Learning Objectives

By the end of this worksheet, you will:

- Translate sentences between natural English and predicate logic.
- Use definitions of predicates to simplify or expand statements in predicate logic.
- Apply negation equivalence rules to simplify statements in predicate logic.

1. **Translation with predicates**. Suppose we have a set P of computer programs that are each meant to solve the same task. Some of the programs are written in the Python programming language, and some are written in a different language. Some of the programs correctly solve the task, and others do not.

   Let's define the following predicates:

   $$Python(x) : \text{``}x \text{ is a Python program''}, \qquad \text{where } x \in P$$
   $$Correct(x) : \text{``}x \text{ solves the task correctly''}, \qquad \text{where } x \in P$$

   Express each English statement below as a sentence in predicate logic.

   (a) Program *my_prog* is correct and is written in Python.

   (b) An incorrect program is written in Python.

   (c) No Python program is correct.

   (d) Every incorrect program is written in Python.

   Translate each predicate logic statement below into natural English.

   (a) $\exists x \in P, \ Python(x) \land Correct(x)$

   (b) $\forall x \in P, \ \neg Python(x) \land Correct(x)$

   (c) $\neg\big(\forall x \in P, \ Correct(x) \Rightarrow Python(x)\big)$

   (d) $\forall x \in P, \ \neg Python(x) \Leftrightarrow Correct(x)$

2. **Quantifiers in subformulas**. So far, we have seen quantifiers only as the leftmost components of our formulas. However, because all predicate statements have truth values (i.e., are either True or False), they too can be combined using the standard propositional operators. Let's see some examples of this (refer to the same predicates as Question 1).

   (a) Translate the following statement into English.

   $$\Big(\forall x \in P,\; Python(x) \Rightarrow Correct(x)\Big) \vee \Big(\forall y \in P,\; Python(y) \Rightarrow \neg Correct(y)\Big)$$

   (b) Translate the following statement into predicate logic. "If at least one Python program is correct, then all Python programs are correct."

   (c) Finally, consider the following two statements:

   $$\big(\exists x_1 \in \mathbb{N},\; x_1 \mid 165\big) \wedge \big(\exists x_2 \in \mathbb{N},\; 7 \mid x_2\big) \qquad\qquad \text{(Statement 1)}$$
   $$\exists x \in \mathbb{N},\; x \mid 165 \wedge 7 \mid x \qquad\qquad \text{(Statement 2)}$$

   What is the difference between these two statements? Are they True or False?

3. **Expanding definitions**. Consider the following statement:

   If m and n are odd integers, then mn is an odd integer.

   If we want to express this statement using predicate logic, we need to start with a definition of the term "odd". Let $n \in \mathbb{Z}$. We say that n is **odd** when $2 \mid (n+1)$, i.e., when $\exists k \in \mathbb{Z}, n+1 = 2k$.

   (a) Write the definition of a predicate over the integers named *Odd* that is True when its argument is odd.

   (b) Using the predicate *Odd* and the notation of predicate logic, express the statement:
       For every pair of odd integers m and n, mn is an odd integer.

   (c) Repeat part (b) but fully expand the definitions of the predicates *Odd* and $\mid$.

(d) Repeat parts (b) and (c) using the following statement (which states the converse of the original implication).

For every pair of integers m and n, if mn is odd, then m and n are odd.

4. **Simplifying negated formulas**. Recall the rules governing how to simplify negations of predicate formulas:

- $\neg(\neg p)$ becomes p.
- $\neg(p \vee q)$ becomes $\neg p \wedge \neg q$.
- $\neg(p \wedge q)$ becomes $\neg p \vee \neg q$.
- $\neg(p \Rightarrow q)$ becomes $p \wedge \neg q$.
- $\neg(p \Leftrightarrow q)$ becomes $(p \wedge \neg q) \vee (\neg p \wedge q)$.
- $\neg(\exists x \in S, \ P(x))$ becomes $\forall x \in S, \ \neg P(x)$.
- $\neg(\forall x \in S, \ P(x))$ becomes $\exists x \in S, \ \neg P(x)$.

Using these rules, simplify each of the following formulas so that the negations are applied directly to predicates/propositional variables. Note: this is a pretty mechanical exercise, but an extremely valuable one: once we get to the next chapter, we will be assuming you can take negations of statements very quickly as a first step in some proofs.

(a) $\neg\Big((a \wedge b) \Leftrightarrow c\Big)$

(b) $\neg\Big(\forall x, y \in S, \ \exists z \in S, \ P(x, y) \wedge Q(x, z)\Big)$

(c) $\neg\Big((\exists x \in S, \ P(x)) \Rightarrow (\exists y \in S, \ Q(y))\Big)$

5. **Choosing a universe and predicates**. Consider the statement

$$\left[\left(\exists x \in U,\ P(x)\right) \wedge \left(\exists y \in U,\ Q(y)\right)\right] \Rightarrow \left[\exists z \in U,\ P(z) \wedge Q(z)\right].$$

Define a non-empty domain U and predicates P and Q for which this statement is False.

Hint: The statement says: "If some $x \in U$ makes $P(x)$ True and some $y \in U$ makes $Q(y)$ True, then some $z \in U$ makes both $P(z)$ and $Q(z)$ True." Think about how this statement could be False, and use this to construct a U, P and Q.