Problem Set : Week 13 Due: 5 pm on 14/11/25

Instructions

- Discussion is allowed and infact encouraged
- Answers must be written by yousef.
- All sources that are used to reach the solution must be mentioned.

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

Recall the following definitions:

- Literals are variables or their negation
- Clauses are ORs of literals
- 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

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

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

We then form a new clause

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

and simplify it by removing repeated literals and literals & their negation if both are present in the same clause.

We then add this clause to $\mathcal{C}$.

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]