Basic Counting Techniques

Vladimir Podolskii

Computer Science Department, Higher School of Economics

Outline

Why counting

Rule of Sum

Convenient Language: Sets

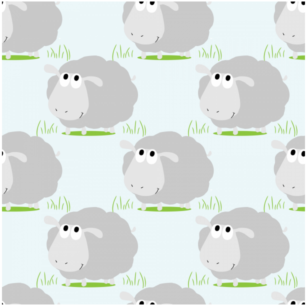
Generalizing Rule of Sum

Recursive Counting: Number of Paths

Rule of Product

Why Counting?

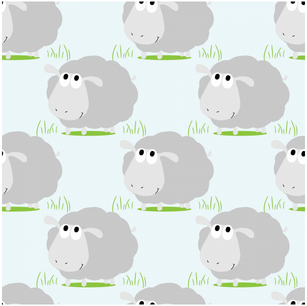
- Counting is one of the basic tasks related to mathematics



Needpix.com

Why Counting?

- Counting is one of the basic tasks related to mathematics
- Goal: to tell how many objects are there without actually counting them one by one



Needpix.com

Why Counting?

- Used a lot in other parts of mathematics and applications

Why Counting?

- Used a lot in other parts of mathematics and applications
- Important application: counting number of steps of algorithms

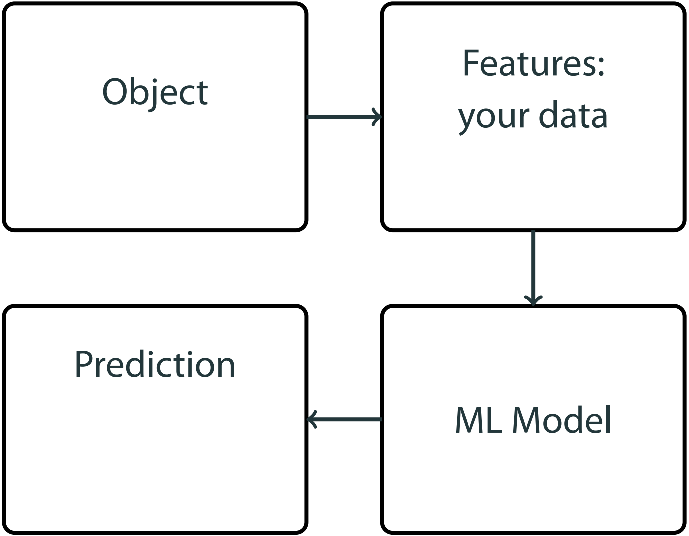
Why Counting?

- Used a lot in other parts of mathematics and applications
- Important application: counting number of steps of algorithms
- Important application: computing probabilities

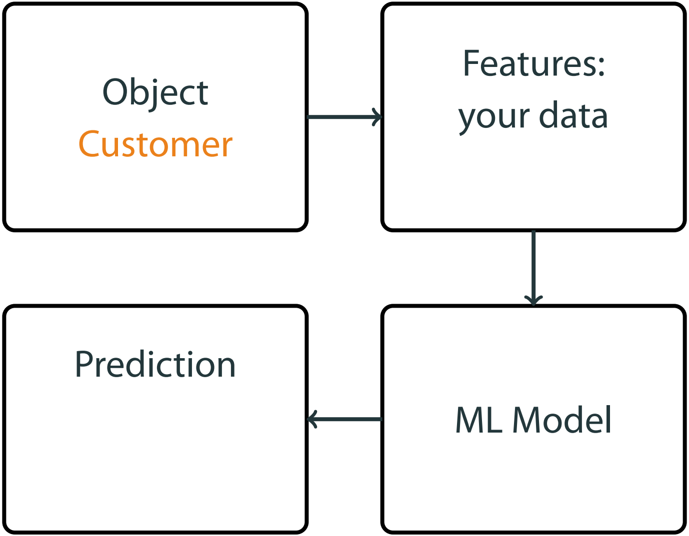
Why Counting?

- Used a lot in other parts of mathematics and applications
- Important application: counting number of steps of algorithms
- Important application: computing probabilities
- Important application: estimating the size of data

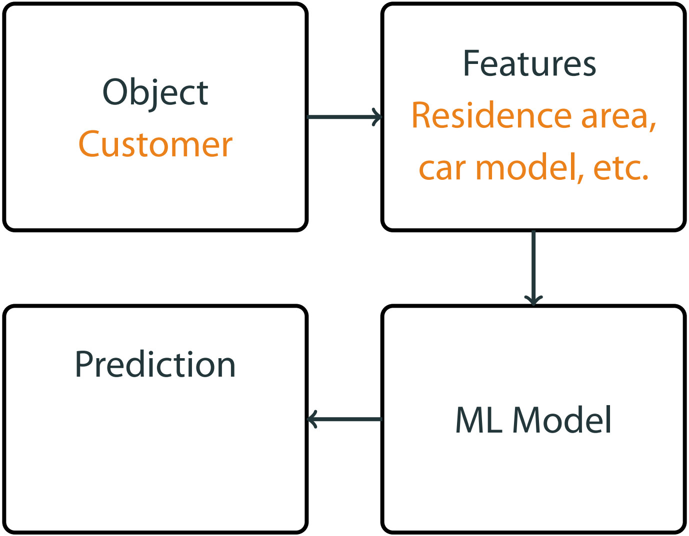
Counting in Data Analysis?



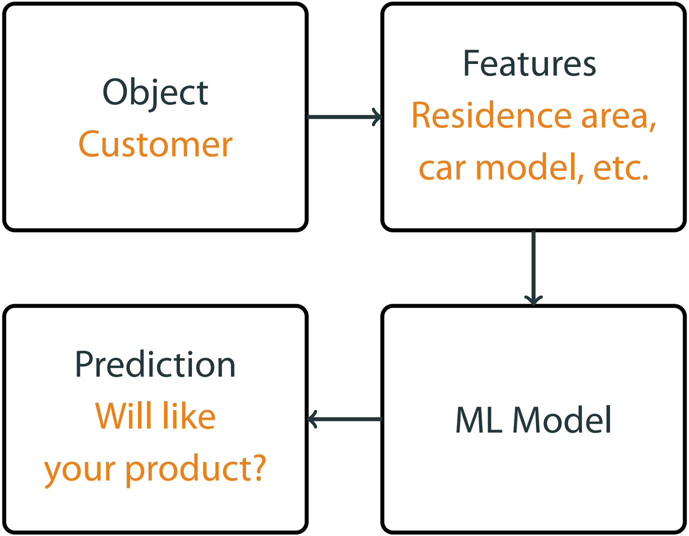
Counting in Data Analysis?



Counting in Data Analysis?



Counting in Data Analysis?



Counting in Data Analysis?

- Before you use your model it is important to teach it

Counting in Data Analysis?

- Before you use your model it is important to teach it
- How much data you need to teach?

Counting in Data Analysis?

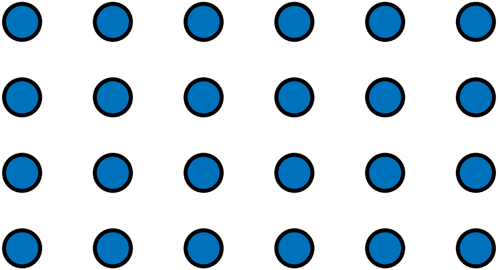
- Before you use your model it is important to teach it
- How much data you need to teach?
- In many models the amount of data needed is comparable to the number of all possible different objects in your feature setting

Counting in Data Analysis?

- Before you use your model it is important to teach it
- How much data you need to teach?
- In many models the amount of data needed is comparable to the number of all possible different objects in your feature setting
- It is useful to estimate the number of all possible collections of feature values

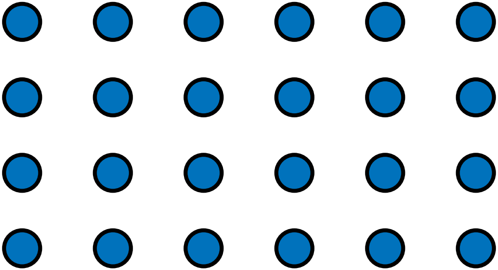
Why Study Counting?

- On the other hand, there are important ideas on how to count



Why Study Counting?

- On the other hand, there are important ideas on how to count
- Toy example: can you tell how many blue circles are in the picture without counting them by hand?



Real Life Example, Preview

wikimedia.org

- Suppose a country, a state or a region introduces a new format of a license plate

Real Life Example, Preview

wikimedia.org

- Suppose a country, a state or a region introduces a new format of a license plate
- Russian example: 3 digits, 3 letters; 78 is a regional code

Real Life Example, Preview

wikimedia.org

- Suppose a country, a state or a region introduces a new format of a license plate
- Russian example: 3 digits, 3 letters; 78 is a regional code
- We have: 10 options for digits, 12 options for letters (they must look similar to Latin letters; Cyrillic letters like Щ, Ю, Й are not allowed)

Real Life Example, Preview

wikimedia.org

- Suppose a country, a state or a region introduces a new format of a license plate
- Russian example: 3 digits, 3 letters; 78 is a regional code
- We have: 10 options for digits, 12 options for letters (they must look similar to Latin letters; Cyrillic letters like Щ, Ю, Й are not allowed)
- Will we have enough plates for everyone?

Outline

- Why counting
- Rule of Sum
- Convenient Language: Sets
- Generalizing Rule of Sum
- Recursive Counting: Number of Paths
- Rule of Product

Rule of Sum

Rule of Sum

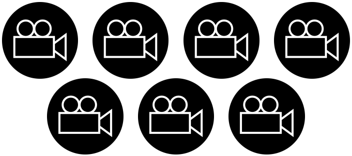
If there are k objects of the first type and there are n objects of the second type, then there are $n + k$ objects of one of two types

Rule of Sum

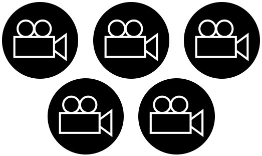
Rule of Sum

If there are k objects of the first type and there are n objects of the second type, then there are $n + k$ objects of one of two types

Videos $\geq$ 10 min



Videos $<$ 10 min

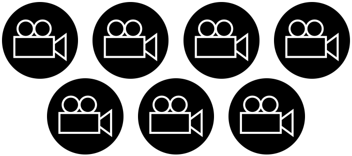


Rule of Sum

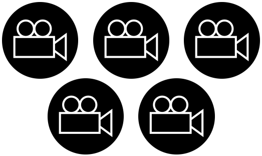
Rule of Sum

If there are k objects of the first type and there are n objects of the second type, then there are $n + k$ objects of one of two types

Videos $\geq$ 10 min



Videos $<$ 10 min

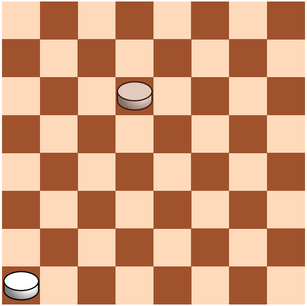


7+5=12 videos in total

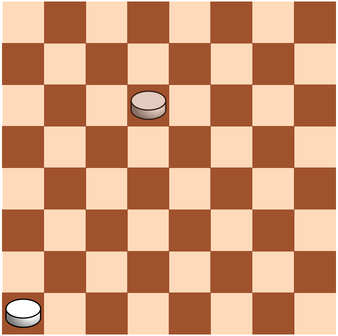
Piece on a Chessboard

Piece on a chessboard

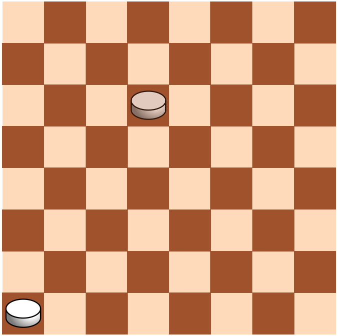
A piece stays in the bottom left corner of a chessboard. In one move it can move one step to the right or one step up. How many moves are needed to get to the position on the picture?



Piece on a Chessboard

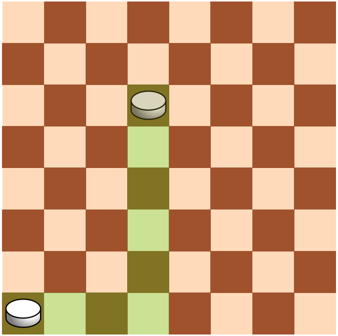


Piece on a Chessboard



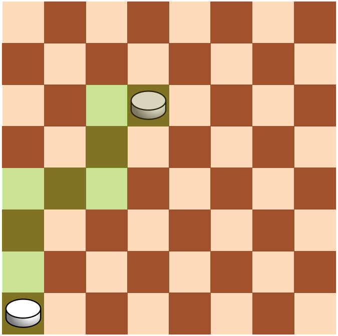
- We can do it in several ways

Piece on a Chessboard



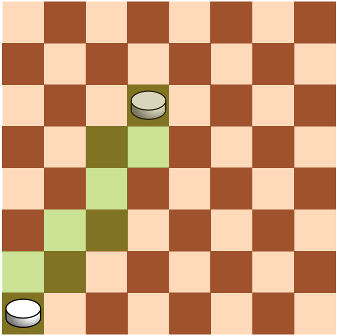
- We can do it in several ways

Piece on a Chessboard



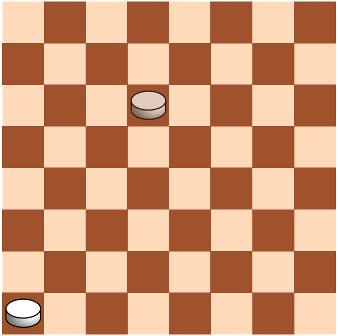
- We can do it in several ways

Piece on a Chessboard



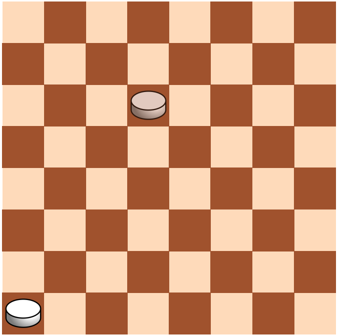
- We can do it in several ways

Piece on a Chessboard



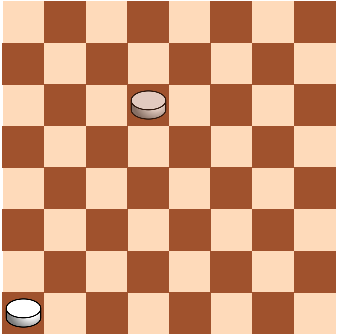
- We can do it in several ways
- In all cases we need 8 moves

Piece on a Chessboard



- We can do it in several ways
- In all cases we need 8 moves
- This is not a coincidence

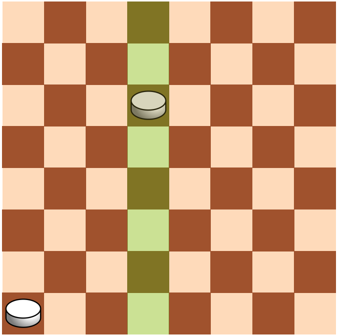
Piece on a Chessboard



- We can do it in several ways
- In all cases we need 8 moves
- This is not a coincidence

1. There are two types of moves: move right and move up

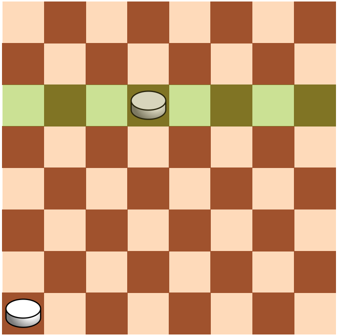
Piece on a Chessboard



- We can do it in several ways
- In all cases we need 8 moves
- This is not a coincidence

1. There are two types of moves: move right and move up
2. To get to the column 4 we need 3 moves to the right

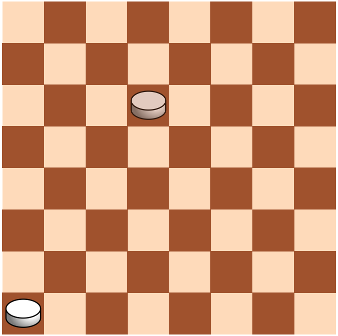
Piece on a Chessboard



- We can do it in several ways
- In all cases we need 8 moves
- This is not a coincidence

1. There are two types of moves: move right and move up
2. To get to the column 4 we need 3 moves to the right
3. To get to the row 6 we need 5 moves up

Piece on a Chessboard



- We can do it in several ways
- In all cases we need 8 moves
- This is not a coincidence

1. There are two types of moves: move right and move up
2. To get to the column 4 we need 3 moves to the right
3. To get to the row 6 we need 5 moves up
4. In total we need 3+5=8 moves

One More Example

- Suppose in our data we have 7 videos of length ≥ 4 minutes and 5 music videos

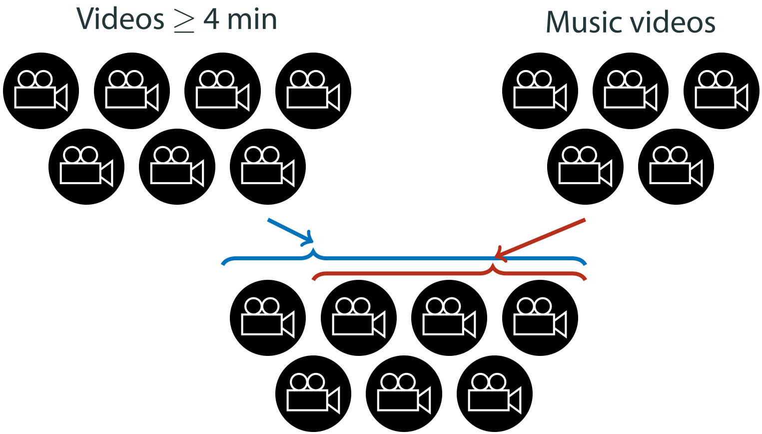
One More Example

- Suppose in our data we have 7 videos of length ≥ 4 minutes and 5 music videos
- Can we say that there are 12 videos that are either longer than 4 minutes or are music videos?

One More Example

- Suppose in our data we have 7 videos of length ≥ 4 minutes and 5 music videos
- Can we say that there are 12 videos that are either longer than 4 minutes or are music videos?
- What if some of our music videos are longer than 4 minutes?

One More Example



Rule of sum does not work here!

More Subtle Example

- Suppose your video service categorize videos: music, comedy, gardening, cats, etc.

More Subtle Example

- Suppose your video service categorize videos: music, comedy, gardening, cats, etc.
- Suppose in your data there are 7 videos of one category and 5 videos of another category

More Subtle Example

- Suppose your video service categorize videos: music, comedy, gardening, cats, etc.
- Suppose in your data there are 7 videos of one category and 5 videos of another category
- Do you have 12 videos that fall in one of the two categories?

More Subtle Example

- Suppose your video service categorize videos: music, comedy, gardening, cats, etc.
- Suppose in your data there are 7 videos of one category and 5 videos of another category
- Do you have 12 videos that fall in one of the two categories?
- Not necessary! Categories might overlap

More Subtle Example

- Suppose your video service categorize videos: music, comedy, gardening, cats, etc.
- Suppose in your data there are 7 videos of one category and 5 videos of another category
- Do you have 12 videos that fall in one of the two categories?
- Not necessary! Categories might overlap
- For example, there might be funny videos with cats

Rule of Sum Revisited

Rule of Sum

If there are k objects of the first type and there are n objects of the second type, then there are $n + k$ objects of one of two types

- Important lesson: in the rule of sum no object should belong to both classes!

Outline

Why counting

Rule of Sum

Convenient Language: Sets

Generalizing Rule of Sum

Recursive Counting: Number of Paths

Rule of Product

Convenient Language: Sets

- Set is an arbitrary group of arbitrary objects

Convenient Language: Sets

- Set is an arbitrary group of arbitrary objects
- We will denote sets by capital letters: A, B, C, S, etc.

Convenient Language: Sets

- Set is an arbitrary group of arbitrary objects
- We will denote sets by capital letters: A, B, C, S, etc.
- Sets can be given by the list of their elements: $S = \{0, 1, 2, 3\}$; this set consists of four elements, 0, 1, 2, 3

Convenient Language: Sets

- Set is an arbitrary group of arbitrary objects
- We will denote sets by capital letters: A, B, C, S, etc.
- Sets can be given by the list of their elements: $S = \{0, 1, 2, 3\}$; this set consists of four elements, 0, 1, 2, 3
- The order of elements is not important: $\{0, 1, 2, 3\} = \{2, 0, 3, 1\}$

Convenient Language: Sets

- Set is an arbitrary group of arbitrary objects
- We will denote sets by capital letters: A, B, C, S, etc.
- Sets can be given by the list of their elements: $S = \{0, 1, 2, 3\}$; this set consists of four elements, 0, 1, 2, 3
- The order of elements is not important: $\{0, 1, 2, 3\} = \{2, 0, 3, 1\}$
- Repetitions in the list of elements are not important: $\{0, 1, 2, 3\} = \{1, 0, 1, 3, 2, 3\}$

Convenient Language: Sets

- For us sets provide a convenient language

Convenient Language: Sets

- For us sets provide a convenient language
- In Mathematics sets play an important and fundamental role

Convenient Language: Sets

- For us sets provide a convenient language
- In Mathematics sets play an important and fundamental role
- For us sets can consist of anything:
  $S = \{0, \sqrt{2}, \text{ Isaac Newton}, \text{ a leprechaun}\}$

Convenient Language: Sets

- For us sets provide a convenient language
- In Mathematics sets play an important and fundamental role
- For us sets can consist of anything:
  $S = \{0, \sqrt{2}, \text{ Isaac Newton}, \text{ a leprechaun}\}$
- However, there are pitfalls

Convenient Language: Sets

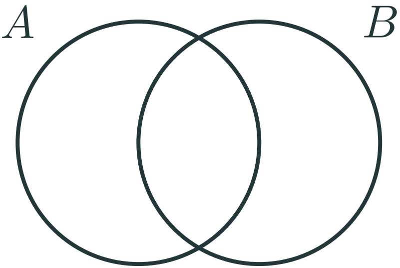
- For us sets provide a convenient language
- In Mathematics sets play an important and fundamental role
- For us sets can consist of anything:
  $S = \{0, \sqrt{2}, \text{ Isaac Newton}, \text{ a leprechaun}\}$
- However, there are pitfalls
- “Set consisting of all sets” is a dangerous construction

Convenient Language: Sets

- For us sets provide a convenient language
- In Mathematics sets play an important and fundamental role
- For us sets can consist of anything:
  $S = \{0, \sqrt{2}, \text{ Isaac Newton}, \text{ a leprechaun}\}$
- However, there are pitfalls
- “Set consisting of all sets” is a dangerous construction
- We will not encounter these difficulties in our program and will not discuss them

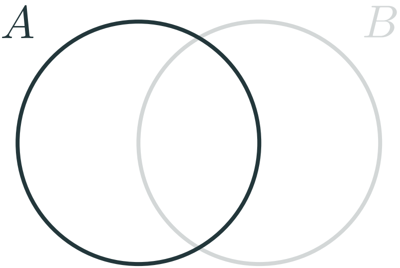
Venn Diagrams

- A convenient way to view sets is by the means of Venn Diagrams



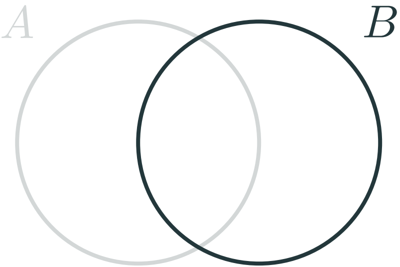
Venn Diagrams

- A convenient way to view sets is by the means of Venn Diagrams
- Elements of A are within the left circle



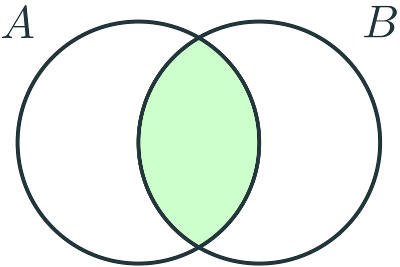
Venn Diagrams

- A convenient way to view sets is by the means of Venn Diagrams
- Elements of A are within the left circle
- Elements of B are within the right circle

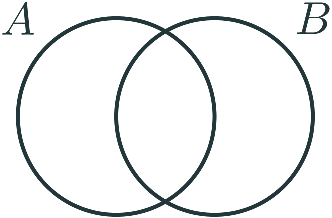


Venn Diagrams

- A convenient way to view sets is by the means of Venn Diagrams
- Elements of A are within the left circle
- Elements of B are within the right circle
- Intersection corresponds to elements belonging to both sets

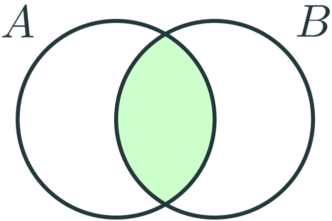


Useful Notations



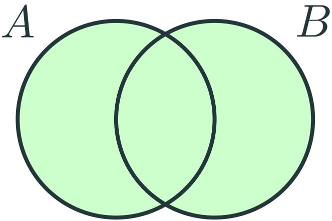
- Suppose we have two sets A and B

Useful Notations



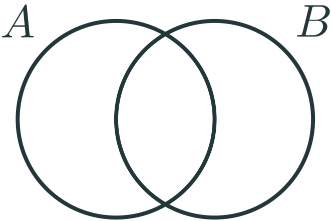
- Suppose we have two sets A and B
- The set $A \cap B$ is an intersection of these sets: it consists of elements belonging to both sets

Useful Notations



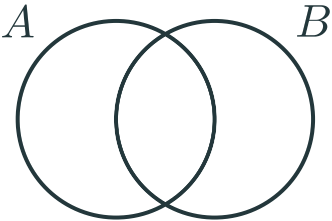
- Suppose we have two sets A and B
- The set $A \cap B$ is an intersection of these sets: it consists of elements belonging to both sets
- The set $A \cup B$ is a union of these sets: it consists of elements belonging to at least one of the sets

Useful Notations



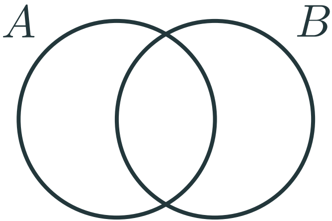
- Suppose we have two sets A and B
- The set $A \cap B$ is an intersection of these sets: it consists of elements belonging to both sets
- The set $A \cup B$ is a union of these sets: it consists of elements belonging to at least one of the sets
- If every element of A is also an element of B, then A is a subset of B; we write $A \subseteq B$

Useful Notations



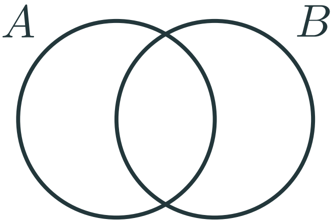
- If some object x is an element of A we write $x \in A$

Useful Notations



- If some object x is an element of A we write $x \in A$
- The number of elements in the set A is $|A|$ (can be infinite)

Useful Notations



- If some object x is an element of A we write $x \in A$
- The number of elements in the set A is $|A|$ (can be infinite)
- A set without elements is denoted by $\emptyset$ and is called empty set

Outline

Why counting

Rule of Sum

Convenient Language: Sets

Generalizing Rule of Sum

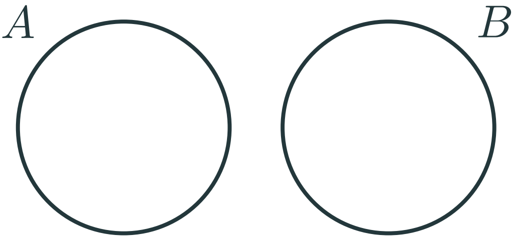
Recursive Counting: Number of Paths

Rule of Product

Rule of Sum in the Set Language

Rule of Sum

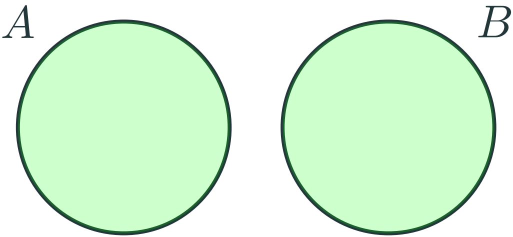
If there is a set A with k elements, a set B with n elements and these sets do not have common elements, then the set $A \cup B$ has $n + k$ elements



Rule of Sum in the Set Language

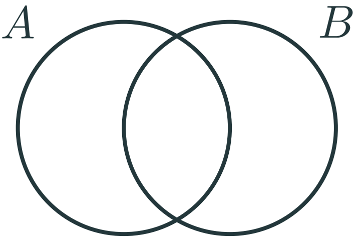
Rule of Sum

If there is a set A with k elements, a set B with n elements and these sets do not have common elements, then the set $A \cup B$ has $n + k$ elements



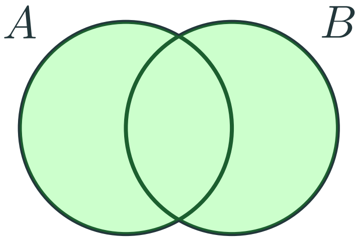
Generalized Rule of Sum

But what if we want to count $|A \cup B|$ in the setting below?

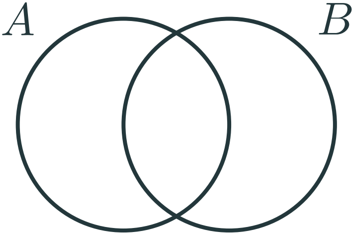


Generalized Rule of Sum

But what if we want to count $|A \cup B|$ in the setting below?

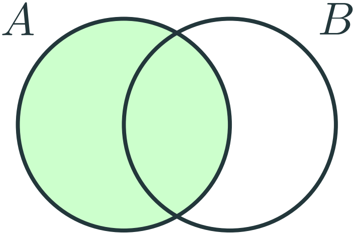


Generalized Rule of Sum



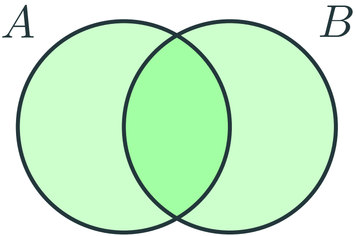
- If we just consider $|A| + |B|$ as in the rule of sum, then we will be wrong

Generalized Rule of Sum



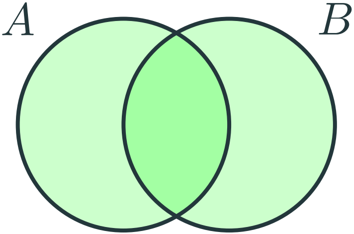
- If we just consider $|A| + |B|$ as in the rule of sum, then we will be wrong

Generalized Rule of Sum



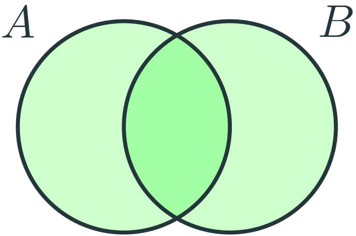
- If we just consider $|A| + |B|$ as in the rule of sum, then we will be wrong

Generalized Rule of Sum



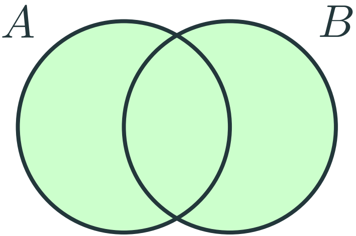
- If we just consider $|A| + |B|$ as in the rule of sum, then we will be wrong
- We count elements that belong to both A and B twice

Generalized Rule of Sum



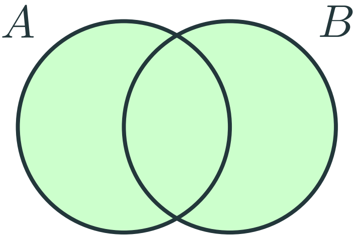
- If we just consider $|A| + |B|$ as in the rule of sum, then we will be wrong
- We count elements that belong to both A and B twice
- So let's subtract them now!

Generalized Rule of Sum



- If we just consider $|A| + |B|$ as in the rule of sum, then we will be wrong
- We count elements that belong to both A and B twice
- So let's subtract them now!

Generalized Rule of Sum



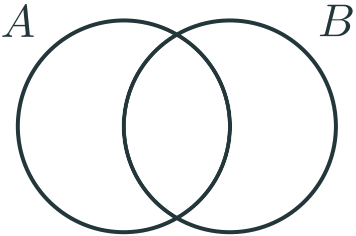
- If we just consider $|A| + |B|$ as in the rule of sum, then we will be wrong
- We count elements that belong to both A and B twice
- So let's subtract them now!
- This gives the right result: $|A \cup B| = |A| + |B| - |A \cap B|$

Generalized Rule of Sum

Rule of Sum

If there are finite sets A and B, then

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



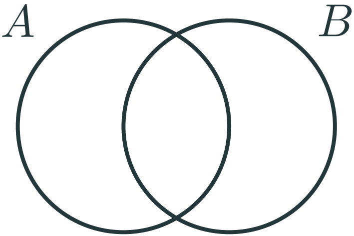
Generalized Rule of Sum

Rule of Sum

If there are finite sets A and B, then

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

Covers the original rule: $|A \cap B| = 0$

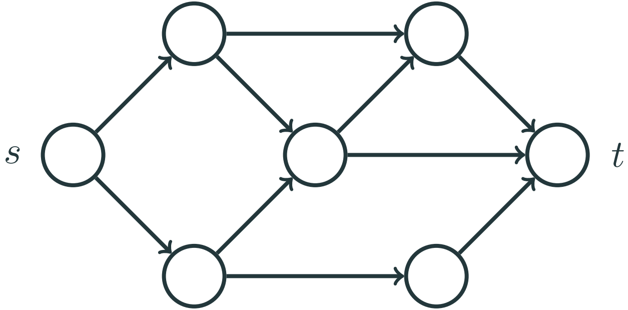


Outline

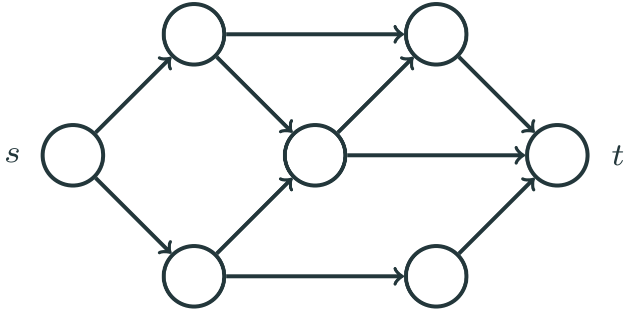
Recursive Counting: Number of Paths

Problem

Suppose there are several points connected by arrows. There is a starting point s (called source) and a final point t (called sink). How many different ways are there to get from s to t?

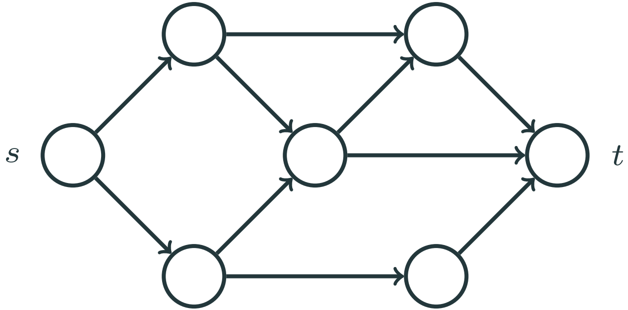


Number of Paths



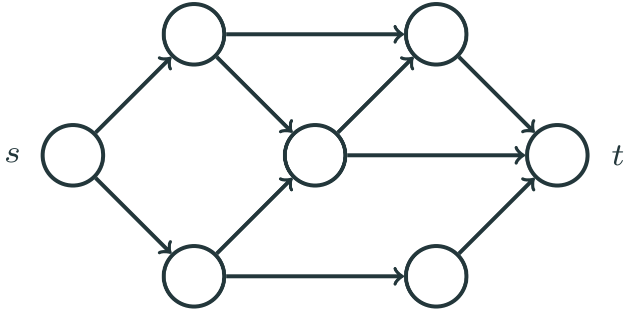
- There are several various paths; how not to miss anything when counting?

Number of Paths



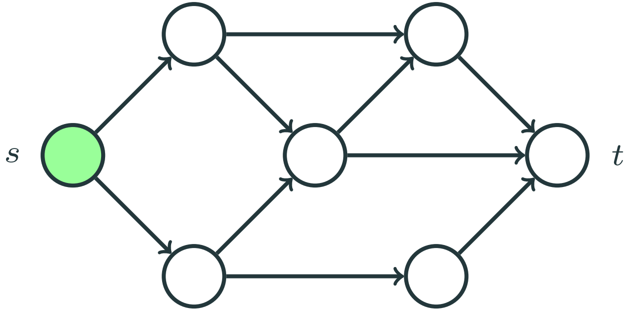
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node

Number of Paths



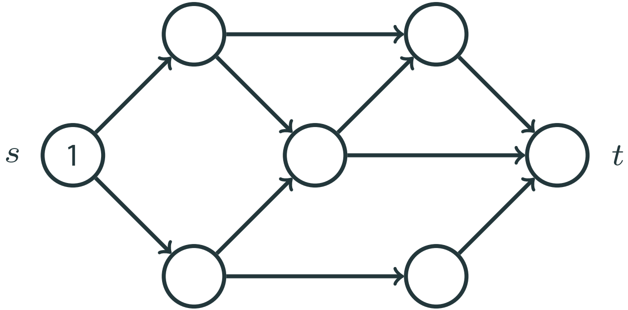
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



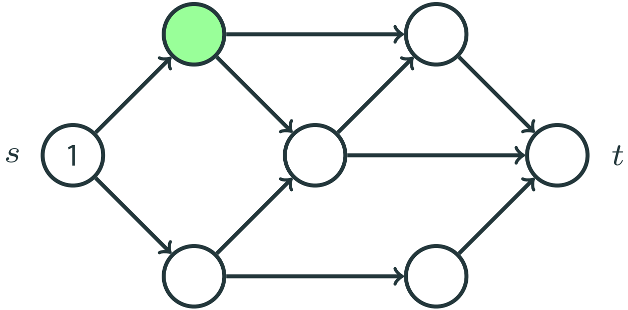
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



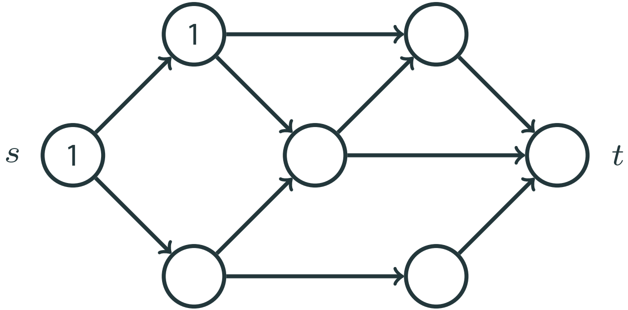
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



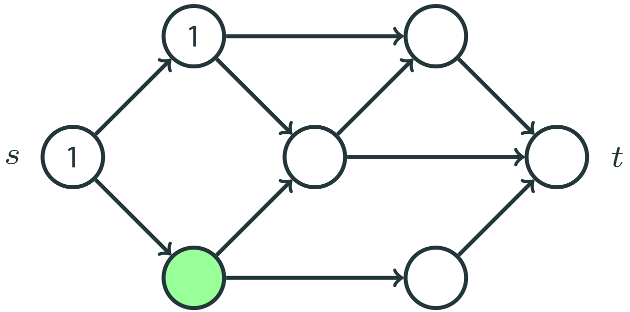
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



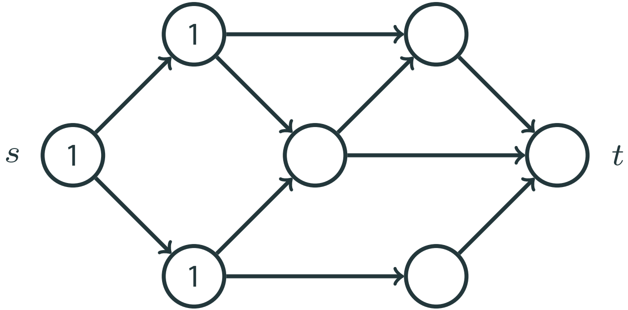
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



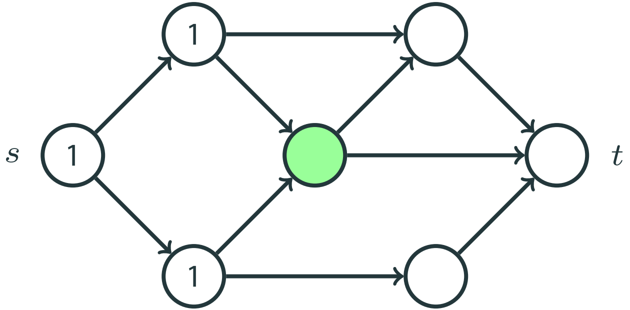
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



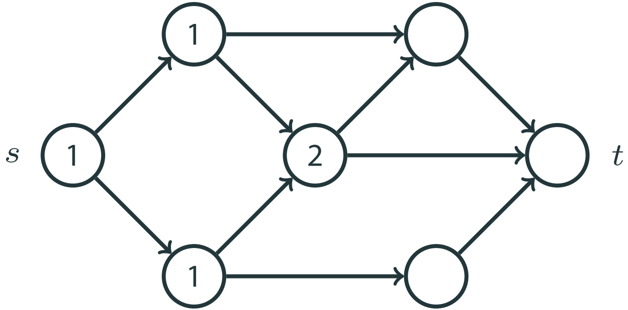
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



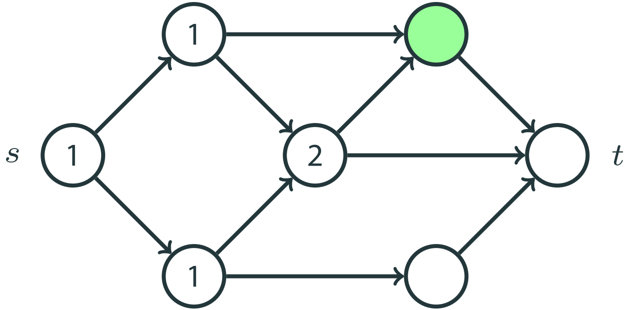
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



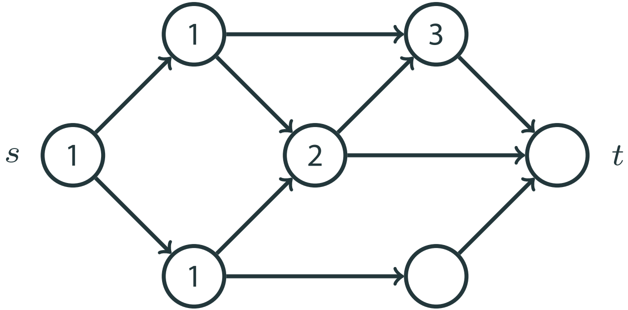
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



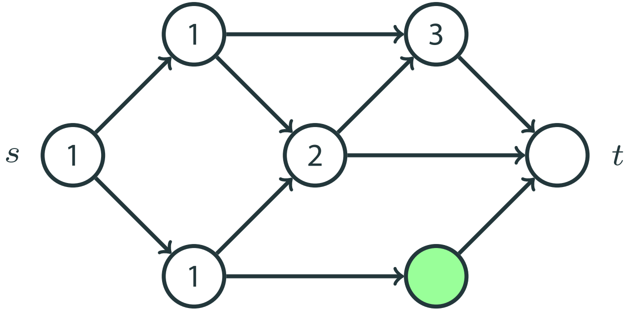
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



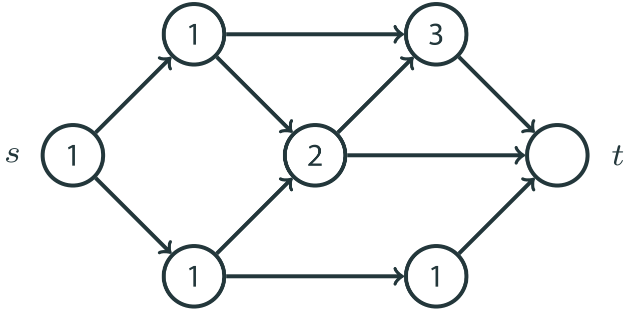
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



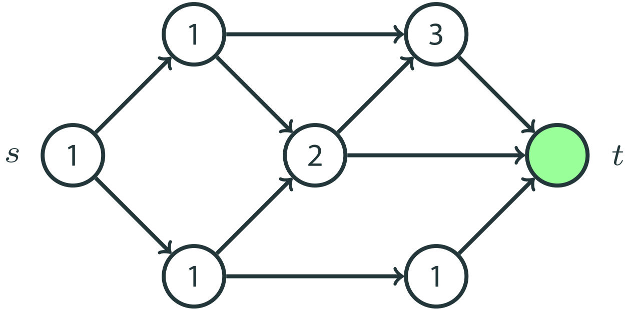
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



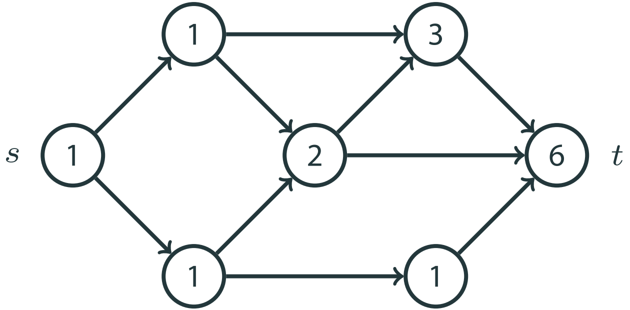
- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Number of Paths



- There are several various paths; how not to miss anything when counting?
- We can count them recursively: for each node count the number of paths from s to this node
- We use the rule of sum!

Outline

Why counting

Rule of Sum

Convenient Language: Sets

Generalizing Rule of Sum

Recursive Counting: Number of Paths

Rule of Product

Rule of Product

Rule of Product

If there are k object of the first type and there are n object of the second type, then there are $k \times n$ pairs of objects, the first of the first type and the second of the second type

Rule of Product

Rule of Product

If there are k object of the first type and there are n object of the second type, then there are $k \times n$ pairs of objects, the first of the first type and the second of the second type

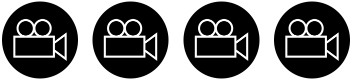
Videos Persons

Rule of Product

Rule of Product

If there are k object of the first type and there are n object of the second type, then there are $k \times n$ pairs of objects, the first of the first type and the second of the second type

Videos



Persons

$4 \times 3 = 12$ evaluations

List of All Evaluations

	5	3	2	7
	8	7	5	1
	1	7	8	10

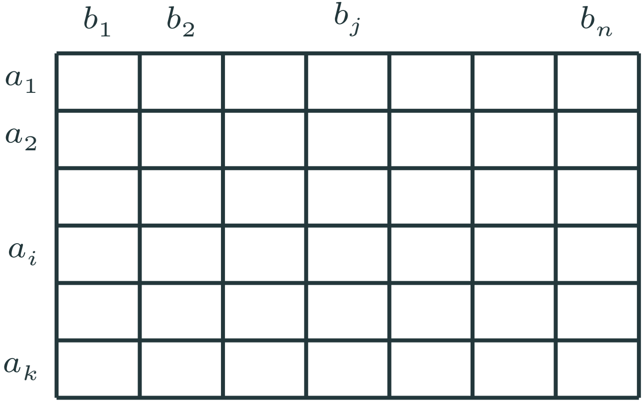
Rule of Product in the Set Language

Rule of Product

If there is a finite set A and a finite set B, then there are $|A| \times |B|$ pairs of objects, the first from A and the second from B

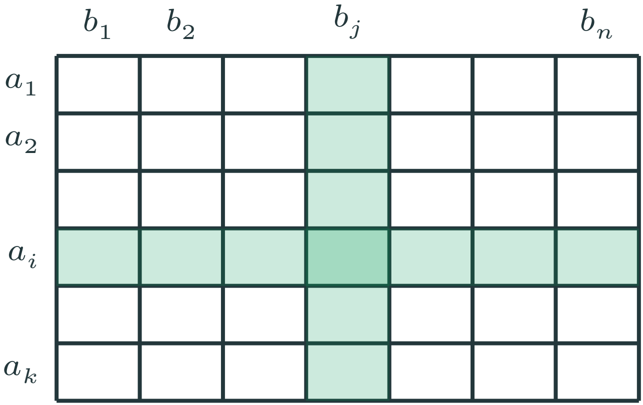
Why the Rule of Product is True?

$$A = \{a_1, \dots, a_k\}$$
$$B = \{b_1, \dots, b_n\}$$



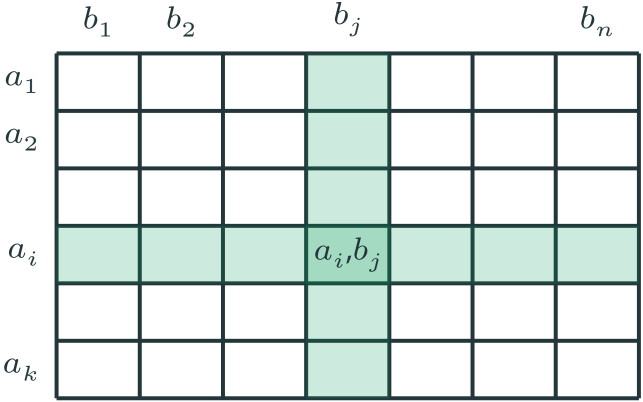
Why the Rule of Product is True?

$A = \{a_1, \dots, a_k\}$
$B = \{b_1, \dots, b_n\}$



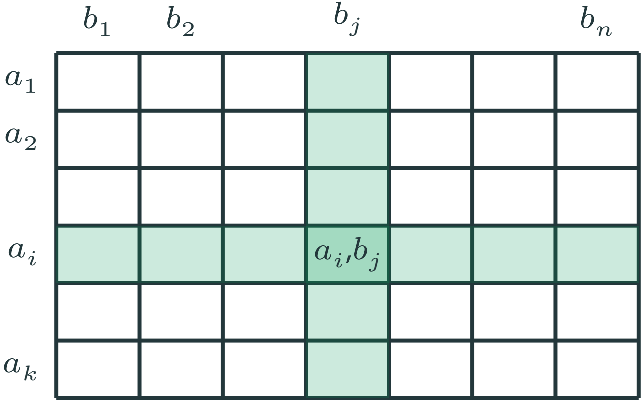
Why the Rule of Product is True?

$A = \{a_1, \ldots, a_k\}$
$B = \{b_1, \ldots, b_n\}$



Why the Rule of Product is True?

$A = \{a_1, \ldots, a_k\}$
$B = \{b_1, \ldots, b_n\}$



There are as many pairs as cells in this table

Rule of Product as the Number of Paths

Rule of Product

If there is a finite set A and a finite set B, then there are $|A| \times |B|$ pairs of objects, the first from A and the second from B

Rule of Product as the Number of Paths

Rule of Product

If there is a finite set A and a finite set B, then there are $|A| \times |B|$ pairs of objects, the first from A and the second from B

Can we express this counting rule in terms of counting the number of paths?

Rule of Product as the Number of Paths

Rule of Product

If there is a finite set A and a finite set B, then there are $|A| \times |B|$ pairs of objects, the first from A and the second from B

Rule of Product as the Number of Paths

Rule of Product

If there is a finite set A and a finite set B, then there are $|A| \times |B|$ pairs of objects, the first from A and the second from B

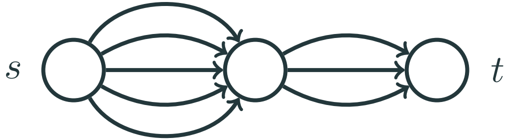
Consider for simplicity $|A| = 5$ and $|B| = 3$

Rule of Product as the Number of Paths

Rule of Product

If there is a finite set A and a finite set B, then there are $|A| \times |B|$ pairs of objects, the first from A and the second from B

Consider for simplicity $|A| = 5$ and $|B| = 3$

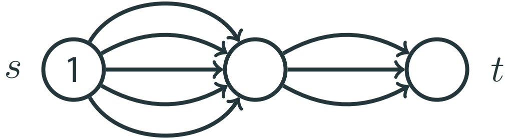


Rule of Product as the Number of Paths

Rule of Product

If there is a finite set A and a finite set B, then there are $|A| \times |B|$ pairs of objects, the first from A and the second from B

Consider for simplicity $|A| = 5$ and $|B| = 3$

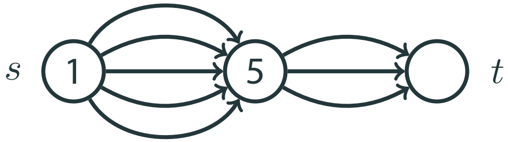


Rule of Product as the Number of Paths

Rule of Product

If there is a finite set A and a finite set B, then there are $|A| \times |B|$ pairs of objects, the first from A and the second from B

Consider for simplicity $|A| = 5$ and $|B| = 3$



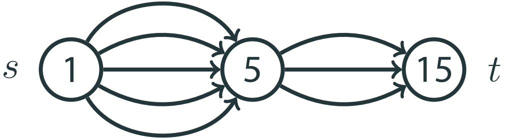
$$1 + 1 + 1 + 1 + 1 = 5$$

Rule of Product as the Number of Paths

Rule of Product

If there is a finite set A and a finite set B, then there are $|A| \times |B|$ pairs of objects, the first from A and the second from B

Consider for simplicity $|A| = 5$ and $|B| = 3$



$$5 + 5 + 5 = 3 \times 5 = 15$$

Summary

- Counting starts with simple things

Summary

- Counting starts with simple things
- But even rule of sum can be tricky

Summary

- Counting starts with simple things
- But even rule of sum can be tricky
- Some simple basic ideas can already be helpful

Summary

- Counting starts with simple things
- But even rule of sum can be tricky
- Some simple basic ideas can already be helpful
- Next we will see how to build something more involved from the basic building blocks