EIDMA

Introduction to Discrete Mathematics

Instructor Tomasz Traczyk, PhD.
Email: t.traczyk@mini.pw.edu.pl, Tomasz.Traczyk1@pw.edu.pl
website: www.mini.pw.edu.pl/~tomtracz
consultations: on Zoom, by appointment via email

LECTURE 1.

- Propositions, logical connectives.
- Laws of propositional calculus.

A *statement* or *proposition* is a declarative sentence, one with a fixed "logical value" (meaning it is true or false - even if we do not know which). "The Universe is infinite" is a statement even though we do not know if true. On the other hand "a man is an idiot" is not because we do not know who the man is. The word "man" plays here the part of a *variable*. For some instances of "man" it may be true, for other it may be false. A statement is not supposed to contain (free) variables. If we say "every man is an idiot" it is different story. The variable "man" is *bound* by the quantifier "every". Hence "every man is an idiot" is a statement, hopefully false.

Comprehension.

Construct other examples of non-statements.

Logical connectives

Formal logic is about how the truth value of a compound statement (called *formulas* or *formulae*) depends on truth values of its components. We form compound statements from elementary ones using so called *logical connectives*. To simplify the notation we use letters (*p*, *q*, *r* etc.) to represent elementary statements much like letters (*a*, *b*, *x*, *y*) to represent unspecified numbers in algebra. The letters are referred to as *propositional variables* (they stand for propositions just like numerical variables stand for numbers). In logic, in place of arithmetic operations of addition or multiplication we use *logical connectives*, AND, OR, IF … THEN and the like. We use constants 0 and 1 to denote logical *false* and *true*, respectively.

Definition. (Logical connectives)

Conjunction, AND, ∧		
p	q	p∧q
0	0	0
0	1	0
1	0	0
1	1	1

Disjunction, OR, ∨		
p	q	p∨q
0	0	0
0	1	1
1	0	1
1	1	1

Negation, NOT, ~, ¬	
p	~p
0	1
1	0

Implication, IF…THEN, ⇒ This one is tricky!		
p	q	p⇒q
0	0	1
0	1	1
1	0	0
1	1	1

Double implication, IF AND ONLY IF, IFF ⇔		
p	q	p⇔q
0	0	1
0	1	0
1	0	0
1	1	1

Notice that conjunction, disjunction and both implications take two arguments each (like addition or subtraction). Such operations are called *binary operations*. The negation takes only one (like assigning the opposite to a number, $a \to -a$). Such operations are called *unary*, for obvious reason.

Comprehension.

How many different binary operations can we define? Or is this number infinite?

Logical equivalence

Formal logic (propositional calculus) is constructed in the image of algebra. So far, we have seen constants (0 and 1 or *true* and *false*, variables and operations. What we lack is a way to compare our formulas. In algebra we use "=" to denote that, whatever numbers we replace the variables with, we get the same result on the left- and on the right-hand side of =. In formal logic "=" means that the two formulas are *identical*, symbol for symbol, so it is not particularly exciting.

The analogue of the relation of equality is the relation of "logical equivalence", ≡".

Definition.
$\varphi(p_1, p_2, \dots, p_k) \equiv \psi(p_1, p_2, \dots, p_k)$ if and only if for every assignment of zeroes and ones to $p_1, p_2, \dots, p_k$ the resulting logical values of φ and ψ are equal. We say then that φ and ψ are *logically equivalent.*

Remark.
Logical equivalence should not be confused with double implication, even though they are closely related. If you write $\varphi(p_1, p_2, \dots, p_k) \Leftrightarrow \psi(p_1, p_2, \dots, p_k)$ you have just created a new formula out of φ and ψ. If you write $\varphi(p_1, p_2, \dots, p_k) \equiv \psi(p_1, p_2, \dots, p_k)$ you state that for every assignment of zeroes and ones …etc. The result of the first is a new formula, the result of the second is *true* or *false*.

Logical equivalence can be used to define logical connectives. For example, the table

Conjunction, AND, $\wedge$		
p	q	$p\wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

can be replaced with: ($0\wedge 0\equiv 0$, $0\wedge 1\equiv 0$, $1\wedge 0\equiv 0$ and $1\wedge 1\equiv 1$).

Instead, one can say that the conjunction of two statements is true if and only if both statements are true. Similar tricks can be done for other connectives.

Checking logical equivalence of formulas

The question in formal logic is: "Given two formulas, are they logically equivalent or not".

For example: Is $p \Rightarrow (q \land r)$ equivalent to $(p \Rightarrow q) \land (p \Rightarrow r)$?

Truth table (brutal force method):

p	q	r	$q \land r$	$p \Rightarrow (q \land r)$	$p \Rightarrow q$	$p \Rightarrow r$	$(p \Rightarrow q) \land (p \Rightarrow r)$
0	0	0	0		1	1	
0	0	1	0		1	1	
0	1	0	0		1	1	
0	1	1	1		1	1	
1	0	0	0		0	0	
1	0	1	0		0	1	
1	1	0	0		1	0	
1	1	1	1		1	1	

Another way of doing this (more subtle and more fun) is to develop a number laws for propositional calculus, much like properties of arithmetic operations in algebra, and attempt to transform one formula into the other. The laws themselves must be verified by truth tables.

Theorem.

For every three propositions *p,q,r*

commutativity law:

1. $p \wedge q \equiv q \wedge p$, $p \vee q \equiv q \vee p$
2. associativity law:
   $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$, $p \vee (q \vee r) \equiv (p \vee q) \vee r$
3. distributivity law:
   $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$,
   $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
4. excluded middle: $p \vee \sim p \equiv 1$
5. contradiction: $p \wedge \sim p \equiv 0$
6. absorption: $p \wedge (p \vee q) \equiv p$, $p \vee (p \wedge q) \equiv p$
7. double negation: $\sim(\sim p) \equiv p$

Comprehension.

Verify the last theorem using the truth-table method.

Comments.

Putting q=$\neg p$ in $p \wedge (p \vee q) \equiv p$, we get $p \wedge (p \vee \neg \mathrm{p}) \equiv p$ and from the *excluded middle* we get $p \wedge 1 \equiv p$, the *identity law for conjunction*.

Putting q=1 in $p \vee (p \wedge q) \equiv p$ we get $p \vee (p \wedge 1) \equiv p$. Combining this with the freshly developed *identity law* we get $p \vee p \equiv p$, the *simplification law for disjunction.*

Comprehension.

In a similar way develop the *identity law for disjunction*, $p \vee 0 \equiv p$ and the *simplification law for conjunction*, $p \wedge p \equiv p$.

Theorem.(De Morgan's Law)

For every two propositions p, q

(a) $\neg(p \wedge q) \equiv \neg p \vee \neg q$ and (b) $\neg(p \vee q) \equiv \neg p \wedge \neg q$.

Proof. Simplified truth-table method.

Instead of checking all possible cases we notice that the left-hand side of (a) is false **only** when $p \wedge q$ is true, i.e. when both p, q are true. On the other hand the disjunction $\neg p \vee \neg q$ is false **only** when both $\neg p$ and $\neg q$ are false, i.e. when both p, q are true. Hence the two sides are equivalent.

(b) can be deduced from (a): $\neg(p \vee q) \equiv \neg(\neg(\neg p) \vee \neg(\neg q)) \equiv \neg(\neg(\neg p \wedge \neg q)) \equiv \neg p \wedge \neg q$. QED

Theorem.(Laws involving the "if … then" connective)

For every two propositions p, q

(a) $p \Rightarrow q \equiv \neg p \lor q$ (*conditional disjunction)*

(b) $p \Rightarrow q \equiv \neg q \Rightarrow \neg p$ (*contrapositive law*)

Proof.

(a)Simplified truth-table method. The LHS is **only** false when p is true and q is false, the RHS is **only** false when $\neg p$ is false and q is false, in other words when p is true and q is false.

(b) By (a), $\neg q \Rightarrow \neg p \equiv \neg(\neg q) \lor \neg p \equiv q \lor \neg p \equiv \neg p \lor q \equiv$

$p \Rightarrow q$. QED

Of course this can be done by truth-table as well, but the transformation method is much more fun.

Remark.

The contraposition law is the background for so called indirect proofs, or proofs by contradiction. In these proofs, instead of showing that the assertion of out theorem follows from our assumptions we show that if the assertion is not true then our assumption could not have been satisfied. People tend to make silly logical mistakes here so be careful.

Tautologies

Definition.

A formula $\varphi(p_1, p_2, \dots, p_k)$ is called a *tautology* iff
$\varphi(p_1, p_2, \dots, p_k) \equiv 1$.
In other words a *tautology* is a formula, such that for every assignment of 0-s and 1-s to its variables the logical value of the resulting statement is 1 (*true*).

Fact.

Now we can explain the relation between $\equiv$ and $\Leftrightarrow$:
$\varphi(p_1, p_2, \dots, p_k) \equiv \psi(p_1, p_2, \dots, p_k)$ if and only if
$\varphi(p_1, p_2, \dots, p_k) \Leftrightarrow \psi(p_1, p_2, \dots, p_k)$ is a tautology.

Every algebraic formula can be reduced to a sum of products. This is possible thanks to distributivity of multiplication with respect to addition. For example (a+b)(a+c)+e = $a^2 + ac + ab + bc + e$. Similarly, every logical formula can be reduced (shown to be logically equivalent) to a conjunction of *clauses*, i.e. expression involving only individual variables, their negations and the OR connectives. Such an expression is called a *conjunctive normal form* of the formula.

This trick is often used to verify logical equivalence of formulas – you reduce both formulas to their CNF-s and compare the CNF-s term to term.

Examples.

1. p, $\neg p$, $p \wedge \neg$q, $p \vee \neg$q are CNF formulas
2. $p \Rightarrow q$ is not, but $p \Rightarrow q \equiv \neg p \vee q$ and $\neg p \vee q$ is a (single clause) CNF.
3. $(p \Rightarrow q) \Rightarrow r \equiv \neg(p \Rightarrow q) \vee r \equiv \neg(\neg p \vee q) \vee r$ which is not good enough because the first $\neg$ sits in front of an expression, not an individual variable. Using de Morgan's Law we get $(\neg\neg p \wedge \neg q) \vee r \equiv (p \wedge \neg q) \vee r \equiv (p \vee r) \wedge (\neg q \vee r)$ – a CNF
4. Is the formula from 3 equivalent to $p \Rightarrow (q \Rightarrow r)$? $p \Rightarrow (q \Rightarrow r) \equiv \neg p \vee (q \Rightarrow r) \equiv \neg p \vee (\neg q \vee r) \equiv \neg p \vee \neg q \vee r$ which means it is not. To be on the safe side: the last CNF is only false for p,q,r = 1,1,0 while the first is also false for example for 0,1,0