

Learning Objectives

By the end of this worksheet, you will:

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

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

Let's define the following predicates:

$\text{Python}(x)$: “ x is a Python program”, where $x \in P$

$\text{Correct}(x)$: “ x solves the task correctly”, where $x \in P$

Express each English statement below as a sentence in predicate logic. Note that all variables should be quantified over P (the set of computer programs that we're considering for this question).

- (a) Some incorrect program is written in Python.
- (b) Program *my_prog* is correct and is written in Python.
- (c) No Python program is correct.
- (d) Every incorrect program is written in Python.

Now, give a natural English sentence that captures the meaning of each symbolic sentence below.

- (a) $\exists x \in P, \text{Python}(x) \wedge \text{Correct}(x)$
- (b) $\forall x \in P, \neg \text{Python}(x) \wedge \text{Correct}(x)$
- (c) $\neg(\forall x \in P, \text{Correct}(x) \Rightarrow \text{Python}(x))$

$$(d) \forall x \in P, \neg \text{Python}(x) \iff \text{Correct}(x)$$

2. **Quantifiers in subformulas.** So far, we have generally 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.

- (a) Using the same predicates as Question 1, translate the following statement into English.

$$(\forall x \in P, \text{Python}(x) \Rightarrow \text{Correct}(x)) \vee (\forall y \in P, \text{Python}(y) \Rightarrow \neg \text{Correct}(y))$$

- (b) Again using the same predicates as Question 1, 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:

$$(\exists x_1 \in \mathbb{N}, x_1 \mid 165) \wedge (\exists x_2 \in \mathbb{N}, 7 \mid x_2) \\ \exists x \in \mathbb{N}, x \mid 165 \wedge 7 \mid x$$

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 mathematical logic, we need to start with a definition of the term "odd". An integer n is said to be **odd** if and only if $2 \mid (n + 1)$. That is, n is odd if and only if $\exists k \in \mathbb{Z}, n + 1 = 2k$.

- (a) Write the definition of a predicate over the integers named *Odd* that is True if and only if 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 do not use the predicates *Odd* or \mid . Instead, use the full definition of *Odd* that includes a quantified statement.

(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:¹

- $\neg(\neg p)$ 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) $\neg((a \wedge b) \Leftrightarrow c)$

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

¹Found on page 25 of the Course Notes.

$$(c) \neg \left((\exists x \in S, P(x)) \Rightarrow (\exists y \in S, Q(y)) \right).$$

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

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

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

Hint: Start by noting that 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 what must be so for the statement to be False, and use this to construct a U , P and Q .