CS 2742 (Logic in Computer Science) – Lecture 1

Antonina Kolokolova*

Sep 9th, 2009

1 What is logic in Computer Science?

Why do we study mathematical logic? Natural languages (such as English) are too ambiguous:

> "Every student knows this" and "Any student knows this" have the same meaning.
>
> "She'd be happy if she passes every course" and "She'd be happy if she passes any course" mean very different things.

Mathematical logic is a kind of language, where everything is designed to have a precise meaning. This is the language of unambiguous reasoning. So this course is somewhat like a foreign language course: there will be a lot of vocabulary to memorize. Also, like in a language course, you need to practice "speaking" logic language for it to become natural.

Logic comes into computer science in many different ways:

- Digital circuit logic
- Proof of correctness of programs, verification
- Automated reasoning in artificial intelligence

*The material in this set of notes came from many sources, in particular "Discrete mathematics with applications" by Susanne Epp, several books by Raymond Smallian, course notes of U. of Toronto CS 238 by Vassos Hadzilacos.

- Database query languages
- ...

We will try to cover at least some of these applications.

2 Propositional logic

Puzzle 1 (Twins puzzle) There are two identical twins, John and Jim. One of them always lies, another always says the truth. Suppose you run into one of them and want to find out whether you met John or Jim. Which 3-word question with a yes/no answer should you ask to learn the name of the twin in front of you? You don't know which one of them is the liar.

The basic unit of our reasoning is a sentence that can have a truth value, that is it can be either true or false. We call such a sentence a *proposition*. For example, "it is raining" is a proposition. It can be either true or false at any particular point in time and space. So is "I am a dolphin", which happens to be false, as far as I know. When referring to a proposition, we often will give it a short name (a variable); for example, we can use a variable p to denote the sentence "it is raining". Now, if we say "p is true" we will mean that it is, indeed, raining. Such variables denoting propositions are called *propositional variables*.

Simple propositions can be combined to create more complicated ones. For example, we can say "if it is raining, then it must be cloudy". Here, we mean that "raining" (that is, our proposition p) implies that it is also "cloudy". If we use a propositional variable q to mean "it is cloudy" then in the language of propositional logic we say that "p implies q". Implication is one of *logical connectives* that allow us to make more complicated statements, *propositional formulas*, out of propositions. The following table lists several common logical connectives.

Meaning	**Name**	**Notation**	**Pronunciation**
if p is true then q is true	*implication*	$p \to q$	p implies q
both p and q are true	*conjunction*	$p \wedge q$	p and q
at least one of p, q is true	*disjunction*	$p \vee q$	p or q
opposite of p is true	*negation*	$\neg p$	not p

Here instead of p and q we could have whole propositional formulas which are themselves made out of propositions, logical connectives and parentheses (to specify what logical connectives apply to). For example, if we denote "I am a dolphin" by a propositional variable r, then the following is a propositional formula: $(\neg p \vee q) \wedge r$, which reads as "It is not raining or it is cloudy and, besides, I am a dolphin". Here, the precedence order is first $\neg$, then $\wedge$,

then $\vee$, and then $\rightarrow$: that is, $\neg q \rightarrow \neg p \vee q \wedge r$ is parenthesized as $(\neg q) \rightarrow ((\neg p) \vee (q \wedge r))$. It may help to think of $\neg, \wedge$ and $\vee$ as unary $-$, $*$ and $+$ in arithmetic formulas: we will show in this course that there is a deep connection between them. So parenthesizing $\neg p \vee q \wedge r$ is just like parenthesizing $-5 + 3 * 8$.

Note that "or" in mathematical logic has a bit different meaning from English "either/or": here, both propositions can be true, whereas in English we often mean "or" as an exclusive: either the first one is true, or the second, but not both. For example, "it is raining or I am a dolphin" is a perfectly valid propositional formula which is true if either it is raining, or I am a dolphin, or both it is raining and I am a dolphin.

Similarly, the implication is only false if its left hand side (i.e., p) is true while the right hand side (q) is false. That is, "if it is raining then it is cloudy" is false only when it is raining out of blue sky. If it is not raining this propositional formula is true no matter whether it is cloudy or not.

One way to think of the implication $p \rightarrow q$ is to consider all possible scenarios for the values of p and q. If both p and q are true, then $p \rightarrow q$ is true. If p is true and q is false, then $p \rightarrow q$ is false. Finally, if p is false then $p \rightarrow q$ is true both when q is true and when q is false.

We will talk later that some of the logical connectives here are "redundant": we can express the same functionality as $p \rightarrow q$, for example, using just $\wedge$ and ./That is, $p \rightarrow q$ is false only when p is true and q is false, that is, when $p \wedge \neg q$ is true. Then saying $\neg(p \wedge \neg q)$ will be false when p is true and q is false, and true everywhere else, just like $p \rightarrow q$. As an example, think of saying "if it rains it must be cloudy" as having the same meaning as "it can't happen that both it's not cloudy and raining".

Puzzle 2 (Variation of the twins puzzle) Suppose that instead of finding out the name of a twin you would want to know the name of the liar twin. Which 3-word question will give you the answer?