Definition 1 (Concatenation) *Consider α, and β words over the A alphabet, namely words constructed from symbols of the alphabet. The result of $\alpha\beta$ is the concatenation of the two words, so that $\gamma = \alpha\beta$, where $|\gamma| = |\alpha| + |\beta|$, so the length of the new word is the sum of the length of the two components.*

Definition 3 (Power of Words)

$$\alpha^0 = \varepsilon$$
$$\alpha^n = \alpha^{n-1}\alpha$$

Then, if $n \geq 1$, namely the nth power of word α is the n times concatenation of the word.

Definition 4 (Reversal of Words) *In case of word $\alpha = a_1,a_2,\ldots,a_m$ word $\alpha^T = a_m, a_{m-1},\ldots,a_1$ is the reversal of α. If $\alpha^T = \alpha$, the word is a palindrome.*

Definition 5 (Subword) *Word β is subword of word α if there are words γ, and δ in a way that $\alpha = \gamma\beta\delta$, and $\gamma\delta \neq \varepsilon$, namely if β is a real subword of α.*

Definition 6 (Subwords with Various Length) *Denote the set of k length subwords of word α $R_k(\alpha)$. $R(\alpha)$ is the set of all such subwords so*

$$R(\alpha) = \bigcup_{k=1}^{|\alpha|} R_k(\alpha)$$

Definition 7 (Complexity of Words) *The complexity of a word is the number of its subwords of different length. The number of k length subwords of word α is $r_\alpha(k)$.*

Definition 8 (Maximal Complexity) *Maximal complexity can only be interpreted on finite words and*

$$\text{Max}(\alpha) = max\{r_\alpha(k)|k \neq 1\}, \quad \alpha \in A^*,$$

where $A*$ is the Kleene star derived from the particular alphabet. (On infinite words we can interpret bottom or top maximal complexity.)

Definition 9 (Global Maximal Complexity) *Global maximal complexity is the sum of the number of nonempty subwords of a word, namely*

$$Tb(\alpha) = \sum_{i=1}^{|\alpha|} r_\alpha(i), \quad \alpha \in A^*,$$

Definition 10 (Mc Cabe's cyclomatic number) *The cyclomatic number $V(G)$ of control graph $G = (v, e)$ is $V(G) = e-v+2p$, where p denotes the number of graph components, which is the same as the number of linearly coherent cycles in a highly coherent graph.*

Definition 11 (Infinite Words) *Consider A_w to denote the set of right infinite words, and the set of finite and infinite words over the A alphabet is denoted:*

$$A_{all} = A^* \cup A_w.$$

Definition 17 (Kleene star of V) *$V^0 := \{\varepsilon\}$, and $V^n := V \otimes V^{n-1}$, $n \geq 1$, and the*

$$V^* := \bigcup_{i=0}^{\infty} V^i = V^0 \cup V^1 \cup V^2 \cup \ldots$$

set is called the Kleene star of "V".

Definition 18 (positive closure of V) *The*

$$V+ := \bigcup_{i=1}^{\infty} V^i = V^1 \cup V^2 \cup \dots$$

set is the positive closure of "V". (Namely $V^* = V+ \cup \varepsilon$.)

Definition 19 (formal language) *Consider V*1 « A. We call an $L \subseteq V^*$ set a formal language over alphabet "A".*

Definition 20 (contextual multiplication of languages) *If L_1,L_2 are two formal languages, then $L_1 * L_2 := \{\alpha\beta \mid \alpha \in L_1 \text{ and } \beta \in L_2\}$. This operation is called contextual multiplication of languages.*

Definition 21 (Generative Grammars) A *G(V, W, S, P) formal quadruple is called a generative grammar, where:*

- *V : is the alphabet of terminal symbols,*
- *W : is the alphabet of nonterminal symbols,*
- *$V \cap W =$, the two sets are disjunct, thy have no shared element,*
- *S : $S \in W$ is a special nonterminal symbol, the start symbol,*
- *P : is the set of replacement rules, if $A := V \cup W$, then $P \subseteq A^* x A^*$.*

Definition 22 (Rule of Derivability) *Consider G(V, W, S, P) generative grammar and $X, Y \in (V \cup W)^*$. From X Y is derivable if $X \Rightarrow +Y$ or $X = Y$. It is denoted: $X \Rightarrow {}^*Y$.*

Definition 23 (Derivable in One Step) *Consider G(V, W, S, P) generative grammar and $X, Y \in (V \cup W)^*$. $X = \alpha\gamma\beta$, $Y = \alpha\omega\beta$ form words where $\alpha, \beta, \gamma, \omega \in (V \cup W)^*$. We say that Y can be derived in one step, if there is ak $\gamma \to \omega \in P$. It is denoted: $X \to Y$.*

Definition 24 (Derivable in Multiple Steps) *Consider G(V, W, S, P) generative grammar and $X, Y \in (V \cup W)^*$. Y can be derived from X in multiple steps if ($n \geq 1$), there is $\exists X_1, X_2, \dots, X_n \in (V \cup W)^*$, so that $X \to X_1 \to X_2 \to X_3 \to \dots \to X_n = Y$. It is denoted: $X \to +Y$.*

Definition 25 (Phrase-structure) *Consider G(V, W, S, P) generative grammar. G generates a $X \in (V \cup W)*$ word if $S \Rightarrow {}^*X$. Then X is called Phrase-structure.*

Definition 26 (Phrase) *Consider G(V, W, S, P) generative grammar. If G generates an X word and $X \in V^*$ (it does not contain nonterminal symbols), then X is called a phrase.*

Definition 27 (Generated Language) *Consider G(V, W, S, P) generative grammar. The $L(G) = \{\alpha | \alpha \in V^*\}$ and $S \to \alpha$ language is called a generated language by grammar G.*

Definition 28 (Rule of Equivalence) *A $G_1(V, W_1, S_1, P_1)$ generative grammar is equivalent to a $G_2(V, W_2, S_2, P_2)$ generative grammar if $L(G_1) = L(G_2)$, namely if the language generated by G_1 is the same as the language generated by G_2.*

Definition 29 (Phrase-structure Languages) *An arbitrary language is type-0 or phrase-structure if every rule generating the language is in the $\alpha A\beta \to \gamma$ form.*

Definition 30 (Context Sensitive Languages) *An arbitrary language is type-1 or context sensitive if every rule of the grammar generating the language is in the $\alpha A\beta \to \alpha\omega\beta$ form, and the use of rule $S \to \varepsilon$ is allowed.*

Definition 31 (Context Free Languages) *An arbitrary language is type2 or context free if every rule of the grammar generating the language is in the $A \to \omega$ form and rule $S \to \varepsilon$ is allowed*

Definition 32 (Regular Languages) *An arbitrary language is type-3 or regular if every rule of the grammar generating the language is in the $A \to a$ and $A \to Ba$, or $A \to a$ and $A \to aB$ form. In the first case we call it right regular language and in the second one we call it left regular language.*

Definition 33 ((Chomsky) Normal Form) *A context free grammar is in Chomsky normal form if its every rule is in form $A \to a$ or $A \to BC$ where $A, B, C \in W$ and $a \in V$.*

Definition 36 (Finite Automaton) *A $G(K, V, \delta, q_0, F)$ formal quintuple is called a finite automaton where:*

- *K: is the finite set of states,*
- *V: is the input alphabet, namely the symbols that can be written on the input tape, which are the terminal symbols of the grammar implemented in the automaton*
- *δ : is the state transition function, $\delta \subseteq KxV \to K$*
- *$q_0 \in K$: is a special state, the initial state*
- *$F \subseteq K$: is the set of terminal, (accepting) states*

Definition 37 (Accepting) *A finite $A(K, V, \delta, q_0, F)$ automaton accepts an L language if it halts in case of any $\alpha \in L$ word accepting it but refuses any $\beta \in L$ word.*

Definition 38 (The Accepted Word) *Consider language L, recognized by the automaton, if it consists of words that are accepted by a $A(K, V, \delta, q_0, F)$ finite automaton.*

Definition 39 (Equivalence of Automata) *$A(K, V, \delta, q_0, F)$ finite automaton is equivalent to $A'(K', V', \delta', q_0', F')$ if the two automata recognize the same language.*

Definition 40 (Equivalence of Automata) *It is always possible to construct a $A'(K', V', \delta', q_0', F')$ complete finite automaton to a $A(K, V, \delta, q_0, F)$ partial finite automaton so that they are equivalent.*

Definition 42 (Configuration of Automata) *The configuration of a $A(K, V, \alpha, q_0, F)$ finite automaton is a (α, q) formal couplet where α is the unread part of the word on the input tape and q is the actual state.*

Definition 43 (Definition of Accepting in Finite Automata) *A A(K, V, δ, q_0, F) finite automaton accepts a ω input word if there is a sequence of configurations in which with finite iteration of δ mapping the initial configuration of the automaton (δ, q0) transits into the terminal configuration (ε, q_0) and $q_0 \in F$. Otherwise the automaton rejects the input word.*

Definition 45 (Computational Capacity) *A set of automata of the same type is called abstract machine class and its computational capacity is the set of formal languages which are recognized by an automaton of the machine class.*

Definition 53 (Turing Machine) *A A(K, V, W, δ, q_0, B, F) formal septuple is called a Turing automaton where*

- *K: is the finite set of states,*
- *V: is the input alphabet, namely the alphabet of the language to be recognized,*
- *W: is the output alphabet, namely the alphabet of symbols that can be written on the tape, where $V \subseteq W$,*
- *δ : is the state transition function, $\delta \subseteq KxW \rightarrow KxW\, x\{\leftarrow,\rightarrow\}$,*
- *q_0: is the initial state of the automaton, $q0 \in K$,*
- *B: is the blank symbol, $B \in W$,*
- *F: is the finite set of accepting states $F \subseteq K$*