Due: Tuesday, October 11, 11:59pm

1. (12 points) In class, we discussed the DLL procedure for finding Resolution refutations. This procedure was inherently bottom-up in that we started at the root, and worked our way to the leaves (initial clauses). Another procedure for finding Resolution refutations is top-down and is called the Davis-Putnam (DP) procedure. The procedure is as follows. First, order the variables $x_1, ..., x_n$. Let C_0 be the original set of clauses. Apply all possible resolution steps to the initial clauses where we resolve only on the variable x_1. Let C_1 be all clauses obtained so far (including the original clauses) and not containing the literals x_1 or $\neg x_1$. Now apply all possible resolution steps to clauses in C_1 where now we resolve only on the variable x_2, and let C_2 be the resulting set of all clauses obtained from C_1 (including clauses in C_1) and not containing the literals x_2 or $\neg x_2$. Continue in this fashion until we have either derived the empty clause, or until we have used up all of our variables Note that the clauses in C_i will involve only the variables $x_{i+1}, ..., x_n$.

   (a) Use the above DP procedure to obtain a refutation of the following formula.

   $$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_2) \wedge (\neg x_2 \vee \neg x_3)$$

   (b) Show that the DP procedure, when applied to 2CNF formulas always terminates in polynomial time.

   (c) Prove completeness of the DP procedure. That is, show that for every unsatisfiable CNF formula f, there exists a DP refutation of f.

2. (10 points) Recall that a tree-like Resolution refutation of an unsatisfiable CNF formula f is a Resolution refutation where each derived clause is used at most once. Prove that tree-like Resolution refutations are closed under restrictions. Let $f(x_1, \ldots, x_n)$ be an unsatisfiable CNF formula. Let S be a subset of the underlying variables, and let ρ be a partial truth assignment that sets each variable in S to either 0 or 1. Prove that if T is tree-like Resolution refutation of F of size s, then we can apply the restriction ρ to T to obtain a tree-like Resolution refutation of $F|_\rho$ of size at most s.

3. (5 points) Let A and B be propositional formulas, and let S be the the set of propositional variables/atoms that occur in both A and B. Prove that if $A \rightarrow B$ is valid, then there is a formula C involving on the variables from S such that $A \rightarrow C$ and $C \rightarrow B$.

4. (10 points) Exercise 14, page 17 of notes on propositional calculus.

5. (5 points) Exercise 5, page 25 of notes. (Show that $\forall x(gfx = x)$ is not a logical consequence of $\forall x(fgx = x)$.)

6. (5 points) Let $\Phi = \{A_1, A_2, ...\}$ be an infinite set of sentences. Suppose that for all n, A_{n+1} is not a logical consequence of $\{A_1, ..., A_n\}$. Now let B be any sentence such that $\Phi \models B$. Prove that there exists n such that A_n is not a logical consequence of B.

7. (5 points) Exercise 10, page 25 of notes on Predicate Calculus

8. (**Extra Credit**) The pigeonhole principle, PHP_n^{n+1} asserts that $n+1$ pigeons cannot be mapped in a one-to-one way to n holes. The negation of the propositional principle, $\neg PHP_n^{n+1}$ is a CNF formula with underlying variables $P_{i,j}$ for $i \leq n+1$ and $j \leq n$. $P_{i,j}$ is intended to represent whether or not pigeon i is mapped to hole j. The clauses of $\neg PHP_n^{n+1}$ are of two types: First, for every $i \leq n+1$ there are pigeon clauses $\mathcal{P}_i$: $(P_{i,1} \vee P_{i,2} \vee \cdots \vee P_{i,n})$ stating that each pigeon goes to at least one hole. Secondly, for every $i1, i2 \leq n+1$, $i1 \neq i2$ and $j \leq n$, there are hole clauses $\mathcal{H}_{i1,i2,j}$: $(\neg P_{i1,j} \vee \neg P_{i2,j})$ stating that each hole has at most one pigeon mapped to it.

   (a) Prove that for n sufficiently large, any DPLL refutation of $\neg PHP_n^{n+1}$ requires size $2^{O(n)}$.

   Hint: Recall the proof of completeness for Resolution discussed in class. In the proof, we showed that a tree-like Resolution refutation of a CNF formula $f(x_1, \ldots, x_n)$ can be viewed as a decision tree that queries the variables $x_1, \ldots, x_n$ of f, and such that each leaf node l of the decision tree is labelled with a clause from f that is falsified by the partial assignment to that leaf node.

   (b) Prove the stronger lower bound of $2^{\Omega(n \log n)}$ on the size of any tree-like Resolution refutation of PHP_n^{n+1}.