Theoretical Computer Science

Tutorial Week 2

Prof. Andrey Frolov

INNOPOLIS UNIVERSITY

Agenda

- **Alphabets and Strings**
- Formal Languages
- Operations

Alphabets and Strings

Definition

An **alphabet** is a finite set of symbols

Examples

$$\{0, 1\}$$

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\{a, b, c, d, \ldots, x, y, z\}$$

Alphabets and Strings

Definition

A **string** over an alphabet Σ is a finite sequence of symbols in Σ

Examples

For $\Sigma = \{0, 1\}$,

$$010011$$

$$11100011$$

Alphabets and Strings

Definition

A **string** (word) over an alphabet Σ is a finite sequence of symbols in Σ

Examples

For $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$,

$$123456$$

$$666$$

$$2023$$

Alphabets and Strings

Definition

A **string** over an alphabet Σ is a finite sequence of symbols in Σ

Examples

For $\Sigma = \{a, b, c, d, \ldots, x, y, z\}$,

$$peace$$

$$war$$

$$dfklgnkjrbgjrbg$$

Alphabets and Strings

Definition

The **length** of a string s is the number of symbols of s and denotes as $|s|$

Examples

$$|peace| = 5$$

$$|war| = 3$$

$$|dfklgnkjrbgjrbg| = 15$$

Alphabets and Strings

Definition

ϵ is the **null** string (empty string) over any alphabet.

Property

$$|\epsilon| = 0$$

Alphabets and Strings

Definition

For two strings x and y, the concatenation $x \cdot y$ is the operation of joining "end-to-end".

Examples

For $x =$"123" and $y =$"987",

$$x \cdot y = \text{"123987"}$$

Alphabets and Strings

Definition

For two strings x and y, the concatenation $x \cdot y$ is the operation of joining “end-to-end”.

Examples

For $x =$ “*back*” and $y =$ “*end*”,

$$x \cdot y = \text{“}\textit{backend}\text{”}$$

$$y \cdot x = \text{“}\textit{endback}\text{”}$$

Alphabets and Strings

Definition

For two strings x and y, the concatenation $x \cdot y$ is the operation of joining "end-to-end".

Examples

For $x =$ "*back*" and $y =$ "*end*",

$$x \cdot y = \text{``}\textit{backend}\text{''}$$

$$y \cdot x = \text{``}\textit{endback}\text{''}$$

Non-commutative!

Alphabets and Strings

Property

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

Examples

For $x =$ "*ab*", $y =$ "*cd*" and $z =$ "*ef*",

$$(x \cdot y) \cdot z = \text{``}abcd\text{''} \cdot \text{``}ef\text{''} = \text{``}abcdef\text{''}$$

$$x \cdot (y \cdot z) = \text{``}ab\text{''} \cdot \text{``}cdef\text{''} = \text{``}abcdef\text{''}$$

Alphabets and Strings

Property

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

Examples

For $x =$ "*ab*", $y =$ "*cd*" and $z =$ "*ef*",

$$(x \cdot y) \cdot z = \text{"}abcd\text{"} \cdot \text{"}ef\text{"} = \text{"}abcdef\text{"}$$

$$x \cdot (y \cdot z) = \text{"}ab\text{"} \cdot \text{"}cdef\text{"} = \text{"}abcdef\text{"}$$

Associative!

Alphabets and Strings

Property with null

$$\forall x \, (x \cdot \epsilon = \epsilon \cdot x = x)$$

ϵ is an identity element

Agenda

- Alphabets and Strings
- **Formal Languages**
- Operations

Formal Languages

Definition

The set of all strings over Σ is denoted by Σ^*

Examples

For $\Sigma = \{0, 1\}$,

$$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, \ldots\}$$

Formal Languages

Definition

A language L is a **set** of strings over an alphabet Σ.

Equivalent definition

$$L \subseteq \Sigma^*$$

The naive set theory

Definition of a set

$$A = \{x \in \mathbf{U} \mid P(x)\}$$

$$A = \{a_1, a_2, \ldots, a_n\}$$

Example

$$\{x \in \mathbb{Z} \mid x < 0\}$$

Formal Languages

Alphabet

For $\Sigma = \{0, 1\}$,

$$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, \ldots\}$$

Languages

$$L_1 = \{00000000, 00000001, \ldots, 11111110, 11111111\} =$$

$$= \{x \in \{0,1\}^* \mid |x| = 8\}$$

$$L_2 = \{0, 00, 01, 000, 001, 010, \ldots\} = \{0x \mid x \in \Sigma^*\}$$

Formal Languages

Alphabet

For $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$,

$$\Sigma^* = \mathbb{N} \cup \{\epsilon\}$$

Languages

$$L_1 = \{0, 2, 4, 6, 8, 10, \ldots\} = \{x \in \Sigma^* \mid x \text{ is even } \}$$

$$L_2 = \{2, 3, 5, 7, 13, \ldots\} = \{x \in \Sigma^* \mid x \text{ is prime } \}$$

Formal Languages

Alphabet

For $\Sigma = \{a, b, c, d, \ldots, x, y, z\}$

Languages

English, Italian, French,...

Formal Languages

Alphabet

For $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, =\}$

Arithmetic

$$\{0 + 0 = 0, 0 - 0 = 0, \ldots, 12 + 32 = 44, \ldots, 52 - 39 = 13, \ldots\}$$

Agenda

- Alphabets and Strings
- Formal Languages
- **Operations**
  - Operations from Set Theory
  - Special operations

1. Complement

Complement of a set

$$A^c = \overline{A} = \{x \in \mathbf{U} \mid x \notin A\}$$

Example

If $\mathbf{U} = \{1, 2, 3, 4\}$ and $A = \{1, 3\}$, then

$$\overline{A} = \{2, 4\}$$

1. Complement

Complement of a language

For an alphabet Σ,

$$L^c = \overline{L} = \{x \in \Sigma^* \mid x \notin L\}$$

Example

For $\Sigma = \{0,1\}$, if $L = \{0x \mid x \in \Sigma^*\}$, then

$$\overline{L} =$$

1. Complement

Complement of a language

For an alphabet Σ,

$$L^c = \overline{L} = \{x \in \Sigma^* \mid x \notin L\}$$

Example

For $\Sigma = \{0, 1\}$, if $L = \{0x \mid x \in \Sigma^*\}$, then

$$\overline{L} = \{1x \mid x \in \Sigma^*\} \cup \{\epsilon\}$$

The naive set theory

Union

$$A \cup B = \{x \in \mathbf{U} \mid x \in A \vee x \in B\}$$

Example

If $A = \{1, 2, 3\}$ and $B = \{2, 3, 4\}$, then

$$A \cup B = \{1, 2, 3, 4\}$$

Intersection

$$A \cap B = \{x \in \mathbf{U} \mid x \in A \,\&\, x \in B\}$$

Example

If $A = \{1, 2, 3\}$ and $B = \{2, 3, 4\}$, then

$$A \cap B = \{2, 3\}$$

The naive set theory

Difference

$$A \setminus B = \{x \in \mathbf{U} \mid x \in A \,\&\, x \notin B\}$$

Example

If $A = \{1, 2, 3\}$ and $B = \{2, 3, 4\}$, then

$$A \setminus B = \{1\}$$

Formal Languages

2. Union

$$L_1 \cup L_2 = \{s \in \Sigma^* \mid s \in L_1 \vee L_2\}$$

3. Intersection

$$L_1 \cap L_2 = \{s \in \Sigma^* \mid s \in L_1 \,\&\, s \in L_2\}$$

4. Difference

$$L_1 \setminus L_2 = \{s \in \Sigma^* \mid s \in L_1 \,\&\, s \notin L_2\}$$

The naive set theory

Definition

$X \times Y = \{(x, y) \mid x \in X \,\&\, y \in Y\}$

Example

If $A = \{1, 2, 3\}$ and $B = \{a, b\}$, then

$$A \times B = \{(1, a), (1, b), (2, a), (2, b), (3, a), (3, b)\}$$

Definition

$$X_1 \times \cdots \times X_n = \{(x_1, \ldots, x_n) \mid x_1 \in X_1 \mathbin{\&} \ldots \mathbin{\&} x_n \in X_n\}$$

Example

$$\underbrace{X \times \cdots \times X}_{n \text{ times}} = X^n$$

The naive set theory

Definition

For a set A, the power of A is the set

$$2^A = \mathcal{P}(A) = \{B \mid B \subseteq A\}$$

Examples

1) If $A = \{a\}$ then $\mathcal{P}(A) = \{\emptyset, \{a\}\}$
2) If $A = \{a, b\}$ then $\mathcal{P}(A) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$

The naive set theory

Definition

Intuitively, the cardinality of a set A, denotes by $|A|$, is the number of elements of A.

Examples

1. $|\emptyset| = 0$
2. if $A = \{2\}$ then $|A| = 1$
2. if $A = \{1, 2, 3\}$ then $|A| = 3$
3. $|\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}| = \omega$
4. $|\mathbb{R}| = 2^{\omega}$

Relationship

Question 1

What different between sets and strings?

Question 2

What different between $\emptyset$ and ϵ?

Question 3

What different between the cardinality and the length?

Agenda

- Alphabets and Strings
- Formal Languages
- Operations
  - Operations from Set Theory
  - **Special operations**

Formal Languages

Concatenation

$$L_1 \cdot L_2 = \{x \cdot y \mid x \in L_1 \,\&\, y \in L_2\}$$

Example

If $L_1 = \{1, 2, 3\}$ and $L_2 = \{a, b\}$, then

$$L_1 \cdot L_2 = \{1a, 1b, 2a, 2b, 3a, 3b\}$$

Formal Languages

Concatenation

$$L_1 \cdot L_2 = \{x \cdot y \mid x \in L_1 \,\&\, y \in L_2\}$$

Example

If $L_1 = \{1, 12\}$ and $L_2 = \{\epsilon, 2\}$, then

$$L_1 \cdot L_2 = \{1, 12, 122\}$$

Concatenation

$$L_1 \cdot L_2 = \{x \cdot y \mid x \in L_1 \,\&\, y \in L_2\}$$

Example

If $L_1 = \{\epsilon, a\}$ and $L_2 = \{\epsilon, a, aa, aaa, \ldots\}$, then

$$L_1 \cdot L_2 =$$

Concatenation

$$L_1 \cdot L_2 = \{x \cdot y \mid x \in L_1 \,\&\, y \in L_2\}$$

Example

If $L_1 = \{\epsilon, a\}$ and $L_2 = \{\epsilon, a, aa, aaa, \ldots\}$, then

$$L_1 \cdot L_2 = L_2$$

$L_1 \times L_2 \neq L_2$ for any nonempty L_1, L_2

Formal Languages

Kleene star

$$L^* = \{x_1 x_2 \ldots x_n \mid n \in \mathbb{N}, x_1, x_2, \ldots x_n \in L\}$$

Example

For $\Sigma = \{a, b\}$, if $L = \{a\}$ then

$$L^* = \{\epsilon, a, aa, aaa, \ldots\}$$

Kleene star

$$L^* = \{x_1 x_2 \ldots x_n \mid n \in \mathbb{N}, x_1, x_2, \ldots x_n \in L\}$$

Example

For $\Sigma = \{a, b\}$, if $L = \{ab\}$ then

$$L^* = \{\epsilon, ab, abab, ababab, \ldots\}$$

Kleene star

Let Σ be an alphabet. Kleene star of Σ contains all strings and denotes Σ^* (as before).

Special case

$\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$

Special cases

$$L^k = \{x_1x_2\dots x_k \mid x_1, x_2, \dots x_k \in L\}$$

Example

For $\Sigma = \{a, b\}$, if $L = \{a\}$ then

$$L^k = \{\underbrace{aa\dots a}_{k \text{ times}}\}$$

Formal Languages

Special cases

$$L^k = \{x_1 x_2 \ldots x_k \mid x_1, x_2, \ldots x_k \in L\}$$

Example

For $\Sigma = \{a, b\}$,

$$\Sigma^2 =$$

Formal Languages

Special cases

$$L^k = \{x_1x_2\ldots x_k \mid x_1, x_2, \ldots x_k \in L\}$$

Example

For $\Sigma = \{a, b\}$,

$$\Sigma^2 = \{aa, ab, ba, bb\}$$

Formal Languages

Special cases

$$a^k = \underbrace{aa\ldots a}_{k \text{ times}}$$

Example

For $\Sigma = \{a, b\}$,

$$\Sigma^3 =$$

Formal Languages

Special cases

$$a^k = \underbrace{aa\ldots a}_{k \text{ times}}$$

Example

For $\Sigma = \{a, b\}$,

$$\Sigma^3 = \{a^3, a^2b, aba, ab^2, ba^2, bab, b^2a, b^3\}$$

Languages in Computer Science

Example

Formal Languages

Rewriting identities

$$\begin{cases} a_1 a_2 \ldots a_n & = b_1 b_2 \ldots b_m \\ \ldots & = \ldots \\ c_1 c_2 \ldots c_k & = d_1 d_2 \ldots d_l \end{cases}$$

Example

For $\Sigma = \{a, b\}$,

$$\begin{cases} ab & = b^2 a \\ a^3 & = b^2 \end{cases}$$

$abab = b^2 abba = b^4 aba = b^6 aba = b^8 aa = (b^2)^4 a^2 = a^1 2 a^2 = a^{14}$

Rewriting identities

$$\begin{cases} a_1 a_2 \ldots a_n & = b_1 b_2 \ldots b_m \\ \ldots & = \ldots \\ c_1 c_2 \ldots c_k & = d_1 d_2 \ldots d_l \end{cases}$$

The word problem is

the problem of deciding whether two given expressions are equivalent with respect to a set of rewriting identities.

Formal Languages

Rewriting identities

$$\begin{cases} a_1 a_2 \dots a_n & = b_1 b_2 \dots b_m \\ \dots & = \dots \\ c_1 c_2 \dots c_k & = d_1 d_2 \dots d_l \end{cases}$$

The word problem is

the problem of deciding whether two given expressions are equivalent with respect to a set of rewriting identities.

Undecidable!!!

Formal Languages

Alphabet

For $\Sigma = \{a, b, c, \ldots, x, y, z, +, -, \cdot, /\}$,

$$\begin{cases} a(b+c) & = ab + ac \\ (a+b)^2 & = a^2 + 2ab + b^2 \\ \cdots & = \cdots \end{cases}$$

Problem

Are there an algorithm solving $\Phi_1 = \Phi_2$ for two algebraic formulas?

Thank you for your attention!