Notes for ECE 369 - Discrete Mathematics for Computer Engineering

Ezekiel Ulrich

September 19, 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
   $p \rightarrow q$
   $\therefore q$
3. Modus tonens (mt)
   $p \rightarrow q$
   $\neg q$

$\therefore \neg p$

4. Predicate inference rules:

Name	Abrv.	Given	Can conclude
Existential generalization	eg	$P(a)$	$(\exists x)P(x)$
Existential instantiation	ei	$(\exists x)P(x)$	$P(a)$
Universal generalization	ug	$P(x)$	$(\forall x)P(x)$
Universal instantiation	ui	$(\forall x)P(x)$	$P(a)$

5. Propositional equivalence 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
$p \rightarrow q$	$\neg p \vee q$	Implication - imp
p	$\neg(\neg p)$	Double negation - dn
$p \leftrightarrow q$	$(p \rightarrow q) \wedge (q \rightarrow p)$	Def'n of equivalence - equ

6. Propositional inference rules:

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

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:

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

Table 1: Logical Connectives in Boolean Logic

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

Table 2: Truth table for $\wedge$

We see that p AND q is only true when p is true and q is true. Similarly, p or q is only true when p is true or q is true (or both). An important connective for discovering new truths is the implication $\rightarrow$, which basically says "if the first letter is true, then so is the second". Let p: "I live in Wiley" and q: "I have no AC". In English, the statement $p \rightarrow q$ would be stated as "If I live in Wiley, then I have no AC".

Table 3 shows the truth table for $\rightarrow$. It may not seem immediately clear why, for instance, if p and q are false, then $p \rightarrow q$ is true. If we

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Table 3: Truth table for $\to$

consider what this means in English, then all we know is that I don't live in Wiley. Perhaps I live in Tarkington and still don't have AC, or perhaps I live in Honors and I do. In any case the first letter isn't true, so "if the first letter is true then so is the second" stands as true. If we have the statement $p \to q$, then we call q a *necessary condition* for p. Conversely, p is a *sufficient condition* for q.

Say we have a statement such as $\neg A \vee B \to C$. This is ambiguous, since we can interpret it as either $(A \vee B) \to C$ or $A \vee (B \to C)$. The truth tables will differ in each case, so it becomes necessary to specify in what order we should apply logical connectives.

1. Parentheses "()"
2. Negation "$\neg$"
3. AND and OR "$\wedge$, $\vee$"
4. Implication "$\to$"
5. Biconditional "$\leftrightarrow$"

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 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**:

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

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".

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

Ergo, unicorns exist :)

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

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

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):

$$p$$
$$p \to q$$
$$\therefore q$$

and
Modus tollens (mt):

$$p \to q$$
$$\neg q$$
$$\therefore \neg p$$

Below are two tables for commonly used rules.

Table 4: Equivalence 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

At this point, let us formally define an
Argument: An argument can be symbolized as

$$P_1 \vee P_2 \vee P_3 \vee ... \vee P_n \to Q$$

where P_i is called a hypothesis and Q is the conclusion. If this statement is a tautology, then the argument is *valid*. There are multiple

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

Table 5: Inference rules

ways we could prove a given argument is a tautology. For instance, we could create a truth table and brute force an answer. However, with even four hypotheses this process is tedious, and with each additional hypothesis it becomes exponentially harder. Therefore we instead often turn to the

Proof sequence: a sequence of well-formed formulas in which each formula is either a premise or the result of applying a derivation rule to earlier well-formed formulas. In practice this looks like

$$\begin{aligned} P_1 &\quad \text{(premise)} \\ P_2 &\quad \text{(premise)} \\ P_3 &\quad \text{(premise)} \\ &\quad \dots \\ P_n &\quad \text{(premise)} \\ \text{(formula) } 1 &\quad \text{(obtained from derivation rule)} \\ \text{(formula) } 2 &\quad \text{(obtained from derivation rule)} \\ &\quad \dots \\ \text{(formula) } n &\quad \text{(obtained from derivation rule)} \\ \therefore Q & \end{aligned}$$

Let's use all this new information in a simple proof.

$$\begin{array}{rl} A & \text{(hypothesis)} \\ A \to B & \text{(hypothesis)} \\ B \to C & \text{(hypothesis)} \\ B & \text{(1, 2, mp)} \\ C & \text{(4, 3, mp)} \\ \hline \therefore C & \end{array}$$

If we wish to apply our knowledge of logic to the real world, some practice in translating natural language to formal logic is necessary. Let's test it with this statement: "If chicken is on the menu, then don't order fish, but you should have either fish or salad. So if chicken is on the menu, have salad." Let C: "Chicken is on the menu", F: "You order fish", and S: "You have salad". We know that if chicken is on the menu you don't order fish, that you should have either fish or salad, and we'd like to show that if chicken is on the menu you should have salad.

$$
\begin{array}{rl}
C & \text{(hypothesis)} \\
C \to \neg F & \text{(hypothesis)} \\
F \vee S & \text{(hypothesis)} \\
\neg F & \text{(1, 2, mp)} \\
S & \text{(3, 4, dis)} \\
\hline
\multicolumn{2}{c}{\therefore S}
\end{array}
$$

Predicate logic

Predicate logic: Capable of making statements about entire groups instead of individual letters. In predicate logic, propositions are expressed in terms of predicates, variables and quantifiers, the latter of which propositional logic lacks.
Quantifer: How many objects have a certain property: "for every" or "for some".
Predicate: Property that a variable may have.
Domain of interpretation: Collection of objects from which the variable is taken.
Universal quantifer: "For all": $\forall$. States that a certain property holds for all objects in a domain.
Existential quantifer: "There exists": $\exists$. States that a certain property holds for at least one object in a domain.

As an example of a predicate well-formed formula: $(\forall x)[(\exists y)x > y]$. We would read this statment as "for all x there exists a y such that $x > y$." At first glance it may seem obvious that this statement is true, but consider the domain. What if the domain is all natural numbers? Then we could let $x = 1$ (or zero depending or your definition of natural numbers) and there would be no corresponding lesser y. We can see from this example that the truth value of a predicate logic formula depends on the domain as well as quantifiers and predicates.

Just as with propositional logic, we often need to translate English statements into predicate logic. Take the statement "every movie made by George Lucas is great". We can rephrase this as "for any movie, if the movie is made by George Lucas, it is great". We would write this formula as

$$(\forall x)(GL(x) \to Great(x))$$

(Author's note: no value judgement is associated with this English statement).

Let's examine some rules in predicate logic. First, negation:

$$\neg[\forall x A(x)] \leftrightarrow (\exists)x\neg A(x)$$

Some rules from propositional logic still apply in predicate logic. Take modus ponens as an example:

$$\begin{array}{rl}
(\forall x)(\forall y)L(x,y) \rightarrow [(\exists x)H(x)] & \text{(hypothesis)} \\
\neg[(\exists x)H(x)] & \text{(hypothesis)} \\
\neg[(\forall x)(\forall y)L(x,y)] & \text{(1, 2, mt)} \\
(\exists x)(\exists y)\neg L(x,y) & \text{(3, DM)} \\
\hline
\therefore (\exists x)(\exists y)\neg L(x,y) &
\end{array}$$

Table 6 holds predicate inference rules. These rules hold given certain conditions. Namely:

Name	Abrv.	Given	Can conclude
Existential generalization	eg	$P(a)$	$(\exists x)P(x)$
Existential instantiation	ei	$(\exists x)P(x)$	$P(a)$
Universal generalization	ug	$P(x)$	$(\forall x)P(x)$
Universal instantiation	ui	$(\forall x)P(x)$	$P(a)$

Table 6: Predicate inference rules

(eg) x not in $P(a)$

(ei) Must the be first rule that introduces a

(ug) $P(x)$ not derived from a hypothesis with x as a free variable, and $P(x)$ is not derived by ei from wff with x as a free variable.

(ui) a is a constant.

Let's see some of these rules in action by with a predicate logic proof. Say we have the statement "every ECE student works harder than somebody, and everyone who works harder than any other person gets less sleep than that person. Maria is an ECE student. Ergo, Maria gets less sleep than someone. Let $E(x)$: "x is an ECE student", $W(x,y)$: "x works harder than y", $S(x,y)$: "x gets less sleep than y", and m:

A *free variable* is a variable not bound by a quantifer. For example, in the formula

$$(\forall x)(\forall y)P(x,y)$$

both x and y are bound by quantifers. Constrast this with the formula

$$(\exists x)(\forall y)q(x,y,z)$$

In this example, z is a free variable, since it is not associated with any quantifiers.

Maria. We want to prove $\exists a(S(m,a))$.

$$
\begin{array}{rl}
\forall x, E(x) \to (\exists y)(W(x,y)) & \text{(hypothesis)} \\
\forall x, \forall y(W(x,y) \to S(x,y)) & \text{(hypothesis)} \\
E(m) & \text{(hypothesis)} \\
\exists y(E(m) \to S(m,y)) & \text{(1, ui)} \\
E(m) \to W(m,a) & \text{(4, ei)} \\
\forall y(W(m,y) \to S(m,y)) & \text{(3, ui)} \\
W(m,a) \to S(m,a) & \text{(6, ui)} \\
W(m,a) & \text{(3,5 mp)} \\
S(m,a) & \text{(7,8, mp)} \\
\exists a(S(m,a)) & \text{(9, eg)} \\
\hline
\multicolumn{2}{c}{\therefore \exists a(S(m,a))}
\end{array}
$$

Proofs

List of common proof techniques:

1. Exhaustive proof: In this kind of proof, the statement to be proved is split into a finite number of cases or sets of equivalent cases, and where each type of case is checked to see if the proposition in question holds

2. Refuting by counter-example: If we have a universal statement such as $\forall x(P(x) \to Q(x))$, we may show it to be false by finding a single a such that $\neg P(a)$.

3. Direct proof: Trying to prove $p \to q$, start by assuming p and then show q.

4. Proof by contraposition: The contrapositive of $p \to q$ is $\neg q \to \neg p$. A statement and its contrapositive are logically equivalent, so if proving $p \to q$ is too difficult we may try to prove $\neg q \to \neg p$ instead.

5. Proof by contradiction: Suppose I have to prove $p \to q$. I can begin by saying p is true and $\neg q$ is true. If by a series of steps I arrive a contradiction, then I may say p implies q.

6. Proof by induction: employs a neat trick which allows you to prove a statement about an arbitrary number n by first proving it is true when $n = 1$ (or some other base case), assuming it is true for $n = k$, and then showing it is true for $n = k + 1$. The steps to prove $\forall n P(n)$ are:

   (a) Prove $P(1)$ (this is your *base case*).

(b) Assume for arbitrary $k \geq 1$, $P(k)$ (your *inductive hypothesis*).

(c) Prove $P(k+1)$.

Below are some example proofs using each of these techniques.

1. *Exhaustive proof (cases)*: say we wish to prove $|xy| = |x||y|$. Let's split this into four cases:

   (a) Case 1: x and y positive. Then the absolute values are equal to the original numbers, and we have

   $$\begin{aligned} |x| &= x \\ |y| &= y \\ |xy| &= xy \\ |x||y| &= xy \\ \therefore |xy| &= |x||y| \end{aligned}$$

   (b) Case 2: x and y negative. If x and y are both negative, then xy is positive. We thus have that

   $$\begin{aligned} |x||y| &= xy \\ |xy| &= xy \\ \therefore |xy| &= |x||y| \end{aligned}$$

   (c) Case 3: x negative and y positive. Now we have that xy is negative. Still, though, $|xy|$ will be positive (by def'n of | |), and so will $|x|$ and $|y|$. So we again have that

   $$|xy| = |x||y|$$

   (d) Case 4: x positive and y negative. WLOG, case 3.

   $$\therefore |xy| = |x||y| \quad \square$$

2. *Direct proof*: say we wish to prove the product of two even integers is even. We first need to translate this English sentence to a mathematical statement, which we can do in this case like so:

   $$x = 2a, a \in \mathbb{Z}, y = 2b, b \in \mathbb{Z}, \rightarrow x \times y = 2c, c \in \mathbb{Z}$$

   Our proof is below.

   $$\begin{aligned} x &= 2a, a \in \mathbb{Z} \\ y &= 2b, b \in \mathbb{Z} \\ z &= x \times y \\ &= 2a \times 2b \\ &= 2(2ab) \\ &= 2c, c \in \mathbb{Z} \quad \square \end{aligned}$$

Since c is an integer, $2c$ is even and the proof is complete. Try proving the product of two odds is odd in a similar fashion.

3. *Proof by contradiction*: say we wish to prove $\sqrt{2}$ is irrational. In a theme that will become common as we see more proofs by contradiction, assume the opposite. That is, assume $\sqrt{2}$ is rational. By the definition of rational, we can then write

$$\sqrt{2} = \frac{a}{b}, a, b \in \mathbb{Z}$$

Where a and b share no common factors. We can then perform the following series of steps.

$$\begin{aligned} \sqrt{2} &= \frac{a}{b} \\ b\sqrt{2} &= a \\ 2b^2 &= a^2 \end{aligned}$$

This means that a^2 is even. It can be easily shown that if a^2 is even then a is even. That means that a^2 will actually be divisible by 4. We can rearrange to get

$$\begin{aligned} 2b^2 &= 4c, c \in \mathbb{Z} \\ b^2 &= 2c \end{aligned}$$

So b is likewise even. But if both a and b are even, then they share a common factor and our original supposition is false. Ergo $\sqrt{2}$ is irrational. $\square$.

4. *Poof by induction*: say we wish to show

$$\sum_{i=1}^{n} = \frac{n(n+1)}{2}$$

Begin with the base case $n = 1$.

$$\begin{aligned} \sum_{i=1}^{1} &= 1 \\ &= \frac{1(1+1)}{2} \\ &= \frac{n(n+1)}{2} \end{aligned}$$

Since we have shown the base case to be true, we may now make our inductive hypothesis and assume that for arbitrary $k \geq 1$,

$$\sum_{i=1}^{k} = \frac{k(k+1)}{2}$$

Let us now show that the formula holds for $k+1$.

$$\begin{aligned}
\sum_{i=1}^{k+1} &= \sum_{i=1}^{k} +(k+1) \\
&= \frac{k(k+1)}{2} + (k+1) \\
&= \frac{k(k+1)+2(k+1)}{2} \\
&= \frac{k^2+3k+2}{2} \\
&= \frac{(k+1)(k+2)}{2} \\
&= \frac{(k+1)((k+1)+1)}{2}
\end{aligned}$$

And we are done $\square$.

The form of induction we have just seen is the *weak* form of induction. The *strong* form of induction is the following. To prove $\forall x, P(x)$, we still prove the base case ($P(n)$). Now, however, we assume for arbitrary $k \geq 1$ that $P(r)$ is true for $1 \leq r \leq k$ and try to prove $P(k+1)$. Let's see this in action. Say we'd like to prove any postage greater than or equal to 8 cents can be created with a combination of 3 cent and 5 cent postage stamps. First, the base case. 8 can be created like so: $3+5=8$. Now let's assume for all $8 \leq r \leq k$, $P(r)$. Now the tricky part. To prove $P(k+1)$, notice that

The strong form is also known as the *second principle of mathematical induction*

$$\begin{aligned}
P(k+1) &= k+1 \\
&= (k-2)+3 \\
&= (3a+5b)+3 \\
&= 3c+5b \quad \square
\end{aligned}$$

We had to rewrite $k+1$ as $k-2+3$ so we could use our assumption that $P(k-2)$ is true. The astute among you will recognize that our proof is technically imcomplete. We have assumed $P(k-2)$, but what if we wish to prove $P(9)$? This is not included in our inductive step, since we only assume $P(r)$ for $8 \leq r \leq k$. We are in the domain of natural numbers (unless you have somehow managed to find a postage stamp with negative or fractional value), then there is no r for which this is true. Therefore we also need to prove $P(9)$ and $P(10)$, which is pretty simple. Let's see a more advanced example of induction.

We see that simply assuming $P(k)$ wouldn't be sufficient in this case, so there are proofs where we can use strong induction but not weak induction. Anything that can be proved with weak induction can be proved with strong induction, since in strong induction we assume $P(k)$ in addition to $P(r)$ for all r between 1 and k. Hence, "strong"

Say you have a set of n elements, and you want to create subsets of this set. You are interested in knowing how many subsets exist. After trying it out with $n = 1,2,3$ you suspect the number of subsets that can exist is 2^n, and you see that this problem is a good candidate for induction. Your base case is $P(1) = 2 = 2^1$, so that's out of the way. Now assume $P(k)$. That is, for any set with $k < n$ elements, the number of subsets is 2^k. You now need to show $P(k+1)$, which can be done by noticing that if we add an additional element to the set, then for other group you can add the new element to get k new groups, bringing your total number of subsets to $k + k = 2k = 2^{k+1}$, and we are done.