1 Chapter 1

1.1 Propositional Logic

Proposition - a statement with an answer of either *true* or *false*. Every proposition should have a negation
Negation - let p be a proposition. Then the negation of p is the statement "It is not the case that p", or, $\neg p$. A negation of a proposition forms an entirely new proposition.

Similarly, we join two propositions via *conjunctions* and *disjunctions*:
Conjunction - Let p, q be propositions. Their conjunction is p **and** q or $p \wedge q$ e.g. The steak was cooked and the grill was hot.
Disjunction - p and q's disjunction is p or q, or $p \vee q$, e.g. it is cloudy or it is sunny. THere also exists an exclusive or

Q: Why is it that I can only think of two successive propositions (Because the grill was hot, the steak was cooked?) What is that called? What are the implications?

Conditional Statement - the proposition if p, then q, or $p \rightarrow q$. Also known as an *implication*. It is only false when p is true and q is false (in other words, when p does not imply q or $F \rightarrow F$). Think iof it like the binding terms of a contract; you only break it if you fulfill the rhs, and don't fulfill the lhs p is the antecedent while q is the consequence. Only care about rhs when lhs is truex
Tricky terminology:

- "q unless $\neg p$"
- "p only if q" - p is only true when q is true. The very definition of a conditional statement

1.1.1 Truth Tables

Holds all possible combinations of the truth values of those combinations. Useful for proving equivalences of simple (low number of variables) propositions.
2^n rows where n is the number of variables in the compound proposition.

1.1.2 Properties of implications

Consider the following statement:
If it is raining outside, then it is cloudy

Inverse - $\neg p \rightarrow \neg q$. Equivalent to *converse*. e.g. If it is not raining outside, then it is not cloudy.
Converse - $q \rightarrow p$. Equivalent to *inverse*. e.g. If it is cloudy, then it is raining outside. Simpilest one, so the shortest.
Contrapositive - $\neg q \rightarrow \neg p$. The only one to always has the same truth value as the original implication, $p \rightarrow q$. e.g. If it is not cloudy outside, then it is not raining

1.1.3 Biconditionals

$(p \iff q) \vee [(p \rightarrow q) \wedge (q \rightarrow p)]$
Biconditional - iff and only if; q must be a "subset" of p. T here is only one path forward, e.g. "you can take a flight iff you buy a ticket. "two way street" of a conditional statement. "It is necessary and sufficient" **Q:** are

$p \iff q$ and $q \iff p$ logically equivalent? **A:** Yes! $\iff$ is commutative

1.1.4 Precedence

1. $\neg$
2. $\wedge$ & $\vee$
3. $\rightarrow$ & $\iff$

1.2 Propositional Equivalences

Replacement of one compound propositional statement with another is the foundation of mathematical proof
Tautology - A statement which is always true, no matter the "input" conditionals (p/q), e.g. $p \lor \neg p$
Contradiction - A statement which is always false, no matter the "input" conditionals. e.g. $p \land \neg p$.
Contingency - A statement which varies based on the "input" conditionals.

1.2.1 Logical Equivalence

The ability to replace one compound proposition with another which evaluates to the same truth table, is known as **logical equivalence** or $p \equiv q$. This new symbol, $\equiv$, is not connector but rather a shortening for the statement that $p \iff q$ is a tautology. Two compound propositions p and q are equivalent when $p \iff q$ is a tautology.

1.2.2 Important Equivalences

1. $p \implies q \equiv \neg p \lor r$ Definition of Implication
2. $p \implies r \equiv \neg q \implies p$
3. $p \iff r \equiv (p \implies r) \land (r \implies p)$ Definition of the Biconditional
4. $p \land T \equiv p$, $p \lor F \equiv p$ Identity Law
5. $p \land F \equiv F$, $p \lor T \equiv T$ Domination Law
6. $p \lor p \equiv p$, $p \land p \equiv p$ Idempotent Law
7. $\neg(\neg p) \equiv p$ Double Negation
8. $p \lor q \equiv q \lor p$ Commutative Law, also works for conjunctions
9. $(p \lor q) \lor r \equiv p \lor (q \lor r)$ Associative Law, works with conjunctions
10. $p \lor (q \land r) \equiv (p \lor q) \land (p \lor q)$ Distributiive law, opposite for conjunctions
11. $p \lor (p \land q) \equiv p, p \lor (p \land q) \equiv p$ Absorption Laws

De Morgans Law

De Morgan's laws provide the following two equivalences:

$$\neg(p \lor q) \equiv \neg p \land \neg q \tag{1}$$
$$\neg(p \land q) \equiv \neg p \lor \neg q \tag{2}$$

Essentially, a "distribution" of the negation (not to be confused with the *distributive laws*)

Q: Is there a simple derivation of De Morgan's laws using logical equivalences? When does it become appropriate to use logical equivalences as opposed to truth tables?

1.2.3 Proof Using Logical Equivalence

The simplest type of proof involves using a series of logical equivalences to conclude a different statement given and initial conditional. It is important to list all steps along with the associated law.

1.2.4 Satisfiability

A proposition is *satisfiable* if, for some set of values of its conditions $\{p_1, p_2 \dots p_{n-1}, p_n\}$, the proposition can evaluate to True. If a proposition is not satisfiable, then it is considered *unsatisfiable*

1.3 Predicates and Quantifiers

Subject - object which holds some property, or a *predicate*
Predicate - some quality describes a *subject*, e.g. "is greater than 3". Each predicate can have a predicate function, $P(x)$, which allows you to evaluate the truth of a predicate
• **Precondition** - conditions which describe valid input
• **Postcondition** - all conditions the output of a program must satisfy

1.3.1 Quantifiers

Quantification - creating a proposition from a propositional function. 2 types: *universal quantification* and *existential quantification.* I really like thinking of it in terms of set theory/probability space; the universal quantifier characterizes an entire class while the existential quantifier describes an object as a part of another class.

1.3.2 The Universal Quantifier

A domain must be specified when working with universal quantifiers. Most mathematical statements are universal quantifiers (hey, that's an existential quantifier right there!). Variables used immediately after both the universal and existential quantifier are considered *it.* Any other variables are *free*

"P(x) for all values of x in the domain"

or...

$$P(x) \; \forall \; x$$

Any statement for which P(x) is false is called a *counterexample.* Just like any other conditional, this can evaluate to true or false:

$$x + 1 > x \forall x \in \mathbb{R}$$

evaluates to true while:

$$2x < x \; \forall \; x \in \mathbb{R}$$

evaluates to false. Even a single counter example leads to a universal quantifier evaluating to false. It is implicitly assumed that the domain is non-empty.

$$\forall x P(x) \equiv P(x_1) \wedge P(x_2) \wedge ... \wedge P(x_n)$$

1.3.3 Existential Quantifier

The existential quantifier can be written as:

"There exists an element x in the domain s.t. P(x)"

otherwise known as..

$$\exists \; x \; P(x)$$

Q: are we *always* implicitly working with positive integers?

Similarily, $\exists$ evaluates to true when even a single example exists where the predicate function evaluates to true. In this way can it be thought of as the negation of $\forall$?

$$\exists x P(x) \equiv P(x_1) \vee P(x_2) \vee ... \vee P(x_n)$$

both $\exists$ AND $\forall$ have the highest order of precedence.

1.3.4 Quantifiers with Restricted Domains

Short hand for:

$$\exists(x < 0 \implies x - 1 < 0)$$

is:

$$\exists x < 0 \ (x - 1 < 0)$$

Additionally:

$$\exists x < 0(P(X)) \equiv \exists(x < 0 \land P(x))$$

1.4 Negating Quantified Expressions

$$\neg\forall x P(x) \equiv \exists x \neg P(x) \neg\exists x Q(x) \equiv \forall x \neg Q(x)$$

This is mainly derived from De Morgan's law(s) and the expansion of the universal/existential quantifiers.

2 Nested Quantifiers

consecutive quantifiers are simply *and*-ing both (or many) of the quantifiers. Pretty simple to read. The book compares them to nested loops in a computer program. If an inner loop is false (or true! in the case of the existential quantifier), then the outer loop can also be concluded to be false.

While quantifiers of the same type can be exchanged, quantifiers of the opposite type do not play as nice.

2.1 Negating Nested Quantifiers

Many times, we move the negation inside the quantifers and flip the quantifier so that the negation is on the innermost statement.