A. Why?
- We use propositions and predicates to write program specifications.
- Propositions and predicates can be related or manipulated syntactically or semantically.
- States describe memory; an expression has a value relative to a state.

B. Objectives
At the end of this homework, you should be able to
- Describe the relationship between syntactic equality and semantic equality.
- Translate expressions, propositions, and predicates to and from English.
- Design predicate functions for simple properties on values and arrays.

C. Group Work; What and How to Submit
- See the general instructions for homework.

D. Problems [50 points total]
Quantified variables range over \( \mathbb{Z} \) unless otherwise specified.

1. [6 = 2 * 3 points] What is the full parenthesization of
   a. \( p \land \neg r \land s \rightarrow \neg q \lor r \rightarrow \neg p \leftrightarrow \neg s \rightarrow t \) ?
   b. \( \exists m \cdot 0 \leq m < n \land \forall j \cdot 0 \leq j < m \rightarrow b[0] \leq b[j] \leq b[m] \)

2. [6 = 3 * 2 points] Give the minimal parenthesization of each of the following by showing what remains after removing all redundant parentheses. Hint: To avoid getting confused about which parentheses match each other, try rewriting the given parentheses with subscripts: (\(1, \ldots, 1\)) versus (\(2, \ldots, 2\)) and so on.
   a. \( ((\neg(p \lor q) \lor r) \rightarrow (((\neg q) \lor r) \rightarrow ((p \lor (\neg r)) \lor (q \land s))) \)
   b. \( \exists i \cdot (((0 \leq i) \land (i < m)) \land (\forall j \cdot ((m \leq j) \land (j < n)) \rightarrow (b[i] = b[j]))) \). (This predicate asks “Is there a value in \( b[0..m-1] \) > every value in \( b[m..n-1] \)?)
   c. \( \forall x.((\exists y.(p \rightarrow q)) \rightarrow (\forall z.(q \lor (r \land s)))) \)

* Leave \((0 \leq j < m)\) as is; don't expand it to \(((0 \leq j) \land (j < m))\). Don't forget to parenthesize \((b[0])\), e.g.
3. [4 = 2 * 2 points] Say whether the given propositions or predicates are \( \equiv \) or \( \not\equiv \). Briefly justify your answer.
   a. Is \( p \land q \lor \neg r \rightarrow \neg p \rightarrow q = ((p \land q) \lor ((\neg r \rightarrow ((\neg p) \rightarrow q))) \) ?
   b. Is \( \forall x . p \rightarrow \exists y . q \rightarrow r = ((\forall x . p) \rightarrow (\exists y . q)) \rightarrow r \) ?

4. [6 = 2 * 3 points] Say whether each of the following is a tautology, contradiction, or contingency. If it's a contingency, show an instance when the proposition is true and show an instance where it's false.
   a. \( (p \rightarrow (q \rightarrow r)) \leftrightarrow ((p \rightarrow q) \rightarrow r) \)
   b. \( (\forall x \in \mathbb{Z} . \forall y \in \mathbb{Z} . f(x, y) > 0) \rightarrow (\exists x \in \mathbb{Z} . \exists y \in \mathbb{Z} . f(x, y) > 0) \). Rely on the idea that for \( (\forall u . \varphi) \) to be false, we need some value for \( u \) for which \( \varphi \) is false. I.e., we need \( (\exists u . \neg \varphi) \). Similarly, for \( (\exists v . \psi) \) to be false, we need \( \psi \) to be false for every value of \( v \). I.e., we need \( (\forall v . \neg \psi) \).

5. [4 points] Which of the following mean(s) \( p \rightarrow q \) and which mean \( q \rightarrow p \)?
   a. \( p \) is sufficient for \( q \)
   b. \( p \) only if \( q \)
   c. \( p \) if \( q \)
   d. \( p \) is necessary for \( q \)

6. [4 = 2 * 2 points] Let \( e_1 \) and \( e_2 \) be expressions.
   a. In general, does \( e_1 \neq e_2 \) imply \( e_1 \neq e_2 \)? If yes, briefly justify (a sentence or two is fine); if no, give a counterexample (specific values for \( e_1 \) and \( e_2 \) that show that this implication does not always hold).
   b. In general, does \( e_1 = e_2 \) imply \( e_1 = e_2 \)? Again give a brief justification or counterexample.

7. [6 points] The goal is to show that \( p \land \neg (q \land r) \rightarrow q \land r \rightarrow \neg p \) is a tautology by proving it is \( \leftrightarrow T \). To do this, complete the proof of equivalence below using (only) the propositional logic rules (from Lecture 2). Be sure to include the names of the rules. There's more than one correct answer [just give one of them].
   \[
   \begin{align*}
   p \land \neg (q \land r) & \rightarrow q \land r \rightarrow \neg p \\
   \text{[you fill in]} & \quad \text{Defn} \rightarrow \\
   \text{[you fill in]} & \quad \text{Defn} \rightarrow \\
   \text{[and so on]} &
   \end{align*}
   \]

8. [6 points] Simplify \( \neg (\forall x . (\exists y . x \leq y) \lor \exists z . x \geq z) \) to a predicate that has no uses of \( \neg \). Present a proof of equivalence. You’ll need DeMorgan’s Laws. Also use rules like "\( \neg (e_1 \leq e_2) \leftrightarrow e_1 > e_2 \) by negation of comparison".
9. [6 points] Write the definition of a predicate function \( GT(b, x, m, k) \) that yields true iff \( x > b[m], \ldots b[m+k-1] \). E.g., in the state \( b = (1, 3, -2, 8, 5) \), \( GT(b, 4, 0, 3) \) is true; \( GT(b, 0, 1, 2) \) is false. You can assume without testing that the indexes \( m, \ldots m+k-1 \) are all in range. If \( k \leq 0 \), the sequence \( b[m], b[m+1], \ldots, b[m+k-1] \) is empty and \( GT(b, x, m, k) \) is true. (It's straightforward to write \( GT \) so that this is not a special case.)

Remember, this has to be a **predicate function**, not a program that calculates a boolean value. Hint: Check the discussion in the Class 2 notes about trying to translate programs to predicates.