Revision : Week 12.   Due on Nov 13th, 2024

Instructions : The usual ones ... ☺

(1) Draw the truth table of the following boolean function and write a formula for it in Conjunctive Normal Form.

$XOR : \{0,1\}^3 \to \{0,1\}$ defined as

$XOR(x, y, z) = 1$ iff exactly one of $x, y, z = 1$.

[4 + 3]

(2) Modify the algorithm from Lecture 23 (that showed that all CFLs are in P) to give a polynomial time algorithm that takes as input a CFG and a string and outputs a parse tree for it if the grammar generates the string, else outputs nothing.

[8].

(3) This problem investigates "RESOLUTION", a method for proving the unsatisfiability of CNF-formulas.

Recall the following definitions:

i) Literals are variables or their negation

ii) Clauses are ORs of literals

iii) CNFs are ANDs of clauses

Let $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_m$ be a formula in CNF, where the C_is are clauses.

Define $\mathcal{C} = \{ C : C \text{ is a clause in } \phi \}$

In a resolution step, we take two clauses C_a and C_b in $\mathcal{C}$ which have some variable, say x, appearing positively in one and negatively in another.

For example, say

$C_a = x \vee y_1 \vee \dots \vee y_k$ and

$C_b = \overline{x} \vee z_1 \vee \dots \vee z_\ell$.

We then form a new clause

$$y_1 \vee \cdots \vee y_k \vee z_1 \vee \cdots \vee z_\ell$$

and remove the repeated literals. The resulting clause, say c, is added to the set of clauses $\mathcal{C}$.

We repeat this step until no new clauses can be added to $\mathcal{C}$. At this point if the empty clause is in $\mathcal{C}$, then we declare that ϕ is unsatisfiable.

i) Prove that the above algorithm does not declare a satisfiable formula as unsatisfiable. [5]

ii) Prove that the above algorithm declares every unsatisfiable formula as unsatisfiable. [5]

iii) Show that the above algorithm is efficient if every clause is an OR of exactly two literals. [5]