CS 151: Mathematical Foundations of Computing
Homework Assignment 01

Instructions

This assignment is due on *Blackboard* on Friday, September 15, at 11:55PM (Central Time).

Late submissions will be accepted within 24 hours after the deadline with a penalty of 25% of the assignment grade. No late submissions will be accepted more than 24 hours after the deadline.

This assignment is individual. Offering or receiving any kind of unauthorized or unacknowledged assistance is a violation of the University's academic integrity policies, will result in a grade of zero for the assignment and will be subject to disciplinary action.

Part I: Understanding logical expressions (30 pt.)

1. **(10 pt., 2 pt. each) Write each of the following conditional statements in the form "if p, then q" and state the corresponding converse, inverse, and contrapositive.**
   a. **I will send you the address only if you give me the number.**
   b. **Getting the job implies that you were the best candidate.**
   c. **It is necessary to have a valid password to log on to the server.**
   d. **A sufficient condition for the warranty to be good is that you bought the computer less than a year ago.**
   e. **You will pass the course unless you do not complete the homework.**

2. **(10 pt., 2 pt. each) Let $P(x)$ be the statement "x is a student," $Q(x)$ be the statement "x is a professor," and $R(x, y)$ be the statement "x asked a question to y." If the domain of x and y consists of all people, express each of the following sentences in terms of $P(x)$, $Q(x)$, $R(x, y)$, quantifiers, and logical operators.**
   a. **It is not the case that some student has never asked a question.**
   b. **There is a professor who has never been asked a question by a student.**
   c. **Some student has asked every professor a question.**
   d. **There is a student who has asked a question to exactly one professor.**
   e. **There are two different students who have asked each other a question.**

3. **(10 pt., 2 pt. each) Determine the truth value of each of the following statements if the domain consists of all integers. If it is true, find a domain for which it is false. If it is false, find a domain for which it is true. Justify your answer.**

a. $\exists x(x^3 = -1)$

b. $\forall x(x^2 \neq x)$

c. $\exists x(x^2 = 2)$

d. $\forall x(x^2 \geq 0)$

e. $\exists x(x + 1 > 2x)$

Part II: Proving logical equivalence using laws of propositional logic (30 pt.)

1. **(20 pt., 5 pt. each) Use the laws of propositional logic to prove that the following compound propositions are logically equivalent. Indicate the law(s) used in each step of the proof.**

   a. $(p \to q) \vee (p \to r)$ **and** $p \to (q \vee r)$

   b. $p \leftrightarrow q$ **and** $(p \wedge q) \vee (\neg p \wedge \neg q)$

   c. $p \to (q \vee r)$ **and** $(p \wedge \neg q) \to r$

   d. $\neg\big(r \vee \big(q \wedge (\neg r \to \neg p)\big)\big)$ **and** $\neg r \wedge (p \vee \neg q)$

2. **(10 pt., 5 pt. each) Use the laws of propositional logic to prove that the following compound propositions are tautologies. Indicate the law(s) used in each step of the proof.**

   a. $\big(\neg p \wedge (p \vee q)\big) \to q$

   b. $(p \vee q) \wedge (\neg p \vee r) \to (q \vee r)$

Part III: Constructing logical expressions from truth tables (20 pt.)

Suppose that a truth table with n variables is specified. A compound proposition can be constructed from this truth table by taking the disjunction of the conjunctions of the variables or their negations, with one conjunction included for each row where the compound proposition is true. The resulting compound proposition is said to be in **disjunctive normal form.**

p	q	?
T	T	T
T	F	F
F	T	F
F	F	T

Truth Table A

p	q	?
T	T	F
T	F	T
F	T	T
F	F	F

Truth Table B

p	q	?
T	T	T
T	F	F
F	T	T
F	F	T

Truth Table C

For example, the following compound proposition can be constructed from the true rows of Truth Table A: $(p \wedge q) \vee (\neg p \wedge \neg q)$.

1. **(2.5 pt.) Construct a compound proposition from the true rows of Truth Table B.**

2. **(2.5 pt.) Construct a compound proposition from the true rows of Truth Table C.**

A compound proposition can also be constructed from a truth table by taking the conjunction of the negation of the conjunctions of the variables or their negations, with one conjunction included for each row where the compound proposition is false. Note that this compound proposition is not in **disjunctive normal form.**

For example, the following compound proposition can be constructed from the false rows of Truth Table A: $\neg(p \wedge \neg q) \wedge \neg(\neg p \wedge q)$.

3. **(2.5 pt.) Construct a compound proposition from the false rows of Truth Table B.**

4. **(2.5 pt.) Construct a compound proposition from the false rows of Truth Table C.**

Both compound propositions constructed from Truth Table A are logically equivalent, since they were constructed from the same truth table; that is, $(p \wedge q) \vee (\neg p \wedge \neg q) \equiv \neg(p \wedge \neg q) \wedge \neg(\neg p \wedge q)$.

5. **(5 pt.) Use the laws of propositional logic to show that the compound propositions constructed from Truth Table B in questions (1) and (3) are logically equivalent. Indicate the law(s) used in each step of the proof [Hint: use the distributive law multiple times].**

You may have already realized that Truth Table A corresponds to the logical biconditional operation: $p \leftrightarrow q$. Thus, $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$ and $p \leftrightarrow q \equiv \neg(p \wedge \neg q) \wedge \neg(\neg p \wedge q)$.

6. **(1.5 pt.) Which logical operation studied in class corresponds to Truth Table B?**

7. **(1.5 pt.) Which logical operation studied in class corresponds to Truth Table C?**

A collection of logical operators is called **functionally complete** if every compound proposition is logically equivalent to a compound proposition involving only these logical operators.

8. **(2 pt.) Explain why the logical negation ($\neg$), conjunction ($\wedge$), and disjunction ($\vee$) operators form a functionally complete collection of logical operators [Hint: think about the fact that we can construct logical expressions from truth tables as described above].**

Part IV: Satisfiability (20 pt.)

Many problems, in diverse areas such as artificial intelligence and circuit design, can be modeled in terms of **propositional satisfiability**.

A compound proposition is **satisfiable** if there is an assignment of truth values to its variables that makes it true. For example, $p \wedge q$ is true when $p = T$ and $q = T$; thus, $p \wedge q$ is satisfiable.

When no such assignment exists, the compound proposition is **unsatisfiable**.

1. **(3 pt.) Explain why a compound proposition is unsatisfiable if and only if its negation is a tautology.**

2. **(2 pt.) Give an example of an unsatisfiable compound proposition with two variables, $\boldsymbol{p}$ and $\boldsymbol{q}$. Justify your answer.**

To show that a compound proposition is satisfiable, we need to find a particular assignment of truth values to its variables that makes it true. However, to show that a compound proposition is unsatisfiable, we need to show that *every* assignment of truth values to its variables makes it false.

3. **(15 pt., 5 pt. each) Determine (without using a truth table) whether each of the following compound propositions is satisfiable. Justify your answer.**

   a. $(\boldsymbol{p} \vee \neg \boldsymbol{q}) \wedge (\neg \boldsymbol{p} \vee \boldsymbol{q}) \wedge (\neg \boldsymbol{p} \vee \neg \boldsymbol{q})$

   b. $(\boldsymbol{p} \vee \neg \boldsymbol{q}) \wedge (\neg \boldsymbol{p} \vee \boldsymbol{q}) \wedge (\boldsymbol{q} \vee \boldsymbol{r}) \wedge (\boldsymbol{q} \vee \neg \boldsymbol{r}) \wedge (\neg \boldsymbol{q} \vee \neg \boldsymbol{r})$

   c. $(\boldsymbol{p} \leftrightarrow \boldsymbol{q}) \wedge (\neg \boldsymbol{p} \leftrightarrow \boldsymbol{q})$

A truth table can be used to determine whether a compound proposition is satisfiable. However, as the number of variables in a compound proposition grows, using a truth table to determine whether it is satisfiable becomes impractical.

No algorithm is known to determine in a reasonable amount of time whether an arbitrary compound proposition with a large number of variables (i.e., thousands, millions) is satisfiable. This is an important unsolved problem in Computer Science.