Logic for Computer Science - Week 1
Introduction to Informal Logic

October 21, 2018

1 Propositions

A *proposition* is a statement that can be true or false. Propositions are sometimes called *sentences*. Here are examples of propositions:

1. “I wear a blue shirt.”
2. “You own a laptop computer.”
3. “You own a laptop computer and a tablet computer, but no smartphone.”
4. “I own: a laptop, a tablet, a smartphone, a server.”.
5. “I will buy a laptop or a tablet”.
6. “I can install the software on my smartphone or on my tablet.”
7. “It is raining outside, but I have an umbrella.”
8. “If I get a passing grade in Logic, I will buy everyone beer.”
9. “I play games often and I study very well.”
10. “Snow is white.”
11. “It is raining.”
12. “It is not raining.”
13. “I will pass Logic only if I study hard.”
14. “Either white wins or black wins in a game of chess.”
15. “$2 + 2 = 4$.” (“Two plus two is four.”)
16. “$1 + 1 = 1$.” (“One plus one is one.”)
17. “$1 + 1 \neq 1$.” (“One plus one is not one.”)

18. "If $1 + 1 = 1$, then I'm a banana."
19. "All natural numbers are integers."
20. "All rational numbers are integers."

Here are examples of things that are not propositions:

1. "Red and Black." (not a statement)
2. "π." (not a statement)
3. "Is it raining?" (question, not a statement)
4. "Go fish!" (imperative)
5. "x is greater than 7." (cannot tell unless I know who x is)
6. "This sentence is false."

Sometimes it is debatable whether something is truly a proposition. For example, we generally agree that "Snow is white" is true, but someone might argue that they have seen black snow, so the status of "Snow is white" is put in question. Arguing about whether something is a proposition or not is more a matter of philosophical logic than computer science logic and we will therefore not be too concerned about these sort of issues.

2 Informal Propositional Logic

Propositional Logic is the logic of propositions linked together with *logical connectives* such as *or*, *and* and *not*. In this section, we will go over the basics of propositional logic.

2.1 Atomic Propositions

Some propositions are atomic, in that they cannot be decomposed further into smaller propositions:

1. "I wear a blue shirt."
2. "You own a laptop computer."
3. "$2 + 2 = 4$." ("Two plus two is four.")

2.2 Conjunctions

Others however seem to be composed of smaller parts. For example, the proposition "I play games often and I study very well" is composed of two smaller propositions "I play games often" and "I study very well", joined together by "and". When two propositions "φ" and "ψ" are joined by an "and", the resulting proposition "φ and ψ" is called a *conjunction* or *the conjunction of* φ *and* ψ. The propositions "φ" and "ψ" are called the *conjuncts* of the proposition "φ and ψ".

Intuitively, a conjunction is true if both of its conjuncts are true. For example, the proposition "I play games often and I study very well." is true if both "I play games often." and "I study very well." are true. In particular, as I do not play games often, this proposition is false (when I say it).

Note that a conjunction need not use explicitly the word "and". For example, the proposition "It is raining outside, but I have an umbrella." is also a conjunction, and it conjuncts are "It is raining outside" and "I have an umbrella.". This particular conjunction uses the adversative conjunction "but".

Exercise 2.1. *Find the conjuncts of "I play at home and I study at school.".*

Exercise 2.2. *Give an example of a conjunction that is false.*

2.3 Disjunctions

Disjunctions are propositions linked together by "or". For example, "I can install the software on my smartphone or on my tablet." is a disjunction between "I can install the software on my smartphone" and "I can install the software on my tablet.". The two parts of the disjunction are called the *disjuncts*.

In the example above, note that the English grammar allows us to omit "I can install the software ...", as it is implicit in our understanding of the language. However, when we find the disjuncts, is helps to state them explicitly.

Exercise 2.3. *Find the disjuncts of "I will buy a laptop or a tablet.".*

A disjunction is true if at least one of the disjuncts is true. For example, "I am Darth Vader or I teach." is true because "I teach" is true (I do not have to worry about being Darth Vader). "I teach or I program." is also true (it happens that both disjuncts are true).

This meaning of disjunctions is called the *inclusive or*. It is standard in mathematics. Sometimes people use "or" in natural language to mean *exclusive or*. For example "Either white wins or black wins in a game of chess." is an example where the "or" is exclusive. The meaning of the sentence is that "white wins" or "black wins", but not both. Here is an example of a false proposition that uses "exclusive or": "Either I program or I teach.". When you see "either" in a sentence, it is a sign that you are dealing with an "exclusive or". In this case, do not say that it is a disjunction.

Exercise 2.4. *Give an example of a false disjunction.*

Exercise 2.5. *When is a disjunction "φ or ψ" false?*

2.4 Implications

Implications are propositions of the form "if φ then ψ". The proposition φ is called the *antecedent* and the proposition ψ is called the *conclusion* of the implication.

An example of an implication is "If I get a passing grade in Logic, I will buy everyone beer.". The antecedent is "I get a passing grade in Logic." and the conclusion is "I will buy everyone beer.". When is an implication true? Actually, it is easier to say when it is false. An implication is false if and only if the antecedent is true but the conclusion is false. Assume that I got a passing grade in Logic. Therefore, the proposition "I get a passing grade in Logic" is true. However, I will not buy beer for everyone (just a few select friends). Therefore the proposition "I will buy everyone beer" is false. Therefore the implication "If I get a passing grade in Logic, I will buy everyone beer." as a whole is false (antecedent is true, but conclusion is false).

The meaning of implications is worth a more detailed discussion as it is somewhat controversial. This is mostly because implication as we understand it in mathematics can sometimes be subtly different from implication as we understand it in everyday life. In everyday life, when we say "If I pass Logic, I buy beer.", we understand that there is a cause-and-effect relation between passing Logic and buying beer. This subtle cause-and-effect relation is evident in a number of "if-then" statements that we use in real life: "If I have money, I will buy a car.", "If you help me, I will help you", etc. We would never connect two unrelated sentences with an implication: the proposition "If the Earth is round, then 2+2=4." would not be very helpful, even though it is true (both the antecedent and the conclusion are true).

This implication is called *material implication* or *truth functional implication*, because the truth value of the implication as a whole depends only on the truth values of the antecedent and the conclusion, not on the antecedent and the conclusion itself. This meaning of implication sometimes does not correspond to the meaning of natural language implications, but it turns out that it is the only sensible interpretation of implications in mathematics (and computer science).

In particular, we will take both the propositions "If the Earth is flat, then $2 + 2 = 5$." and "If the Earth is flat, then $2 + 2 = 4$." to be true, because the antecedent is false. Implications that are true because the antecedent is false are called *vacuously true*.

Exercise 2.6. *What are the truth values of "If $2 + 2 = 4$, then the Earth is flat." and "If $2 + 2 = 5$, then the Earth is flat."?*

The truth value of an implication "if φ then ψ" depending on the truth values of its antecedent φ and its conclusion ψ is summarized in the truth-table below:

φ	ψ	if φ then ψ
false	*false*	*true*
false	*true*	*true*
true	*false*	*false*
true	*true*	*true*

The following example aims at convincing you that the truth table above is the only good one. You must agree that every natural number is also an integer. Otherwise put, the proposition "for any number x, if x is a natural, then x is an integer" is true. In particular, you will agree that the proposition above holds for $x = -10$, $x = 10$ and $x = 1.2$. In particular, the propositions "If -10 is a natural, then -10 is an integer.", "If 10 is a natural, then 10 is an integer." and "If 1.2 is a natural, then 1.2 is an integer." must all be true. This accounts for the first, second and fourth lines of the truth table above (typically, the second line is controversial). As for the third line, false is the only reasonable truth value for an implication "if φ then ψ" where φ is true but ψ is false. Otherwise, we would accept propositions such as "If $2+2=4$, then $2+2=5$." (antecedent $2+2=4$ true, conclusion $2+2=5$ false) as being true.

Implications are sometimes subtle to spot and identify correctly. For example, in the proposition "I will pass Logic only if I study hard.", the antecedent is "I will pass Logic" and the conclusion is "I study hard". In particular, the above proposition does not have the same meaning as "If I study hard, then I will pass Logic.".

Implications can sometimes not make use of "if". For example, take the proposition "I will pass Logic or I will drop school.". This proposition can be understood as "If I do not pass Logic, then I will drop school".

2.5 Negations

A proposition of the form "it is not the case that φ" is the *negation* of φ. For example, "It is not raining." is the negation of "It is raining.". The negation of a proposition takes the opposite truth value. For example, as I am writing this text, the proposition "It is raining." is false, and therefore the proposition "It is not raining." is true.

The words "and", "or", "if-then", "not" are called *logical connectives*, as they can be used to connect smaller propositions in order to obtain larger propositions.

Exercise 2.7. *Give an example of a false proposition that uses both a negation and a conjunction.*

2.6 Equivalence

A proposition of the form "φ if and only if ψ" is called an *equivalence*. Such a proposition, as a whole, it true if φ and ψ have the same truth value (both false or both true).

For example, when I am writing this text, “It is raining if and only if it is snowing” is true. Why? Because both of the propositions “It is raining.” and “It is snowing.” are false.

Exercise 2.8. *What is the truth value of the proposition “The number* 7 *is odd if and only if* 7 *is a prime.”?*

3 Ambiguities in Natural Language

We have described informally the language of propositional logic: atomic propositions connected with and, or, not, etc. So far, our approach has used English. However, English (or any other natural language) is not suitable for our purposes because it exhibits imprecisions in the form of ambiguitites.

Consider the following argument:

1. “John and Mary are married.”
2. “Mary is a student.”
3. So, “John’s wife is a student.”

Is the argument valid? It depends on how you resolve the ambiguity of the first sentence. The first sentence could mean any of the following two things:

1. that John and Mary are married to each other;
2. or that John and Mary are married, but not necessarily to each other (they could be married to unnamed 3rd parties).

In the first interpretation, the argument is valid, while in the second interpretation, it is not. In the study of the laws of logic, such ambiguities can get in the way, just like a wrong computation could impact the resistance of buildings or bridges in civil engineering.

The state of the art in Logic for over 2.000 years, from Aristotle up to the development of Symbolic Logic in the 19th century, has been to use natural language. Symbolic logic (formal logic) has changed the game by introducing languages so precise that there is no risk of misunderstandings.

The first such language that we study will be the language of propositional logic.

3.1 On the Use of Natural Language

As we have seen in the previous lecture, natural language (English, French, Romanian, etc.) features inherent ambiguities. We have already seen an example of such an ambiguity: “John and Mary are married” could mean either that they are married to each other or that they are married, not necessarily to each other.

We are now anticipating first-order logic (discussed during the second half of the course).

When we will get to first-order logic, we will see that we can disambiguate between the two possible meanings of the sentence by:

- either using a two-argument predicate "MarriedToEachOther(X, Y)" that means that "X" and "Y" are married to each other and writing the sentence as "MarriedToEachOther(John, Mary)";
- or using a single-argument predicate "Married(X)" that means that "X" is married (without saying who "X" is married to) and writing the sentence as "Married(John) and Married(Mary)".

End of divagation into first-order logic.

Such ambiguities also occur in (informal) propositional logic. Take the sentence "It is not true that John is tall and Jane is short." It could either mean that the conjunction "John is tall and Jane is short" is not true, or simply that "John is not tall and Jane is short".

Such ambiguities impede the study of propositional logic. Therefore we will design a *formal language*, the language of propositional logic, where no ambiguity can occur.

Normally, when people say "formal", they mean it in a bad way, such as having to dress or act formally to go to dinner.

However, in computer science (and in mathematics), formal is a good thing: it means making things so precise that there is no possibility of misunderstanding.

4 The Formal Syntax of Propositional Logic

Propositional formulae will be strings (sequences of characters) over the alphabet of propositional logic.

The alphabet of propositional logic is simply a set of symbols. The symbols are categorised as follows:

1. $A = \{\mathtt{p}, \mathtt{q}, \mathtt{r}, \mathtt{p}', \mathtt{q}_1, \ldots\}$ is an infinite set of *propositional variables* that we fix from the very beginning;
2. $\{\neg, \wedge, \vee\}$ is the set of *logical connectives*;
3. $\{(,)\}$ is the set of auxiliary symbols; in our case, it consists of two symbols called *brakets*.

The set $L = A \cup \{\neg, \wedge, \vee, (,)\}$ is called the alphabet of propositional logic. We call a set an alphabet in computers science when we will use the elements of the set to make words.

Here are a few examples of words over L: $))\mathtt{p} \vee \wedge$, $\vee \vee \neg(\mathtt{p})$, $\neg(\mathtt{p} \vee \mathtt{q})$. Words, or strings, are simply sequences of symbols of the alphabet L. Some of these words will be formulae or, equivalently, well-formed formulae (wff). Some authors prefer to use the terminology "wff", but we will simply use "formula" by default in these lecture notes.

As an example, the last word above, $\neg(\mathtt{p} \vee \mathtt{q})$ is a formula of propositional logic, but $\vee \vee \neg(\mathtt{p})$ is not. The following definition captures exactly the set of propositional formulae.

Definition 4.1 (The Set of Propositional Formulae (PL))**.** *The set of formulae of propositional logic, denoted PL from hereon, is the only set of word over L satisfying the following conditions:*

1. *(Base Case) Any propositional variable, seen as a 1-symbol word, is in PL (equivalently, $A \subseteq PL$);*
2. *(Inductive Step i) If $\varphi \in PL$, then $\neg\varphi \in PL$ (equivalently, if the word φ is a propositional formula, then so is the word starting with the symbol $\neg$ and continuing with the symbols in φ);*
3. *(Inductive Step ii) If $\varphi_1, \varphi_2 \in PL$, then $(\varphi_1 \vee \varphi_2) \in PL$ (equivalently, exercise);*
4. *(Inductive Step iii) If $\varphi_1, \varphi_2 \in PL$, then $(\varphi_1 \wedge \varphi_2) \in PL$ (equivalently, exercise);*
5. *(Minimality Constraint) No other word (other than those constructible using the rules above) is in PL.*

Here are examples of elements of PL:

$$\mathtt{p} \qquad \mathtt{q} \qquad \neg\mathtt{p} \qquad \neg\mathtt{q} \qquad \neg\mathtt{p}' \qquad \neg\neg\mathtt{p}_1 \qquad (\mathtt{p} \vee \mathtt{q}) \qquad (\mathtt{p} \wedge \mathtt{q}) \qquad \neg(\mathtt{p} \vee \mathtt{q})$$

$$(\neg\mathtt{p} \wedge \neg\mathtt{q}) \qquad \neg(\neg\neg\mathtt{p} \vee \mathtt{p}) \qquad ((\mathtt{p} \vee \mathtt{q}) \wedge \mathtt{r}) \qquad (\mathtt{p} \vee (\mathtt{q} \wedge \mathtt{r}))$$

Here are examples of words not in PL:

$$\mathtt{pp} \qquad \mathtt{q}\neg\mathtt{q} \qquad \mathtt{q} \wedge \neg\mathtt{p} \qquad \mathtt{p} + \mathtt{q}$$

The definition of the set PL is an example of an *inductive definition.* Such definitions are really important in computer science and it is a must to understand them very well. In inductive definitions of set, there are usually some base cases, which say what "base" elements are part of the set and some inductive cases, which explain how to obtain new elements of the set from old elements. Another important part of an inductive definition is the minimality constraint, which says that nothing other than what is provable by the base case(s) and the inductive case(s) belongs to the set. It can be shown that the above set PL exists and is unique, but the proof is beyond the scope of this course.