Notes for ECE 369 - Discrete Mathematics for Computer Engineering

Ezekiel Ulrich

August 24, 2023

These are lecture notes for fall 2023 ECE 36900 at Purdue. Modify, use, and distribute as you please.

Contents

Course Introduction

This course introduces discrete mathematical structures and finite-state machines. Students will learn how to use logical and mathematical formalisms to formulate and solve problems in computer engineering. Topics include formal logic, proof techniques, recurrence relations, sets, combinatorics, relations, functions, algebraic structures, and finite-state machines. For more information, see the syllabus.

Equations

1. De Morgan's Theorem:

   $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$

   $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$

2. Modus ponens (mp)

   p

   $p \to q$

   $\therefore q$

3. Modus tonens (mt)

   $p \to q$

   $\neg q$

   $\therefore \neg p$

Propositional Logic

We often wish that others would be more logical, tell the truth, or shower. While studying formal logic cannot help with the latter (in fact, studies have shown a negative correlation between hygiene and studying formal logic) it is a useful way to define what the first mean two. In a formal logic model, we have two constructs:

- **Statements/propostion**: A statement or a proposition is a sentence that is either true or false. Propositions are often represented with letters of the alphabet. For example: "q: the more time you spend coding, the less time you have to buy deodorant."
- **Logical connectives**: Used to connect statements. For example, "and" is a logical connective in English. It can be used to connect two statements, e.g. "the person next to me smells like dog *and* looks like a dog" to obtain a new statement with its own truth value.

  Here are common logical connectives in Boolean logic:

Table 1: Logical Connectives in Boolean Logic

Logical Connective	**Symbol**
Negation (NOT)	$\neg$
Conjunction (AND)	$\wedge$
Disjunction (OR)	$\vee$
Exclusive OR (XOR)	$\oplus$
Implication	$\rightarrow$
Biconditional	$\leftrightarrow$

Truth table: Defines how each of the connectives operate on truth values. Every connective has one. For example, consider $\wedge$ AND:

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Rules and proofs

With each additional variable in your truth table, the number of choices grows exponentially. Specifically, if you have n statement letters, you would have 2^n choices for your truth table.
Tautology: A formula that is true in every model. Example: the Bible is infallible because the Bible itself says it is infallible.

Contradiction: A formula that is false in every model Examples: "it is raining and it is not raining", "I am sleeping and I am awake", "IU is a good school".

Confusion often arises when negating a sentence such as "the book is thick and boring". An natural inclination is to negate it thus: "the book is not thick and not boring". However, consider the truth table for this: p: "the book is thick", q: "the book is boring". We can see the

p	q	$p \wedge q$	$\neg(p \wedge q)$	$\neg p \wedge \neg q$
T	T	T	F	F
T	F	F	T	F
F	T	F	T	F
F	F	F	T	T

last two rows are not identical, therefore the negation of "the book is not thick and not boring" is not "the book is not thick and not boring". For p to be false, either the book must not be thick *or* the book must not be boring. This is summarized by **De Morgan's Theorem**:

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$

Interestingly, it is possible to prove any statement in a system where a contradiction exists. This is known as the *principle of explosion*. To see how it works, consider the following example:

1. p: Donuts are good for you.
2. q: Unicorns exist.

I'll now assume the contradictory statement "donuts are good for you and donuts are not good for you".

$$\begin{array}{rl} \neg p \wedge p & \text{(Contradiction)} \\ p \vee q & \text{(Addition)} \\ \neg p & \\ \therefore q & \end{array}$$

Ergo, unicorns exist :)

We now have a sufficient understanding of truth tables and logical connectives to come up with some useful rules. First of these are **Modus ponens (mp)**:

$$\begin{array}{l} p \\ p \to q \\ \therefore q \end{array}$$

and
Modus tollens (mt):

$$\begin{array}{l} p \to q \\ \neg q \\ \therefore \neg p \end{array}$$

Below are two tables for commonly used rules.

Expression	Equivalent to	Name - abbreviation
$p \vee q$ $p \wedge q$	$q \vee p$ $q \wedge p$	Commutative - comm
$(p \vee q) \vee r$ $(p \wedge q) \wedge r$	$p \vee (q \vee r)$ $p \wedge (q \wedge r)$	Associative - ass
$\neg(p \wedge q)$ $\neg(p \vee q)$	$\neg p \vee \neg q$ $\neg p \wedge \neg q$	De Morgan's Laws - De Morgan
$p \to q$	$\neg p \vee q$	Implication - imp
p	$\neg(\neg p)$	Double negation - dn
$p \leftrightarrow q$	$(p \to q) \wedge (q \to) p$	Def'n of equivalence - equ

Table 2: Equivalence rules

From	Can derive	Name - abbreviation
$p, p \to q$ $p \to q, \neg q$	q $\neg p$	Modus ponens - mp Modus tollens - mt
p, q	$p \vee q$	Conjunction - con
$p \wedge q$	p, q	Simplification - sim
p	$p \vee q$	Addition - add

Table 3: Inference rules