LIN241, Winter 2021
Week 2 summary

Note: this covers a bit more than what was discussed in our second lecture (we stopped on slide 48). The contents of slides 49 to 61 will be practiced in tutorial and will be discussed in our third lecture.

1. Lexicon and syntax of propositional logic

(1) Propositional logic (PL) studies the logical relations between certain kinds of **formulas**. The formulas that PL studies have either no internal structure (in which case they are called atomic formulas), or else they are formed by combining atomic formulas with connectives.

(2) The connectives that we will use in our version of PL are **negation** (~), **conjunction** (&), **disjunction** (∨), **material implication** (→) and the **biconditional** (↔).

(3) You should remember this vocabulary, and you should also remember how to define the lexicon and syntax of PL:

(4) Lexicon of PL:

An infinite set of atomic formulas: p, q, r, s, t, p′, p′′, …

A finite set of connectives: ~, &, ∨, →, ↔

Parentheses: (,)

(5) Syntax of PL:

1. Any atomic formula is itself a well-formed formula (wff)

2. If φ is a wff, then ~φ is a wff

3. If φ and ψ are wff, then (φ & ψ), (φ ∨ ψ), (φ → ψ), (φ ↔ ψ) are wff

4. Nothing else is a wff.

(6) Comment: in (5), φ and ψ are metavariables over well-formed formulas of PL. Rules 2 and 3 tell you that if you substitute φ and ψ by well-formed formulas (like p, ~p or p&q), you will get a well-formed formula as a result.

2. Semantics of PL

(7) The semantics of PL tells us how to calculate the truth-value of a complex formula based on its syntactic structure, and on the truth-value of its atomic parts (the atomic formulas that make up the complex formula).

(8) The semantics of PL is defined by giving truth tables for each connective in the lexicon.

(9) Negation:

p	$\sim p$
T	F
F	T

(10) Conjunction:

p	q	$p \,\&\, q$
T	T	T
T	F	F
F	T	F
F	F	F

(11) Disjunction:

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

(12) Material Implication (if … then ...):

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

(13) Biconditional:

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

(14) Comment: in these truth tables, the columns on the left side indicate the truth values of the atomic formulas that make up the complex formula in the right column. The right column gives you the truth value of that complex formula.

Each row is called a valuation. A valuation is a way to give a truth value to the atomic formulas in the table.

Instead of using the letters T for true and F for false, we could use 1 for true and 0 for false. Another commonly used symbol for false is ⊥.

(15) Using these basic truth tables, we can build truth tables for more complex formulas. To do so, we must break down the complex formulas into their immediate parts, then we must break down these parts into their immediate parts and continue until we reach atomic formulas.

To illustrate, consider the formula:

- r & (~p → q)

This a conjunction of two formulas:

- r
- ~p → q (we omit the outer parentheses since this does not result in ambiguity)

The second of these is itself made up of two formulas:

- ~p
- q

And ~p is itself made up only of negation and the atomic formula:

- p

When building the truth table for r & (~p → q), we create columns for each formula, going from left to right by order of increasing complexity. This is illustrated in the following table.

p	q	r	~ p	~ p → q	r & (~ p → q)
T	T	T	F	T	T
T	T	F	F	T	F
T	F	T	F	T	T
T	F	F	F	T	F
F	T	T	T	T	T
F	T	F	T	T	F
F	F	T	T	F	F
F	F	F	T	F	F

3. Logical properties of formulas, and logical relation between formulas

(16) A **contradiction** is a formula that is false in all valuations. A simple example is:

- p & ~p

If you draw the truth table for p & ~p, you will see that the truth value that you get in the right column is F, in all the rows of the table.

(17) A **tautology** is a formula that is true in all valuations. A simple example is:

- p ∨ ~p

If you draw the truth table for p ∨ ~p, you will see that the truth value that you get in the right column is T, in all the rows of the table.

(18) Two formulas are **equivalent** if and only if they have the same truth value in all valuations. A simple example is given by the two following formulas:

- ~(p & q)
- ~p ∨ ~q

If you draw a truth table for these two formulas, you will see that in each valuation, the truth values of these two formulas are identical.

4. Arguments

(19) An argument is a sequence of formulas where the last formula is claimed to follow from the preceding formulas, which are called the premises. An example is given in (20).

(20) $p \to q$ [a premise]
p [another premise]
$\therefore q$ ['∴' indicates the conclusion]

(21) An argument is valid just in case there is no valuation in which all the premises are true and the conclusion is false.

(22) In order to determine whether an argument is valid, you can draw a combined truth table for the premises and the conclusion, leaving the conclusion in the last column.

Then, you check whether there is at least one row in which all the premises are true, but the conclusion is false. If there is, the argument is not valid. If there isn't, the argument is valid.

Here is an example with the valid argument in (20)

p q	p → q	p	q
T T	T	T	T
T F	F	T	F
F T	T	F	T
F F	T	F	F

5. Translating sentence of English into PL

(23) Translating sentences of English into PL is an art. We can provide general guidelines, but you will have to use your intuition.

In general, keep in mind that the translation of an English sentence into PL must preserve two properties of the sentence:

1. The sentence and its translation must have the same truth-conditions.

2. The clausal connectives in the sentence must be represented in its translation into PL.

The first condition is imperative. If it is not satisfied, the translation is plainly incorrect. The second translation is just preferred for clarity. In many cases we can find equivalent translations of a sentence into PL, with different numbers and types of connectives. We will favor the translation that is syntactically most similar to the original sentence.

Another issue with 2 is that there are many connectives of English and other languages whose meaning cannot be fully captured in PL because they are not entirely truth-functional.

Remember that a connective is truth-functional if it is interpreted as a function that maps (pairs of) truth values to truth values.

To illustrate, 'although' has a truth-functional meaning that is the same as conjunction, but it also conveys a form of concession, which cannot be represented truth-functionally in PL.

The way we address this problem when we do translations into PL is that we pick the connective of PL that captures the truth-functional meaning of the English connective, and we ignore aspects of meaning that are not truth-functional.

(24) See tutorial the lecture slides and tutorial exercises for examples of translations of English sentences into PL.

(25) General guidelines for translating sentences of English into PL:

1. identify simple clauses without connectives

2. write a small dictionary (a key) in which each simple clause is associated with an atomic statement of PL

3. replace the simple clauses by their translation as atomic statements of PL

4. translate any connectives into PL

5. represent the order in which the atomic statements combine with connectives using parentheses

(26) Example:

Sentence: If it snows, it is cold.

Key: s = it snows
c = it is cold

Translation: $s \rightarrow c$