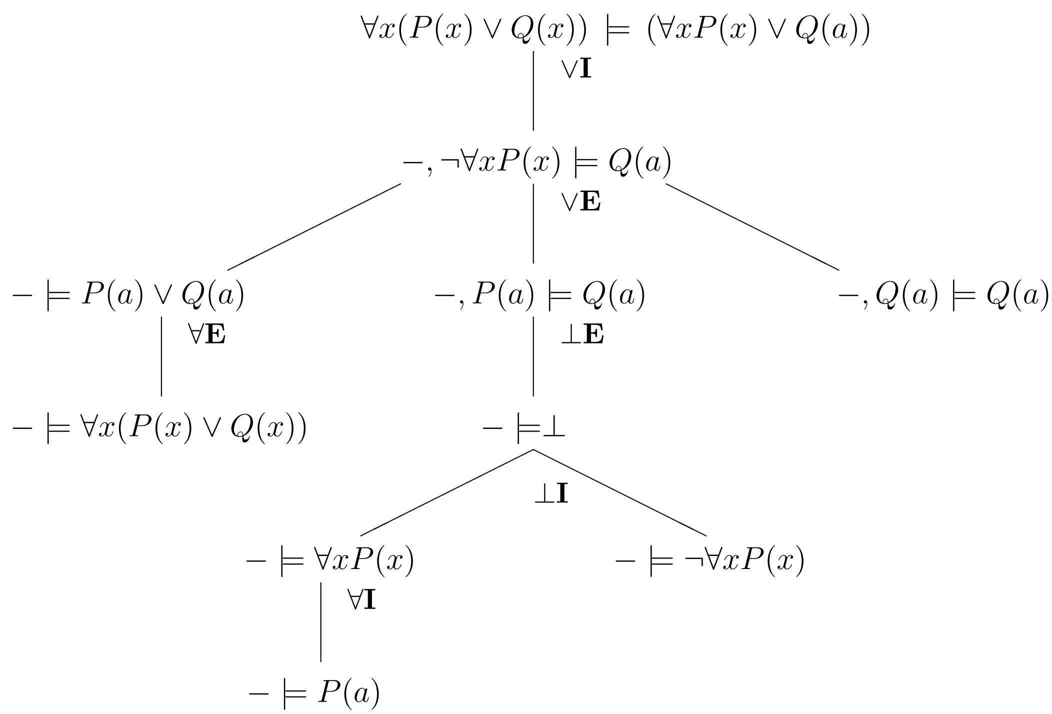
CSC 330F 2005, Assignment 4, St. George Campus
Due: Nov. 29, 2:10 PM

On the cover page of your assignment, you must list everyone with whom you discussed this assignment, and which problems you discussed with each person. You must also write **and sign** *the following statement: "I have read and understood the policy on collaboration on homework assignments stated in the Course Information handout." Without these, your homework will not be marked.*

---

1. Prove by using a tree derivation with 9 nodes that $\{\forall x(\neg P(x) \supset Q(x)), \neg\exists x P(x)\}$ logically entails $\forall y Q(y)$. [15]

2. The following purports to be a tree derivation showing that $\forall x(P(x) \vee Q(x))$ logically entails $(\forall x P(x) \vee Q(a))$: [10]



Explain why this is not a tree derivation by showing which inference rules are not being used correctly.

3. In this question we will simulate the DPLL procedure (by hand) showing a non-entailment in propositional logic. Let Σ be the following set of formulas:

$$\{\, q,\ \ (q \supset (a \vee b)),\ \ (a \wedge p \supset r),\ \ (b \wedge \neg p \supset r)\,\}.$$

Show that Σ does not entail r by:

(a) writing Σ together with $\neg r$ as a list of clauses; [5]

(b) tracing the execution of DPLL on this list of clauses, and showing that 1 is returned. [5]

If an atom must be chosen to split on, it can be selected arbitrarily.

4. Convert the following to clausal formulas (list of clauses): [15]

(a) $\neg(P(a) \supset P(c)) \vee P(c)$

(b) $[P(a) \supset (Q(a) \wedge \neg Q(a))] \equiv \neg P(a)$

(c) $\exists x P(x) \equiv \exists y P(y)$

(d) $\exists x \forall y R(x, y) \supset \forall y \exists x R(x, y)$

(e) $\exists x[\forall y \exists z(P(x, y) \wedge Q(x, z)) \vee \forall w R(x, w)]$

5. This question is a continuation of Question 2 from the previous assignment (involving Mary and her membership in two sports clubs) but now using Resolution. To do this question, you will need a correct version of the given facts (1), (2), (3), and the query F from part (b) written in first-order logic.

(a) Write (1), (2) (3) and $\neg F$ as a set of clauses. [5]

(b) Prove that (1), (2) (3) logically entail F by starting with the clauses from part (a) and showing how the Resolution procedure can derive the empty clause. Do this by drawing a diagram like we did in class. [10]

6. A *clique* of a graph is defined as a subset V of the vertices of the graph such that every pair of elements of V has an edge between them.

(a) Given a number k and a graph as input, show how to construct a sentence of propositional logic that is satisfiable iff the graph has a clique with k or more vertices. You may assume that the graph is presented by giving the number n of vertices and a list of pairs (i, j), where $1 \leq i < j \leq n$, indicating an edge between vertex i and vertex j. You may assume that $k \leq n$. [10]

(b) Consider the graph with 5 nodes, and the following edges: (1,3), (1,4), (2,4), (4,5). Show the sentence you would construct for this graph and $k = 3$ and explain informally why it is correct. [5]

(c) Consider the graph like the one above but with one more edge, (2,5). Show the sentence you would construct for this graph and $k = 3$ and explain again why it is correct. [5]

7. While function symbols are quite convenient in first-order logic, it appears that only one new rule of inference is needed to deal with them in a sound and complete way. The rule is as follows: [15]

If $\Sigma \models \forall x F$ then $\Sigma \models F[x/t]$, for any term t without variables.

We call this rule $\forall\mathbf{E}^{+}$ since it subsumes the original rule $\forall\mathbf{E}$ which only dealt with constants. Prove using a tree derivation with this new rule that the sentence $\forall x \forall y (x = y \supset f(x) = f(y))$ is valid. This can be done in a tree with 8 nodes.

Bonus question: The method proposed in class for converting a sentence of propositional logic into CNF may result in a sentence that is *exponentially* longer than the original. For example, each time a sentence $(F \equiv G)$ is encountered, it is replaced by the sentence $(F \supset G) \wedge (G \supset F)$, which has twice the size. However, for any sentence of propositional logic F of length n, it is possible to efficiently construct another sentence F' in CNF such that F is satisfiable iff F' is satisfiable as before, but where the length of F' is $O(n)$. This is done by introducing new atomic sentences into F'.

Explain the construction.

You may use books and Google as necessary for help. *Do not attempt any bonus work until the regular part of your assignment is complete.*