ITCS 531: Logic 1 - Semantics for propositional formulas

Rob Egrot

The logic of mathematical proofs

- Formal mathematical arguments:
  - Start with assumptions (axioms).
  - Sequence of logical deductions.
  - Desired conclusion.
- This axiom-theorem-proof style dates back to the Ancient Greeks, e.g. Euclid (around 300 BCE).
- This neat picture of mathematics does not really correspond to how mathematicians actually work as.
- Mathematicians use a lot of informal intuition.
- The modern style of being very explicit about assumptions and definitions started in the late 19th century.

The evolution of rigour

- As mathematics became more advanced, mathematicians started proving contradictory things (e.g. in calculus).
- To resolve this, mathematicians became very precise.
- By doing this they were able to see that the contradictions often came from people starting from different assumptions.
- Imre Lakatos' book *proofs and refutations* explores this.
- The formal style is not how mathematicians *think*, but it is important to communication.
- It also helps prevent logic errors in mathematical reasoning.
- In practice: Think informally, write formally.
  - Often this is taken too far!

The role of formal logic

- It is debatable whether the formal style captures the true essence of mathematics.
- But, mathematics should be, in principle, capable of being expressed as a formal procession of axioms and deductions.
- I.e., mathematics can be treated as a formal system.
- So we can use mathematical reasoning on mathematics!
- Mathematics about mathematics (metamathematics).
  - Spoiler: This leads to the development of computers, which we will study next semester.
- Computers also behave very much like formal systems...

Applying logic

- But, before we can understand the role of formal logic in:
  - The theory of computation.
  - Analysis of the difficulty of computation problems.
  - Understanding the behaviour of software.
  - Etc.
- We need to understand the basics.
- That is what this course is about.

What is logic anyway?

- To properly describe mathematical and computational ideas symbolically we will need a complex language.
- We will get to this later in the course.
- First, we can think about the abstract structure of logical arguments with a relatively simple formal system.
- The Ancient Greeks thought a lot about this.
- For example, Aristotle gave the following example of a logical deduction:
  1. *All humans are mortal.*
  2. *All Greeks are human.*
  3. *Therefore, all Greeks are mortal.*

The syllogism

- Aristotle's argument again:
  1. *All humans are mortal.*
  2. *All Greeks are human.*
  3. *Therefore, all Greeks are mortal.*
- This is an example of something call a *syllogism*.
- The conclusion here is true in reality, but also, it *has* to be true if the assumptions are true:
  1. *All X have property Y.*
  2. *Z is X.*
  3. *Therefore, Z has property Y.*
- Medieval Christian scholars studied syllogisms in great detail.
- But for us syllogisms are not enough.

Propositional logic

- For our formal system of propositional logic we need three things:
  - Basic propositions (AKA *propositional variables*), $\{p_0, p_1, p_2, \ldots\}$. Abstract true/false statements.
  - Logical connectives $\{\wedge, \vee, \neg, \rightarrow, \leftrightarrow\}$.

    $\wedge$: $p \wedge q$ is supposed to mean "p and q".

    $\vee$: $p \vee q$ is supposed to mean "p or q".

    $\neg$: $\neg p$ is supposed to mean "not p".

    $\rightarrow$: $p \rightarrow q$ is supposed to mean "p implies q" specifically, *material implication*.

    $\leftrightarrow$: $p \leftrightarrow q$ for mutual implication.
  - Brackets (and). We use these to delimit formulas.

Giving meaning to propositional statements

- If we assign meaning to some of the basic propositions we can combine them into new statements using the logical connectives and brackets.

Example 1

- Let $a, b, c \in \mathbb{N}$, and suppose p means "$a|b$", q means "$a|(b+c)$", and r means "$a|c$".
- Then $(p \wedge q) \rightarrow r$ means "If a divides b, and a divides $(b+c)$, then a divides c".
- This statement is true, which we proved in the number theory class.

Another example

Example 2

- Again let $a, b, c \in \mathbb{N}$.
- Suppose p means "$a|b$", q means "$a|c$", and r means "$a|bc$".
- Then $(p \wedge q) \leftrightarrow r$ means "a divides b, and a divides b, if and only if a divides bc".
- This is not true (why?).
- The 'only if' part is true, but the 'if' part is not (though it looks similar to a true statement).

What is a 'formula'?

- Not every string we can make using basic propositions and logical connectives makes sense.

Example 3

$(p \to \wedge q) \vee \neg r$ doesn't make sense, whatever meaning we give to p, q and r. It's not true or false, it just doesn't mean anything.

- A **well-formed formula** (WFF) is a string in propositional logic that is capable of making sense.
- A recursive definition:
  - Individual basic proposition symbols are well-formed formulas.
  - If ϕ is well-formed then $\neg\phi$ is well-formed.
  - If ϕ and ψ are well-formed then $(\phi * \psi)$ is well-formed for all $* \in \{\wedge, \vee, \to, \leftrightarrow\}$.
  - Everything else is not well-formed.
  - (We sometimes cheat with the brackets).

Subformulas

- In propositional logic we often refer to well-formed formulas as just *formulas*, and sometimes as *sentences*.
- If ϕ is a formula, then a **subformula** of ϕ is a substring of ϕ that is also a sentence (i.e. can be obtained by our recursive construction).
- E.g. $(p \wedge q)$ is a subformula of $(p \wedge q) \to r$, and so is e.g. r.
- We consider ϕ to be a subformula of itself.
- We define the **length** of a sentence ϕ to be the number of logical connectives that occur in ϕ.
- E.g. if $\phi = \neg((p \vee q) \wedge q)$ then the length of ϕ is 3.

When is a sentence true?

- Every basic proposition must be either true or false, and cannot be both.
- The same applies to sentences.
- Whether a sentence is true or false is depends only on the true/false values of the basic propositions it is built from.
- This truth value can be calculated recursively from the truth values of the basic propositions.
- We use **truth tables** to represent the recursion rules.

Truth tables

¬:

ϕ	$\neg\phi$
T	F
F	T

∧:

ϕ	ψ	$\phi \wedge \psi$
T	T	T
T	F	F
F	T	F
F	F	F

∨:

ϕ	ψ	$\phi \vee \psi$
T	T	T
T	F	T
F	T	T
F	F	F

→:

ϕ	ψ	$\phi \to \psi$
T	T	T
T	F	F
F	T	T
F	F	T

↔:

ϕ	ψ	$\phi \leftrightarrow \psi$
T	T	T
T	F	F
F	T	F
F	F	T

Using truth tables

Example 4

p	q	r	$p \wedge q$	$(p \wedge q) \to r$
T	T	T	T	T
T	T	F	T	F
T	F	T	F	T
T	F	F	F	T
F	T	T	F	T
F	T	F	F	T
F	F	T	F	T
F	F	F	F	T

Implication

- Truth table for $\to$ gives the *material conditional*.
  - $\phi \to \psi$ is true whenever ϕ is false or ψ is true.
- Contrast with *subjunctive implication*:
  - E.g: "if I dropped it, then it would break".
  - Should be true for a chicken egg, false for a tennis ball.
  - According to material implication, this is true for tennis balls and eggs if I don't drop them!
- Consider also *indicative implication*:
  - E.g. suppose a student likes yoghurt, studies hard and does well in her exams.
  - "studied hard so did well in her exams" is probably true.
  - "likes yoghurt so did well in her exams" is probably false.
  - I.e. the first part should be relevant to the second part.
  - Material conditional says both are true.
- So, material implication is not appropriate for everything.
- But it is good for formal systems.
  - I.e. if ϕ is false then $\phi \to \psi$ makes no claim, so is true.

Satisfaction

- Setting every propositional variable to be either true or false is making a **truth assignment** (or just **assignment**).
- If a sentence is true under some assignment we say it is **satisfied** by that assignment.
- A sentence is **satisfiable** if there is some assignment that satisfies it.
  - I.e. if there is a way we can interpret each basic proposition as true or false so that the whole thing becomes true.
- If Γ is a set of sentences, then we say Γ is satisfiable if there is an assignment that satisfies every sentence in Γ.

Tautologies and contradictions

- A sentence that is satisfied by every assignment is called a **tautology**.
  - I.e. a tautology is something that is always true.
  - E.g. $p \vee \neg p$ (a proposition must be either true or false).
  - Warning: in some logic systems this is not something we can assume! (see the next class).
  - We sometimes use the symbol $\top$ to denote a tautology.

- A sentence that is not satisfiable is called a **contradiction**.
  - I.e. a contradiction can never be true.
  - E.g. $p \wedge \neg p$ (a proposition cannot be both true and false at the same time).
  - We sometimes use the symbol $\perp$ to denote a contradiction.

- If ϕ is a tautology then $\neg\phi$ is a contradiction, and vice versa.

Logical implication

- Given ϕ and ψ, we say that ϕ **logically implies** ψ if whenever an assignment satisfies ϕ, it also satisfies ψ.
- We also say ψ is a **logical consequence** of ϕ.
- We write $\phi \models \psi$.
- This is another way of saying that $\phi \to \psi$ is true.
- We say that ϕ and ψ are **logically equivalent** if each is a logical consequence of the other.
- In this case we write $\phi \models \mathrel{=}\!\mid \psi$.

Theories

- We can also do this with sets of sentences.
- If Γ is a set of sentences and ψ is a sentence, then ψ is a logical consequence of Γ if, whenever an assignment satisfies ϕ for all $\phi \in \Gamma$, it also satisfies ψ.
- We write $\Gamma \models \psi$.
- We sometimes call a set of sentences a **theory**, and then we might say that ψ is a consequence of the theory Γ.
- We want to be able to say things like "the fact that the set of primes is infinite is a consequence of the theory of numbers".
- A theory can be empty (i.e. have no members).
- We write $\models \phi$ if ϕ follows from the empty theory, i.e. if ϕ is a tautology.

A simple theory example

- Let Γ be the theory $\{p \wedge \neg q, q \vee r\}$.
- Then $r \wedge p$ is a logical consequence of Γ (i.e. $\Gamma \models p \wedge r$).

p	q	r	$p \wedge \neg q$	$q \vee r$	$r \wedge p$
T	T	T	F	T	T
T	T	F	F	T	F
T	F	T	T	T	T
T	F	F	T	F	F
F	T	T	F	T	F
F	T	F	F	T	F
F	F	T	F	T	F
F	F	F	F	F	F

- There is only one assignment that makes both $p \wedge \neg q$ and $q \vee r$ true, and that is $p = r = T$, $q = F$.
- $r \wedge p$ is also true with this assignment.
- So, every assignment that makes everything in Γ true must also make $r \wedge p$ true, which means $\Gamma \models r \wedge p$.

Avoiding truth tables

- We could also work this out without writing out the whole truth table.
  - E.g. we might notice just by looking at the formulas that $p \wedge \neg q$ being true means p is true and q is false.
  - Then $q \vee r$ can only be true if r is true.
  - Which means $r \wedge p$ must be true too.
- For complicated formulas, writing out a truth table is quite a lot of effort.
- So it's usually a good idea to look at the formulas first and see if you can find a quick argument for why one thing logically implies another.
- But, if you get stuck, the option of working out the truth table is always there.

Eliminating $\to$

- The set $\{\wedge, \vee, \neg, \to, \leftrightarrow\}$ is bigger than we need.
- We can use truth tables to check that some of the connectives can be reproduced using combinations of different ones.

Lemma 5

If ϕ and ψ are sentences, then $\neg\phi \vee \psi \models\Dashv \phi \to \psi$.

Proof.

ϕ	ψ	$\phi \to \psi$	$\neg\phi \vee \psi$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

We can see that the last two columns are the same. □

Why is this important?

- What this means is that whenever $\phi \to \psi$ appears in a formula, we could replace it with $\neg\phi \vee \psi$ without changing the truth value of the formula.
- In other words, we don't really need the connective $\to$, because for every formula involving $\to$ there's an equivalent one where it does not appear.
- This might be intuitively obvious, but we will provide proof now, as the proof method will be very important.

The formal version

Corollary 6

If ϕ is a sentence, then there is a sentence ϕ' where the symbol $\to$ does not occur, and with $\phi \models\, \Dashv \phi'$.

Proof.

- We induct on the length of ϕ.
- Base case: $n = 0$, so ϕ is a basic proposition. Define $\phi = \phi'$.
- Inductive step: suppose true for length n, and let ϕ have length $n + 1$. There are three cases:
  1. $\phi = \neg\psi$ for some ψ.
     - ψ has length n, so there is ψ' that does not contain '$\to$' and with $\psi \models\, \Dashv \psi'$.
     - Define $\phi' = \neg\psi'$ to complete the proof, as, since $\psi \models\, \Dashv \psi'$ we must have $\neg\psi \models\, \Dashv \neg\psi'$, and $\phi = \neg\psi$.
  2. $\phi = \psi_1 * \psi_2$ for some $* \in \{\wedge, \vee, \leftrightarrow\}$.
     - Define $\phi' = \psi_1' * \psi_2'$.
  3. $\phi = \psi_1 \to \psi_2$.
     - By Lemma 5 we can define $\phi' = \neg\psi_1' \vee \psi_2'$.

Functional completeness

- A set of connectives defined by truth tables, S, is **functionally complete** if every formula that can be constructed from $\{\wedge, \vee, \neg, \rightarrow, \leftrightarrow\}$ is logically equivalent to one constructed from S.

Proposition 7

$\{\wedge, \vee, \neg, \leftrightarrow\}$ *is functionally complete.*

Proof.

This is what we showed in corollary 6. $\square$

- The proof of corollary 6 generalizes to other connectives.
- If any sentence containing a particular connective is equivalent to another not containing it, then we still have a functionally complete set of connectives if we eliminate it.
- We will see in the exercises that $\{\wedge, \neg\}$ is functionally complete, and the same is true for $\{\vee, \neg\}$.