**Assignment 2** (*Marks: 7, Due date: Sept. 21*) **COL703, Aug-Nov 2022**

---

All questions in this assignment carry marks.

Please note that there will be zero tolerance for dishonest means like copying solutions from others, and even letting others copy your solution, in any assignment/quiz/exam. If you are found indulging in such an activity, your answer-paper/code will not be evaluated and your participation/submission will not be counted. Second-time offenders will be summarily awarded an F grade. The onus will be on the supposed offender to prove his or her innocence.

---

1. **[1 marks]** Let $X \subseteq \Phi$ and $\alpha \in \Phi$. Prove that $X \models \alpha$ *iff* $X \vdash \alpha$.

2. Let $X$ be a set of formulas. $X$ is said to be a *finitely satisfiable set (FSS)* if every $Y \subseteq_{fin} X$ is satisfiable.

   X is sat

   Equivalently, $X$ is an FSS if there is no finite subset $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ of $X$ such that $\neg(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n)$ is valid.

   (Note that if $X$ is an FSS we are not promised a single valuation $v$ which satisfies every finite subset of $X$. Each finite subset could be satisfied by a *different* valuation.)

   Show that:

   (a) **[0.5 marks]** Every FSS can be extended to a maximal FSS.

   (b) **[0.5 marks]** If $X$ is a maximal FSS then for every formula $\alpha$, $\alpha \in X$ *iff* $\neg\alpha \notin X$.

   (c) **[0.5 marks]** If $X$ is a maximal FSS then for all formulas $\alpha, \beta$, $(\alpha \vee \beta) \in X$ *iff* $(\alpha \in X$ *or* $\beta \in X)$.

   (d) **[1 marks]** Every maximal FSS $X$ generates a valuation $v_X$ such that for every formula $\alpha$, $v_X \models \alpha$ *iff* $\alpha \in X$.

   From these facts, conclude that:

   (e) **[0.5 marks]** Any FSS $X$ is simultaneously satisfiable (that is, for any FSS $X$, there exists $v_X$ such that $v_X \models X$).

   (f) **[1 marks]** For all $X$ and all $\alpha$, $X \models \alpha$ *iff* there exists $Y \subseteq_{fin} X$ such that $Y \models \alpha$.

3. **[2 marks]** Recall the coding exercise from Assignment 1. Assume that we only have unsatisfiable formulas as inputs, in our *formula* files. Suppose we modify the *proof* file of our sample input (from Assignment 1) as follows:

```
3f 4f
4f 5f
1p 2p
6f 3p
```

   This modified proof, as you can see, only contains the *clause-ids* in each line. We can check the correctness of this *proof*, under the assumption that once the clauses (line numbers) have been identified, the corresponding resolvent is correct and unique. For uniqueness, assume that if there are more than one pair of complementary literal, resolution is done by picking the pair corresponding to the smaller atom, i.e. the pair (1, -1) should be used before (2, -2), and so on.

   Correctness of such a proof essentially depends on three things:

- the clauses indicated by the clause ids in each line of proof file indeed contains (at least) a pair of complementary literals,
- the clauses of the last line of the proof file must give an empty clause as resolvent, *and*
- the clause ids in each line in the proof file only refers to clauses that are in the formula file, or already discovered clauses in the proof file.

Here's what you have to do. Given a formula and a modified proof file (like the one shown above) but with holes (denoted by "??"), you need to output a complete proof file (like the sample proof file shown in Assignment 1). Write a Python code for this task. You are guaranteed that there will be at most one hole (in place of only one of the clause ids) in every line of the input proof. If the proof with holes cannot be completed anyhow, return the proof file with every "??" replaced with "np".

Here's a sample *formula*:

```
c CNF formula (p1 ∨ !p2) ∧ (p2 ∨ p3) ∧ (!p1 ∨ !p2 ∨ p3) ∧ (!p3)
p cnf 3 4
1 -2 0
2 3 0
-1 -2 3 0
-3 0
```

And, here is a sample *proof* with holes:

```
3f ??
??  5f
1p ??
??  3p
```

And here is the sample output:

```
3f 4f 1 3 0
4f 5f -1 3 0
1p 2p 3 0
6f 3p 0
```

*(This is not a part of the assignment, but note that if you are able to output a complete proof, you can verify the correctness of your output using the code that you had submitted for Assignment 1, assuming that the code was correct!)*