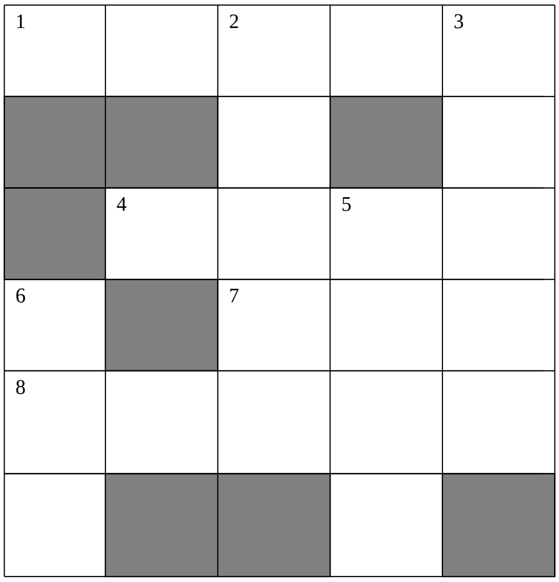
Rina Dechter, Fall-01

ICS 275A, Assignment 4

1. Read chapters 5 and provide a short review answering the following questions.
   - Is the chapter clear? readable?
   - Is the chapter technically difficult?
   - Enough examples and illustrations?
   - Any typos? technical inaccuracies?
   - Please provide an overall evaluation

2. 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 of the algorithm:.



Word List

aft	laser
ale	lee
eel	line
heel	sails
hike	sheet
hoses	steer
keel	tie
knot	

   (a) forward-checking

   (b) DVFC

   (c) Partial look-ahead.

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

3. Apply the following algorithms to the 6-queens problem. Show the search space and data structures generated for finding all solutions (no more than one page per algorithm).

(a) backmarking

(b) Forward-checking

(c) Dynamic variable ordering, DVFC.

(d) arc-consistency look-ahead.

4. Analyze and compare the complexity overhead of each of the following algorithms along a path of the search tree: backtracking, backmarking, forward-checking, partial look-ahead arc-consistency look-ahead (assume that you keep a table of current domains at each level of the search tree.) Compare with the overhead of a state generation.

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 an graph having induced-width of 2.)

6. (a possible project.) Implement DVO algorithms for 3 levels of look-ahead: forward-checking, arc-consistency and partial look-ahead. Run these algorithms on various random and real benchmarks and compute three performance measures: the size of the search space (or number of dead-ends), the number of consistency checks, and CPU time. Present the comparison of your experiments.

7. (Optional) DPLL algorithm for propositional satisfiability is applying unit resolution at each node of the tree. It also dynamically chooses unit variables that are associated with a unit clause. Can you define a general look-ahead for constraints that have exactly the same level of look-ahead?

8. (Extra credit) Extend algorithm forward-checking (FC) and arc-consistency look-ahead (ACLH) to exploit the generalized arc-consistency definition.