# Discrete Structures in Computer Science

Ethan Anthony

January 9, 2025

## Contents

# 1 Logical Expressions

A statement needs to be either true or false.

**Mathematical Proposition** **1.1**

A statement that is either true or false, but **not both**. Some texts call it a mathematical proposition.

Some examples of Mathematical Propositions are:

- All primes are odd (false)
- For all real numbers $x$, $x^2 > 0$
- Every even integer greater than 2 can be expressed as the sum of two primes.

Some non-examples are:

- Math is fun
- Go directly to jail, do not pass go, do not collect $200
- This sentence is false

**Propositional Function** **1.2**

Also called an open sentence, is a sentence with at least one free variable that becomes a statement once the variable is substituted with a value.

Some examples are:

- The truth value of $P(x)$ depends on $x$.
- $P(0)$ is false.
- $P(5)$ is true.

These all aren't true *or* false, unless we substitute the variables with a specific value. For example, $x > 0$ isn't true or false, but $x > 0; x = 1$ is false.

## 1.1 Logical Operators

**Logical Operator** **1.3**

A function that modifies one or more mathematical propositions.

The operator **not** negates a proposition $p$. It is written as $\neg p$.

| $p$ | $\neg p$ |
|---|---|
| F | T |
| T | F |

Figure 1: Not Operator

When we have two statements $p$ and $q$, we can modify and combine them in different ways. The operator **and**, written as $\wedge p$ combines them in such a way that the result is true *only if* both $p$ and $q$ are true individually.

The operator **or** operates such that the result is true if *either* $p$ or $q$ is true individually. If $p$ and $q$ are both true individually, the or operator still recognizes that as truth.

| $p$ | $q$ | $p \wedge q$ |
|---|---|---|
| F | F | F |
| T | F | F |
| F | T | F |
| T | T | T |

(a) And

| $p$ | $q$ | $p \vee q$ |
|---|---|---|
| F | F | F |
| T | F | T |
| F | T | T |
| T | T | T |

(b) Or

The **Exclusive Or** operator is similar to the or operator in that it is true in the case of one of $p$ or $q$ being true, but limits it to only a single one being true. If both are true, then the result is false.

| $p$ | $q$ | $p \oplus q$ |
|---|---|---|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | F |

(a) xor

| $p$ | $q$ | $p \Rightarrow q$ |
|---|---|---|
| F | F | T |
| F | T | T |
| T | F | F |
| T | T | T |

(b) If $\Rightarrow$ Then

## 1.2 Conditionals

**Consitional Statement** **1.4**

Contains a **hypothesis** (or premise), and results in a **conclusion** (or consequence). This statement takes the hypothesis, and determines the state of the conclusion based on the hypothesis.

An **If, then** statement is an example of a conditional statement. The truth table of the statement is in Figure 3b.

Given the conditional statement $p \Rightarrow q$, its **converse** is the statement $q \Rightarrow p$. A statement and its converse are not logiaclly equivalent. For example, "If it is raining, then it is cloudy." is not equivalent to its converse, "If it is cloudy, then it is raining."

Given a statement $p \Rightarrow q$, its **contrapositive** is the statement $\neg p \Rightarrow \neg q$. These statements *are* logically equivalent.

| $p$ | $q$ | $p \Rightarrow q$ | $q \Rightarrow p$ |
|---|---|---|---|
| F | F | T | T |
| F | T | T | F |
| T | F | F | T |
| T | T | T | T |

(a) Converse

| $p$ | $q$ | $p \Rightarrow q$ | $\neg p$ | $\neg q$ | $\neg p \Rightarrow \neg q$ |
|---|---|---|---|---|---|
| F | F | T | T | T | T |
| F | T | T | F | T | T |
| T | F | F | T | F | F |
| T | T | T | T | F | T |

(b) Contrapositive

Figure 4: Converse and Contrapositive

The statement $p \Leftrightarrow q$ (if and only if), results in truth *only if* both statements are the same, whether it be both false or both true.

| $p$ | $q$ | $p \Leftrightarrow q$ |
|---|---|---|
| F | F | T |
| F | T | F |
| T | F | F |
| T | T | T |

Figure 5: If and Only If

## 1.3 Vocabulary

**Tautology** **1.5**

A compound proposition that is always true for all truth values of the propositional variables it contains.

| $p$ | $\neg p$ | $p \vee \neg p$ |
|---|---|---|
| F | F | T |
| F | T | T |
| T | F | T |
| T | T | T |

Figure 6: Tautology

**Contradiction** **1.6**

A compound proposition that is always false.

| $p$ | $\neg p$ | $p \wedge \neg p$ |
|---|---|---|
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | F |

Figure 7: Contradiction

**Contingency** **1.7**

A compound proposition that is **neither** a tautology or a contradiction.

| $p$ | $q$ | $p \wedge q$ |
|---|---|---|
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | T |

Figure 8: Contingency

**Satisfiable** **1.8**

A compound proposition is considered **satisfiable** if there is at least one assignment of truth values to its variables that makes it true.

## 1.4 Logical Equivalences

Figure 9 shows one of **DeMorgan's Laws**, proving that the proposition: $\neg(p \wedge q)$ is logically equivalent to $\neg p \vee \neg q$.

| $p$ | $q$ | $\neg p$ | $\neg q$ | $p \wedge q$ | $\neg(p \wedge q)$ | $\neg p \vee \neg q$ |
|---|---|---|---|---|---|---|
| F | F | T | T | F | T | T |
| F | T | T | F | F | T | T |
| T | F | F | T | F | T | T |
| T | T | F | F | T | F | F |

Figure 9: DeMorgan's Laws

Apart from DeMorgan's Law, there are several other logical equivalences in Figure 10. These can be used to manipulate logical expressions algebraically without breaking the rules of logic.

| **Mathematical Representation** | **Name of the Law** |
|---|---|
| $p \vee \neg q \equiv T$ | Law of Excluded Middle |
| $p \wedge \neg p \equiv F$ | Law of Non-Contradiction |
| $p \wedge T \equiv p$ | Identity Laws |
| $p \vee F \equiv p$ | |
| $p \wedge F \equiv F$ | Domination Laws |
| $p \vee T \equiv T$ | |
| $p \vee p \equiv p$ | Idempotent Laws |
| $p \wedge p \equiv p$ | |
| $\neg\neg p \equiv p$ | Laws of Double Negation Elimination |
| $p \vee q \equiv q \vee q$ | Commutative Laws |
| $p \wedge q \equiv q \wedge q$ | |
| $(p \vee q) \vee r \equiv p \vee (q \vee r)$ | Associative Laws |
| $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$ | |
| $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ | Distributive Laws |
| $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ | |
| $\neg(p \wedge q) \equiv \neg p \vee \neg q$ | DeMorgan's Laws |
| $\neg(p \vee q) \equiv \neg p \wedge \neg q$ | |
| $p \vee (p \wedge q) \equiv p$ | Absorption Laws |
| $p \wedge (p \vee q) \equiv p$ | |
| $p \Rightarrow q \equiv q \vee \neg p$ | $\vee$ Restatement of Implication |
| $p \Rightarrow q \equiv \neg q \Rightarrow \neg p$ | Contraposition |
| $(p \Rightarrow q) \wedge (p \Rightarrow r) \equiv p \Rightarrow (q \wedge r)$ | Conjunction of Implications |
| $(p \Rightarrow r) \wedge (q \Rightarrow r) \equiv (p \vee q) \Rightarrow r$ | Will Show This Below |
| $(p \Rightarrow q) \vee (p \Rightarrow r) \equiv p \Rightarrow q \vee r$ | Disjunction of Implications |
| $(p \Rightarrow r) \vee (q \Rightarrow r) \equiv (p \wedge q) \Rightarrow r$ | |
| $p \Leftrightarrow q \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$ | Conjunction of Implications |
| $p \Leftrightarrow q \equiv \neg p \Leftrightarrow \neg q$ | Negation Restatement |
| $\neg(p \Leftrightarrow q) \equiv \neg p \Leftrightarrow q$ | Negation |

Figure 10: List of Logical Equivalences

Figures 11 and 12 are both examples of tautologies. By breaking them down into smaller parts, their logical meaning can be better understood.

| $p$ | $q$ | $p \Rightarrow q$ | $p \wedge (p \Rightarrow q)$ | $(p \wedge (p \Rightarrow q)) \Rightarrow q$ |
|---|---|---|---|---|
| F | F | T | F | T |
| F | T | T | F | T |
| T | F | F | F | T |
| T | T | T | T | T |

Figure 11: Tautology Example One

| $r$ | $v$ | $r \vee v$ | $\neg r$ | $(r \vee v) \wedge \neg r$ | $((r \vee v) \wedge \neg r) \Rightarrow v$ |
|---|---|---|---|---|---|
| F | F | F | T | F | T |
| F | T | T | T | F | T |
| T | F | Y | F | F | T |
| T | T | T | F | T | T |

Figure 12: Tautology Example Two

**Example** **1.1**

By using the logical equivalences in Figure 10, you can decompose a proposition into simpler terms to more clearly see what characteristics it has. For example, decomposing $(p \wedge p \Rightarrow q) \Rightarrow q \equiv T$ shows that it is a tautology.

$$\begin{aligned}
(p \wedge p \Rightarrow q) \Rightarrow q &\equiv T \\
[p \wedge (q \vee \neg p)] \Rightarrow q &\equiv T \\
[(p \wedge q) \vee (p \wedge \neg p)] \Rightarrow q &\equiv T \\
[(p \wedge q) \vee (F)] \Rightarrow q &\equiv T \\
(p \wedge q) \Rightarrow q &\equiv T \\
q \vee \neg(p \wedge q) &\equiv T \\
q \vee \neg p \vee \neg q &\equiv T \\
q \vee \neg q \vee \neg p &\equiv T \\
T \vee \neg p &\equiv T \\
T &\equiv T
\end{aligned}$$

**Syllogism** **1.9**

$$
\begin{aligned}
[(p \Rightarrow q) \wedge (q \Rightarrow r)] &\Rightarrow (p \Rightarrow r) \\
(q \Rightarrow \neg p) \wedge (r \Rightarrow \neg q) &\Rightarrow (p \Rightarrow r) \\
[q \wedge (r \vee \neg q)] \vee [\neg p \wedge (r \vee \neg q)] &\Rightarrow (p \Rightarrow r) \\
[(q \wedge r) \vee (q \wedge \neg q)] \vee [(\neg p \wedge r) \vee (\neg p \wedge \neg q)] &\Rightarrow (p \Rightarrow r) \\
[(q \wedge r) \vee (\neg p \wedge r) \vee (\neg p \wedge \neg q)] &\Rightarrow (p \Rightarrow r) \\
[\neg(q \wedge r) \wedge \neg(\neg p \wedge r) \wedge \neg(\neg p \wedge \neg q)] &\vee (p \Rightarrow r) \\
[(\neg q \vee \neg r) \wedge (p \vee \neg r) \wedge (p \vee q)] &\vee (p \Rightarrow r)
\end{aligned}
$$

A syllogism is a form in which, if $p$ implies $q$, and $q$ implies $r$, then $p$ implies $r$. It can be decomposed as seen above to show that the form of $[(p \Rightarrow q) \wedge (q \Rightarrow r)] \Rightarrow (p \Rightarrow r)$ is a tautology.

**Modus Tollens** **1.10**

$$
\begin{aligned}
(\neg \wedge p \Rightarrow q) &\Rightarrow \neg p \\
(\neg q \wedge (q \vee \neg p)) &\Rightarrow \neg p \\
[(\neg q \wedge q) \vee (\neg q \wedge \neg p)] &\Rightarrow \neg p \\
[F \vee (\neg q \wedge \neg p)] &\Rightarrow \neg p \\
(\neg q \wedge \neg p) &\Rightarrow \neg p
\end{aligned}
$$

## 1.5 Quantifiers

In the context of propositional functions, certain restrictions can be applied to what can be considered as a value to use in the function. These restrictions are determined by quantifiers.

A propositional function can be quantified with a **universal quantifier**. For example:

"All integers are either positive or negative"

This is equivalent to:

$$\forall n \in \mathbb{Z}, (n \geq 0) \vee (n < 0)$$

---

A propositional function can also be quantified with a **existential quantifier**. For example:

"There is an integer $n$ that is even *and* prime"

This is equivalent to:

$$\exists n \in \mathbb{Z}, (n \text{ is even}) \wedge (n \text{ is prime})$$

## 1.6 Canonical Normal Form

Canonical Normal Forms are standardized forms of boolean logic that serve some purpose to simplify the expression of boolean values.

**Disjunctive Normal Form** **1.11**

A canonical logical form consisting of a disjunction (OR) of conjunctions (ANDs). More broadly, a logical formula is said to be in Disjunctive Normal Form (DNF) if it consists of one or more conjunctions or literals joined in disjunction.

All of the following formulas follow Disjunctive Normal Form.

$$(A \land \neg B \land \neg C) \lor (\neg D \land E \land F \land D \land F)$$
$$(A \land B) \lor (C)$$
$$(A \land B)$$
$$(A)$$

# 2 Sets

## 2.1 Introduction to Sets

**Set** **2.1**

A collection of objects defined either by listing its members or by defining a pattern.

The ways to specifically define certain elements as part of a set is as follows:

| Notation | Meaning |
|---|---|
| $5 \in A$ | 5 is a member of set $A$ |
| $3 \notin B$ | 3 is not a member of set $B$ |
| $C = \{1, 2, 3\}$ | Set $C$ contains 1, 2, and 3 |

Additionally, set builder notation allows you to construct a set based on a defined pattern:

$$A = \{2k + 1 | k \in \mathbb{Z}\}$$
$$A = \{n \mid \exists k \in \mathbb{Z} \text{ for which } n = 2k + 1\}$$

Figure 14: Defines Set $A$ as all odd numbers

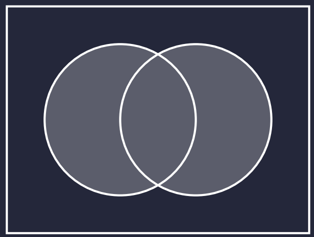
Sets can also be defined to be empty. $A = \{\}$, $A = \varnothing$, and $A = \{\varnothing\}$ all define the set $A$ to be empty. Additionally, this can be done through set builder notation:

$$A = \{x \mid \forall x, x \ \exists! \ U\}$$

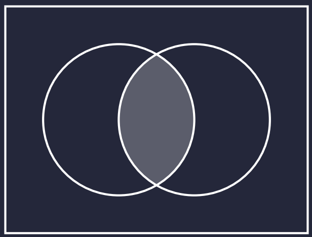
---

## 2.2 Set Operations

Once a set exists, operations can be performed on the set at large. Four basic operations are the union of two sets ($\cup$), the intersection of two sets ($\cap$), the complement of a set ($\complement$), and the difference of two sets ($\setminus$).



(a) $A \cup B$



(b) $A \cap B$

(a) $A^{\mathbb{C}}$ (b) $A \backslash B$

Figure 16: Set Operations

## 2.3 Subsets and Supersets

**Subset** **2.2**

**Superset** **2.3**

## 2.4 Powersets and Cartesian Products

**Powerset** **2.4**

**Cartesian Product** **2.5**

## 2.5 Set Identities

## 2.6 Set Notation

| Notation | Meaning |
|---|---|
| $\varnothing$ | Empty Set |
| $\#$ | Size of a Set |
| $\in$ | Is a member of |
| $\notin$ | Is not a member of |
| $\subset$ | Is a subset of |
| $\supset$ | Is a superset of |
| $\supset$ | Is a superset of |
| $\cup$ | Set union |
| $\cap$ | Set intersection |
| $\setminus$ | Set difference |
| $\complement$ | Set complement |

(a) Set Theory

| Notation | Meaning |
|---|---|
| $\neg$ | Logical Negation |
| $\wedge$ | Logical And |
| $\vee$ | Logical Or |
| $\oplus$ | Logical Xor |
| $\forall$ | Universal Quantification |
| $\exists$ | Existential Quantification |
| $\exists!$ | Uniqueness Quantification |
| $\Rightarrow$ | Material Conditional |
| $\Leftrightarrow$ | Logical Equivalence |
| $\bot$ | False Predicate |
| $\top$ | True Predicate |

(b) Formal Logic

| Notation | Meaning |
|---|---|
| $\mathbb{N}$ | Natural Numbers |
| $\mathbb{Z}$ | Integers |
| $\mathbb{Q}$ | Rational Numbers |
| $\mathbb{R}$ | Real Numbers |
| $\mathbb{C}$ | Complex Numbers |

(c) Number Systems

Figure 17: Logic and Set Theory Notation

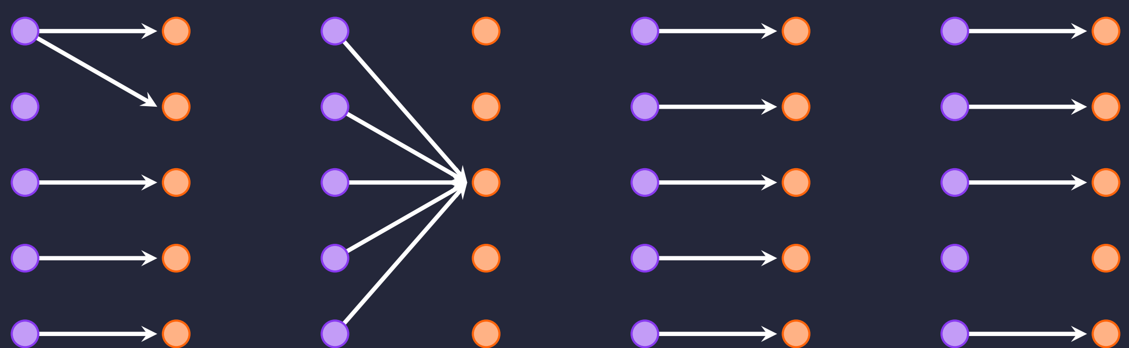
# 3 Functions

## 3.1 Function Definition

A function is defined as $f : A \to B$, where each element in array $A$ is assigned to exactly one element in array $B$. $f(a) = b$ denotes a single element from set $B$ as the assignment from an element from set $A$.

A function $f : A \to B$ can also be defined as a subset of $A \times B$. This is because a function $f : A \to B$ contains one, and only one, ordered pair for each element in $A$.

$$\forall x[x \in A \to \exists y[y \in B \wedge (x, y) \in f]]$$



(a) Not a Function (b) Function (c) Function (d) Not a Function

In the working definition of a function as $f : A \to B$, the **domain** of the function is $A$, and the **codomain** is $B$. In terms of $f(a) = b$, $a$ is the **preimage** and $b$ is the **image**.

**Range of a Function** **3.1**

The range of a function is the set of all images of the function.

$$\text{Range of a Set} = \{x | \forall a \in A, f(a) = x \wedge x \in B\}$$

## 3.2 Properties of Functions

**Injection (One to one)** **3.2**

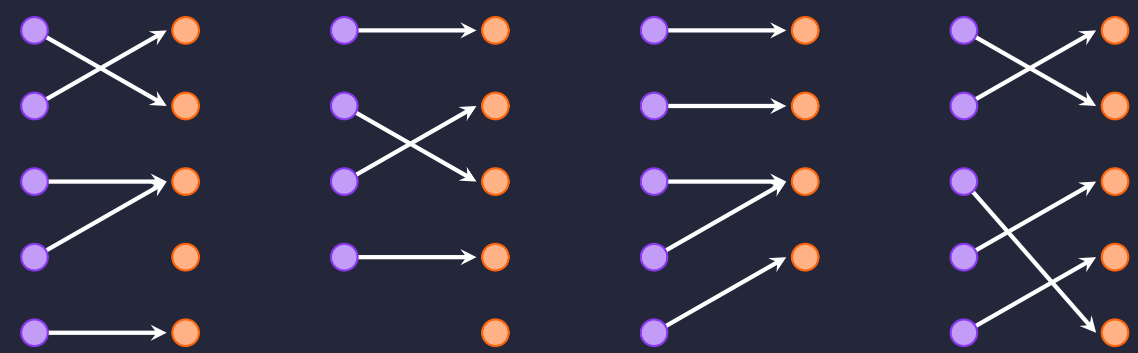
A function is said to be *injective* if and only if $f(a) = f(b) \to a = b$. An injective function has only one input for each element in its range.

**Surjection (Onto)** **3.3**

A function is said to be *surjective* if and only if for every element $b$ in $B$, there is an element $a$ in $A$ such that $f(a) = b$. In other words, the range of the function must be equivalent to the codomain.

**Bijection** **3.4**

A function is said to be bijective if and only if it is both *injective* and *surjective*.



(a) General Function (b) Injective (c) Surjective (d) Bijective

Figure 19: Examples of Functions

Functions can also be inversed. If there is a function $f$, the inverse is denoted as $f^{-1}$. A function can only have an inverse if it is a **bijective** function. This is because, if the function were to be injective, but not surjective, there would be elements in the domain of $f^{-1}$ that don't map onto the codomain of $f^{-1}$. Similarly, if the function were to be surjective, but not injective, elements in the domain of $f^{-1}$ would map onto multiple values.

# 4 Sequences

## 4.1 Sequence Definition

**Sequence** **4.1**

A sequence is an **ordered** list of elements. The elements in a sequence are derived from a subset of integers, but aren't necessarily integers themselves.

Some sequences, such as the Fibonacci Sequence, are defined recursively. That is each term in the sequence is defined in terms of some previous terms. The Fibonacci Sequence, for example, defines each term as the sum of the two previous terms. These recursively defined sequences are said to have **Recursive Relations** between their terms.

## 4.2 Types of Sequences

### Geometric Progression

A geometric progression is a sequence of the form:

$$ar^0, ar^1, ar^2, ar^3, ..., ar^n$$

where the ***initial term a*** and the ***common ratio r*** are both real numbers.

| a | r | Sequence |
|---|---|---|
| 1 | $-1$ | $\{1, -1, 1, -1, 1, -1, ...\}$ |
| 2 | 5 | $\{2, 10, 50, 250, 1250, ...\}$ |
| 6 | $\frac{1}{3}$ | $\{6, 2, \frac{2}{3}, \frac{2}{9}, \frac{2}{27}, \frac{2}{81}, \frac{2}{243}, ...\}$ |

### Artithmetic Progression

| a | d | Sequence |
|---|---|---|
| 1 | $-1$ | $\{1, 0, -1, -2, -3, ...\}$ |
| 2 | 5 | $\{2, 7, 12, 17, 22, 27, ...\}$ |
| 6 | $\frac{1}{3}$ | $\{6, \frac{19}{3}, \frac{20}{3}, \frac{21}{3}, \frac{22}{3}, \frac{23}{3}, ...\}$ |

A arithmetic progression is a sequence of the form:

$$a + 0d, a + 1d, a + 2d, a + 3d, ..., a + nd$$

where the ***initial term a*** and the ***common difference d*** are both real numbers.

### Fibonacci Sequence

The Fibonacci Sequence defined the $n^{th}$ term as the sum of the $n - 1^{th}$ term and the $n - 2^{th}$ term. The first two terms cannot be defined recursively, and are usually defined as $F_0 = 0$ and $F_1 = 1$.

$$F_0, F_1, (F_0 + F_1), (F_1 + F_2), (F_2 + F_3), (F_3 + F_4), ...$$

The Fibonacci Sequence with $F_0 = 0$ and $F_1 = 1$ is as follows:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...$$

# 5 Number Theory

## 5.1 Last-Digit Algebra

| $i$ | $n \cdot i$ | $n\%10$ |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 7 | 7 |
| 2 | 14 | 4 |
| 3 | 21 | 1 |
| 4 | 28 | 8 |
| 5 | 35 | 5 |
| 6 | 42 | 2 |
| 7 | 49 | 9 |
| 8 | 56 | 6 |
| 9 | 63 | 3 |
| 10 | 70 | 0 |
| 11 | 77 | 7 |
| 12 | 84 | 4 |
| 13 | 91 | 1 |

Adding 7 to itself infinitely will result in an infinitely increasing sequence of integers. By focusing only on the last digit (the one's digit) of each item in the sequence, a pattern emerges: $0, 7, 4, 1, 8, 5, 2, 9, 6, 3, 0, 7, 4, 1, ....$

This sequence is showing that when 7 is added to a number ending in a 0, the sum will end in a 7; when 7 is added to a number ending in a 7, the sum will end in a 4; etc. This idea can be stated as:

$$(7 \cdot 1)\%10 = 7$$
$$(7 \cdot 2)\%10 = 4$$

Any number ending in a 7 can replace the seven in these equations. Thus:

$$\forall x \mid x\%10 = 7 \Rightarrow (x \cdot 1)\%10 = 7$$
$$\forall x \mid x\%10 = 7 \Rightarrow (x \cdot 2)\%10 = 4$$

By considering only the last digit of a number, rules can be derived about the result of some arbitrary algebraic operation.

## 5.2 Divisibility and Modular Arithmetic

**Divides** **5.1**

$a$ is said to *divide* $b$ if $\frac{b}{a} \in \mathbb{Z}$. In other words, $a$ divides $b$ if there exists some integer $c$ such that $b = c \cdot a$. Denoted as $a|b$.

There are three **basic properties of divisibility**. Considering all *integer* values of $a$, $b$, and $c$ where $a \neq 0$, then:

$$a|b \wedge a|c \Rightarrow a|(b + c)$$
$$a|b \Rightarrow a|bc$$
$$a|b \wedge b|c \Rightarrow a|c$$