第二章 知识表示方法

**1.状态空间法、问题归约法、谓词逻辑法和语义网络法的要点是什么？它们有何本质上的联系及异同点?**

答：状态空间法：基于解答空间的问题表示和求解方法，它是以状态和算符为基础来表示和求解问题的。一般用状态空间法来表示下述方法：从某个初始状态开始，每次加一个操作符，递增的建立起操作符的试验序列，直到达到目标状态为止。

问题规约法：已知问题的描述，通过一系列变换把此问题最终变成一个子问题集合：这些子问题的解可以直接得到，从而解决了初始问题。问题规约的实质：从目标（要解决的问题）出发逆向推理，建立子问题以及子问题的子问题，直至最后把出示问题规约为一个平凡的本原问题集合。

谓词逻辑法：采用谓词合式公式和一阶谓词算法。要解决的问题变为一个有待证明的问题，然后采用消解定理和消解反演莱证明一个新语句是从已知的正确语句导出的，从而证明这个新语句也是正确的。

语义网络法：是一种结构化表示方法，它由节点和弧线或链组成。节点用于表示物体、概念和状态，弧线用于表示节点间的关系。语义网络的解答是一个经过推理和匹配而得到的具有明确结果的新的语义网络。语义网络可用于表示多元关系，扩展后可以表示更复杂的问题。

**2  设有3个传教士和3个野人来到河边，打算乘一只船从右岸渡到左岸去。该船的负载能力为两人。在任何时候，如果野人人数超过传教士人数，那么野人就会把传教士吃掉。他们怎样才能用这条船安全地把所有人都渡过河去?**

答：见ppt

3．利用图2.3，用状态空间法规划一个最短的旅行路程：此旅程从城市A开始，访问其他城市不多于一次，并返回A。选择一个状态表示，表示出所求得的状态空间的节点及弧线，标出适当的代价，并指明图中从起始节点到目标节点的最佳路径。

A

B

100

D

C

E

100

125

75

50

100

50

75

125

125

E D E C D C

B C D E

A

C D E

D E C E C D

150 75 125

100 125 100 50 125 50

50 50 125 125 100 100

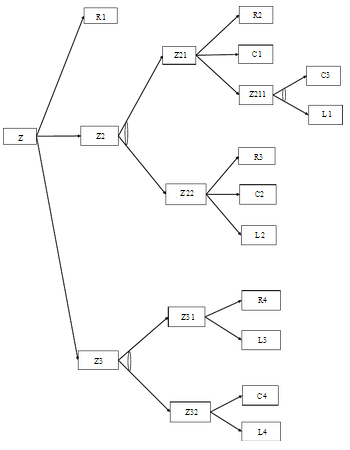
100 125 100 75

A A A A A A

75 100 75 125 100 125

4．试说明怎样把一棵与或解树用来表达图2.28所示的电网络阻抗的计算。单独的R、L或C可分别用R、jωL或1/jωC来计算，这个事实用作本原问题。后继算符应以复合并联和串联阻抗的规则为基础。

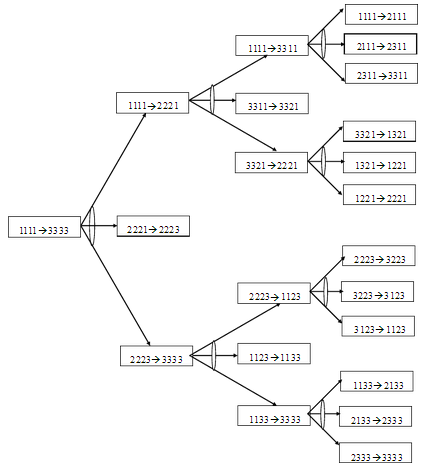
答：约定，用原来的与后继算法用来表达并联关系，用原来的或后继算法用来表达串联关系



5．试用四元数列结构表示四圆盘梵塔问题，并画出求解该问题的与或图。

用四元数列 (nA, nB, nC, nD) 来表示状态，其中nA表示A盘落在第nA号柱子上，nB表示B盘落在第nB号柱子上，nC表示C盘落在第nC号柱子上，nD表示D盘落在第nD号柱子上。

初始状态为 1111，目标状态为 3333



如图所示，按从上往下的顺序，依次处理每一个叶结点，搬动圆盘，问题得解。

6  把下列句子变换成子句形式：

(1)  (x)｛P(x)→P(x)｝ (2)  xy(On(x,y)→Above(x,y))

(3)  xyz(Above(x,y)∧Above(y,z)→Above(x,z))

(4)  ～｛(x)｛P(x)→｛（y）［p(y)→p(f(x,y))］∧(y)［Q(x,y)→P(y)］｝｝｝

答：

1. (ANY x) { P(x)→P(x) }

(ANY x) {~P(x) OR P(x)}

 ~P(x) OR P(x)

最后子句为 ~P(x) OR P(x)

(2)  (ANY x) (ANY y) { On(x,y) →Above(x,y) }

(ANY x) (ANY y) { ~On(x,y) OR Above(x,y) } ~On(x,y) OR Above(x,y) 最后子句为

~On(x,y) OR Above(x,y)

(3) (ANY x) (ANY y) (ANY z) { Above(x,y) AND Above(y,z) →Above(x,z) }

(命题联结词之优先级如下:否定→合取→析取→蕴涵→等价)

(ANY x) (ANY y) (ANY z) { ~ [ Above(x,y) AND Above(y,z) ] OR Above (x,z) } ~ [ Above(x,y) AND Above(y,z) ] OR Above (x,z)

最后子句为

~[Above(x,y), Above(y,z)] OR Above(x,z)

(4) ~{ (ANY x) { P(x) { (ANY y) [ p(y) →p(f(x,y)) ] AND (ANY y) [ Q(x,y) →P(y) ] } } }

~ { (ANY x) { ~P(x) OR { (ANY y) [ ~p(y) OR p(f(x,y)) ] AND (ANY y) [ ~Q(x,y) OR P(y) ] } } }

(EXT x) { P(x) AND { (EXT x) [ p(y) AND ~p(f(x,y)) ] OR (EXT y) [ Q(x,y) AND ~P(y) ] } }

(EXT x) { P(x) AND { (EXT w) [ p(y) AND ~p(f(w,y)) ] OR (EXT v) [ Q(x,v) AND ~P(v) ] } }

P(A) AND { [ p(y) AND ~p(f(B,y)) ] OR [ Q(A,C) AND ~P(C) ] }

P(A) AND { [ p(y) AND ~p(f(B,y)) OR Q(A,C) ] AND [ p(y) AND ~p(f(B,y)) OR ~P(C) ] } P(A) AND { { p(y), ~p(f(B,y)) } OR Q(A,C) } AND { { p(y), ~p(f(B,y)) } OR ~P(C) }

最后子句为 P(A)

{ p(x), ~p(f(B,x)) } OR Q(A,C)

{ p(y), ~p(f(B,y)) } OR ~P(C)

7  用谓词演算公式表示下列英文句子(多用而不是省用不同谓词和项。例如不要用单一的谓词字母来表示每个句子。)

A computer system is intelligent if it can perform a task which, if performed by a human, requires intelligence.

 答：

先定义基本的谓词

INTLT(x) means x is intelligent

PERFORM(x,y) means x can perform y

REQUIRE(x) means x requires intelligence

CMP(x) means x is a computer system HMN(x) means x is a human

上面的句子可以表达为 (任意x)

{ (存在t) (存在y) [ HMN(y) 合取 PERFORM(y,t) 合取 REQUIRE(t) 合取 CMP(x) 合取 PERFORM(x,t) ] → INTLT(x) }

8 把下列语句表示成语义网络描述： (1) All man are mortal.

(2) Every cloud has a silver lining.

(3) All branch managers of DEC participate in a profit-sharing plan.

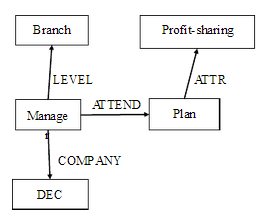
 (1)



(2)



(3)



9. 试构造一个描述你的寝室或办公室的框架系统。

