cse303
ELEMENTS OF THE THEORY OF COMPUTATION

Professor Anita Wasilewska

LECTURE 4

CHAPTER 1

SETS, RELATIONS, and LANGUAGES

7. Alphabets and Languages
8. Finite Representation of Languages

CHAPTER 1

PART 7: 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 Σ to denote the **alphabet**

Remember

Σ 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

E1 $\Sigma = \{\ddagger,\ \emptyset,\ \partial,\ \oint,\ \bigotimes,\ \vec{a},\ \nabla\}$

E2 $\Sigma = \{a,\ b,\ c\}$

E3 $\Sigma = \{n \in N: \quad n \leq 10^5\}$

E4 $\Sigma = \{0, 1\}$ 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 Σ

Σ is a finite set and we will write

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

Finite Sequences Revisited

Definition

A **finite sequence** of elements of a set A is any function

$f: \{1, 2, \ldots, n\} \longrightarrow A$ for $n \in N$

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 Σ

Let Σ be an **alphabet**

We call finite sequences of the alphabet Σ **words** or **strings** over Σ

We denote by e the **empty word** over Σ

Some books use symbol λ for the **empty word**

Words over Σ

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

We will write some words (strings) over Σ in a **shorthand** notaiton as for example

$$aaa, \ ab, \ bbb$$

instead using the formal definition:

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

such that $f(1) = a, f(2) = a, f(3) = a$ for the word aaa or $g: \{1,2\} \longrightarrow \Sigma$ such that $g(1) = b, g(2) = b$ for the word bb .. etc..

Words in Σ^*

Let Σ be an **alphabet**. We denote by

$$\Sigma^*$$

the set of **all finite** sequences over Σ

Elements of Σ^* are called **words** over Σ

We write $w \in \Sigma^*$ to express that w is a **word** over Σ

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 Σ^*

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** Σ,

$$\Sigma^* \neq \emptyset$$

Chapter 1

Some Short Questions and Answers

Short Questions

Q1 Let $\Sigma = \{a, b\}$

How many are there all possible **words** of length 5 over Σ ?

A1 By definition, words over Σ are finite sequences;

Hence words of a **length 5** are functions

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

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

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 Lecture 2, 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 Σ^*?

A2

By the **Counting Functions Theorem** there are

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

words of length $\leq n$ over Σ 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 Σ^*?

A3

There are **infinitely countably** many **words** in Σ^* 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 Σ^* of all words over Σ is **countably infinite**

Languages over Σ

Language Definition

Given an alphabet Σ, any set L such that

$$L \subseteq \Sigma^*$$

is called a **language over** Σ

Fact 1

For any alphabet Σ, any language over Σ is **countable**

Languages over Σ

Fact 2

For any alphabet $\Sigma \neq \emptyset$, there are uncountably many languages over Σ

More precisely, there are exactly $\mathcal{C} = |R|$ of **languages** over any non - empty alphabet Σ

Languages over Σ

Fact 1

For any alphabet Σ, any language over Σ 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 Σ, both finite, so **countable**
2. Let $\Sigma \neq \emptyset$, then Σ^* is countably infinite, so obviously any $L \subseteq \Sigma^*$ is finite or countably infinite, hence **countable**

Languages over Σ

Fact 2

For any alphabet $\Sigma \neq \emptyset$, there are exactly $\mathcal{C} = |R|$ of **languages**

over any non - empty alphabet Σ

Proof

We proved that $|\Sigma^*| = \aleph_0$

By definition $L \subseteq \Sigma^*$, so there is as many languages over Σ

as all subsets of a set of cardinality $\aleph_0$— that is

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

Languages over Σ

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

There is $\aleph_0$ **languages over** Σ

NO

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

$$\aleph_0 < \mathcal{C}$$

Some Basic Definitions

Some Basic Definitions

Definition

Given an alphabet Σ and a word $w \in \Sigma^*$

We say that w has a **length** $n = l(w) = |w|$ when

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

We re-write w as

$$w : \{1, 2, \ldots, |w|\} \longrightarrow \Sigma$$

Some Basic Definitions

We define now a **position** of $\sigma \in \Sigma$ in a word $w \in \Sigma^*$ as follows

Definition

Given $\sigma \in \Sigma$ and a word $w \in \Sigma^*$

$\sigma \in \Sigma$ 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

$$w = anita$$

By formal definition we have

$w(1) = a, \;\; w(2) = n, \;\; w(3) = i, \;\; w(4) = t, \;\; w(5) = a$

and a occurs in the 1st and 5th position

Some Examples

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

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

$w(1) = 0,\ w(2) = 1,\ w(3) = 1,\ w(4) = 0,\ w(5) = 1,$
$w(6) = 1,\ w(7) = 0,\ w(8) = 1$

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

0 occurs in the positions 1, 4, 7

Informal Concatenation

Concatenation (Informal Definition)

Given an alphabet Σ 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

Informal Concatenation

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 Σ 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 $j = 1, 2, \ldots, |x|$

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

Formal Concatenation

Properties

Directly from definition we have that

$$w \circ e = e \circ w = w$$

Concatenation of words is associative

$$(x \circ y) \circ z = x \circ (y \circ z) = x \circ y \circ z$$

Formal Concatenation

Remark

We need to define a concatenation of two words only 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

$$w = x_1 \circ x_2 \circ \cdots \circ x_n$$

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 recursively as follows

$$w^0 = e$$

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

This type of definition is called **definition by induction**

E8

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

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

$$w^R = au^R \text{ for } |u| < n + 1$$

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

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

$$(we)^R = (w)^R = ew^R = e^R w^R$$

Reversal Proof

Inductive Assumption

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

Let now $|x| = n + 1$, so $x = ua$ for certain $a \in \Sigma$, $u \in \Sigma^*$ and $|u| = n$

We evaluate

$$(wx)^R = (w(ua))^R = ((wu)a)^R$$

$$=^{def} a(wu)^R =^{ind} a(u^R w^R) = (au^R)w^R =^{def} (ua)^R w^R = x^R w^R$$

Languages over Σ

Definition

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

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

We have proved

Theorem

Any language L over Σ, is finite or infinitely countable

Languages over Σ

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

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

E2

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

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

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

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

and we have that all properties of **algebra of sets** hold for any languages over a given alphabet Σ

Special Operations on Languages

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

Concatenation of Languages

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

$$\Sigma = \Sigma_1 \cup \Sigma_2 \quad \text{and get} \quad L_1, L_2 \subseteq \Sigma^*$$

Concatenation Examples

E5

Let $L_1, \ L_2$ be languages defined below

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

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

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

Domain Σ 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

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

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

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

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

where

$x \in \{a, b\}^*$, $y \in \{0, 1\}^*$ 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

$L_1 = \{e, \ a, \ b\}, \quad L_2 = \{e, \ 0, \ 1, \ 01, \ 00, 11, \ 10\}$

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

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

and

$$L_2 = \{w \in \{0,1\}^* : \quad w = 0xx, \ x \in \Sigma^*\}$$

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

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

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

Observe that

$$1000 \in L_1 \circ L_2 \quad \text{and} \quad 1000 \notin L_2 \circ L_1$$

This proves that

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

We hence **proved** the following

Fact

Concatenation of languages **is not** commutative

Concatenation Examples

E8

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

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

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

Describe the language $L_2 \circ L_1$

Here it is

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

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

$$L_2 \circ L_1 = \{w \in \Sigma^* : \ \text{w starts with } 0 \text{ and ends with } 1\}$$

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 $n \in N$, $n \geq 2$

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 $\circ$)

$(w \in L_1 \; or \; w \in L_2) \; and \; w \in L$ iff (by distributivity of and over or)

$(w \in L_1 \, and \; w \in L) \; or \; (w \in L_2 \; and \; w \in L)$ iff (by definition of $\circ$)

$(w \in L_1 \circ L) or \; (w \in L_2 \circ L)$ 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

$$L^* = \{w_1 w_2 \dots w_k : \text{for some} \quad k \geq 0 \quad \text{and} \quad w_1, \dots, w_k \in L\}$$

We also write as

$$L^* = \{w_1 w_2 \dots w_k : \quad k \geq 0, \quad w_i \in L, \quad i = 1, 2, \dots, k\}$$

or in a form of Generalized Union

$$L^* = \bigcup_{k \geq 0} \{w_1 w_2 \dots w_k : \quad w_1, \dots, w_k \in L\}$$

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

$$01 \notin L^* \quad \text{and} \quad 01 \in \Sigma^*$$

Example

Observation

The property **P4** provides a quite trivial example of a language L over an alphabet Σ 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

$L = \{w \in \{0,1\}^* : \ w$ 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 the following property of languages:

P: If $L_1 \subseteq L_2$, then $L_1{}^\star \subseteq L_2{}^\star$

and get that

$$L^* \subseteq \{0,1\}^{**} = \{0,1\}^* \text{ i.e. } L^* \subseteq \Sigma^*$$

Example Proof

Now we have to show that $\Sigma^* \subseteq L^*$, i.e.

$\{0, 1\}^* \subseteq \{w \in 0, 1^* : \; w$ has an unequal number of 0 and $1\}$

Observe that

$0 \in L$ because 0 regarded as a string over Σ 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

$L^+ = \{w_1 w_2 \ldots w_k$: for some $k \geq 1$ 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 : $L^+ = L \circ L^*$ **P2** : $e \in L^+$ iff $e \in L$

L^* and L^+

We know that

$$e \in L^* \quad \text{for all } L$$

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
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**:

$$\{,\ \ \},\ (,\ \ ),\ *,\ \ \cup$$

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

$$(,\ ),\ *,\ \cup,\ \emptyset, \quad \text{and symbols } \sigma \in \Sigma$$

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 has **two** or **three** occurrences of 1 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 **finitely represent** all languages over an alphabet $\Sigma \neq \emptyset$?

Observation

O1. Different languages must have different representations

O2. **Finite representations** are finite strings over a finite set, so we have that

there are $\aleph_0$ possible **finite representations**

Problem with Finite Representation

O3. There are **uncountably** many, precisely exactly $\mathcal{C} = |R|$) 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 Σ^* that is as many as

$$|2^{\Sigma^*}| = 2^{\aleph_0} = \mathcal{C}$$

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 $\mathcal{C}$) 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. $\mathcal{C}$ 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

Definition

We define a $\mathcal{R}$ of **regular expressions** over an alphabet Σ 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

$$\emptyset \in \mathcal{R} \text{ and } \forall_{\sigma \in \Sigma} (\sigma \in \mathcal{R})$$

2. If $\alpha, \beta \in \mathcal{R}$, then

$(\alpha\beta) \in \mathcal{R}$ **concatenation**

$(\alpha \cup \beta) \in \mathcal{R}$ **union**

$\alpha^* \in \mathcal{R}$ **Kleene's Star**

Regular Expressions Theorem

Theorem

The set $\mathcal{R}$ of **regular expressions** over an alphabet Σ 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. $\emptyset \in \mathcal{R}, \quad 1 \in \mathcal{R}, \quad 0 \in \mathcal{R}$

2. $(01) \in \mathcal{R} \quad 1^* \in \mathcal{R}, \quad 0^* \in \mathcal{R}, \quad \emptyset^* \in \mathcal{R}, \quad (\emptyset \cup 1) \in \mathcal{R}, \ldots,$

$\ldots, \quad (((0^* \cup 1^*) \cup \emptyset)1)^* \in \mathcal{R}$

Shorthand Notation when writing regular expressions we will **keep only** essential parenthesis

For example, we will write

$$((0^* \cup 1^* \cup \emptyset)1)^* \quad \text{instead of} \quad (((0^* \cup 1^*) \cup \emptyset)1)^*$$

$$1^*01^* \cup (01)^* \quad \text{instead of} \quad (((1^*0)1^*) \cup (01)^*)$$

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

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 establish by a **function** $\mathcal{L}$ such that if $\alpha \in \mathcal{R}$ is any regular expression, then $\mathcal{L}(\alpha)$ is the **language represented** by α

Definition of Representation

Formal Definition

The function $\mathcal{L} : \mathcal{R} \longrightarrow 2^{\Sigma^*}$ is defined recursively as follows

1. $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\sigma) = \{\sigma\}$ for all $\sigma \in \Sigma$

2. If $\alpha, \beta \in \mathcal{R}$, then

$$\mathcal{L}(\alpha\beta) = \mathcal{L}(\alpha) \circ \mathcal{L}(\beta)$$ **concatenation**

$$\mathcal{L}(\alpha \cup \beta) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta)$$ **union**

$$\mathcal{L}(\alpha^*) = \mathcal{L}(\alpha)^*$$ **Kleene's Star**

Regular Language Definition

Definition

A language $L \subseteq \Sigma^*$ is **regular**

if and only if

L is represented by a **regular expression**, i.e.

when there is $\alpha \in \mathcal{R}$ such that $L = \mathcal{L}(\alpha)$

where $\mathcal{L} : \mathcal{R} \longrightarrow 2^{\Sigma^*}$ is the **representation function**

We use a **shorthand notation**

$L = \alpha$ for $L = \mathcal{L}(\alpha)$

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

$$\mathcal{L}((a \cup b)^* a) = \mathcal{L}((a \cup b)^*) \circ \mathcal{L}(a) = \mathcal{L}((a \cup b)^*) \circ \{a\} =$$

$$(\mathcal{L}(a \cup b))^* \{a\} = (\mathcal{L}(a) \cup \mathcal{L}(b))^* \{a\} = (\{a\} \cup \{b\})^* \{a\}$$

Observe that

$$(\{a\} \cup \{b\})^* \{a\} = \{a, b\}^* \{a\} = \Sigma^* \{a\}$$

so we get

$$L = \mathcal{L}((a \cup b)^* a) = \Sigma^* \{a\}$$

$$L = \{w \in \{a, b\}^* : \ w \text{ ends with } a\}$$

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 α 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 α 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

$$L = \{w \in \{0,\ 1\}^* : \ w \text{ does not contain a substring } 111\}$$