Basic Concepts: Alphabets, Strings, and Language

<u>Alphabet:</u> A finite and nonempty set of symbols denoted by **Σ**. The elements of an alphabet are ***letters,*** but sometimes are named also ***symbols.***

e.g.,

$\emptyset=\{\}$	ليست أبجدية لأنها خالية (لا تحقق التعريف)
$\Sigma=\{a, b, \ldots, z\}$	أبجدية اللغة الانكليزية (أحرفها الصغيرة فقط)
$\Sigma=\{0, 1\}$	أبجدية نظام العد الثنائي
Σ={ا, ب, ..., ي}	أبجدية اللغة العربية
$\Sigma=\{1, 2, 3, \ldots, 9\}$	أبجدية نظام العد العشري
N مجموعة الاعداد الطبيعية	ليست أبجدية لأنها مجموعة غير منتهية
$\Sigma=\{0, 1, 01\}$	ليست أبجدية لأنها تحوي عنصر قابل للتجزئة (01) وبالتالي هو ليس رمزاً

<u>Strings (words):</u> is a finite ordered sequence of symbols from a given Alphabet. Note that repetitions are allowed. The words (strings) $u = a_1a_2 \ldots a_m$ and $v = b_1b_2 \ldots b_n$ are equal (i.e., $u = v$), if $m = n$ and $a_i = b_i$, $i = 1, 2, \ldots, n$.

Some Important Notations

ε (Empty String): is a special string its length is 0, some text denoted empty string with (λ).

Σ^*: All strings composed from the alphabet Σ with length >= 0.

Σ^+: All strings composed from the alphabet Σ with length > 0.

e.g., Let Σ= {0,1}

$\{0, 1\}^*$ is all strings over (all strings composed from) {0,1}

$\{0, 1\}^* =\{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, \ldots\ldots\}$

$\{0, 1\}^+ =\{0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, \ldots\ldots\}$

Σ_i : all strings over Σ with length i.

e.g.,: Suppose Σ = {a, b, c}

$\Sigma_0 = \{\varepsilon\}$

$\Sigma_1 = \{a, b, c\}$

$\Sigma_2 = \{a, b, c\}_2 = \{aa, ab, ac, ba, bb, bc, ca, cb, cc, \ldots\ldots\}$

$\Sigma_3 = \{a,b\}_3 = \{aaa, aab, aba, abb, baa, bab, bba, bbb, \ldots\ldots\}$

$\Sigma^* = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \ldots\ldots \cup \Sigma_n \cup \Sigma_{n+1} \cup \ldots\ldots$

$\Sigma^+ = \Sigma_1 \cup \Sigma_2 \cup \ldots\ldots \cup \Sigma_n \cup \Sigma_{n+1} \cup \ldots\ldots$

Length of string $|s|$: The length of a string is the number of symbols in the string, with repetitions counted.

e.g., |a| is 1, |ab| is 2, |aba| is 3, |ε| is 0.

String Operations

Concatenation: The concatenation (or product) of the words $u = a_1a_2 \ldots am$ and $v = b_1b_2 \ldots bn$ is the word $uv = a_1a_2 \ldots amb_1b_2 \ldots bn$. $|uv| = |u|+|v|$.

e.g., if Σ = {a, b}, x = aba, y = bbb

then,

xy = ababbb

yx = bbbaba

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

The ***reversal*** (or ***mirror image***) of the word $u = a_1a_2 \ldots a_n$ is $u^{-1} = a_na_{n-1} \ldots a_1$. The reversal of u sometimes is denoted by u^R or $\tilde{u}$. It is clear that $u^{-1-1} = u$ and $(uv)^{-1} = v^{-1}u^{-1}$.

Word ***v*** is a ***prefix*** of the word ***u*** if there exists a word ***z*** such that ***u = vz***. If $z = \varepsilon$ then v is a proper prefix of u.

e.g.,

v = ababbbaa

Prefix of the string v is: {ε, a, ab, aba, abab, ababb, ababbb, ababbba, ababbbaa}

Similarly, v is a ***suffix*** of u if there exists a word x such that $u = xv$. The proper suffix can also be defined.

e.g.,

v = ababbbaa

Suffix of the string v is: {ε, a, aa, baa, bbaa, bbbaa, abbbaa, babbbaa, ababbbaa}

Word ***v*** is a ***subword*** of the word ***u*** if there are words ***p*** and ***q*** such that ***u = pvq***. If $pq \neq \varepsilon$ then v is a ***proper subword***.

e.g.,

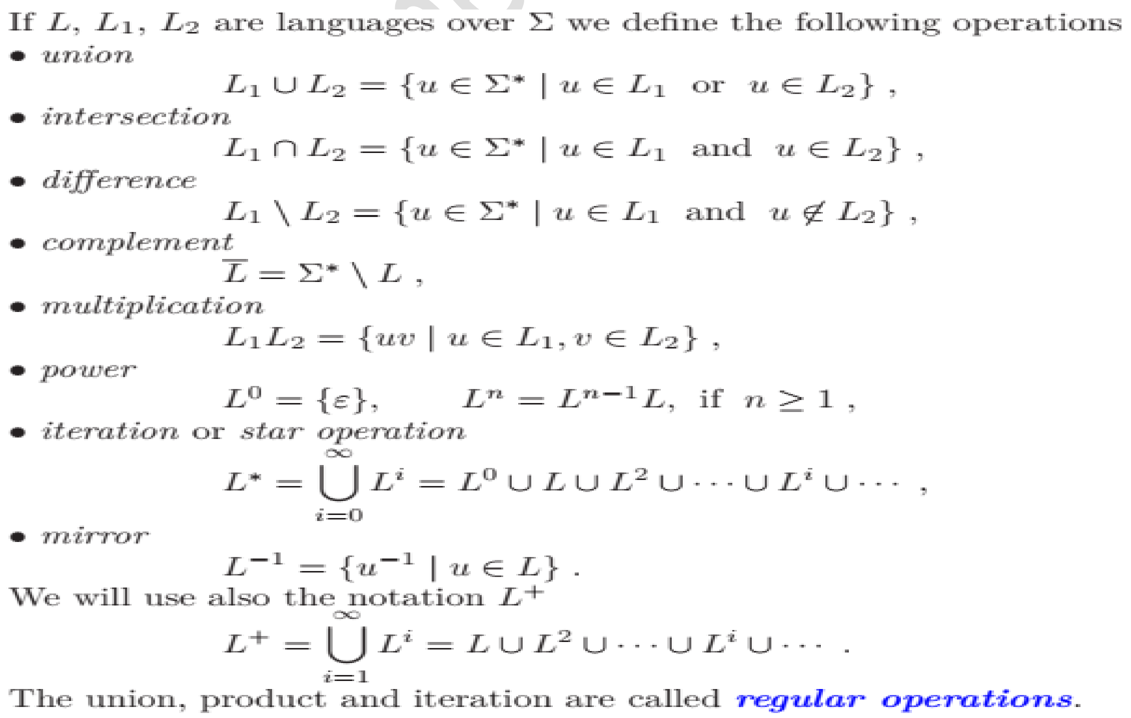
v = ababbbaa

Subwords from the string v are: {babb, bab, abb, ab, baa, ……}

Language: A subset L of Σ^* is called a ***language*** over the alphabet Σ. Sometimes this is called a ***formal language*** because the words are here considered without any meanings. Note that ∅ is the empty language while {ε} is a language which contains the empty word.

اللغة: هي مجموعة السلاسل المختارة من المجموعة *Σ والمولدة من الأبجدية Σ ونرمز لها (L).

Operations on Languages



If L, L_1, L_2 are languages over Σ we define the following operations

- *union*
$$L_1 \cup L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ or } u \in L_2\} ,$$
- *intersection*
$$L_1 \cap L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ and } u \in L_2\} ,$$
- *difference*
$$L_1 \setminus L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ and } u \notin L_2\} ,$$
- *complement*
$$\overline{L} = \Sigma^* \setminus L ,$$
- *multiplication*
$$L_1L_2 = \{uv \mid u \in L_1, v \in L_2\} ,$$
- *power*
$$L^0 = \{\varepsilon\}, \qquad L^n = L^{n-1}L, \text{ if } n \geq 1 ,$$
- *iteration* or *star operation*
$$L^* = \bigcup_{i=0}^{\infty} L^i = L^0 \cup L \cup L^2 \cup \cdots \cup L^i \cup \cdots ,$$
- *mirror*
$$L^{-1} = \{u^{-1} \mid u \in L\} .$$

We will use also the notation L^+
$$L^+ = \bigcup_{i=1}^{\infty} L^i = L \cup L^2 \cup \cdots \cup L^i \cup \cdots .$$
The union, product and iteration are called ***regular operations***.

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

Specifying languages

Languages can be specified in several ways. For example, a language can be specified using: 1) the enumeration of its words; 2) a property, such that all words of the language have this property, but other words have not; 3) Regular Expression; and 4) a grammar.

Specifying languages by enumeration of their words (listing their elements)

For example, the following are languages:

$L_1 = \{\varepsilon, 0, 1\}$,

$L_2 = \{a, aa, aaa, ab, ba, aba\}$.

Even if we cannot enumerate the elements of an infinite set, infinite languages can be specified by enumeration if after enumerating the first some elements we can continue the enumeration using a rule. The following is such a language:

$L_3 = \{\varepsilon, ab, aabb, aaabbb, aaaabbbb, \ldots\ldots\}$.

Specifying languages by properties

The following sets are languages:

$L_4 = \{a^n b^n \mid n = 0, 1, 2, \ldots\}$,

$L_5 = \{u\, u^{-1} \mid u \in \Sigma^*\}$,

$L_6 = \{u \in \{a, b\}^* \mid |u|_a = |u|_b\}$,

e.g., baba, aabb, bbaa,

Where $|u|_a$ denotes the number of letters a in word u, and $|u|_b$ the number of letters b.

Regular Expressions التعابير المنتظمة

A regular expression is a formula, and the corresponding language is a language over Σ. For example, if $\Sigma = \{a, b\}$, then a^*, b^*, a^*+b^* are regular expressions over Σ, which represent, respectively languages $\{a\}^*$, $\{b\}^*$, $\{a\}^* \cup \{b\}^*$.

التعابير المنتظمة هي عبارة عن سلسلة نصّية تُستخدم في وصف العديد من الأنماط الشائعة والتعرف عليها مثل عنوان البريد الالكتروني وعناوين مواقع الانترنت، كما تستخدم في معالجة النصوص وفي صناعة المترجمات (Compilers) حيث قامت ويكيبيديا (Wikipedia) بتصحيح أكثر من 250 ألف خطأ املائي في مقالاتها باستخدام تطبيق يعتمد على التعابير المنتظمة، حيث يكون التعبير المنتظم هام للتعرف على أنماط معينة من السلاسل. وكمثال للتوضيح: لجعل الآلة تتعرف على بريد الكتروني من نص ما، نستخدم التعابير المنتظمة للبريد الالكتروني ولتكن بالشكل:

(charcter)$^+$(Digits, character)*@(charcter)$^+$.(character)$^+$

حيث: $^+$ تعني انه يمكن تكرار الحرف مرة أو أكثر (أي حرف واحد على الأقل).

* تعني أنه يمكن تكرار الحرف صفر مرة أو أكثر (أي ولا حرف أو أكثر).

Definition: *Define recursively a regular expression over Σ and the language it represents:*

- ∅ is a regular expression representing the empty language. e.g., $L(\emptyset) = \emptyset = \{\}$

  ∅ هو التعبير المنتظم الذي يحدد اللغة الفارغة.

- ε is a regular expression representing language {ε}. e.g., $L(\varepsilon) = \{\varepsilon\}$

  ε هو التعبير المنتظم الذي يحدد اللغة التي تحوي السلسلة الفارغة.

- If a ∈ Σ, then a is a regular expression representing language {a}. e.g., $L(a) = \{a\}$

  اذا كان a حرف من حروف الأبجدية Σ عندئذ تكون المجموعة {a} لغة منتظمة على Σ.

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

- If x, y are regular expressions representing languages X and Y, respectively, then (x+y), (xy), (x^*) are regular expressions representing languages X∪Y, XY and X^* respectively.

اذا كان X, Y تعبيران منتظمان فإن التعبير X+Y هو تعبير منتظم يحدد اجتماع اللغتين L(X) و L(Y). والتعبير XY هو تعبير منتظم يحدد تعاقب اللغتين L(X) و L(Y). و X^* هو تعبير منتظم (تكرار E من الصفر الى تكرارات غير منتهية).

Properties of regular expressions

$x+y$	$\equiv$	$y+x$
$(xy)z$	$\equiv$	$x(yz)$
$(x+y)+z$	$\equiv$	$x+(y+z)$
$x(y+z)$	$\equiv$	$xy+xz$
$(x+y)^*$	$\equiv$	$(x^*+y^*)^*$
$(x^*)^*$	$\equiv$	x^*
x^*x	$\equiv$	xx^*
$xx^*+\epsilon$	$\equiv$	x^*

Regular Expression: Examples

Consider Σ = {0,1} and w ϵ Σ, Find RE for each of the following languages:

1- {w| w contains a single 1}
   Answer: 0^*10^* e.g., 0010, 100, 01, 1, 0000100000, …...

2- {w| w has at least single 1}
   Answer: $(0+1)^*1(0+1)^*$ e.g., 0 1 010, 1, 10, 01 1 0, 001, 111111111, …..

3- {w| w contains the string 001 as a substring}.
   Answer: $(0+1)^*001(0+1)^*$ e.g., 100101, 0001, ~~100~~, 1010010, ~~010101~~, …..

4- {w| every 0 in w is followed by at least single 1}
   Answer: $1^*(011^*)^*$ e.g., 1101101111, 111, ε, 01, ~~0010~~, …..

5- {w| w is a string of even length}
   Answer: $((0+1)(0+1))^*$ e.g., 01, 1001, 11, 1100, 0000, 1111, …..

6- {w| the length of w is a multiple of three}
   Answer: $((0+1)(0+1)(0+1))^+$ e.g., ***HW***

7- {w| w starts and ends with the same symbol}.
   Answer: $(0(0+1)^*0)+(1(0+1)^*1)+0+1$ e.g., ***HW***

HW. Does the string dabc ∈ $L(ab^*c+d^*)$ or it ∈ $L(ab^*c+d)^*$?

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.