Languages

Language: a set of strings

String: a sequence of symbols from some alphabet

Example:

Strings: cat, dog, house

Language: {cat, dog, house}

Alphabet: $\Sigma = \{a, b, c, \ldots, z\}$

Languages are used to describe computation problems:

$$PRIMES = \{2,3,5,7,11,13,17,\ldots\}$$

$$EVEN = \{0,2,4,6,\ldots\}$$

Alphabet: $\Sigma = \{0,1,2,\ldots,9\}$

Computation is translated to set membership

Example computation problem:

Is number x prime?

Equivalent set membership problem:

$$x \in PRIMES = \{2,3,5,7,11,13,17,\ldots\}?$$

Alphabets and Strings

An alphabet is a set of symbols

Example Alphabet: $\Sigma = \{a, b\}$

A string is a sequence of symbols from the alphabet

Example Strings

a

ab

$abba$

$aaabbbaabab$

String variables

$u = ab$

$v = bbbaaa$

$w = abba$

Decimal numbers alphabet $\Sigma = \{0,1,2,\ldots,9\}$

102345 $\qquad$ 567463386

Binary numbers alphabet $\Sigma = \{0,1\}$

100010001 $\qquad$ 101101111

Unary numbers alphabet $\Sigma = \{1\}$

Unary number:	1	11	111	1111	11111
Decimal number:	1	2	3	4	5

String Operations

$$w = a_1 a_2 \cdots a_n \qquad aaa$$

$$v = b_1 b_2 \cdots b_m \qquad bbb$$

Concatenation

$$wv = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m \qquad aaabbb$$

$$v = a_1 a_2 \cdots a_n \qquad w = ababaaabbb$$

Reverse

$$w^R = bbbaaababa$$

String Length

$$w = a_1 a_2 \cdots a_n$$

Length: $|w| = n$

Examples:

$$|abba| = 4$$

$$|aa| = 2$$

$$|a| = 1$$

Length of Concatenation

$$|uv| = |u| + |v|$$

Example:

$$u = aab, \quad |u| = 3$$

$$v = abaab, \quad |v| = 5$$

$$|uv| = |aababaab| = 8$$

$$|uv| = |u| + |v| = 3 + 5 = 8$$

Empty String

A string with no letters is denoted: λ or ε

Acts as a neutral element

Observations: $|\lambda| = 0$

$$\lambda w = w\lambda = w$$

$$\lambda abba = abba\lambda = ab\lambda ba = abba$$

Substring

Substring of string:
a subsequence of consecutive characters

String	Substring
$\underline{ab}bab$	ab
$\underline{abba}b$	$abba$
$ab\underline{b}ab$	b
$a\underline{bbab}$	$bbab$

Prefix and Suffix

string $abbab$

Prefixes	Suffixes
λ	$abbab$
a	$bbab$
ab	bab
abb	ab
$abba$	b
$abbab$	λ

$$w = \underbrace{u}_{\text{prefix}}\underbrace{v}_{\text{suffix}}$$

Exponent Operation

$$w^n = \underbrace{ww \cdots w}_{n}$$

Example: $(abba)^2 = abbaabba$

Definition: $w^0 = \lambda$

$$(abba)^0 = \lambda$$

The * Operation

Σ^* : the set of all possible strings from alphabet Σ

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

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

The + Operation

Σ^+ : the set of all possible strings from alphabet Σ except λ

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

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

$$\Sigma^+ = \Sigma^* - \lambda$$

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

Powers of an alphabet

Let Σ be an alphabet.

- Σ^k = the set of all strings of length k

- $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots$

- $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$

Languages

A language over alphabet Σ

is any subset of Σ^*

Example: $\Sigma = \{a, b\}$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, \ldots\}$$

Languages: $\{\}$

$\{\lambda\}$

$\{a, aa, aab\}$

$\{\lambda, abba, baba, aa, ab, aaaaaa\}$

More Language Examples

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

An infinite language $L = \{a^n b^n : n \geq 0\}$

$$\left.\begin{array}{l} \lambda \\ ab \\ aabb \\ aaaaabbbbb \end{array}\right\} \in L$$

$$bbabb \notin L$$

$$abb \notin L$$

Prime numbers

Rule: Numbers divisible by 1 and itself

Alphabet $\Sigma = \{0,1,2,\ldots,9\}$

Language:

$$PRIMES = \{x : x \in \Sigma^{*} \text{ and } x \text{ is prime}\}$$

$$PRIMES = \{2,3,5,7,11,13,17,\ldots\}$$

Even and odd numbers

Alphabet $\Sigma = \{0,1,2,\ldots,9\}$

Languages:

$$EVEN = \{x : x \in \Sigma^* \text{ and } x \text{ is even}\}$$

$$EVEN = \{0,2,4,6,\ldots\}$$

$$ODD = \{x : x \in \Sigma^* \text{ and } x \text{ is odd}\}$$

$$ODD = \{1,3,5,7,\ldots\}$$

Addition (of unary numbers)

Alphabet: $\Sigma = \{1, +, =\}$

Language:

$$ADDITION = \{x + y = z : \mathrm{x} = 1^{\mathrm{n}}, \mathrm{y} = 1^{\mathrm{m}}, \mathrm{z} = 1^{k},$$
$$\mathrm{n} + m = k, n \geq 1, m \geq 1\}$$

$$11 + 111 = 11111 \in ADDITION$$

$$111 + 111 = 111 \notin ADDITION$$

$$ADDITION = \{1 + 1 = 11, 1 + 11 = 111, 11 + 1 = 111, 11 + 11 = 1111, \ldots\}$$

Squares (of unary numbers)

Alphabet: $\Sigma = \{1, \#\}$

Language:

$$SQUARES = \{x\#y \ : x = 1^n, y = 1^m, m = n^2\}$$

$$11\#1111 \in SQUARES$$

$$111\#1111 \notin SQUARES$$

$$SQUARES = \{1\#1, 11\#1111, 111\#111111111, \ldots\}$$

Two special languages

Empty language

$\{\,\}$ or $\varnothing$

Language with empty string

$\{\lambda\}$

Size of a language (number of elements):

$$|\{\,\}| = 0$$

$$|\{\lambda\}| = 1$$

$$|\{a, aa, ab\}| = 3$$

$$|\{\lambda, aa, bb, abba, baba\}| = 5$$

Note that:

Sets $\emptyset = \{\ \} \neq \{\lambda\}$

Set size $|\{\ \}| = |\emptyset| = 0$

Set size $|\{\lambda\}| = 1$

String length $|\lambda| = 0$

Operations on Languages

The usual set operations:

$$\{a, ab, aaaa\} \cup \{bb, ab\} = \{a, ab, bb, aaaa\}$$ union

$$\{a, ab, aaaa\} \cap \{bb, ab\} = \{ab\}$$ intersection

$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$ difference

Complement: $\overline{L} = \Sigma^* - L$

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \ldots\}$$

Reverse

Definition: $L^R = \{w^R : w \in L\}$

Examples: $\{ab, aab, baba\}^R = \{ba, baa, abab\}$

$$L = \{a^n b^n : n \geq 0\}$$

$$L^R = \{b^n a^n : n \geq 0\}$$

Concatenation

Definition: $$L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$$

Example: $$\{a, ab, ba\}\{b, aa\}$$

$$= \{ab, aaa, abb, abaa, bab, baaa\}$$

Another Operation

Definition: $L^n = \underbrace{LL \cdots L}_{n}$

$$\{a,b\}^3 = \{a,b\}\{a,b\}\{a,b\} =$$
$$\{aaa,aab,aba,abb,baa,bab,bba,bbb\}$$

Special case: $L^0 = \{\lambda\}$

$$\{a,bba,aaa\}^0 = \{\lambda\}$$

Example

$$L = \{a^n b^n : n \geq 0\}$$

$$L^2 = \{a^n b^n a^m b^m : n, m \geq 0\}$$

$$aabbaaabbb \in L^2$$

$$??????? \notin L^2$$

Star-Closure (Kleene *)

All strings that can be constructed from L

Definition: $L^* = L^0 \cup L^1 \cup L^2 \cup \cdots$

Example:

$$\{a, bb\}^* = \left\{ \begin{array}{l} \lambda, \\ a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \cdots \end{array} \right\} \begin{array}{l} L^0 \\ L^1 \\ L^2 \\ L^3 \end{array}$$

Kleene's Closure/ Kleene's Star

- Given Σ, then the Kleene Star Closure of the alphabet Σ, denoted by Σ^*, is the collection of all strings defined over Σ, including Λ.
- It is to be noted that Kleene Star Closure can be defined over any set of strings.

Examples

If $\Sigma = \{x\}$

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots$

Then $\Sigma^* = \{\Lambda, x, xx, xxx, xxxx, \ldots.\}$

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

Then $\Sigma^* = \{\Lambda, 0, 1, 00, 01, 10, 11, \ldots.\}$

If $\Sigma = \{aaB, c\}$

Then $\Sigma^* = \{\Lambda, aaB, c, aaBaaB, aaBc, caaB, cc, \ldots.\}$

Note

Languages generated by Kleene Star Closure of set of strings, are infinite languages. (By infinite language, it is supposed that the language contains infinite many words, each of finite length).

Positive Closure

Definition: $$L^{+} = L^{1} \cup L^{2} \cup L^{3} \cup \ldots$$

Note that: $L^{*} = L^{0} \cup L^{+}$

$$\{a, bb\}^{+} = \left\{ \begin{array}{l} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\} \begin{array}{l} L^{1} \\ L^{2} \\ L^{3} \end{array}$$

Valid/In-valid alphabets

While defining an alphabet, an alphabet may contain letters consisting of group of symbols for

Example

$\Sigma 1 = \{B, aB, bab, d\}$.

Now consider an alphabet

$\Sigma 2 = \{B, Ba, bab, d\}$ and a string **BababB**.

This string can be tokenized in two different ways

(Ba), (bab), (B)

(B), (abab), (B)

Which shows that the second group cannot be identified as a string, defined overΣ = {a, b}.

As when this string is scanned by the compiler (Lexical Analyzer), first symbol B is identified as a letter belonging to Σ, while for the second letter the lexical analyzer would not be able to identify, so while defining an alphabet it should be kept in mind that ambiguity should not be created.

Remarks

- While defining an alphabet of letters consisting of more than one symbols, no letter should be started with the letter of the same alphabet *i.e. one letter should not be the prefix of another. However, a letter may be ended in a* letter of same alphabet.

Conclusion

Σ1= {B, aB, bab, d}

Σ2= {B, Ba, bab, d}

Σ1 is a valid alphabet while Σ2 is an in-valid alphabet.

Descriptive definition of language

The language is defined, describing the conditions imposed on its words.

Example

The language L of strings of odd length, defined over $\Sigma=\{a\}$, can be written as

$L=\{a, aaa, aaaaa,\ldots..\}$

Example

The language L of strings that does not start with a, defined over $\Sigma =\{a,b,c\}$, can be written as

$L =\{ b, c, ba, bb, bc, ca, cb, cc, \ldots\}$

Example

The language L of strings of length 2, defined over Σ ={0,1,2}, can be written as

L={00, 01, 02,10, 11,12,20,21,22}

Example

The language L of strings ending in 0, defined over Σ ={0,1}, can be written as

L={00,10,000,010,100,110,…}

Example

The language **EQUAL**, of strings with number of a's equal to number of b's, defined over Σ={a,b}, can be written as

L={Λ ,ab,ba,aabb,abab,baba,abba,…}

Example

The language **EVEN-EVEN**, of strings with even number of a's and even number of b's, defined over Σ={a,b}, can be written as

L={Λ, aa, bb, aaaa,aabb,abab, abba, baab, baba, bbaa,bbbb,…}

Example

The language **INTEGER**, of strings defined over $\Sigma=\{-,0,1,2,3,4,5,6,7,8,9\}$, can be written as

INTEGER = {…,-2,-1,0,1,2,…}

Example

The language **EVEN**, of stings defined over $\Sigma=\{-,0,1,2,3,4,5,6,7,8,9\}$, can be written as

EVEN = { …,-4,-2,0,2,4,…}

Example

The language $\{a^n b^n\}$, of strings defined over $\Sigma=\{a,b\}$, as $\{a^n b^n : n=1,2,3,\ldots\}$, can be written as

L={ab, aabb, aaabbb,aaaabbbb,…}

Example

The language $\{a^n b^n a^n\}$, of strings defined over $\Sigma=\{a,b\}$, as $\{a^n b^n a^n: n=1,2,3,\ldots\}$, can be written as

L={aba, aabbaa, aaabbbaaa,aaaabbbbaaaa,…}

Example

The language **factorial**, of strings defined over $\Sigma=\{0,1,2,3,4,5,6,7,8,9\}$ *i.e.*

$\{1,2,6,24,120,\ldots\}$

Example

The language **FACTORIAL**, of strings defined over $\Sigma=\{a\}$, as $\{a^n! : n=1,2,3,\ldots\}$, can be written as $\{a,aa,aaaaaa,\ldots\}$.

It is to be noted that the language FACTORIAL can be defined over any single letter alphabet.

Example

The language **DOUBLE-FACTORIAL**, of strings defined over $\Sigma=\{a, b\}$, as $\{a^n!b^n! : n=1,2,3,\ldots\}$, can be written as

$\{ab, aabb, aaaaaabbbbbb,\ldots\}$

CLASS ACTIVITY#01