**Problem1**

State: Suppose there are totally N node. For node i, j in N, D(i,j) represents the distance between node i and node j, T(i,j) = 1 if the salesman did travel from node i to j else T(i,j) = 0.

Initial State: Any given starting point, with known distance matrix D, and T(i,j) = 0 for each pair of i,j.

Goal Test: Find a least cost path that starts and ends at the starting node and goes through each other node in the graph once and exactly once. In math, that’s minimize while

Operators: move to one adjacent node j from current node i, that’s set T(i,j) = 1

Operator Cost: The number of the label, or length.

**Problem2**

Variables: the variables are each row ri and each column ci.

Domains: List of words in the dictionary.

Constraints:

1. For each word w, if it is placed horizontally in row ri, then length of w should be equal to length of ri; else it is placed vertically, length of w should be equal to length of c*i*
2. If a row ri and a column cj are intersected, and the point is the mth index of ri and nth index of cj, then intersect(i,j) = (m,n), and r­i[m] = cj[n], means the cross over point letter should be the same.

**Problem3**

State: [Account Balance (P), Year(Y)]. Initial State is [P, 0], where P means the principle.

Terminate State: if Year = 4, stop.

Action: {Buy CD, Buy Stock}

Transition:

Reward:

Reward (s = [P,Y], Buy CD,s’=[1.1P,Y+1]) = 0.1\*P

Reward (s = [P,Y], Buy Stock, s’ = [1.3P,Y+1]) = 0.3\*P

Reward (s = [P, Y], Buy Stock, s’ = [0.9P, Y+1]) = -0.1\*P

Where P is the current balance.

Set discount factor

**Problem4**

4.1

State: the vector of input sensor, S = [s1,s2,s3,s4,s5,s6,s7,s8]

Terminate State: S = [1,1,1,1,1,1,1,1], which means the robot is blocked. Else the robot should run.

Action:{Go North, Go South, Go West, Go East}

Transition: If current action is a legal move, the state will be become the position that the robot moved. Else the action is illegal, the state remains the same. It is impossible to give a transition matrix with probability.

Reward:

Set x1 = s2\*s3 x2 = s3\*s4 x3 = s5\*s6 x4 = s7\*s8

If , Reward (s, North, s’) = 1 else Reward(s,North,s) = -1

If , Reward (s, West, s’) = 1 else Reward(s,West,s) = -1

If , Reward (s, South, s’) = 1 else Reward(s,South,s) = -1

If , Reward (s, East, s’) = 1 else Reward(s,East,s) = -1

If none of above, the robot go north in last condition, Reward(s, North, s’) = 0

Policy:

**4.2**

State: the vector of input sensor, S = [s1,s2,s3,s4,s5,s6,s7,s8]

Terminate State: S = [1,1,1,1,1,1,1,1], which means the robot is blocked. Else the robot should run.

Action:{Go North, Go South, Go West, Go East}

Policy:

Reward:

Set x1 = s2\*s3 x2 = s3\*s4 x3 = s5\*s6 x4 = s7\*s8

If , if action = Go North, reward = 1;else reward = -1

If , if action = Go West, reward = 1;else reward = -1

If , if action = Go South, reward = 1;else reward = -1

If , ,if action = Go East, reward = 1;else reward = -1

If none of above, the robot go north in last condition, Reward = 0

Then, Use Monte Carlo method to record series of states and actions:

Estimate:

**Problem5**

**图示

描述已自动生成**

Solution: S -> B -> A -> G

**Problem6**

I = min(0,7) = 0

F = max(I,5) = 5

G = max(K,L) = 8

C = min(F,G,H) = 4

B = min(D,E) = 3

A = max(B,C) = 4

**Problem7**

From left to right: Pruned Node N and L

From right to left: Pruned Node K and I