CSC110 Lecture 6: Introduction to Formal Logic

Print this handout

Exercise 1: Propositional and Predicate Logic

1. Let p be a propositional variable representing the statement "All cats are cute", and q be a propositional variable representing the statement "All dogs are cute".

   a. Translate the statement $p \Rightarrow q$ into English.

   b. What is the *hypothesis* of this implication?

   c. What is the *conclusion* of this implication?

   d. Write down an English translation of the *converse* of this implication.

   e. Write down an English translation of the *contrapositive* of this implication.

2. 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 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:

$$\begin{aligned} Python(x) &: x \text{ is a Python program}, \qquad && \text{where } x \in P \\ Correct(x) &: x \text{ solves the task correctly}, \qquad && \text{where } x \in P \end{aligned}$$

   Translate each statement below from English into predicate logic, or vice versa.

   a. Program *my_prog* is correct and is written in Python. (*my_prog* is an element of P)

   b. At least one incorrect program is written in Python.

   c. $\forall x \in P, Python(x) \land Correct(x)$

Exercise 2: Translating logical statements into Python

1. Now let's practice translating these statements into Python expressions. First, our set-up:

   - Suppose we have a variable `programs` which refers to a set of values representing Pythoi programs. (You can imagine these are strings, but the exact data type doesn't matter for this exercise.)
   - Suppose we have a variable `my_prog` which refers to one of these programs (e.g., `my_prog in programs` is `True`).
   - Suppose we also have Python functions `is_python` and `is_correct`, which each take a "program" value and returns a `bool` representing whether that program "is a Python

program” and “is a correct program”, respectively.

Given these Python variables and functions, write expressions to express each of the statemen from Question 3 in the previous exercise. The first one should use the `my_prog` variable, and the other two will require using `any`/`all` and comprehensions.

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

b. At least one incorrect program is written in Python.

c. $\forall x \in P,\ Python(x) \land Correct(x)$

Exercise 3: Filtering collections

1. Using the same set and predicates as Exercise 1 Question 2, translate the following statements into predicate logic.

    a. Every Python program is correct.

    b. No Python program is correct.

2. a. Given the following function: (1) complete its doctest example, and (2) implement it usin a *filtering comprehension*.

```python
def longest_cool_string(strings: list) -> int:
    """Return the length of the longest given string that contains the
        substring 'cool'.

    >>> longest_cool_string(['cool beans', 'hello', 'David is cool'])



    """
```

b. Suppose we have a Python variable `strings` that refers to a set of strings. Translate the following English statement into a Python expression (using variable `strings` and one c `any`/`all`):

> At least one string in `strings` contains the substring `'cool'`.

Additional exercises

1. Write a Python function that, given three boolean values p, q, and r, returns the value of the propositional formula $((p \Rightarrow q) \land r) \Leftrightarrow (p \Rightarrow (q \land r))$. We have begun the Function Design Recipe for you.

```python
def propositional_formula(p: bool, q: bool, r: bool) -> bool:
    """Return the value of ((p ⇒ q) ∧ r) ⇔ (p ⇒ (q ∧ r)).

    >>> propositional_formula(True, False, False)
    True
    >>> propositional_formula(False, False, False)
    False
    """
```

2. Here are some additional statements that you can use to practice your translations. For each one, translate it twice: into English/symbolic logic, and then into a Python expression, using tl same variables/functions we introduced earlier in this worksheet.

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

b. Every incorrect program is written in Python.

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

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

3. 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 car be combined using the standard propositional operators. Let's see some examples of this.

   a. Using the same predicates as before, translate the following statement into English.

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

   b. Again using the same predicates as before, translate the following statement into predica logic. "If at least one Python program is correct, then all Python programs are correct."