Definition: $\forall n \in \mathbb{N}^*$, we define a word of length n in the alphabet A as being any string of the form $a_1, a_2, ..., a_n$ $s.t.$ $a_i \in A \quad \forall i, 1 \leq i \leq n$. Let A^n be the set of all words of length n over the alphabet A.

Remark: There is a one-to-one correspondence between the string $a_1 a_2 ... a_n$ and the ordered n-tuple $(a_1, a_2, ..., a_n) \in A^n = \underbrace{A \times ... \times A}_{n\ times}$, the Cartesian product of n copies of A.

Definition: Let $A^+ = \bigcup_{n=1}^{\infty} A^n = A^1 \cup A^2 \cup A^3 \cup ....$ A^+ is the set of all words of positive length over the alphabet A.

Examples:

1. $A = \{0, 1\}$, A^+ is the set of all binary strings of finite length that is at least one, **i.e.** $0, 1, 01, 10, 00, 11$, etc.
2. If $A =$ letters of the English alphabet, then A^+ consists of all non-empty strings of finite length of letters from the English alphabet.

It is useful to also have the empty word ε in our set of strings. ε has length 0. Define $A^0 = \{\varepsilon\}$ and then adjoin the empty word ε to A^+. We get $A^* = \{\varepsilon\} \cup A^+ = A^0 \cup \bigcup_{n=1}^{\infty} A^n = \bigcup_{n=0}^{\infty} A^n$.

Notation: We denote the length of a word w by $\mid w \mid$.

Next introduce an operation on A^*.

Definition: Let A be a finite set, and let w_1 and w_2 be words in A^*. $w_1 = a_1 a_2 ... a_m$ and $w_2 = b_1 b_2 ... b_n$. The concatenation of w_1 and w_2 is the word $w_1 \circ w_2$, where $w_1 \circ w_2 = a_1 a_2 ... a_m b_1 b_2 ... b_n$. Sometimes $w_1 \circ w_2$ is denoted as just $w_1 w_2$. Note that $\mid w_1 \circ w_2 \mid = \mid w_1 \mid + \mid w_2 \mid$.
Concatenation of words is:

1. associative
2. NOT commutative if A has more than one element.

Proof of (1): Let $w_1, w_2, w_3 \in A^*$. $w_1 = a_1 a_2 ... a_m$ for some $m \in \mathbb{N}$, $w_2 = b_1 b_2 ... b_n$ for some $m \in \mathbb{N}$, and $w_3 = c_1 c_2 ... c_p$ for some $p \in \mathbb{N}$. $(w_1 \circ w_2) \circ w_3 = w_1 \circ (w_2 \circ w_3) = a_1 a_2 ... a_m b_1 b_2 ... b_n c_1 c_2 ... c_p$.

qed

Proof of (2): Since A has at least two elements, $\exists a, b \in A$ s.t. $a \neq b$.

$a \circ b = ab \neq ba = b \circ a.$

qed

A^* is closed under the operation of concatenation $\Rightarrow$ concatenation is a binary operation on A^* as $\forall w_1, w_2 \in A^*$, $w_1 \circ w_2 \in A^*$.

Theorem Let A be a finite set. $(A^*, \circ)$ is a monoid with identity element ε.

Proof: Concatenation $\circ$ is an associative binary operation on A^* as we showed above. Moreover, $\forall w \in A^*, \varepsilon \circ w = w \circ \varepsilon = w$, so ε is the identity element of A^*.

qed

Definition: Let A be a finite set. A <u>language</u> over A is a subset of A^*. A language L over A is called a <u>formal language</u> is $\exists$ a finite set of rules or algorithm that generates exactly L, **i.e.** all words that belong to L and no other words.

Theorem: Let A be a finite set.

1. If L_1 and L_2 are languages over $A, L_1 \cup L_2$ is a language over A.
2. If L_1 and L_2 are languages over $A, L_1 \cap L_2$ is a language over A.
3. If L_1 and L_2 are languages over A, the concatenation of L_1 and L_2 given by $L_1 \circ L_2 = \{w_1 \circ w_2 \in A^* \mid w_1 \in L_1 \wedge w_2 \in L_2\}$ is a language over A.
4. Let L be a language over A. Define $L^1 = L$ and inductively for any $n \geq 1$, $L^n = L \circ L^{n-1}$. L^n is a language over A. Furthermore, $L^* = \{\varepsilon\} \cup L^1 \cup L^2 \cup L^3 \cup ... = \bigcup\limits_{n=0}^{\infty} L^n$ is a language over A.

Proof: By definition, a language over A is a subset of A^*. Therefore, if $L_1 \subseteq A^*$ and $L_2 \subseteq A^*$, then $L_1 \cup L_2 \subseteq A^*$ and $L_1 \cap L_2 \subseteq A^*$. $\forall w_1 \circ w_2 \in L_1 \circ L_2$, $w_1 \circ w_2 \in A^*$ because $w_1 \in A^n$ for some n and $w_2 \in A^m$ for some m, so $w_1 \circ w_2 \in A^{m+n} \subseteq A^* = \bigcup\limits_{n=0}^{\infty} A^n$.

Applying the same reasoning inductively, we see that $L \subset A^* \Rightarrow L^* \subseteq A^*$ as $L^n \subseteq A^* \, \forall n \geq 0$.

qed

Remark: This theorem gives us a theoretic way of building languages, but we need a practical way. The practical way of building a language is through the notion of a grammar.

Definition: A (formal) grammar is a set of production rules for strings in a language.
To generate a language we use:

1. the set A, which is the alphabet of the language;
2. a start symbol <s>;
3. a set of production rules.

Example: $A = \{0, 1\}$; start symbol <s>; 2 production rules given by:

1. <s> $\to$ 0<s>1
2. <s> $\to$ 01

Let's see what we generate: via rule 2, <s> $\to$ 01, so we get <s> $\Rightarrow$ 01 Via rule 1, <s> $\to$ 0<s>1, then via rule 2, 0<s>1 $\to$ 0011. We write the process as <s> $\Rightarrow$ 0<s>1 $\Rightarrow$ 0011.