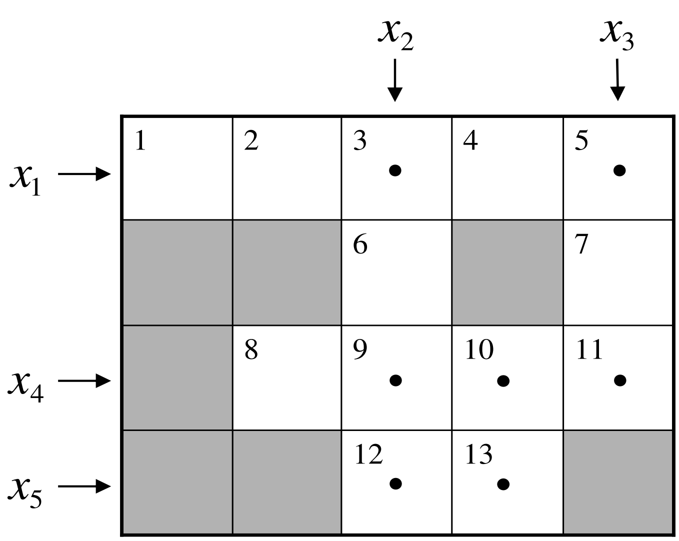
Rina Dechter, Fall-2022

ICS 275, Assignment 4

Read chapter 5 in the textbook and answer the following questions:

1. (20 pts. question 4 chapter 5) Consider The crossword puzzle formulated using the dual-graph representation. Using the ordering $x_1, x_2, x_5, x_6, x_4, x_3$ show a trace, (and data structure) whose length is limited by a 1 page description, for each algorithm. You should show explicitly the data structure you use for each level of the search tree and how it is updated at each level.



P_{26}

   (a) Forward-checking

   (b) Using graph parameters, bound the complexity of partial look-ahead on constraint problems whose graph is identical to the crossword puzzle's graph.

   (c) (optional) Formulate the problem in minizinc and run with different optional solvers or heuristics available. Describe what were your choices and what worked best.
   (Hint:https://www.minizinc.org/doc-2.5.5/en/mzn_search.html#finite-domain-search Here is the documentation link that will allow you to specify how domain values are chosen, as well as different methods for determining variable orderings including specifying your own variable ordering as well)

2. (30 pts. question 7, chapter 5) Trace the following algorithms for the 6-queens problem. Show the search tree and data structures generated for finding all solutions (no more than one page per algorithm).

   (a) Backtracking

   (b) Dynamic variable ordering, DVFC.

   (c) Arc-consistency look-ahead.

3. (10 pts. question 9, chapter 5) We say that a problem is backtrack-free at level m of the search if a backtracking algorithm, once it reaches this level is guaranteed to be backtrack-free. Assume we apply DPC as a look-ahead propagation at each level of the search.

   (a) Can you identify a level for which DPC is backtrack-free?

   (b) Can you bound the complexity of DPC as a function of a width-2 cutset (a cutset is a width-2 if its removal from the graph yields a graph having induced-width of 2.)

4. (15 pts.) Langford Problem: L(k,n): Arrange k copies of the n digits $1, \ldots, n$ such that there is one digit between the 1s, two digits between the 2s, three digits between the 3s and so on. For example, the solution of $L(2,3)$ is 231213 and the solution of $L(2,4)$ is 23421314. n is also called as the order of the Langford problem.

   (a) (5 pts) Provide one way of encoding the Langford problem as a CSP.

   (b) (5 pts) Suggest a scheme to transform your CSP encoding into a SAT formula.

   (c) (5 pts) What are the pros and cons of modeling the Langford problem as a SAT/CSP.

   (d) (10 pts.) Formulate the Langford problem in minizinc and run with different optional solvers or heuristics available. Describe what were your choices and what worked best.
   (Hint:`https://www.minizinc.org/doc-2.5.5/en/mzn_search.html#finite-domain-search` Here is the documentation link that will allow you to specify how domain values are chosen, as well as different methods for determining variable orderings including specifying your own variable ordering as well)