CS360 Homework 12

Constraint Satisfaction

1) Consider the following constraint satisfaction problem with variables x, y and z, each with domain $\{1, 2, 3\}$, and constraints C_1 and C_2, defined as follows:

- C_1 is defined between x and y and allows the pairs $(1, 1), (2, 2), (3, 1), (3, 2), (3, 3)$. (A pair (a, b) means that assigning $x = a$ and $y = b$ does not violate the constraint. Any assignment that does not appear in the list violate the constraint.)
- C_2 is defined between y and z and allows the pairs $(1, 1), (1, 2), (3, 1), (3, 2), (3, 3)$.

Which values does arc consistency rule out from the domain of each variable? Suppose that we started search after establishing arc consistency, and we assign $x = 1$. Which values does a) forward checking and b) maintaining arc consistency rule out from the domain of each variable?

2) Show how a single ternary constraint such as "A + B = C" can be turned into three binary constraints by using an auxiliary variable. You may assume finite domains. (Hint: Consider a new variable that takes on values that are pairs of other values, and consider constraints such as "X is the first element of the pair Y.") Next, show how constraints with more than three variables can be treated similarly. Finally, show how unary constraints can be eliminated by altering the domains of variables. This completes the demonstration that any CSP can be transformed into a CSP with only binary constraints.

Search

3) Solve Problem 1 from Homework 11 for A* with a) the (consistent) Manhattan distance heuristic and b) the straight line distance heuristic. The Manhattan distance heuristic between two grid cells (x_1, y_1) and (x_2, y_2) is $|x_1 - x_2| + |y_1 - y_2|$ (the length of a shortest path between the two cells, assuming that there are no obstacles on the grid). For instance, the Manhattan distance between A1 and E3 is $|1 - 5| + |1 - 3| = 6$. Remember to expand every state at most once.

4) We are given a sequence of integers and want to sort them in ascending order. The only operation available to us is to reverse the order of the elements in some prefix of the sequence. For instance, by reversing the first three elements of (1 2 3 4), we get (3 2 1 4). This problem is also known as the "pancake flipping" problem. We model this problem as a search problem, where each state corresponds to a different ordering of the elements in the sequence. Given an initial sequence (2 4 1 3), in which order does A* expand the states, using the breakpoint heuristic described below? Assume that ties are broken toward states with larger g-values, and, if there are still ties, they are broken in lexicographic order. That is, (2 1 4 3) is preferred to (2 4 1 3).

Breakpoint heuristic: A breakpoint exists between two consecutive integers if their difference is more than one. Additionally, a breakpoint exists after the last integer in the sequence if it is not the largest integer in the sequence. For instance, in (2 1 4 3), there are two breakpoints: one between 1 and 4 (since their difference is more than 1), the other after 3 (since it is at the end and of the sequence and is not the largest integer in the sequence). The breakpoint heuristic is the number of breakpoints in a given sequence. (Bonus question: Is this heuristic a) admissible and b) consistent? Why?)

5) Does A* always terminate if a finite-cost path exists? Why?

6) Given two consistent heuristics h_1 and h_2, we compute a new heuristic h_3 by taking the maximum of h_1 and h_2. That is, $h_3(s) = max(h_1(s), h_2(s))$. Is h_3 consistent? Why?

7) In the arrow puzzle, we have a series of arrows pointing up or down, and we are trying to make all the arrows point up with a minimum number of action executions. The only action available to us is to chose a pair of adjacent arrows and flip both of their directions. Using problem relaxation, come up with a good heuristic for this problem.

8) Explain why heuristics obtained via problem relaxation are not only admissible but also consistent.

9) What are the advantages and disadvantages of a) uniform-cost search and b) greedy best-first search over A* search?

10) Give a simple example that shows that A* with inconsistent but admissible heuristic values is not guaranteed to find a cost-minimal path if it expands every state at most once.

11) Solve Problem 1 from Homework 11 for Iterative Deepening search.