CS206 Mid-Semster Examination

Max marks: 50 **Time: 3 hours**

- *Be brief, complete and stick to what has been asked.*
- *If needed, you may cite results/proofs covered in class without reproducing them.*
- ***If you need to make any assumptions, state them clearly.***
- ***Do not copy solutions from others or indulge in unfair means.***

1. *[10 + 10 marks]* In this question, we wish to use a propositional satisfiability solver (henceforth, called a SAT solver) to help us win a board game, whenever possible. The game is as described below.

   We have a 3×3 board, divided into 9 squares as shown in Figure 1. In this figure, squares on the board are numbered $(0, 0)$ through $(2, 2)$. The game is to be played between you and an opponent. You are required to insert A's in the squares of the board, and your opponent must insert B's, one at a time, and in alternation.

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

Figure 1:

   Suppose you start the game by first inserting an A in one of the 9 squares. Your opponent then inserts a B in one of the remaining 8 squares. You are now allowed to insert an A only in those remaining squares that cannot be reached by a diagonal from the square in which a B has been placed by your opponent. If you cannot find such an empty square, you lose the game. Otherwise, you can insert an A in an empty square not reachable along a diagonal from the square containing B. Your opponent then inserts a B in one of the remaining 6 squares, and you are again required to insert an A in one of the remaining 5 squares that cannot be reached along a diagonal from any of the squares containing B. If you can find such a square and insert the third A in that square, you win the game. Otherwise, you lose the game. Thus, if you start the game, your goal at each step is to insert an A in one of the unfilled squares that is not reachable along a diagonal from any of the squares *already containing a B*. You win the game in this case if you can insert three A's in this way.

   If your opponent starts the game, however, by first filling in a B in one of the 9 squares, then your goal at each step is to insert an A in one of the unfilled squares such that your opponent eventually can't insert three B's, none of which are reachable along a diagonal from previously inserted A's. In this case, you win the game, if you can prevent the opponent from inserting three B's.

   We wish to solve the problem of determining your move in the above board game using propositional logic. For this purpose, we will use 27 propositions named $a_{i,j}$ and $b_{i,j}$ and $m_{i,j}$, where $i, j \in \{0, 1, 2\}$. We wish to associate the following meanings with these propositions:

   - $a_{i,j}$ is True iff an A has already been inserted in the i^{th} row and j^{th} column of the board in an earlier step.
   - $b_{i,j}$ is True iff your opponent has already inserted a B in the i^{th} row and j^{th} column of the board in an earlier move.
   - If $m_{i,j}$ is True, then in the current move, you can place an A in the i^{th} row and j^{th} column to eventually win the game.

   (a) Give a propositional logic formula, ϕ_1 using the above propositions such that if you start the game, and wish to determine a move that will eventually lead you to a win (for all possible moves of the opponent), you can use the following procedure.

   - Simplify ϕ_1 by assigning True and False values for $a_{i,j}$ and $b_{i,j}$ for depending on previous moves made by you and the opponent.

- Feed the simplified formula to a SAT solver
- If the SAT solver returns a satisfying assignment, you place an A in the $(i,j)^{th}$ square, if $m_{i,j}$ is set to True in the satisfying assignment. Note that there could be several choices of (i,j) from the satisfying assignment, and you can pick one at random.
- If the SAT solver says that the simplified formula is unsatisfiable, then you give up and declare yourself to be a loser.

(b) Using only the above propositions, is it possible to give a propositional logic formula, ϕ_2 that is to be used exactly as in the previous subquestion to determine a move that will eventually lead you to a win, but only if your opponent starts the game. Give justification if your answer is in the negative. Otherwise, describe how would obtain the required formula ϕ_2.

2. *[10 marks]* You are given (i) a set of propositional clauses $\mathcal{C} = \{C_1, C_2, \ldots C_r\}$ on the propositions $\{p_1, \ldots p_n\}$, and (ii) a fraction f such that $0 \leq f \leq 0.5$.

   Design an algorithm that takes the above two inputs and returns YES if there is *at least one assignment of truth values* to $\{p_1, \ldots p_n\}$ that makes $f.r$ or more clauses in $\mathcal{C}$ evaluate to True. The algorithm must return NO if there no such assignment of truth values to $\{p_1, \ldots p_n\}$.

   Your algorithm must have no more than $O(n)$ complexity, where n is the number of distinct propositions.

3. *[10 marks]* Let $\mathcal{C} = \{C_1, C_2, \ldots C_r\}$ be a set of propositional clauses on the propositions $\{p_1, \ldots p_n\}$. Let $\phi_{\mathcal{C}}$ denote the CNF formula $\bigwedge_{i=1}^{r} C_i$.

   You are given an algorithm `RandSAT` that takes as argument a set $\mathcal{C}$ of propositional clauses, and returns a *randomly chosen* satisfying assignment of $\phi_{\mathcal{C}}$, if $\phi_{\mathcal{C}}$ is satisfiable. Otherwise, `RandSAT` prints out `NOT POSSIBLE`. Since the satisfying assignment returned by `RandSAT` is random, two invokations to `RandSAT` with the same set $\mathcal{C}$ of propositional clauses can potentially return two different satisfying assignments of $\phi_{\mathcal{C}}$, if $\phi_{\mathcal{C}}$ is satisfiable.

   Given $\mathcal{C}$, we now wish to use `RandSAT` as a black-box to check whether there are at least four satisfying assignments of $\phi_{\mathcal{C}}$ that differ only in the truth assignments of propositions p_1 and p_2.

   Describe (in pseudo-code form) how you would design such an algorithm that invokes `RandSAT` exactly once and uses its return value to answer the above question.

4. *[2.5 × 4 marks]* In this question, we wish to express statements about a directed acyclic graph (DAG) using predicate logic. Thus, our domain (or universe) is the set of all nodes in a DAG.

   Consider the following predicates and their interpretations

Predicate	Arity	Evaluates to True iff ...
$Edge(x, y)$	2	there is a directed edge from x to y
$Path(x, y)$	3	there is a directed path of at least one edge from x to y

   Using the above predicates, express the following properties about DAGs in predicate logic.

   (a) Whenever there is a path from one node to another, there is at least one edge along the path.

   (b) There is a node that is part of two distinct cycles, one of which has exactly two nodes on it.

   (c) There are no more than five rooted trees in the DAG.

   (d) The DAG is a rooted tree.