Problem Solving (Second Partial Test)

January 16, 2023, Classroom Q3/0003, 15:30–17:30

No notes, no communication.
The **number between brackets** [ ] indicate the points of the exercise.
Write the answers to questions 1–3 and to questions 4–6 in **a different piece of paper**.
Always **justify your answer** (JYA); otherwise, it will not be considered.
Do not forget **to write you name** and coordinates (row and column) in **every** page.

1. In a two-player zero-sum game, the initial state is A and is the turn of player 1. The succesor states are as follows: $succ(A) = (B, C, D)$ by actions a_1, a_2, a_3; $succ(B) = (E, F, G)$ by actions b_1, b_2, b_3; $succ(C) = (H, I, J)$ by actions c_1, c_2, c_3; $succ(D) = (K, L, M)$ by actions d_1, d_2, d_3. The evaluation function $eval$ applied to states of the second level is as follows:

   $eval(E) = 0.8, eval(F) = 0.6, eval(G) = 0.4$
   $eval(H) = 0.2, eval(I) = 0.3, eval(J) = 0.4$
   $eval(K) = 0.5, eval(L) = 0.1, eval(M) = 0.6$

   (a) **[1]** If A is a MAX node and B, C, D are MIN nodes, develop the game tree and show how minimax propagates the evaluation values, identifying the best action for player 1 in A.

   (b) **[1]** Explain how to apply alpha-beta prunning and show the pruned branches.

2. For a particular zero-sum game successors are generated in *good order*, that is, the successors s' of a state s are generated in increasing or decreasing order of the evaluation function applied to those states, depending whether s is a MAX node or a MIN node. Then, you could develop a new version of the MINIMAX algorithm as follows: from a state s, you generate a single successor s'_1, the one with maximum value of the eval function if s is a MAX node, or with minimum value of the evaluation function if s is a MIN node. Formally:

   $succ(s) = (s'_1, s'_2, \ldots, s'_k)$; $succ(s)$ is an ordered set
   $eval(s'_1) \geq eval(s'_2) \geq \ldots \geq eval(s'_k)$ if s is a MAX node
   $eval(s'_1) \leq eval(s'_2) \leq \ldots \leq eval(s'_k)$ if s is a MIN node

   In this way, you may produce thinner trees with can be searched deeper.

   (a) **[1]** Does this new version of MINIMAX produce optimal moves? JYA.

   (b) **[1]** How alpha-beta prunning will occur in this schema? JYA.

3. **[1]** The idea of *singular extensions* was developed by the Deep Blue team and contributed substantially to its winning result in the chess tournament against Kasparov. Explain what a singular extension is and include an example.

4. **[2]** Translate into a CNF formula the constraint $X_1 + \overline{X_2} + X_3 + X_4 \leq 2$ using **ONE** of the methods explained in the course at your convenience (sequential counters, bubble-sort sorting networks or merge-sort sorting networks). Indicate the chosen method.

5. **[2]** Draw a sequential counter, bubble-sort network, and a merge-sort network, all of them with input x_1, x_2, x_3, x_4 and outputs y_1, y_2, y_3, y_4.

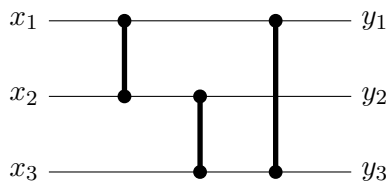
   (a) Indicate how many AND and OR gates has each one of them.

   (b) If you use them to encode $X_1 + \overline{X_2} + X_3 + X_4 \leq 2$, which of the outputs will you use? How many AND and OR gates will have the resulting circuit in each case (after removing useless gates)?

   (c) Is it correct to encode $\overline{X_1} + X_2 + \overline{X_3} + \overline{X_4} \geq 2$ instead of $X_1 + \overline{X_2} + X_3 + X_4 \leq 2$? In this case, which output of the circuit will you use, and how many gates will you obtain for each one of the three methods?

6. **[1]** We want to use a SAT solver to verify if a given sorting network is correct. Given a sorting network, explain how to construct a CNF formula whose models are counter-examples showing that the network is incorrect.

   Compute the formula to test if the following sorting network is correct:



   This network is unable to sort correctly the sequence $(1, 1, 0)$. Can you give the values of the variables in the model computed by the SAT solver for the formula?