ELEMENTARY COMBINATORICS & PROBABILITY

Lecture 1

Sets
Basic counting

TODAY

- Introduction
  - about this course;
  - about me.
- What is combinatorics (and why it's worth studying).
- Basic counting principles.
- Examples.

ABOUT ME

- EVGENIYA Korneva evgeniakorneva@gmail.com

ABOUT ME

- EVGENIYA Korneva evgeniakorneva@gmail.com
- Education:
  - *2015* - Bachelor of Applied Mathematics *(Moscow, Russia)*
  - *2016* - Master of Artificial Intelligence *(Leuven, Belgium)*

ABOUT ME

- EVGENIYA Korneva evgeniakorneva@gmail.com
- Education:
  - *2015* - Bachelor of Applied Mathematics *(Moscow, Russia)*
  - *2016* - Master of Artificial Intelligence *(Leuven, Belgium)*
- *2016* - … Doctoral researcher at KU Leuven

ABOUT ME

- EVGENIYA Korneva evgeniakorneva@gmail.com
- Education:
  - *2015* - Bachelor of Applied Mathematics *(Moscow, Russia)*
  - *2016* - Master of Artificial Intelligence *(Leuven, Belgium)*
- *2016* - … Doctoral researcher at KU Leuven

Leuven, Belgium

LOGISTICS

- Class: 09:00 – 12:20

LOGISTICS

- Class: 09:00 – 12:20

- Week 1: Combinatorics.
- Weeks 2 & 3: Probability.

LOGISTICS

- Class: 09:00 – 12:20

- Week 1: Combinatorics.
- Weeks 2 & 3: Probability.

- Evaluation:
  - 5 graded assignments
  - 2 interim exams (March 22 & 29)
  - Final exam (April 2)

LOGISTICS

- Class: 09:00 – 12:20

- Week 1: Combinatorics.
- Weeks 2 & 3: Probability.

- Evaluation:
  - 5 graded assignments *5% each*;
  - 2 interim exams (March 22 & 29) *25% each*;
  - Final exam (April 2), *25%*.

LET'S START!

WHAT IS COMBINATORICS

- Fine art of counting.

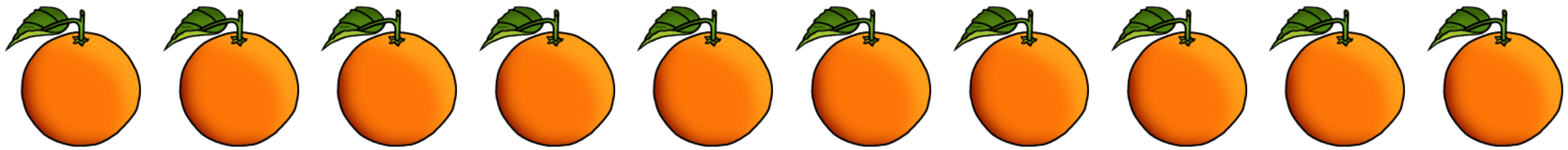
WHAT IS COMBINATORICS

WHAT IS COMBINATORICS

- We can easily count by enumeration:

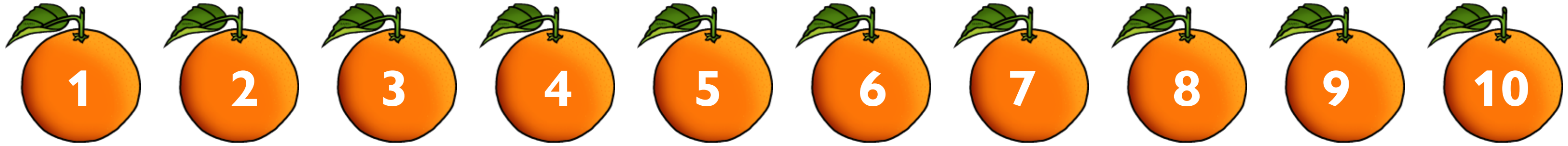
WHAT IS COMBINATORICS

- We can easily count by enumeration:



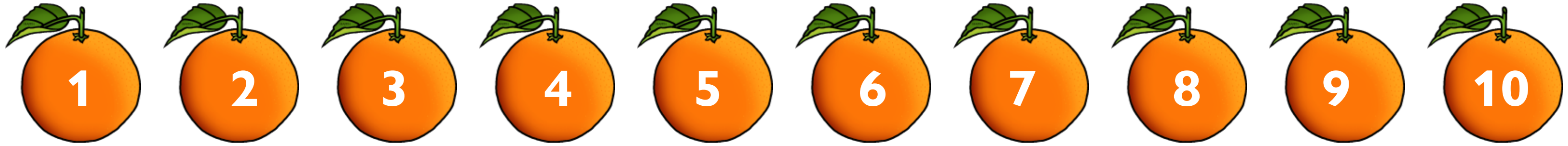
WHAT IS COMBINATORICS

- We can easily count by enumeration:



WHAT IS COMBINATORICS

- We can easily count by enumeration:

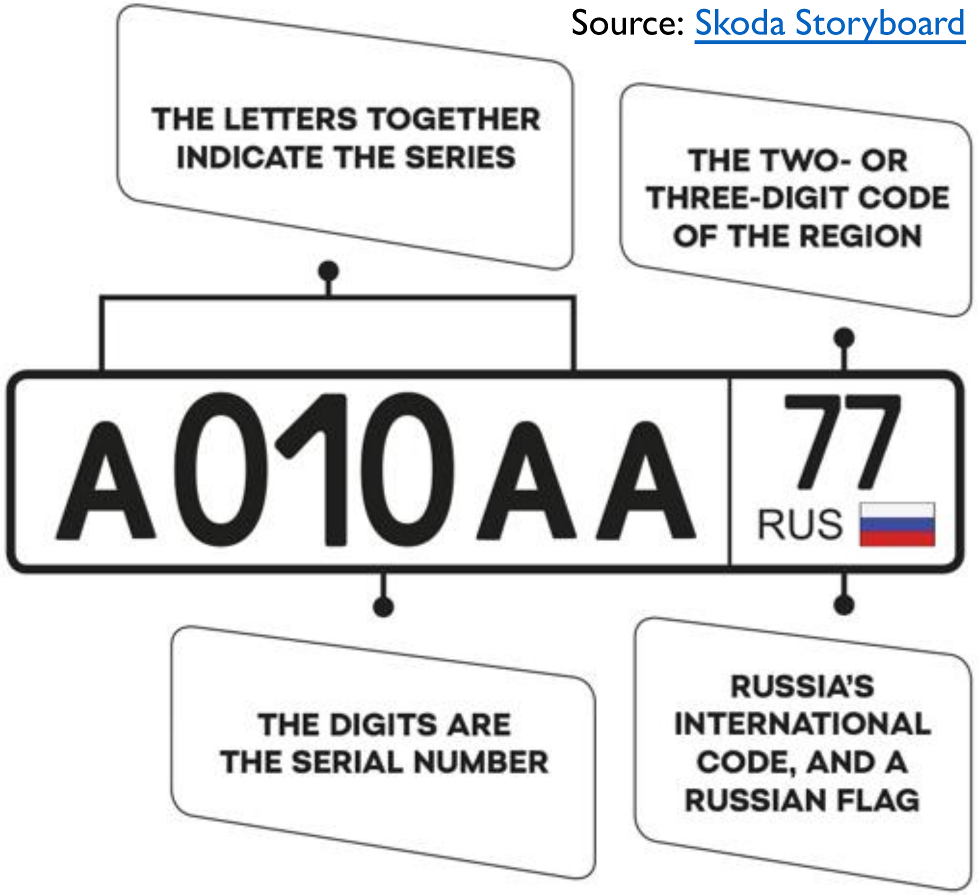


- But enumeration is not always easy.

MOTIVATING EXAMPLE: LICENSE PLATES

MOTIVATING EXAMPLE: LICENSE PLATES

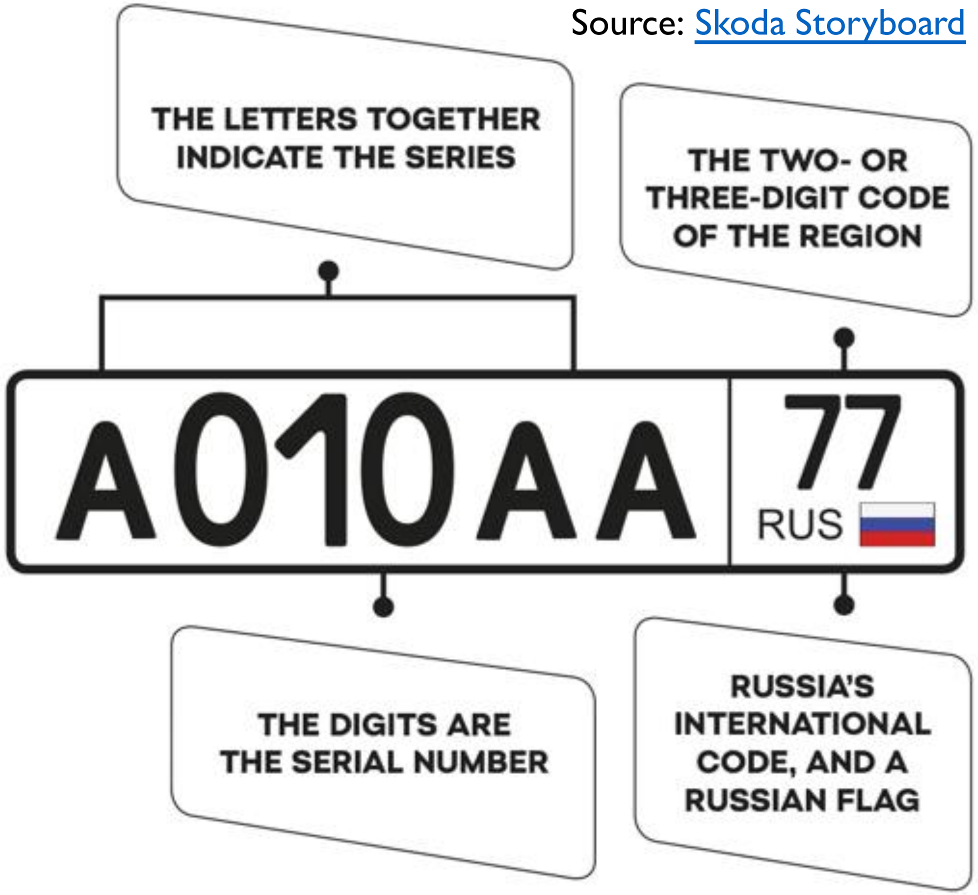
Source: [Skoda Storyboard]



- A Russian license plate:
  3 letters + 3 digits + regional code

MOTIVATING EXAMPLE: LICENSE PLATES

Source: [Skoda Storyboard]

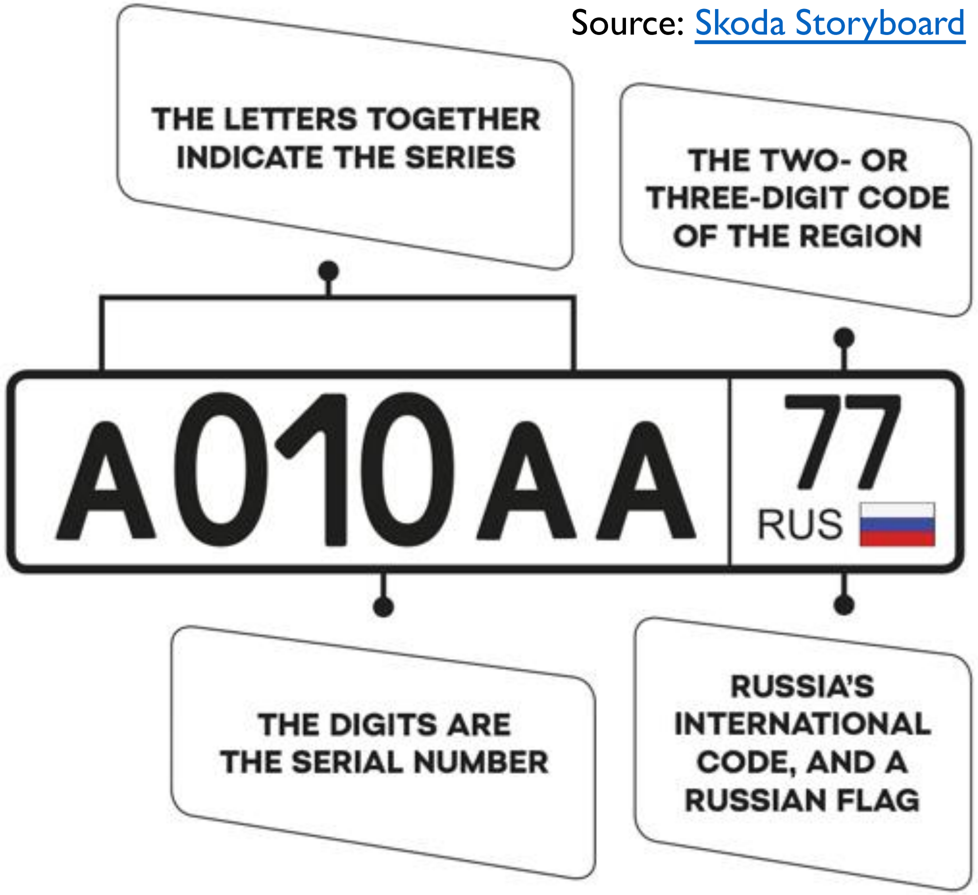


- A Russian license plate:

  3 letters + 3 digits + regional code
- 12 letters and 10 digits can be used.

MOTIVATING EXAMPLE: LICENSE PLATES

Source: [Skoda Storyboard]

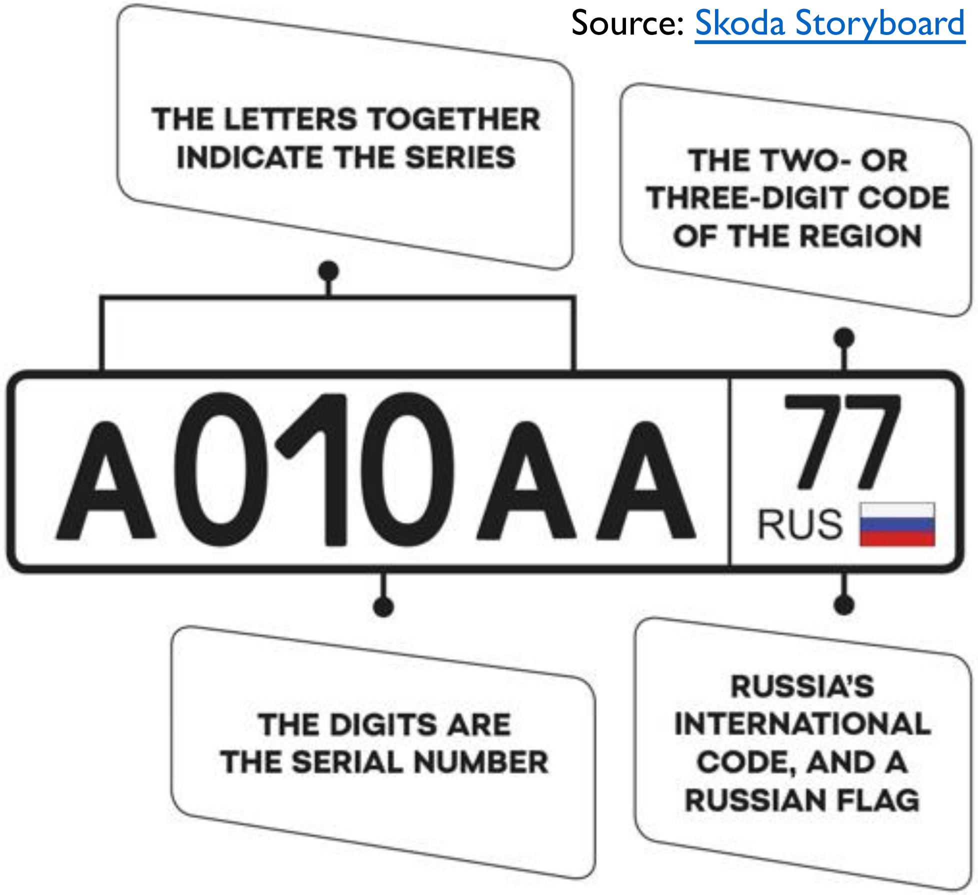


- A Russian license plate:

  3 letters + 3 digits + regional code
- 12 letters and 10 digits can be used.
- How many combinations are there per region?

MOTIVATING EXAMPLE: LICENSE PLATES

Source: Skoda Storyboard

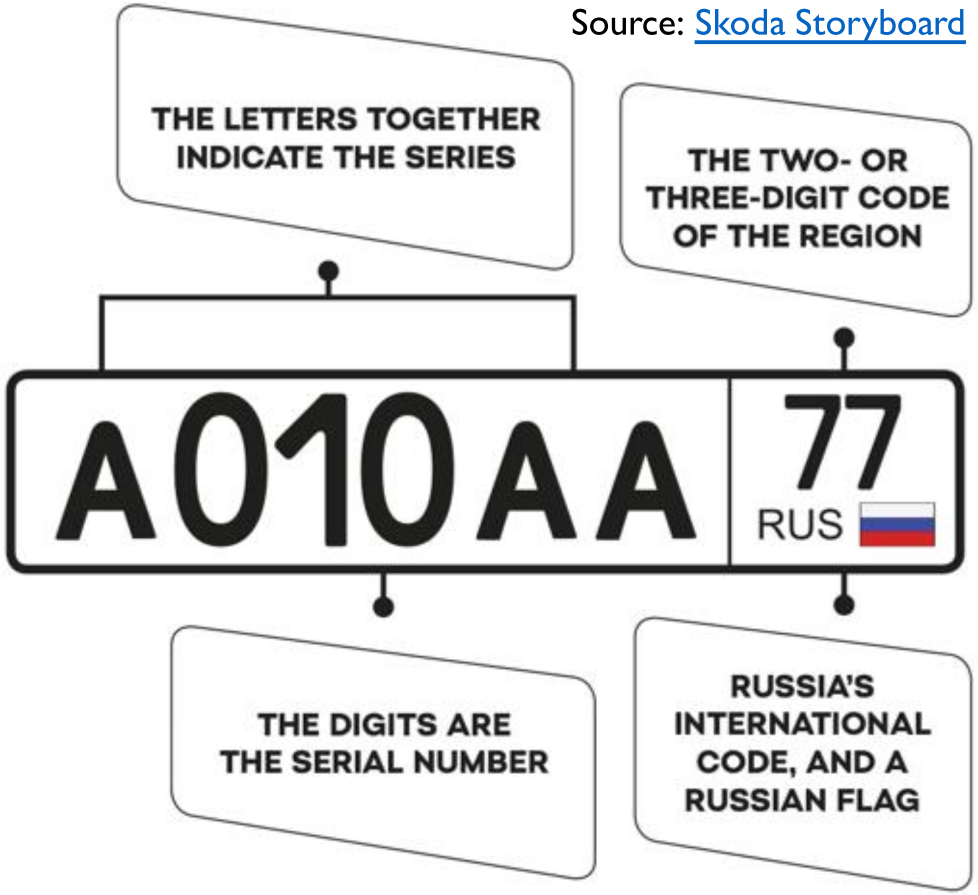


- A Russian license plate:

  3 letters + 3 digits + regional code
- 12 letters and 10 digits can be used.
- How many combinations are there per region?
  - ~ 1.7 million unique plates

MOTIVATING EXAMPLE: LICENSE PLATES

Source: Skoda Storyboard



- A Russian license plate:

  3 letters + 3 digits + regional code
- 12 letters and 10 digits can be used.
- How many combinations are there per region?
  - ~ 1.7 million unique plates
- Moscow has **nine** regional codes.
  - ~ 1.7 million -> ~15.5 million unique plates.

MOTIVATING EXAMPLE: PHONE NUMBERS

MOTIVATING EXAMPLE: PHONE NUMBERS

Japan is running out of phone numbers, so it's making longer ones

Its current 11-digit numbers could run out by 2022

By Jon Porter | @JonPorty | May 16, 2019, 8:45am EDT

SHARE

Photo by Amelia Holowaty Krales / The Verge

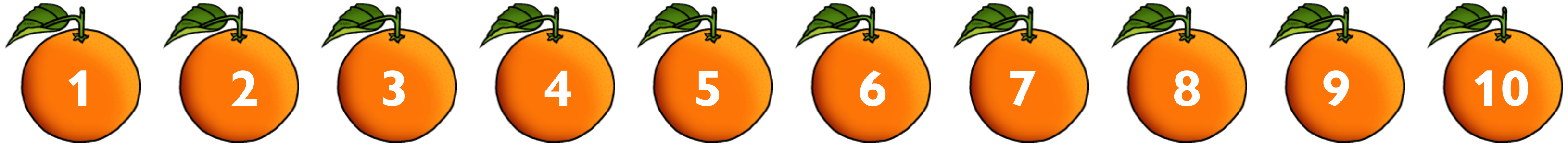
Source: [The Verge]

Source: [Independent]

Local calls using area codes seen as a better alternative to adding another digit

WHAT IS COMBINATORICS

- We can easily count by enumeration:



- But enumeration is not always easy.

Combinatorics = fine art of counting
(without *actually* enumerating).

COMBINATORICS: WHAT ELSE FOR?

COMBINATORICS: WHAT ELSE FOR?

- Computer science: determining the time and storage required to solve a problem.

COMBINATORICS: WHAT ELSE FOR?

- Computer science: determining the time and storage required to solve a problem.

- Basis of probability theory
  - *we'll see it later in the course.*

COMBINATORICS: WHAT ELSE FOR?

- Computer science: determining the time and storage required to solve a problem.

- Basis of probability theory
  - *we'll see it later in the course.*

- Some proof techniques in mathematics rely on combinatorics.

SET THEORY

SETS

- We almost always have a collection of things we are trying to study.

SETS

- We almost always have a collection of things we are trying to study.

- **Set** is a collection of *distinct* objects in which the *order doesn't matter*.

SETS

- Set is a collection of *distinct* objects in which the *order doesn't matter*

SETS

- Set is a collection of *distinct* objects in which the *order doesn't matter*

SETS

- Set is a collection of *distinct* objects in which the *order doesn't matter*
    - {apple, peach} is a set

SETS

- Set is a collection of *distinct* objects in which the *order doesn't matter*
  - {apple, peach} is a set
  - {apple, apple, peach} is **not** a set

SETS

- Set is a collection of <u>*distinct*</u> objects in which the <u>*order doesn't matter*</u>

  - {apple, peach} is a set

  - {apple, apple, peach} is **not** a set

  - {apple, peach, pear} is the same as {peach, apple, pear}

SOME BASIC NOTATION

SOME BASIC NOTATION

- Set A contains elements a, b and c: $A = \{a, b, c\}$

SOME BASIC NOTATION

- Set A contains elements a, b and c: $A = \{a, b, c\}$
- Element a is in the set A: $a \in A$

SOME BASIC NOTATION

- Set A contains elements a, b and c: $A = \{a, b, c\}$
- Element a is in the set A: $a \in A$
- Set B contains all even numbers:

$$B = \{x \mid x = 2n, n = 0, 1, 2, \ldots\}$$

SOME BASIC NOTATION

- Set A contains elements a, b and c: $A = \{a, b, c\}$
- Element a is in the set A: $a \in A$
- Set B contains all even numbers:

$$B = \{x \mid x = 2n, n = 0, 1, 2, \ldots\}$$

- Element 5 doesn't belong to set B: $5 \notin B$

FINITE AND INFINITE SETS

- Sets can contain finite number of elements
  - digits: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
  - Latin characters: {A, B, C, …, X, Y, Z}
  - all possible 10-digit phone numbers

FINITE AND INFINITE SETS

- Sets can contain finite number of elements
  - digits: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
  - Latin characters: {A, B, C, …, X, Y, Z}
  - all possible 10-digit phone numbers

- But sets can also be infinite:
  - natural numbers $\mathbb{N}$: {1, 2, 3, …}
  - integer numbers $\mathbb{Z}$: {…, -2, -1, 0, 1, 2, …}

FINITE AND INFINITE SETS

- Sets can contain finite number of elements
  - digits: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
  - Latin characters: {A, B, C, …, X, Y, Z}
  - all possible 10-digit phone numbers
- But sets can also be infinite:
  - natural numbers $\mathbb{N}$: {1, 2, 3, …}
  - integer numbers $\mathbb{Z}$: {…, -2, -1, 0, 1, 2, …}

EMPTY SET

- A set with no elements:

EMPTY SET

- A set with no elements:

$$A = \{\}, \qquad A = \emptyset$$

CARDINALITY OF A SET

- Cardinality = “how large the set is”.

CARDINALITY OF A SET

- Cardinality = “how large the set is”.

- Finite sets: cardinality = number of elements.

CARDINALITY OF A SET

- Cardinality = "how large the set is".

- Finite sets: cardinality = number of elements.
  - A = {apple, banana, peach}, $|A| = 3$

CARDINALITY OF A SET

- Cardinality = “how large the set is”.

- Finite sets: cardinality = number of elements.
  - A = {apple, banana, peach}, $|A| = 3$
  - B – a set of Latin characters,

CARDINALITY OF A SET

- Cardinality = “how large the set is”.

- Finite sets: cardinality = number of elements.
  - A = {apple, banana, peach}, $|A| = 3$
  - B – a set of Latin characters, $|B| = 26$

CARDINALITY OF A SET

- Cardinality = "how large the set is".

- Finite sets: cardinality = number of elements.
  - A = {apple, banana, peach}, |A| = 3
  - B – a set of Latin characters, |B| = 26
  - C = {} – an empty set,

CARDINALITY OF A SET

- Cardinality = “how large the set is”.

- Finite sets: cardinality = number of elements.
  - A = {apple, banana, peach}, $|A| = 3$
  - B – a set of Latin characters, $|B| = 26$
  - C = {} – an empty set, $|C| = 0$

SUBSET OF A SET

- If each element of a set A is also an element of a set B, we say that A is a subset of B:

$$A \subseteq B$$

SUBSET OF A SET

- If each element of a set A is also an element of a set B, we say that A is a subset of B:

$$A \subseteq B$$

- Example:

SUBSET OF A SET

- If each element of a set A is also an element of a set B, we say that A is a subset of B:

$$A \subseteq B$$

- Example:

  A – a set of all numbers between 0 and 100 divisible by 3.
  B – a set of all numbers between 0 and 100.

SUBSET OF A SET

- If each element of a set A is also an element of a set B, we say that A is a subset of B:

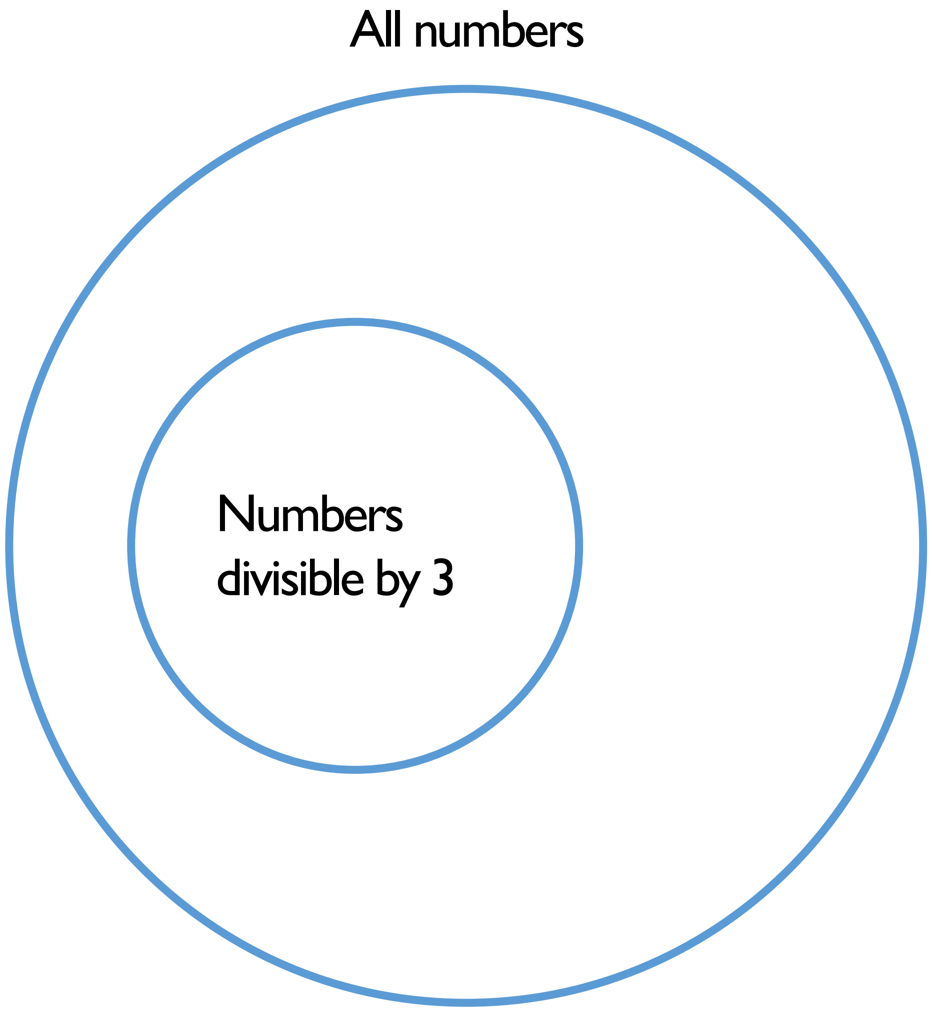
$$A \subseteq B$$

- Example:

  A – a set of all numbers between 0 and 100 divisible by 3.
  B – a set of all numbers between 0 and 100.

  $A \subseteq B$

VISUALIZATION



OPERATIONS WITH SETS

OPERATIONS ON SETS: UNION

- The **union** of A and B is

$$A \cup B = \{x \mid x \in A \text{ \underline{or} } x \in B\}$$

OPERATIONS ON SETS: UNION

- The **union** of A and B is

$$A \cup B = \{x \mid x \in A \ \underline{\text{or}}\ x \in B\}$$

- Example:

OPERATIONS ON SETS: UNION

- The **union** of A and B is

$$A \cup B = \{x \mid x \in A \ \underline{\text{or}}\ x \in B \}$$

- Example:

  A = {apple, banana}
  B = {orange, banana, peach}

  $A \cup B = \{ \qquad \}$

OPERATIONS ON SETS: UNION

- The **union** of A and B is

$$A \cup B = \{x \mid x \in A \ \underline{\text{or}}\ x \in B\}$$

- Example:

  A = {apple, banana}
  B = {orange, banana, peach}

  A ∪ B = {apple, banana, orange, peach}

OPERATIONS ON SETS: INTERSECTION

- The **intersection** of A and B is

$$A \cap B = \{x \mid x \in A \ \underline{\text{and}}\ x \in B\}$$

OPERATIONS ON SETS: INTERSECTION

- The **intersection** of A and B is

$$A \cap B = \{x \mid x \in A \ \underline{\text{and}}\ x \in B\}$$

- Example:

OPERATIONS ON SETS: INTERSECTION

- The **intersection** of A and B is

$$A \cap B = \{x \mid x \in A \ \underline{\text{and}}\ x \in B\}$$

- Example:

  A = {apple, banana}
  B = {orange, banana, peach}

OPERATIONS ON SETS: INTERSECTION

- The **intersection** of A and B is

$$A \cap B = \{x \mid x \in A \ \underline{\text{and}}\ x \in B\}$$

- Example:

  A = {apple, banana}
  B = {orange, banana, peach}

  $A \cap B = \{ \qquad\qquad \}$

OPERATIONS ON SETS: INTERSECTION

- The **intersection** of A and B is

$$A \cap B = \{x \mid x \in A \ \underline{\text{and}}\ x \in B\}$$

- Example:

  A = {apple, banana}
  B = {orange, banana, peach}

  $A \cap B$ = {banana}

OPERATIONS ON SETS: INTERSECTION

- Sets A and B are called **disjoint** if they don't have a common element:

$$A \cap B = \emptyset$$

OPERATIONS ON SETS: INTERSECTION

- Sets A and B are called **disjoint** if they don't have a common element:

$$A \cap B = \emptyset$$

- Example:

  A = {apple, banana}
  B = {orange, peach}

OPERATIONS ON SETS: INTERSECTION

- Sets A and B are called **disjoint** if they don't have a common element:

$$A \cap B = \emptyset$$

- Example:

  $A = \{\text{apple, banana}\}$
  $B = \{\text{orange, peach}\}$

  $A \cap B = \emptyset$

OPERATIONS ON SETS: DIFFERENCE

- The **set difference** of A and B is

$$A \setminus B = A - B = \{x \mid x \in A \text{ and } x \notin B\}$$

OPERATIONS ON SETS: DIFFERENCE

- The **set difference** of A and B is

$$A \setminus B = A - B = \{x \mid x \in A \text{ and } x \notin B\}$$

- Example:

OPERATIONS ON SETS: DIFFERENCE

- The **set difference** of A and B is

$$A \setminus B = A - B = \{x \mid x \in A \text{ and } x \notin B\}$$

- Example:

  A = {apple, banana}
  B = {orange, banana, peach}

OPERATIONS ON SETS: DIFFERENCE

- The **set difference** of A and B is

$$A \setminus B = A - B = \{x \mid x \in A \text{ and } x \notin B\}$$

- Example:

  $A = \{\text{apple, banana}\}$
  $B = \{\text{orange, banana, peach}\}$

  $A \setminus B = \{ \qquad \}$

OPERATIONS ON SETS: DIFFERENCE

- The **set difference** of A and B is

$$A \setminus B = A - B = \{x \mid x \in A \text{ and } x \notin B \}$$

- Example:

  $A = \{\text{apple, banana}\}$
  $B = \{\text{orange, banana, peach}\}$

  $A \setminus B = \{\text{apple}\}$

OPERATIONS ON SETS: DIFFERENCE

- The **set difference** of A and B is

$$A \setminus B = A - B = \{x \mid x \in A \text{ and } x \notin B \}$$

- Example:

  $A = \{\text{apple, banana}\}$
  $B = \{\text{orange, banana, peach}\}$

  $A \setminus B = \{\text{apple}\}$ $\qquad$ $B \setminus A = \{ \qquad \}$

OPERATIONS ON SETS: DIFFERENCE

- The **set difference** of A and B is

$$A \setminus B = A - B = \{x \mid x \in A \text{ and } x \notin B\}$$

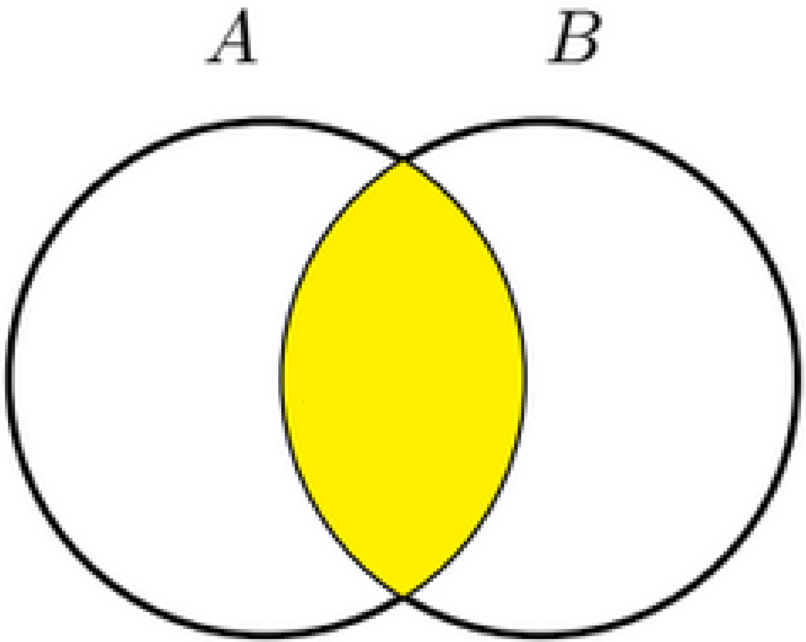
- Example:

  $A = \{\text{apple, banana}\}$
  $B = \{\text{orange, banana, peach}\}$

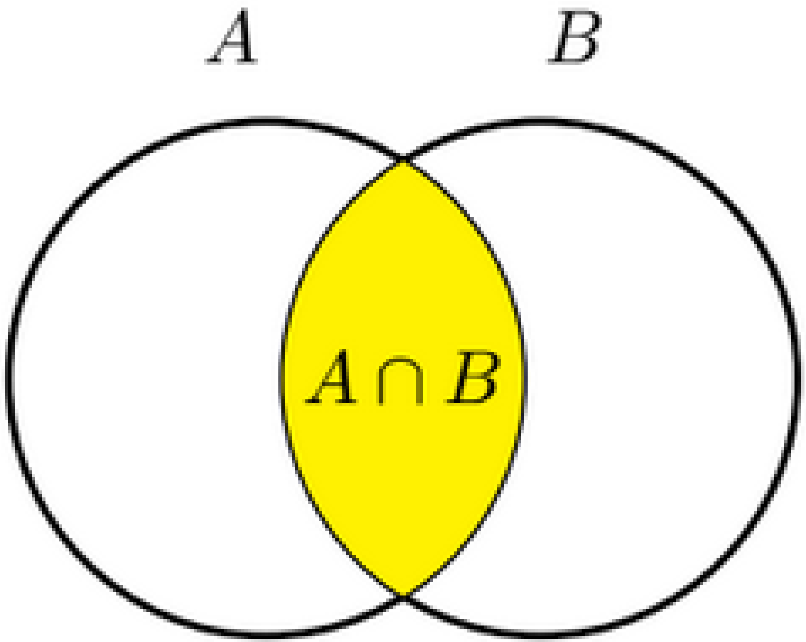
  $A \setminus B = \{\text{apple}\}$ $B \setminus A = \{\text{orange, peach}\}$

OPERATIONS WITH SETS: VENN'S DIAGRAMS

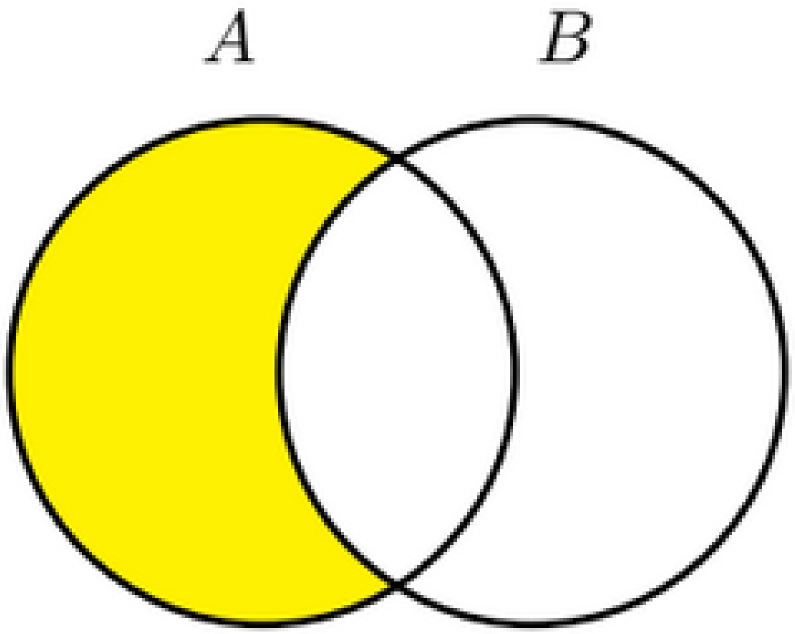
OPERATIONS WITH SETS: VENN'S DIAGRAMS



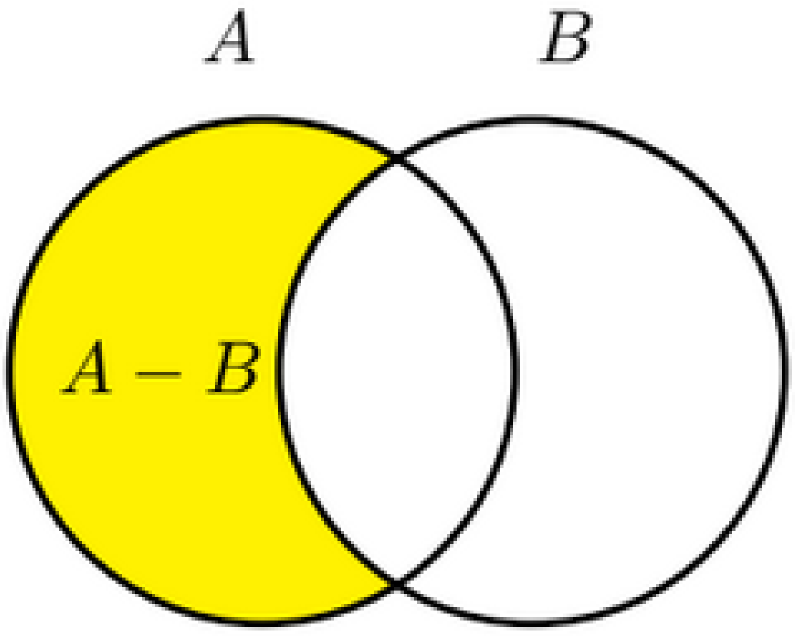
OPERATIONS WITH SETS: VENN'S DIAGRAMS



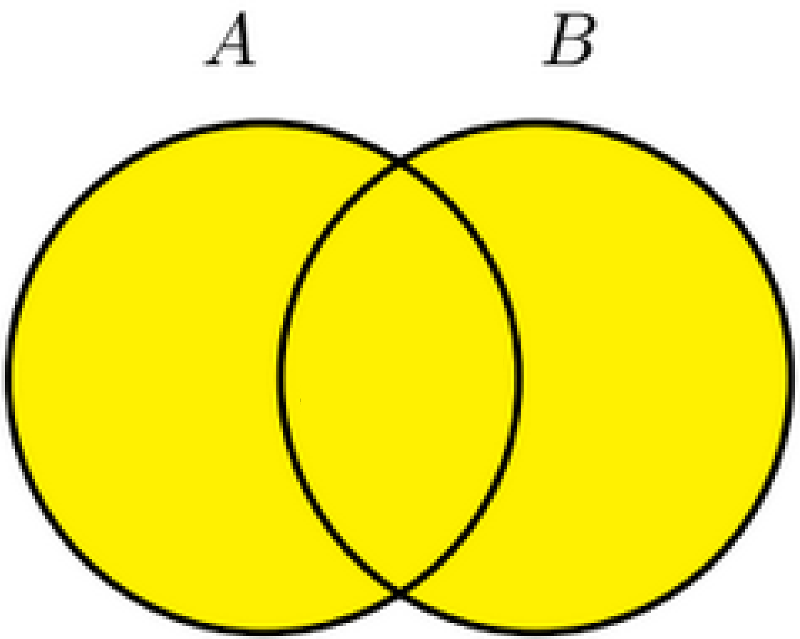
OPERATIONS WITH SETS: VENN'S DIAGRAMS



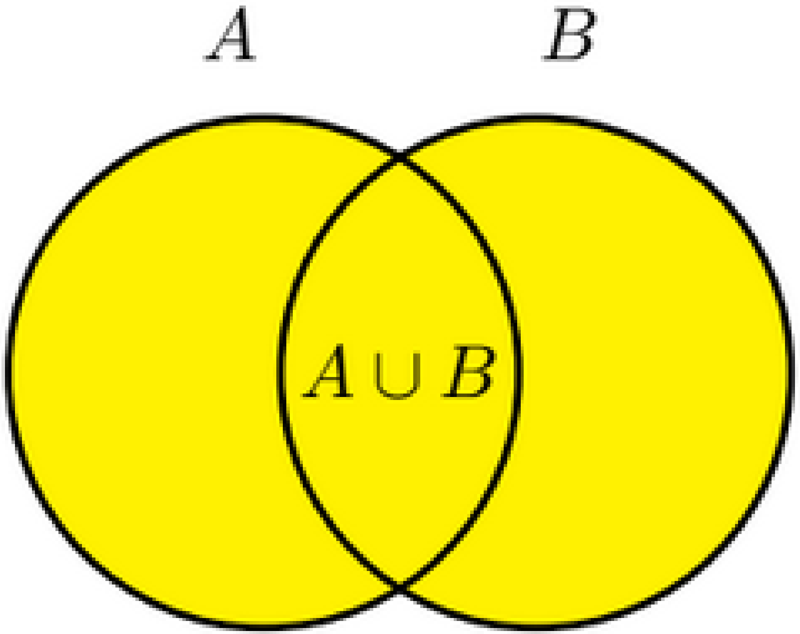
OPERATIONS WITH SETS: VENN'S DIAGRAMS



OPERATIONS WITH SETS: VENN'S DIAGRAMS



OPERATIONS WITH SETS: VENN'S DIAGRAMS

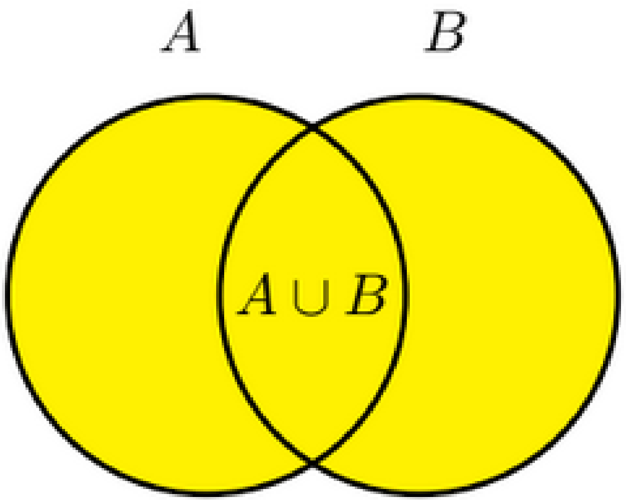


ABSORPTION LAW

$A \cap (A \cup B) = \ldots ?$

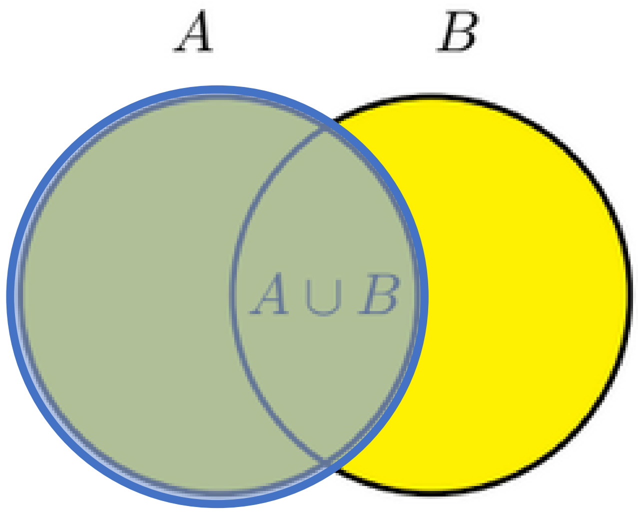
ABSORPTION LAW

$A \cap (A \cup B) = \ldots ?$



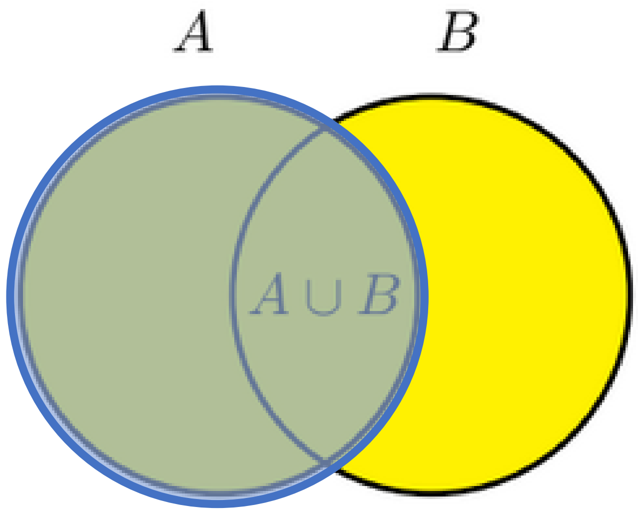
ABSORPTION LAW

$A \cap (A \cup B) = \ldots?$



ABSORPTION LAW

$A \cap (A \cup B) = A$



COMPLEMENT OF A SET

- Let be U the universal set.
  - “*a set of everything*”, whatever that is in the given context.

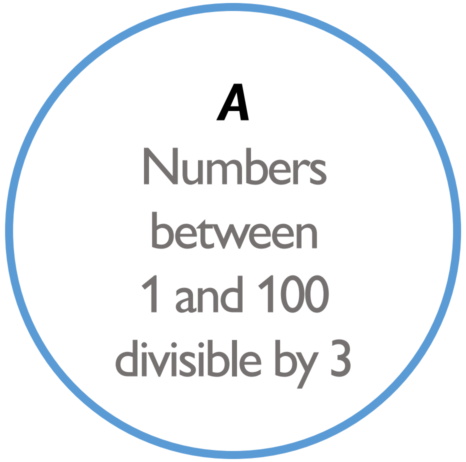
COMPLEMENT OF A SET

- Let be U the universal set.
  - “*a set of everything*”, whatever that is in the given context.

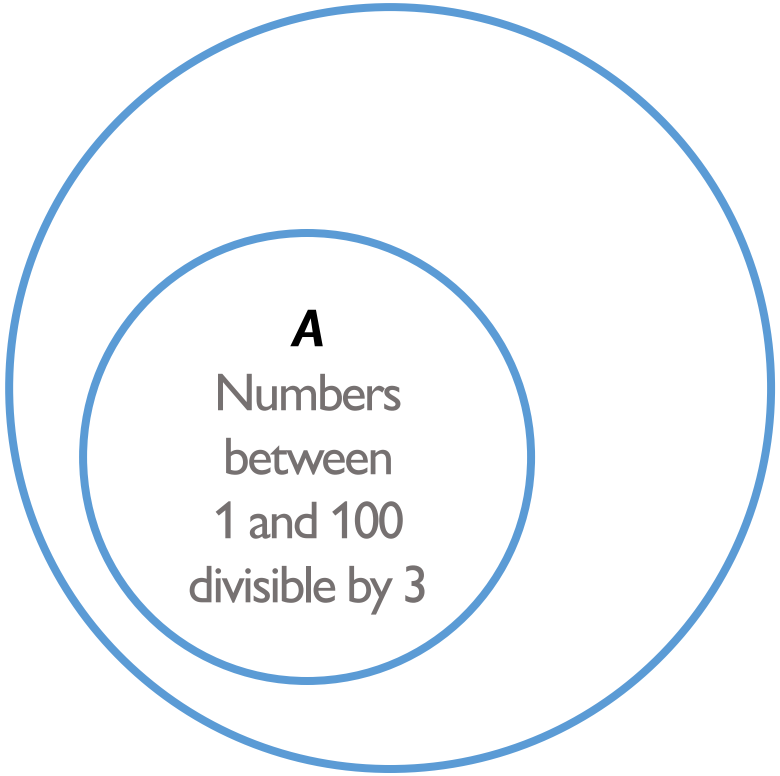
- **Complement** of A is

$$A^c = A = U \setminus A$$

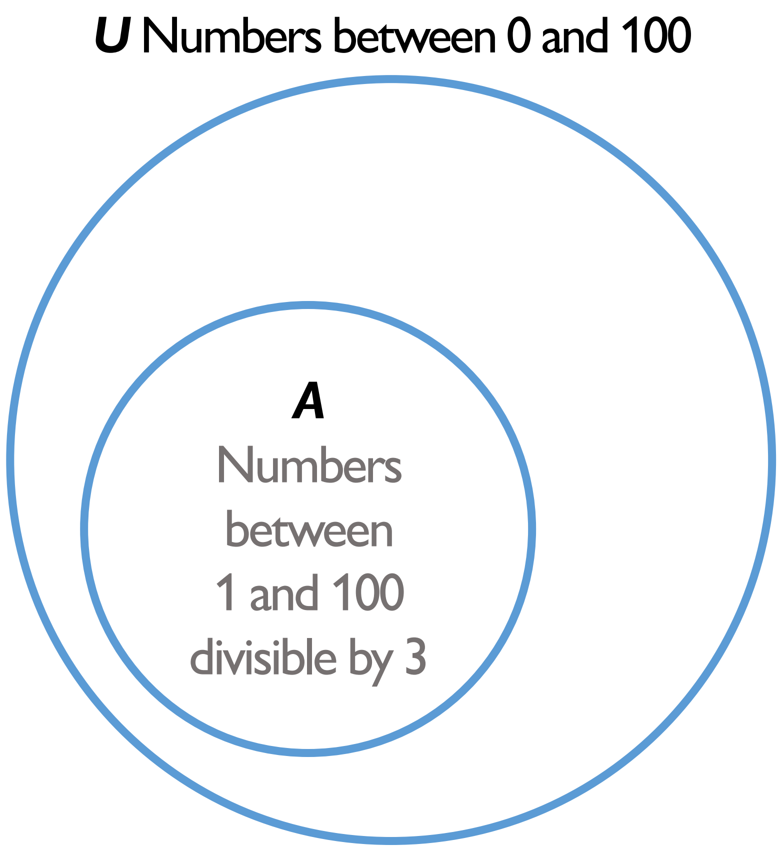
COMPLEMENT OF A SET: EXAMPLE



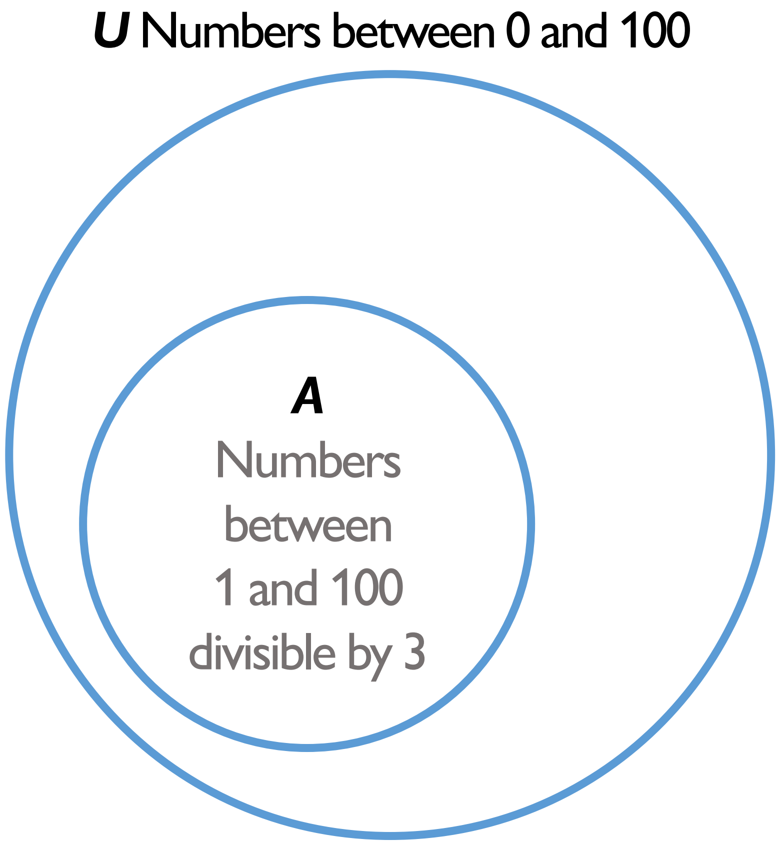
COMPLEMENT OF A SET: EXAMPLE



COMPLEMENT OF A SET: EXAMPLE

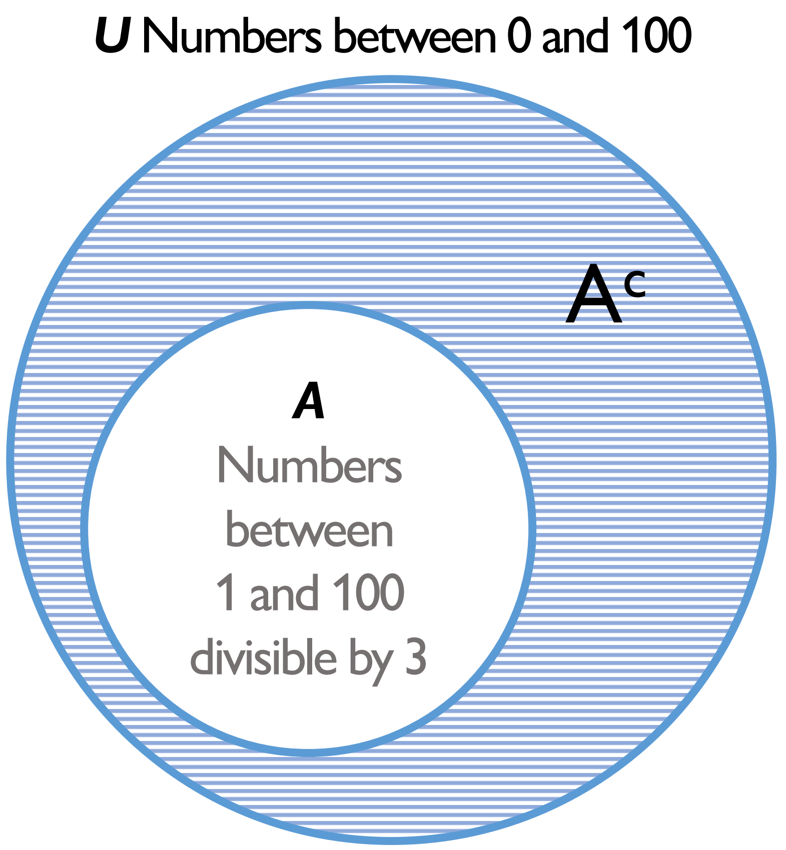


COMPLEMENT OF A SET: EXAMPLE

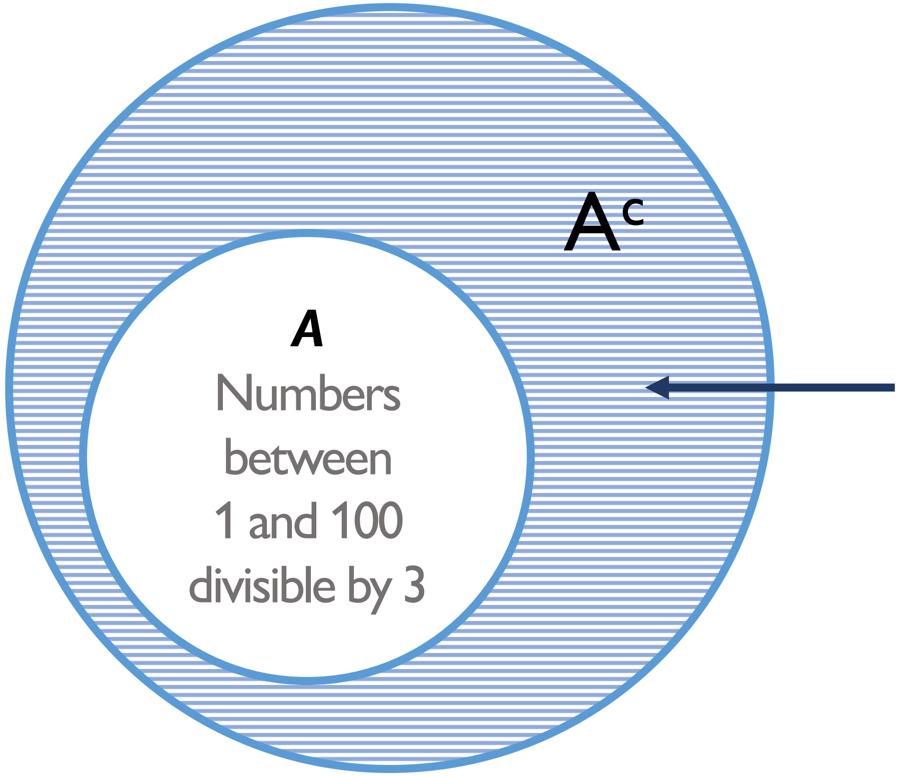


$A^c = ?$

COMPLEMENT OF A SET: EXAMPLE



COMPLEMENT OF A SET: EXAMPLE



Numbers between 0 and 100
not divisible by 3

SHORT QUIZ

1. $A \cup \emptyset =$

SHORT QUIZ

1. $A \cup \emptyset = A$

SHORT QUIZ

1. $A \cup \emptyset = A$

2. $A \cap \emptyset =$

SHORT QUIZ

1. $A \cup \emptyset = A$

2. $A \cap \emptyset = \emptyset$

SHORT QUIZ

1. $A \cup \emptyset = A$

2. $A \cap \emptyset = \emptyset$

3. $A \cup U =$

SHORT QUIZ

1. $A \cup \emptyset = A$

2. $A \cap \emptyset = \emptyset$

3. $A \cup U = U$

SHORT QUIZ

1. $A \cup \emptyset = A$
2. $A \cap \emptyset = \emptyset$
3. $A \cup U = U$
4. $A \cap U =$

SHORT QUIZ

1. $A \cup \emptyset = A$
2. $A \cap \emptyset = \emptyset$
3. $A \cup U = U$
4. $A \cap U = A$

OTHER LAWS

Name	Rule
Commutative laws	$A \cap B = B \cap A, \quad A \cup B = B \cup A$
Associative laws	$A \cap (B \cap C) = (A \cap B) \cap C$
	$A \cup (B \cup C) = (A \cup B) \cup C$
Distributive laws	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
De Morgan's laws	$(A \cup B)^c = A^c \cap B^c, \quad (A \cap B)^c = A^c \cup B^c$
Complement laws	$A \cap A^c = \varnothing, \quad A \cup A^c = U$
Double complement law	$(A^c)^c = A$
Idempotent laws	$A \cap A = A, \quad A \cup A = A$
Absorption laws	$A \cap (A \cup B) = A, \quad A \cup (A \cap B) = A$
Dominance laws	$A \cap \varnothing = \varnothing, \quad A \cup U = U$
Identity laws	$A \cup \varnothing = A, \quad A \cap U = A$

CARTESIAN PRODUCT

- Let A and B be two sets.
- **Cartesian product** of two sets A and B is a set, containing all sequences, where the first element is from set A and the second is from set B:

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

CARTESIAN PRODUCT: EXAMPLE

- D = {wine, beer, water}
- F = {spaghetti, burgers}

CARTESIAN PRODUCT: EXAMPLE

- D = {wine, beer, water}
- F = {spaghetti, burgers}

D x F = {

}

CARTESIAN PRODUCT: EXAMPLE

- D = {wine, beer, water}
- F = {spaghetti, burgers}

D x F = {(wine, spaghetti),
(wine, burgers),

}

CARTESIAN PRODUCT: EXAMPLE

- D = {wine, beer, water}
- F = {spaghetti, burgers}

D x F = {(wine, spaghetti),
(wine, burgers),
(beer, spaghetti),
(beer, burgers),

}

CARTESIAN PRODUCT: EXAMPLE

- D = {wine, beer, water}
- F = {spaghetti, burgers}

D x F = {(wine, spaghetti),
(wine, burgers),
(beer, spaghetti),
(beer, burgers),
(water, spaghetti),
(water, burgers)}

BACK TO COUNTING...

ADDITION PRINCIPLE

(aka Rule of sum)

RULE OF SUM: EXAMPLE

- We want to order dinner for tonight:
  - Pizza:
  - Sushi:

RULE OF SUM: EXAMPLE

- We want to order dinner for tonight:
  - Pizza:

    Dominos, Pizza Hut or King Slice (3 restaurants)
  - Sushi:

RULE OF SUM: EXAMPLE

- We want to order dinner for tonight:
  - Pizza:

    Dominos, Pizza Hut or King Slice (3 restaurants)
  - Sushi:

    Wabi Sabi or Sakura (2 restaurants)

RULE OF SUM: EXAMPLE

- We want to order dinner for tonight:
  - Pizza:

    Dominos, Pizza Hut or King Slice (3 restaurants)
  - Sushi:

    Wabi Sabi or Sakura (2 restaurants)

- How many options are there in total?

RULE OF SUM: EXAMPLE

- We want to order dinner for tonight:
  - Pizza:

    Dominos, Pizza Hut or King Slice (3 restaurants)
  - Sushi:

    Wabi Sabi or Sakura (2 restaurants)

- How many options are there in total?

  3 pizza places **+** **2** sushi places =

RULE OF SUM: EXAMPLE

- We want to order dinner for tonight:
  - Pizza:

    Dominos, Pizza Hut or King Slice (3 restaurants)
  - Sushi:

    Wabi Sabi or Sakura (2 restaurants)

- How many options are there in total?

  3 pizza places **+** **2** sushi places = **3** **+** **2** = 5 options in total

RULE OF SUM

- If there are **a** ways of doing something and **b** number of ways of doing another thing and we can not do both at the same time, then there are **a + b** ways to choose one of the actions.

RULE OF SUM (SET THEORY)

- If A and B are <u>disjoint sets</u> ($A \cap B = \emptyset$), then

RULE OF SUM (SET THEORY)

- If A and B are <u>disjoint sets</u> ($A \cap B = \emptyset$), then

$$|A \cup B| =$$

RULE OF SUM (SET THEORY)

- If A and B are disjoint sets ($A \cap B = \emptyset$), then

$$|A \cup B| = |A| + |B|$$

RULE OF SUM (SET THEORY)

- If A and B are <u>disjoint sets</u> ($A \cap B = \emptyset$), then

$$|A \cup B| = |A| + |B|$$

- **Generalized rule of sum**:

RULE OF SUM (SET THEORY)

- If A and B are disjoint sets ($A \cap B = \emptyset$), then

$$|A \cup B| = |A| + |B|$$

- **Generalized rule of sum**:

  If $A_1 \cap A_2 \cap \ldots \cap A_n = \emptyset$, then

RULE OF SUM (SET THEORY)

- If A and B are disjoint sets ($A \cap B = \emptyset$), then

$$|A \cup B| = |A| + |B|$$

- **Generalized rule of sum**:

  If $A_1 \cap A_2 \cap \ldots \cap A_n = \emptyset$, then

$$|A_1 \cup A_2 \cup \ldots \cup A_n| = |A_1| + |A_2| + \ldots + |A_n|$$

MULTIPLICATION PRINCIPLE

(aka Rule of product)

RULE OF PRODUCT INFORMALLY

- If there are **a** ways of doing something and **b** ways of doing another thing, then there are **a** · **b** ways of performing both actions.

RULE OF PRODUCT: EXAMPLE 1

- You want to order pizza:
  - first, choose the type of crust:
  - second, choose one topping:

RULE OF PRODUCT: EXAMPLE 1

- You want to order pizza:
  - first, choose the type of crust:
    thin or thick (2 choices);
  - second, choose one topping:

RULE OF PRODUCT: EXAMPLE 1

- You want to order pizza:
  - first, choose the type of crust:
    thin or thick (2 choices);
  - second, choose one topping:
    cheese, pepperoni, or sausage (3 choices).

RULE OF PRODUCT: EXAMPLE 1

- You want to order pizza:
  - first, choose the type of crust:
    thin or thick (2 choices);
  - second, choose one topping:
    cheese, pepperoni, or sausage (3 choices).
- How many different pizzas are there?

RULE OF PRODUCT: EXAMPLE 1

- You want to order pizza:
  - first, choose the type of crust:
    thin or thick (2 choices);
  - second, choose one topping:
    cheese, pepperoni, or sausage (3 choices).
- How many different pizzas are there?

2 ways of choosing crust **x** **3** ways of choosing topping =

RULE OF PRODUCT: EXAMPLE 1

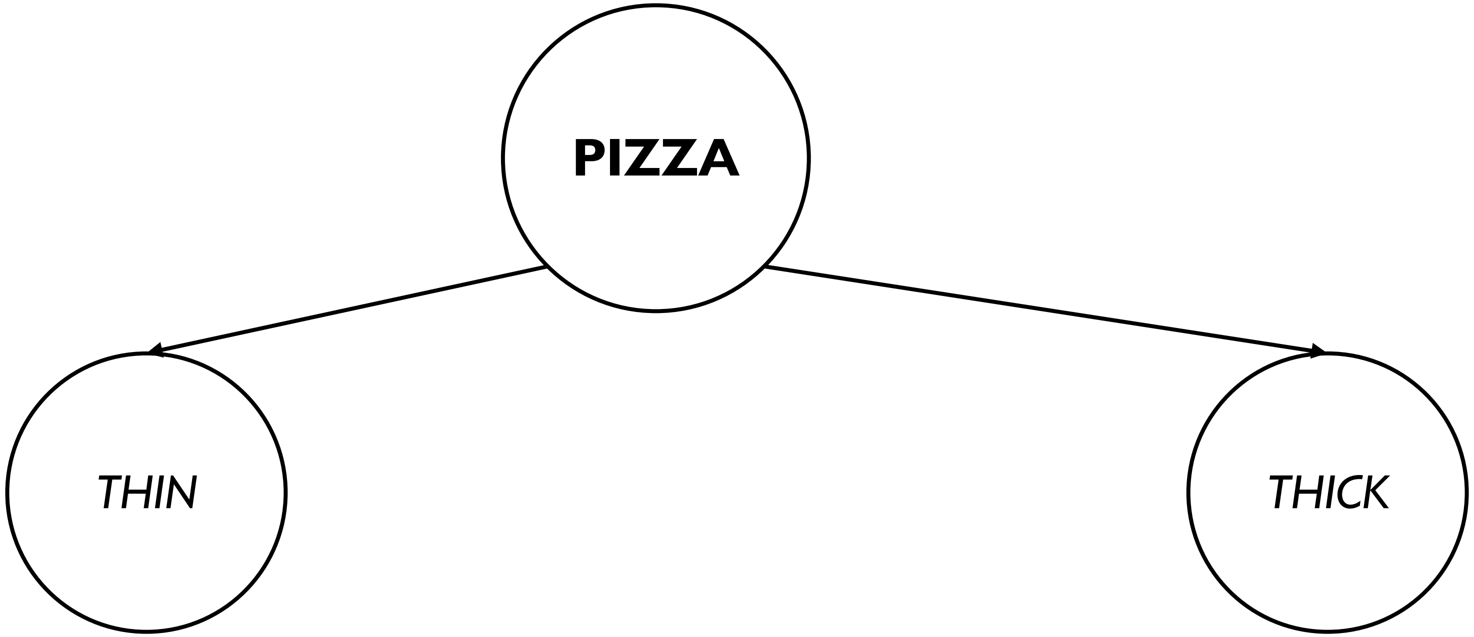
- You want to order pizza:
  - first, choose the type of crust:
    thin or thick (2 choices);
  - second, choose one topping:
    cheese, pepperoni, or sausage (3 choices).
- How many different pizzas are there?

2 ways of choosing crust **x** **3** ways of choosing topping =
= **2 x 3** = 6 different pizzas

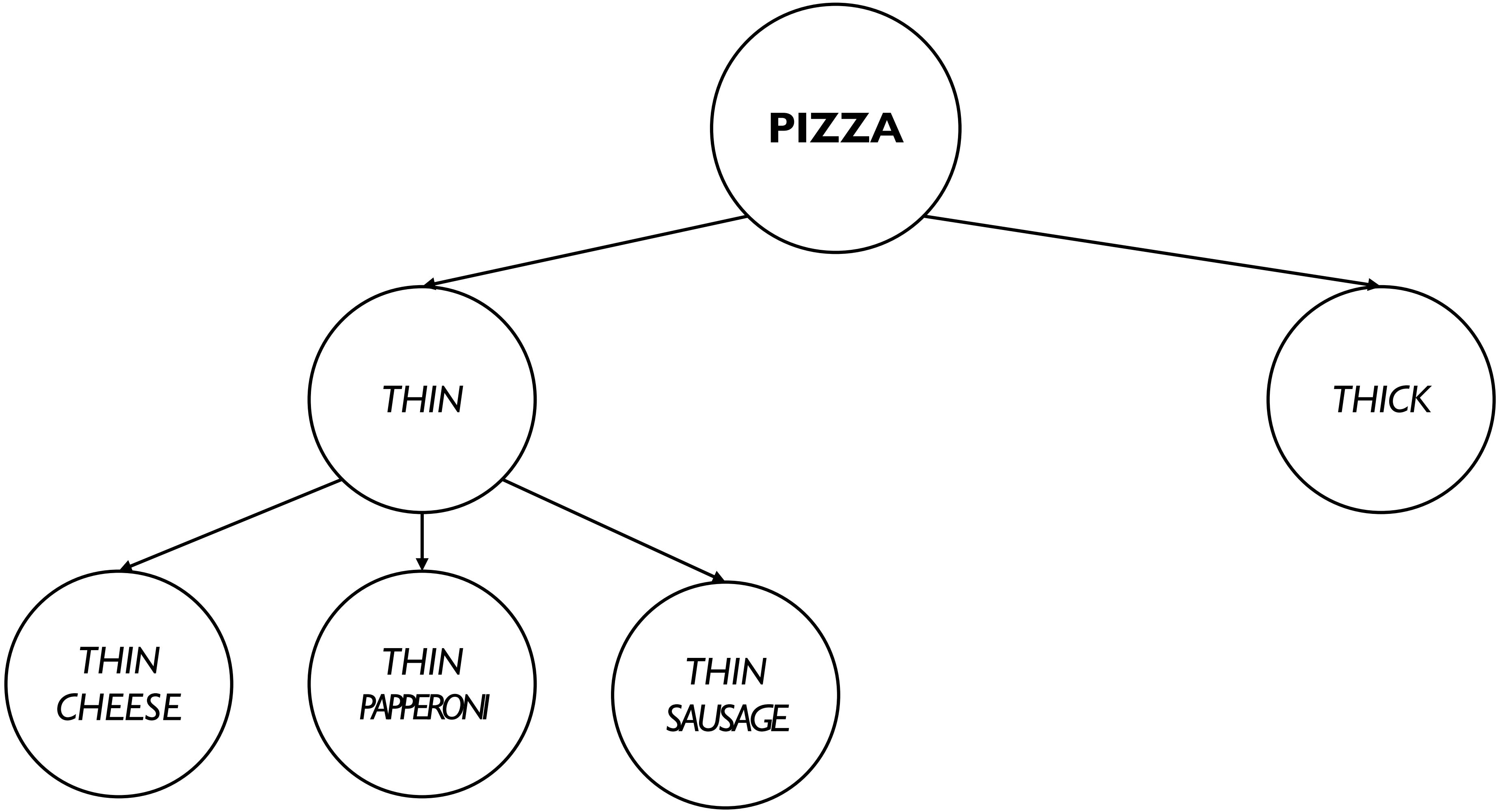
RULE OF PRODUCT: EXAMPLE 1

PIZZA

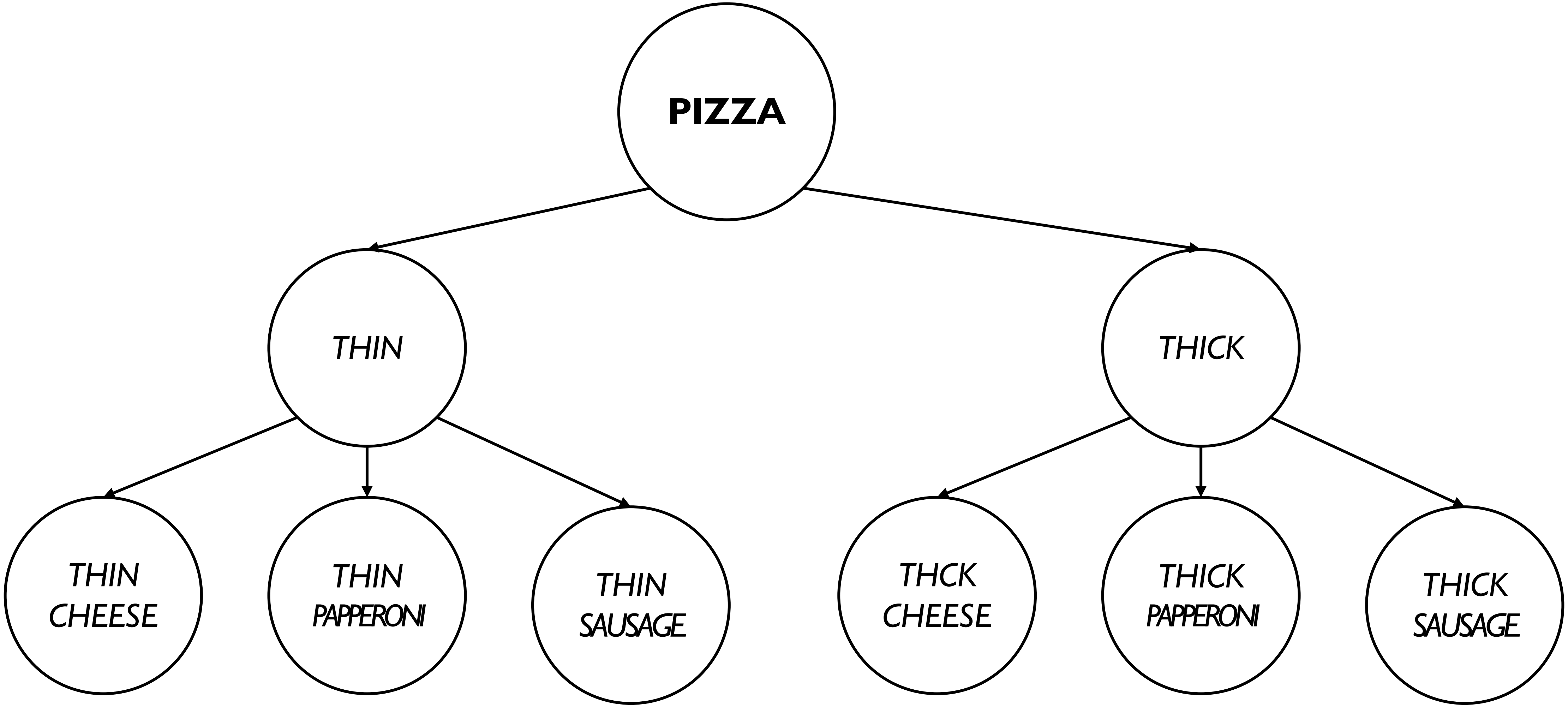
RULE OF PRODUCT: EXAMPLE 1



RULE OF PRODUCT: EXAMPLE 1



RULE OF PRODUCT: EXAMPLE 1



RULE OF PRODUCT (SET THEORY)

RULE OF PRODUCT (SET THEORY)

- For any two sets A and B

RULE OF PRODUCT (SET THEORY)

- For any two sets A and B

$$|A \times B| =$$

RULE OF PRODUCT (SET THEORY)

- For any two sets A and B

$$|A \times B| = |A| \cdot |B|$$

RULE OF PRODUCT (SET THEORY)

- For any two sets A and B

$$|A \times B| = |A| \cdot |B|$$

- **Generalized product rule:**

RULE OF PRODUCT (SET THEORY)

- For any two sets A and B

$$|A \times B| = |A| \cdot |B|$$

- **Generalized product rule:**

  For any sets $A_{1,}\ A_2,\ \ldots,\ A_n$

RULE OF PRODUCT (SET THEORY)

- For any two sets A and B

$$|A \times B| = |A| \cdot |B|$$

- **Generalized product rule:**

  For any sets $A_1, A_2, \ldots, A_n$

$$|A_1 \times A_2 \times \ldots \times A_n| = |A_1| \cdot |A_2| \cdot \ldots \cdot |A_n|$$

RULE OF PRODUCT: EXAMPLE 2

- One can chose from the following meals during the day:
  - **B**reakfast = {pancakes, eggs, bagel}
  - **L**unch = {salad, soup, bagel}
  - **D**inner = {soup, pasta, burgers}

RULE OF PRODUCT: EXAMPLE 2

- One can chose from the following meals during the day:
  - **B**reakfast = {pancakes, eggs, bagel}
  - **L**unch = {salad, soup, bagel}
  - **D**inner = {soup, pasta, burgers}

How many different day menus can one have?

RULE OF PRODUCT: EXAMPLE 2

- One can chose from the following meals during the day:
  - **B**reakfast = {pancakes, eggs, bagel}
  - **L**unch = {salad, soup, bagel}
  - **D**inner = {soup, pasta, burgers}

How many different day menus can one have?

$$|B| \cdot |L| \cdot |D| = 3 \cdot 3 \cdot 3 = 27$$

RULE OF PRODUCT: EXAMPLE 3

- License plate:

3 letters + 3 digits + regional code

- How many unique plates are there per regional code?

RULE OF PRODUCT: EXAMPLE 3

- License plate:

3 letters + 3 digits + regional code

- How many unique plates are there per regional code?
- 12 letters and 10 digits can be used:

RULE OF PRODUCT: EXAMPLE 3

- License plate:

3 letters + 3 digits + regional code

- How many unique plates are there per regional code?

- 12 letters and 10 digits can be used:

$$12 \cdot 12 \cdot 12 \cdot 10 \cdot 10 \cdot 10$$

RULE OF PRODUCT: EXAMPLE 3

- License plate:

3 letters + 3 digits + regional code

- How many unique plates are there per regional code?

- 12 letters and 10 digits can be used:

$$12 \cdot 12 \cdot 12 \cdot 10 \cdot 10 \cdot 10 =$$

1 728 000 unique plates

MORE EXAMPLES

Sum and product rules in action

SUM & PRODUCT RULES IN PROGRAMMING

```
for i in range(n):
    print('hello')

for j in range(m):
    print('hello')
```

```
for i in range(n):
    for j in range(m):
        print('hello')
```

SUM & PRODUCT RULES IN PROGRAMMING

RULE OF SUM

```
for i in range(n):
     print('hello')

for j in range(m):
     print('hello')
```

```
for i in range(n):
     for j in range(m):
          print('hello')
```

SUM & PRODUCT RULES IN PROGRAMMING

RULE OF SUM

```
for i in range(n):
     print('hello')

for j in range(m):
     print('hello')
```

```
for i in range(n):
     for j in range(m):
          print('hello')
```

hello will be printed **m + n** times

SUM & PRODUCT RULES IN PROGRAMMING

RULE OF SUM

```
for i in range(n):
     print('hello')

for j in range(m):
     print('hello')
```

hello will be printed **m + n** times

RULE OF PRODUCT

```
for i in range(n):
     for j in range(m):
          print('hello')
```

SUM & PRODUCT RULES IN PROGRAMMING

RULE OF SUM

```
for i in range(n):
    print('hello')

for j in range(m):
    print('hello')
```

hello will be printed **m + n** times

RULE OF PRODUCT

```
for i in range(n):
    for j in range(m):
        print('hello')
```

hello will be printed **m*n** times

NUMBER OF 6-BIT STRINGS

- Example of a 6-bit string:

0	**1**	**1**	**1**	**0**	**1**

- How many such strings are there?

NUMBER OF 6-BIT STRINGS

- Example of a 6-bit string:

0	**1**	**1**	**1**	**0**	**1**

- How many such strings are there?
- Solution:

NUMBER OF 6-BIT STRINGS

- Example of a 6-bit string:

0	1	1	1	0	1

- How many such strings are there?

- Solution:

*	*	*	*	*	*

NUMBER OF 6-BIT STRINGS

- Example of a 6-bit string:

0	**1**	**1**	**1**	**0**	**1**

- How many such strings are there?

- Solution:

*	*	*	*	*	*

Each position:

NUMBER OF 6-BIT STRINGS

- Example of a 6-bit string:

0	**1**	**1**	**1**	**0**	**1**

- How many such strings are there?

- Solution:

*	*	*	*	*	*

Each position: 0 or 1 (2 choices)

NUMBER OF 6-BIT STRINGS

- Example of a 6-bit string:

0	**1**	**1**	**1**	**0**	**1**

- How many such strings are there?

- Solution:

*	*	*	*	*	*

Each position: 0 or 1 (2 choices)
Product rule:

NUMBER OF 6-BIT STRINGS

- Example of a 6-bit string:

0	**1**	**1**	**1**	**0**	**1**

- How many such strings are there?

- Solution:

*	*	*	*	*	*

Each position: 0 or 1 (2 choices)
Product rule:

$$2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^6 = 64 \text{ different 6-bit strings.}$$

NUMBER OF 6-BIT STRINGS

- How many such strings are there that start with *10*?

NUMBER OF 6-BIT STRINGS

- How many such strings are there that start with *10*?
- Solution:

1	**0**	*	*	*	*

NUMBER OF 6-BIT STRINGS

- How many such strings are there that start with *10*?
- Solution:

1	**0**	*	*	*	*

Positions 1 & 2: 10 Other positions: 0 or 1 (2 choices)

NUMBER OF 6-BIT STRINGS

- How many such strings are there that start with *10*?
- Solution:

1	**0**	*	*	*	*

Positions 1 & 2: 10 Other positions: 0 or 1 (2 choices)

Product rule:

NUMBER OF 6-BIT STRINGS

- How many such strings are there that start with *10*?
- Solution:

1	**0**	*	*	*	*

Positions 1 & 2: 10 Other positions: 0 or 1 (2 choices)

Product rule:

$$1 \cdot 1 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^4 = 16$$

different 6-bit strings that start with **10**.

NUMBER OF 6-BIT STRINGS

- How many such strings are there that **don’t** start with *10*?

NUMBER OF 6-BIT STRINGS

- How many such strings are there that **don't** start with *10*?
- Solution:

not 10

*	*	*	*	*	*

NUMBER OF 6-BIT STRINGS

- How many such strings are there that **don’t** start with *10*?

- Solution:

not 10

*	*	*	*	*	*

SOLUTION 1 |strings that **don’t** start with *10*| =

NUMBER OF 6-BIT STRINGS

- How many such strings are there that **don’t** start with *10*?

- Solution: **not 10**

*	*	*	*	*	*

SOLUTION 1 |strings that **don’t** start with *10*| =
|strings that start with *00*| + |strings that start with *01*| +
+ |strings that start with *11*|

NUMBER OF 6-BIT STRINGS

- How many such strings are there that **don’t** start with *10*?

- Solution:

not 10

*	*	*	*	*	*

SOLUTION 1 |strings that **don’t** start with *10*| =
|strings that start with *00*| + |strings that start with *01*| +
+ |strings that start with *11*|

$2^4 + 2^4 + 2^4 = 16 + 16 + 16 = 48$

NUMBER OF 6-BIT STRINGS

- How many such strings are there that **don’t** start with *10*?

- Solution:

not 10

*	*	*	*	*	*

SOLUTION 2 |strings that **don’t** start with *10*| =

NUMBER OF 6-BIT STRINGS

- How many such strings are there that **don’t** start with *10*?

- Solution:

not 10

*	*	*	*	*	*

SOLUTION 2

|strings that **don’t** start with *10*| =
|all strings| - |strings that **do** start with *10*|

NUMBER OF 6-BIT STRINGS

- How many such strings are there that **don't** start with *10*?

- Solution:

not 10

*	*	*	*	*	*

SOLUTION 2

|strings that **don't** start with *10*| =
|all strings| - |strings that **do** start with *10*|

$2^6 - 2^4 = 64 – 16 = 48$

NUMBER OF BIT STRINGS OF LENGTH 6 OR LESS

- How many bit strings are there of length 6 or less?

NUMBER OF BIT STRINGS OF LENGTH 6 OR LESS

- How many bit strings are there of length 6 or less?
- Solution:

NUMBER OF BIT STRINGS OF LENGTH 6 OR LESS

- How many bit strings are there of length 6 or less?

- Solution:

1. Sum rule:

$N = N_6 + N_5 + N_4 + N_3 + N_2 + N_1$, $\quad N_i$ – # bit strings of length i

NUMBER OF BIT STRINGS OF LENGTH 6 OR LESS

- How many bit strings are there of length 6 or less?

- Solution:

1. *Sum rule:*

$N = N_6 + N_5 + N_4 + N_3 + N_2 + N_1,$ $\quad N_i$ – # bit strings of length i

$$N_i =$$

NUMBER OF BIT STRINGS OF LENGTH 6 OR LESS

- How many bit strings are there of length 6 or less?
- Solution:

1. *Sum rule:*

   $N = N_6 + N_5 + N_4 + N_3 + N_2 + N_1,$ $\quad N_i$ – # bit strings of length i

2. *Product rule:*

$$N_i = 2^i$$

NUMBER OF BIT STRINGS OF LENGTH 6 OR LESS

- How many bit strings are there of length 6 or less?

- Solution:

1. *Sum rule:*

   $N = N_6 + N_5 + N_4 + N_3 + N_2 + N_1$, N_i – # bit strings of length i

2. *Product rule:*

$$N_i = 2^i$$

$$N = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 = 126$$

SUBSETS OF A SET

- Consider a set {a, b, c}. How many subsets does it have? List them all.

SUBSETS OF A SET

- Consider a set {a, b, c}. How many subsets does it have? List them all.
- 0 elements:
- 1 element:
- 2 elements:
- 3 elements:

SUBSETS OF A SET

- Consider a set {a, b, c}. How many subsets does it have? List them all.
- 0 elements:

{}

- 1 element:
- 2 elements:
- 3 elements:

SUBSETS OF A SET

- Consider a set {a, b, c}. How many subsets does it have? List them all.
- 0 elements:

  {}

- 1 element:

  {a} {b} {c}

- 2 elements:
- 3 elements:

SUBSETS OF A SET

- Consider a set {a, b, c}. How many subsets does it have? List them all.
- 0 elements:

{}

- 1 element:

{a} {b} {c}

- 2 elements:

{a, b} {a, c} {b, c}

- 3 elements:

SUBSETS OF A SET

- Consider a set {a, b, c}. How many subsets does it have? List them all.

- 0 elements:

{}

- 1 element:

{a} {b} {c}

- 2 elements:

{a, b} {a, c} {b, c}

- 3 elements:

{a, b, c}

NUMBER OF SUBSETS OF A SET

- A 3-element set has 8 subsets.
- How many subsets does a set of *n* elements have?

NUMBER OF SUBSETS OF A SET

- A 3-element set has 8 subsets.
- How many subsets does a set of *n* elements have?
- **Solution:**

NUMBER OF SUBSETS OF A SET

- A 3-element set has 8 subsets.
- How many subsets does a set of *n* elements have?

- **Solution:**

  Each element can either be included in the subset (1) or not (0):

NUMBER OF SUBSETS OF A SET

- A 3-element set has 8 subsets.
- How many subsets does a set of *n* elements have?

- **Solution:**

  Each element can either be included in the subset (1) or not (0):

0	**1**	**0**	**…**	**0**	**0**	**1**

NUMBER OF SUBSETS OF A SET

- A 3-element set has 8 subsets.
- How many subsets does a set of *n* elements have?

- **Solution:**

  Each element can either be included in the subset (1) or not (0):

0	**1**	**0**	**…**	**0**	**0**	**1**

  2 options per each of the *n* element ***(just like bit-strings!)***.

NUMBER OF SUBSETS OF A SET

- A 3-element set has 8 subsets.
- How many subsets does a set of *n* elements have?

- **Solution:**

  Each element can either be included in the subset (1) or not (0):

0	**1**	**0**	**...**	**0**	**0**	**1**

  2 options per each of the *n* element ***(just like bit-strings!)***.

  Product rule:

NUMBER OF SUBSETS OF A SET

- A 3-element set has 8 subsets.
- How many subsets does a set of *n* elements have?

- **Solution:**

  Each element can either be included in the subset (1) or not (0):

0	**1**	**0**	**…**	**0**	**0**	**1**

  2 options per each of the *n* element ***(just like bit-strings!)***.

  Product rule: $2 \cdot 2 \cdot \ldots \cdot 2 = 2^n$ subsets

EXAMPLE: NUMBER OF PASSWORDS

- A valid password contains 6 to 8 symbols
  - first symbol is a letter (upper- or lowercase);
  - other symbols are letters (upper- or lowercase) or digits.
- How many different passwords are there?

EXAMPLE: NUMBER OF PASSWORDS

- A valid password contains 6 to 8 symbols
  - first symbol is a letter (upper- or lowercase);
  - other symbols are letters (upper- or lowercase) or digits.
- How many different passwords are there?

- Solution:

EXAMPLE: NUMBER OF PASSWORDS

- A valid password contains 6 to 8 symbols
  - first symbol is a letter (upper- or lowercase);
  - other symbols are letters (upper- or lowercase) or digits.
- How many different passwords are there?

- Solution:

  Sum rule:

EXAMPLE: NUMBER OF PASSWORDS

- A valid password contains 6 to 8 symbols
  - first symbol is a letter (upper- or lowercase);
  - other symbols are letters (upper- or lowercase) or digits.
- How many different passwords are there?

- Solution:

  Sum rule:

$$N = P_6 + P_7 + P_8 \qquad P_i \text{ – \# passwords of length i}$$

EXAMPLE: NUMBER OF PASSWORDS

- Solution:

Sum rule: $N = P_6 + P_7 + P_8$ P_i – # passwords of length i

EXAMPLE: NUMBER OF PASSWORDS

- Solution:

Sum rule: $N = P_6 + P_7 + P_8$ P_i – # passwords of length i

First symbol:

upper- or lowercase letter

Other symbols:

upper- or lowercase letter or digit

EXAMPLE: NUMBER OF PASSWORDS

- Solution:

Sum rule: $N = P_6 + P_7 + P_8$ P_i – # passwords of length i

First symbol:

upper- or lowercase letter – $2 \cdot 26 = 52$ options

Other symbols:

upper- or lowercase letter or digit

EXAMPLE: NUMBER OF PASSWORDS

- Solution:

Sum rule: $N = P_6 + P_7 + P_8$ P_i – # passwords of length i

First symbol:

upper- or lowercase letter – $2 \cdot 26 = 52$ options

Other symbols:

upper- or lowercase letter or digit – $52 + 10 = 62$ options

EXAMPLE: NUMBER OF PASSWORDS

- Solution:

Sum rule: $N = P_6 + P_7 + P_8$ P_i – # passwords of length i

First symbol:

upper- or lowercase letter – $2 \cdot 26 = 52$ options

Other symbols:

upper- or lowercase letter or digit – $52 + 10 = 62$ options

$P_6 =$ $P_7 =$ $P_8 =$

EXAMPLE: NUMBER OF PASSWORDS

- Solution:

Sum rule: $N = P_6 + P_7 + P_8$ P_i – # passwords of length i

First symbol:

upper- or lowercase letter – $2 \cdot 26 = 52$ options

Other symbols:

upper- or lowercase letter or digit – $52 + 10 = 62$ options

Product rule: $P_6 =$ $P_7 =$ $P_8 =$

EXAMPLE: NUMBER OF PASSWORDS

- Solution:

Sum rule: $N = P_6 + P_7 + P_8$ P_i – # passwords of length i

First symbol:

upper- or lowercase letter – $2 \cdot 26 = 52$ options

Other symbols:

upper- or lowercase letter or digit – $52 + 10 = 62$ options

Product rule: $P_6 = 52 \cdot 62^5$ $P_7 =$ $P_8 =$

EXAMPLE: NUMBER OF PASSWORDS

- Solution:

Sum rule: $N = P_6 + P_7 + P_8$ P_i – # passwords of length i

First symbol:

upper- or lowercase letter – $2 \cdot 26 = 52$ options

Other symbols:

upper- or lowercase letter or digit – $52 + 10 = 62$ options

Product rule: $P_6 = 52 \cdot 62^5$ $P_7 = 52 \cdot 62^6$ $P_8 = 52 \cdot 62^7$

EXAMPLE: NUMBER OF PASSWORDS

- Solution:

Sum rule: $N = P_6 + P_7 + P_8$ $\qquad$ P_i – # passwords of length i

First symbol:

upper- or lowercase letter – $2 \cdot 26 = 52$ options

Other symbols:

upper- or lowercase letter or digit – $52 + 10 = 62$ options

Product rule: $P_6 = 52 \cdot 62^5$ $\qquad$ $P_7 = 52 \cdot 62^6$ $\qquad$ $P_8 = 52 \cdot 62^7$

$$N = 52 \cdot 62^5 + 52 \cdot 62^6 + 52 \cdot 62^7$$

OTHER USEFUL TRICKS

COUNTING ONE THING BY COUNTING ANOTHER

- How to count the number of people in the class?

COUNTING ONE THING BY COUNTING ANOTHER

- How to count the number of people in the class?
  - Count heads

COUNTING ONE THING BY COUNTING ANOTHER

- How to count the number of people in the class?
  - Count heads
  - Count their laptops

COUNTING ONE THING BY COUNTING ANOTHER

- How to count the number of people in the class?
  - Count heads
  - Count their laptops
  - Count ears and divide by 2 ☺
    (assuming that everyone has exactly two ears)

NUMBER OF HANDSHAKES

- There are n people in a room. Everyone shakes hands with everyone. How many handshakes are there?

NUMBER OF HANDSHAKES

- There are n people in a room. Everyone shakes hands with everyone. How many handshakes are there?

- Solution:

NUMBER OF HANDSHAKES

- There are n people in a room. Everyone shakes hands with everyone. How many handshakes are there?

- Solution:

  Every person (n) shakes hands with everyone else:

NUMBER OF HANDSHAKES

- There are n people in a room. Everyone shakes hands with everyone. How many handshakes are there?

- Solution:

  Every person (n) shakes hands with everyone else (n-1):

$$n(n-1) \text{ handshakes?}$$

NUMBER OF HANDSHAKES

- There are n people in a room. Everyone shakes hands with everyone. How many handshakes are there?

- Solution:

  Every person (n) shakes hands with everyone else (n-1):

$$n(n-1) \text{ handshakes?}$$

- We counted each handshake twice!

NUMBER OF HANDSHAKES

- There are n people in a room. Everyone shakes hands with everyone. How many handshakes are there?

- Solution:

  Every person (n) shakes hands with everyone else (n-1):

  $$n(n-1) \text{ handshakes?}$$

- We counted each handshake twice!

  Fixing it:

NUMBER OF HANDSHAKES

- There are n people in a room. Everyone shakes hands with everyone. How many handshakes are there?

- Solution:

  Every person (n) shakes hands with everyone else (n-1):

  $$n(n-1) \text{ handshakes?}$$

- We counted each handshake twice!

  Fixing it: $\frac{n(n-1)}{2}$

LET'S PRACTICE

https://docs.google.com/document/d/1XM0qoghJNkjnrVL3QVb_cSYRnfz1qdf7CLK6yFezF18/edit?usp=sharing

TO SUM UP

- The basics of set theory.
- Basic counting principles:
  - sum rule;
  - product rule;
  - combining the rules.
- Other tricks
  - using complement of a set;
  - overcounting.