More Logic

Kevin Gold

September 12, 2017

1 Propositional Logic and Implication

Logic isn't just about circuits; it also describes a way of thinking. Consider someone saying "If it rains, I will stay inside today." Later, you look outside and see it is raining. You can conclude that the speaker will stay indoors. How? Because the speaker gave you an *implication*, or a statement that one state of affairs implies another. We could let P be the assertion that it is raining, and Q be the assertion that the speaker will stay inside. The speaker has just given you a statement of the form "If P, then Q" - if it is raining, then the speaker will stay inside. On seeing the truth of P (it's raining), you conclude Q (the person will stay inside), assuming the truth of the speaker's earlier statement. (And if you saw the speaker outside while it was raining, you'd conclude the speaker's earlier statement was false.)

Propositional logic refers to a logic in which the variables represent statements about the world that are true or false. Such statements can be combined using OR, AND, and NOT, just as abstract logical variables can be combined in this way. For example, consider the statement "I will buy either an apple or an orange, or both." If P is the statement "I will buy an apple" and Q is the statement "I will buy an orange," we could write the original statement as "$P \vee Q$".

The value in formalizing things in this way is that we can turn logical reasoning into a rules-based problem fit for a computer to solve. Very complex logic problems may be tricky to solve without some resort to symbol manipulation. "I will buy at least one of an apple, an orange, and a pear, and I will buy at least one of a pear, a pineapple, and a plum, and ..." could be written formally $(A \vee B \vee C) \wedge (C \vee D \vee E) \wedge \ldots$.... and then an algorithmic process could solve for what satisfies all the constraints. Such *satisfiability* problems, looking for what truth values satisfy a complex set of propositions, are at the heart of many complex problems in computer science. They often don't have known efficient solutions, but formalization of the reasoning is what allows a computer to tackle these problems at all.

One of the main ways to perform logical deduction is with implication - statements of the form "If P, then Q." Just as there is shorthand for "and," "or," and "not," there is shorthand for this: "If P then Q" can be written "$P \implies Q$." The arrow is read as "implies." If P is true, and $P \implies Q$ is true, you can conclude Q is true. If P is false, however, one can't assume anything about Q; maybe it's true for a different reason. For example, if P is "It is raining" and Q is "I will stay inside," and $P \implies Q$, we know for certain that I will stay inside if it's raining. But if it's not raining, maybe I'll be lazy and stay inside anyway. We can't deduce anything from "not P."

Note that "implies" doesn't necessarily mean that P *causes* Q to be true. It may be the opposite, that Q is the only logical explanation for P, and therefore if P is true, we can conclude Q. Or perhaps P and Q are mutually caused by something else.

If P implies Q and Q implies P, these facts have a very special relationship - they are always either both true or both false. We can say "P if and only if Q," which can be written in the shorthand "P iff Q" or "$P \iff Q$." If two statements logically imply each other, it's often because one is defined in terms of the other. "A shape is a square iff it has four sides of equal length and four right angles." "You will get an A in this class iff you achieve 90 percent or better with your final weighted percentage." In math, less obvious statements can be proven to have such a tight relationship with each other - always either both true, or both false.

Interestingly, the implies operator is equivalent to a statement that we can write without using the implies symbol at all; $P \implies Q$ is logically equivalent to $\neg P \vee Q$. For example, "If it is raining, I will stay indoors" can be rephrased as "Either it is not raining, or I am staying indoors." Recall that an "or" can be satisfied with three out of the four settings of the two variables it combines. The three states of affairs that satisfy the implication's "OR" are: the condition P doesn't hold and Q is false; the condition P doesn't hold but Q happens to be true anyway; and condition P holds and Q is true as a result. The one combination forbidden by the implication is that you can't have P true, but Q false - because P implies Q.

Deduction in propositional logic consists of making use of a "knowledge base" of facts and implications, and trying to deduce the truth of a particular statement using these statements and the rules of logic. For example, if statement P is "It is raining," and Q is "The baseball game is going on," and R is "there is heavy traffic near the stadium," with the rules $P \implies \neg Q$ and $Q \iff R$, I could deduce from statement P the truth of $\neg R$: if it is raining, there will not be a baseball game, and there will therefore be no traffic.

In the early years of artificial intelligence, many researchers tried to emulate human intelligence with large knowledge bases of facts and implications. This had some amount of success, but more recent approaches use probability (introduced later in this course) to address how reasoning should work when the facts are uncertain.

2 Quantifiers and First-Order Logic

Propositional logic is a bit unwieldy for making complex statements. If we want to make a statement such as "The sum of two even numbers is also an even number," we would be better able to express it in a form of logic called *first-order logic*. First order logic allows one to make generalizations about objects that do or don't have different properties.

First-order logic contains *quantifiers* that come in two types: “there exists” ($\exists$) and “for all” ($\forall$). “There exists” is a claim that there is at least one object that has a particular property, while “for all” is a claim that all objects have some property. The properties themselves are called *predicates*, and they might be best demonstrated with an example. We can write our preceding statement, “the sum of two even numbers is also an even number,” in the following way:

$$\forall x, y : Even(x) \wedge Even(y) \implies Even(x+y). \quad (1)$$

Here we have a claim that whatever *Even* means, if it’s true of both x and y, no matter what those values are, then it is also true of their sum.

As an example of “there exists,” we could claim that for every number x, there is some number y such that $x + y = 0$:

$$\forall x \exists y : x + y = 0 \quad (2)$$

The order of these quantifiers matters. If we were to make the above claim quantified with $\exists y \forall x$ (thus swapping the order of the quantifiers), we would be claiming that there exists some number y such that any number x added to it equals zero - which isn’t true of any numbers we’re familiar with. English sentences are sometimes ambiguous in how they’re quantifying things, unless you apply common sense: “Everyone has a favorite teacher” has a natural meaning of $\forall x \exists y IsFavoriteTeacher(y,x)$, so everyone’s favorite teacher is different, but the sentence could also be understood to mean $\exists y \forall x IsFavoriteTeacher(y,x)$ meaning that there’s one specific teacher who is the favorite of everybody.

When thinking about math, it can be useful to realize that mathematical theorems are often making claims that implicitly use these operators. A theorem often is claiming something exists or doesn’t exist, or that something is true of all objects or of no objects.

Notice that the claim “$\forall x, P(x)$” is the same as “$\neg \exists x, \neg P(x)$.” If something is true of all objects, then there is no object that it is not true of. Such rules are sometimes used implicitly in mathematical proofs, a subject we will return to later.