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. Formatting and Submitting Your Work
- [Website update]: You may work in groups of \( \leq 4 \) people. Submit your answers just one time: Pick someone from your group and be sure to include the names and A-id numbers of all the group members at the top of the submission. Submit your homework to the assignment folder in Blackboard, as a pdf file (scanned or computer-generated).
- You don’t have to use a word processor to write out your answers: Feel free to convert logical symbols into ASCII text: For \( \land, \lor, \rightarrow, \neg, \exists, \forall \), write and, or, ->, !, all, and exist. For \( \Rightarrow, \Leftrightarrow, \equiv, \not\equiv \), write =>, <=>, ==, and !=.

D. Problems [100 points total]

Part 1: Logic Review
As usual \( p, q, \) and \( r \) are propositions or predicates. Quantified variables range over \( \mathbb{Z} \) unless otherwise specified.

1. \([-12 = 4 * 3 \text{ points}]\) Give the full parenthesization of each of the following. Include the outermost parentheses.
   a. \( p \land \neg q \land r \rightarrow \neg q \lor r \rightarrow \neg p \leftrightarrow \neg s \rightarrow t \)
   b. \( p \rightarrow q \land r \rightarrow ((p \rightarrow \neg q) \rightarrow r) \land (\neg(q \land r) \rightarrow \neg p) \)
   c. \( \exists m. 0 \leq m < n \land \forall j. 0 \leq j < m \rightarrow b[0] \leq b[j] \land b[j] \leq b[m] \) (Don’t expand \( 0 \leq m < n \) to \( 0 \leq m \land m < n \))
   d. \( (\exists x. \forall y.f(x) > g(y)) \rightarrow \forall y. \exists x.f(x) \geq g(y) \)

2. \([-16 = 4 * 4 \text{ points}]\) 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: \( (1 \ldots) \) versus \( (2 \ldots) \) and so on.
   a. \( (((\neg(p \lor q) \lor r) \lor (((\neg q) \lor r) \lor ((p \lor (\neg r)) \lor (q \land s)))) \)
   b. \( (((p \lor (q \lor r)) \lor ((q \rightarrow (r \rightarrow (s \land (p \land r)))))) \lor (\neg(p \land q)) \)
3. [12 = 4 * 3 points] Say whether the given propositions or predicates are \( \equiv \) or \( \neq \). Briefly justify your answer.
   a. \( p \land q \lor \neg r \rightarrow \neg p \lor q \equiv ((p \land q) \lor ((\neg r \rightarrow ((\neg p) \rightarrow q)))) ? \)
   b. \( (p \rightarrow (\neg q \land s)) \rightarrow (r \land (s \rightarrow p)) \equiv p \rightarrow (\neg q \land s \rightarrow r \land (s \rightarrow p)) ? \)
   c. \( \forall x, p \rightarrow \exists y, q \rightarrow r \equiv (\forall x, p \rightarrow (\exists y, q)) \rightarrow r ? \)
   d. \( \forall x, \forall y, p \land q \equiv \forall y, \forall x, q \land p ? \)

4. [12 = 4 * 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. \( s \land t \lor u \rightarrow s \lor u \)
   b. \( (p \rightarrow (q \rightarrow r)) \leftrightarrow ((p \rightarrow q) \rightarrow r) \)

For c and d, argue based on our informal understanding of the quantifiers (not on the definition of \( \equiv \), since we don't get to that until Lecture 4).

   c. \( (\forall x \in \mathbb{Z}, \forall y \in \mathbb{Z} \cdot f(x, y) > 0) \rightarrow (\exists x \in \mathbb{Z}, \exists y \in \mathbb{Z} \cdot f(x, y) > 0) \)
   d. \( \neg (\exists x \in \mathbb{Z}, \forall y \in \mathbb{Z} \cdot f(x, y) > 0) \rightarrow (\exists x \in \mathbb{Z}, \exists y \in \mathbb{Z} \cdot f(x, y) \leq 0) \)

5. [4 points] Which of the following mean \( p \rightarrow q \) and which mean \( q \rightarrow p \)?
   - \( p \) is sufficient for \( q \)
   - \( p \) only if \( q \)
   - \( p \) \( \iff \) \( q \)
   - \( p \) is necessary for \( q \)

6. [6 = 3 * 2 points] Let \( e_1 \) and \( e_2 \) be expressions.
   a. Does \( e_1 \equiv e_2 \) imply \( e_1 = e_2 \), in general? 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. Does the converse hold? (Does \( e_1 = e_2 \) imply \( e_1 \equiv e_2 \), in general?) Again give a brief justification or counterexample.
   c. Repeat, with whether or not \( e_1 \neq e_2 \) implies \( e_1 \equiv e_2 ? \)

7. [8 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 \( \iff \) \( T \). 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].

\[
p \land \neg(q \land r) \rightarrow q \land r \rightarrow \neg p
\]
\[
\text{[you fill in]} \quad \text{Defn} \rightarrow
\]
\[
\text{[you fill in]} \quad \text{Defn} \rightarrow
\]
\[
\text{[and so on]}
\]

8. [8 points] Simplify \( \neg(\forall x \cdot (\exists y \cdot x \leq y)) \lor \exists z \cdot x \geq z \) to a predicate that has no uses of \( \neg \). Present a proof of equivalence. You'll need DeMorgan's Laws. Also use rules like "\( \neg(e_1 \leq e_2) \iff e_1 > e_2 \) by negation of comparison".

9. [10 points] Write the definition of a predicate function \( \text{GT}(b, x, m, k) \) that yields true iff \( x > b[m] \), \( \ldots \) \( b[m+k-1] \). E.g., in the state \( \{ b = (1, 3, -2, 8, 5) \} \), \( \text{GT}(b, 4, 0, 3) \) is true; \( \text{GT}(b, 0, 1, 2) \) 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 \( \text{GT}(b, x, m, k) \) is true. (It's straightforward to write \( \text{GT} \) so that this is not a special case.) Remember, this has to be a **predicate function**, not a program that calculates a boolean value.

**Part 2: Expressions, States**

10. \([6 = 3 * 2 \text{ points}]\) Which of the following expressions are legal or illegal according to the syntax we’re using? Assume \( x, y, z \) are integer variables, \( f \) is function, and \( b \) and \( b' \) are array names.
   a. \((x < y ? T : 17)\)
   b. \((i = 3 ? b[1] : b'[2])[k]\)
   c. \(f(b, b', x)\)

11. \([6 = 3 * 2 \text{ points}]\) For each of the following, there are three possibilities:
   - The state is **illegal** (doesn't meet the criteria for being a state).
   - The state is **legal but not proper** for the given expression.
   - The state is legal, proper, and the **expression evaluates** to a value or runtime error (say which).

   **Notation:** \( x, y, \ldots, i, \ldots \) are variables of type integer. \( b \) is a variable of type array of integer. \( \alpha, \beta, \gamma, \ldots \) are semantic values, as are items spelled out in English like two plus two.
   a. \(\{x = \text{ten}, y = \text{eight plus one}\} \text{ and } x+y\)
   b. \(\{b = \gamma, i = 1\} \text{ where } \gamma(0) = 2 \text{ and } \gamma(1) = 5, \text{ and } b[b[i]-1]\)\)
   c. \(\{c = \alpha, d = 2\alpha, e = 3\alpha\} \text{ (for some } \alpha\) \text{ and } d/c+(0*z)\)