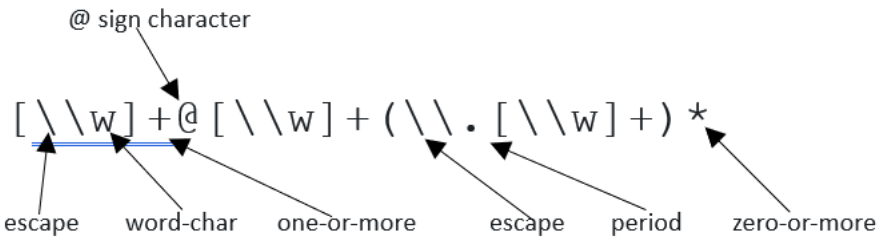
Section 12.1 – Formal Languages and Regular Expressions

Regular Expressions occur frequently, and in many different contexts, in CS, IT, MA, and DS, most often to match patterns.

For example, in CS 360 – Programming Languages, we discuss their use during compilation, when source code is translated into machine executable code. And in CS 240 – Computer Science 2, they're used to identify email id's and URLs in web-pages. In Java, a simple regular expression to capture, many, but not all email id's is:



For more on Regular Expressions, https://beginnersbook.com/2014/08/java-regex-tutorial/.

Here, we will define them mathematically, and then discuss their uses. We begin with some foundational definitions.

A Greek sigma is the traditional symbol used to denote an alphabet in a formal language. It is smaller than the Greek sigma used for a summation.

Alphabet Σ:	a finite set of characters
String over Σ:	(1) a finite juxtaposition of elements (called **characters**) of Σ or (2) the null string λ
Length of a string over Σ:	the number of characters that made up the string, with the null string having length 0
Formal language over Σ:	a set of strings over the alphabet

So, we define a language as a set of strings derived from a given alphabet. For example, here are two languages:

Let the alphabet $\Sigma = \{a, b\}$.

a. Define a language L_1 over Σ to be the set of all strings that begin with the character a and have length at most three characters. Find L_1.

b. A **palindrome** is a string that looks the same if the order of its characters is reversed. For instance, aba and $baab$ are palindromes. Define a language L_2 over Σ to be the set of all palindromes obtained using the characters of Σ. Write ten elements of L_2.

Solution

a. $L_1 = \{a, aa, ab, aaa, aab, aba, abb\}$

b. L_2 contains the following ten strings (among infinitely many others):

$$\lambda, a, b, aa, bb, aaa, bab, abba, babaabab, abaabbbbbaaba$$

Let Σ be an alphabet. For each nonnegative integer n, let

$$\Sigma^n = \text{the set of all strings over } \Sigma \text{ that have length } n,$$
$$\Sigma^+ = \text{the set of all strings over } \Sigma \text{ that have length at least 1, and}$$
$$\Sigma^* = \text{the set of all strings over } \Sigma.$$

These superscripts are important, especially the **+** and *****, which we will use when we create our regular expressions, which is one of the most useful ways to define a language.

Given an alphabet Σ, the following are **regular expressions over Σ:**

I. Base: $\emptyset$, λ, and each individual symbol in Σ are regular expressions over Σ.

II. Recursion: If r and s are regular expressions over Σ, then the following are also regular expressions over Σ:

(i) (rs) (ii) $(r \mid s)$ (iii) (r^*),

where rs denotes the concatenation of r and s, r^* denotes the concatenation of r with itself any finite number (including zero) of times, and $r \mid s$ denotes either one of the strings r or s. The regular expression r^* is called the **Kleene closure** of r.

III. Restriction: Nothing is a regular expression over Σ except for objects defined in (I) and (II) above.

Every regular expression defines a language, which we know is a set of strings.

Partner up

Let $\Sigma = \{0, 1\}$. Use words to describe the languages defined by the following regular expressions over Σ.

a. $0^*1^* \mid 1^*0^*$

b. $0(0 \mid 1)^*$

==========

a. The strings in this language consist either of a string of 0's followed by a string of 1's or of a string of 1's followed by a string of 0's. However, in either case the strings could be empty, which means that λ is also in the language.

b. The strings in this language have to start with a 0. The 0 may be followed by any finite number (including zero) of 0's and 1's in any order. Thus the language is the set of all strings of 0's and 1's that start with a 0.

Partner up

In each of (a) and (b), let $\Sigma = \{a, b\}$ and consider the language L over Σ defined by the given regular expression.

a. The regular expression is $a^*b(a|b)^*$. Write five strings that belong to L.

b. The regular expression is $a^*|(ab)^*$. Indicate which of the following strings belong to L:

$$a \quad b \quad aaaa \quad abba \quad ababab$$

==========

a. The strings b, ab, $abbb$, $abaaa$, and $ababba$ are five strings from the infinitely many in L.

b. The following strings are the only ones listed that belong to L: a, $aaaa$, and $ababab$. The string b does not belong to L because it is neither a string of a's nor a string of possibly repeated ab's. The string $abba$ does not belong to L because any two b's that might occur in a string of L are separated by an a.

Partner up

Let $\Sigma = \{0, 1\}$. Find regular expressions over Σ that define the following languages:

a. The language consisting of all strings of 0's and 1's that have even length and in which the 0's and 1's alternate.

b. The language consisting of all strings of 0's and 1's with an even number of 1's. Such strings are said to have *even parity*.

c. The language consisting of all strings of 0's and 1's that do not contain two consecutive 1's.

==========

a. If a string in the language starts with a 1, the pattern 10 must continue for the length of the string. If it starts with 0, the pattern 01 must continue for the length of the string. Also, the null string satisfies the condition by default. Thus an answer is

$$(10)^*|(01)^*.$$

b. Some basic strings with even parity are λ, 0, and 10^*1. Concatenation of strings with even parity also have even parity. Because such a string may start or end with a string of 0's, one answer is

$$(0|10^*1)^*.$$

c. Note that a string may end in a 1, but any other 1 must be followed immediately by a 0. Thus, it is enough to enforce the rule that a 1 must be followed by a 0, unless the 1 is at the end of the string. A regular expression satisfying these conditions is

$$(0|10)^*(\lambda|1).$$