

Discrete Mathematics - Final

Daniel Rosel

December 10, 2022

Contents

1	Domains	3
2	Terminology	4
3	Propositional Logic	4
3.1	Operators	4
3.2	DeMorgan's Laws	6
3.3	Fallacies	6
3.4	Converse, Contrapositive, Inverse	6
3.5	Disjunctive Normal Form	7
3.6	Logical Equivalences	7
3.7	Satisfiability	8
3.8	Practice Problems	8
4	Predicate Logic	13
4.1	Predicates	13
4.2	Quantifiers	13
4.3	Universal Quantifier	13
4.4	Existential Quantifier	14
4.5	Singular Existential Quantifier	14
4.6	Negation	14
4.7	Nesting	14
4.8	Practice Problems	15
5	Inference	15
5.1	Rules of Inference	16
5.2	Rules of Inference for Quantified Propositions	17
5.3	Practice Problems	17

6 Formal Proofs	18
7 Informal Proofs	19
7.1 Methods of Proof	19
8 Proofs Practice Problems	22
8.1 Problems from the Book	23
9 Set Theory	28
9.1 Sets	28
9.2 Subsets	28
9.3 Set Cardinality	29
9.4 Power Set	29
9.5 Cartesian Product	29
9.6 Set Operations	30
9.7 Union	31
9.8 Intersection	31
9.9 Difference	31
9.10 Complement	32
9.11 Logical Operations with Sets	33
10 Functions	33
10.1 Domain, Codomain & Range	34
10.2 One-to-One	34
10.3 Onto	34
10.4 Composite Functions	35
10.5 Inverse Functions	35
10.6 Partial Functions	36
10.7 Practice Questions	36
11 Sequences	37
11.1 Arithmetic Sequences	37
11.2 Geometric Sequences	37
11.3 Recurrence Relations	38
12 Summations	38
13 Divisibility	39
13.1 Divisibility Rules	39

14 Modular Arithmetic	40
14.1 Rules of Modular Arithmetic	40
14.2 Inverse of a Modulo	41
14.3 Practice Problems	41
15 Integer Representation	42
16 Binary Representation	42
17 Hexadecimal Representation	43
18 Prime Numbers	43
19 Algorithms	44
19.1 Complexity of Algorithms	44
19.2 Big O Notation	44
20 Greatest Common Divisor	45
20.1 Least Common Multiple	46
21 Applications of Number Theory	46
21.1 Databases	46
21.2 Hash Functions	46
21.3 Asymmetric Encryption	46
22 Relations	47
22.1 Reflexive Relations	48
22.2 Symmetric Relations	48
22.3 Antisymmetric Relations	48
22.4 Transitive Relations	48
22.5 Composite Relations	49
23 Directed Graphs	49
24 Grammars	50
24.1 Railroad Diagrams	50

1 Domains

Naturals \mathbb{N} $[1, 2, 3, \dots, \infty)$

Integers \mathbb{Z} $(-\infty, \dots, -2, 0, 2, \dots, \infty)$

Rationals \mathbb{Q} Fractions of integers $\{\frac{p}{q} | p \in \mathbb{Z} \wedge q \in \mathbb{Z} \wedge q \neq 0\}$

Reals \mathbb{R} All real numbers

Complex \mathbb{C} All complex numbers

2 Terminology

Theorem A statement we can prove using some axioms

Proof A valid argument/set of arguments that demonstrate a theorem/proposition

Axioms Statements which are assumed to be true, they are in the most basic form

Lemma A small proof that is an element of another larger proof

3 Propositional Logic

A proposition, is a sentence, which can be given some **truth value**. There are two types of propositions:

Tautology A proposition that will always be **true** (\top)

Contradiction A proposition that will never be true (\perp)

3.1 Operators

There are three main operators which can be used to express any logical expression. These consists of **Not**, **Conjunction**, **Disjunction**.

3.1.1 Not

It flips the truth value of a proposition.

$$\neg p, \bar{A}$$

3.1.2 Conjunction

This operators returns a new truth value which is true **if both of the propositions are true**

$$p \wedge q, AB$$

3.1.3 Disjunction

Like the conjunction, it operates on two propositions. It will return true as long as either of the propositions is true.

$$p \vee q, A + B$$

3.1.4 Exclusive OR (XOR)

It is very similar to the **Disjunction** except that it must be exclusively one value that is true.

$$p \oplus q = (p \vee q) \wedge \neg(p \wedge q)$$

3.1.5 Conditional Statements / Implication

A conditional statement consists of two parts, the **premise** and **consequence**. The truth value is **only false when** $T \rightarrow F$.

$$p \rightarrow q \equiv \neg p \vee q$$

- If p is True, then q must also be True, we say that p is a **sufficient condition** for q . $T \rightarrow T$
- If p is False, q could still be True. $F \rightarrow T$
- p cannot be true if q is not true. $F \rightarrow F$
- p could still be false, even if q is true. $F \rightarrow T$

$$((p \rightarrow q) \wedge p) \rightarrow q$$

$$((p \rightarrow q) \wedge \neg q) \rightarrow \neg p$$

3.1.6 Bi-conditionals

It expressed a **sufficient** and **necessary** conditional relationship between two propositions. It is read as *if and only if p , then q* .

$$p \Leftrightarrow q$$

This operator will return True only if both p and q have equal values. The expression can be represented with the 3 basic operators as:

$$(p \rightarrow q) \wedge (q \rightarrow p) \equiv (\neg p \wedge \neg q) \vee (p \wedge q)$$

$$p \equiv q = \neg(p \oplus q)$$

3.1.7 Order of Operators

1. \neg
2. \wedge
3. \vee
4. \rightarrow
5. \Leftrightarrow

3.2 DeMorgan's Laws

$$\neg(p \wedge q) \equiv \neg p \vee \neg q \tag{1}$$

$$\neg(p \vee q) \equiv \neg p \wedge \neg q \tag{2}$$

3.3 Fallacies

The fallacies stem from the fact that an implication is true, and the truth value of either p, q .

Affirming the consequent $((p \rightarrow q) \wedge q) \rightarrow p$. We falsely assume that the premise is **true** because the consequence is **true**

Nageting the antecedent $((p \rightarrow q) \wedge \neg p) \rightarrow \neg q$. We falsely assume that the consequence is **false** because the premise is **false**.

3.4 Converse, Contrapositive, Inverse

Converse $q \rightarrow p$

Contrapositive $\neg q \rightarrow \neg p$

Inverse $\neg p \rightarrow \neg q$

The fallacies are a product of assuming that the **Converse and Inverse** are equivalent to the conditional statement.

3.5 Disjunctive Normal Form

This is a proposition created from the values in the truth table.

3.6 Logical Equivalences

These are various way you can arrive at some conclusion given a few basic propositions.

3.6.1 Identity Laws

$$p \wedge \top \equiv p$$

$$p \vee \perp \equiv p$$

3.6.2 Denomination Laws

$$p \vee \top \equiv \top$$

$$p \wedge \perp \equiv \perp$$

3.6.3 Idempotent Laws

$$p \vee p \equiv p$$

$$p \wedge p \equiv p$$

3.6.4 Double Negation Law

$$\neg(\neg p) \equiv p$$

3.6.5 Commutative Laws

$$p \vee q \equiv q \vee p$$

$$p \wedge q \equiv q \wedge p$$

3.6.6 Associative Laws

$$(p \vee q) \vee r \equiv p \vee (q \vee r)$$

$$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$$

3.6.7 Distributive Laws

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

3.6.8 3.2

3.6.9 Absorption Laws

$$p \vee (p \wedge q) \equiv p$$

$$p \wedge (p \vee q) \equiv p$$

3.6.10 Negation Laws

$$p \vee \neg p \equiv \top$$

$$p \wedge \neg p \equiv \perp$$

3.7 Satisfiability

A proposition is satisfiable if it has at least one case for which it is **true**, it is anything but a **contradiction**.

3.8 Practice Problems

For the following, create a truth table and a DNF for each (a DNF is a proposition created from the values in the truth table)

1. $p \vee \neg p$

Truth Table:

p	$\neg p$	$p \vee \neg p$
T	F	T
F	T	T

DNF:

$$p \vee \neg p = \top$$

2. $p \wedge \neg p$

Truth Table:

p	$\neg p$	$p \wedge \neg p$
T	F	F
F	T	F

DNF:

$$p \wedge \neg p = \perp$$

3. $p \vee (p \wedge q)$

Truth Table:

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

DNF:

$$p \vee (p \wedge q) = p$$

4. $p \wedge (p \vee q)$

Truth Table:

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

DNF:

$$p \wedge (p \vee q) = p$$

Now for implications:

1. $p \rightarrow q$

Truth Table:

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

DNF:

$$p \rightarrow q = \neg p \vee q$$

2. $p \rightarrow \neg q$

Truth Table:

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

DNF:

$$p \rightarrow \neg q = \neg p \vee \neg q$$

3. $(\neg p) \rightarrow q$

Truth Table:

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

DNF:

$$(\neg p) \rightarrow q = p \vee q$$

4. $\neg p \rightarrow \neg q$

Truth Table:

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

DNF:

$$\neg p \rightarrow \neg q = p \vee \neg q$$

Prove the following:

1. $p \rightarrow q \equiv \neg p \vee q$

Proof:

p $\neg p \vee q$	q	$p \rightarrow q$
T T	T	T
T F	F	F
F T	T	T
F T	F	T

Therefore $p \rightarrow q \equiv \neg p \vee q$ is true.

2. $p \rightarrow q \equiv \neg q \rightarrow \neg p$

Proof:

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

Therefore $p \rightarrow q \equiv \neg q \rightarrow \neg p$ is true.

Which of the following are satisfiable:

1. $p \wedge \neg p$

p	$\neg p$	$p \wedge \neg p$
T	F	F
F	T	F

Therefore $p \wedge \neg p$ is unsatisfiable.

2. $p \vee \neg p$

p	$\neg p$	$p \vee \neg p$
T	F	T
F	T	T

Therefore $p \vee \neg p$ is satisfiable.

3. $p \rightarrow q$

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

Therefore $p \rightarrow q$ is satisfiable.

4. $p \rightarrow \neg q$

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

Therefore $p \rightarrow \neg q$ is satisfiable.

5. $\neg p \rightarrow q$

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

Therefore $\neg p \rightarrow q$ is satisfiable.

4 Predicate Logic

With predicate logic, we can break down a complex proposition into predicates and quantifiers.

4.1 Predicates

It is a statement, which is made up of a proposition, with the only difference being that it can take a parameter.

X is faster than me

= Subject + Predicate

$\rightarrow P(x)$

We say that the it is instantiated once it gets evaluated with the passed parameter.

$$P(x_1, x_2, x_3, \dots, x_n)$$

And once instantiated with variables:

$$P(a_1, a_2, a_3, \dots, a_n) \equiv p$$

4.2 Quantifiers

With a quantifier, we can specify for which values on a certain domain, a predicate is true or false. There are various ways to express this: (**for all**, **for many**, **for at least one**, **for none**).

4.3 Universal Quantifier

If we have some predicate for which we can say that all the possible values in a certain domain will make it true, we use the universal quantifier.

$P(x)$ for all value of x within its domain

$$\forall x P(x)$$

- To prove that this quantifier is true, we must demonstrate that it holds true for all the possible values.
- To **dis**-prove the quantifier, it is sufficient to show that there exists just one that does not satisfy the predicate
- If we are considering an empty set, it will always be false.

4.4 Existential Quantifier

$P(x)$ for at least one value of x within its domain

$$\exists x P(x)$$

- To prove this quantifier, we must find at least one value for which the predicate holds true
- To disprove the quantifier, we must show that all the values will not hold for the predicate
- Again, if we consider an empty set, it will always be false

4.5 Singular Existential Quantifier

We can also use the existential quantifier, and with a bit of modification we can use it to specify how many values will satisfy the predicate.

There is exactly one x such that $P(x)$

$$\exists! x P(x)$$

The definition for this is as follows:

$$\exists x (P(x) \wedge \forall y (P(y) \Leftrightarrow x = y))$$

We can generalize this to a specific number n for which the predicate is satisfiable.

$$\exists_n x P(x)$$

4.6 Negation

When negating a quantifier, we have to consider both the outside and inside, that being **the quantifier itself** and the **predicate**.

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

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

4.7 Nesting

If we have more than one variable for which we quantify a predicate, we have to **consider the order**.

$$\exists x \forall y P(x, y) \not\equiv \forall y \exists x P(x, y)$$

4.8 Practice Problems

Evaluate the following:

1. $\forall x \exists y P(x, y)$ where $P(x, y) = x > y$

This quantifier is true, because for any value of x , there will always be a value of y that will satisfy the predicate.

2. $\exists x \forall y P(x, y)$ where $P(x, y) = x > y$

This quantifier is false, because there will always be a value of x for which there will not be a value of y that will satisfy the predicate.

3. $\forall x \exists y P(x, y)$ where $P(x, y) = x < y$

This quantifier is true, because for any value of x , there will always be a value of y that will satisfy the predicate. This because no matter the value of x , there will always be a value of y that will be greater than x .

Prove the following:

1. $\forall x \exists y P(x, y) \equiv \exists y \forall x P(x, y)$

Nested quantifiers:

1. $\exists x \forall y \exists z P(x, y, z)$ where $P(x, y, z) = x > y \wedge y > z$

Negating quantifiers:

1. $\neg \forall x \exists y P(x, y)$ where $P(x, y) = x > y$

5 Inference

A process of arriving at some conclusion through logical steps. This process consists of **arguments** which can either be valid or not. The logical form of an argument is vital, because it is the only way to determine if it is true or not.

$$\begin{array}{c} p \rightarrow q \\ p \\ \hline \therefore q \end{array}$$

It is another thing to call an argument **sound**, for this we need to have an argument for which **all premises are true and valid**. A **valid argument implies a tautology**. How can we tell if an argument is valid? We can use the **truth table**. Another way to determine if an argument is valid is to see if it makes use of any **existing argument forms**.

5.1 Rules of Inference

An argument form which would generate a valid implication falls under the rules of inference

5.1.1 Modus Ponens

$$\frac{p \quad p \rightarrow q}{q}$$

1. If p is true, then q is true
2. p is true
3. Therefore q is true

5.1.2 Modus Tollens

$$\frac{\neg q \quad p \rightarrow q}{\neg p}$$

5.1.3 Hypothetical Syllogism

$$\frac{p \rightarrow q \quad q \rightarrow r}{p \rightarrow r}$$

5.1.4 Disjunctive Syllogism

$$\frac{p \vee q \quad \neg p}{q}$$

5.1.5 Addition

$$\frac{p}{p \vee q}$$

5.1.6 Simplification

$$\frac{p \wedge q}{p}$$

5.1.7 Conjunction

$$\frac{\begin{array}{c} p \\ q \end{array}}{p \wedge q}$$

5.1.8 Resolution

$$\frac{\begin{array}{c} p \vee q \\ \neg p \vee r \end{array}}{\therefore q \vee r}$$

5.2 Rules of Inference for Quantified Propositions

5.2.1 Universal Instantiation

$$\frac{\forall x P(x)}{\therefore P(c)}$$

5.2.2 Universal Generalization

$$\frac{P(c) \text{ for an arbitrary } c}{\therefore \forall x P(x)}$$

5.2.3 Existential Instantiation

$$\frac{\exists x P(x)}{\therefore P(c)}$$

5.2.4 Existential Generalization

$$\frac{P(c) \text{ for an arbitrary } c}{\therefore \exists x P(x)}$$

5.3 Practice Problems

Use the rules of inference to prove the following:

1. $\neg(p \vee q) \rightarrow \neg p \wedge \neg q$

$$\frac{\begin{array}{c} \neg(p \vee q) \\ p \vee q \end{array}}{\therefore \neg p \wedge \neg q}$$

The rules of inference used are Modus Tollens and Disjunctive Syllogism.

$$2. \neg(p \wedge q) \rightarrow \neg p \vee \neg q$$

$$\frac{\neg(p \wedge q) \quad p \wedge q}{\therefore \neg p \vee \neg q}$$

The rules of inference used are Modus Tollens and Conjunction.

Prove the following using the rules of inference for quantified propositions:

$$1. \forall x P(x) \rightarrow P(c) \text{ where } P(x) = x > 0$$

$$\frac{\forall x P(x) \quad P(c)}{\therefore P(c)}$$

The rules of inference used are Universal Instantiation.

$$2. \exists x P(x) \rightarrow P(c) \text{ where } P(x) = x > 0$$

$$\frac{\exists x P(x) \quad P(c)}{\therefore P(c)}$$

The rules of inference used are Existential Instantiation.

Turn the following phrases into logical statements using the rules of inference for quantified propositions:

$$1. \text{ For all } x, x > 0$$

$$\forall x(x > 0)$$

$$2. \text{ There exists } x \text{ such that } x > 0$$

$$\exists x(x > 0)$$

$$3. \text{ There exists } x \text{ such that } x > 0 \text{ and } x < 10$$

$$\exists x(x > 0 \wedge x < 10)$$

6 Formal Proofs

This kind of proof is made up of statements which we can infer from previous statements. This we can construct using a table in which we state the logical expression and how we got to it.

$$(p \vee q) \rightarrow (r \wedge s) \quad \text{Premise} \quad (3)$$

$$r \rightarrow t \quad \text{Premise} \quad (4)$$

$$\neg t \quad \text{Premise} \quad (5)$$

$$\neg r \quad \text{Modus Tollens from 2 and 3} \quad (6)$$

$$(\neg p \wedge \neg q) \quad \text{Modus Tollens from 4 and 1} \quad (7)$$

$$\neg p \quad \text{Simplification of 5} \quad (8)$$

7 Informal Proofs

Because a formal proof can get convoluted, we use informal proofs. The main difference here is that we don't need to provide a step-by-step description of how the proof works; we just show how the conclusion is true by describing a few intermediate steps between the theorem and the corollary. (These are called **premises** and **conclusions** in logic.) There are various methods of proof we can use. The two primary categories for these proofs are **direct** and **indirect**.

7.1 Methods of Proof

Upon deciding which axioms to use, you then need to use some method of proof. The most common statements we will be proving are implications.

7.1.1 Direct Proof

We want to prove "**If p, then q**". This is done by assuming that the **premise** is true and then proving that the consequence is also true.

1. We assume p to be true.
2. By applying the laws of inference, we deduce q from p .

7.1.2 Indirect: Proof by Contraposition

If we cannot arrive to the consequence using just the premise, we can use contraposition, which states that $(p \rightarrow q) \equiv (\neg q \rightarrow \neg p)$.

1. We assume that $\neg q$ is true
2. We find a way to infer that $\neg p$ is true
3. Since this verifies the contrapositive, we have proved that $p \rightarrow q$

7.1.3 Indirect: Vacuous and Trivial proofs

If we are saying that $p \rightarrow q$ is **True**, we can assume that:

1. p is False (Vacuous proof)
2. q is True (Trivial proof)

It is enough to prove just one of these (as long as we do not accept the premise being True) we can prove the condition to be True.

1. Example If n is an integer and an even number, then $2n$ must be an integer.

Solution Since an integer multiplied by another integer will also be an integer, we can say that q will be true. This is enough to prove that $p \rightarrow q$ is also True.

7.1.4 Indirect: Proof by Contradiction

This can be used to demonstrate any statement p . We say that $\neg p$ implies a contradiction. A nicer way to think about this is by the following:

$$\neg p \rightarrow \perp \equiv \neg(\neg p) \vee \perp \equiv p \vee \perp \equiv p$$

If we should use proof by contradiction on an implication, we must prove that $p \wedge \neg q \rightarrow \perp$. This is because in this case, we need to show that the negation of the implication is a contradiction.

$$\neg(p \rightarrow q) \equiv \neg(\neg p \vee q) \equiv p \wedge \neg q \equiv \perp$$

1. Example Show that $\sqrt{2}$ is an irrational number.

Solution An irrational number is a fraction of 2 integers. We assume that $\sqrt{2}$ is rational. This would mean that $\sqrt{2} = \frac{p}{q}$ where both p, q are integers. The key thing to assume here is that the fraction is in its smallest form. From the previous equation we can get to $p^2 = 2q^2$ which tells us that p is an even number (if the square of a number even, then so is that number). Further, we can also come to the conclusion that q is even. Since both p, q are even, the fraction is not its simplest form - implying a contradiction.

7.1.5 Proof by Cases

This is a proof which can be used as a part of another proof when proving implications. If we have an implication where the premise consists of disjunctions, we can apply the following:

$$(p_1 \vee p_2 \vee \dots \vee p_n) \rightarrow q \quad (9)$$

$$(p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge \dots \wedge (p_n \rightarrow q) \quad (10)$$

This proof is very helpful when we want to prove that a condition is satisfiable for a group of elements.

We can also use proof by cases for absolute values. If we have $|x| = y$, we can say that $x = y$ or $x = -y$. This is because $|x|$ is the distance from x to 0.

1. Example Prove that if n is not a multiple of 3, $2n$ is not a multiple of 3.

Solution To define a value as not a multiple of 3, we have two formulas $n = 3k + 1$ $n = 3k + 2$. We can reformulate the proposition by saying $(n = 3k + 1) \vee (n = 3k') \rightarrow \neg(2n = 3k'')$. We then have to show that both cases imply the consequence.

7.1.6 Equivalence Proof

As previously mentioned $p \leftrightarrow q = p \equiv q$, so if we want to prove some equivalence, we can use $(p \rightarrow q) \wedge (q \rightarrow p)$.

7.1.7 Existential Statements Proof

We have two strategies for this.

Constructive Proof We show an example that verifies a predicate and apply existential generalization to prove the proposition.

Non Constructive Proof We show that there exists an element which validates the predicate but we don't specify which one. This is most commonly proof by contradiction.

7.1.8 Universal Statements Proof

Proving We show that a predicate holds for an arbitrary value and then apply universal generalization.

Counterexamples To disprove a universal statement we need to find a counterexample. This action is a process of negation, making the US an ES.

7.1.9 Strategy for Proofs

So should you use? It comes down to trial and error, but also practice and intuition.

Trial and error If direct proof doesn't work, try indirect lol.

Complex propositions If there are complex propositions in an implication, they quite possibly can be negated, allowing use to use an **indirect method**.

Using proof by contradiction with conditional statements We show that $\neg(p \rightarrow q)$ implies \perp .

8 Proofs Practice Problems

1. Prove that if n is an integer, then n^2 is an integer.

To prove this we need to show that $p \rightarrow q$ is true. We can assume that p is true and then show that q is also true. We can do this by applying the laws of inference. We can say that $n^2 = (n + 1)(n - 1)$, which is an integer. This is enough to prove that $p \rightarrow q$ is true.

2. Prove that if n is an integer, then n^2 is even.

This is not true. It is not necessary for n^2 to be even. We can show this by taking $n = 3$. This would mean that $n^2 = 9$ which is odd.

3. Define (using truth tables) the disjunction, conjunction, exclusive or, conditional, and biconditional of the propositions p and q :

Truth Table for Disjunction and Conjunction and Exclusive Or:

p	q	$p \vee q$	$p \wedge q$	$p \oplus q$
T	T	T	T	F
T	F	T	F	T
F	T	T	F	T
F	F	F	F	F

Truth Table for Conditional and Biconditional:

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

4. Give a direct proof, a proof by contraposition, and a proof by contradiction of the statement: "If n is even, then $n + 4$ is even."

Direct Proof Let $n =$ an even integer. Then, $n = 2k$ for some integer k . Since $n + 4 = 2k + 4$, $n + 4$ is also even. Thus, if n is even, then $n + 4$ is also even.

Proof by Contraposition Let $n + 4 =$ an odd integer. Then, $n + 4 = 2j + 1$ for some integer j . Since $n = 2j - 3$, n is also odd. Thus, if $n + 4$ is odd, then n is also odd.

Proof by Contradiction Assume that n is even but $n + 4$ is odd. Then, $n = 2k$ for some integer k , but $n + 4 = 2j + 1$ for some integer j . Since $2k = 2j + 1$, $k = j + 1/2$. Since j and k are integers, this is a contradiction. Thus, if n is even, then $n + 4$ is also even.

1. Explain how a proof by cases can be used to prove a result about absolute values, such as the fact that $|xy| = |x||y|$ for all real numbers x and y

A proof by cases can be used to prove a result about absolute values by breaking the statement into cases and proving each case separately. In the case of $|xy| = |x||y|$, the statement is broken into two cases: x and y are both positive, and x and y are both negative. For the first case, it can be shown that $|xy| = xy$, and $|x||y| = xy$, which proves the statement. For the second case, it can be shown that $|xy| = -xy$, and $|x||y| = -xy$, which also proves the statement. Together, these two cases prove that $|xy| = |x||y|$ for all real numbers x and y .

8.1 Problems from the Book

- Show that each of the following conditional statements is a tautology using the fact that a conditional is false exactly when the hypothesis is true and the conclusion is false. (pg 38 Ex140)

1. $(\neg p \wedge (p \vee q)) \rightarrow q$

Solution: If we use the laws of inference we will get that $\neg p \wedge q \rightarrow q$. If we try to make this false, we will should consider the case where the premise is true, for the p is false and q is true. This will make the conclusion false, which is not possible. Therefore, the statement is a tautology.

$$2. ((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$$

Solution: It is pretty easy to see that this is the hypothetical syllogism. We can use the laws of inference to show that $(p \rightarrow q) \wedge (q \rightarrow r)$ is equal to $p \rightarrow r$. Now we just need to show that this is a tautology. We can do this by considering the case where the premise is true, and since the premise and conclusion are the same, we can conclude that the statement is a tautology.

$$3. (p \wedge (p \rightarrow q)) \rightarrow q$$

Solution: We transform the statement into $(p \wedge q) \rightarrow q$. We can see that this is a tautology by considering the case where the premise is true. For the premise to be true, the conclusion also has to be true. This makes the case of the implication being false not possible, which means that the statement is a tautology.

$$4. ((p \vee q) \wedge (p \rightarrow r) \wedge (q \rightarrow r)) \rightarrow r$$

Solution: By simplifying the premise we get that $\top \vee r \rightarrow r$.

- Show that each of the following conditional statements is a tautology using the chain of logical identities. (Pg 38 Ex 16)

$$1. (\neg p \wedge (p \vee q)) \rightarrow q$$

Solution:

$$(\neg p \wedge (p \vee q)) \rightarrow q \quad (11)$$

$$\neg p \wedge p \vee \neg p \wedge q \rightarrow q \quad (12)$$

$$\perp \vee \neg p \wedge q \rightarrow q \quad (13)$$

$$\neg(\neg p \wedge q) \vee q \quad (14)$$

$$p \vee \neg q \vee qp \vee \top \quad (15)$$

$$\top \quad (16)$$

$$2. ((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$$

Solution:

$$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r) \quad (17)$$

$$(\neg p \vee q \wedge \neg q \vee r) \rightarrow (\neg p \vee r) \quad (18)$$

$$\neg p \vee \perp \vee r \rightarrow \neg p \vee r \quad (19)$$

$$\neg(\neg p \vee r) \vee \neg p \vee r \quad (20)$$

$$p \vee \neg r \vee \neg p \vee r \quad (21)$$

$$(p \vee \neg p) \vee (r \vee \neg r) \quad (22)$$

$$\top \vee \top \quad (23)$$

$$\top \quad (24)$$

$$3. (p \wedge (p \rightarrow q)) \rightarrow q$$

Solution:

$$(p \wedge (p \rightarrow q)) \rightarrow q \quad (25)$$

$$(p \wedge (\neg p \vee q)) \rightarrow q \quad (26)$$

$$(p \wedge \neg p \vee p \wedge q) \rightarrow q \quad (27)$$

$$(\perp \vee p \wedge q) \rightarrow q \quad (28)$$

$$\neg(p \wedge q) \vee q \quad (29)$$

$$p \vee \neg q \vee q \quad (30)$$

$$p \vee \top \vee \top \quad (31)$$

$$4. ((p \vee q) \wedge (p \rightarrow r) \wedge (q \rightarrow r)) \rightarrow r$$

Solution:

$$((p \vee q) \wedge (p \rightarrow r) \wedge (q \rightarrow r)) \rightarrow r \quad (32)$$

$$\neg((p \vee q) \wedge (\neg p \vee r) \wedge (\neg q \vee r)) \vee r \quad (33)$$

$$\neg p \wedge \neg q \vee p \wedge \neg r \vee (q \wedge \neg r) \vee r \quad (34)$$

$$(\neg p \wedge \neg q) \vee (p \wedge \neg r) \vee (q \vee r) \wedge (\neg r \vee r) \quad (35)$$

$$(\neg p \wedge \neg q) \vee (p \wedge \neg r) \vee (q \vee r) \wedge \top \quad (36)$$

$$q \vee (\neg p \wedge \neg q) \vee r \vee (p \wedge \neg r) \quad (37)$$

$$q \vee \neg p \wedge q \vee \neg q \vee r \vee p \wedge r \vee \neg r \quad (38)$$

$$q \vee \neg p \wedge \top \vee r \vee p \wedge \top \quad (39)$$

$$q \vee \neg p \vee r \vee p \quad (40)$$

$$q \vee r \vee \top \quad (41)$$

$$\top \quad (42)$$

- Express the negation of the propositions using quantifiers:

$$1. \forall x(x > 1)$$

$$\text{Solution: } \exists x(x \leq 1)$$

$$2. \forall x((x < -1) \vee (x > 2))$$

$$\text{Solution: } \exists x((x \geq -1) \wedge (x \leq 2))$$

- Determine whether $\forall x(P(x) \rightarrow Q(x))$ is equivalent to $\forall xP(x) \rightarrow \forall xQ(x)$. (pg 59 ex 45) Solution:

$$\forall x(P(x) \rightarrow Q(x)) \equiv \forall xP(x) \rightarrow \forall xQ(x) \quad (43)$$

$$\neg \exists x(P(x) \rightarrow Q(x)) \vee \exists xP(x) \rightarrow \exists xQ(x) \quad (44)$$

$$\neg \exists x(P(x) \rightarrow Q(x)) \vee \neg \exists xP(x) \vee \exists xQ(x) \quad (45)$$

$$\neg \exists x(P(x) \rightarrow Q(x)) \vee \neg \exists xP(x) \vee \neg \exists xQ(x) \quad (46)$$

- Use rules of inference to show that the hypotheses “If it does not rain or if it is not foggy, then the sailing race will be held and the lifesaving demonstration will go on,” “If the sailing race is held, then the trophy will be awarded,” and “The trophy was not awarded” imply the conclusion “It rained.”

First we identify the premises:

- p = it rained
- q = it was foggy
- r = the race was held
- s = the lifesaving demonstration went on
- t = the trophy was awarded

$$\neg p \vee \neg q \rightarrow r \wedge s \quad (47)$$

$$r \rightarrow t \quad (48)$$

$$\neg t \rightarrow \neg r \quad \text{from 2nd and 3rd (modus tollens)} \quad (49)$$

$$(\neg r \wedge s) \quad \text{from 1st and 4th} \quad (50)$$

$$\neg(\neg p \vee \neg q) \quad (51)$$

$$p \wedge q \quad (52)$$

$$p \quad \text{it rained} \quad (53)$$

- Find the argument form for the following argument and determine whether it is valid. Can we conclude that the conclusion is true if the premises are true?

If George does not have eight legs, then he is not a spider.

George is a spider

George has eight legs.

Solution: The form of the argument is Modus Tollens, which is valid.

- What rules of inference are used in this famous argument? “All men are mortal. Socrates is a man. Therefore, Socrates is mortal.”

$$\forall x(M(x) \rightarrow Mo(x))$$

$$\text{Solution: } \frac{M(S)}{Mo(S)}$$

Here we make use of universal instantiation and then modus ponens.

- Use a direct proof to show that every odd integer is the difference of two squares. Solution:

$$n = 2k + 1 \quad (54)$$

$$n = (k + 1)^2 - k^2 \quad (55)$$

$$n = k^2 + 2k + 1 - k^2 \quad (56)$$

$$n = 2k + 1 \quad (57)$$

9 Set Theory

9.1 Sets

A set is a collection of objects. We use the notation $\{a, b, c\}$ to denote a set. We can also use the notation $\{x \in S \mid P(x)\}$ to denote a set of elements x in S that satisfy the predicate $P(x)$. A set can be empty, which is denoted by \emptyset .

A more concrete definition of a set using quantifiers is as follows:

$$\forall x((x \in S) \leftrightarrow P(x))$$

We also often use **interval definitions** to define sets. These are defined as follows:

$$\{x \in \mathbb{R} \mid a \leq x \leq b\}$$

A common way to define an interval is with brackets and parenthesis. For example, (a, b) is an open interval, while $[a, b]$ is a closed interval.

Two sets are equal if they have the same elements. We denote this as $A = B$. A more rigorous definition of this is as follows:

$$A = B \equiv \forall x(x \in A \leftrightarrow x \in B)$$

The equality of two sets can also be defined using subsets:

$$A = B \equiv A \subseteq B \wedge B \subseteq A$$

A way to disprove the equality of two sets is by showing that they have at least one element that is not in the other set. This is called a **counterexample**. It can be represented with an existential quantifier and XOR: $\exists x(x \in A \oplus x \in B)$.

Some important elements are the aforementioned empty set \emptyset and the universal set \mathbb{U} . The universal set is the set of all elements in a domain. The empty set is the set that contains no elements. The Domains, mentioned at the beginning are also sets.

9.2 Subsets

A subset is a set that is contained in another set. We denote this as $A \subseteq B$. A more rigorous definition of this is as follows:

$$A \subseteq B \equiv \forall x(x \in A \rightarrow x \in B)$$

For all sets, it is true that $A \subseteq A$ and $\emptyset \subseteq A$. Which means that every set is a subset of itself and the empty set is a subset of every set. We can also say that $A \subseteq B$ and $B \subseteq A$ implies $A = B$. This is called the **subset equality**.

What can be confusing is the proper subset. A proper subset is a subset that is not equal to the set. We denote this as $A \subset B$. A more rigorous definition of this is as follows:

$$A \subset B \equiv A \subseteq B \wedge A \neq B$$

A quantified version of this is as follows:

$$\forall x((x \in A) \rightarrow (x \in B) \wedge (A \neq B))$$

9.3 Set Cardinality

This is not the absolute value of a set. It is the number of elements in a set. We denote this as $|A|$. A more rigorous definition of this is as follows:

$$|A| \equiv \#\{x \in A \mid x \in A\}$$

9.4 Power Set

The power set is the set of all subsets of a set. We denote this as $P(A)$. What is meant by all subsets is that the power set contains the empty set and the universal set as well as all the subsets of the set. A more rigorous definition of this is as follows:

$$P(A) \equiv \{B \mid B \subseteq A\}$$

The cardinality of a power set is $2^{|A|}$.

An example of this is the power set of $\{1, 2, 3\}$ is $\{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$.

What is important to remember is that the power set of a power set is the original set. This is because the power set of a set is the set of all subsets of a set. So the power set of the power set is the set of all subsets of all subsets of a set. This is the original set.

9.5 Cartesian Product

The cartesian product is the set of all ordered pairs of elements from two sets. We denote this as $A \times B$. In other words, the cartesian product is the

set of all possible combinations of elements from two sets. A more rigorous definition of this is as follows:

$$A \times B \equiv \{(a, b) \mid a \in A \wedge b \in B\}$$

The cardinality of the cartesian product is $|A| \times |B|$.

An example of this is the cartesian product of $\{1, 2\}$ and $\{3, 4\}$ is $\{(1, 3), (1, 4), (2, 3), (2, 4)\}$.

The relation of the Cartesian product to the power set is that the power set of a cartesian product is the cartesian product of the power sets. This is because the power set of a set is the set of all subsets of a set. So the power set of the cartesian product is the set of all subsets of all ordered pairs of elements from two sets. This is the cartesian product of the power sets. A mathematical proof of this is as follows:

$$P(A \times B) = \{C \mid C \subseteq A \times B\} = \{C \mid C \subseteq \{(a, b) \mid a \in A \wedge b \in B\}\} = \{C \mid C \subseteq \{(a, b) \mid a \in P(A) \wedge b \in P(B)\}\}$$

Or simply: $P(A \times B) = P(A) \times P(B)$.

9.6 Set Operations

Set operations are operations that can be performed on sets. The most common set operations are union, intersection, difference, and complement.

It is important to remember that if we have an expression like $AA \cup B$, something that could be tricky is that the order of operations is not the same as in arithmetic. In arithmetic, we would do the multiplication and division first, then addition and subtraction. In set operations, we do the complement first, then the difference, then the intersection, and finally the union. So the expression $AA \cup B$ would be read as $A(A \cup B)$.

Once again, **the order of operations for sets** is:

1. Complement
2. Difference
3. Intersection
4. Union

9.7 Union

Simply explained, the union of two sets is the set of all elements that are in either set. We denote this as $A \cup B$. A more rigorous definition of this is as follows:

$$A \cup B \equiv \{x \mid x \in A \vee x \in B\}$$

And using quantifiers:

$$\forall((x \in A \cup B) \equiv (x \in A \vee x \in B))$$

The cardinality of the union is $|A| + |B| - |A \cap B|$. The reason why we have to subtract the cardinality of the intersection is because the intersection is counted twice. Once in A and once in B .

An example of this is the union of $\{1, 2\}$ and $\{2, 3\}$ is $\{1, 2, 3\}$. The reason why this is the union is because $1 \in A$, $2 \in A$, $2 \in B$, and $3 \in B$. So $1, 2, 3$ are all in the union.

9.8 Intersection

The intersection of two sets is the set of all elements that are in both sets. We denote this as $A \cap B$. A more rigorous definition of this is as follows:

$$A \cap B \equiv \{x \mid x \in A \wedge x \in B\}$$

The cardinality of the intersection is $|A \cap B|$. What is important to remember is that the intersection of a set with itself is the set itself. This is because the intersection of a set with itself is the set of all elements that are in the set and the set itself. So the intersection of a set with itself is the set itself.

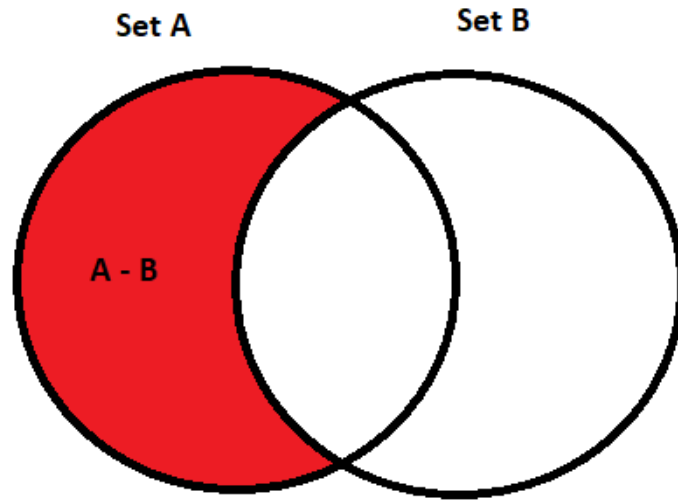
An example of this is the intersection of $\{1, 2\}$ and $\{2, 3\}$ is $\{2\}$. The reason why this is the intersection is because $2 \in A$ and $2 \in B$. So 2 is in the intersection.

9.9 Difference

The difference of two sets is the set of all elements that are in the first set but not the second set. We denote this as $A \setminus B$. A more rigorous definition of this is as follows:

$$A \setminus B \equiv \{x \mid x \in A \wedge x \notin B\}$$

A visual representation of this is as follows:



The cardinality of the difference is $|A \setminus B|$ or $|A| - |A \cap B|$. The reason why we have to subtract the cardinality of the intersection is because the intersection is counted twice. Once in A and once in B .

An example of this is the difference of $\{1, 2\}$ and $\{2, 3\}$ is $\{1\}$. The reason why this is the difference is because $1 \in A$ and $1 \notin B$. So 1 is in the difference.

9.10 Complement

The complement of a set is the set of all elements that are not in the set. We denote this as A^c . A more rigorous definition of this is as follows:

$$A^c \equiv \{x \mid x \notin A\}$$

In terms of the universal set, the complement of a set is the set of all elements that are not in the set and are in the universal set. It is denoted as $A^c = U \setminus A$. A more rigorous definition of this is as follows:

$$A^c \equiv \{x \mid x \in U \wedge x \notin A\}$$

The cardinality of the complement is $|U| - |A|$. The reason why we have to subtract the cardinality of the set is because the set is counted twice. Once in U and once in A .

An example of this is the complement of $\{1, 2\}$ is $\{3, 4\}$. The reason why this is the complement is because $3 \in U$ and $3 \notin A$. So 3 is in the complement. This of course depends on the universal set being $\{1, 2, 3, 4\}$.

9.11 Logical Operations with Sets

We can relate sets to logical operations. The most common logical operations are conjunction, disjunction, and negation. We can relate these to sets as follows:

$$A \cup B \sim p \vee q$$

$$A \cap B \sim p \wedge q$$

$$\bar{A} \sim \neg p$$

The reason why we can do this is that the definition of a set, is based on a predicate. So we can relate the set to the predicate and the predicate to the logical operation. For example, the definition of a set is as follows:

$$A \equiv \{x \mid x \in A\}$$

So we can relate the set to the predicate as follows:

$$A \equiv \{x \mid x \in A\} \equiv \{x \mid p(x)\}$$

10 Functions

A function, is the assignment of a single value $b \in B$ to each element a in a set A . We denote this as $f : A \rightarrow B$. We can also call a function a mapping, transformation or a black box (in some cases even an oracle). A more rigorous definition of this is as follows:

$$f : A \rightarrow B \equiv \{f(a) \mid a \in A\}$$

We are mostly interested in what makes a function a function. So we will define what makes a function a function. A function is a function if the following conditions are met:

1. The domain of the function is a set
2. The range of the function is a set
3. The function is one-to-one
4. The function is onto

A mathematical way to express this is as follows:

$$\forall a \in A \exists! b \in B \wedge f(a) = b$$

The reasons why some assignments might not be functions is that not all elements in A are assigned to an element in B . This is called a partial function. Another reason why some assignments might not be functions is that not all elements in B are assigned to an element in A . This is called a surjection. Another reason why some assignments might not be functions is that not all elements in A are assigned to a unique element in B . This is called a bijection.

How can we tell if a function is a function? We can tell if a function is a function by checking if the function is one-to-one and onto. If the function is one-to-one and onto, then the function is a function. If the function is not one-to-one and onto, then the function is not a function.

10.1 Domain, Codomain & Range

If $f(a) = b$, then we call a the **pre-image** and b the **image** of f . We call A the **domain** of f and B the **codomain** of f . We call $f(A)$ the **range** of f .

The main difference between the **codomain** and the **range** is that the **codomain** is the set of all possible images of f and the **range** is the set of all images of f .

10.2 One-to-One

A function is one-to-one if each element in the range is assigned to a unique element in the domain. We denote this as $f : A \rightarrow B$ is one-to-one. A more rigorous definition of this is as follows:

$$f : A \rightarrow B \text{ is one-to-one} \equiv \forall a_1, a_2 \in A \wedge f(a_1) = f(a_2) \implies a_1 = a_2$$

$$\text{or simply: } f(a_1) = f(a_2) \implies a_1 = a_2.$$

If we want to check if a functions is one-to-one, we have to test if $f(a_1) = f(a_2) \implies a_1 = a_2$ for all $a_1, a_2 \in A$. If this is true for all $a_1, a_2 \in A$, then the function is one-to-one.

10.3 Onto

A function is onto if each element in the codomain is assigned to an element in the domain. We denote this as $f : A \rightarrow B$ is onto. This is to say that for

each element in the codomain, there is an element in the domain that maps to it. A more rigorous definition of this is as follows:

$$f : A \rightarrow B \text{ is onto} \equiv \forall b \in B \exists a \in A \wedge f(a) = b$$

or simply: $\forall b \in B \exists a \in A \wedge f(a) = b$.

To see if a function is onto, we have to test if $\forall b \in B \exists a \in A \wedge f(a) = b$. If this is true for all $b \in B$, then the function is onto. For example if $f : \{1, 2\} \rightarrow \{1, 2, 3\}$, then $f(1) = 1$, $f(2) = 2$ and $f(3) = 3$. So f is onto. If $f : \{1, 2\} \rightarrow \{1, 2\}$, then $f(1) = 1$, $f(2) = 2$ and $f(3) = 3$. So f is not onto. In this case, the function is not onto because 3 is not assigned to an element in the domain.

10.4 Composite Functions

We can combine functions to create composite functions. We can combine functions by applying one function to the output of another function. We denote this as $f \circ g$. We can also call this function composition. A more rigorous definition of this is as follows:

$$f \circ g \equiv \{f(g(a)) \mid a \in A\}$$

If the **codomain** of g is the **domain** of f , then we can combine g and f to create a composite function. We denote this as $f \circ g : A \rightarrow C$.

When it comes to functions properties after composition we have the following properties:

1. $f \circ g$ is one-to-one if f is one-to-one and g is one-to-one
2. $f \circ g$ is onto if f is onto and g is onto

If some of the functions are not one-to-one or onto, then the composite function is not one-to-one or onto.

10.5 Inverse Functions

An inverse function is a function that undoes the effects of another function. We denote this as f^{-1} . The definition is as follows:

$$f^{-1} \equiv \{a \mid f(a) = b\}$$

If f is one-to-one and onto, then f^{-1} is also one-to-one and onto.

10.6 Partial Functions

A partial function is a function that is not one-to-one or onto. We denote this as $f : A \rightarrow B$ is a partial function. Definition:

$$f : A \rightarrow B \text{ is a partial function} \equiv \exists a_1, a_2 \in A \wedge f(a_1) = f(a_2) \wedge a_1 \neq a_2$$

or simply: $\exists a_1, a_2 \in A \wedge f(a_1) = f(a_2) \wedge a_1 \neq a_2$. If this is true for some $a_1, a_2 \in A$, then f is a partial function.

10.7 Practice Questions

1. Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be defined by $f(x) = x^2$. Is f one-to-one? Is f onto? Is f a partial function?

Solution f is one-to-one because $f(x) = f(y) \implies x = y$. f is not onto because $f(x) = x^2$ and x^2 is not in the domain. f is not a partial function because $f(x) = f(y) \implies x = y$.

2. Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be defined by $f(x) = x^2$. Is f^{-1} one-to-one? Is f^{-1} onto? Is f^{-1} a partial function?

Solution f^{-1} is one-to-one because $f^{-1}(x) = f^{-1}(y) \implies x = y$. f^{-1} is onto because $f^{-1}(x) = x^2$ and x^2 is in the domain. f^{-1} is not a partial function because $f^{-1}(x) = f^{-1}(y) \implies x = y$.

3. Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be defined by $f(x) = x^2$. Is $f \circ f$ one-to-one? Is $f \circ f$ onto? Is $f \circ f$ a partial function?

Solution $f \circ f$ is one-to-one because f is one-to-one and f is one-to-one. $f \circ f$ is onto because f is onto and f is onto. $f \circ f$ is not a partial function because $f \circ f$ is one-to-one and $f \circ f$ is onto.

4. Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be defined by $f(x) = x^2$. Is $f \circ f^{-1}$ one-to-one? Is $f \circ f^{-1}$ onto? Is $f \circ f^{-1}$ a partial function?

Solution $f \circ f^{-1}$ is one-to-one because f is one-to-one and f^{-1} is one-to-one. $f \circ f^{-1}$ is onto because f is onto and f^{-1} is onto. $f \circ f^{-1}$ is not a partial function because $f \circ f^{-1}$ is one-to-one and $f \circ f^{-1}$ is onto.

11 Sequences

A sequence is a set of elements, which can be continued by using intuition or mathematical patterns. We denote this as $\{a_n \mid n \in \mathbb{N}\}$. We will mostly focus on **arithmetic** sequences and **geometric** sequences.

11.1 Arithmetic Sequences

An arithmetic sequence is a sequence where the difference between two consecutive terms is constant. We denote this as $\{a_n \mid n \in \mathbb{N}\}$ where $a_n = a_1 + (n-1)d$ for some $a_1, d \in \mathbb{R}$. We can also write this as $a_n = a_1 + nd$, where n is the **nth** term of the sequence and d is the **common difference**.

An arithmetic sequence is defined by the following properties:

1. $a_n = a_1 + (n-1)d$
2. $a_n = a_1 + nd$

This sequence can also be thought of as a linear function. We can write this as $a_n = f(n)$ where $f(n) = a_1 + (n-1)d$. The main elements of a sequence are the **first term** and the **common difference**.

The expansion for the arithmetic sequence is as follows:

$$a_1 = a_1 + 0d = a_1 \tag{58}$$

$$a_2 = a_1 + 1d = a_1 + d \tag{59}$$

$$a_3 = a_1 + 2d = a_1 + 2d \tag{60}$$

$$a_4 = a_1 + 3d = a_1 + 3d \tag{61}$$

$$\vdots = \vdots \tag{62}$$

$$a_n = a_1 + (n-1)d = a_1 + (n-1)d \tag{63}$$

11.2 Geometric Sequences

A geometric sequence is a sequence where the ratio between two consecutive terms is constant. We denote this as $\{a_n \mid n \in \mathbb{N}\}$ where $a_n = a_1 r^{n-1}$ for some $a_1, r \in \mathbb{R}$. We can also write this as $a_n = a_1 r^n$, where n is the **nth** term of the sequence and r is the **common ratio**.

An expansion of such as sequence can look something like this:

$$a_1 = a_1 \quad (64)$$

$$a_2 = a_1 r \quad (65)$$

$$a_3 = a_1 r^2 \quad (66)$$

$$a_4 = a_1 r^3 \quad (67)$$

$$\vdots = \vdots \quad (68)$$

$$a_n = a_1 r^{n-1} \quad (69)$$

11.3 Recurrence Relations

A recurrence relation is a sequence that is defined by a formula. We denote this as $\{a_n \mid n \in \mathbb{N}\}$ where $a_n = f(a_{n-1})$ for some $f : \mathbb{R} \rightarrow \mathbb{R}$. It takes the previous term and applies a function to it to get the next term.

We can be told to find the closed formula for a sequence. This is a formula that can be used to find the n th term of a sequence. To do this, we need to use iteration and induction. We can use the following rules to find the closed formula for a sequence:

1. $a_n = a_1 + (n - 1)d$ where $a_1, d \in \mathbb{R}$
2. $a_n = a_1 r^{n-1}$ where $a_1, r \in \mathbb{R}$
3. $a_n = f(a_{n-1})$ where $f : \mathbb{R} \rightarrow \mathbb{R}$
4. $a_n = f(a_{n-1}, a_{n-2})$ where $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$

In short, we need to find the pattern in the sequence and use that to find the closed formula.

12 Summations

A summation is a way to add up a sequence of numbers. We denote this as $\sum_{i=1}^n a_i$. There are many ways to define a summation. To find the sum of a sequence, we can use the following rules:

1. $\sum_{i=1}^n i = \frac{n(n+1)}{2}$
2. $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$
3. $\sum_{i=1}^n i^3 = \frac{n^2(n+1)^2}{4}$

$$4. \sum_{i=1}^n C * f(i) = \frac{C}{1-r}$$

If we subtract 1 from the upper bound, we can find the sum of the sequence. For example, $\sum_{i=1}^n i = \sum_{i=0}^{n-1} i + n = \sum_{i=0}^{n-1} i + \sum_{i=n}^n i = \sum_{i=0}^{n-1} i + n = \frac{n(n-1)}{2} + n = \frac{n(n+1)}{2}$.

If we have nested summations, we can expand the far-right summation and include it in each iteration of the left summation.

13 Divisibility

A number is divisible by another number if the remainder is 0. We denote this as $a \mid b$ where a is the divisor and b is the dividend. We can also write this as $a = q \cdot d + r$ where r is the remainder and q is the quotient. It is important to note that r is always less than d and always positive.

The elements in the primary equation are as follows:

1. a is the dividend
2. d is the divisor
3. q is the quotient
4. r is the remainder

From this equation we gain two new operations, the **division** operation and the **modulus** operation. We can use these operations to find the remainder and quotient of a division problem.

If we perform division of a negative number, we cannot have a negative remainder. To solve this, we can add the divisor to the dividend until the remainder is positive. This is called the **floor division** operation. In other words, we want to 'overshoot' with the dividend and then compensate with the remainder.

- Division Function: This function is onto but not one-to-one
- Modulus Function: This function neither one-to-one or onto

13.1 Divisibility Rules

We can use divisibility rules to determine if a number is divisible by another number. The rules are as follows:

1. If a divides both b and c , then a divides $b + c$. Expression: $a \mid b \wedge a \mid c \Rightarrow a \mid (b + c)$
2. If a divides b , then a divides any multiple of b . Expression: $a \mid b \Rightarrow a \mid bm$ for some $m \in \mathbb{N}$
3. If a divides b and b divides c , then a divides c . Expression: $a \mid b \wedge b \mid c \Rightarrow a \mid c$

14 Modular Arithmetic

Modular arithmetic uses the modulus operation to find the remainder of a division problem. We use the notation $a \equiv b \pmod{m}$ to denote that a is congruent to b modulo m .

If we want to identify if two integers are congruent modulo m , we can test this in the following way:

1. $a \equiv b \pmod{m} \wedge b \equiv c \pmod{m} \Rightarrow a \equiv c \pmod{m}$

In other words, if the remainder of the modulus operation is the same, then the two numbers are congruent modulo m .

14.1 Rules of Modular Arithmetic

We can use the following rules to perform modular arithmetic:

1. $a \equiv b \pmod{m} \wedge c \equiv d \pmod{m} \Rightarrow a + c \equiv b + d \pmod{m}$
2. $a \equiv b \pmod{m} \wedge c \equiv d \pmod{m} \Rightarrow ac \equiv bd \pmod{m}$
3. $a \equiv b \pmod{m} \wedge c \equiv d \pmod{m} \Rightarrow a - c \equiv b - d \pmod{m}$
4. $a \equiv b \pmod{m} \wedge c \equiv d \pmod{m} \Rightarrow a/c \equiv b/d \pmod{m}$
5. $a \equiv b \pmod{m} \wedge c \equiv d \pmod{m} \Rightarrow a^c \equiv b^d \pmod{m}$

The difference between congruency and the modulo operation is that congruency is a relation between two numbers, while the modulo operation is an operation that returns a number. For example, $a \equiv b \pmod{m}$ is a relation between a and b , while $a \pmod{m}$ is an operation that returns a number.

14.2 Inverse of a Modulo

We can find the inverse of a modulo by using the following equation: $a \cdot a^{-1} \equiv 1 \pmod{m}$. In other words, if we multiply the inverse of a modulo by the modulo, we get 1 modulo m .

To find the inverse of a modulo, we can use the Extended Euclidean Algorithm. This algorithm is as follows:

1. Set $a = m \cdot q + r$ where q is the quotient and r is the remainder
2. Set $b = a \cdot q + r$ where q is the quotient and r is the remainder
3. Repeat step 2 until $r = 0$
4. The inverse of m modulo a is q where q is the quotient of the last step

For example, if we want to find the inverse of 3 modulo 7, we can use the Extended Euclidean Algorithm as follows:

1. $7 = 3 \cdot 2 + 1$
2. $3 = 1 \cdot 3 + 0$
3. $r = 0$ so we stop
4. The inverse of 3 modulo 7 is 3

14.3 Practice Problems

1. $2 \equiv 3 \pmod{5}$
Solution: False, 2 is not congruent to 3 modulo 5
2. $3 \equiv 4 \pmod{5}$
Solution: False, 3 is not congruent to 4 modulo 5
3. $4x \equiv 7 \pmod{5}$
Solution: This is true when $x = 0.5$ because $4(0.5) \equiv 7 \pmod{5}$ is true.
4. $x \equiv 3 \pmod{5} \wedge x \equiv 4 \pmod{5}$
Solution: This is not true because $3 \not\equiv 4 \pmod{5}$ so $x \not\equiv 3 \pmod{5} \wedge x \not\equiv 4 \pmod{5}$

Practice problems using powers of modular arithmetic:

1. $2^3 \equiv 8 \pmod{5}$
Solution: This is true because $2^3 = 8 \equiv 8 \pmod{5}$

15 Integer Representation

Integers can be represented in many ways. In every day life, we use the base 10 system. This means that we use the digits 0-9 to represent numbers. We can also use the base 2 system. This means that we use the digits 0-1 to represent numbers. We can also use the base 16 system. This means that we use the digits 0-9 and A-F to represent numbers. We can also use the base 26 system. This means that we use the digits A-Z to represent numbers.

If we want to convert the value of base b to base b' , we can use the following formula:

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$$

The general algorithm for calculating the number representation is as follows:

1. Divide the number by the base
2. Take the remainder and add it to the list of digits
3. Repeat until the number is 0

In mathematics:

$$n = a_0 + b q_0 \tag{70}$$

$$n = a_1 + b(a_1 + b q_1) \tag{71}$$

$$n = a_2 + b(a_1 + b(a_2 + b q_2)) \tag{72}$$

16 Binary Representation

Binary representation is the representation of numbers in base 2. We can use the following rules to convert from base 10 to base 2:

1. Divide the number by 2
2. Take the remainder and add it to the list of digits
3. Repeat until the number is 0

To then put together the binary string we take each bit in reverse order and put it together. For example, the binary representation of 13 is 1101.

To make life a bit easier, we can use a table to convert from base 10 to base 2. The table is made up of the powers of 2 and the digits 0-1.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

17 Hexadecimal Representation

Hexadecimal representation is the representation of numbers in base 16. We can use the following rules to convert from base 10 to base 16:

1. Divide the number by 16
2. Take the remainder and add it to the list of digits
3. Repeat until the number is 0

To then put together the hexadecimal string we take each digit in reverse order and put it together. For example, the hexadecimal representation of 15816 is 3DC8. The steps taken to get to this number are as follows:

1. $15816 / 16 = 988$ remainder 8
2. $988 / 16 = 61$ remainder 12 (12 is C in hexadecimal)
3. $61 / 16 = 3$ remainder 13 (13 is represented as D in hexadecimal)
4. $3 / 16 = 0$ remainder 3
5. 3DC8

18 Prime Numbers

A prime number is a number that is only divisible by 1 and itself. We can use the following rules to determine if a number is prime:

1. If n is divisible by a , then n is not prime
2. If n is not divisible by a , then n is prime

Any number that is not prime is called a composite number. This relates to the Fundamental Theorem of Arithmetic. This theorem states that every integer greater than 1 can be expressed as a product of prime numbers. This theorem is important because it allows us to factorize numbers.

How to determine if a number is a prime or composite? We can use the following proposition:

$$(n \text{ is composite}) \Leftrightarrow \exists a \in \mathbb{N}((a \mid n) \wedge (a \neq 1) \wedge (a \neq n))$$

19 Algorithms

An algorithm is a collection of instructions that are made to solve a problem. To create any algorithm, we need three basic components:

Sequence A sequence is a list of instructions that are executed in order

Selection A selection is a list of instructions that are executed based on a condition

Iteration An iteration is a list of instructions that are executed multiple times

19.1 Complexity of Algorithms

The complexity of an algorithm is given by the relation between the input size and the number of steps required to solve the problem. We can use the following notations to describe the complexity of an algorithm:

$O(n)$ This notation describes the worst case scenario for an algorithm. This means that the algorithm will take at least this amount of time to solve the problem.

$\Omega(n)$ This notation describes the best case scenario for an algorithm. This means that the algorithm will take at most this amount of time to solve the problem.

$\Theta(n)$ This notation describes the average case scenario for an algorithm. This means that the algorithm will take this amount of time to solve the problem.

We will not need omega and theta notation for the exam, but it is good to know.

19.2 Big O Notation

Big O notation is used to describe the worst case scenario for an algorithm. We can use the following rules to determine the complexity of an algorithm:

The idea of this notation is that we can use it to compare the complexity of different algorithms. For example, if we have two algorithms that solve the same problem, we can use big O notation to determine which algorithm is better. If one algorithm is $O(n)$ and the other is $O(n^2)$, then the first algorithm is better than the second algorithm.

Algorithm	Complexity	Example
$O(1)$	Constant	Finding the first element in a list
$O(\log n)$	Logarithmic	Binary search
$O(n)$	Linear	Finding the maximum value in a list
$O(n \log n)$	Linearithmic	Merge sort
$O(n^2)$	Quadratic	Bubble sort
$O(n^3)$	Cubic	Matrix multiplication
$O(2^n)$	Exponential	Tower of Hanoi
$O(n!)$	Factorial	Travelling salesman problem

Contrary to intuition, to solve a large problem you should always use the most efficient algorithm, but for a smaller size, higher complexity algorithms are better.

20 Greatest Common Divisor

The greatest common divisor of two numbers is the largest number that divides both numbers. We can easily find this if we know the factorizations of each number. Another way to find the greatest common divisor is to use Euclid's algorithm. This algorithm is as follows:

1. Let a be the larger number and b be the smaller number
2. Divide a by b and take the remainder
3. If the remainder is 0, then b is the greatest common divisor
4. If the remainder is not 0, then set a to b and b to the remainder
5. Repeat until the remainder is 0

An example of this algorithm is as follows:

1. Let $a = 24$ and $b = 16$
2. $24 \div 16 = 1$ remainder 8
3. $16 \div 8 = 2$ remainder 0
4. $b = 8$ is the greatest common divisor

20.1 Least Common Multiple

The least common multiple of two numbers is the smallest number that is divisible by both numbers. We can easily find this if we know the factorizations of each number. Another way to find the least common multiple is to use the following formula:

$$\text{lcm}(a, b) = \frac{a \cdot b}{\text{gcd}(a, b)}$$

21 Applications of Number Theory

Number theory is used in many different fields. Some of these fields are as follows: cryptography, coding theory, and combinatorics.

21.1 Databases

If we know the number of records in a database and how much space they take up, as well as the starting position, we can use the following formula to determine the position of the n th record:

$$\text{position} = \text{starting position} + \text{record size} \cdot (n - 1)$$

21.2 Hash Functions

If we want to store some records into a database, we can use a hash function which will map each record to a unique location in the database. This is done by taking the record and applying a hash function to it. The hash function will then return a number which is the location of the record in the database. We might use the *mod* operator to create a hash function.

If each entry has some numerical value we can use *mod10* to map it to a location in the database. This method is not very practical because we might get some sort of collision. A collision is when two records map to the same location in the database. To avoid collisions we can make use of a secondary hash function.

21.3 Asymmetric Encryption

Asymmetric encryption is a method of encryption that uses two keys. One key is used to encrypt the message and the other key is used to decrypt the message. The two keys are called the **public key** and the **private key**.

The public key is used to encrypt the message and the private key is used to decrypt the message. The public key is made public and the private key is kept secret.

22 Relations

A relation is a set of ordered pairs. It is also a subset of the Cartesian product of the set. We can use the following notation to represent a relation:

$$R = \{(x, y) \mid x \in A \wedge y \in B\}$$

The easiest way to represent a relation is to use a table. The table will have the elements of the first set as the rows and the elements of the second set as the columns. If the ordered pair (x, y) is in the relation, then we will put a 1 in the cell that corresponds to the row x and the column y . If the ordered pair (x, y) is not in the relation, then we will put a 0 in the cell that corresponds to the row x and the column y .

Here is an example of a relation:

$$R = \{(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)\}$$

We can represent this relation as a table as follows:

	1	2	3
1	0	1	1
2	1	0	1
3	1	1	0

The relationship between functions and relations is that a function is a relation where each ordered pair has a unique first element.

To find the cardinality of a relation for a **single set**, we can use the following formula:

$$2^{n^2}$$

This is because there are 2 possibilities for each ordered pair. The ordered pair can either be in the relation or it can not be in the relation. If we have n elements in the set, then there are n^2 ordered pairs. Therefore, there are 2^{n^2} possible relations.

This should also be equivalent to the power set of the Cartesian product of the set.

22.1 Reflexive Relations

A relation is reflexive if it contains the ordered pair (x, x) for every element x in the set. We can use the following notation to represent a reflexive relation:

$$R = \{(x, y) \mid x \in A \wedge y \in A \wedge x = y\}$$

As a matrix, it would look like this:

	1	2	3
1	1	0	0
2	0	1	0
3	0	0	1

22.2 Symmetric Relations

A relation is symmetric if it contains the ordered pair (x, y) for every ordered pair (y, x) . We can use the following notation to represent a symmetric relation:

$$R = \{(x, y) \mid x \in A \wedge y \in A \wedge (y, x) \in R\}$$

As a matrix, it would look like this (or any other symmetric relation):

	1	2	3
1	0	1	1
2	1	1	1
3	1	1	0

22.3 Antisymmetric Relations

If a relation is antisymmetric, it will contain some order pair (x, y) but not (y, x) for every ordered pair (x, y) . We can use the following notation to represent an antisymmetric relation:

$$R = \{(x, y) \mid x \in A \wedge y \in A \wedge (x, y) \in R \wedge (y, x) \notin R\}$$

22.4 Transitive Relations

A relation is transitive if it contains the ordered pair (x, z) for every ordered pair (x, y) and (y, z) . We can use the following notation to represent a transitive relation:

$$R = \{(x, z) \mid x \in A \wedge z \in A \wedge (x, y) \in R \wedge (y, z) \in R\}$$

What its saying is that the relation will be transitive if for every n steps, we can get from x to z in just one step.

22.5 Composite Relations

We can combine two relations to create a composite relation. The composite relation will contain the ordered pair (x, z) if the first relation contains the ordered pair (x, y) and the second relation contains the ordered pair (y, z) . We can use the following notation to represent a composite relation:

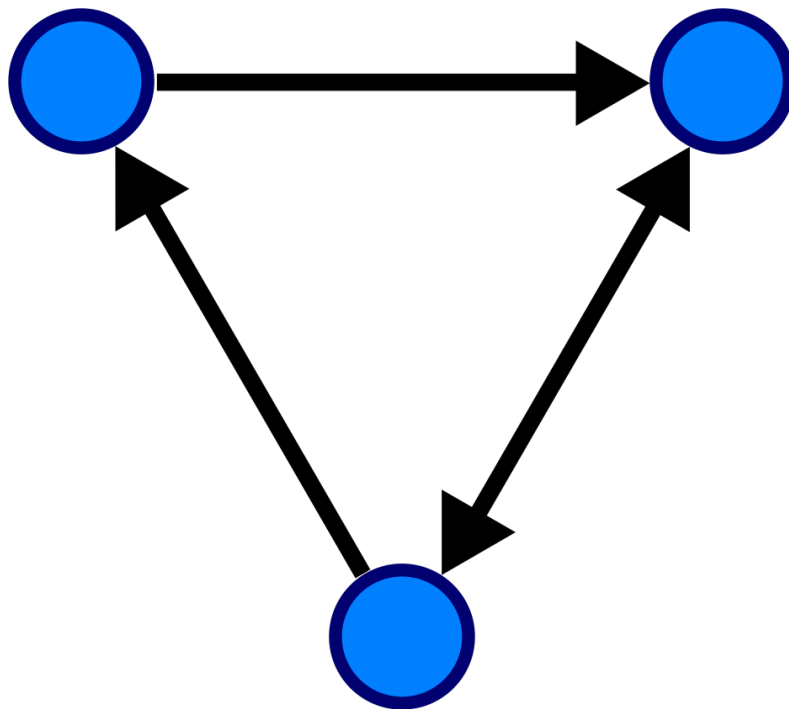
$$R \circ S = \{(x, z) \mid x \in A \wedge z \in A \wedge (x, y) \in R \wedge (y, z) \in S\}$$

We can use **powers of a binary relation** to represent the composite relation. The n th power of a binary relation will contain the ordered pair (x, z) if the relation contains the ordered pair (x, y) and the relation contains the ordered pair (y, z) n times.

23 Directed Graphs

If we want to represent a relation, we can use a directed graph. A directed graph is a graph where the edges have a direction. The edges can be represented by arrows. The arrows will point from the first element of the ordered pair to the second element of the ordered pair. If we have an ordered pair (x, y) in the relation, then we will have an arrow from x to y .

An example of a directed graph is the following:



24 Grammars

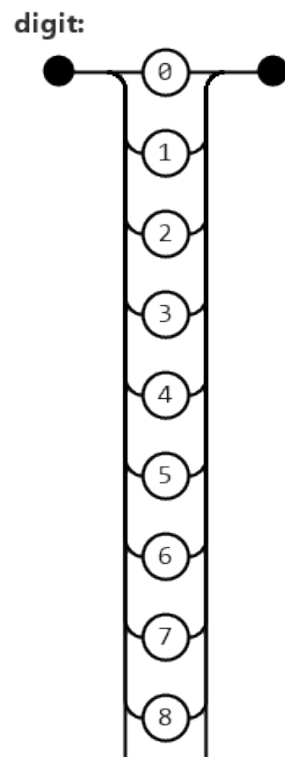
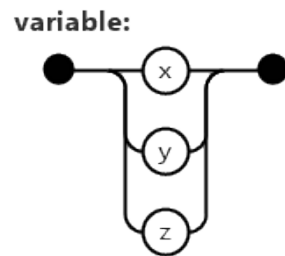
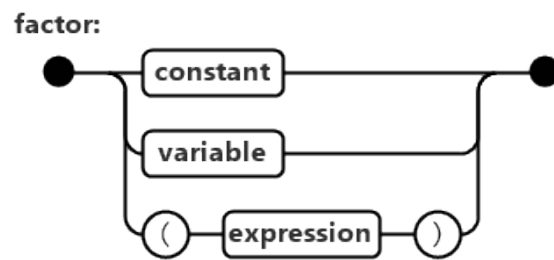
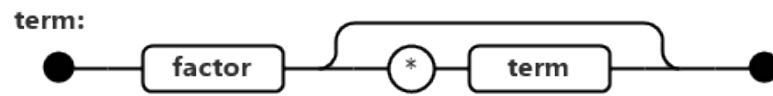
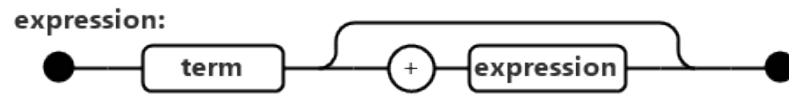
A grammar is a set of rules that can be used to generate strings. The strings that are generated by the grammar are called the language of the grammar. The language of the grammar is the set of all strings that can be generated by the grammar. The language of the grammar is denoted by $L(G)$.

24.1 Railroad Diagrams

Vocabulary This is a set of symbols which we use to represent the language of the grammar. The vocabulary of the grammar is denoted by V .

Grammar This is a set of rules that can be used to generate strings. The grammar is denoted by G . It is a combination of the vocabulary. By using grammar and vocabulary, we can generate strings which are part of **formal language**.

An example of a railroad diagram is the following:



We write the railroad diagram as follows (not the one above):

```
<expression> ::= <term> | <term> "+" <expression>
<term>       ::= <factor> | <factor> "*" <term>
<factor>     ::= <constant> | <variable> | "(" <expression> ")"
<variable>   ::= "x" | "y" | "z"
<constant>   ::= <digit> | <digit> <constant>
<digit>      ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

more coming soon