

Basic Logic

Proposition

A proposition is a statement that is either true or false. For instance ‘Philadelphia is the capital of Pennsylvania’ is a proposition, since that is a false statement.

Things that do not have a true-false answer are not propositions. For instance, the most commonly asked ‘What is today’s weather going to be like’.

Propositional variables such as p , q and r can be used to denote arbitrary propositions, as in “ p : January has 31 days”.

Boolean operators

This section will be assumed knowledge. The only reason we go through this is to introduce the syntax we will use for our course.

Conjunction operator (AND) will be denoted by \wedge - use `\wedge` in latex.

Disjunction operator (OR) will be denoted by \vee - use `\vee` in latex.

Exclusive or (XOR) will be denoted by \oplus - use `\oplus` in latex. Remember that $p \oplus q$ is true only when exactly one of the propositions is true.

NOT will be denoted by \neg - use `\neg` in latex.

One of the first things to learn in logic is expressing English sentences in a more mathematical form using logical connectives.

As an example consider the statement - ‘Dinner is eaten in our house at 7pm or at 8pm’. If we wanted to express this in terms of logical connectives, we can break this down into

$$p : \text{I eat dinner at 7 pm}$$
$$q : \text{I eat dinner at 8 pm}$$

Then it seems like the most logical way of expressing the statement would be to write $p \vee q$. But if we pause for a moment we realize that one aspect of the English sentence is not really being captured. If we eat dinner at 7pm, can we say something about dinner being eaten at 8pm? Generally speaking, the English language uses the word or in the sense of exclusive or. So really if we wanted to capture the full meaning of our English sentence we should be saying $p \oplus q$.

This example might seem like an exaggeration of English ambiguities, but it is important to understand that if we want to make a computer work in a normal deterministic fashion,

then the more clearly we can express thoughts and commands, the simpler it is to get the computer to do its job.

The order of operations for Boolean algebra. The normal order of operations for Boolean algebra is the following

1. \neg
2. \wedge
3. \vee

Simplifying Boolean Expressions and Truth Tables

Truth tables are definitely things that we assume you have seen by this point. Either as a past course or in CIT593 (Note: will be covered in class if someone says they have not seen them at all).

For the simplification of Boolean expressions, perhaps the only thing that you need to get used to is that the mathematical symbols look slightly different from what you might have seen in other areas, especially if you have done logic as part of a circuit design course.

Example $(p \vee r) \wedge (p \wedge s \vee p \wedge \neg s) \vee (p \wedge r) \vee r$

Worked out on in class.

Conditional statements

The "if-then" type of statement has specific rules when it comes to logic. Generally called an implication, it is expressed as $p \rightarrow q$ where p and q are propositions.

Very importantly, $p \rightarrow q$ is itself a proposition and it has a truth value which is determined from the following truth table.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

The first part of a conditional is called the hypothesis and the second is called the conclusion. The hypothesis being false gives you no evidence for the truth value of the implication and the way that mathematical logic works, you go with the 'innocent until proven guilty' idea.

We have been using this already when we have discussed whether or not the less-than relationship is anti-symmetric.

Expressing conditionals in English

The following all express $p \rightarrow q$ in English

1. If p then q
2. If p, q
3. q if p
4. p only if q
5. p implies q
6. p is sufficient for q
7. q is necessary for p

Contrapositive, Converse and Inverse

There are 3 statements that are closely related to $p \rightarrow q$.

They are

- Contrapositive $\neg q \rightarrow \neg p$
- Converse $q \rightarrow p$
- Inverse $\neg p \rightarrow \neg q$

Predicates and Quantifiers

Many mathematical statements contain variables. Statements that contain variables are different from propositions because their truth value depends on the value of the variable. A logical statement that is a function of a variable is called a predicate. A predicate always has an underlying domain associated with it. Variables are only allowed to take values from within the domain.

Universal quantifier

Sometimes we want to assert that a predicate is true regardless of the value taken from the domain. This is a universally quantified predicate and is represented with the \forall .

$$\forall x \in \mathbb{R}^+ \frac{1}{x+1} < 1$$

Proving a universally quantified statement is **True** requires showing that it is true regardless of the value of any variables that are present in the statement.

$$\frac{1}{x+1} = 1 - \frac{x}{x+1}$$

$$x \in \mathbb{R}^+ \rightarrow x+1 \in \mathbb{R}^+$$

$$\text{Therefore } \frac{x}{x+1} \in \mathbb{R}^+$$

$$\text{Therefore } 1 - \frac{x}{x+1} < 1, \forall x \in \mathbb{R}^+$$

Proving a universally quantified statement is **False** is easier. It just requires one counter example.

For example $\forall x \in \mathbb{R}, x^2 > x$ is a false statement because for $x = 0, x^2 = x$.

Existential quantifier

If there is at least one value in the domain that causes the predicate to evaluate to true, that is expressed using an existential quantifier \exists .

To prove an existentially quantified statement, you just need to show one single value that makes the statement true.

For instance, to prove the statement ‘There is a natural number n such that $2^n > n!$ ’ all you have to do is say that for $n = 2, 2^2 > 2!$ because after all $4 > 2$.

What will proving an existential statement is False involve? That is the same as proving that no value in the domain will make the statement True.

For instance to prove that $\exists x \in \mathbb{R}^+, x+1 < x$ is a false statement the following approach can be used.

$$x+1 < x$$

$$\implies 1 < 0 \quad (\text{ by subtracting } x \text{ from both sides})$$

And $1 \not< 0$ therefore proved.

Writing English statements with quantifiers

Now that we have seen quantifiers, we can express more English sentences using them.

Every CIT592 student has taken the midterm.

Assume x is a variable that comes from the domain of all MCIT first year students.

Then this would be one way of writing the above statement.

Let $P(x)$: x is a student in 592.

Let $Q(x)$: x has taken the midterm.

$\forall x P(x) \rightarrow Q(x)$

Negating quantified statements

It is important to be able to express the negation of a quantified statement.

Consider the statement ‘All birds fly’. To write this as a quantified statement you will probably consider x to come from the domain of all birds and let $F(x)$ be a predicate that is true when x can fly. $\forall x, F(x)$.

What if we wanted to negate that statement. In English we would simply say ‘Not all birds fly’. Which actually can be equivalently stated as ‘There is at least one bird that does not fly’. That brings us to more familiar territory as far as expressing things with quantifiers goes.

$$\neg(\forall x F(x)) \equiv \exists x \neg F(x)$$

The sign placed between the two quantified statements is the logical equivalence sign. That is saying that whenever the left statement is true, the right statement is true and whenever the left statement is false then the right statement is false.

Similarly consider the statement ‘There is some student that proved Fermat’s last theorem’. Let us assume the domain is the set of all students. And let $FLT(x)$ be true if the student proved Fermat’s last theorem. Then the statement is saying $\exists x FLT(x)$.

What if we negate the statement. In English we might simply say there is no student that proved Fermat’s last theorem. That is the equivalent to saying for every student, it is true that they have not proven Fermat’s last theorem. Or in other words,

$$\neg(\exists x FLT(x)) \equiv \forall x \neg FLT(x)$$

In general, here are the two rules to remember when it comes to negating a quantified statement

$$\neg \forall x P(x) \equiv \exists \neg P(x)$$

$$\neg \exists x P(x) \equiv \forall \neg P(x)$$