# *Logic Review*

*CS 536: Science of Programming, Fall 2019*

*Due Wed Sep 4, 11:59 pm*

## *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*

- You don't have to use a word processor to write out your answers: Feel free to convert logical symbols into ASCII text: For $\wedge$, $\vee$, $\rightarrow$, $\neg$, $\forall$, $\exists$,write `and`, `or`, `->`, `!`, `all`, and `exist`. For $\Rightarrow$, $\Leftrightarrow$, $\equiv$, and $\not\equiv$, write `=>`, `<=>`, `==`, and `!==`.

## *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. $p \wedge \neg r \wedge s \rightarrow \neg q \vee r \rightarrow \neg p \leftrightarrow \neg s \rightarrow t$ ?
   b. $\exists\, \mathtt{m} \,.\, 0 \le \mathtt{m} < \mathtt{n} \wedge \forall\, \mathtt{j} \,.\, 0 \le \mathtt{j} < \mathtt{m} \rightarrow \mathtt{b[0]} \le \mathtt{b[j]} \le \mathtt{b[m]}$ *

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 $(_2$ and $)_2$ and so on.
   a. $((\neg(p \vee q) \vee r) \rightarrow (((\neg q) \vee r) \rightarrow ((p \vee (\neg r)) \vee (q \wedge s))))$
   b. $(\exists\, \mathtt{i} \,.\, (((0 \le \mathtt{i}) \wedge (\mathtt{i} < \mathtt{m})) \wedge (\forall\, \mathtt{j} \,.\, (((\mathtt{m} \le \mathtt{j}) \wedge (\mathtt{j} < \mathtt{n})) \rightarrow (\mathtt{b[i]} = \mathtt{b[j]})))))$. (This predicate asks "Is there a value in `b[0..m-1]` > every value in `b[m..n-1]`?)
   c. $\forall \mathtt{x}.((\exists \mathtt{y}.(p \rightarrow q)) \rightarrow (\forall \mathtt{z}.(q \vee (r \wedge s))))$

* Leave $(0 \le \mathtt{j} < \mathtt{m})$ as is; don't expand it to $((0 \le \mathtt{j}) \wedge (\mathtt{j} < \mathtt{m}))$. 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 $p \wedge q \vee \neg r \to \neg p \to q \equiv ((p \wedge q) \vee ((\neg r \to ((\neg p) \to q))))$ ?
   b. Is $\forall\, \mathtt{x} \,.\, p \to \exists\, \mathtt{y} \,.\, q \to r \equiv ((\forall\, \mathtt{x} \,.\, p) \to (\exists\, \mathtt{y} \,.\, q)) \to r$ ?

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. $(p \to (q \to r)) \leftrightarrow ((p \to q) \to r)$
   b. $(\forall\, \mathtt{x} \in \mathbb{Z} \,.\, \forall\, \mathtt{y} \in \mathbb{Z} \,.\, \mathtt{f}(\mathtt{x}, \mathtt{y}) > 0) \to (\exists\, \mathtt{x} \in \mathbb{Z} \,.\, \exists\, \mathtt{y} \in \mathbb{Z} \,.\, \mathtt{f}(\mathtt{x}, \mathtt{y}) > 0)$. 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 $\psi$ to be false for every value of $v$. I.e., we need $(\forall v.\ \neg \psi)$.

5. [2 points] Which of the following mean(s) $p \to q$ and which mean $q \to p$?
   a. $p$ is sufficient for $q$
   b. $p$ only if $q$

6. [6 = 3 * 2 points] Let $e_1$ and $e_2$ be expressions.
   a. In general, does $e_1 \neq e_2$ imply $e_1 \not\equiv e_2$? 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 $e_1 \equiv e_2$,? Again give a brief justification or counterexample.

7. [6 points] The goal is to show that $p \wedge \neg(q \wedge r) \to q \wedge r \to \neg p$ 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].

   $p \wedge \neg(q \wedge r) \to q \wedge r \to \neg p$
   *[you fill in]* Defn $\to$
   *[you fill in]* Defn $\to$
   *[and so on]*

8. [6 points] Simplify $\neg(\forall\, \mathtt{x} \,.\, (\exists\, \mathtt{y} \,.\, \mathtt{x} \le \mathtt{y}) \vee \exists\, \mathtt{z} \,.\, \mathtt{x} \ge \mathtt{z})$ to a predicate that has no uses of $\neg$. Present a proof of equivalence. You'll need DeMorgan's Laws. Also use rules like "$\neg(e_1 \le e_2) \Leftrightarrow e_1 > e_2$ by negation of comparison".

9. [6 points] Write the definition of a predicate function $\mathtt{GT}(\mathtt{b}, \mathtt{x}, \mathtt{m}, \mathtt{k})$ that yields true iff $\mathtt{x} > \mathtt{b}[\mathtt{m}], \ldots \mathtt{b}[\mathtt{m+k-1}]$. E.g., in the state $\{\mathtt{b} = (1, 3, -2, 8, 5)\}$, $\mathtt{GT}(\mathtt{b}, 4, 0, 3)$ is true; $\mathtt{GT}(\mathtt{b}, 0, 1, 2)$ is false. You can assume without testing that the indexes $\mathtt{m}, \ldots \mathtt{m+k-1}$ are all in range. If $\mathtt{k} \le 0$, the sequence $\mathtt{b}[\mathtt{m}], \mathtt{b}[\mathtt{m+1}], \ldots, \mathtt{b}[\mathtt{m+k-1}]$ is empty and $\mathtt{GT}(\mathtt{b}, \mathtt{x}, \mathtt{m}, \mathtt{k})$ 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 Lecture 2 about trying to translate

 

10. [4 = 2 * 2 points] For each of the following, there are three possibilities:

- The state is *ill-formed* (doesn't meet the criteria for being a state).
- The state is *well-formed but not proper* for the given predicate
- The state is legal, proper, and satisfies the predicate.

**Notation**: `x`, `y`, …, `i`, … are variables of type integer. `b` is a variable of type array of integer. α, β, γ, … are semantic values; so are items spelled out in English like *two plus two*.

a. {`x` = *ten*, `y` = *eight plus one*} ⊨ `x+y` > 0.

b. {`c` = α, `d` = 2α, `e` = 3α} (for some α} and `d/c+(0*z)` ≤ 0.

 