Not so Discrete Math

Christian Rudder

August 2024

Contents

This page is left intentionally blank.

1 Sets

Opening Remarks: Before you start reading, skim through and look at each definition. Every and anything that's bolded or underlined, and all the funky symbols. Ask yourself "What does this mean?" Then move on without giving it thought. This will prime your brain and engage your curiosity. It becomes a game of discovery rather than completion.

1.1 Introduction to Sets

In discrete math we work with some group of 'things,' a thing or something we fancily call an **object**. A group or categorization of objects is called a set.

Definition 1.1: Set

Is a collection of objects.

For Example:

- S = The set of all students in a classroom.
- A = The set of all vowels in the English alphabet.
- $\mathbb{Z}$ = The set of all integers.

Objects in a **set** are called **elements**.

Definition 1.2: Element

An object that is a member of a given set.

To expand on the previous example:

- $S = \{s_1, s_2, s_3\}$, where s_1, s_2, s_3 are students, elements of the set.
- $A = \{a, e, i, o, u\}$, where a, e, i, o, u are elements.
- $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$, elements of integer set.

Curly braces denote a set, commas separate elements, and the '...' (ellipse) indicates an indefinite continuation, <u>used only when the pattern **is clear.**</u>

There is also notation to denote members of a set.

Definition 1.3: Membership

If x is an element of set A, $x \in A$. If x is not an element of set A, $x \notin A$.

For Example: Given $A = \{a, e, i, o, u\}$,
$a \in A$, "a is an element of A," and $b \notin A$, "b is not an element of A."

Order nor repetition matter:

- $A = \{1, 2, 3\} = \{3, 2, 1\} = \{1, 2, 3, 3, 3, 3, 3\}$.
- $B = \{a, b, c\} = \{a, b, c, a, b, c\}$.

Definition 1.4: Properties of a Set

- The order of elements do not matter.
- Duplicate elements are not counted.

A subset is a set contained within another set. If the set B is a subset of set A, then every element in B is also in A as shown in Figure 1:

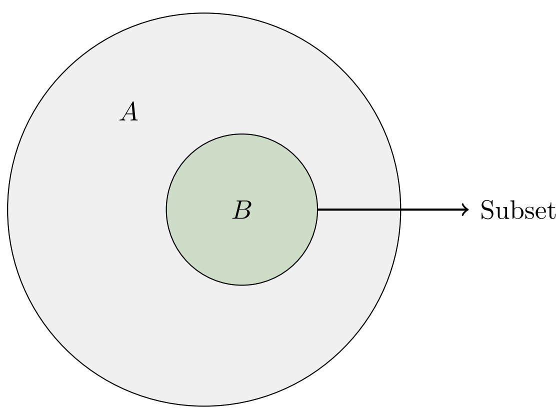

Figure 1: B is a subset of A.

Written $B \subseteq A$ or $A \supseteq B$, similar to the less than or equal to signs '$\leq$' and '$\geq$'.

Definition 1.5: Subset

If every element in set B is also in set A, then B is a subset of A.
Denoted: $B \subseteq A$ or $A \supseteq B$.

For Example:

- $\{-1, 0\} \subseteq \{-1, 0, 1, 2, 3\}$
- $\{-1, 1, 3\} \subseteq \{-1, 0, 1, 2, 3\}$
- $\{-1, 0, 1, 2, 3\} \subseteq \{-1, 0, 1, 2, 3\}$
- $\{-1, 7\} \not\subseteq \{-1, 0, 1, 2, 3\}$

$\not\subseteq$ denotes 'not a subset of.'

A set with no elements is called the empty set.

Definition 1.6: Empty Set

Commonly denoted by $\emptyset$ or $\{\}$, refers to a collection with no objects.

Questions:

1. How many elements are in the set $\{\emptyset\}$?
2. True or False: $\emptyset \subseteq \{\emptyset\}$.
3. True or False: $\emptyset \in \{\emptyset\}$.
4. True or False: $\emptyset \subseteq \emptyset$.
5. True or False: $\emptyset \subseteq \mathbb{Z}$.
6. True or False: $\emptyset \in \mathbb{Z}$.

Tip: Mathematicians define things? So can you! Let's define a collection that infinitely repeats the string "bees." We will fancily call it "**Bioths Non-determinant Sequence**," or a β_{seq} for short.

$$\beta_{seq} = \{\text{"bees"}, \text{"bees"}, \text{"bees"}, \text{"bees"}, \text{"bees"}, \text{"bees"}, \ldots\}$$

Names are names, no matter how fancy, they were labeled by another human, like you. They thought,... "Damn, this would be a *kick-ass* name." Never be intimidated, complex ideas are just groupings of basic concepts.

Answers:

1. 1 element, the empty set.
2. True, the empty set is a subset of $\{\emptyset\}$.
3. True, the empty set is an element of $\{\emptyset\}$.
4. True, the empty set is a subset of itself.
5. True, the empty set is a subset of all sets.
6. False, the empty set is not an element of the integers.

Why (1.): A collection is an object. The emptyset is a collection, a collection without objects. Likewise, a house is still a house without furniture.

Why (5.): Take sets $A = \{\}$ and $B = \{1, 2, 3\}$
By definition of a subset, every element in A must be in B. It's difficult to argue elements in A are indeed in B, but it's undeniable that elements in A are not in B. Since our statement cannot be denied, it's **Vacuously true**.

Say we have an empty box. How many objects do we have? **Zero**.
Put an empty box inside our original box. How many objects now? **One**!

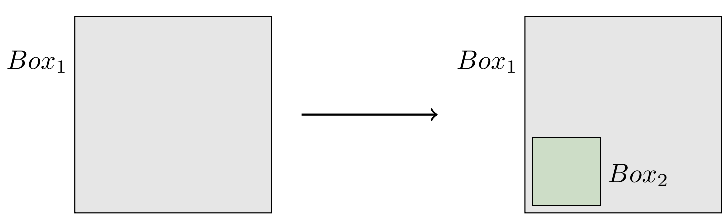

Figure 2: Box_1 contains 1 object, which is Box_2, an empty box. Hence Box_1 represents $\{\{\}\}$ or $\{\emptyset\}$.

Counting the number of elements in a set is called the **cardinality** of the set.

Definition 1.7: Cardinality

The number of elements in a set.
Denoted over a set A as $|A|$.

For Example:

- $A = \{1, 2, 3\}$, $|A| = 3$.
- $B = \{a, e, i, o, u\}$, $|B| = 5$.
- $\mathbb{Z}$ the set of all integers, $|\mathbb{Z}| = \infty$.

Questions:
What are the cardinalities of the following sets?

1. $|\{1, 2, 3\}|$
2. $|\emptyset|$
3. $|\{\}|$
4. $|\{\emptyset\}|$
5. $|\{1, \{2, 3\}\}|$
6. $|\{1, 2, 2, 3, 3, 3\}|$

Try to think about the answer before looking at the solution.
Things stick when you struggle.

Tip: Whenever you approach a problem, always break things down into simple components. "What defines a set? What defines an element? What defines a subset? What defines cardinality?"

Answers:

1. 3
2. 0
3. 0
4. 1
5. 2
6. 3

Explicitly defining a set, say $\{1, 2, 3, ...\}$, is called **set-roster notation**. **set-builder notation** enables us to create more complex definitions.

Definition 1.8: Set-Builder Notation

General form: $\{x \mid P(x)\}$,

- x = defines some variable.
- " | " = is short hand for "such that."
- $P(x)$ = describes the properties x must satisfy.

For Example: Lets Define the set of even integers

- $\{x \mid x \text{ is an even integer}\}$: "$x$, such that, x is an even integer."
- $\{x \in \mathbb{Z} \mid x \text{ is even}\}$: "$x$ in Integers, such that, x is an even."
- $\{x \in \mathbb{Z} \mid x \text{ is not odd}\}$: "$x$ in Integers, such that, x is not odd."

It's important to Define exactly what variables are. In the above, x was stated directly as an integer. If not, x could be **water-balloons** or **puppies**.

1.2 Set Operations

Combining the two sets, $\{1, 2, 3\}$ and $\{a, b, c\}$, produce the set $\{1, 2, 3, a, b, c\}$, which is called the **union**.

Definition 1.9: Union

The set of elements that appear in either set A or set B is the union.
Denoted: $A \cup B$.

This is also known as a **disjunction**, which is a fancy term for the word "OR".

For Example:

- $\{1, 2\} \cup \{2, 3\} = \{1, 2, 3\}$.
- $\{1, 2\} \cup \emptyset = \{1, 2\}$. There is nothing to add.
- $\{1\} \cup \{\emptyset\} = \{1, \emptyset\}$. The $\emptyset$ is an element in this case.

The common elements of the two sets, $\{1, 2, 3\}$ and $\{2, 3, 4\}$, produce the set $\{2, 3\}$, the **intersection**.

Definition 1.10: Intersection

The set of elements that appear in both sets A and B is the intersection. Denoted: $A \cap B$.

This is also known as a **conjunction**, which is a fancy term for the word "AND".

For Example:

- $\{1, 2\} \cap \{2, 3\} = \{2\}$.
- $\{1\} \cap \{2\} = \emptyset$. There is nothing in common.
- $\{1\} \cap \emptyset = \emptyset$. There is nothing to compare.

Tip: To lessen the confusion between $\cup$ and $\cap$, think, "$\cap$" for "AND", since $\cap$ looks like a curved "A" without the line.

The combination of $A = \{1, 2, 3\}$ and $B = \{a, b, c\}$ in order pairs are:

$$(1, a), (1, b), (1, c),$$

$$(2, a), (2, b), (2, c),$$

$$(3, a), (3, b), (3, c)$$

Putting the above objects in a set yields the **cartesian product** of A and B.

Definition 1.11: Cartesian Product

The set of all possible order pairs of elements from sets A and B. Denoted: $A \times B$.

For Example:

- $\{1, 2\} \times \{a, b\} = \{(1, a), (1, b), (2, a), (2, b)\}$.
- $\{1, 2\} \times \emptyset = \emptyset$. There is nothing to pair.
- $\{1\} \times \{\emptyset\} = \{(1, \emptyset)\}$. The $\emptyset$ is an element in this case.

Note: Visit '**Figure 2**' in the previous section if $\emptyset$ causes confusion.

We have sets $A = \{1, 2, 3\}$ and $B = \{2, 3, 4\}$, to remove the common elements A has with B, i.e., take all in A that is not in B, yields the set $\{1\}$, the **difference**.

Definition 1.12: Difference

The set of all elements that are in set A but not in set B.
Denoted: $A - B$.

For Example:

- $\{1, 2\} - \{2, 3\} = \{1\}$.
- $\{1\} - \{1\} = \emptyset$.
- $\{1, 2\} - \emptyset = \{1, 2\}$. There is nothing to remove.

Summary: A **set** is a collection of 'things' or objects. An object that is a member of a set is an **element** $4 \in \mathbb{Z}$. In a set **order and repetition do not matter**. Sets can contain other sets, these subsections/slices/portions are called **subsets**. An **empty set** is denoted $\emptyset$ or $\{\}$.

Cardinality is the element count of a set, it does not count sets beyond the top layer. $\{1, 2, 3\}$ has a cardinality of 2, not 3. **set-roster** notation is the explicit listing of a set like $\{..., 1, 2, 3, ...\}$. **set-builder** notation is the description of a set $\{x \in \mathbb{Z} | x \textbf{ is even}\}$.

Combining two sets is called a **union**, the **intersection** is the common elements between two sets. The **cartesian product** of two sets is the set of all possible ordered pairs. The **difference** between sets A and B are elements in A that are not in B. There are many more operations that we could discuss, but we will just stick to these for now.

2 Functions

2.1 Introduction to Functions

To talk about functions, is to talk about **relationships**. Take the '$<$' sign, this is a relationship. $x < y$ means x relates to y, such that x is less than y.

Let $a = \{1, 2, 3, 4\}$, $b = \{0, 1, 2, 3\}$. Let R be the relation '$<$', aRb yields:

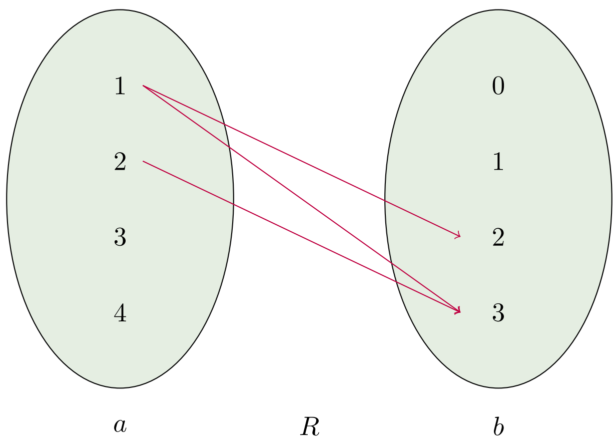

Figure 3: R produces ordered pairs $\{(1, 2), (1, 3), (2, 3)\}$, as $1 < 2$, $1 < 3$, $2 < 3$
.

Definition 2.1: Relation

A relation R on sets a and b, aRb, is a subset of $a \times b$.

Since $a \times b$ is the set of all possible ordered pairs. R's pairings must contain some or all pairing of $a \times b$. This includes no pairing at all, the emptyset.

The arrows in **Figure 5** are often referred to as **mappings**. 1 maps to 2, 1 maps to 3, and 2 maps to 3.

Tip: When looking at Definitions, come up with examples on your own. The ability to explain a concept to someone else proves understanding.

Functions are a type of relation, where **each input has exactly one output.**
We can visualize this as a machine:

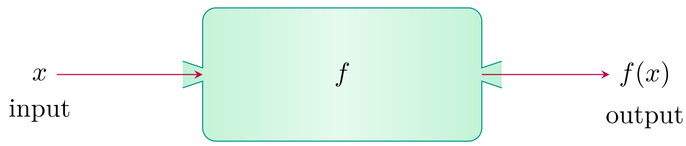

Figure 4: The machine f takes an input x and produces an output $f(x)$.

In our previous example we used '$<$' as a relation. This is not a function, as 1 relates to 2 and 3. Instead let's use the absolute function, $f(x) = |x|$.

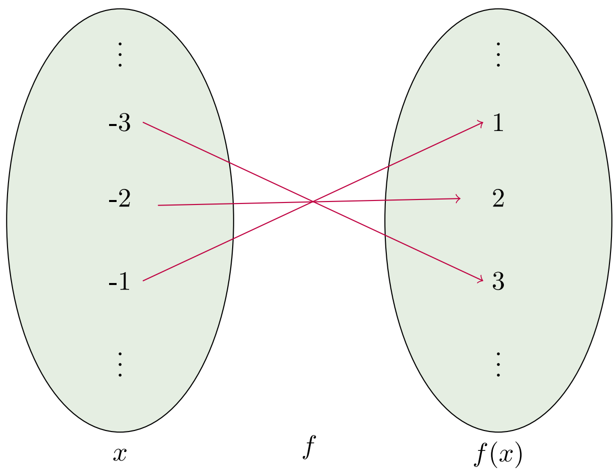

Figure 5: Pairs $\{(-3, 3), (-2, 2), (-1, 1)\}$, as $f(-3) = 3$, $f(-2) = 2$, $f(-1) = 1$
.

Note: f(x)=|x| is where x hasn't been chosen yet, choose 3, f(-3)=|-3|=3.

In **Figure 5**, assume inputs are integers. Then the absolute functions maps to integers, i.e., if I put in an integer, I get out an integer. In the figure we represent this by the two green ovals; The left (inputs), the right (outputs).

Our inputs are called the **domain**, the outputs the **codomain**,
and all the possible mappings the **range**.

Definition 2.2: Function

A function f is a relation between two sets, A (the domain) and B (the codomain), such that each element in A is mapped to exactly one element in B. The set of all possible outputs of f is called the range of f.

Say we create a function $f(x)$, which tells us if someone prefers cats or dogs.

$$\text{Let } P = \{\text{“Lois”}, \text{“Pete”}, \text{“Stuart”}\},\ Q = \{\text{”Cats”}, \text{”Dogs”}\}.$$

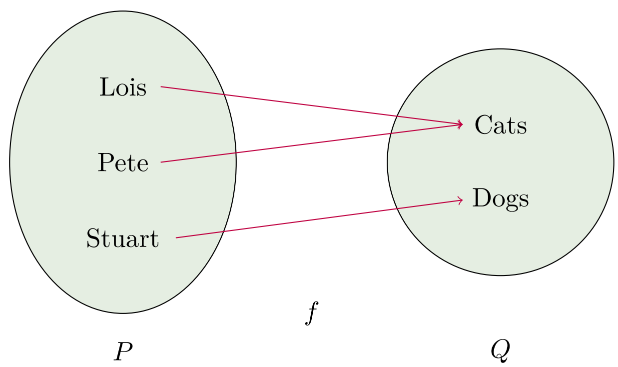

Figure 6: Produced ordered pairs $\{(\text{Lois}, \text{Cats}), (\text{Pete}, \text{Cats}), (\text{Stuart}, \text{Dogs})\}$.

In function f, all elements in our domain map to an element in our codomain. This makes our function **onto**, or **surjective**.

Definition 2.3: Onto (Surjective)

A function f is onto if every element in the codomain is mapped to by some element in the domain.

Now say we have a function $g(x)$, which tells us a person's student ID.

$$\text{Let } T = \{\text{“Raven”}, \text{“Stella”}, \text{“Robbert”}\},\ U = \{\text{”U001”}, \text{”U002”}, \text{U003}, ...\}.$$

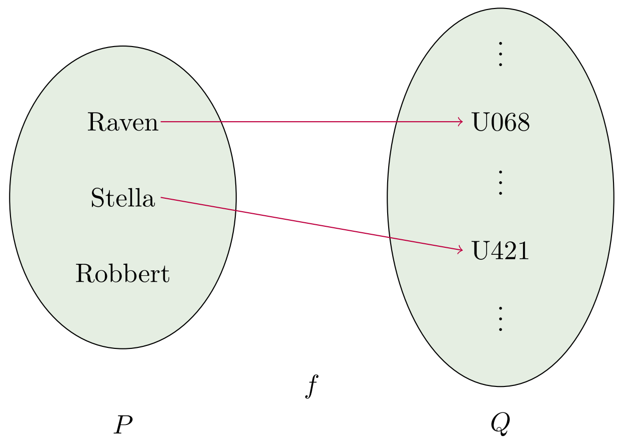

Figure 7: Produced ordered pairs $\{(\text{Raven}, \text{U068}), (\text{Stella}, \text{U421})\}$. not all elements have a mapping, meaning it is **not onto**.

In our example this suggests that “Robbert” is not a registered student.

Each element in the domain maps to exactly one element in the codomain. This makes our function **one-to-one**, or **injective**.

Definition 2.4: One-to-One (Injective)

A function f is one-to-one if each element in the domain is mapped to exactly one element in the codomain.

Tip: To remember the difference between ‘surjective’ and ‘injective’: think, one-to-one, one-to-one means a personal connection, something unique, one goes **into** one, **‘i’** for **‘into’**, **‘i’** for **‘injective’**.

If we were to change the function g, restricting the domain to only include registered students, then our function would be **onto** and **one-to-one**. Since each student has a unique ID, and each ID belongs to a student.

When a function is both onto and one-to-one, it is called a **bijection**.

Definition 2.5: Bijection

A function f is a bijection if it is both onto and one-to-one.

Summary: A **function** is a **relationship** between two sets. A set for which we use as an input, and a set that will house our outputs. A relationship can be described in ordered pairs, take sets A and B, over some relation R, $x \in A$ relates to $y \in B$ such that $(x, y) \in R$. The ordered pairs in a relation is a subset of $A \times B$, which includes the emptyset.

A function takes in one input and produces one output. The set of all our inputs is called the **domain**, the set of our outputs the **codomain**. The set of all possible mappings from the domain to the codomain is called the **range**.

If elements in the codomain all have a mapping, the function is **onto** or **surjective**. If elements in the codomain have a unique mapping, the function is **one-to-one** or **injective**. If a function is both **onto** and **one-to-one**, it is **bijective**.

3 Logic

3.1 Introduction to Logic

Observe the claim below:

"You're a rotten cook and your breath is sickly."

This **assertion** is a **statement** or **proposition** that is either **TRUE** or **FALSE**.

Note: claim, assertion, statement, and proposition are all synonyms.

Most Importantly:

- Am I actually a rotten cook?
- Is my breath really that sickly?

We can boil down these two propositions to the following:

- R: You're a rotten cook
- S: Your breath is sickly

The original claim can be rewritten as:

- "R and S" formally written, "$R \wedge S$", "$\wedge$" shorthand for "AND."
  Both proposition must be true for the claim to be true.

Altering the claim to *"You're a rotten cook or your breath is sickly"* yields:

- "R or S" formally written, "$R \vee S$", "$\vee$" shorthand for "OR."
  At least one of the proposition must me true for the claim to be true.

  "*Maybe you're a rotten cook, maybe your breath is sickly, maybe both.*"

Tip: To remember the difference between "$\wedge$" and "$\vee$", think: $\wedge$ looks like an "A" without the line for "AND".

Also remember "And" is CONJUNCTION, "OR" is DISJUNCTION. To help, think conjunction means "conjoin" to join together,
"I'm putting together one and one, I'm *conjoining* them."

Observe the claim below:

"I'm not a rotten cook, but I admit my breath is sickly."

The above states, "Not R" formally written, "$\neg R$", "$\neg$" shorthand for "NOT". "but" in this context is a conjunction. The claim writes as: "$\neg R \wedge S$".

Tip: Try to uncover what a statement is truly saying, not literally. In language we often obfuscate sentences to hide intent or to be more polite.

Observe the claim below:

"Either I'm a rotten cook or my breath is sickly, not both."

This is an example of the **exclusive OR** (XOR), we'll use the symbol "$\oplus$". Written as "$R \oplus S$", **only one** of the proposition can be true, not both.

This is more obvious in statements such as:

- "They either went to the party or stayed home."
- "You are either with me or against me."
- "They either bought lunch or saved their money."

We call "$\wedge, \vee, \neg, \oplus$", **logical operators**.

Definition 3.1: Logical Operators

A symbol that represents a logical operation.

- $\wedge$: AND, both must be true.
- $\vee$: OR, at least one must be true.
- $\neg$: NOT, negation, opposite.
- $\oplus$: XOR, exclusive OR, either or, but not both.

3.2 Statements vs. Predicates:

Below reads, **"4 is less than 2."** This is false, but still **is a statement**:

$$4 < 2$$

A statement evaluates to either true or false, **predicates don't**. Such as:

$$x < 2$$

This is a predicate, it's neither true or false until we assign a value to x.

Definition 3.2: Proposition

A statement that is either true or false, independent of any variables.

Definition 3.3: Predicate

A statement where its truth values depend on one or more variables.

3.3 Truth Tables

Writing whole sentences in logic can be cumbersome even in their reduced logical forms. So we use **truth tables** to help keep track our evaluations.

To demonstrate, let's use a table to show the values P and $\neg P$ Our goal is to find all possible values of P to then evaluate $\neg P$.

P	$\neg P$
T	F
F	T

T = "True" and F = "False", we refer to these truth values as **booleans**.

Definition 3.4: Boolean

A Boolean is a value that can only be either true or false.

Our goal here is to find all possible value combinations of P and Q to then evaluate $P \wedge Q$ and $P \vee Q$.

P	Q	$P \wedge Q$	$P \vee Q$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

We will read this table's P then Q boolean values respectively:

1. T then T, So $T \wedge T = T$, and $T \vee T = T$.
2. T then F, So $T \wedge F = F$, and $T \vee F = T$.
3. F then T, So $F \wedge T = F$, and $F \vee T = T$.
4. F then F, So $F \wedge F = F$, and $F \vee F = F$.

Tip: When working with new concepts, ask yourself what was going through the creator's mind. Why this way and not some other way? Try to understand the intuition behind an idea rather than pure memorization.

Now lets evaluate $\neg(P) \vee \neg(Q)$, the full table will be on the next page:

P	Q	$\neg P$	$\neg Q$	$\neg P \vee \neg Q$
T	T			
T	F			
F	T			
F	F			

Try to think and fill this out on your own, then compare your answers with the next page.

This is the complete table for $\neg(P) \vee \neg(Q)$:

P	Q	$\neg P$	$\neg Q$	$\neg P \vee \neg Q$
T	T	F	F	F
T	F	F	T	T
F	T	T	F	T
F	F	T	T	T

Now let's try to evaluate $P \wedge \neg Q \vee W$. The filled table will be on the next page.

P	Q	W	$\neg Q$	$P \wedge \neg Q$	$P \wedge \neg Q \vee W$
T	T	T			
T	T	F			
T	F	T			
T	F	F			
F	T	T			
F	T	F			
F	F	T			
F	F	F			

Notice a pattern is starting to begin with our variables, P, Q, and W. To get all possible combinations, we increase the period of repeating trues and false.

- Starting with W we start with T alternate each row F.
- For Q we alternate every 2 rows.
- For P we alternate every 4 rows.
- With 4 variables, we alternate every 16 rows, and 5 every 32, and so on.

The pattern is 2^n where n is the number of variables.

Definition 3.5: Growth of Truth Table

The number of rows in a table grow 2^n where n is the number of variables.

This is the complete table for $P \wedge \neg Q \vee W$:

P	Q	W	$\neg Q$	$P \wedge \neg Q$	$P \wedge \neg Q \vee W$
T	T	T	F	F	T
T	T	F	F	F	F
T	F	T	T	T	T
T	F	F	T	T	T
F	T	T	F	F	T
F	T	F	F	F	F
F	F	T	T	F	T
F	F	F	T	F	F

Truth tables are functions, given a set of inputs, it will produce an output. Our above table could be written as the function $f(P, Q, W) = P \wedge \neg Q \vee W$.

We can leverage tables to derive new functions. Observe the table below:

P	Q	$f(x)$
T	T	F
T	F	T
F	T	F
F	F	T

Note the highlighted rows, our function is true if:

"P is true and Q is false" or "P is false and Q is false."

Let's convert our statements with into propositional logic with our boolean operators:

- "P is true and Q is false" $= P \wedge \neg Q$.
- "P is false and Q is false." $= \neg P \wedge \neg Q$.
- Adding back our "or" yields: $(P \wedge \neg Q) \vee (\neg P \wedge \neg Q)$.

Note: P and Q are propositions, and propositions are claims that we assert to be true. To say "P is false," is to say "not P is true," i.e, "$\neg P$".

Known as **disjunctive normal form (DNF)**, where we take statements held by "AND" and join them by "OR." For reference: $(P \wedge \neg Q) \vee (\neg P \wedge \neg Q)$:

Definition 3.6: Disjunctive Normal Form (DNF)

Statements held by "AND" joined by "OR," i.e., conjunctions joined by disjunctions.

We can also derive a function by negating false rows, which also means true:

P	Q	$f(x)$
T	T	F
T	F	T
F	T	F
F	F	T

"not row one" and "not row three."
"$\neg(P \wedge Q)$" and "$\neg(\neg P \wedge Q)$".

To simplify, we will use **De Morgan's Laws**.

Definition 3.7: De Morgan's Laws

For booleans P and Q:
First Law: $\neg(P \wedge Q) = \neg P \vee \neg Q$.
Second Law: $\neg(P \vee Q) = \neg P \wedge \neg Q$.

i.e., negation distributes, $\vee$ to $\wedge$ and $\wedge$ to $\vee$.

And we'll need the **law of double negation**.

Definition 3.8: Double Negation

For a boolean P, $\neg(\neg P) = P$.

3.4 Functional completeness