Computational Intelligence

A Logical Approach

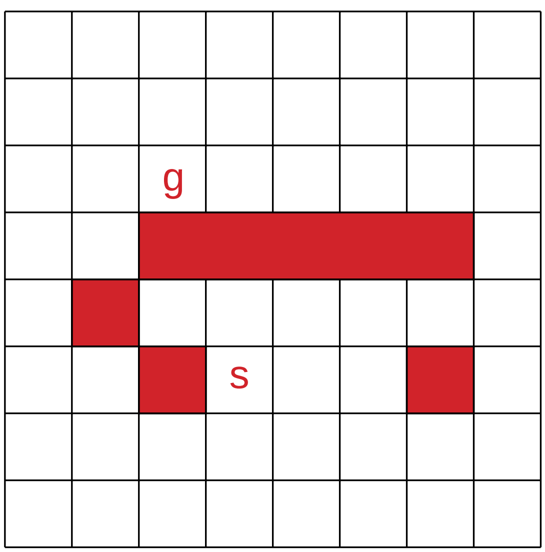
Problems for Chapter 4

Here are some problems to help you understand the material in Computational Intelligence: *A Logical Approach*. They are designed to help students understand the material and practice for exams.

This file is available in html, or in pdf format, either without solutions or with solutions. (The pdf can be read using the free acrobat reader or with recent versions of Ghostscript).

1 Finding Paths in a Grid

Consider the problem of finding a path in the grid shown below from the position s to the position g. The robot can move on the grid horizontally and vertically, one square at a time (each step has a cost of one). No step may be made into a forbidden shaded area.



(a) On the grid, number the nodes in the order in which they are removed from the frontier in a depth-first search from s to g, given that the order of the operators you will test is: up, left, right, then down. Assume there is a cycle check.

(b) Number the nodes in order in which they are taken off the frontier for an A^* search for the same graph. Manhattan distance should be used as the heuristic function. That is, $h(n)$ for any node n is the Manhattan distance from n to g. The Manhattan distance between two points is the distance in the x-direction plus the distance in the y-direction. It corresponds to the distance traveled along city streets arranged in a grid. For example, the Manhattan distance between g and s is 4. What is the path that is found by the A^* search?

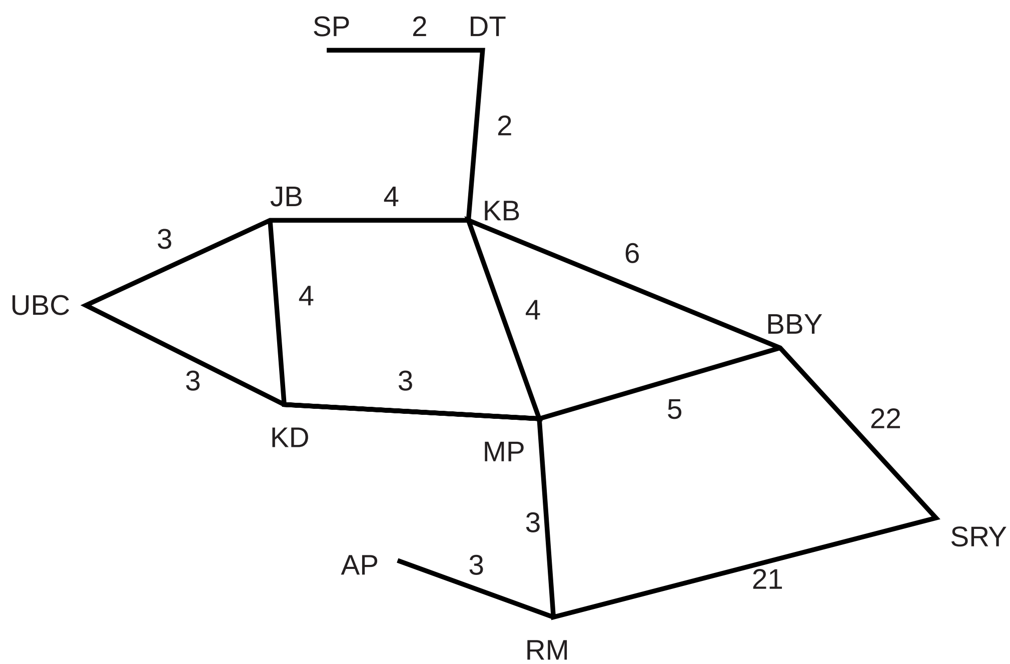
(c) Assume that you were to solve the same problem using dynamic programming. Give the *dist* value for each node, and show which path is found.

(d) Based on this experience, discuss which algorithms are best suited for this problem.

(e) Suppose that the graph extended infinitely in all directions. That is, there is no boundary, but s, g, and the forbidden area are in the same relative positions to each other. Which methods would no longer find a path? Which would be the best method, and why?

2 Searching on a simple graph

Consider the graph (not drawn to scale) with arc lengths shown on the arcs:



Suppose we have the following heuristic values for the distance to *sp*.

h(sp)=0	h(dt)=2	h(kb)=3
h(jb)=3	h(ubc)=5	h(kd)=6
h(mp)=7	h(bby)=8	h(ap)=8
h(rm)=9	h(sry)=29	

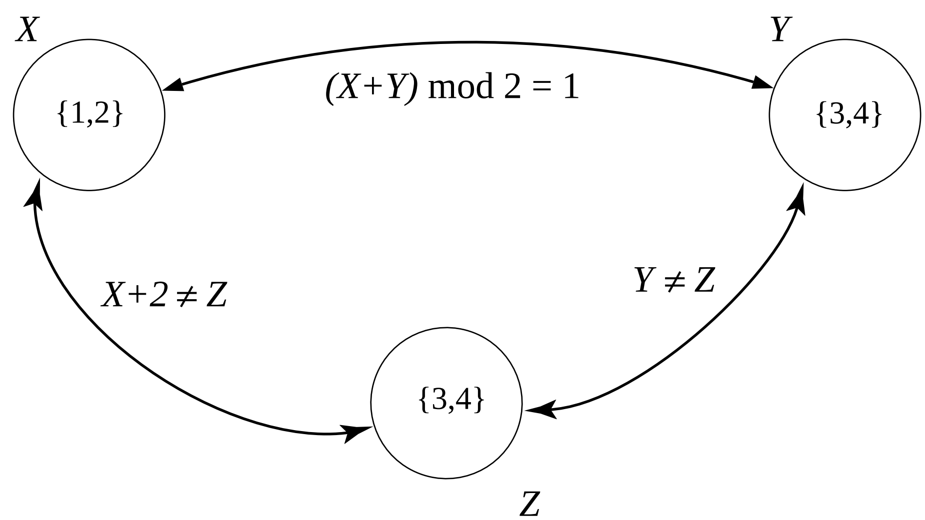
(a) Show the nodes expanded (taken off the frontier), in order, and the f-value for each node added to the frontier for an A^* search from *ubc* to *sp*. Assume that multiple-path pruning is used, and that the search stops after the first path is found. Show clearly the path found. [Explain clearly what your notation means.]

(b) Show how dynamic programming can be used to find a path from *ubc* to *sp*. Show all distance values that are computed, and how these are used to find the shortest path. What path is found?

(c) Suppose you were contacted by TecnoTaxi to advise on a method for finding routes between locations in your city. What method would you recommend, and why. Give one shortcoming of the method you propose. You must use full sentences.

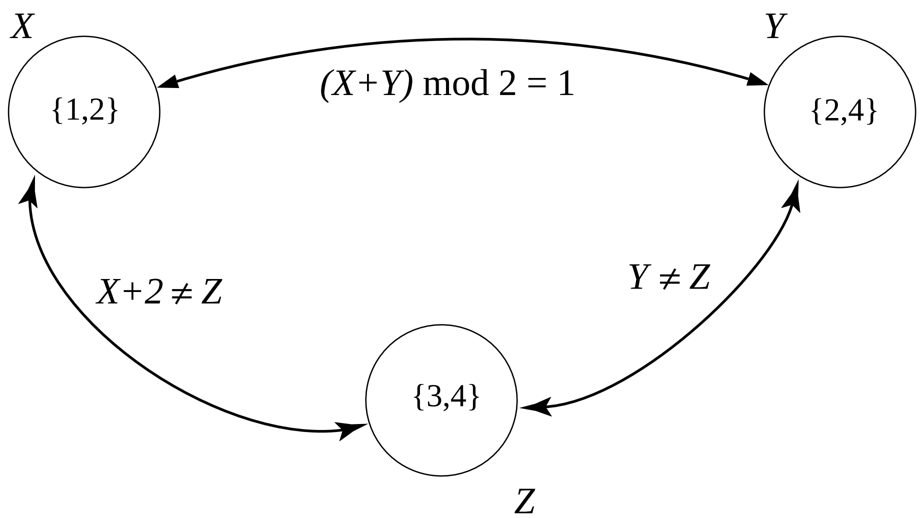
3 Arc Consistency

(a) Consider the following constraint network. Note that $(X+Y) \bmod 2 = 1$ means that $X+Y$ is odd.



Is this constraint network arc consistent? If it is, explain why. If it isn't, explain which arc is not arc consistent and why it isn't arc consistent.

(b) Consider the following constraint network:



Is this constraint network arc consistent? If it is, explain why. If it isn't, explain which arc is not arc consistent and why it isn't arc consistent.

4 Arc Consistency

Suppose you have a relation $r(X, Y)$ that is true of there is a word in the word list below with first letter X and second letter Y.

The word list is:

add	arc	bad	bud
cup	dip	fad	odd

Suppose the domain of X is $\{a, b, c, d\}$ and that of Y is $\{a, d, i, r\}$.

(a) Is the arc $\langle X, Y\rangle$ arc consistent? If so, explain why. If not, show what element(s) can be removed from a domain to make it arc consistent.

(b) Is the arc $\langle Y, X\rangle$ arc consistent? If so, explain why. If not, show what element(s) can be removed from a domain to make it arc consistent.

5 Solving a CSP via backtracking, arc consistency, hillclimbing

In this question you will look at backtracking, arc consistency, and hill climbing for solving the same CSP problem.

Consider a scheduling problem, where there are five variables A, B, C, D, and E, each with domain $\{1, 2, 3, 4\}$. Suppose the constraints are: $E - A$ is even, $C \neq D$, $C > E$, $C \neq A$, $B > D$, $D > E$,$B > C$.

(a) Show how backtracking can be used to solve this problem, using the variable ordering A, B, C, D, E. To do this you should draw the search tree generated to find all answers. Indicate clearly the satisfying assignments.

To indicate the search tree, write it in text form with each branch on one line. For example, suppose we had variables X, Y and Z with domains t, f, and constraints $X \neq Y$, $Y \neq Z$. The corresponding search tree can be written as:

```
X=t Y=t failure
    Y=f Z=t solution
        Z=f failure
X=f Y=t Z=t failure
        Z=f solution
    Y=f failure
```

Hint: the easiest way to solve this problem is to write a program to generate the tree (using whatever programming language you like).

(b) Is there a different variable ordering that results in a smaller tree? Give the variable ordering that results in the smallest tree (or a small tree). Explain how you determined this was the optimal ordering. (E.g., what was the search strategy through the space of orderings that you used to solve this. A good explanation as to why your ordering is good is more important than the right answer.)

(c) Show how arc consistency can be used to solve this problem. To do this you need to

i) Draw the constraint graph,

ii) Show which elements of a domain are deleted at each step, and which arc is responsible for removing the element.

iii) Show explicitly the constraint graph after arc consistency has stopped.

iv) Show how splitting domains can be used to solve this problem. Include all arc consistency steps.

(d) Show how hill climbing can be used for the problem. Suppose a neighbor is obtained by increasing or decreasing the value of one of the variables by 1, the heuristic function to be maximized is the number of satisfied constraints, and you always choose a neighbor with the maximal heuristic value.

i) Show what happens when we start with the assignment $A = 1, B = 1, C = 1, D = 1, E = 1$.

ii) Show what happens when we start with $A = 3$, $B = 3$, $C = 2$, $D = 1$, $E = 4$.

iii) Can you think of a better heuristic function? Explain why or why not.