Graph Theory and Combinatorics

Lecture 1: Introduction.
Binomial and Multinomial Numbers

Isabela Drămnesc UVT

Computer Science Department,
West University of Timișoara,
Romania

October 1, 2018
Purpose of this lecture

Become familiar with the basic notions from combinatorics and graph theory.

2. Principle of inclusion and exclusion, enumeration techniques.
3. Combinations
4. The cyclic structure of permutations. Advanced counting techniques.
5. Polya’s theory of counting
6. Graph theory: basic notions
7. Data structures for the representation of graphs
8. Transport networks, maximal flows, minimal cuts
9. Trees: definitions; generating trees; minimum cost trees
10. Paths, circuits, chains, and cycles
11. The traveling salesman problem. Planar graphs
12. Chromatic theory of graphs
13. Matchings
Organizational items

- Lecturer and TA: Isabela Drămnesc
- Course webpage: staff.fmi.uvt.ro/~isabela.dramnesc
  - Exercises
  - Seminar/Lab: working with Combinatorica in Mathematica
- Handouts: will be posted on the webpage of the lecture
- Grading:
  - 50% : weekly seminar assignments and a partial exam (week 8)
  - 50% : 1 written exam at the end of the semester
Basic counting principles
- The product rule
- The sum rule
- Combinatorial proofs; examples

Counting techniques for
- combinations - unordered selections of distinct elements of a finite set
- permutations - ordered selections of distinct elements of a finite set

Generalizations
- permutations with repetition
- combinations with repetition
- permutations with indistinguishable elements

Binomial and multinomial numbers
**Product rule.** If a procedure can be broken down into a sequence of two tasks, such that

- first task can be done in $n_1$ ways
- second task can be done in $n_2$ ways

then there are $n_1 \cdot n_2$ ways to do the procedure.
Basic counting principles

1. The product rule

**Product rule.** If a procedure can be broken down into a sequence of two tasks, such that

- first task can be done in \( n_1 \) ways
- second task can be done in \( n_2 \) ways

then there are \( n_1 \cdot n_2 \) ways to do the procedure.

**Generalized product rule.** If a procedure can be broken down into a sequence of \( m \) tasks, such that

- first task can be done in \( n_1 \) ways
- second task can be done in \( n_2 \) ways
- \( \ldots \)
- \( m \)-th task can be done in \( n_m \) ways

then there are \( n_1 \cdot n_2 \cdot \ldots \cdot n_m \) ways to do the procedure.
A new company with just 2 employees, John and Wayne, rents a floor of a building with 12 offices. How many ways are there to assign different offices to these two employees?
(1) A new company with just 2 employees, John and Wayne, rents a floor of a building with 12 offices. How many ways are there to assign different offices to these two employees?

Answer
(1) A new company with just 2 employees, John and Wayne, rents a floor of a building with 12 offices. How many ways are there to assign different offices to these two employees?

**Answer**

- The task can be broken down into a sequence of 2 tasks: choosing an office for John, followed by choosing an office for Wayne.
(1) A new company with just 2 employees, John and Wayne, rents a floor of a building with 12 offices. How many ways are there to assign different offices to these two employees?

**Answer**

- The task can be broken down into a sequence of 2 tasks: choosing an office for John, followed by choosing an office for Wayne.
- There are 12 ways to choose an office for John, because there are 12 offices available.
(1) A new company with just 2 employees, John and Wayne, rents a floor of a building with 12 offices. How many ways are there to assign different offices to these two employees?

**Answer**

- The task can be broken down into a sequence of 2 tasks: choosing an office for John, followed by choosing an office for Wayne.
- There are 12 ways to choose an office for John, because there are 12 offices available.
- There are 11 choices for the office of Wayne, because only John’s office is unavailable.
Applications of the product rule

(1) A new company with just 2 employees, John and Wayne, rents a floor of a building with 12 offices. How many ways are there to assign different offices to these two employees?

**Answer**

- The task can be broken down into a sequence of 2 tasks: choosing an office for John, followed by choosing an office for Wayne.
- There are 12 ways to choose an office for John, because there are 12 offices available.
- There are 11 choices for the office of Wayne, because only John’s office is unavailable.

⇒ by the **product rule**, there are $12 \cdot 11 = 132$ ways.
(2) The seats of a lab room are to be labelled with a letter and a number not exceeding 25. What is the largest number of chairs that can be labeled differently?
(2) The seats of a lab room are to be labelled with a letter and a number not exceeding 25. What is the largest number of chairs that can be labeled differently?

**Answer**
(2) The seats of a lab room are to be labelled with a letter and a number not exceeding 25. What is the largest number of chairs that can be labeled differently?

**Answer**

- This task can be broken down into a sequence of 2 tasks: assign one of the 26 letters, followed by assigning one of the 25 possible numbers to the seat.

\[
\text{According to the product rule, there are } 26 \cdot 25 = 650 \text{ ways to do so.}
\]
(2) The seats of a lab room are to be labelled with a letter and a number not exceeding 25. What is the largest number of chairs that can be labeled differently?

**Answer**

- This task can be broken down into a sequence of 2 tasks: assign one of the 26 letters, followed by assigning one of the 25 possible numbers to the seat.
- According to the product rule, there are $26 \cdot 25 = 650$ ways to do so.
(2) The seats of a lab room are to be labelled with a letter and a number not exceeding 25. What is the largest number of chairs that can be labeled differently?

**Answer**
- This task can be broken down into a sequence of 2 tasks: assign one of the 26 letters, followed by assigning one of the 25 possible numbers to the seat.
- According to the product rule, there are $26 \cdot 25 = 650$ ways to do so.

(3) How many different bit strings of length 7 are there?
(2) The seats of a lab room are to be labelled with a letter and a number not exceeding 25. What is the largest number of chairs that can be labeled differently?

**Answer**

- This task can be broken down into a sequence of 2 tasks: assign one of the 26 letters, followed by assigning one of the 25 possible numbers to the seat.
- According to the product rule, there are $26 \cdot 25 = 650$ ways to do so.

(3) How many different bit strings of length 7 are there?

**Answer**
(2) The seats of a lab room are to be labelled with a letter and a number not exceeding 25. What is the largest number of chairs that can be labeled differently?

**Answer**

- This task can be broken down into a sequence of 2 tasks: assign one of the 26 letter, followed by assigning one of the 25 possible numbers to the seat.
- According to the product rule, there are $26 \cdot 25 = 650$ ways to do so.

(3) How many different bit strings of length 7 are there?

**Answer**

- Each of the 7 bits can be chosen in 2 ways, because each bit is either 0 or 1.
Applications of the product rule

(2) The seats of a lab room are to be labelled with a letter and a number not exceeding 25. What is the largest number of chairs that can be labeled differently?

**Answer**

- This task can be broken down into a sequence of 2 tasks: assign one of the 26 letter, followed by assigning one of the 25 possible numbers to the seat.
- According to the product rule, there are $26 \cdot 25 = 650$ ways to do so.

(3) How many different bit strings of length 7 are there?

**Answer**

- Each of the 7 bits can be chosen in 2 ways, because each bit is either 0 or 1.
- $\Rightarrow$ by the product rule, there are $2^7 = 128$ ways.
(4) How many functions are there from a set with \( m \) elements to a set with \( n \) elements?

**Answer**

- The procedure to define such a function can be broken down into a sequence of \( m \) subtasks, where each subtask fixes the output value for an input argument.
- Each subtask can be done in \( n \) ways (there are \( n \) possible output values)

\[ \Rightarrow \text{by product rule, the number of functions is } n \cdot \ldots \cdot n = n^m \]

\( m \) times
(5) How many one-to-one functions are there from a set with \( m \) elements to one with \( n \) elements?
(5) How many one-to-one functions are there from a set with $m$ elements to one with $n$ elements?

**Remark.** A one-to-one function is a function that maps different elements to different values.
(5) How many one-to-one functions are there from a set with \( m \) elements to one with \( n \) elements?

**Remark.** A one-to-one function is a function that maps different elements to different values.

**Answer:** assume \( f : \{a_1, \ldots, a_m\} \rightarrow \{b_1, \ldots, b_n\} \)
How many one-to-one functions are there from a set with \( m \) elements to one with \( n \) elements?

**Remark.** A one-to-one function is a function that maps different elements to different values.

**Answer:** assume \( f : \{a_1, \ldots, a_m\} \rightarrow \{b_1, \ldots, b_n\} \)
- Note that we must have \( m \leq n \).
(5) How many one-to-one functions are there from a set with \( m \) elements to one with \( n \) elements?

**Remark.** A one-to-one function is a function that maps different elements to different values.

**Answer:** assume \( f : \{a_1, \ldots, a_m\} \to \{b_1, \ldots, b_n\} \)

- Note that we must have \( m \leq n \).
- There are \( n \) ways to pick a value for \( f(a_1) \in \{b_1, \ldots, b_m\} \).
(5) How many one-to-one functions are there from a set with $m$ elements to one with $n$ elements?

**Remark.** A one-to-one function is a function that maps different elements to different values.

**Answer:** assume $f : \{a_1, \ldots, a_m\} \rightarrow \{b_1, \ldots, b_n\}$

- Note that we must have $m \leq n$.
- There are $n$ ways to pick a value for $f(a_1) \in \{b_1, \ldots, b_m\}$.
- There are $n - 1$ ways to pick a value for $f(a_2) \in \{b_1, \ldots, b_m\} - \{f(a_1)\}$
(5) How many one-to-one functions are there from a set with \( m \) elements to one with \( n \) elements?

**Remark.** A one-to-one function is a function that maps different elements to different values.

**Answer:** assume \( f : \{a_1, \ldots, a_m\} \rightarrow \{b_1, \ldots, b_n\} \)

- Note that we must have \( m \leq n \).
- There are \( n \) ways to pick a value for \( f(a_1) \in \{b_1, \ldots, b_m\} \).
- There are \( n - 1 \) ways to pick a value for \( f(a_2) \in \{b_1, \ldots, b_m\} - \{f(a_1)\} \).
How many one-to-one functions are there from a set with \( m \) elements to one with \( n \) elements?

**Remark.** A one-to-one function is a function that maps different elements to different values.

**Answer:** assume \( f : \{a_1, \ldots, a_m\} \rightarrow \{b_1, \ldots, b_n\} \)

- Note that we must have \( m \leq n \).
- There are \( n \) ways to pick a value for \( f(a_1) \in \{b_1, \ldots, b_m\} \).
- There are \( n-1 \) ways to pick a value for \( f(a_2) \in \{b_1, \ldots, b_m\} - \{f(a_1)\} \).
- ... 
- There are \( n - m + 1 \) ways to pick the function value for \( f(a_m) \in \{b_1, \ldots, b_m\} - \{f(a_1), \ldots, f(a_{m-1})\} \).

\[ \Rightarrow \] By the product rule, there are \( n \cdot (n-1) \cdot \ldots \cdot (n - m + 1) \) one-to-one functions.
How many one-to-one functions are there from a set with $m$ elements to one with $n$ elements?

**Remark.** A one-to-one function is a function that maps different elements to different values.

**Answer:** assume $f : \{a_1, \ldots, a_m\} \rightarrow \{b_1, \ldots, b_n\}$

- Note that we must have $m \leq n$.
- There are $n$ ways to pick a value for $f(a_1) \in \{b_1, \ldots, b_m\}$.
- There are $n - 1$ ways to pick a value for $f(a_2) \in \{b_1, \ldots, b_m\} - \{f(a_1)\}$.
- $\vdots$
- There are $n - m + 1$ ways to pick the function value for $f(a_m) \in \{b_1, \ldots, b_m\} - \{f(a_1), \ldots, f(a_{m-1})\}$.

$\Rightarrow$ By product rule, there are $n \cdot (n - 1) \cdot \ldots \cdot (n - m + 1)$ one-to-one functions.
Applications of the product rule
Counting the subsets of a finite set

(6) The number of subsets of a finite set $S = \{a_1, a_2, \ldots, a_n\}$ is $2^n$. 
(6) The number of subsets of a finite set $S = \{a_1, a_2, \ldots, a_n\}$ is $2^n$.

**Answer**

...
(6) The number of subsets of a finite set \( S = \{a_1, a_2, \ldots, a_n\} \) is \( 2^n \).

**Answer**

- For every subset \( B \) of \( A \) we define the bit string \( b_1 b_2 \ldots b_n \) with

\[
    b_i = \begin{cases} 
        1 & \text{if } a_i \in B \\
        0 & \text{otherwise}
    \end{cases}
\]
(6) The number of subsets of a finite set $S = \{a_1, a_2, \ldots, a_n\}$ is $2^n$.

Answer
- For every subset $B$ of $A$ we define the bit string $b_1 b_2 \ldots b_n$ with

$$b_i = \begin{cases} 1 & \text{if } a_i \in B \\ 0 & \text{otherwise} \end{cases}$$

- There is a one-to-one correspondence between the subsets of $A$ and the bit strings of length $n$. 
Applications of the product rule
Counting the subsets of a finite set

(6) The number of subsets of a finite set \( S = \{a_1, a_2, \ldots, a_n\} \) is \( 2^n \).

**Answer**
- For every subset \( B \) of \( A \) we define the bit string \( b_1 b_2 \ldots b_n \) with

\[
b_i = \begin{cases} 
1 & \text{if } a_i \in B \\
0 & \text{otherwise}
\end{cases}
\]

- There is a one-to-one correspondence between the subsets of \( A \) and the bit strings of length \( n \).
- The procedure to define a bit string of length \( n \) is a sequence of \( n \) tasks, where each task chooses the value of a different bit from the bit string.
(6) The number of subsets of a finite set $S = \{a_1, a_2, \ldots, a_n\}$ is $2^n$.

Answer
- For every subset $B$ of $A$ we define the bit string $b_1 b_2 \ldots b_n$ with
  \[
  b_i = \begin{cases} 
  1 & \text{if } a_i \in B \\
  0 & \text{otherwise}
  \end{cases}
  \]
- There is a one-to-one correspondence between the subsets of $A$ and the bit strings of length $n$.
- The procedure to define a bit string of length $n$ is a sequence of $n$ tasks, where each task chooses the value of a different bit from the bit string.
- By the product rule, there are $2^n$ such bit strings.
Applications of the product rule
Counting the subsets of a finite set

(6) The number of subsets of a finite set \( S = \{a_1, a_2, \ldots, a_n\} \) is \( 2^n \).

Answer
- For every subset \( B \) of \( A \) we define the bit string \( b_1b_2\ldots b_n \) with
  \[
  b_i = \begin{cases} 
  1 & \text{if } a_i \in B \\
  0 & \text{otherwise}
  \end{cases}
  \]
- There is a one-to-one correspondence between the subsets of \( A \)
  and the bit strings of length \( n \).
- The procedure to define a bit string of length \( n \) is a sequence
  of \( n \) tasks, where each task chooses the value of a different bit
  from the bit string.
- By the product rule, there are \( 2^n \) such bit strings.

\[ \Rightarrow \text{there are } 2^n \text{ subsets of } S. \]
Basic counting principles

2. The sum rule

Sum rule. If a procedure can be done either in one of \( n_1 \) ways or in one of \( n_2 \) ways, and none of the set of \( n_1 \) ways is the same as any of the \( n_2 \) ways, then there are \( n_1 + n_2 \) ways to do the procedure.

Generalized sum rule. Suppose that a procedure can be done in one of \( n_1 \) ways, in one of \( n_2 \) ways, . . . , or in one of \( n_m \) ways, where none of the set of \( n_i \) ways of doing the task is the same as any of the set of \( n_j \) ways, for all pairs \( i \) and \( j \) with \( 1 \leq i < j \leq m \). Then the number of ways to do the task is \( n_1 + n_2 + . . . + n_m \).
Sum rule. If a procedure can be done either in one of $n_1$ ways or in one of $n_2$ ways, and none of the set of $n_1$ ways is the same as any of the $n_2$ ways, then there are $n_1 + n_2$ ways to do the procedure.
2. The sum rule

**Sum rule.** If a procedure can be done either in one of \( n_1 \) ways or in one of \( n_2 \) ways, and none of the set of \( n_1 \) ways is the same as any of the \( n_2 \) ways, then there are \( n_1 + n_2 \) ways to do the procedure.

**Generalized sum rule.** Suppose that a procedure can be done in one of \( n_1 \) ways, in one of \( n_2 \) ways, . . ., or in one of \( n_m \) ways, where none of the set of \( n_i \) ways of doing the task is the same as any of the set of \( n_j \) ways, for all pairs \( i \) and \( j \) with \( 1 \leq i < j \leq m \). Then the number of ways to do the task is \( n_1 + n_2 + \ldots + n_m \).
(1) Suppose a student can choose a computer project from one of 3 lists. The three lists contain 9, 8, and 12 possible projects respectively. No project is on more than one list. How many possible project are there to choose from?
(1) Suppose a student can choose a computer project from one of 3 lists. The three lists contain 9, 8, and 12 possible projects respectively. No project is on more than one list. How many possible project are there to choose from?

**Answer**

The project can be chosen by selecting from the first list, the second list, or the third list. Because no project is in more than one list, we can apply the sum rule \( \Rightarrow \) there are 9 + 8 + 12 = 29 ways to choose a project.
(1) Suppose a student can choose a computer project from one of 3 lists. The three lists contain 9, 8, and 12 possible projects respectively. No project is on more than one list. How many possible project are there to choose from?

Answer

- The project can be chosen by selecting from the first list, the second list, or the third list.
(1) Suppose a student can choose a computer project from one of 3 lists. The three lists contain 9, 8, and 12 possible projects respectively. No project is on more than one list. How many possible project are there to choose from?

**Answer**

- The project can be chosen by selecting from the first list, the second list, or the third list.
- Because no project is in more than one list, we can apply the sum rule $\Rightarrow$ there are $9 + 8 + 12 = 29$ ways to choose a project.
Many complicated counting problems can not be solved using just the sum rule or just the product rule. But they can be solved using both rules in combination.

EXAMPLES
Many complicated counting problems can not be solved using just the sum rule or just the product rule. But they can be solved using both rules in combination.

**Examples**

(1) Assume a password is 6 to 8 characters long, where each character is either an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?
More complex counting problems

Many complicated counting problems can not be solved using just the sum rule or just the product rule. But they can be solved using both rules in combination.

**Examples**

(1) Assume a password is 6 to 8 characters long, where each character is either an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

**Answer**
More complex counting problems

Many complicated counting problems can not be solved using just the sum rule or just the product rule. But they can be solved using both rules in combination.

**Examples**

(1) Assume a password is 6 to 8 characters long, where each character is either an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

**Answer**

- Let $P$ be the total number of passwords, and $P_6$, $P_7$ and $P_8$ be the number of passwords of length 6, 7, and 8, respectively.
More complex counting problems

Many complicated counting problems can not be solved using just the sum rule or just the product rule. But they can be solved using both rules in combination.

Examples

(1) Assume a password is 6 to 8 characters long, where each character is either an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

Answer

- Let $P$ be the total number of passwords, and $P_6$, $P_7$ and $P_8$ be the number of passwords of length 6, 7, and 8, respectively.
- By sum rule, we have $P = P_6 + P_7 + P_8$. 

Isabela Drămnesc UVT
More complex counting problems

Many complicated counting problems cannot be solved using just the sum rule or just the product rule. But they can be solved using both rules in combination.

**Examples**

(1) Assume a password is 6 to 8 characters long, where each character is either an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

**Answer**

- Let $P$ be the total number of passwords, and $P_6$, $P_7$ and $P_8$ be the number of passwords of length 6, 7, and 8, respectively.
- By sum rule, we have $P = P_6 + P_7 + P_8$.
- To compute $P_m$ for $m \in \{6, 7, 8\}$, we can proceed as follows:
More complex counting problems

Many complicated counting problems cannot be solved using just the sum rule or just the product rule. But they can be solved using both rules in combination.

**Examples**

1. Assume a password is 6 to 8 characters long, where each character is either an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

**Answer**

- Let $P$ be the total number of passwords, and $P_6$, $P_7$ and $P_8$ be the number of passwords of length 6, 7, and 8, respectively.
- By sum rule, we have $P = P_6 + P_7 + P_8$.
- To compute $P_m$ for $m \in \{6, 7, 8\}$, we can proceed as follows:
  - Let $W_m$ be the number of strings of uppercase letters and digits of length $m$. By product rule, $W_m = (26 + 10)^m = 36^m$
More complex counting problems

Many complicated counting problems cannot be solved using just the sum rule or just the product rule. But they can be solved using both rules in combination.

**Examples**

(1) Assume a password is 6 to 8 characters long, where each character is either an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

**Answer**

- Let $P$ be the total number of passwords, and $P_6$, $P_7$ and $P_8$ be the number of passwords of length 6, 7, and 8, respectively.
- By sum rule, we have $P = P_6 + P_7 + P_8$.
- To compute $P_m$ for $m \in \{6, 7, 8\}$, we can proceed as follows:
  - Let $W_m$ be the number of strings of uppercase letters and digits of length $m$. By product rule, $W_m = (26 + 10)^m = 36^m$.
  - Let $N_m$ be the number of strings of uppercase letters of length $m$. By product rule, $N_m = 26^m$. 

More complex counting problems

Many complicated counting problems can not be solved using just the sum rule or just the product rule. But they can be solved using both rules in combination.

Examples

(1) Assume a password is 6 to 8 characters long, where each character is either an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

Answer

- Let $P$ be the total number of passwords, and $P_6$, $P_7$ and $P_8$ be the number of passwords of length 6, 7, and 8, respectively.
- By sum rule, we have $P = P_6 + P_7 + P_8$.
- To compute $P_m$ for $m \in \{6, 7, 8\}$, we can proceed as follows:
  - Let $W_m$ be the number of strings of uppercase letters and digits of length $m$. By product rule, $W_m = (26 + 10)^m = 36^m$.
  - Let $N_m$ be the number of strings of uppercase letters of length $m$. By product rule, $N_m = 26^m$.
- Note that $P_m = W_m - N_m$ (explain why).
More complex counting problems

Many complicated counting problems cannot be solved using just the sum rule or just the product rule. But they can be solved using both rules in combination.

**Examples**

(1) Assume a password is 6 to 8 characters long, where each character is either an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

**Answer**

- Let $P$ be the total number of passwords, and $P_6$, $P_7$ and $P_8$ be the number of passwords of length 6, 7, and 8, respectively.
- By **sum rule**, we have $P = P_6 + P_7 + P_8$.
- To compute $P_m$ for $m \in \{6, 7, 8\}$, we can proceed as follows:
  - Let $W_m$ be the number of strings of **uppercase letters and digits** of length $m$. By **product rule**, $W_m = (26 + 10)^m = 36^m$.
  - Let $N_m$ be the number of strings of **uppercase letters** of length $m$. By **product rule**, $N_m = 26^m$.
- Note that $P_m = W_m - N_m$ (explain why).

$\Rightarrow P = W_6 - N_6 + W_7 - N_7 + W_8 - N_8 = 36^6 - 26^6 + 36^7 - 26^7 + 36^8 - 26^8$. 
(2) In how many ways can we choose 2 books of different languages among 5 books in Romanian, 9 in English, and 10 in German?
(2) In how many ways can we choose 2 books of different languages among 5 books in Romanian, 9 in English, and 10 in German?

**Answer**

\[
\begin{align*}
R&E &= 5 \times 9 = 45 \quad \text{by product rule} \\
R&G &= 5 \times 10 = 50 \quad \text{by product rule} \\
E&G &= 9 \times 10 = 90 \quad \text{by product rule}
\end{align*}
\]

\[\Rightarrow 45 + 50 + 90 = 185 \text{ ways (by sum rule).}\]
A **combinatorial proof** is a proof that uses counting arguments, such as the sum rule and product rule to prove something.

The proofs illustrated in the previous examples are combinatorial proofs.
Assumption: $A$ is a finite set with $n$ elements.

- An $r$-permutation is an ordered sequence $\langle a_1, a_2, \ldots, a_r \rangle$ of $r$ elements of $A$.
- A permutation of $A$ is an ordered sequence $\langle a_1, a_2, \ldots, a_n \rangle$ of all elements of $A$.
- An $r$-combination of $A$ is an unordered selection $\{a_1, a_2, \ldots, a_r\}$ of $r$ elements of $A$. 

Example $\langle 3, 1, 2 \rangle$ and $\langle 1, 3, 2 \rangle$ are permutations of $\{1, 2, 3\}$.

$P(n, r) :=$ the number of $r$-permutations of a set with $n$ elements.

$C(n, r) :=$ the number of $r$-combinations of a set with $n$ elements. Alternative notation: $(n \choose r)$. 

Isabela Drămnescu UVT Graph Theory and Combinatorics – Lecture 1 16 / 31
**Assumption:** $A$ is a finite set with $n$ elements.

- An $r$-permutation is an ordered sequence $\langle a_1, a_2, \ldots, a_r \rangle$ of $r$ elements of $A$.
- A permutation of $A$ is an ordered sequence $\langle a_1, a_2, \ldots, a_n \rangle$ of all elements of $A$.
- An $r$-combination of $A$ is an unordered selection $\{a_1, a_2, \ldots, a_r\}$ of $r$ elements of $A$.

**Example**

\[ \langle 3, 1, 2 \rangle \text{ and } \langle 1, 3, 2 \rangle \text{ are permutations of } \{1, 2, 3\}. \]

\[ \langle 3, 1 \rangle \text{ and } \langle 1, 2 \rangle \text{ are 2-permutations of } \{1, 2, 3\}. \]
Assumption: A is a finite set with $n$ elements.

- An $r$-permutation is an ordered sequence $\langle a_1, a_2, \ldots, a_r \rangle$ of $r$ elements of $A$.
- A permutation of $A$ is an ordered sequence $\langle a_1, a_2, \ldots, a_n \rangle$ of all elements of $A$.
- An $r$-combination of $A$ is an unordered selection $\{a_1, a_2, \ldots, a_r\}$ of $r$ elements of $A$.

Example

$\langle 3, 1, 2 \rangle$ and $\langle 1, 3, 2 \rangle$ are permutations of $\{1, 2, 3\}$.
$\langle 3, 1 \rangle$ and $\langle 1, 2 \rangle$ are 2-permutations of $\{1, 2, 3\}$.

- $P(n, r) :=$ the number of $r$-permutations of a set with $n$ elements.
- $C(n, r) :=$ the number of $r$-combinations of a set with $n$ elements. Alternative notation: $\binom{n}{r}$. 
Theorem

\[ P(n, r) = n \cdot (n - 1) \cdot \ldots \cdot (n - r + 1). \]

Proof
**Theorem**

\[ P(n, r) = n \cdot (n - 1) \cdot \ldots \cdot (n - r + 1). \]

**Proof**

Let \( A = \{a_1, \ldots, a_n\} \)

An \( r \)-permutation is \( p_1, p_2, \ldots, p_r \)

<table>
<thead>
<tr>
<th>choice tasks</th>
<th>( p_1 \in A )</th>
<th>( p_2 \in A - {p_1} )</th>
<th>( \ldots )</th>
<th>( p_r \in A - {p_1, \ldots, p_{r-1}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td># of choices</td>
<td>( n )</td>
<td>( n - 1 )</td>
<td>( \ldots )</td>
<td>( n - r + 1 )</td>
</tr>
</tbody>
</table>

Remark. \( n! \) denotes the product 
\[ 1 \cdot 2 \cdot \ldots \cdot (n-1) \cdot n. \]
Permutations
What is the value of $P(n, r)$?

**Theorem**

$P(n, r) = n \cdot (n - 1) \cdot \ldots \cdot (n - r + 1)$.

**Proof**

$A = \{a_1, \ldots, a_n\}$

$r$-permutation $= p_1, p_2, \ldots, p_r$

<table>
<thead>
<tr>
<th>choice tasks</th>
<th>$p_1 \in A$</th>
<th>$p_2 \in A - {p_1}$</th>
<th>$\ldots$</th>
<th>$p_r \in A - {p_1, \ldots, p_{r-1}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td># of choices</td>
<td>$n$</td>
<td>$n - 1$</td>
<td>$\ldots$</td>
<td>$n - r + 1$</td>
</tr>
</tbody>
</table>

$\Rightarrow P(n, r) = n \cdot (n - 1) \cdot \ldots \cdot (n - r + 1) = \frac{n!}{(n - r)!}$
**Permutations**

What is the value of $P(n, r)$?

**Theorem**

$$P(n, r) = n \cdot (n - 1) \cdot \ldots \cdot (n - r + 1).$$

**Proof**

$A = \{a_1, \ldots, a_n\}$

$r$-permutation $= p_1, p_2, \ldots, p_r$

<table>
<thead>
<tr>
<th>choice tasks</th>
<th>$p_1 \in A$</th>
<th>$p_2 \in A - {p_1}$</th>
<th>$\ldots$</th>
<th>$p_r \in A - {p_1, \ldots, p_{r-1}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td># of choices</td>
<td>$n$</td>
<td>$n - 1$</td>
<td>$\ldots$</td>
<td>$n - r + 1$</td>
</tr>
</tbody>
</table>

$\Rightarrow P(n, r) = n \cdot (n - 1) \cdot \ldots \cdot (n - r + 1) = \frac{n!}{(n - r)!}$

**Remark.** $n!$ denotes the product $1 \cdot 2 \cdot \ldots \cdot (n - 1) \cdot n$. 

Isabela Drămnesc UVT

Graph Theory and Combinatorics – Lecture 1 17 / 31
Theorem

\[ P(n, r) = C(n, r) \times P(r, r). \]

**Combinatorial proof**
Theorem

\[ P(n, r) = C(n, r) \times P(r, r). \]

**Combinatorial proof**

- An \( r \)-permutation of a set with \( n \) elements can be performed by a sequence of 2 tasks:
  1. choose \( r \) elements from the set with \( n \) elements
  2. arrange them.
Theorem

\[ P(n, r) = C(n, r) \times P(r, r). \]

**Combinatorial proof**

- An \( r \)-permutation of a set with \( n \) elements can be performed by a sequence of 2 tasks:
  1. choose \( r \) elements from the set with \( n \) elements
  2. arrange them.
- There are \( C(n, r) \) ways to choose \( r \) elements out of \( n \) \( \Rightarrow \) task (1) can be done in \( C(n, r) \) ways.
Theorem

\[ P(n, r) = C(n, r) \times P(r, r). \]

**Combinatorial proof**

- An \( r \)-permutation of a set with \( n \) elements can be performed by a sequence of 2 tasks:
  1. choose \( r \) elements from the set with \( n \) elements
  2. arrange them.
- There are \( C(n, r) \) ways to choose \( r \) elements out of \( n \) \( \Rightarrow \) task (1) can be done in \( C(n, r) \) ways.
- There are \( P(r, r) \) ways to arrange \( r \) elements \( \Rightarrow \) task (2) can be done in \( P(r, r) \) ways.
Theorem

\[ P(n, r) = C(n, r) \times P(r, r). \]

Combinatorial proof

- An \( r \)-permutation of a set with \( n \) elements can be performed by a sequence of 2 tasks:
  1. choose \( r \) elements from the set with \( n \) elements
  2. arrange them.
- There are \( C(n, r) \) ways to choose \( r \) elements out of \( n \) \( \Rightarrow \) task (1) can be done in \( C(n, r) \) ways.
- There are \( P(r, r) \) ways to arrange \( r \) elements \( \Rightarrow \) task (2) can be done in \( P(r, r) \) ways.
- \( \Rightarrow \) by product rule, we obtain \( P(n, r) = C(n, r) \times P(r, r) \).
Combinations
Counting combinations

$C(n, r) = ?$

We already know how to compute $P(n, r)$, it is $\frac{n!}{(n - r)!}$.

We proved that $P(n, r) = C(n, r) \times P(r, r)$.

$\Rightarrow C(n, r) = \frac{n!}{(n - r)!} \times \frac{0!}{r!} = \frac{n!}{r!(n - r)!}$.
Combinations
Counting combinations

\[ C(n, r) = ? \]

- We already know how to compute \( P(n, r) \), it is \( \frac{n!}{(n-r)!} \)
Combinations

Counting combinations

$$C(n, r) =?$$

- We already know how to compute $P(n, r)$, it is $\frac{n!}{(n-r)!}$
- We proved that $P(n, r) = C(n, r) \times P(r, r)$
We already know how to compute $P(n, r)$, it is $\frac{n!}{(n-r)!}$.

We proved that $P(n, r) = C(n, r) \times P(r, r)$

$\Rightarrow C(n, r) = \frac{P(n, r)}{P(r, r)} = \frac{n!}{(n-r)!} \cdot \frac{0!}{r!} = \frac{n!}{r!(n-r)!}$
Theorem

\[ C(n, r) = C(n - 1, r - 1) + C(n - 1, r) \] for all \( n > r > 0 \).
Properties of combinations

**Theorem**

\[ C(n, r) = C(n − 1, r − 1) + C(n − 1, r) \text{ for all } n > r > 0. \]

**Combinatorial proof**

- Let \( S = \{a_1, a_2, \ldots, a_n\} \). There are \( C(n, r) \) ways to choose \( r \) elements from \( S \). We distinguish 2 distinct possibilities:
Properties of combinations

**Theorem**

\[ C(n, r) = C(n - 1, r - 1) + C(n - 1, r) \text{ for all } n > r > 0. \]

**Combinatorial proof**

- Let \( S = \{a_1, a_2, \ldots, a_n\} \). There are \( C(n, r) \) ways to choose \( r \) elements from \( S \). We distinguish 2 distinct possibilities:
  1. The choice of \( r \) elements from \( S \) contains \( a_1 \). Let \( N_1 \) be the number of such choices.
  2. The choice of \( r \) elements from \( S \) does not contain \( a_1 \). Let \( N_2 \) be the number of such choices.

By sum rule, \( C(n, r) = N_1 + N_2 \).

- \( N_1 = C(n - 1, r - 1) \) because we have to choose \( r - 1 \) elements from \( \{a_2, \ldots, a_n\} \).
- \( N_2 = C(n - 1, r) \) because we have to choose \( r \) elements from \( \{a_2, \ldots, a_n\} \).

\[ \Rightarrow C(n, r) = C(n - 1, r - 1) + C(n - 1, r) \]
Properties of combinations

**Theorem**

\[ C(n, r) = C(n - 1, r - 1) + C(n - 1, r) \text{ for all } n > r > 0. \]

**Combinatorial proof**

- Let \( S = \{a_1, a_2, \ldots, a_n\} \). There are \( C(n, r) \) ways to choose \( r \) elements from \( S \). We distinguish 2 distinct possibilities:
  1. The choice of \( r \) elements from \( S \) contains \( a_1 \). Let \( N_1 \) be the number of such choices.
  2. The choice of \( r \) elements from \( S \) does not contain \( a_1 \). Let \( N_2 \) be the number of such choices.

By sum rule,
\[ C(n, r) = N_1 + N_2. \]

But:
\[ N_1 = C(n - 1, r - 1) \text{ because we have to choose } r - 1 \text{ elements from } \{a_2, \ldots, a_n\}. \]
\[ N_2 = C(n - 1, r) \text{ because we have to choose } r \text{ elements from } \{a_2, \ldots, a_n\}. \]
Theorem

\[ C(n, r) = C(n - 1, r - 1) + C(n - 1, r) \text{ for all } n > r > 0. \]

Combinatorial proof

Let \( S = \{a_1, a_2, \ldots, a_n\} \). There are \( C(n, r) \) ways to choose \( r \) elements from \( S \). We distinguish 2 distinct possibilities:

1. The choice of \( r \) elements from \( S \) contains \( a_1 \). Let \( N_1 \) be the number of such choices.

2. The choice of \( r \) elements from \( S \) does not contain \( a_1 \). Let \( N_2 \) be the number of such choices.

By sum rule, \( C(n, r) = N_1 + N_2 \). But:
### Theorem

\[ C(n, r) = C(n - 1, r - 1) + C(n - 1, r) \quad \text{for all } n > r > 0. \]

**Combinatorial proof**

- Let \( S = \{a_1, a_2, \ldots, a_n\} \). There are \( C(n, r) \) ways to choose \( r \) elements from \( S \). We distinguish 2 distinct possibilities:
  1. The choice of \( r \) elements from \( S \) contains \( a_1 \). Let \( N_1 \) be the number of such choices.
  2. The choice of \( r \) elements from \( S \) does not contain \( a_1 \). Let \( N_2 \) be the number of such choices.

By sum rule, \( C(n, r) = N_1 + N_2 \). But:

- \( N_1 = C(n - 1, r - 1) \) because we have to choose \( r - 1 \) elements from \( \{a_2, \ldots, a_n\} \)
- \( N_2 = C(n - 1, r) \) because we have to choose \( r \) elements from \( \{a_2, \ldots, a_n\} \)
**Theorem**

\[ C(n, r) = C(n - 1, r - 1) + C(n - 1, r) \quad \text{for all } n > r > 0. \]

**Combinatorial proof**

- Let \( S = \{a_1, a_2, \ldots, a_n\} \). There are \( C(n, r) \) ways to choose \( r \) elements from \( S \). We distinguish 2 distinct possibilities:
  1. The choice of \( r \) elements from \( S \) contains \( a_1 \). Let \( N_1 \) be the number of such choices.
  2. The choice of \( r \) elements from \( S \) does not contain \( a_1 \). Let \( N_2 \) be the number of such choices.

By sum rule, \( C(n, r) = N_1 + N_2 \). But:

- \( N_1 = C(n - 1, r - 1) \) because we have to choose \( r - 1 \) elements from \( \{a_2, \ldots, a_n\} \)
- \( N_2 = C(n - 1, r) \) because we have to choose \( r \) elements from \( \{a_2, \ldots, a_n\} \)

\[ \Rightarrow C(n, r) = C(n - 1, r - 1) + C(n - 1, r). \]
1. Give an algebraic proof, using the formulas for $C(n, r)$, of the fact that $C(n, r) = C(n - 1, r - 1) + C(n - 1, r)$.

2. Give a combinatorial proof of the fact that $C(n, r) = C(n, n - r)$.

3. How many ways are there to select a first-prize winner, a second-prize winner, and a third-prize winner from 100 different people who have entered a contest?

4. In how many ways can $n$ people stand to form a ring?

5. How many permutations of the letters $ABCDEFGH$ contain the string $ABC$?

6. How many bit strings of length $n$ contain exactly $r$ 1s?
Generalized permutations and combinations
Permutations with repetition

In many counting problems, we want to use elements repeatedly. Permutations and combinations assume that every item appears only once. An \( r \)-permutation with repetition of a set of \( n \) elements is an arrangement of \( r \) elements from that set, where elements may occur more than once.

Example
How many strings of length \( n \) can be formed with the lowercase and uppercase letters of the English alphabet?

Answer:

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( 52 )</td>
</tr>
</tbody>
</table>

\[ \Rightarrow 52^n \text{ strings (by product rule)} \]

Theorem
The number of \( r \)-permutations of a set of \( n \) elements with repetition is \( n^r \).
In many counting problems, we want to use elements repeatedly.
In many counting problems, we want to use elements repeatedly.

Permutations and combinations assume that every item appears only once.

- **Example**
  - How many strings of length $n$ can be formed with the lowercase and uppercase letters of the English alphabet?
  - **Answer:** By the product rule, there are $52$ strings of length $n$.

- **Theorem**
  - The number of $r$-permutations of a set of $n$ elements with repetition is $n^r$. 

Isabela Drămnesc UVT

Graph Theory and Combinatorics – Lecture 1
In many counting problems, we want to use elements repeatedly. Permutations and combinations assume that every item appears only once.

An \( r \)-permutation with repetition of a set of \( n \) elements is an arrangement of \( r \) elements from that set, where elements may occur more than once.

**Example**

How many strings of length \( n \) can be formed with the lowercase and uppercase letters of the English alphabet?

Answer: \(|\text{Alphabet}_{\text{English}}| = 52 \Rightarrow 52^n \text{ strings} \quad \text{(by product rule)}\)
In many counting problems, we want to use elements repeatedly.

Permutations and combinations assume that every item appears only once.

An \( r \)-permutation with repetition of a set of \( n \) elements is an arrangement of \( r \) elements from that set, where elements may occur more than once.

**Example**

How many strings of length \( n \) can be formed with the lowercase and uppercase letters of the English alphabet?

**Answer:** \(|\text{Alphabet}_{\text{English}}| = 52 \Rightarrow 52^n \text{ strings} \) (by product rule)

**Theorem**

*The number of \( r \)-permutations of a set of \( n \) elements with repetition is \( n^r \).*
An \( r \)-combination with repetition of a set of \( n \) elements is a choice of \( r \) elements from a bag of elements of \( n \) kinds, where we can choose the same kind of element any number of times.

**Q:** How many \( r \)-combinations with repetition of a set of \( n \) elements are there?

**Example**

How many ways are there to select 5 bills from a cash box containing bills of $1, $2, $5, $10, $20, $50. Assume that: the order in which the bills are chosen does not matter; the bills are indistinguishable; there are at least 5 bills of each type.
Combinations with repetition

Example – continued

Five not necessarily distinct bills = a 5-combination with repetition from the set \{\$1, \$2, \$5, \$10, \$20, \$50\} of bill kinds = a placement of five * in the slots of the cash box depicted below:

- The number of * in a slot represents the number of bills taken from that place.

⇒ The number of 5-combinations with repetition of a set with 6 elements = the number of ways to place 5 stars in 6 slots.

<table>
<thead>
<tr>
<th>$1$</th>
<th>$2$</th>
<th>$5$</th>
<th>$10$</th>
<th>$20$</th>
<th>$50$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>**</td>
<td></td>
<td>***</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>**</td>
<td></td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>**</td>
<td></td>
</tr>
</tbody>
</table>

... cash box with 6 types of bills
Note that

- Every placement of 5 stars in 6 possible slots is uniquely described by a string of 5 stars and 5 red bars.
- In general, the number of $r$-combinations with repetition of a set with $n$ elements = the number of strings with $r$ stars and $n - 1$ red bars.

Q: In how many ways can we arrange $n - 1$ bars and $r$ stars?
Note that

- Every placement of 5 stars in 6 possible slots is uniquely described by a string of 5 stars and 5 red bars.
- In general, the number of \( r \)-combinations with repetition of a set with \( n \) elements = the number of strings with \( r \) stars and \( n-1 \) red bars.

Q: In how many ways can we arrange \( n-1 \) bars and \( r \) stars?

A: The sequence has length \( n+r-1 \)

\[ \Rightarrow \text{there are } n+r-1 \text{ positions in the sequence} \]
\[ \Rightarrow \text{we must choose } r \text{ positions out of } n+r-1 \text{ to be filled with stars; the others will be filled with red bars.} \]
Note that

- Every placement of 5 stars in 6 possible slots is uniquely described by a string of 5 stars and 5 red bars.
- In general, the number of $r$-combinations with repetition of a set with $n$ elements = the number of strings with $r$ stars and $n-1$ red bars.

Q: In how many ways can we arrange $n-1$ bars and $r$ stars?

A: The sequence has length $n + r - 1$
  $\Rightarrow$ there are $n + r - 1$ positions in the sequence
  $\Rightarrow$ we must choose $r$ positions out of $n + r - 1$ to be filled with stars; the others will be filled with red bars.

There are $C(n + r - 1, r)$ such choices.
Note that

- Every placement of 5 stars in 6 possible slots is uniquely described by a string of 5 stars and 5 red bars.
- In general, the number of $r$-combinations with repetition of a set with $n$ elements = the number of strings with $r$ stars and $n - 1$ red bars.

Q: In how many ways can we arrange $n - 1$ bars and $r$ stars?
A: The sequence has length $n + r - 1$

⇒ there are $n + r - 1$ positions in the sequence
⇒ we must choose $r$ positions out of $n + r - 1$ to be filled with stars; the others will be filled with red bars.

There are $\binom{n + r - 1}{r}$ such choices.

Theorem
The number of $r$-combinations with repetition of $n$ elements is $\binom{r + n - 1}{r}$. 
### Permutations and Combinations

**Summary**

<table>
<thead>
<tr>
<th>Type</th>
<th>Repetition allowed?</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$-permutations</td>
<td>No</td>
<td>$P(n, r) = \frac{n!}{(n-r)!}$</td>
</tr>
<tr>
<td>$r$-combinations</td>
<td>No</td>
<td>$C(n, r) = \frac{n!}{r!(n-r)!}$</td>
</tr>
<tr>
<td>$r$-permutations with repetition</td>
<td>Yes</td>
<td>$n^r$</td>
</tr>
<tr>
<td>$r$-combinations with repetition</td>
<td>Yes</td>
<td>$C(n + r - 1, r) = \frac{(n + r - 1)!}{r!(n-1)!}$</td>
</tr>
</tbody>
</table>
Problem

How many strings can be made by reordering the string SUCCESS?

SUCCESS contains 3 Ss, 2 Cs, 1 U, 1 E.

placements of 3 Ss among 7 places: \( C(7, 3) \) ⇒ 4 places left.

placements of 2 Cs among 4 places: \( C(4, 2) \) ⇒ 2 places left.

placements of 1 U among 2 places: \( C(2, 1) \) ⇒ 1 place left.

placements of 1 E among 1 place: \( C(1, 1) \) ⇒ 1 place left.

by product rule, the number is \( C(7, 3) \times C(4, 2) \times C(2, 1) \times C(1, 1) = 7! / 3!2!1!1! \)
Problem

How many strings can be made by reordering the string SUCCESS?

- SUCCESS contains 3 Ss, 2 Cs, 1U, 1 E.
- placements of 3 Ss among 7 places: \( C(7, 3) \Rightarrow 4 \) places left.
- placements of 2 Cs among 4 places: \( C(4, 2) \Rightarrow 2 \) places left.
- placements of 1 U among 2 places: \( C(2, 1) \Rightarrow 1 \) place left.
- placements of 1 E among 1 place: \( C(1, 1) \).

\( \Rightarrow \) by product rule, the number is

\[
C(7, 3) \times C(4, 2) \times C(2, 1) \times C(1, 1) = \frac{7!}{3!2!1!1!}
\]
Theorem

The number of different permutations of $n$ objects, where there are

- $n_1$ indistinguishable elements of type 1
- $n_2$ indistinguishable elements of type 2

... 

- $n_k$ indistinguishable elements of type $n_k$

is

$$\frac{n!}{n_1!n_2! \cdots n_k!}.$$
Binomial and multinomial numbers

- The binomial numbers are the coefficients $c_{n,k}$ in the formula

\[(x + y)^n = \sum_{k=0}^{n} c_{n,k} \cdot x^{n-k} y^{k}\]

- The multinomial numbers are the coefficients $c_{n,k_1,\ldots,k_r}$ in the formula

\[(x_1 + \ldots + x_r)^n = \sum_{k_1+\ldots+k_r=n} c_{n,k_1,\ldots,k_r} \cdot x_1^{k_1} x_2^{k_2} \ldots x_r^{k_r}\]

Example

\[(x + y)^3 = 1 \cdot x^3 + 3 \cdot x^2 y + 3 \cdot x y^2 + 1 \cdot y^3\]
\[(x_1 + x_2 + x_3)^2 = 1 \cdot x_1^2 + 1 \cdot x_2^2 + 1 \cdot x_3^2 + 2 \cdot x_1 x_2 + 2 \cdot x_1 x_3 + 2 \cdot x_2 x_3\]
Binomial numbers and multinomial numbers

How to compute them?

\[(x_1 + \ldots + x_r)^n = \sum_{k_1+\ldots+k_r=n} \frac{n!}{k_1!\ldots k_r!} x_1^{k_1} x_2^{k_2} \ldots x_r^{k_r}\]
Binomial numbers and multinomial numbers
How to compute them?

\[(x_1 + \ldots + x_r)^n = \sum_{k_1 + \ldots + k_r = n} \frac{n!}{k_1! \ldots k_r!} \cdot x_1^{k_1} x_2^{k_2} \ldots x_r^{k_r}\]

**Combinatorial proof**

\[ (x_1 + \ldots + x_r)^n = \underbrace{(x_1 + \ldots + x_r) \cdot \ldots \cdot (x_1 + \ldots + x_r)}_{n \text{ parenthesized expressions}} \]
Binomial numbers and multinomial numbers
How to compute them?

\[(x_1 + \ldots + x_r)^n = \sum_{k_1+\ldots+k_r=n}^{n} \frac{n!}{k_1!\ldots k_r!} \cdot x_1^{k_1} x_2^{k_2} \ldots x_r^{k_r}\]

**Combinatorial proof**

\[(x_1 + \ldots + x_r)^n = \underbrace{(x_1 + \ldots + x_r) \cdot \ldots \cdot (x_1 + \ldots + x_r)}_{n \text{ parenthesized expressions}}\]

In how many ways can we produce the monomial \(x_1^{k_1} \cdot \ldots \cdot x_r^{k_r}\)?
Binomial numbers and multinomial numbers

How to compute them?

\[(x_1 + \ldots + x_r)^n = \sum_{k_1+\ldots+k_r=n}^{n} \frac{n!}{k_1! \ldots k_r!} \cdot x_1^{k_1} x_2^{k_2} \ldots x_r^{k_r}\]

**Combinatorial proof**

\[(x_1 + \ldots + x_r)^n = \underbrace{(x_1 + \ldots + x_r) \cdot \ldots \cdot (x_1 + \ldots + x_r)}_{n \text{ parenthesized expressions}}\]

**In how many ways can we produce the monomial** \(x_1^{k_1} \cdot \ldots \cdot x_r^{k_r}\)?

▷ Choose \(k_1\) parentheses from where \(x_1\) originates \(\Rightarrow \binom{n}{k_1}\) choices.

▷ Choose \(k_2\) parentheses from where \(x_2\) originates \(\Rightarrow \binom{n-k_1}{k_2}\) choices.

\[\ldots\]

▷ Choose \(k_r\) parentheses from where \(x_r\) originates \(\Rightarrow \binom{n-\sum_{i=1}^{r-1}k_i}{k_r}\) choices.
Binomial numbers and multinomial numbers
How to compute them?

\[(x_1 + \ldots + x_r)^n = \sum_{k_1+\ldots+k_r=n}^{n} \frac{n!}{k_1! \ldots k_r!} \cdot x_1^{k_1} x_2^{k_2} \ldots x_r^{k_r}\]

**Combinatorial proof**

\[(x_1 + \ldots + x_r)^n = \underbrace{(x_1 + \ldots + x_r) \cdot \ldots \cdot (x_1 + \ldots + x_r)}_{n \text{ parenthesized expressions}}\]

In how many ways can we produce the monomial \(x_1^{k_1} \cdot \ldots \cdot x_r^{k_r}\)?

▷ Choose \(k_1\) parentheses from where \(x_1\) originates \(\Rightarrow \binom{n}{k_1}\) choices.

▷ Choose \(k_2\) parentheses from where \(x_2\) originates \(\Rightarrow \binom{n-k_1}{k_2}\) choices.

▷ \ldots

▷ Choose \(k_r\) parentheses from where \(x_r\) originates \(\Rightarrow \binom{n-\sum_{i=1}^{r-1} k_i}{k_r}\) choices.

\(\Rightarrow\) by the product rule, the number of occurrences of \(x_1^{k_1} \cdot \ldots \cdot x_r^{k_r}\) in the right hand side is \(\binom{n}{k_1} \binom{n-k_1}{k_2} \ldots \binom{n-\sum_{i=1}^{r-1} k_i}{k_r} = \frac{n!}{k_1! \ldots k_r!}\).
Binomial numbers and multinomial numbers

Conclusions

- For the formula \( \binom{n}{k_1! \cdots k_r!} \) with \( k_1 + \ldots + k_r = n \) we often use the notation \( \binom{n}{k_1, \ldots, k_r} \).
- The binomial formula is

\[
(x + y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k
\]

- The multinomial formulas

\[
(x_1 + \ldots + x_r)^n = \sum_{k_1 + \ldots + k_r = n} \binom{n}{k_1, \ldots, k_r} x_1^{k_1} \ldots x_r^{k_r}
\]

**Remark.** \( \binom{n}{k} = \binom{n}{k, n-k} \) and

\[
(x_1 + x_2)^n = \sum_{k=0}^{n} \binom{n}{k} x_1^k x_2^{n-k} = \sum_{k_1 + k_2 = n} \binom{n}{k_1, k_2} x_1^{k_1} x_2^{k_2}.
\]