S_{31})$
- (18) $S_{31}=x \rightarrow \text{exchange}(S_{31}, S_{21})$
- (19) $S_{31}=x \rightarrow \text{exchange}(S_{31}, S_{32})$
- (20) $S_{32}=x \rightarrow \text{exchange}(S_{32}, S_{31})$
- (21) $S_{32}=x \rightarrow \text{exchange}(S_{32}, S_{22})$
- (22) $S_{32}=x \rightarrow \text{exchange}(S_{32}, S_{33})$
- (23) $S_{33}=x \rightarrow \text{exchange}(S_{33}, S_{23})$
- (24) $S_{33}=x \rightarrow \text{exchange}(S_{33}, S_{32})$

还有一种给出规则的方式：

- (1) $S_{i,j}=x, i>1 \rightarrow \text{exchange}(S_{ij}, S_{i-1,j})$
- (2) $S_{i,j}=x, i<3 \rightarrow \text{exchange}(S_{ij}, S_{i+1,j})$
- (3) $S_{i,j}=x, j>1 \rightarrow \text{exchange}(S_{ij}, S_{i,j-1})$
- (4) $S_{i,j}=x, j<3 \rightarrow \text{exchange}(S_{ij}, S_{i,j+1})$

第一种规则穷尽式地列出所有可能的状态，给出该状态下的状态变换方式。第二种规则列出状态的某种特征，给出满足这种特征时的状态变换方式。第一种规则形式可以认为是外延式，因为它的条件部分是穷尽地列举状态。这种规则形式的优点是状态与规则条件的匹配非常简单，直接比较数字就可以决定一条规则能不能用，若能则可以直接执行动作。缺点是规则比较多，储存规则所需要的存储开销大；为寻找适用规则需将当前状态同规则

的条件逐条匹配，相当费时间。第二种规则形式可以认为是内涵式，它的条件部分用数学表达式展示条件的内涵。这种规则形式的优点是节省存储空间，寻找适用规则时需要检查的规则较少。缺点是每一次匹配的过程比较复杂，要计算数学表达式，而且当匹配成功时需要记住参数值（本例中是阵列下标 i 和 j ），因为要执行的动作依赖于参数值。实际问题中采用哪种形式的规则往往需要视 ([具体情况 ([进行 ([权衡 ([。下面列出的 8 数码问题的规则是一种折中的形式：

- (1) $S_{1j}=x \rightarrow \text{exchange}(S_{1j}, S_{2j})$
- (2) $S_{2j}=x \rightarrow \text{exchange}(S_{2j}, S_{1j})$
- (3) $S_{2j}=x \rightarrow \text{exchange}(S_{2j}, S_{3j})$
- (4) $S_{3j}=x \rightarrow \text{exchange}(S_{3j}, S_{2j})$
- (5) $S_{i1}=x \rightarrow \text{exchange}(S_{i1}, S_{i2})$
- (6) $S_{i2}=x \rightarrow \text{exchange}(S_{i2}, S_{i1})$
- (7) $S_{i2}=x \rightarrow \text{exchange}(S_{i2}, S_{i3})$
- (8) $S_{i3}=x \rightarrow \text{exchange}(S_{i3}, S_{i2})$

下面以第二种规则形式画出局部的状态空间图

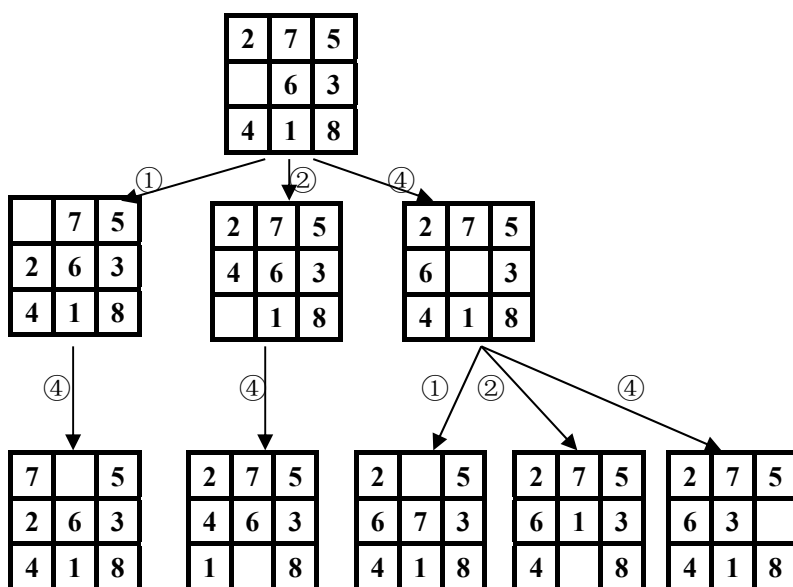


图 3-6

3. 问题求解的任务

这个状态空间图有两点需要注意：首先，它只是局部的。许多情况下状态空间涉及的状态太多，有些甚至可以无限伸展，无法一一列出，所以只画局部的。第二，有的状态节点的子节点不止一个，或者说适用于该状态的规则不止一个。为了使得由一个状态生成新状态（即由一个节点生成其子节点）的工作符合算法化的要求，就需要有一种方法，在适用的规则中选定一个，也就是在从一个状态节点出发的不同路径中选定一条路径。

从状态空间的观点看，一项具体的问题求解工作的任务是：

搜索从起始状态到达目标状态的路径。

搜索从起始状态到达目标状态的最优路径。

研究一般的问题求解工作的任务是：

寻找一种最好的方法，用于在状态空间中搜索到达目标状态的最优路径。

实际的人工智能问题非常复杂，交给机器做的往往已经不是原问题了。这时，机器寻找的

目标节点未必就是真正的目标节点，可能只是近似的符合要求的节点；机器寻找的最优路径未必真是最优的，可能只是比较好的路径。这是从解决问题的效果看，人工智能计算问题同传统计算问题的差别。

后面要详细介绍有关状态空间的两个基本技术：知识表示技术，用于状态和规则的形式化；搜索技术，用于寻找路径。

4. 例：直线图问题的状态和规则

从第2章已知，在相应的限制条件之下，直线图只有以下类型的顶点（叉形顶点增加两种，以便不必判断旋转同构）：

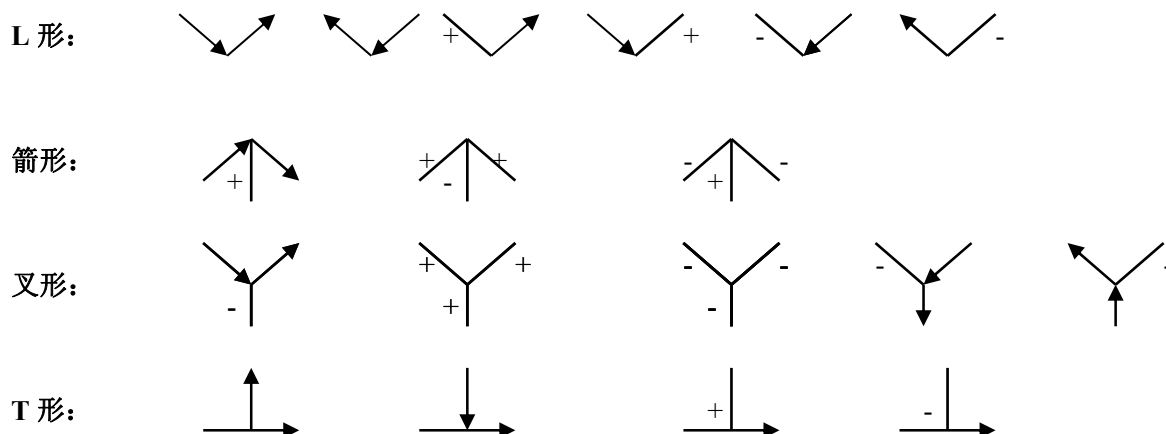


图 2-13

顶点有 4 种：L 形、箭形、叉形、T 形，分别用字符 L、X、Y、T 表示。构成顶点的边有 4 种：入边（边界线，箭头指向顶点）、出边（边界线，箭头离开顶点）、凸边、凹边，分别用字符 i、u、+、- 表示。未知边用 q 表示。

再规定每一类顶点中边的排列顺序：

L 形顶点当缺口朝上时，顺序为左边和右边。

X 形顶点当箭头朝上时，顺序为左翼、右翼、箭杆。

Y 形顶点某个叉口朝上时，顺序为左叉、右叉、下叉。

T 形顶点当摆成倒 T 形时，顺序为垂直边、左水平边、右水平边。

如此，可能的标记完的顶点可以表示成 Lxx、Xxxx、Yxxx 和 Txxx 的形式，它们是：

Liu, Lui, L+u, Li+, L-i, Lu-, Xiu+, X++-, X--+, Yiu-, Yu-i, Y-iu, Y+++ , Y---, Tuii, Tiiu, T+iu, T-iu

未标记完的顶点带有字符 q。可以设计一种规则系统，其中每一条规则就是把带有 q 标记边的顶点变成没有 q 标记边的顶点。规则如下：

Liq→Liu

Lqu→Liu

Luq→Lui

Lqi→Lui

L+q→L+u

Lqu→L+u

Liq→Li+

Lq+→Li+

L-q→L-i
 Lqi→L-i
 Luq→Lu-
 Lq-→Lu-
 Xiqq→Xiu+
 Xquq→Xiu+
 Xqq+→Xiu+
 Xiuq→Xiu+
 Xiq+→Xiu+
 Xqu+→Xiu+
 X+qq→X++-
 Xq+q→X++-
 Xqq-→X++-
 X++q→X++-
 X+q-→X++-
 Xq+-→X++-
 X-qq→X--+
 Xq-q→X--+
 Xqq+→X--+
 X--q→X--+
 X-q+→X--+
 Xq-+→X--+
 Yiqq→Yiu-
 Yquq→Yiu-
 Yqq-→Yiu-
 Yiuq→Yiu-
 Yiq-→Yiu-
 Yqu-→Yiu-
 Yuqq→Yu-i
 Yq-q→Yu-i
 Yqqi→Yu-i
 Yu-q→Yu-i
 Yuqi→Yu-i
 Yq-i→Yu-i
 Y-qq→Y-iu
 Yqiq→Y-iu
 Yqqu→Y-iu
 Y-iq→Y-iu
 Y-qu→Y-iu
 Yqiu→Y-iu
 Y+qq→Y+++
 Y+qq→Y+++
 Yqq+→Y+++
 Y++q→Y+++

$Y+q+\rightarrow Y+++$
 $Yq++\rightarrow Y+++$
 $Y-qq\rightarrow Y---$
 $Yq-q\rightarrow Y---$
 $Yqq-\rightarrow Y---$
 $Y--q\rightarrow Y---$
 $Y-q-\rightarrow Y---$
 $Yq--\rightarrow Y---$
 $Tuqq\rightarrow Tuiu$
 $Tqiq\rightarrow Tuiu$
 $Tqqu\rightarrow Tuiu$
 $Tuiq\rightarrow Tuiu$
 $Tuqu\rightarrow Tuiu$
 $Tqiu\rightarrow Tuiu$
 $Tiqq\rightarrow Tiiu$
 $Tqiq\rightarrow Tiiu$
 $Tqqu\rightarrow Tiiu$
 $Tiiq\rightarrow Tiiu$
 $Tiqu\rightarrow Tiiu$
 $Tqiu\rightarrow Tiiu$
 $T+qq\rightarrow T+iu$
 $Tqiq\rightarrow T+iu$
 $Tqqu\rightarrow T+iu$
 $T+iq\rightarrow T+iu$
 $T+qu\rightarrow T+iu$
 $Tqiu\rightarrow T+iu$
 $T-qq\rightarrow T-iu$
 $Tqiq\rightarrow T-iu$
 $Tqqu\rightarrow T-iu$
 $T-iq\rightarrow T-iu$
 $T-qu\rightarrow T-iu$
 $Tqiu\rightarrow T-iu$
 $Lqq\rightarrow Liu$
 $Lqq\rightarrow Lui$
 $Lqq\rightarrow L+u$
 $Lqq\rightarrow Li+$
 $Lqq\rightarrow L-i$
 $Lqq\rightarrow Lu-$
 $Xqqq\rightarrow Xiu+$
 $Xqqq\rightarrow X++-$
 $Xqqq\rightarrow X--+$
 $Yqqq\rightarrow Yiu-$
 $Yqqq\rightarrow Yu-i$
 $Yqqq\rightarrow Y-iu$

```

Yqqq→Y+++
Yqqq→Y---
Tqqq→Tuiu
Tqqq→Tiiu
Tqqq→T+iu
Tqqq→T-iu
Liu→go
Lui→go
L+u→go
Li+→go
L-i→go
Lu-→go
Xiu+→go
X++-→go
X--+→go
Yiu-→go
Yu-i→go
Y-iu→go
Y+++→go
Y---→go
Tuiu→go
Tiiu→go
T+iu→go
T-iu→go
other→error

```

当规则左部是 V_{xx} 或 V_{xxx} 形式，其中没有一个 x 是 q 时，说明当前顶点已经正确地标记完，此时应转到尚未标记的顶点去；如果所有顶点都标记完了，系统就停止。本规则系统用 go 表示这一含义。

上面的规则集中，出最后一条以外，都是彼此独立的，前后顺序可以颠倒，但最后一条的位置不能变，只有当前面所有规则都匹配失败后才能使用这一条规则。

这样写的规则属于外延型规则，规则数目太多，不仅大量消耗时间和空间资源，还容易引起错误。下面给出内涵式规则的例子。

$Data=\{Liu, Lui, L+u, Li+, L-i, Lu-, Xiu+, X++-, X--+, Yiu-, Yu-i, Y-iu, Y+++, Y---, Tuiu, Tiiu, T+iu, T-iu\}$

```

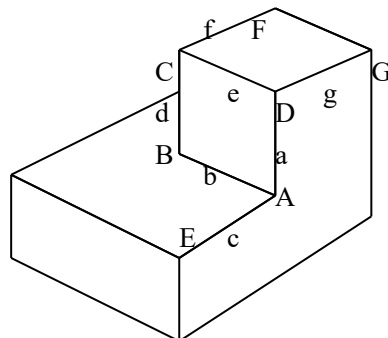
s= match(v, data)
  s=null→error
  s={v}→go
  others→v=select(s)

```

其中 $data$ 是标记已完成的顶点的符号表示集， v 是待完成标记的顶点的符号表示，它的边的符号表示中含有 q 。函数 $match$ 将 v 同 $data$ 中的顶点符号一一相配，字符 q 可以对任何字符匹配成功，其他字符必须同该字符本身匹配。 $match$ 送回 $data$ 中所有可同 v 匹配的顶点的符号表示。如果送回的是空集合，即没有可匹配的完成顶点，则 v 是错误的，应当改标记。如果送回的是一个元素集合，而且这元素就是 v 本身，说明 v 在 $data$ 中，或者说 v 已被正确地完成标记。否则，从送回的顶点符号表示中选出一个，作为 v 的完成标记。这

种内涵式的表示方法使得规则数量大大减少，而且使规则与数据（data）完全独立，易于扩充和修改。

例 3. 标注下图中的顶点 A、B、C、D。（即第二章例 1）



待标记顶点可以表示为：

A: Xacb, B: Ldb, C:Xdfe, D:Yega

其中顶点类型符号 X、L、X、Y 后面跟着的符号表示构成顶点的边，每一条边的名字相当于一个变量，它的值则为该边被标记的字符。边的排列顺序遵从前面的规定。

4 个顶点涉及 abcdefg7 条边，它们的值形成 7 个字符的串，这些字符顺序地同边的名字相对应，就是本问题的状态。

显然，初始状态为 qqqqqqq，停止状态也是 7 个字符的串，但其中不含字符 q。

按照上面的讨论，充作边的值的字符除了 q 之外，应当是 i、u、+、-，其中 i、u 是有问题的。在参照固定顶点的情况下，i 和 u 分别表示进这个顶点和出这个顶点，意义是明确的。但一般地看一条边时，它有两个顶点，i 和 u 就发生歧义了。为此，需要把 i 和 u 换成顶点字符，即边界线的箭头指向哪个顶点，就标哪个顶点的字符。也就是说，在使用规则把状态字符串中的 q 改成其他字符时，需要将规则中的 i 和 u 换成对应的顶点字符；另一方面，在将状态字符串中相应字符同规则的条件进行匹配时，需要把顶点字符转换成 i 或 u。

上述转换是容易的。只要有顶点构成形式表和边的端点表：

A: Xacb, B: Ldb, C:Xdfe, D:Yega

a:AD, b:AB, c:AE, d:BC, e:CD, f:CF, g:DG

比如，如果在顶点 B 中 d 边被表示成入边 i，那么 d 在状态字符串中取值应为 B；若顶点 B 中 d 边被表示成出边 u，那么 d 在状态字符串中取值应为 C。

下面是状态空间图，每个状态除了列出状态字符串之外还列出当时正处理的顶点字符，右边列出所用的规则。状态的向右缩进表示状态空间的搜索过程深入进行，向左退回则表示搜索过程回退。

```

qqqqqqq
A:A+Eqqqq      \\Xqqq→Xiu+
  B:A+EBqqq      \\Lq+→Li+
    C:error       \\Xuqq→error
A: +-+qqqq      \\Xqqq→X++-
  B: +-+Cqqq      \\Lq-→Lu-
    C: +-+C+Fq     \\Xiqq→Xiu+
      D: +-+C+F+    \\Y+q+→Y+++
A: -+-qqqq      \\Xqqq→X--+

```

B:-+Bqqq \\Lq+→Li+
C:error \\Xuqq→error

作业:

按照本例的形式表示出第 2 章例 2 的状态空间图。

5*. 一个较复杂的实例

例 4. 称球问题。

设有若干个球，其中有一个坏球，其余都是好球。好球的重量相同，坏球同好球重量不同，但未知其轻还是重。现有一个无砝码的天平，不能称量物体的确切重量，但能比较轻重。要求找出一种称量方案，能以最少的称量次数保证找到坏球。

状态形式：(正, 轻, 重, 正轻, 正重, 轻重, 正轻重), 表示人对于球的好坏的认知情况。
其中,

1. “正”表示被认知的正常球的个数;
2. “轻”表示被认知的轻球的个数;
3. “重”表示被认知的重球的个数;
4. “正轻”表示被认知的既可能是轻球又可能是正常球但不可能是重球的个数;
5. “正重”表示被认知的既可能是重球又可能是正常球但不可能是轻球的个数;
6. “轻重”表示被认知的既可能是轻球又可能是重球但不可能是正常球的个数;
7. “正轻重”表示既可能是轻球又可能是重球还可能是正常球的个数，即完全未被认知的球的个数。

设有 k 个球,

起始状态为: (0,0,0,0,0,0,k),

停止状态为: (k-1,1,0,0,0,0,0)或(k-1,0,1,0,0,0,0)或(k-1,0,0,0,0,1,0)。

规则:

左盘状态 右盘状态 盘外状态 天平表现→全集状态

其中,

左盘状态、右盘状态分别是天平左盘和右盘中球的状态,

盘外状态是在天平之外的其他球的状态。

天平表现有三种: -, +, =, 分别表示左盘轻、左盘重、两盘平衡。

全集状态是由左盘状态、右盘状态、盘外状态、天平表现推理得知的全部球的状态。

- a. (0,0,0,0,0,0,n)(0,0,0,0,0,0,n)(x1,0,0,x4,x5,0,x7)-+ →(x1+x4+x5+x7,0,0,n,n,0,0)
b. (0,0,0,0,0,0,n)(0,0,0,0,0,0,n)(x1,0,0,x4,x5,0,x7)= →(x1+2n,0,0,x4,x5,0,x7)
c. (0,0,0,m,n,0,0)(m+n,0,0,0,0,0,0)(x1,0,0,x4,x5,0,x7)- →(x1+x4+x5+x7+m+2n,0,0,m,0,0,0)
d. (0,0,0,m,n,0,0)(m+n,0,0,0,0,0,0)(x1,0,0,x4,x5,0,x7)+ →(x1+x4+x5+x7+2m+n,0,0,0,n,0,0)
e. (0,0,0,m,n,0,0)(m+n,0,0,0,0,0,0)(x1,0,0,x4,x5,0,x7)= →(x1+2m+2n,0,0,x4,x5,0,x7)
f. (0,0,0,n,0,0,0)(0,0,0,n,0,0,0)(x1,0,0,x4,x5,0,x7)-+ →(x1+x4+x5+x7+n,0,0,n,0,0,0)
g. (0,0,0,n,0,0,0)(0,0,0,n,0,0,0)(x1,0,0,x4,x5,0,x7)= →(x1+2n,0,0,x4,x5,0,x7)
h. (0,0,0,0,n,0,0)(0,0,0,0,n,0,0)(x1,0,0,x4,x5,0,x7)-+ →(x1+x4+x5+x7+n,0,0,0,n,0,0)
i. (0,0,0,0,n,0,0)(0,0,0,0,n,0,0)(x1,0,0,x4,x5,0,x7)= →(x1+2n,0,0,x4,x5,0,x7)
j. (0,0,0,n,0,0,0)(n,0,0,0,0,0,0)(x1,0,0,x4,x5,0,x7)- →(x1+x4+x5+x7+n,0,0,n,0,0,0)

k. $(0,0,0,n,0,0,0)(n,0,0,0,0,0,0)(x1,0,0,x4,x5,0,x7) \rightarrow (x1+2n,0,0,x4,x5,0,x7)$
 l. $(0,0,0,0,n,0,0)(n,0,0,0,0,0,0)(x1,0,0,x4,x5,0,x7) \rightarrow (x1+x4+x5+x7+n,0,0,0,n,0,0)$
 m. $(0,0,0,0,n,0,0)(n,0,0,0,0,0,0)(x1,0,0,x4,x5,0,x7) \rightarrow (x1+2n,0,0,x4,x5,0,x7)$
 n. $(0,0,0,0,0,n)(n,0,0,0,0,0,0)(x1,0,0,x4,x5,0,x7) \rightarrow (x1+x4+x5+x7+n,0,0,n,0,0,0)$
 o. $(0,0,0,0,0,0,n)(n,0,0,0,0,0,0)(x1,0,0,x4,x5,0,x7) \rightarrow (x1+x4+x5+x7+n,0,0,0,n,0,0)$
 p. $(0,0,0,0,0,0,n)(n,0,0,0,0,0,0)(x1,0,0,x4,x5,0,x7) \rightarrow (x1+2n,0,0,x4,x5,0,x7)$
 q. $(x1,0,0,1,0,0,0) \rightarrow (x1,1,0,0,0,0,0)$ stop
 r. $(x1,0,0,0,1,0,0) \rightarrow (x1,0,1,0,0,0,0)$ stop
 s. $(x1,0,0,0,0,0,1) \rightarrow (x1,0,0,0,0,1,0)$ stop

状态空间的搜索（最佳搜索路径）：

全集状态，（左盘状态 右盘状态 盘外状态 天平表现）施用规则号→全集状态

其中推出符→左边的全集状态为称量之前对于全体球的认知的情况，右边为根据称量结果对于全体球的新的认知。

$(0,0,0,0,0,0,13), ((0,0,0,0,0,0,4) (0,0,0,0,0,0,4) (0,0,0,0,0,0,5) -)a \rightarrow (5,0,0,4,4,0,0)$
 $(5,0,0,4,4,0,0), ((0,0,0,3,2,0,0)(5,0,0,0,0,0,0)(0,0,0,1,2,0,0)-)c \rightarrow (10,0,0,3,0,0,0)$
 $(10,0,0,3,0,0,0), ((0,0,0,1,0,0,0)(0,0,0,1,0,0,0)(10,0,0,1,0,0,0)-)fq \rightarrow (12,1,0,0,0,0,0)$
 $(10,0,0,3,0,0,0), ((0,0,0,1,0,0,0)(0,0,0,1,0,0,0)(10,0,0,1,0,0,0)+)fq \rightarrow (12,1,0,0,0,0,0)$
 $(10,0,0,3,0,0,0), ((0,0,0,1,0,0,0)(0,0,0,1,0,0,0)(10,0,0,1,0,0,0)=)gq \rightarrow (12,1,0,0,0,0,0)$
 $(5,0,0,4,4,0,0), ((0,0,0,3,2,0,0)(5,0,0,0,0,0,0)(0,0,0,1,2,0,0)+)d \rightarrow (11,0,0,0,2,0,0)$
 $(11,0,0,0,2,0,0), ((0,0,0,0,1,0,0) (0,0,0,0,1,0,0)(11,0,0,0,0,0,0)-)hr \rightarrow (12,0,1,0,0,0,0)$
 $(11,0,0,0,2,0,0), ((0,0,0,0,1,0,0) (0,0,0,0,1,0,0)(11,0,0,0,0,0,0)+)hr \rightarrow (12,0,1,0,0,0,0)$
 $(5,0,0,4,4,0,0), ((0,0,0,3,2,0,0)(5,0,0,0,0,0,0)(0,0,0,1,2,0,0)=)e \rightarrow (10,0,0,1,2,0,0)$
 $(10,0,0,1,2,0,0), ((0,0,0,1,1,0,0)(2,0,0,0,0,0,0)(8,0,0,0,1,0,0)-)cq \rightarrow (12,1,0,0,0,0,0)$
 $(10,0,0,1,2,0,0), ((0,0,0,1,1,0,0)(2,0,0,0,0,0,0)(8,0,0,0,1,0,0)+)dr \rightarrow (12,0,1,0,0,0,0)$
 $(10,0,0,1,2,0,0), ((0,0,0,1,1,0,0)(2,0,0,0,0,0,0)(8,0,0,0,1,0,0)=)er \rightarrow (12,0,0,0,1,0,0)$
 $(0,0,0,0,0,0,13), ((0,0,0,0,0,0,4) (0,0,0,0,0,0,4) (0,0,0,0,0,0,5) +)a \rightarrow (5,0,0,4,4,0,0)$
 $(0,0,0,0,0,0,13), ((0,0,0,0,0,0,4) (0,0,0,0,0,0,4) (0,0,0,0,0,0,5) =)b \rightarrow (8,0,0,0,0,0,5)$
 $(8,0,0,0,0,0,5), ((0,0,0,0,0,0,3) (3,0,0,0,0,0,0)(5,0,0,0,0,0,2)-)n \rightarrow (10,0,0,3,0,0,0)$
 $(8,0,0,0,0,0,5), ((0,0,0,0,0,0,3) (3,0,0,0,0,0,0)(5,0,0,0,0,0,2)+)o \rightarrow (10,0,0,3,0,0,0)$
 $(10,0,0,3,0,0,0), ((0,0,0,0,1,0,0) (0,0,0,0,1,0,0)(10,0,0,0,1,0,0)-)hr \rightarrow (12,0,1,0,0,0,0)$
 $(10,0,0,3,0,0,0), ((0,0,0,0,1,0,0) (0,0,0,0,1,0,0)(10,0,0,0,1,0,0)+)hr \rightarrow (12,0,1,0,0,0,0)$
 $(10,0,0,3,0,0,0), ((0,0,0,0,1,0,0) (0,0,0,0,1,0,0)(10,0,0,0,1,0,0)=)ir \rightarrow (12,0,1,0,0,0,0)$
 $(8,0,0,0,0,0,5), ((0,0,0,0,0,0,3) (3,0,0,0,0,0,0)(5,0,0,0,0,0,2)=)p \rightarrow (11,0,0,0,0,0,2)$
 $(11,0,0,0,0,0,2), ((0,0,0,0,0,0,1)(1,0,0,0,0,0,0)(10,0,0,0,0,0,1)-)nq \rightarrow (12,1,0,0,0,0,0)$
 $(11,0,0,0,0,0,2), ((0,0,0,0,0,0,1)(1,0,0,0,0,0,0)(10,0,0,0,0,0,1)+)or \rightarrow (12,0,1,0,0,0,0)$
 $(11,0,0,0,0,0,2), ((0,0,0,0,0,0,1)(1,0,0,0,0,0,0)(10,0,0,0,0,0,1)=)ps \rightarrow (12,0,0,0,0,0,1,0)$
 可知采用该称量方案，称量 3 次必能找到坏球。

作业：

说明直线图识别问题中的状态和规则如何表示。

