INTRODUCTION TO THE THEORY OF COMPUTATION
LECTURE NOTES

Professor Anita Wasilewska
Stony Brook University (SUNY at Stony Brook)
Course Text Book

ELEMENTS OF THE THEORY OF COMPUTATION
Harry R. Lewis, and Christos H. Papadimitriou
Prentice Hall, 2nd Edition
CHAPTER 1
SETS, RELATIONS, and LANGUAGES

LECTURE SLIDES
Chapter 1
Sets, Relations, and Languages

Slides Set 1
PART 1: Sets
PART 2: Relations and Functions
PART 3: Special types of Binary Relations

Slides Set 2
PART 4: Finite and Infinite Sets
PART 5: Fundamental Proof Techniques
Chapter 1
Sets, Relations, and Languages

Slides Set 3
PART 6: Closures and Algorithms

Slides Set 4
PART 7: Alphabets and languages
PART 8: Finite Representation of Languages
Chapter 1
Sets, Relations, and Languages

Slides Set 1
PART 1: Sets
Sets

Set  A set is a collection of objects

Elements  The objects comprising a set are called its elements or members

\[ a \in A \] denotes that \( a \) is an element of a set \( A \)
\[ a \notin A \] denotes that \( a \) is not an element of \( A \)

Empty Set  is a set without elements

Empty Set  is denoted by \( \emptyset \)
Sets

Sets can be defined by listing their elements;

Example
The set

\[ A = \{a, \emptyset, \{a, \emptyset\}\} \]

has 3 elements:

\[ a \in A, \quad \emptyset \in A, \quad \{a, \emptyset\} \in A \]
Sets can be defined by referring to other sets and to properties \( P(x) \) that elements may or may not have.

We write it as

\[
B = \{x \in A : P(x)\}
\]

Example

Let \( N \) be a set of natural numbers

\[
B = \{n \in N : n < 0\} = \emptyset
\]
Operations on Sets

Set Inclusion
\( A \subseteq B \) if and only if \( \forall a (a \in A \Rightarrow a \in B) \) is a true statement

Set Equality
\( A = B \) if and only if \( A \subseteq B \) and \( B \subseteq A \)

Proper Subset
\( A \subset B \) if and only if \( A \subseteq B \) and \( A \neq B \)
Operations on Sets

Subset Notations

$A \subseteq B$ for a subset (might be improper)

$A \subset B$ for a proper subset

Power Set

Set of all subsets of a given set

$\mathcal{P}(A) = \{B : B \subseteq A\}$

Other Notation

$2^A = \{B : B \subseteq A\}$
Operations on Sets

Union
\[ A \cup B = \{ x : \ x \in A \quad \text{or} \quad x \in B \} \]
We write:
\[ x \in A \cup B \quad \text{if and only if} \quad x \in A \cup x \in B \]

Intersection
\[ A \cap B = \{ x : \ x \in A \quad \text{and} \quad x \in B \} \]
We write:
\[ x \in A \cap B \quad \text{if and only if} \quad x \in A \cap x \in B \]
Operations on Sets

Relative Complement
\[ x \in (A - B) \quad \text{if and only if} \quad x \in A \quad \text{and} \quad x \notin B \]
We write:
\[ A - B = \{ x : x \in A \cap x \notin B \} \]

Complement is defined only for \( A \subseteq U \), where \( U \) is called an universe
\[ -A = U - A \]
We write for \( x \in U \),
\[ x \in -A \quad \text{if and only if} \quad x \notin A \]
Operations on Sets

**Algebra of sets** consists of properties of sets that are **true** for all sets involved.

We use **tautologies** of propositional logic to prove **basic** properties of the **algebra of sets**.

We then use the **basic properties** to **prove** more **elaborated** properties of sets.
Operations on Sets

It is possible to form intersections and unions of more than two, or even a finite number of sets.

Let $\mathcal{F}$ denote is any collection of sets.

We write $\bigcup \mathcal{F}$ for the set whose elements are the elements of all of the sets in $\mathcal{F}$.

**Example** Let

$$\mathcal{F} = \{\{a\}, \{\emptyset\}, \{a, \emptyset, b\}\}$$

We get

$$\bigcup \mathcal{F} = \{a, \emptyset, b\}$$
Operations on Sets

Observe that given

\[ \mathcal{F} = \{\{a\}, \emptyset, \{a, \emptyset, b\}\} = \{A_1, A_2, A_3\} \]

we have that

\[ \{a\} \cup \emptyset \cup \{a, \emptyset, b\} = A_1 \cup A_2 \cup A_3 = \{a, \emptyset, b\} = \bigcup \mathcal{F} \]

Hence we have that for any element \( x \),

\[ x \in \bigcup \mathcal{F} \text{ if and only if } \text{ there exists } i, \text{ such that } x \in A_i \]
Operations on Sets

We define formally

**Generalized Union** of any family \( F \) of sets is

\[
\bigcup F = \{ x : \text{exists a set } S \in F \text{ such that } x \in S \}
\]

We write it also as

\[
x \in \bigcup F \text{ if and only if } \exists_{S \in F} x \in S
\]
Operations on Sets

**Generalized Intersection** of any family $F$ of sets is

$$\bigcap F = \{ x : \forall S \in F \ x \in S \}$$

We write

$$x \in \bigcap F \text{ if and only if } \forall S \in F \ x \in S$$
Operations on Sets

Ordered Pair

Given two sets $A, B$ we denote by $(a, b)$ an ordered pair, where $a \in A$ and $b \in B$. We call $a$ a first coordinate of $(a, b)$ and $b$ its second coordinate.

We define

$$(a, b) = (c, d)$$

if and only if $a = c$ and $b = d$.
Operations on Sets

Cartesian Product
Given two sets $A$ and $B$, the set

$$A \times B = \{(a, b) : a \in A \text{ and } b \in B\}$$

is called a **Cartesian product** (cross product) of sets $A, B$.

We write

$$(a, b) \in A \times B \quad \text{if and only if} \quad a \in A \text{ and } b \in B$$
Chapter 1
Sets, Relations, and Languages

Slides Set 1
PART 2: Relations and Functions
Binary Relations

Binary Relation
Any set $R$ such that $R \subseteq A \times A$ is called a binary relation defined in a set $A$.

Domain, Range of $R$
Given a binary relation $R \subseteq A \times A$, the set

$$D_R = \{a \in A : (a, b) \in R\}$$

is called a domain of the relation $R$.
The set

$$V_R = \{b \in A : (a, b) \in R\}$$

is called a range (set of values) of the relation $R$. 
Ordered tuple

Given sets $A_1, \ldots, A_n$, an element $(a_1, a_2, \ldots, a_n)$ such that $a_i \in A_i$ for $i = 1, 2, \ldots, n$ is called an ordered tuple.

Cartesian Product of sets $A_1, \ldots, A_n$ is a set

$$A_1 \times A_2 \times \ldots \times A_n = \{(a_1, a_2, \ldots, a_n) : a_i \in A_i, i = 1, 2, \ldots, n\}$$

n-ary Relation on sets $A_1, \ldots, A_n$ is any subset of $A_1 \times A_2 \times \ldots \times A_n$, i.e. the set

$$R \subseteq A_1 \times A_2 \times \ldots \times A_n$$
Function as Relation

Definition
A binary relation \( R \subseteq A \times B \) is a **function** from \( A \) to \( B \) if and only if the following condition holds

\[
\forall a \in A \exists! b \in B \ (a, b) \in R
\]

where \( \exists! b \in B \) means there is **exactly one** \( b \in B \)

Because the condition says that for any \( a \in A \) we have **exactly one** \( b \in B \), we write

\[
R(a) = b \quad \text{for} \quad (a, b) \in R
\]
Function as Relation

Given a binary relation

\[ R \subseteq A \times B \]

that is a function

The set \( A \) is called a domain of the function \( R \) and we write:

\[ R : A \rightarrow B \]

To denote that the relation \( R \) is a function and say that \( R \) maps the set \( A \) into the set \( B \)
Functions

Function notation
We denote relations that are functions by letters f, g, h,... and write

\[ f : A \rightarrow B \]

say that the function f maps the set A into the set B

Domain, Codomain
Let \( f : A \rightarrow B, \)
the set A is called a domain of f, and the set B is called a codomain of f
Functions

Range
Given a function \( f : A \rightarrow B \)
The set
\[
R_f = \{ b \in B : b = f(a) \text{ and } a \in A \}
\]
is called a range of the function \( f \)
By definition, the range of \( f \) is a subset of its codomain \( B \)
We write \( R_f = \{ b \in B : \exists a \in A \ b = f(a) \} \)

The set
\[
f = \{(a, b) \in A \times B : b = f(a)\}
\]
is called a graph of the function \( f \)
Functions

Function "onto"

The function \( f : A \rightarrow B \) is an onto function if and only if the following condition holds:

\[
\forall b \in B \ \exists a \in A \ f(a) = b
\]

We denote it by:

\[
f : A \xrightarrow{\text{onto}} B
\]
Functions

Function "one-to-one"

The function \( f : A \longrightarrow B \)
is called a one-to-one function and denoted by

\[
f : A \overset{1-1}{\longrightarrow} B
\]

if and only if the following condition holds

\[
\forall x, y \in A (x \neq y \Rightarrow f(x) \neq f(y))
\]
Functions

A function \( f : A \rightarrow B \) is not one-to-one function if and only if the following condition holds

\[
\exists x, y \in A ( x \neq y \cap f(x) = f(y) )
\]

If a function \( f \) is 1-1 and onto we denote it as

\[
f : A^{1-1,onto} \rightarrow B
\]
Functions

Composition of functions

Let \( f \) and \( g \) be two functions such that

\[
 f : \ A \rightarrow B \quad \text{and} \quad g : \ B \rightarrow C
\]

We define a new function

\[
 h : \ A \rightarrow C
\]

called a composition of functions \( f \) and \( g \) as follows:
for any \( x \in A \) we put

\[
 h(x) = g(f(x))
\]
Functions

Composition notation
Given function \( f \) and \( g \) such that

\[
f : \ A \rightarrow B \quad \text{and} \quad g : \ B \rightarrow C
\]

We denote the composition of \( f \) and \( g \) by \((f \circ g)\)
in order to stress that the function

\[
f : \ A \rightarrow B
\]

"goes first" followed by the function

\[
g : \ B \rightarrow C
\]

with a shared set \( B \) between them
Functions

We write now the **definition** of **composition** of functions \( f \) and \( g \) using the **composition notation** (name for the composition function) \( (f \circ g) \) as follows:

The composition \( (f \circ g) \) is a **new** function

\[
(f \circ g) : A \rightarrow C
\]

such that for any \( x \in A \) we put

\[
(f \circ g)(x) = g(f(x))
\]
Functions

There is also other notation (name) for the composition of \( f \) and \( g \) that uses the symbol \((g \circ f)\), i.e. we put

\[
(g \circ f)(x) = g(f(x)) \quad \text{for all } x \in A
\]

This notation was invented to help calculus students to remember the formula \( g(f(x)) \) defining the composition of functions \( f \) and \( g \)
Functions

Inverse function

Let \( f : A \rightarrow B \) and \( g : B \rightarrow A \)

\( g \) is called an **inverse** function to \( f \) if and only if the following condition holds

\[
\forall a \in A (f \circ g)(a) = g(f(a)) = a
\]

If \( g \) is an **inverse** function to \( f \) we denote by \( g = f^{-1} \)
Functions

Identity function
A function \( f : A \rightarrow A \) is called an **identity** on \( A \) if and only if the following condition holds

\[ \forall a \in A \quad f(a) = a \]

Inverse and Identity
Let \( f : A \rightarrow B \) and let \( f^{-1} : B \rightarrow A \) be an **inverse** to \( f \), then the following hold

\[ (f \circ f^{-1})(a) = f^{-1}(f(a)) = I(a) = a, \quad \text{for all} \quad a \in A \]

\[ (f^{-1} \circ f)(b) = f(f^{-1}(b)) = I(b) = b, \quad \text{for all} \quad b \in B \]
Functions: Image and Inverse Image

Image
Given a function \( f : X \rightarrow Y \) and a set \( A \subseteq X \n\)
The set

\[
 f[A] = \{ y \in Y : \exists x ( x \in A \cap y = f(x)) \}
\]

is called an image of the set \( A \subseteq X \) under the function \( f \n\)
We write

\[ y \in f[A] \text{ if and only if } \exists x ( x \in A \text{ and } y = f(x)) \]

Other symbols used to denote the image are

\[ f^{-1}(A) \text{ or } f(A) \]
Functions: Image and Inverse Image

Inverse Image
Given a function \( f : X \rightarrow Y \) and a set \( B \subseteq Y \)

The set

\[
f^{-1}[B] = \{x \in X : f(x) \in B\}
\]

is called an inverse image of the set \( B \subseteq Y \) under the function \( f \)

We write

\[
x \in f^{-1}[B] \quad \text{if and only if} \quad f(x) \in B
\]

Other symbol used to denote the inverse image are

\[
f^{-1}(B) \quad \text{or} \quad f\leftarrow(B)
\]
Sequences

Definition

A **sequence** of elements of a set $A$ is any **function** from the set of natural numbers $N$ into the set $A$, i.e. any function

$$f : N \rightarrow A$$

Any $f(n) = a_n$ is called **n-th term** of the sequence $f$

Notations

$$f = \{a_n\}_{n \in N}, \quad \{a_n\}_{n \in N}, \quad \{a_n\}$$
Sequences Example

Example
We define a sequence $f$ of real numbers $\mathbb{R}$ as follows

$$f : \mathbb{N} \rightarrow \mathbb{R}$$

such that

$$f(n) = n + \sqrt{n}$$

We also use a shorthand notation for the function $f$ and write it as

$$a_n = n + \sqrt{n}$$
Sequences Example

We often write the function \( f = \{a_n\} \) in an even **shorter** and **informal** form as

\[
\begin{align*}
a_0 &= 0, \\
a_1 &= 1 + 1 = 2, \\
a_2 &= 2 + \sqrt{2} \ldots 
\end{align*}
\]

or even as

\[
\begin{align*}
0, \quad 2, \quad 2 + \sqrt{2}, \quad 3 + \sqrt{3}, \quad \ldots \ldots \ldots \ldots \ldots \ldots \quad n + \sqrt{n} \ldots 
\end{align*}
\]
Observations

Observation 1
By definition, sequence of elements of any set is always infinite (countably infinite) because the domain of the sequence function \( f \) is a set \( N \) of natural numbers.

Observation 2
We can enumerate elements of a sequence by any infinite subset of \( N \).
We usually take a set \( N - \{0\} \) as a sequence domain (enumeration).
Observations

Observation 3
We can choose as a set of indexes of a sequence any countably infinite set $T$, i.e., not only the set $N$ of natural numbers.

We often choose $T = N - \{0\} = N^+$, i.e. we consider sequences that "start" with $n = 1$.
In this case we write sequences as

$$a_1, a_2, a_3, \ldots, a_n, \ldots \ldots$$
Finite Sequences

Finite Sequence
Given a finite set $K = \{1, 2, \ldots, n\}$, for $n \in \mathbb{N}$ and any set $A$
Any function $f : \{1, 2, \ldots, n\} \rightarrow A$

is called a \textbf{finite sequence} of elements of the set $A$
of the length $n$

Case $n=0$
In this case the function $f$ is an empty set and we call it an\textbf{empty sequence}
We denote the \textbf{empty sequence} by $e$
Example

Consider a sequence given by a formula

\[ a_n = \frac{n}{(n - 2)(n - 5)} \]

The domain of the function \( f(n) = a_n \) is the set \( N - \{2, 5\} \) and the sequence \( f \) is a function

\[ f : N - \{2, 5\} \rightarrow \mathbb{R} \]

The first elements of the sequence \( f \) are

\[ a_0 = f(0), \ a_1 = f(1), \ a_3 = f(3), \ a_4 = f(4) \ a_5 = f(5), \ a_6 = f(6), \ldots \]
Example

Let \( T = \{-1, -2, 3, 4\} \) be a finite set and

\[
f : \{-1, -2, 3, 4\} \rightarrow \mathbb{R}
\]

be a function given by a formula

\[
f(n) = a_n = \frac{n}{(n-2)(n-5)}
\]

\( f \) is a finite sequence of length 4 with elements

\[
a_{-1} = f(-1), \quad a_{-2} = f(-2), \quad a_3 = f(3), \quad a_4 = f(4)
\]
Families of Sets Revisited

Family of sets
Any collection of sets is called a family of sets
We denote the family of sets by
\[ F \]

Sequence of sets
Any function
\[ f : \mathbb{N} \rightarrow F \]
is a sequence of sets, i.e., a sequence where all its elements are sets
We use capital letters to denote sets and write the sequence of sets as
\[ \{A_n\}_{n \in \mathbb{N}} \]
Generalized Union

Given a sequence \( \{A_n\}_{n \in \mathbb{N}} \) of sets, we define the Generalized Union of the sequence of sets as

\[
\bigcup_{n \in \mathbb{N}} A_n = \{x : \exists n \in \mathbb{N} \ x \in A_n\}
\]

We write \( x \in \bigcup_{n \in \mathbb{N}} A_n \) if and only if \( \exists n \in \mathbb{N} \ x \in A_n \).
Generalized Intersection

Given a sequence \( \{A_n\}_{n \in N} \) of sets, we define that **Generalized Intersection** of the sequence of sets as

\[
\bigcap_{n \in N} A_n = \{x : \forall n \in N \ x \in A_n\}
\]

We write

\[
x \in \bigcap_{n \in N} A_n \quad \text{if and only if} \quad \forall n \in N \ x \in A_n
\]
Indexed Family of Sets

Given \( \mathcal{F} \) be a family of sets

Let \( T \neq \emptyset \) be any non-empty set

Any function

\[
f : T \rightarrow \mathcal{F}
\]

is called an indexed family of sets with the set of indexes \( T \)

We write it

\[
\{A_t\}_{t \in T}
\]

Notice

Any sequence of sets is an indexed family of sets for \( T = \mathbb{N} \)
Chapter 1

Some Simple Questions and Answers
Simple Short Questions

Here are some short Yes/No questions
Answer them and write a short justification of your answer

Q1 \( 2^{\{1,2\}} \cap \{1, 2\} \neq \emptyset \)

Q2 \( \{\{a, b\}\} \in 2^{\{a,b,\{a,b\}\}} \)

Q3 \( \emptyset \in 2^{\{a,b,\{a,b\}\}} \)

Q4 Any function \( f \) from \( A \neq \emptyset \) onto \( A \), has property

\[ f(a) \neq a \text{ for certain } a \in A \]
Simple Short Questions

Q5 Let \( f : \mathbb{N} \rightarrow \mathcal{P}(\mathbb{N}) \) be given by a formula:

\[
f(n) = \{ m \in \mathbb{N} : m < n^2 \}
\]

then \( \emptyset \in f[\{0, 1, 2\}] \)

Q6 Some relations \( R \subseteq A \times B \)

are functions that map the set \( A \) into the set \( B \)
Answers to Short Questions

Q1 \(2^{\{1, 2\}} \cap \{1, 2\} \neq \emptyset\)
\[\text{NO because} \quad 2^{\{1, 2\}} = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\} \cap \{1, 2\} = \emptyset\]

Q2 \\{a, b\}\} \in 2^{\{a, b, \{a, b\}}
\[\text{YES because} \quad \{a, b\} \subseteq \{a, b, \{a, b\}\} \quad \text{and hence}\]
\[\\{a, b\}\} \in 2^{\{a, b, \{a, b\}}\]
by definition of the set of all subsets of a given set
Answers to Short Questions

Q2 \( \{\{a, b\}\} \in 2^{\{a, b, \{a, b\}\}} \)

YES other solution

We list all subsets of the set \( \{a, b, \{a, b\}\} \), i.e. all elements of the set

\[ 2^{\{a, b, \{a, b\}\}} \]

We start as follows

\[ \emptyset, \{a\}, \{b\}, \{\{a, b\}\}, \ldots, \ldots \]

and observe that we can stop listing because we reached the set \( \{\{a, b\}\} \)

This proves that \( \{\{a, b\}\} \in 2^{\{a, b, \{a, b\}\}} \)
Q3  ∅ ∈ 2^{a,b,\{a,b\}}

YES  because for any set  A,  we have that  ∅ ⊆ A

Q4  Any function  f  from  A ≠ ∅  onto  A  has a property

  \[ f(a) \neq a \quad \text{for certain} \quad a \in A \]

NO
Take a function such that  f(a) = a  for all  a \in A
Obviously  f  is "onto" and and  there is no  a \in A
for which  f(a) ≠ a
Answers to Short Questions

Q5  Let $f : N \rightarrow \mathcal{P}(N)$ be given by formula:
$f(n) = \{m \in N : m < n^2\}$, then $\emptyset \in f[\{0, 1, 2\}]$

YES  We evaluate
$f(0) = \{m \in N : m < 0\} = \emptyset$
$f(1) = \{m \in N : m < 1\} = \{0\}$
$f(2) = \{m \in N : m < 2^2\} = \{0, 1, 2, 3\}$
and so by definition of $f[A]$ get that
$f[\{0, 1, 2\}] = \{\emptyset, \{0\}, \{0, 1, 2, 3\}\}$ and hence $\emptyset \in f[\{0, 1, 2\}]$

Q6  Some $R \subseteq A \times B$ are functions that map $A$ into $B$

YES: Functions are special type of relations
Simple Short Questions

Q7 \(((1, 2), (a, 1))\) is a binary relation on \(\{1, 2\}\)

Q8 For any binary relation \(R \subseteq A \times A\), the inverse relation \(R^{-1}\) exists

Q9 For any binary relation \(R \subseteq A \times A\) that is a function, the inverse function \(R^{-1}\) exists
Simple Short Questions

Q10  Let \( A = \{ a, \{ a \}, \emptyset \} \) and \( B = \{ \emptyset, \{ \emptyset \}, \emptyset \} \)
there is a function \( f : A \rightarrow_{1-1}^{\text{onto}} B \)

Q11  Let \( f : A \rightarrow B \) and \( g : B \rightarrow^{\text{onto}} A \),
then the compositions \( (g \circ f) \) and \( (f \circ g) \) exist

Q12  The function \( f : \mathbb{N} \rightarrow \mathcal{P}(\mathbb{R}) \) given by the formula:
\[
f(n) = \{ x \in \mathbb{R} : x > \frac{\ln(n^3 + 1)}{\sqrt{n + 6}} \}
\]
is a sequence
Answers to Short Questions

Q7 \[\{(1, 2), (a, 1)\}\] is a binary relation on \{1, 2\}
NO because \((a, 1) \notin \{1, 2\} \times \{1, 2\}\)

Q8 For any binary relation \(R \subseteq A \times A\), the inverse relation \(R^{-1}\) exists
YES By definition, the inverse relation to \(R \subseteq A \times A\) is the set
\[R^{-1} = \{(b, a) : (a, b) \in R\}\]
and it is a well defined relation in the set \(A\)
Answers to Short Questions

Q9  For any binary relation $R \subseteq A \times A$ that is a function, the inverse function $R^{-1}$ exists.

NO  $R$ must be also a 1–1 and onto function.

Q10  Let $A = \{a, \{a\}, \emptyset\}$ and $B = \{\emptyset, \{\emptyset\}, \emptyset\}$.

there is a function $f : A \rightarrow_{onto}^{1-1} B$.

NO  The set $A$ has 3 elements and the set $B = \{\emptyset, \{\emptyset\}, \emptyset\} = \{\emptyset, \{\emptyset\}\}$ has 2 elements and an onto function does not exists.
Q11 Let \( f : A \rightarrow B \) and \( g : B \rightarrow^\text{onto} A \), then the **compositions** \( (g \circ f) \) and \( (f \circ g) \) **exist**

YES The composition \( (f \circ g) \) **exists** because the functions \( f : A \rightarrow B \) and \( g : B \rightarrow^\text{onto} A \) share the same set \( B \)

The composition \( (g \circ f) \) **exists** because the functions \( g : B \rightarrow^\text{onto} A \) and \( f : A \rightarrow B \) share the same set \( A \)

The information ”onto” is **irrelevant**
Q12  The function \( f : \mathbb{N} \rightarrow \mathcal{P}(\mathbb{R}) \) given by the formula:

\[
f(n) = \{ x \in \mathbb{R} : x > \frac{\ln(n^3 + 1)}{\sqrt{n + 6}} \}
\]

is a **sequence**

**YES**  It is a sequence as the **domain** of the function \( f \) is the set \( \mathbb{N} \) of natural numbers and the formula for \( f(n) \) assigns to each natural number \( n \) a certain **subset** of \( \mathbb{R} \), i.e. an **element** of \( \mathcal{P}(\mathbb{R}) \)
Chapter 1
Sets, Relations, and Languages

Slides Set 1
PART 3: Special types of Binary Relations

SPECIAL RELATION: Equivalence Relation
Equivalence Relation

Equivalence relation
A binary relation \( R \subseteq A \times A \) is an equivalence relation defined in the set \( A \) if and only if it is reflexive, symmetric, and transitive.

Symbols
We denote equivalence relation by symbols \( \sim, \approx, \equiv \)

We will use the symbol \( \approx \) to denote the equivalence relation.
Equivalence Relation

Equivalence class
Let \( \approx \subseteq A \times A \) be an equivalence relation on \( A \)
The set
\[
E(a) = \{ b \in A : \ a \approx b \}
\]
is called an equivalence class

Symbol
The equivalence classes are usually denoted by
\[
[a] = \{ b \in A : \ a \approx b \}
\]
The element \( a \) is called a representative of the equivalence class \([a]\) defined in \( A\)
Partitions

Partition

A family of sets $\mathcal{P} \subseteq \mathcal{P}(A)$ is called a partition of the set $A$ if and only if the following conditions hold

1. $\forall X \in \mathcal{P} \ (X \neq \emptyset)$
   i.e. all sets in the partition are non-empty

2. $\forall X, Y \in \mathcal{P} \ (X \cap Y = \emptyset)$
   i.e. all sets in the partition are disjoint

3. $\bigcup \mathcal{P} = A$
   i.e. union of all sets from $\mathcal{P}$ is the set $A$
Equivalence and Partitions

Notation

$A / \equiv$ denotes the set of all equivalence classes of the equivalence relation $\equiv$, i.e.

$$A / \equiv = \{ [a] : a \in A \}$$

We prove the following theorem 1.3.1

Theorem 1

Let $A \neq \emptyset$

If $\equiv$ is an equivalence relation on $A$, then the set $A / \equiv$ is a partition of $A$
Theorem 1 (full statement)

Let \( A \neq \emptyset \)

If \( \approx \) is an equivalence relation on \( A \),
then the set \( A/\approx \) is a partition of \( A \), i.e.

1. \( \forall [a] \in A/\approx \) \( ([a] \neq \emptyset) \)
   i.e. all equivalence classes are non-empty
2. \( \forall [a] \neq [b] \in A/\approx \) \( ([a] \cap [b] = \emptyset) \)
   i.e. all different equivalence classes are disjoint
3. \( \bigcup A/\approx = A \)
   i.e the union of all equivalence classes is equal to the set \( A \)
Partition and Equivalence

We also prove a following

**Theorem 2**

For any partition \( P \subseteq \mathcal{P}(A) \) of the set \( A \), one can construct a binary relation \( R \) on \( A \) such that \( R \) is an equivalence on \( A \) and its equivalence classes are exactly the sets of the partition \( P \).
Partition and Equivalence

Observe that we can consider, for any binary relation $R$ on set $A$ the sets that "look" like equivalence classes i.e. that are defined as follows:

$$R(a) = \{ b \in A; \ a R b \} = \{ b \in A; \ (a, b) \in R \}$$

Fact 1
If the relation $R$ is an equivalence on $A$, then the family $\{ R(a) \}_{a \in A}$ is a partition of $A$

Fact 2
If the family $\{ R(a) \}_{a \in A}$ is not a partition of $A$, then $R$ is not an equivalence on $A$
Proof of Theorem 1

Theorem 1
Let $A \neq \emptyset$
If $\approx$ is an equivalence relation on $A$,
then the set $A/\approx$ is a partition of $A$

Proof
Let $A/\approx = \{[a] : a \in A\} = P$
We must show that all sets in $P$ are nonempty, disjoint, and together exhaust the set $A$
Proof of Theorem 1

1. All equivalence classes are nonempty,
   This holds as \( a \in [a] \) for all \( a \in A \), reflexivity of equivalence relation

2. All different equivalence classes are disjoint
   Consider two different equivalence classes \([a] \neq [b]\)
   Assume that \([a] \cap [b] \neq \emptyset\).
   We have that \([a] \neq [b]\), thus there is an element \(c\)
   such that \(c \in [a]\) and \(c \in [b]\)
   Hence \((a, c) \in \approx\) and \((c, b) \in \approx\)
   Since \(\approx\) is transitive, we get \((a, b) \in \approx\)
Proof of Theorem 1

Since \( \approx \) is symmetric, we have that also \( (a, b) \in \approx \).

Now take any element \( d \in [a] \); then \( (d, a) \in \approx \), and by transitivity, \( (d, b) \in \approx \).
Hence \( d \in [b] \), so that \( [a] \subseteq [b] \).

Likewise \( [b] \subseteq [a] \) and \( [a] = [b] \) what contradicts the assumption that \( [a] \neq [b] \).
Proof of Theorem 1

3. To prove that
\[ \bigcup \mathcal{A} / \approx = \bigcup \mathcal{P} = A \]
we simply notice that each element \( a \in A \) is in some set in \( \mathcal{P} \).
Namely, we have by reflexivity that always
\[ a \in [a] \]
This ends the proof of Theorem 1
Proof of the Theorem 2

Now we are going to prove that the Theorem 1 can be reversed, namely that the following is also true

Theorem 2
For any partition $P \subseteq \mathcal{P}(A)$ of $A$, one can construct a binary relation $R$ on $A$ such that $R$ is an equivalence and its equivalence classes are exactly the sets of the partition $P$.

Proof
We define a binary relation $R$ as follows

$$R = \{(a, b) : a, b \in X \text{ for some } X \in P\}$$
Chapter 1
Sets, Relations, and Languages

Slides Set 1

PART 3: Equivalence Relations - Short and Long Questions
Short Questions

Q1 Let $R \subseteq A \times A$ for $A \neq \emptyset$, then the set

$$[a] = \{b \in A : (a, b) \in R\}$$

is an equivalence class with a representative $a$.

Q2 The set

$$\{(\emptyset, \emptyset), (\{a\}, \{a\}), (3, 3)\}$$

represents a transitive relation.
Short Questions

**Q3** There is an **equivalence** relation on the set

\[ A = \{\{0\}, \{0, 1\}, 1, 2\} \]

with **3** equivalence classes

**Q4** Let \( A \neq \emptyset \) be such that there are exactly **25 partitions** of \( A \)

It is **possible** to define **20 equivalence** relations on \( A \)
Short Questions Answers

Q1  Let \( R \subseteq A \times A \) then the set

\[
[a] = \{ b \in A : (a, b) \in R \}
\]

is an **equivalence** class with a **representative** \( a \)

**NO**  The set \([a] = \{ b \in A : (a, b) \in R \}\) is an equivalence class **only** when the relation \( R \) is an **equivalence** relation

Q2  The set

\[
\{ (\emptyset, \emptyset), (\{a\}, \{a\}), (3, 3) \}
\]

represents a **transitive** relation

**YES**  Transitivity condition is **vacuously** true
Short Questions Answers

Q3  There is an equivalence relation on

\[ A = \{\{0\}, \{0, 1\}, 1, 2\} \]

with 3 equivalence classes

YES  For example, a relation \( R \) defined by the partition

\[ P = \{\{0\}, \{0, 1\}, \{1, 2\}\} \]

and so By proof of Theorem 2

\[ R = \{(a, b) : a, b \in X \text{ for some } X \in P\} \]

i.e. \( a = b = \{0\} \) or \( a = b = \{0, 1\} \) or \( (a = 1 \text{ and } b= 2) \)
Q4
Let $A \neq \emptyset$ be such that there are exactly 25 partitions of $A$
It is possible to define 2 equivalence relations on $A$

**YES** By Theorem 2 one can define up to 25 (as many as partitions) of equivalence classes
Equivalence Relations

Some Long Questions
Some Long Questions

Q1  Consider a function \( f : A \rightarrow B \)

Show that \( R = \{(a, b) \in A \times A : f(a) = f(b)\} \)
is an equivalence relation on \( A \)

Q2  Let \( f : N \rightarrow N \) be such that

\[
f(n) = \begin{cases} 
1 & \text{if } n \leq 6 \\
2 & \text{if } n > 6 
\end{cases}
\]

Find equivalence classes of \( R \) from Q1 for this particular function \( f \)
Q1  Consider a function  \( f : A \rightarrow B \)

Show that

\[ R = \{(a, b) \in A \times A : f(a) = f(b)\} \]

is an equivalence relation on \( A \)

Solution

1. \( R \) is reflexive

\( (a, a) \in R \) for all \( a \in A \) because \( f(a) = f(a) \)
Long Questions Solutions

2. \( R \) is symmetric

Let \((a, b) \in R\), by definition \( f(a) = f(b) \) and \( f(b) = f(a) \)

Consequently \((b, a) \in R\)

3. \( R \) is transitive

For any \( a, b, c \in A \) we get that \( f(a) = f(b) \) and \( f(b) = f(c) \)

implies that \( f(a) = f(c) \)
Q2  Let \( f : N \rightarrow N \) be such that

\[
f(n) = \begin{cases} 
1 & \text{if } n \leq 6 \\
2 & \text{if } n > 6 
\end{cases}
\]

Find equivalence classes of

\[
R = \{(a, b) \in A \times A : f(a) = f(b)\}
\]

for this particular \( f \)
Solution
We evaluate

\[ [0] = \{ n \in N : f(0) = f(n) \} = \{ n \in N : f(n) = 1 \} = \{ n \in N : n \leq 6 \} \]

\[ [7] = \{ n \in N : f(7) = f(n) \} = \{ n \in N : f(n) = 2 \} = \{ n \in N : n > 6 \} \]

There are two equivalence classes:

\[ A_1 = \{ n \in N : n \leq 6 \}, \quad A_2 = \{ n \in N : n > 6 \} \]
Chapter 1
Sets, Relations, and Languages

Slides Set 1
PART 3: Special types of Binary Relations

SPECIAL RELATIONS: Order Relations
Order Relations

We introduce now of another type of important binary relations: the order relations

**Definition**

\( R \subseteq A \times A \) is an order relation on \( A \) iff \( R \) is 1. Reflexive, 2. Antisymmetric, and 3. Transitive, i.e. the following conditions are satisfied

1. \( \forall a \in A (a, a) \in R \)
2. \( \forall a, b \in A ((a, b) \in R \cap (b, a) \in R \Rightarrow a = b) \)
3. \( \forall a, b, c \in A ((a, b) \in R \cap (b, c) \in R \Rightarrow (a, c) \in R) \)
Order Relations

Definition

$R \subseteq (A \times A)$ is a total order on $A$ iff $R$ is an order and any two elements of $A$ are comparable, i.e. additionally the following condition is satisfied

4. $\forall_{a,b \in A} ((a, b) \in R \cup (b, a) \in R)$

Names: order relation is also called historically a partial order
total order is also called historically a linear order
Order Relations

**Notations**

Order relations are usually denoted by \( \leq \), or when we want to make a clear distinction from the natural order in sets of numbers, we denote it by \( \preceq \).

**Remember**, that even if we use \( \leq \) as the order relation symbol, it is a **symbol for ANY order relation** and not only a symbol for a natural order \( \leq \) in sets of numbers.
Posets

A set $A \neq \emptyset$ ordered by an order relation $R$ is called a poset. We write it as a tuple (depending of symbols used) $(A, R), (A, \leq), (A, \preceq)$.

Name poset stands historically for "partially ordered set".

Diagram of order relation is a graphical representation of a poset. It is a simplified graph constructed as follows.

1. As the order relation is reflexive, i.e. $(a, a) \in R$ for all $a \in A$, we draw a point with symbol $a$ instead of a point with symbol $a$ and the loop.

2. As the order relation is antisymmetric we draw a point $b$ above point $a$ (connected, but without the arrow) to indicate that $(a, b) \in R$.

3. As the order relation is transitive, we connect points $a, b, c$ without arrows.
Posets Special Elements

**Special elements** in a poset \((A, \leq)\) are: maximal, minimal, greatest (largest) and smallest (least) and are defined below.

**Smallest (least)** \(a_0 \in A\) is a smallest (least) element in the poset \((A, \leq)\) iff \(\forall a \in A (a_0 \leq a)\)

**Greatest (largest)** \(a_0 \in A\) is a greatest (largest) element in the poset \((A, \leq)\) iff \(\forall a \in A (a \leq a_0)\)
Posets Special Elements

Maximal (formal) \( a_0 \in A \) is a maximal element in the poset \((A, \leq)\) iff \( \neg \exists_{a \in A} (a_0 \leq a \land a_0 \neq a) \)

Maximal (informal) \( a_0 \in A \) is a maximal element in the poset \((A, \leq)\) iff on a diagram of \((A, \leq)\) there is no element placed above \( a_0 \)

Minimal (formal) \( a_0 \in A \) is a minimal element in the poset \((A, \leq)\) iff \( \neg \exists_{a \in A} (a \leq a_0 \land a_0 \neq a) \)

Minimal (informal) \( a_0 \in A \) is a minimal element in the poset \((A, \leq)\) iff on the diagram of \((A, \leq)\) there is no element placed below \( a_0 \)
Some Properties of Posets

Use **Mathematical Induction** to prove the following property of finite posets

**Property 1** Every non-empty finite poset has at least one maximal element

**Proof**

Let \((A, \leq)\) be a finite, not empty poset (partially ordered set by \(\leq\), such that \(A\) has \(n\)-elements, i.e. \(|A| = n\)

We carry the Mathematical Induction over \(n \in N - \{0\}\)

**Reminder:** an element \(a_o \in A\) ia a maximal element in a poset \((A, \leq)\) iff the following is true.

\[ \neg \exists_{a \in A} (a_0 \neq a \land a_0 \leq a) \]
Inductive Proof

**Base case:**  \( n = 1 \), so \( A = \{a\} \) and \( a \) is maximal (and minimal, and smallest, and largest) in the poset \((\{a\}, \leq)\)

**Inductive step:** Assume that any set \( A \) such that \(|A| = n\) has a maximal element;
Denote by \( a_0 \) the maximal element in \((A, \leq)\)
Let \( B \) be a set with \( n + 1 \) elements; i.e. we can write \( B \) as \( B = A \cup \{b_0\} \) for \( b_0 \not\in A \), for some \( A \) with \( n \) elements
Inductive Proof

By **Inductive Assumption** the poset \((A, \leq)\) has a maximal element \(a_0\)

To show that \((B, \leq)\) has a maximal element we need to consider 3 cases.

1. \(b_0 \leq a_0\); in this case \(a_0\) is also a maximal element in \((B, \leq)\)
2. \(a_0 \leq b_0\); in this case \(b_0\) is a new maximal in \((B, \leq)\)
3. \(a_0, b_0\) are not compatible; in this case \(a_0\) remains maximal in \((B, \leq)\)

By Mathematical Induction we have proved that
\[
\forall n \in \mathbb{N} \setminus \{0\} (|A| = n \Rightarrow A \text{ has a maximal element})
\]
Some Properties of Posets

We just proved

Property 1 Every non-empty finite poset has at least one maximal element

Show that the Property 1 is not true for an infinite set

Solution: Consider a poset \((\mathbb{Z}, \leq)\), where \(\mathbb{Z}\) is the set on integers and \(\leq\) is a natural order on \(\mathbb{Z}\). Obviously no maximal element!

Exercise: Prove

Property 2 Every non-empty finite poset has at least one minimal element

Show that the Property 2 is not true for an infinite set
Chapter 1
Sets, Relations, and Languages

Slides Set 2
PART 4: Finite and Infinite Sets
PART 5: Fundamental Proof Techniques
Equinumerous Sets

Equinumerous sets
We call two sets $A$ and $B$ are equinumerous if and only if there is a bijective function $f : A \rightarrow B$, i.e. there is $f$ is such that

$$f : A \overset{1-1,onto}{\rightarrow} B$$

Notation
We write $A \sim B$ to denote that the sets $A$ and $B$ are equinumerous and write symbolically

$$A \sim B \text{ if and only if } f : A \overset{1-1,onto}{\rightarrow} B$$
Equinumerous Relation

Observe that for any set $X$, the relation $\sim$ is an equivalence on the set $2^X$, i.e.

$$\sim \subseteq 2^X \times 2^X$$

is reflexive, symmetric and transitive and for any set $A$ the equivalence class

$$[A] = \{B \in 2^X : A \sim B \}$$

describes for finite sets all sets that have the same number of elements as the set $A$.
Equinumerous Relation

**Observe** also that the relation $\sim$ when considered for any sets $A, B$ is **not** an equivalence relation as its domain would have to be the set of all sets that does not exist.

We extend the notion of "the same number of elements" to any sets by introducing the notion of cardinality of sets.
Cardinality of Sets

Cardinality definition
We say that $A$ and $B$ have the same **cardinality** if and only if they are equipotent, i.e.

$$A \sim B$$

Cardinality notations
If sets $A$ and $B$ have the same **cardinality** we denote it as:

$$|A| = |B| \text{ or } \text{card}A = \text{card}B$$
Cardinality of Sets

Cardinality
We put the above together in one definition

\[ |A| = |B| \text{ if and only if } \]
\[ \text{there is a function } f \text{ is such that} \]
\[ f : A \rightarrow B \]
Finite and Infinite Sets

Definition
A set $A$ is finite if and only if there is $n \in \mathbb{N}$ and there is a function

$$f : \{0, 1, 2, \ldots, n - 1\} \rightarrow A$$

$1-1$, onto

In this case we have that

$$|A| = n$$

and say that the set $A$ has $n$ elements
Finite and Infinite Sets

**Definition**
A set $A$ is **infinite** if and only if $A$ is not finite

Here is a theorem that characterizes infinite sets

**Dedekind Theorem**
A set $A$ is **infinite** if and only if there is a **proper** subset $B$ of the set $A$ such that

$|A| = |B|$
Infinite Sets Examples

E1 Set $N$ of natural numbers is infinite

Consider a function $f$ given by a formula

$$f(n) = 2n \text{ for all } n \in N$$

Obviously

$$f : N \xrightarrow{1-1,onto} 2N$$

By Dedekind Theorem the set $N$ is infinite as the set $2N$ of even numbers are a proper subset of natural numbers $N$
E2  Set $\mathbb{R}$ of real numbers is infinite

Consider a function $f$ given by a formula

$$f(x) = 2^x \quad \text{for all} \quad x \in \mathbb{R}$$

Obviously

$$f : \mathbb{R} \xrightarrow{1-1,onto} \mathbb{R}^+$$

By Dedekind Theorem the set $\mathbb{R}$ is infinite as the set $\mathbb{R}^+$ of positive real numbers are a proper subset of real numbers $\mathbb{R}$
Countably Infinite Sets
Cardinal Number $\aleph_0$

Definition
A set $A$ is called **countably infinite** if and only if it has the same **cardinality** as the set $N$ natural numbers, i.e. when

$$|A| = |N|$$

The **cardinality** of natural numbers $N$ is called $\aleph_0$ (Aleph zero) and we write

$$|N| = \aleph_0$$
Countably Infinite Sets

Definition
For any set $A$,

$$|A| = \aleph_0 \quad \text{if and only if} \quad |A| = |N|$$

Directly from definitions we get the following

Fact 1
A set $A$ is **countably infinite** if and only if $|A| = \aleph_0$
Countably Infinite Sets

Fact 2
A set $A$ is **countably infinite** if and only if all elements of $A$ can be put in a 1-1 sequence.

Other name for **countably infinite** set is **infinitely countable** set and we will use both names.
Countably Infinite Sets

In a case of an infinite set $A$ and not 1-1 sequence we can "prune" all repetitive elements to get a 1-1 sequence, i.e. we prove the following

**Fact 2a**
An infinite set $A$ is **countably infinite** if and only if all elements of $A$ can be put in a sequence
Countable and Uncountable Sets

Definition
A set $A$ is **countable** if and only if $A$ is finite or countably infinite.

Fact 3
A set $A$ is **countable** if and only if $A$ is finite or $|A| = \aleph_0$, i.e. $|A| = |\mathbb{N}|$.
Countable and Uncountable Sets

Definition
A set $A$ is **uncountable** if and only if $A$ is not **countable**

Fact 4
A set $A$ is **uncountable** if and only if $A$ is infinite and $|A| \neq \aleph_0$, i.e. $|A| \neq |\mathbb{N}|$

Fact 5
A set $A$ is **uncountable** if and only if its elements **cannot** be put into a sequence

Proof proof follows directly from definition and Facts 2, 4
Countably Infinite Sets

We have proved the following

**Fact 2a**
An infinite set $A$ is **countably infinite** if and only if all elements of $A$ can be put in a sequence

We use it now to prove two **theorems** about **countably infinite** sets
Countably Infinite Sets

Union Theorem
Union of two countably infinite sets is a countably infinite set

Proof
Let $A$, $B$ be two disjoint infinitely countable sets
By Fact 2 we can list their elements as 1-1 sequences

$$A : a_0, a_1, a_2, \ldots$$
$$B : b_0, b_1, b_2, \ldots$$

and their union can be listed as 1-1 sequence

$$A \cup B : a_0, b_0, a_1, b_1, a_2, b_2, \ldots$$

In a case not disjoint sets we proceed the same and then ”prune” all repetitive elements to get a 1-1 sequence
Countably Infinite Sets

Product Theorem
Cartesian Product of two countably infinite sets is a countably infinite set

Proof
Let A, B be two infinitely countable sets
By Fact 2 we can list their elements as 1-1 sequences

\[ A : a_0, a_1, a_2, \ldots \quad \text{and} \quad B : b_0, b_1, b_2, \ldots \]

We list their Cartesian Product \( A \times B \) as an infinite table
\[
(a_0, b_0), (a_0, b_1), (a_0, b_2), (a_0, b_3), \ldots \\
(a_1, b_0), (a_1, b_1), (a_1, b_2), (a_1, b_3), \ldots \\
(a_2, b_0), (a_2, b_1), (a_2, b_2), (a_2, b_3), \ldots \\
(a_3, b_0), (a_3, b_1), (a_3, b_2), (a_3, b_3), \ldots \\
\ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots,
Cartesian Product Theorem Proof

Observe that even if the table is infinite, each of its diagonals is finite.

\[(a_0, b_0), (a_0, b_1), (a_0, b_2), (a_0, b_3), (a_0, b_4), \ldots, \ldots\]
\[(a_1, b_0), (a_1, b_1), (a_1, b_2), (a_1, b_3), \ldots\]
\[(a_2, b_0), (a_2, b_1), (a_2, b_2), (a_2, b_3), \ldots\]
\[(a_3, b_0), (a_3, b_1), (a_3, b_2), (a_3, b_3), \ldots\]
\[\ldots, \ldots, \ldots, \ldots, \ldots,\]

We list now elements of \(A \times B\) one diagonal after the other. Each diagonal is finite, so now we know when one finishes and other starts.
Cartesian Product Theorem Proof

$A \times B$ becomes now the following sequence

$$(a_0, b_0),$$
$$ (a_1, b_0), (a_0, b_1),$$
$$ (a_2, b_0), (a_1, b_1), (a_0, b_2),$$
$$ (a_3, b_0), (a_2, b_1), (a_1, b_2), (a_0, b_3),$$
$$ (a_3, b_1), (a_2, b_2), (a_1, b_3), (a_0, b_4), \ldots,$$
$$ \ldots, \ldots, \ldots, \ldots,$$

We prove by Mathematical induction that the sequence is well defined for all $n \in N$ and hence that $|A \times B| = |N|$

It ends the proof of the Product Theorem
Observe that the both **Union** and **Product Theorems** can be generalized by **Mathematical Induction** to the case of **Union** or **Cartesian Products** of **any finite** number of sets.
Theorem 1
The set $\mathbb{R}$ of real numbers is uncountable

Proof
We first prove (homework problem 1.5.11) the following

Lemma 1
The set of all real numbers in the interval $[0,1]$ is uncountable

Then we use the Lemma 2 below (to be proved it as an exercise) and the fact that $[0,1] \subseteq \mathbb{R}$ and this ends the proof

Lemma 2
For any sets $A, B$ such that $B \subseteq A$ and $B$ is uncountable we have that also the set $A$ is uncountable
Special Uncountable Sets

Cardinal Number $C$ - Continuum
We denote by $C$ the cardinality of the set $\mathbb{R}$ of real numbers $C$ is a new **cardinal number** called **continuum** and we write

$$|\mathbb{R}| = C$$

**Definition**
We say that a set $A$ has **cardinality** $C$ (continuum) if and only if $|A| = |\mathbb{R}|$
We write it

$$|A| = C$$
Sets of Cardinality $C$

Example
The set of positive real numbers $R^+$ has cardinality $C$ because a function $f$ given by the formula

$$f(x) = 2^x \text{ for all } x \in R$$

is 1-1 function and maps $R$ onto the set $R^+$
Sets of Cardinality $C$

**Theorem 2**
The set $2^\mathbb{N}$ of all subsets of natural numbers is **uncountable**

**Proof**
We will prove it in the PART 5.

**Theorem 3**
The set $2^\mathbb{N}$ has cardinality $C$, i.e.

$$|2^\mathbb{N}| = C$$

**Proof**
The proof of this theorem is not trivial and is not in the scope of this course
Cantor Theorem

Cantor Theorem (1891)

For any set $A$,

$|A| < |2^A|$

where we define

$|A| \leq |B|$ if and only if $A \sim C$ and $C \subseteq B$

$|A| < |B|$ if and only if $|A| \leq |B|$ and $|A| \neq |B|$
Cantor Theorem

Directly from the definition we have the following

**Fact 6**
If $A \subseteq B$ then $|A| \leq |B|$

We know that $|N| = \aleph_0$, $C = |R|$, and $N \subseteq R$ hence from Fact 6, $\aleph_0 \leq C$, but $\aleph_0 \neq C$, as the set $N$ is **countable** and the set $R$ is **uncountable**

Hence we proved

**Fact 7**

$\aleph_0 < C$
Uncountable Sets of Cardinality Greater than $C$

By **Cantor Theorem** we have that

$$|N| < |\mathcal{P}(N)| < |\mathcal{P}(\mathcal{P}(N))| < |\mathcal{P}(\mathcal{P}(\mathcal{P}(N)))| < \ldots$$

All sets

$$\mathcal{P}(\mathcal{P}(N)), \mathcal{P}(\mathcal{P}(\mathcal{P}(N))) \ldots$$

are uncountable with cardinality greater than $C$, as by Theorem 3, Fact 7, and **Cantor Theorem** we have that

$$\aleph_0 < C < |\mathcal{P}(\mathcal{P}(N))| < |\mathcal{P}(\mathcal{P}(\mathcal{P}(N)))| < \ldots$$
Countable and Uncountable Sets

Here are some basic **Theorem** and **Facts**

**Union 1**
Union of two infinitely countable \((\text{of cardinality } \aleph_0)\) sets is an infinitely countable set
This means that
\[
\aleph_0 + \aleph_0 = \aleph_0
\]

**Union 2**
Union of a finite \((\text{of cardinality } n)\) set and infinitely countable \((\text{of cardinality } \aleph_0)\) set is an infinitely countable set
This means that
\[
\aleph_0 + n = \aleph_0
\]
Countable and Uncountable Sets

**Union 3**
Union of an infinitely countable (of cardinality \( \aleph_0 \)) set and a set of the same cardinality as real numbers i.e. of the cardinality \( C \) has the same cardinality as the set of real numbers.
This means that
\[
\aleph_0 + C = C
\]

**Union 4** Union of two sets of cardinality the same as real numbers (of cardinality \( C \)) has the same cardinality as the set of real numbers.
This means that
\[
C + C = C
\]
Countable and Uncountable Sets

Product 1
Cartesian Product of two infinitely countable sets is an infinitely countable set

\[ \aleph_0 \cdot \aleph_0 = \aleph_0 \]

Product 2
Cartesian Product of a non-empty finite set and an infinitely countable set is an infinitely countable set

\[ n \cdot \aleph_0 = \aleph_0 \quad \text{for} \quad n > 0 \]
Countable and Uncountable Sets

Product 3
Cartesian Product of an infinitely countable set and an uncountable set of cardinality $C$ has the cardinality $C$

$$\aleph_0 \cdot C = C$$

Product 4
Cartesian Product of two uncountable sets of cardinality $C$ has the cardinality $C$

$$C \cdot C = C$$
Countable and Uncountable Sets

Power 1
The set $2^\mathbb{N}$ of all subsets of natural numbers (or of any countably infinite set) is uncountable set of cardinality $\mathcal{C}$, i.e. has the same cardinality as the set of real numbers

$$2^{\aleph_0} = \mathcal{C}$$

Power 2
There are $\mathcal{C}$ of all functions that map $\mathbb{N}$ into $\mathbb{N}$

Power 3
There are $\mathcal{C}$ possible sequences that can be form out of an infinitely countable set

$$\aleph_0^{\aleph_0} = \mathcal{C}$$
Countable and Uncountable Sets

Power 4
The set of all functions that map \( \mathbb{R} \) into \( \mathbb{R} \) has the cardinality \( C^C \).

Power 5
The set of all real functions of one variable has the same cardinality as the set of all subsets of real numbers

\[ C^C = 2^C \]
Countable and Uncountable Sets

Theorem 4

\[ n < \aleph_0 < C \]

Theorem 5

For any non empty, finite set \( A \), the set \( A^* \) of all \textbf{finite sequences} formed out of \( A \) is \textbf{countably infinite}, i.e

\[ |A^*| = \aleph_0 \]

We write it as

If \( |A| = n, \ n \geq 1 \), then \( |A^*| = \aleph_0 \)
Simple Short Questions
Simple Short Questions

Q1 Set $A$ is uncountable iff $A \subseteq R$ ($R$ is the set of real numbers)

Q2 Set $A$ is countable iff $N \subseteq A$ where $N$ is the set of natural numbers

Q3 The set $2^N$ is infinitely countable

Q4 The set $A = \{\{n\} \in 2^N : n^2 + 1 \leq 15\}$ is infinite

Q5 The set $A = \{(\{n\}, n) \in 2^N \times N : 1 \leq n \leq n^2\}$ is infinitely countable

Q6 Union of an infinite set and a finite set is an infinitely countable set
Answers to Simple Short Questions

Q1 Set $A$ is uncountable if and only if $A \subseteq \mathbb{R}$ ($\mathbb{R}$ is the set of real numbers)

NO

The set $2^\mathbb{R}$ is uncountable, as $|\mathbb{R}| < |2^\mathbb{R}|$ by Cantor Theorem, but $2^\mathbb{R}$ is not a subset of $\mathbb{R}$

Also for example. $N \subseteq \mathbb{R}$ and $N$ is not uncountable
Answers to Simple Short Questions

Q2  Set $A$ is countable if and only if $N \subseteq A$, where $N$ is the set of natural numbers.

NO

For example, the set $A = \{\emptyset\}$ is countable as it is finite, but $N \not\subseteq \{\emptyset\}$.

In fact, $A$ can be any finite set or any $A$ can be any infinite set that does not include $N$, for example,

$$A = \{\{n\} : n \in N\}$$
Answers to Simple Short Questions

Q3  The set \(2^N\) is infinitely countable
NO
\(|2^N| = |R| = C\) and hence \(2^N\) is uncountable

Q4  The set \(A = \{\{n\} \in 2^N : n^2 + 1 \leq 15\}\) is infinite
NO
The set \(\{n \in N : n^2 + 1 \leq 15\} = \{0, 1, 2, 3\}\),
Hence the set \(A = \{\{0\}, \{1\}, \{2\}, \{3\}\}\) is finite
Answers to Simple Short Questions

**Q5** The set \( A = \{(\{n\}, n) \in 2^N \times N : 1 \leq n \leq n^2\} \) is **infinitely countable** (countably infinite)

**YES**

Observe that the condition \( n \leq n^2 \) holds for all \( n \in N \), so the set \( B = \{n : n \leq n^2\} \) is **infinitely countable**

The set \( C = \{(\{n\} \in 2^N : 1 \leq n \leq n^2\} \) is also **infinitely countable** as the function given by a formula \( f(n) = \{n\} \) is 1–1 and maps \( B \) onto \( C \), i.e \(|B| = |C|\)

The set \( A = C \times B \) is hence **infinitely countable** as the Cartesian Product of two **infinitely countable** sets
Chapter 1
Sets, Relations, and Languages

Slides Set 2
PART 5: Fundamental Proof Techniques
1. Mathematical Induction
2. The Pigeonhole Principle
3. The Diagonalization Principle
Mathematical Induction Applications
Examples

Counting Functions Theorem

For any finite, non empty sets \( A, B \), there are \(|B|^{|A|}\) functions that map \( A \) into \( B \).

Proof

We conduct the proof by Mathematical Induction over the number of elements of the set \( A \), i.e. over \( n \in \mathbb{N} - \{0\} \), where \( n = |A| \).
Counting Functions Theorem Proof

Base case  \( n = 1 \)
We have hence that \( |A| = 1 \) and let \( |B| = m, \ m \geq 1 \), i.e.

\[
A = \{a\} \quad \text{and} \quad B = \{b_1,...b_m\}, \ m \geq 1
\]

We have to prove that there are \( |B|^{|A|} = m^1 \)
functions that map \( A \) into \( B \)
The base case holds as there are exactly \( m^1 = m \) functions \( f : \{a\} \longrightarrow \{b_1,...b_m\} \) defined as follows

\[
f_1(a) = b_1, \ f_2(a) = b_2, \ ...., \ f_m(a) = b_m
\]
Counting Functions Theorem Proof

Inductive Step
Let \( A = A_1 \cup \{a\} \) for \( a \notin A_1 \) and \( |A_1| = n \)

By inductive assumption, there are \( m^n \) functions

\[ f : A \longrightarrow B = \{b_1, \ldots, b_m\} \]

We group all functions that map \( A_1 \) as follows

Group 1 contains all functions \( f_1 \) such that

\[ f_1 : A \longrightarrow B \]

and they have the following property

\[ f_1(a) = b_1, \quad f_1(x) = f(x) \quad \text{for all} \quad f : A \longrightarrow B \quad \text{and} \quad x \in A_1 \]

By inductive assumption there are \( m^n \) functions in

the Group 1
Inductive Step
We define now a Group $i$, for $1 \leq i \leq m$, $m = |B|$ as follows
Each Group $i$ contains all functions $f_i$ such that

$$f_i : A \rightarrow B$$

and they have the following property

$$f_i(a) = b_1, \quad f_i(x) = f(x) \quad \text{for all } f : A \rightarrow B \text{ and } x \in A_1$$

By inductive assumption there are $m^n$ functions in each of the Group $i$
There are $m = |B|$ groups and each of them has $m^n$ elements, so all together there are

$$m(m^n) = m^{n+1}$$

functions, what ends the proof
Pigeonhole Principle Theorem
If \( A \) and \( B \) are non-empty finite sets and \( |A| > |B| \), then there is no one-to-one function from \( A \) to \( B \).

Proof
We conduct the proof by Mathematical Induction over \( n \in \mathbb{N} - \{0\} \), where \( n = |B| \) and \( B \neq \emptyset \).

Base case \( n = 1 \)
Suppose \( |B| = 1 \), that is, \( B = \{b\} \), and \( |A| > 1 \).
If \( f : A \rightarrow \{b\} \), then there are at least two distinct elements \( a_1, a_2 \in A \), such that \( f(a_1) = f(a_2) = \{b\} \).
Hence the function \( f \) is not one-to-one.
Pigeonhole Principle Proof

**Inductive Assumption**
We assume that any $f : A \rightarrow B$ is **not one-to one** provided

$$|A| > |B| \quad \text{and} \quad |B| \leq n, \quad \text{where} \quad n \geq 1$$

**Inductive Step**
Suppose that $f : A \rightarrow B$ is such that

$$|A| > |B| \quad \text{and} \quad |B| = n + 1$$

Choose some $b \in B$
Since $|B| \geq 2$ we have that $B - \{b\} \neq \emptyset$
Pigeonhole Principle Proof

Consider the set $f^{-1}(\{b\}) \subseteq A$. We have two cases

1. $|f^{-1}(\{b\})| \geq 2$

Then by definition there are $a_1, a_2 \in A$, such that $a_1 \neq a_2$ and $f(a_1) = f(a_2) = b$ what proves that the function $f$ is not one-to one

2. $|f^{-1}(\{b\})| \leq 1$

Then we consider a function
g : A - f^{-1}(\{b\}) \rightarrow B - \{b\}
such that

$$g(x) = f(x) \quad \text{for all} \quad x \in A - f^{-1}(\{b\})$$
Pigeonhole Principle Proof

Observe that the inductive assumption applies to the function $g$ because $|B - \{b\}| = n$ for $|B| = n + 1$ and

$$|A - f^{-1}(\{b\})| \geq |A| - 1 \text{ for } |f^{-1}(\{b\})| \leq 1$$

We know that $|A| > |B|$, so

$$|A| - 1 > |B| - 1 = n = |B - \{b\}| \text{ and } |A - f^{-1}(\{b\})| > |B - \{b\}|$$

By the inductive assumption applied to $g$ we get that $g$ is not one-to-one.

Hence $g(a_1) = g(a_2)$ for some distinct $a_1, a_2 \in A - f^{-1}(\{b\})$, but then $f(a_1) = f(a_2)$ and $f$ is not one-to-one either.
Pigeonhole Principle Revisited

We now formulate a bit stronger version of the pigeonhole principle and present its slightly different proof.

Pigeonhole Principle Theorem
If \( A \) and \( B \) are finite sets and \( |A| > |B| \), then there is no one-to-one function from \( A \) to \( B \).

Proof
We conduct the proof by Mathematical Induction over \( n \in \mathbb{N}, \) where \( n = |B| \).

Base case \( n = 0 \)
Assume \( |B| = 0 \), that is, \( B = \emptyset \). Then there is no function \( f : A \rightarrow B \) whatsoever; let alone a one-to-one function.
Pigeonhole Principle Revisited Proof

Inductive Assumption
Any function \( f : A \rightarrow B \) is **not one-to one** provided

\[ |A| > |B| \quad \text{and} \quad |B| \leq n, \quad n \geq 0 \]

Inductive Step
Suppose that \( f : A \rightarrow B \) is such that

\[ |A| > |B| \quad \text{and} \quad |B| = n + 1 \]

We have to show that \( f \) is **not one-to one** under the Inductive Assumption
Pigeonhole Principle Revisited Proof

We proceed as follows
We choose some element \( a \in A \)
Since \(|A| > |B|\), and \(|B| = n + 1 \geq 1\) such choice is possible

Observe now that if there is another element \( a' \in A \) such that \( a' \neq a \) and \( f(a) = f(a') \), then obviously the function \( f \) is not one-to one and we are done

So, suppose now that the chosen \( a \in A \) is the only element mapped by \( f \) to \( f(a) \)
Pigeonhole Principle Revisited Proof

Consider then the sets $A - \{a\}$ and $B - \{f(a)\}$ and a function

$$g : A - \{a\} \rightarrow B - \{f(a)\}$$

such that

$$g(x) = f(x) \text{ for all } x \in A - \{a\}$$

Observe that the inductive assumption applies to $g$ because

$$|B - \{f(a)\}| = n \quad \text{and} \quad |A - \{a\}| = |A| - 1 > |B| - 1 = |B - \{f(a)\}|$$
Pigeonhole Principle Revisited Proof

Hence by the **inductive assumption** the function $g$ is **not one-to-one**

Therefore, there are two distinct elements elements of $A - \{a\}$ that are mapped by $g$ to the same element of $B - \{f(a)\}$

The function $g$ is, by definition, such that

$$g(x) = f(x) \quad \text{for all } x \in A - \{a\}$$

so the function $f$ is **not one-to-one** either

This **ends** the proof
Pigeonhole Principle Theorem Application

The Pigeonhole Principle Theorem is a quite simple fact but is used in a large variety of proofs including many in this course. We present here just one simple application which we will use in later Chapters.

Path Definition
Let \( A \neq \emptyset \) and \( R \subseteq A \times A \) be a binary relation in the set \( A \). A path in the binary relation \( R \) is a finite sequence \( a_1, \ldots, a_n \) such that \( (a_i, a_{i+1}) \in R \), for \( i = 1, 2, \ldots, n-1 \) and \( n \geq 1 \).

The path \( a_1, \ldots, a_n \) is said to be from \( a_1 \) to \( a_n \). The length of the path \( a_1, \ldots, a_n \) is \( n \).

The path \( a_1, \ldots, a_n \) is a cycle if \( a_i \) are all distinct and also \( (a_n, a_1) \in R \).
Path Theorem
Let \( R \) be a binary relation on a finite set \( A \) and let \( a, b \in A \). If there is a path from \( a \) to \( b \) in \( R \), then there is a path of length at most \( |A| \).

Proof
Suppose that \( a_1, \ldots, a_n \) is the shortest path from \( a = a_1 \) to \( b = a_n \), that is, the path with the smallest length, and suppose that \( n > |A| \). By Pigeonhole Principle there is an element in \( A \) that repeats on the path, say \( a_i = a_j \) for some \( 1 \leq i < j \leq n \).

But then \( a_1, \ldots, a_i, a_{j+1}, \ldots, a_n \) is a shorter path from \( a \) to \( b \), contradicting \( a_1, \ldots, a_n \) being the shortest path.
The Diagonalization Principle

Here is yet another Principle which justifies a new important proof technique

**Diagonalization Principle** (Georg Cantor 1845-1918)

Let $R$ be a binary relation on a set $A$, i.e. $R \subseteq A \times A$ and let $D$, the diagonal set for $R$ be as follows

$$D = \{a \in A : (a, a) \notin R\}$$

For each $a \in A$, let

$$R_a = \{b \in A : (a, b) \in R\}$$

Then $D$ is **distinct** from each $R_a$
The Diagonalization Principle Applications

Here are two theorems whose proofs are the "classic" applications of the Diagonalization Principle

**Cantor Theorem 2**
Let \( \mathbb{N} \) be the set on natural numbers

\[
\text{The set } 2^{\mathbb{N}} \text{ is uncountable}
\]

**Cantor Theorem 3**
The set of real numbers in the interval \([0, 1]\) is uncountable
Cantor Theorem 2 Proof

Cantor Theorem 2
Let \( N \) be the set on natural numbers

The set \( 2^N \) is uncountable

Proof
We apply proof by contradiction method and the Diagonalization Principle
Suppose that \( 2^N \) is countably infinite. That is, we assume that we can put sets of \( 2^N \) in a one-to one sequence \( \{R_n\}_{n \in N} \) such that

\[
2^N = \{R_0, R_1, R_2, \ldots \}
\]

We define a binary relation \( R \subseteq N \times N \) as follows

\[
R = \{(i, j) : j \in R_i\}
\]

This means that for any \( i, j \in N \) we have that

\[
(i, j) \in R \text{ if and only if } j \in R_i
\]
Cantor Theorem 2 Proof

In particular, for any $i, j \in N$ we have that 

$$(i, j) \notin R \text{ if and only if } j \notin R_i$$

and the diagonal set $D$ for $R$ is 

$$D = \{n \in N : n \notin R_n\}$$

By definition $D \subseteq N$, i.e.

$$D \in 2^N = \{R_0, R_1, R_2, \ldots\}$$

and hence 

$$D = R_k \text{ for some } k \geq 0$$
Cantor Theorem 2 Proof

We obtain **contradiction** by asking whether \( k \in R_k \) for

\[
D = R_k
\]

We have two cases to consider: \( k \in R_k \) or \( k \notin R_k \)

**c1** Suppose that \( k \in R_k \)

Since \( D = \{ n \in N : \ n \notin R_n \} \) we have that \( k \notin D \)

But \( D = R_k \) and we get \( k \notin R_k \)

**Contradiction**

**c2** Suppose that \( k \notin R_k \)

Since \( D = \{ n \in N : \ n \notin R_n \} \) we have that \( k \in D \)

But \( D = R_k \) and we get \( k \in R_k \)

**Contradiction**

This ends the proof
Cantor Theorem 3 Proof

Cantor Theorem 3
The set of real numbers in the interval \([0, 1]\) is **uncountable**

Proof
We carry the proof by the **contradiction method**
We assume that the set of real numbers in the interval \([0, 1]\) is **infinitely countable**
This means, by definition, that there is a function \(f\) such that \(f : \mathbb{N} \stackrel{1-1,onto}{\rightarrow} [0, 1]\)
Let \(f\) be any such function. We write \(f(n) = d_n\) and denote by
\[
d_0, d_1, \ldots, d_n, \ldots,\]
a sequence of all elements of \([0, 1]\) **defined** by \(f\)
We will get a **contradiction** by showing that one can always find an element \(d \in [0,1]\) such that \(d \neq d_n\) for all \(n \in \mathbb{N}\)
We use **binary** representation of real numbers.
Hence we assume that all numbers in the interval $[01]$ form a one to one sequence:

$$d_0 = 0.a_{00} a_{01} a_{02} a_{03} a_{04} \ldots \ldots$$

$$d_1 = 0.a_{10} a_{11} a_{12} a_{13} a_{04} \ldots \ldots$$

$$d_2 = 0.a_{20} a_{21} a_{22} a_{23} a_{0} \ldots \ldots$$

$$d_3 = 0.a_{30} a_{31} a_{32} a_{33} a_{04} \ldots \ldots$$

$$\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$$

where all $a_{ij} \in \{0, 1\}$
Cantor Theorem 3 Proof

We use Cantor Diagonatization idea to define an element \( d \in [01] \), such that \( d \neq d_n \) for all \( n \in N \) as follows.

For each element \( a_{nn} \) of the "diagonal"

\[
a_{00}, a_{11}, a_{22}, \ldots \ a_{nn}, \ldots , \ldots
\]

of the sequence \( d_0, d_1, \ldots, d_n, \ldots, \) of binary representation of all elements of the interval \([01]\) we define an element \( b_{nn} \neq a_{nn} \) as

\[
b_{nn} = \begin{cases} 
0 & \text{if } a_{nn} = 1 \\
1 & \text{if } a_{nn} = 0 
\end{cases}
\]
Cantor Theorem 3 Proof

Given such defined sequence

\[ b_{00}, \ b_{11}, \ b_{22}, \ b_{33}, \ b_{44}, \ \ldots \ \ldots \]

We now construct a real number \( d \) as

\[ d = b_{00} \ b_{11} \ b_{22} \ b_{33} \ b_{44} \ \ldots \ \ldots \]

Obviously \( d \in [01] \) and by the Diagonatization Principle

\( d \neq d_n \) for all \( n \in N \)

**Contradiction**

This ends the **proof**
Here is another proof of the Cantor Theorem 3

It uses, after Cantor the decimal representation of real numbers

In this case we assume that all numbers in the interval [01] form a one to one sequence

\[ d_0 = 0.a_{00} a_{01} a_{02} a_{03} a_{04} \ldots \ldots \]

\[ d_1 = 0.a_{10} a_{11} a_{12} a_{13} a_{04} \ldots \ldots \]

\[ d_2 = 0.a_{20} a_{21} a_{22} a_{23} a_{0} \ldots \ldots \]

\[ d_3 = 0.a_{30} a_{31} a_{32} a_{33} a_{04} \ldots \ldots \]

\[ \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \]

where all \( a_{ij} \in \{0, 1, 2 \ldots 9\} \)
Cantor Theorem 3 Proof

For each element $a_{nn}$ of the ”diagonal”

$$a_{00}, a_{11}, a_{22}, \ldots a_{nn}, \ldots , \ldots$$

we define now an element (this is not the only possible definition) $b_{nn} \neq a_{nn}$ as

$$b_{nn} = \begin{cases} 
2 & \text{if } a_{nn} = 1 \\
1 & \text{if } a_{nn} \neq 1 
\end{cases}$$

We construct a real number $d \in [01]$ as

$$d = b_{00} \ b_{11} \ b_{22} \ b_{33} \ b_{44} \ \ldots \ \ldots$$
Chapter 1
Sets, Relations, and Languages

Slides Set 3
PART 6: Closures and Algorithms
Closures - Intuitive

Idea

Natural numbers $N$ are **closed** under $+$, i.e. for given two natural numbers $n, m$ we always have that $n + m \in N$

Natural numbers $N$ are **not closed** under subtraction $-$, i.e. there are two natural numbers $n, m$ such that $n – m \notin N$, for example $1, 2 \in N$ and $1 – 2 \notin N$

Integers $Z$ are **closed** under $–$, moreover $Z$ is the **smallest** set containing $N$ and closed under subtraction $–$

The set $Z$ is called a **closure** of $N$ under subtraction $–$
Consider the two directed graphs $R$ (a) and $R^*$ (b) as shown below.

Observe that $R^* = R \cup \{(a_i, a_i) : i = 1, 2, 3, 4\} \cup \{(a_2, a_4)\}$, $R \subseteq R^*$ and is $R^*$ is reflexive and transitive whereas $R$ is neither, moreover $R^*$ is also the smallest set containing $R$ that is reflexive and transitive.

We call such relation $R^*$ the reflexive, transitive closure of $R$.

We define this concept formally in two ways and prove the equivalence of the two definitions.
Two Definitions of $R^*$

**Definition 1 of $R^*$**

$R^*$ is called a reflexive, transitive closure of $R$ iff $R \subseteq R^*$ and is $R^*$ is reflexive and transitive and is the smallest set with these properties.

This definition is based on a notion of a **closure property** which is any property of the form ” the set $B$ is closed under relations $R_1, R_2, \ldots, R_m$”.

We define it formally and prove that reflexivity and transitivity are closures properties.

Hence we **justify** the name: reflexive, transitive closure of $R$ for $R^*$. 
Two Definitions of $R^*$

**Definition 2 of $R^*$**

Let $R$ be a binary relation on a set $A$

The **reflexive, transitive closure of $R$** is the relation

$$R^* = \{(a, b) \in A \times A : \text{there is a path from } a \text{ to } b \text{ in } R\}$$

This is a much simpler definition- and algorithmically more interesting as it uses a simple notion of a path

We hence **start our investigations** from it- and only later introduce all notions needed for the **Definition 1** in order to prove that the $R^*$ defined above is really what its name says: the **reflexive, transitive closure of $R$**
Definition 2 of $R^*$

We bring back the following

**Path Definition**

A *path* in the binary relation $R$ is a finite sequence $a_1, \ldots, a_n$ such that $(a_i, a_{i+1}) \in R$, for $i = 1, 2, \ldots n - 1$ and $n \geq 1$

The path $a_1, \ldots, a_n$ is said to be from $a_1$ to $a_n$

The path $a_1$ (case when $n = 1$) always exist and is called a *trivial path* from $a_1$ to $a_1$

**Definition 2**

Let $R$ be a binary relation on a set $A$

The *reflexive, transitive closure of $R$* is the relation

$$R^* = \{(a, b) \in A \times A : \text{ there is a path from } a \text{ to } b \text{ in } R \}$$
Algorithms

Definition 2 immediately suggests an following algorithm for computing the reflexive transitive closure $R^*$ of any given binary relation $R$ over some finite set $A = \{a_1, a_2, \ldots, a_n\}$

Algorithm 1
Initially $R^* := 0$
for $i = 1, 2, \ldots, n$ do
for each i- tuple $(b_1, \ldots, b_i) \in A^i$ do
if $b_1, \ldots, b_i$ is a path in $R$ then add $(b_1, b_n)$ to $R^*$
The Book develops and prove correctness of a following much faster algorithm

Algorithm 2

Initially $R^* := R \cup \{(a_i, a_i) : a_i \in A\}$
for $j = 1, 2, \ldots, n$ do
for $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, n$ do
if $(a_i, a_j), (a_j, a_k) \in R^*$ but $(a_i, a_k) \notin R^*$
then add $(a_i, a_k)$ to $R^*$
Closure Property Formal

We introduce now formally a concept of a closure property of a given set

**Definition**

Let \( D \) be a set, let \( n \geq 0 \) and let \( R \subseteq D^{n+1} \) be a \((n + 1)\)-ary relation on \( D \)

Then the subset \( B \) of \( D \) is said to be **closed under** \( R \) if \( b_{n+1} \in B \) whenever \((b_1, \ldots, b_n, b_{n+1}) \in R\)

Any property of the form ”the set \( B \) is closed under relations \( R_1, R_2, \ldots, R_m \)” is called a **closure property** of \( B \)
Closure Property Examples

Observe that any function \( f : D^n \to D \) is a special relation \( f \subseteq D^{n+1} \) so we have also defined what does it mean that a set \( A \subseteq D \) is closed under the function \( f \).

**E1:** \( + \) is a closure property of \( N \).

Adition is a function \( + : N \times N \to N \) defined by a formula \( +(n, m) = n + m \), i.e. it is a relation \( + \subseteq N \times N \times N \) such that

\[
+ = \{(n, m, n + m) : n, m \in N\}
\]

Obviously the set \( N \subseteq N \) is (formally) closed under + because for any \( n, m \in N \) we have that \( (n, m, n + m) \in + \).
Closures Property Examples

E2: \( \cap \) is a closure property of \( 2^N \)
\( \cap \subseteq 2^N \times 2^N \times 2^N \) is defined as

\[(A, B, C) \in \cap \quad \text{iff} \quad A \cap B = C\]

and the following is true for all \( A, B, C \in 2^N \)

if \( A, B \in 2^N \) and \( (A, B, C) \in \cap \) then \( C \in 2^N \)
Closure Property Fact 1

Since relations are sets, we can speak of one relation as being closed under one or more others.

We show now the following

CP Fact 1

Transitivity is a closure property

Proof

Let $D$ be a set, let $Q$ be a ternary relation on $D \times D$, i.e. $Q \subseteq (D \times D)^3$ be such that

$$Q = \{((a, b), (b, c), (a, c)) : a, b, c \in D\}$$

Observe that for any binary relation $R \subseteq D \times D$, $R$ is closed under $Q$ if and only if $R$ is transitive.
CP Fact1 Proof

The definition of closure of $R$ under $Q$ says: for any $x, y, z \in D \times D,$

if $x, y \in R$ and $(x, y, z) \in Q$ then $z \in R$

But $(x, y, z) \in Q$ iff $x = (a, b), y = (b, c), z = (a, c)$ and $(a, b), (b, c) \in R$ implies $(a, c) \in R$

is a true statement for all $a, b, c \in D$ iff $R$ is transitive
Closure Property Fact 2

We show now the following

CP Fact 2

Reflexivity is a closure property

Proof

Let $D \neq \emptyset$, we define an unary relation $Q'$ on $D \times D$, i.e. $Q' \subseteq D \times D$ as follows

$$Q' = \{(a, a) : a \in D\}$$

Observe that for any $R$ binary relation on $D$, i.e. $R \subseteq D \times D$ we have that

$R$ is closed under $Q'$ if and only if $R$ is reflexive
Closure Property Theorem

**CP Theorem**
Let $P$ be a closure property defined by relations on a set $D$, and let $A \subseteq D$
Then there is a **unique minimal** set $B$ such that $B \subseteq A$ and $B$ has property $P$
Two Definition of $R^*$ Revisited

Definition 1

$R^*$ is called a reflexive, transitive closure of $R$ iff $R \subseteq R^*$ and is $R^*$ is reflexive and transitive and is the smallest set with these properties

Definition 2

Let $R$ be a binary relation on a set $A$

The reflexive, transitive closure of $R$ is the relation

$$R^* = \{(a, b) \in A \times A : \text{there is a path from } a \text{ to } b \text{ in } R\}$$

EquivalencyTheorem

$R^*$ of the Definition 2 is the same as $R^*$ of the Definition 1 and hence richly deserves its name reflexive, transitive closure of $R$
Equivalency of Two Definition of $R^*$

Proof \quad Let

\[ R^* = \{(a, b) \in A \times A : \text{there is a path from } a \text{ to } b \text{ in } R\} \]

$R^*$ is reflexive \quad for there is a trivial path (case $n=1$) from $a$ to $a$, for any $a \in A$

$R^*$ is transitive \quad as for any $a, b, c \in A$

if there is a path from $a$ to $b$ and a path from $b$ to $c$, then there is a path from $a$ to $c$

Clearly $R \subseteq R^*$ \quad because there is a path from $a$ to $b$ whenever $(a, b) \in R$
Equivalency of Two Definition of $R^*$

Consider a set $S$ of all binary relations on $A$ that contain $R$ and are reflexive and transitive, i.e.

$$S = \{ Q \subseteq A \times A : R \subseteq Q \text{ and } Q \text{ is reflexive and transitive} \}$$

We have just proved that $R^* \in S$

We prove now that $R^*$ is the smallest set in the poset $(S, \subseteq)$, i.e. that for any $Q \in S$ we have that $R^* \subseteq Q$
Equivalency of Two Definition of $R^*$

Assume that $(a, b) \in R^*$. By Definition 2 there is a path $a = a_1, \ldots, a_k = b$ from $a$ to $b$ and let $Q \in S$

We prove by Mathematical Induction over the length $k$ of the path from $a$ to $b$

**Base case:** $k=1$

We have that the path is $a = a_1 = b$, i.e. $(a, a) \in R^*$ and $(a, a) \in Q$ from reflexivity of $Q$

**Inductive Assumption:**

Assume that for any $(a, b) \in R^*$ such that there is a path of length $k$ from $a$ to $b$ we have that $(a, b) \in Q$
Equivalency of Two Definition of $R^*$

Inductive Step:
Let $(a, b) \in R^*$ be now such that there is a path of length $k+1$ from $a$ to $b$, i.e. there is a path $a = a_1, \ldots, a_k, a_{k+1} = b$.

By inductive assumption $(a = a_1, a_k) \in Q$ and by definition of the path $(a_k, a_{k+1} = b) \in R$.

But $R \subseteq Q$ hence $(a_k, a_{k+1} = b) \in Q$ and $(a, b) \in Q$ by transitivity.

This ends the proof that Definition 2 of $R^*$ implies the Definition 1.

The inverse implication follows from the previously proven fact that reflexivity and transitivity are closure properties.
Chapter 1
Sets, Relations, and Languages

Slides Set 4
PART 7: Alphabets and languages
PART 8: Finite Representation of Languages
Chapter 1
Sets, Relations, and Languages

Slides Set 4
PART 7: Alphabets and languages
Alphabets and languages

Introduction

Data are **encoded** in the computers’ memory as strings of bits or other **symbols** appropriate for **manipulation**.

The mathematical study of the **Theory of Computation** begins with understanding of mathematics of **manipulation** of strings of **symbols**.

We first introduce two basic notions: **Alphabet** and **Language**.
Alphabet

Definition

Any finite set is called an alphabet.

Elements of the alphabet are called symbols of the alphabet.

This is why we also say:

Alphabet is any finite set of symbols.
Alphabet

Alphabet Notation
We use a symbol \( \Sigma \) to denote the alphabet

Remember
\( \Sigma \) can be \( \emptyset \) as empty set is a finite set

When we want to study non-empty alphabets we have to say so, i.e to write:
\[
\Sigma \neq \emptyset
\]
Alphabet Examples

\[ \Sigma = \{ \top, \emptyset, \partial, \oint, \bigotimes, \mathbf{a}, \nabla \} \]

\[ \Sigma = \{ a, b, c \} \]

\[ \Sigma = \{ n \in \mathbb{N} : n \leq 10^5 \} \]

\[ \Sigma = \{ 0, 1 \} \] is called a binary alphabet
Alphabet Examples

For simplicity and consistence we will use only as symbols of the alphabet letters (with indices if necessary) or other common characters when needed and specified.

We also write $\sigma \in \Sigma$ for a general form of an element in $\Sigma$.

$\Sigma$ is a finite set and we will write

$$\Sigma = \{a_1, a_2, \ldots, a_n\} \text{ for } n \geq 0$$
Finite Sequences Revisited

Definition
A finite sequence of elements of a set $A$ is any function $f : \{1, 2, \ldots, n\} \rightarrow A$ for $n \in \mathbb{N}$

We call $f(n) = a_n$ the $n$-th element of the sequence $f$
We call $n$ the length of the sequence

$$a_1, a_2, \ldots, a_n$$

Case $n=0$
In this case the function $f$ is empty and we call it an empty sequence and denote by $e$
Words over $\Sigma$

Let $\Sigma$ be an alphabet

We call finite sequences of the alphabet $\Sigma$ words or strings over $\Sigma$

We denote by $e$ the empty word over $\Sigma$

Some books use symbol $\lambda$ for the empty word
Words over $\Sigma$

**E5** Let $\Sigma = \{a, b\}$

We will write some words (strings) over $\Sigma$ in a **shorthand** notation as for example

$$aaa, \ ab, \ bbb$$

instead using the formal definition:

$$f : \{1, 2, 3\} \rightarrow \Sigma$$

such that $f(1) = a, f(2) = a, f(3) = a$ for the word $aaa$

or$$g : \{1, 2\} \rightarrow \Sigma \text{ such that } g(1) = b, g(2) = b$$

for the word $bb \ldots$ etc..
Words in $\Sigma^*$

Let $\Sigma$ be an alphabet. We denote by $\Sigma^*$ the set of all finite sequences over $\Sigma$. Elements of $\Sigma^*$ are called words over $\Sigma$. We write $w \in \Sigma^*$ to express that $w$ is a word over $\Sigma$.

Symbols for words are

$$w, z, v, x, y, z, \alpha, \beta, \gamma \in \Sigma^*$$

$$x_1, x_2, \ldots \in \Sigma^* \quad y_1, y_2, \ldots \in \Sigma^*$$
Words in $\Sigma^*$

**Observe** that the set of all finite sequences include the **empty** sequence i.e. $e \in \Sigma^*$ and we hence have the following

**Fact**
For any **alphabet** $\Sigma$, 

$$\Sigma^* \neq \emptyset$$
Chapter 1

Some Short Questions and Answers
Q1  Let $\Sigma = \{a, b\}$

How many are there all possible words of length 5 over $\Sigma$?

A1  By definition, words over $\Sigma$ are finite sequences; Hence words of a length 5 are functions

$$f : \{1, 2, \ldots, 5\} \rightarrow \{a, b\}$$

So we have by the **Counting Functions Theorem** that there are $2^5$ words of a length 5 over $\Sigma = \{a, b\}$
Counting Functions Theorem

For any finite, non empty sets $A, B$, there are $|B|^{|A|}$ functions that map $A$ into $B$.

The proof is in Part 5.
Short Questions

Q2
Let $\Sigma = \{a_1, \ldots, a_k\}$ where $k \geq 1$
How many are there possible words of length $\leq n$ for $n \geq 0$
in $\Sigma^*$?

A2
By the **Counting Functions Theorem** there are

$$k^0 + k^1 + \cdots + k^n$$

words of length $\leq n$ over $\Sigma$ because for each $m$
there are $k^m$ words of length $m$ over $\Sigma = \{a_1, \ldots, a_k\}$
and $m = 0, 1 \ldots n$
Short Questions

Q3  Given an alphabet $\Sigma \neq \emptyset$

How many are there words in the set $\Sigma^*$?

A3

There are infinitely countably many words in $\Sigma^*$ by the Theorem 5 (Lecture 2) that says: ” for any non empty, finite set $A$, $|A^*| = \aleph_0”$

We hence proved the following

Theorem

For any alphabet $\Sigma \neq \emptyset$, the set $\Sigma^*$ of all words over $\Sigma$ is countably infinite
Languages over $\Sigma$

Language Definition
Given an alphabet $\Sigma$, any set $L$ such that

$$L \subseteq \Sigma^*$$

is called a language over $\Sigma$

Fact 1
For any alphabet $\Sigma$, any language over $\Sigma$ is countable.
Fact 2

For any alphabet \( \Sigma \neq \emptyset \), there are uncountably many languages over \( \Sigma \).

More precisely, there are exactly \( C = |R| \) of languages over any non-empty alphabet \( \Sigma \).
Languages over $\Sigma$

Fact 1
For any alphabet $\Sigma$, any language over $\Sigma$ is countable

Proof
By definition, a set is countable if and only if is finite or countably infinite

1. Let $\Sigma = \emptyset$, hence $\Sigma^* = \{e\}$ and we have two languages $\emptyset, \{e\}$ over $\Sigma$, both finite, so countable

2. Let $\Sigma \neq \emptyset$, then $\Sigma^*$ is countably infinite, so obviously any $L \subseteq \Sigma^*$ is finite or countably infinite, hence countable
Languages over \( \Sigma \)

**Fact 2**
For any alphabet \( \Sigma \neq \emptyset \), there are exactly \( C = |R| \) of languages over any non-empty alphabet \( \Sigma \)

**Proof**
We proved that \( |\Sigma^*| = \aleph_0 \)
By definition \( L \subseteq \Sigma^* \), so there is as many languages over \( \Sigma \) as all subsets of a set of cardinality \( \aleph_0 \) that is as many as \( 2^{\aleph_0} = C \)
Languages over $\Sigma$

Q4 Let $\Sigma = \{a\}$

There is $\aleph_0$ languages over $\Sigma$

NO

We just proved that there is uncountably many, more precisely, exactly $C$ languages over $\Sigma \neq \emptyset$ and we know that

$$\aleph_0 < C$$
Languages over $\Sigma$

Definition
Given an alphabet $\Sigma$ and a word $w \in \Sigma^*$
We say that $w$ has a length $n = |w|$ when

$$w : \{1, 2, \ldots, n\} \rightarrow \Sigma$$

We re-write $w$ as

$$w : \{1, 2, |w|\} \rightarrow \Sigma$$

Definition
Given $\sigma \in \Sigma$ and $w \in \Sigma^*$, we say $\sigma \in \Sigma$ occurs in the $j$-th position in $w \in \Sigma^*$ if and only if $w(j) = \sigma$ for $1 \leq j \leq |w|$
Some Examples

E6  Consider a word $w$ written in a shorthand as

$$w = \text{anita}$$

By formal definition we have

\[
\begin{align*}
    w(1) &= a, & w(2) &= n, & w(3) &= i, & w(4) &= t, & w(5) &= a \\
\end{align*}
\]

and $a$ occurs in the 1st and 5th position

E7  Let $\Sigma = \{0, 1\}$ and $w = 01101101$ (shorthand)

Formally $w : \{1, 2, 8\} \rightarrow \{0, 1\}$ is such that

\[
\begin{align*}
    w(1) &= 0, & w(2) &= 1, & w(3) &= 1, & w(4) &= 0, & w(5) &= 1, \\
    w(6) &= 1, & w(7) &= 0, & w(8) &= 1 \\
\end{align*}
\]

1 occurs in the positions 2, 3, 5, 6 and 8

0 occurs in the positions 1, 4, 7
Informal Concatenation

Informal Definition

Given an alphabet $\Sigma$ and any words $x, y \in \Sigma^*$

We define informally a **concatenation** $\circ$ of words $x, y$ as a word $w$ obtained from $x, y$ by writing the word $x$ followed by the word $y$

We write the **concatenation** of words $x, y$ as

$$w = x \circ y$$

We use the symbol $\circ$ of **concatenation** when it is needed formally, otherwise we will write simply

$$w = xy$$
Formal Concatenation

Definition
Given an alphabet $\Sigma$ and any words $x, y \in \Sigma^*$
We define:

$$w = x \circ y$$

if and only if

1. $|w| = |x| + |y|$
2. $w(j) = x(j)$ for $j = 1, 2, \ldots, |x|$
3. $w(|x| + j) = j(j)$ for $j = 1, 2, \ldots, |y|$
Formal Concatenation

Properties

Directly from definition we have that

\[ w \circ e = e \circ w = w \]

\[(x \circ y) \circ z = x \circ (y \circ z) = x \circ y \circ z \]

Remark: we need to define a concatenation of two words and then we define

\[ x_1 \circ x_2 \circ \cdots \circ x_n = (x_1 \circ x_2 \circ \cdots \circ x_{n-1}) \circ x_n \]

and prove by Mathematical Induction that

\[ w = x_1 \circ x_2 \circ \cdots \circ x_n \text{ is well defined for all } n \geq 2 \]
Substring

Definition
A word \( v \in \Sigma^* \) is a **substring** (sub-word) of \( w \) iff there are \( x, y \in \Sigma^* \) such that

\[
w = xvy
\]

**Remark:** the words \( x, y \in \Sigma^* \), i.e. they can also be empty

**P1**  \( w \) is a substring of \( w \)

**P2**  \( e \) is a substring of any string (any word \( w \))
as we have that \( ew = we = w \)

**Definition**  Let \( w = xy \)

\( x \) is called a **prefix** and \( y \) is called a **suffix** of \( w \)
Power $w^i$

Definition

We define a **power** $w^i$ of $w$ by Mathematical Induction as follows

$$w^0 = e$$

$$w^{i+1} = w^i \circ w$$

E8

$$w^0 = e, \quad w^1 = w^0 \circ w = e \circ w = w, \quad w^2 = w^1 \circ w = w \circ w$$

E9

$$anita^2 = anita^1 \circ anita = e \circ anita \circ anita = anita \circ anita$$
Reversal $w^R$

Definition

Reversal $w^R$ of $w$ is defined by induction over length $|w|$ of $w$ as follows

1. If $|w| = 0$, then $w^R = w = e$
2. If $|w| = n + 1 > 0$, then $w = ua$ for some $a \in \Sigma$, and $u \in \Sigma^*$ and we define

   \[w^R = au^R\] for $|u| < n + 1$

Short Definition of $w^R$

1. $e^R = e$
2. $(ua)^R = au^R$
Reversal Proof

We prove now as an example of Inductive proof the following simple fact

**Fact**

For any \( w, x \in \Sigma^* \)

\[(wx)^R = x^R w^R\]

**Proof** by Mathematical Induction over the length \(|x|\) of \(x\) with \(|w| = constant\)

**Base case** \( n=0 \)

\(|x| = 0\), i.e. \(x=e\) and by definition

\[(we)^R = ew^R = e^R w^R\]
Reversal Proof

Inductive Assumption

\[(wx)^R = x^R w^R \quad \text{for all } |x| \leq n\]

Let now \(|x| = n + 1\), so \(x = ua\) for certain \(a \in \Sigma\) and \(|u| = n\)

We evaluate

\[(wx)^R = (w(ua))^R = ((wu)a)^R\]

\[=_{\text{def}} a(wu)^R =_{\text{ind}} au^R w^R =_{\text{def}} (ua)^R = x^R w^R\]
Languages over $\Sigma$

**Definition**
Given an alphabet $\Sigma$, any set $L$ such that $L \subseteq \Sigma^*$ is called a **language** over $\Sigma$

**Observe** that $\emptyset$, $\Sigma$, $\Sigma^*$ are all languages over $\Sigma$

We have proved

**Theorem**
Any language $L$ over $\Sigma$, is **finite** or **infinitely countable**
Languages over $\Sigma$

Languages are **sets** so we can define them in ways we did for sets, by **listing** elements (for small finite sets) or by giving a **property** $P(w)$ defining $L$, i.e. by setting

$$L = \{w \in \Sigma^* : P(w)\}$$

**E1**

$$L_1 = \{w \in \{0, 1\}^* : \text{w has an even number of 0's} \}$$

**E2**

$$L_2 = \{w \in \{a, b\}^* : \text{w has ab as a sub-string} \}$$
Languages Examples

E3

\[ L_3 = \{ w \in \{0, 1\}^* : |w| \leq 2 \} \]

E4

\[ L_4 = \{ e, 0, 1, 00, 01, 11, 10 \} \]

Observe that \[ L_3 = L_4 \]
Languages Examples

Languages are sets so we can define set operations of union, intersection, generalized union, generalized intersection, complement, Cartesian product, ... etc ... of languages as we did for any sets.

For example, given $L, L_1, L_2 \subseteq \Sigma^*$, we consider

$L_1 \cup L_2, \quad L_1 \cap L_2, \quad L_1 - L_2,$

$-L = \Sigma^* - L, \quad L_1 \times L_2, \ldots \quad$ etc

and we have that all properties of algebra of sets hold for any languages over a given alphabet $\Sigma$. 
We define now a special operation on languages, different from any of the set operation

**Concatenation Definition**

Given \( L_1, L_2 \subseteq \Sigma^* \), a language

\[
L_1 \circ L_2 = \{ w \in \Sigma^* : w = xy \text{ for some } x \in L_1, y \in L_2 \}
\]

is called a *concatenation* of the languages \( L_1 \) and \( L_2 \)
The concatenation $L_1 \circ L_2$ domain issue

We can have that the languages $L_1$, $L_2$ are defined over different domains, i.e. they have two alphabets $\Sigma_1 \neq \Sigma_2$ for

$$L_1 \subseteq \Sigma_1^* \quad \text{and} \quad L_2 \subseteq \Sigma_2^*$$

In this case we always take

$$\Sigma = \Sigma_1 \cup \Sigma_2 \quad \text{and get} \quad L_1, L_2 \subseteq \Sigma^*$$
E5
Let $L_1$, $L_2$ be languages defined below

\[ L_1 = \{ w \in \{a, b\}^* : |w| \leq 1 \} \]

\[ L_2 = \{ w \in \{0, 1\}^* : |w| \leq 2 \} \]

Describe the concatenation $L_1 \circ L_2$ of $L_1$ and $L_2$

Domain $\Sigma$ of $L_1 \circ L_2$
We have that $\Sigma_1 = \{a, b\}$ and $\Sigma_2 = \{0, 1\}$
so we take $\Sigma = \Sigma_1 \cup \Sigma_2 = \{a, b, 0, 1\}$ and

\[ L_1 \circ L_2 \subseteq \Sigma \]
Concatenation Examples

Let $L_1, L_2$ be languages defined below

$L_1 = \{w \in \{a, b\}^*: |w| \leq 1\}$

$L_2 = \{w \in \{0, 1\}^*: |w| \leq 2\}$

We write now a **general formula** for $L_1 \circ L_2$ as follows

$L_1 \circ L_2 = \{w \in \Sigma^*: w = xy \}$

where

$x \in \{a, b\}^*, \ y \in \{0, 1\}^* \text{ and } |x| \leq 1, \ |y| \leq 2$
Concatenation Examples

E5 revisited
Describe the concatenation of $L_1 = \{ w \in \{a, b\}^* : |w| \leq 1 \}$ and $L_2 = \{ w \in \{0, 1\}^* : |w| \leq 2 \}$
As both languages are finite, we list their elements and get $L_1 = \{ e, a, b \}$, $L_2 = \{ e, 0, 1, 01, 00, 11, 10 \}$
We describe their concatenation as

$$L_1 \circ L_2 = \{ ey : y \in L_2 \} \cup \{ ay : y \in L_2 \} \cup \{ by : y \in L_2 \}$$

Here is another general formula for $L_1 \circ L_2$

$$L_1 \circ L_2 = e \circ L_2 \cup (\{a\} \circ L_2) \cup (\{b\} \circ L_2)$$
Concatenation Examples

E6
Describe concatenations \( L_1 \circ L_2 \) and \( L_2 \circ L_1 \) of

\[
L_1 = \{ w \in \{0, 1\}^* : \text{ w has an even number of 0's} \}
\]

and

\[
L_2 = \{ w \in \{0, 1\}^* : \text{ w = 0xx, x \in \Sigma^*} \}
\]

Here the are

\[
L_1 \circ L_2 = \{ w \in \Sigma^* : \text{ w has an odd number of 0's} \}
\]

\[
L_2 \circ L_1 = \{ w \in \Sigma^* : \text{ w starts with 0} \}
\]
Concatenation Examples

We have that

$L_1 \circ L_2 = \{w \in \Sigma^* : w \text{ has an odd number of } 0\text{'s}\}$

$L_2 \circ L_1 = \{w \in \Sigma^* : w \text{ starts with } 0\}$

Observe that

$1000 \in L_1 \circ L_2$ \quad and \quad $1000 \notin L_2 \circ L_1$

This proves that

$L_1 \circ L_2 \neq L_2 \circ L_1$

We hence \underline{proved} the following

\textbf{Fact}

\textbf{Concatenation} of languages is \textbf{not} commutative
Concatenation Examples

**E8**

Let $L_1$, $L_2$ be languages defined below for $\Sigma = \{0, 1\}$

$L_1 = \{w \in \Sigma^* : w = x1, \ x \in \Sigma^*\}$

$L_2 = \{w \in \Sigma^* : w = 0x, \ x \in \Sigma^*\}$

**Describe** the language $L_2 \circ L_1$

Here it is

$$L_2 \circ L_1 = \{w \in \Sigma^* : w = 0xy1, \ x, y \in \Sigma^*\}$$

**Observe** that $L_2 \circ L_1$ can be also defined by a property as follows

$$L_2 \circ L_1 = \{w \in \Sigma^* : w \text{ starts with 0 and ends with 1}\}$$
Distributivity of Concatenation

Theorem
Concatenation is **distributive** over union of languages

More precisely, given languages $L, L_1, L_2, \ldots, L_n$, the following holds for any $n \geq 2$

\[
(L_1 \cup L_2 \cup \cdots \cup L_n) \circ L = (L_1 \circ L) \cup \cdots \cup (L_n \circ L)
\]

\[
L \circ (L_1 \cup L_2 \cup \cdots \cup L_n) = (L \circ L_1) \cup \cdots \cup (L \circ L_n)
\]

**Proof** by Mathematical Induction over $n \in \mathbb{N}, n \geq 2$
Distributivity of Concatenation Proof

We prove the **base case** for the first equation and leave the Inductive argument and a similar proof of the second equation as an exercise

**Base Case**  $n = 2$

We have to prove that

$$(L_1 \cup L_2) \circ L = (L_1 \circ L) \cup (L_2 \circ L)$$

$w \in (L_1 \cup L_2) \circ L$  iff (by definition of $\circ$)

$(w \in L_1 \text{ or } w \in L_2) \text{ and } w \in L$  iff (by distributivity of $\text{and}$ over $\text{or}$)

$(w \in L_1 \text{ and } w \in L) \text{ or } (w \in L_2 \text{ and } w \in L)$  iff (by definition of $\circ$)

$(w \in L_1 \circ L) \text{ or } (w \in L_2 \circ L)$  iff (by definition of $\cup$)

$w \in (L_1 \circ L) \cup (L_2 \circ L)$
Kleene Star - $L^*$

Kleene Star $L^*$ of a language $L$ is yet another operation specific to languages.

It is named after Stephen Cole Kleene (1909 -1994), an American mathematician and world famous logician who also helped lay the foundations for theoretical computer science.

We define $L^*$ as the set of all strings obtained by concatenating zero or more strings from $L$.

Remember that concatenation of zero strings is $e$, and concatenation of one string is the string itself.
Kleene Star - $L^*$

We define $L^*$ formally as

$$L^* = \{w_1 w_2 \ldots w_k : \text{for some } k \geq 0 \text{ and } w_1, \ldots, w_k \in L\}$$

We also write as

$$L^* = \{w_1 w_2 \ldots w_k : k \geq 0, \ w_i \in L, \ i = 1, 2, \ldots, k\}$$

or in a form of Generalized Union

$$L^* = \bigcup_{k \geq 0} \{w_1 w_2 \ldots w_k : w_1, \ldots, w_k \in L\}$$

**Remark** we write $xyz$ for $x \circ y \circ z$. We use the concatenation symbol $\circ$ when we want to stress that we talk about some properties of the concatenation
Kleene Star Properties

Here are some Kleene Star basic properties

P1 \( e \in L^* \), for all \( L \)
Follows directly from the definition as we have case \( k = 0 \)

P2 \( L^* \neq \emptyset \), for all \( L \)
Follows directly from P1, as \( e \in L^* \)

P3 \( \emptyset^* \neq \emptyset \)
Because \( L^* = \emptyset^* = \{e\} \neq \emptyset \)
Kleene Star Properties

Some more Kleene Star basic properties

P4 \[ L^* = \Sigma^* \] for some \( L \)
Take \( L = \Sigma \)

P6 \[ L^* \neq \Sigma^* \] for some \( L \)
Take \( L = \{00, 11\} \) over \( \Sigma = \{0, 1\} \)
We have that \( 01 \notin L^* \) and \( 01 \in \Sigma^* \)
Example

Observation

The property $\mathbf{P4}$ provides a quite trivial example of a language $L$ over an alphabet $\Sigma$ such that $L^* = \Sigma^*$, namely just $L = \Sigma$.

A natural question arises: is there any language $L \neq \Sigma$ such that nevertheless $L^* = \Sigma^*$?
Example

Example

Take $\Sigma = \{0, 1\}$ and take

$$L = \{w \in \Sigma^* : w \text{ has an unequal number of } 0 \text{ and } 1\}$$

Some words in and out of $L$ are

$$100 \in L, \quad 00111 \in L \quad 100011 \notin L$$

We now prove that

$$L^* = \{0, 1\}^* = \Sigma^*$$
Example Proof

Given
$L = \{ w \in \{0, 1\}^* : w \text{ has an unequal number of 0 and 1} \}$
We now prove that

\[ L^* = \{0, 1\}^* = \Sigma^* \]

Proof
By definition we have that $L \subseteq \{0, 1\}^*$ and $\{0, 1\}^{**} = \{0, 1\}^*$
By the following property of languages:

P: If $L_1 \subseteq L_2$, then $L_1^* \subseteq L_2^*$

and get that

\[ L^* \subseteq \{0, 1\}^{**} = \{0, 1\}^* \text{ i.e. } L^* \subseteq \Sigma^* \]
Now we have to show that $\Sigma^* \subseteq L^*$, i.e.

$$\{0, 1\}^* \subseteq \{w \in 0, 1^* : w \text{ has an unequal number of 0 and 1}\}$$

**Observe** that

$0 \in L$ because $0$ regarded as a string over $\Sigma$ has an unequal number appearances of $0$ and $1$

The number of appearances of $1$ is zero and the number of appearances of $0$ is one

$1 \in L$ for the same reason a $0 \in L$

So we proved that $\{0, 1\} \subseteq L$

We now use the property $P$ and get

$$\{0, 1\}^* \subseteq L^* \text{ i.e } \Sigma^* \subseteq L^*$$

what **ends the proof** that $\Sigma^* = L^*$
\( L^* \) and \( L^+ \)

We define

\[
L^+ = \{ w_1 w_2 \ldots w_k : \text{for some } k \geq 1 \text{ and some } w_1, \ldots, w_k \in L \}
\]

We write it also as follows

\[
L^+ = \{ w_1 w_2 \ldots w_k : k \geq 1 \ w_i \in L, \ i = 1, 2, \ldots, k \}
\]

Properties

\( P1 : \quad L^+ = L \circ L^* \quad P2 : \quad e \in L^+ \text{ iff } e \in L \)
We know that
\[ e \in L^* \quad \text{for all} \quad L \]

**Show** that
For some language \( L \) we have that \( e \in L^+ \) and
for some language \( L \) we can have that \( e \notin L^+ \)

**E1**
Obviously, for any \( L \) such that \( e \in L \) we have that \( e \in L^+ \)

**E2**
If \( L \) is such that \( e \notin L \) we have that \( e \notin L^+ \) as \( L^+ \) does not have a case \( k=0 \)
Chapter 1
Sets, Relations, and Languages

Slides Set 4
PART 8: Finite Representation of Languages
Finite Representation of Languages

Introduction

We can represent a finite language by finite means for example listing all its elements.

Languages are often infinite and so a natural question arises if a finite representation is possible and when it is possible when a language is infinite.

The representation of languages by finite specifications is a central issue of the theory of computation.

Of course we have to define first formally what do we mean by representation by finite specifications, or more precisely by a finite representation.
Idea of Finite Representation

We start with an example: let

\[ L = \{a\}^* \cup (\{b\} \circ \{a\}^*) = \{a\}^* \cup (\{b\} \{a\}^*) \]

Observe that by definition of Kleene’s star

\[ \{a\}^* = \{e, a, aa, aaa \ldots \} \]

and \( L \) is an infinite set

\[ L = \{e, a, aa, aaa \ldots \} \cup \{b\} \{e, a, aa, aaa \ldots \} \]

\[ L = \{e, a, aa, aaa \ldots \} \cup \{b, ba, baa, baaa \ldots \} \]

\[ L = \{e, a, b, aa, ba, aaa baa, \ldots \} \]
Idea of Finite Representation

The expression \( \{a\}^* \cup (\{b\}\{a\}^*) \) is built out of a finite number of symbols:

\(\{,\},\,(,),\,\ast,\,\cup\)

and describe an infinite set

\[ L = \{e, a, b, aa, ba, aaa baa, \ldots \} \]

We write it in a simplified form - we skip the set symbols \(\{,\}\) as we know that languages are sets and we have

\[ a^* \cup (ba^*) \]
Idea of Finite Representation

We will call such expressions as

\[ a^* \cup (ba^*) \]

a **finite representation** of a language \( L \)

The idea of the **finite representation** is to use symbols

\((, ), \ast, \cup, \emptyset, \) and symbols \( \sigma \in \Sigma \)

to write **expressions** that **describe** the language \( L \)
Example of a Finite Representation

Let \( L \) be a language defined as follows

\[
L = \{ w \in \{0, 1\}^* : \ w \text{ has two or three occurrences of 1} \text{ the first and the second of which are not consecutive } \}
\]

The language \( L \) can be expressed as

\[
L = \{0\}^*\{1\}\{0\}^*\{0\} \circ \{1\}\{0\}^*(\{1\}\{0\}^* \cup \emptyset^*)
\]

We will define and write the finite representation of \( L \) as

\[
L = 0^*10^*010^*(10^* \cup \emptyset^*)
\]

We call expression above (and others alike) a regular expression
Problem with Finite Representation

Question
Can we \textit{finitely represent} \textit{all languages} over an alphabet $\Sigma \neq \emptyset$?

Observation
\textbf{O1.} Different \textit{languages} must have different \textit{representations}.

\textbf{O2.} \textit{Finite representations} are \textit{finite strings} over a \textit{finite set}, so we have that

there are $\aleph_0$ possible \textit{finite representations}.
Problem with Finite Representation

O3. There are uncountably many, precisely exactly \( C = |\mathcal{R}| \) of possible languages over any alphabet \( \Sigma \neq \emptyset \)

Proof

For any \( \Sigma \neq \emptyset \) we have proved that

\[
|\Sigma^*| = \aleph_0
\]

By definition of language

\[
L \subseteq \Sigma^*
\]

so there are as many languages as subsets of \( \Sigma^* \) that is as many as

\[
|2^{\Sigma^*}| = 2^{\aleph_0} = C
\]
Problem with Finite Representation

Question
Can we finitely represent all languages over an alphabet $\Sigma \neq \emptyset$?

Answer
No, we can’t
By O2 and O3 there are countably many (exactly $\aleph_0$) possible finite representations and there are uncountably many (exactly $C$) possible languages over any $\Sigma \neq \emptyset$

This proves that
NOT ALL LANGUAGES CAN BE FINITELY REPRESENTED
Problem with Finite Representation

Moreover
There are **uncountably** many and exactly as many as Real numbers, i.e. $C$ languages that **can not** be **finitely** represented

We can **finitely represent** only a small, **countable** portion of languages

We **define** and **study** here only **two** classes of languages:

**REGULAR** and **CONTEXT FREE** languages
Regular Expressions Definition

Definition
We define a $\mathcal{R}$ of regular expressions over an alphabet $\Sigma$ as follows.

$\mathcal{R} \subseteq (\Sigma \cup \{ (, ), \emptyset, \cup, * \})^*$ and $\mathcal{R}$ is the smallest set such that

1. $\emptyset \in \mathcal{R}$ and $\Sigma \subseteq \mathcal{R}$, i.e. we have that

$$\emptyset \in \mathcal{R} \quad \text{and} \quad \forall \sigma \in \Sigma \ (\sigma \in \mathcal{R})$$

2. If $\alpha, \beta \in \mathcal{R}$, then

$$(\alpha \beta) \in \mathcal{R} \quad \text{concatenation}$$

$$(\alpha \cup \beta) \in \mathcal{R} \quad \text{union}$$

$$\alpha^* \in \mathcal{R} \quad \text{Kleene’s Star}$$
Regular Expressions Theorem

Theorem
The set $\mathcal{R}$ of regular expressions over an alphabet $\Sigma$ is countably infinite

Proof
Observe that the set $\Sigma \cup \{ (, ), \emptyset, \cup, * \}$ is non-empty and finite, so the set $(\Sigma \cup \{ (, ), \emptyset, \cup, * \})^*$ is countably infinite, and by definition

$$\mathcal{R} \subseteq (\Sigma \cup \{ (, ), \emptyset, \cup, * \})^*$$

hence we $|\mathcal{R}| \leq \aleph_0$

The set $\mathcal{R}$ obviously includes an infinitely countable set

$$\emptyset, \emptyset\emptyset, \emptyset\emptyset\emptyset, \ldots, \ldots,$$

what proves that $|\mathcal{R}| = \aleph_0$
Regular Expressions

Example
Given $\Sigma = \{0, 1\}$, we have that

1. $\emptyset \in R$, $1 \in R$, $0 \in R$

2. $(01) \in R$, $1^* \in R$, $0^* \in R$, $0^* \in R$, $(\emptyset \cup 1) \in R$, ..., 

\[ (((0^* \cup 1^*) \cup \emptyset)1)^* \in R \]

Shorthand Notation when writing regular expressions we will keep only essential parenthesis

For example, we will write

\[ ((0^* \cup 1^* \cup \emptyset)1)^* \] instead of \[ (((0^* \cup 1^*) \cup \emptyset)1)^* \]

\[ 1^*01^* \cup (01)^* \] instead of \[ (((1^*01^*) \cup (01)^*)) \]
Regular Expressions and Regular Languages

We use the regular expressions from the set \( \mathcal{R} \) as a representation of languages.

Languages represented by regular expressions are called regular languages.
Regular Expressions and Regular Languages

The idea of the representation is explained in the following example.

Example

The regular expression (written in a shorthand notion)

\[ 1^*01^* \cup (01)^* \]

represents a language

\[ L = \{1\}^*\{0\}\{1\}^* \cup \{01\}^* \]
Definition of Representation

Definition
The **representation relation** between regular expressions and languages they **represent** is established by a **function** $L$ such that if $\alpha \in R$ is any regular expression, then $L(\alpha)$ is the **language represented** by $\alpha$. 
Definition of Representation

Formal Definition
The function \( L : \mathcal{R} \rightarrow 2^{\Sigma^*} \) is defined recursively as follows

1. \( L(\emptyset) = \emptyset, \quad L(\sigma) = \{\sigma\} \) for all \( \sigma \in \Sigma \)
2. If \( \alpha, \beta \in \mathcal{R} \), then

\[
L(\alpha \beta) = L(\alpha) \circ L(\beta) \quad \text{concatenation}
\]

\[
L(\alpha \cup \beta) = L(\alpha) \cup L(\beta) \quad \text{union}
\]

\[
L(\alpha^*) = L(\alpha)^* \quad \text{Kleene’s Star}
\]
Regular Language Definition

Definition
A language \( L \subseteq \Sigma^* \) is regular if and only if

\[ L \text{ is represented by a regular expression, i.e.} \]

when there is \( \alpha \in \mathcal{R} \) such that \( L = \mathcal{L}(\alpha) \)

where \( \mathcal{L} : \mathcal{R} \rightarrow 2^{\Sigma^*} \) is the representation function

We use a shorthand notation

\[ L = \alpha \quad \text{for} \quad L = \mathcal{L}(\alpha) \]
Examples

E1
Given $\alpha \in \mathcal{R}$, for $\alpha = ((a \cup b)^*a)$

Evaluate $L$ over an alphabet $\Sigma = \{a, b\}$, such that $L = \mathcal{L}(\alpha)$
We write

$$\alpha = ((a \cup b)^*a)$$

in the shorthand notation as

$$\alpha = (a \cup b)^*a$$
Examples

We evaluate $L = (a \cup b)^* a$ as follows

$L((a \cup b)^* a) = L((a \cup b)^*) \circ L(a) = L((a \cup b)^*) \circ \{a\} =$

$(L(a \cup b))^* \{a\} = (L(a) \cup L(b))^* \{a\} = (\{a\} \cup \{b\})^* \{a\}$

Observe that

$(\{a\} \cup \{b\})^* \{a\} = \{a, b\}^* \{a\} = \Sigma^* \{a\}$

so we get

$L = L((a \cup b)^* a) = \Sigma^* \{a\}$

$L = \{w \in \{a, b\}^* : w \text{ ends with } a\}$
Examples

E2 Given $\alpha \in \mathcal{R}$, for $\alpha = ((c^*a) \cup (bc^*))^*$

Evaluate $L = \mathcal{L}(\alpha)$, i.e. describe $L = \alpha$

We write $\alpha$ in the shorthand notation as

$$\alpha = c^*a \cup (bc^*)^*$$

and evaluate $L = c^*a \cup (bc^*)^*$ as follows

$$\mathcal{L}((c^*a \cup (bc^*)^*)) = \mathcal{L}(c^*a) \cup (\mathcal{L}(bc^*))^* = \{c\}^*\{a\} \cup (\{b\}\{c\}^*)^*$$

and we get that

$$L = \{c\}^*\{a\} \cup (\{b\}\{c\}^*)^*$$
Examples

E3  Given $\alpha \in \mathcal{R}$, for

$$\alpha = (0^* \cup (((0^*(1 \cup (11))))((00^*(1 \cup (11)))*0^*))$$

Evaluate $L = \mathcal{L}(\alpha)$, i.e describe the language $L = \alpha$

We write $\alpha$ in the shorthand notation as

$$\alpha = 0^* \cup 0^*((1 \cup 11)((00^*(1 \cup 11))^*)0^*)$$

and evaluate

$$L = \mathcal{L}(\alpha) = 0^* \cup 0^*\{1, 11\}(00^*\{1, 11\})^*0^*$$

Observe that $00^*$ contains at least one $0$ that separates $0^*\{1, 11\}$ on the left with $(00^*(\{1, 11\})^*$ that follows it, so we get that

$$L = \{w \in \{0, 1\}^*: w \text{ does not contain a substring } 111\}$$
Class $\textbf{RL}$ of Regular Languages

**Definition**

Class $\textbf{RL}$ of regular languages over an alphabet $\Sigma$ contains all $L$ such that $L = \mathcal{L}(\alpha)$ for certain $\alpha \in \mathbb{R}$, i.e.

$$\textbf{RL} = \{ L \subseteq \Sigma^* : L = \mathcal{L}(\alpha) \text{ for certain } \alpha \in \mathbb{R} \}$$

**Theorem**

There $\aleph_0$ regular languages over $\Sigma \neq \emptyset$ i.e.

$$|\textbf{RL}| = \aleph_0$$

**Proof**

By definition that each regular language is $L = \mathcal{L}(\alpha)$ for certain $\alpha \in \mathbb{R}$ and the interpretation function $\mathcal{L} : \mathbb{R} \rightarrow 2^{\Sigma^*}$ has an infinitely countable domain, hence $|\textbf{RL}| = \aleph_0$
Class \textbf{RL} of Regular Languages

We can also think about languages in terms of \textit{closure} and get immediately from definitions the following

\textbf{Theorem}

Class \textbf{RL} of regular languages is the \textit{closure} of the set of languages

\[
\{\{\sigma\} : \sigma \in \Sigma\} \cup \{\emptyset\}
\]

with respect to union, concatenation and Kleene Star
Languages that are NOT Regular

Given an alphabet $\Sigma \neq \emptyset$

We have just proved that there are $\aleph_0$ regular languages, and we have also there are $\mathbb{C}$ of all languages over $\Sigma \neq \emptyset$, so we have the following

Fact
There is $\mathbb{C}$ languages that are not regular

Natural Questions
Q1 How to prove that a given language is regular?
A1 Find a regular expression $\alpha$, such that $L = \alpha$, i.e. $L = \mathcal{L}(\alpha)$
Languages that are NOT Regular

Q2 How to prove that a given language is not regular?

A2 Not easy!

We will have a Theorem, called Pumping Lemma which provides a criterium for proving that a given language is not regular.

E1 A language

\[ L = 0^*1^* \]

is regular as it is given by a regular expression \( \alpha = 0^*1^* \).

E2 We will prove, using the Pumping Lemma that the language

\[ L = \{0^n1^n : n \geq 1, \ n \in N\} \]

is not regular.
General Problem

Given a language $L$ over $\Sigma$ and a word $w \in \Sigma^*$, how to recognize whether

$$w \in L \quad \text{or} \quad w \notin L$$

Our Next Subject

Automata - LANGUAGE RECOGNITION devices