

# Test 1 Study Guide

- Moore's Law - Postulates doubling computer hardware every 18 months
- Babbage Difference Engine - Early mechanical computer
- ENIAC - Early tube-based computer
- Program - A syntactically and semantically correct string from a language
- Language - denoted by (L) is a mechanism that permits the expressions of general ideas
- Grammar - Consists of  $G(V_t, V_n, S, P)$ , can be used in a generative or analytic manner
- Alphabet - The list of symbols that can be used to form words and sentences
- Generative - grammar is used to create a string terminal symbols using P
- Analytic - together with specification of G
- $G(V_t, V_n, S, P)$  - ( **$V_t$**  = set of terminal or primitive symbols,  **$V_n$**  = set of nonterminal symbols or variables,  **$P$**  = set of production rules,  **$S$**  = Starting root / Symbol)
- $L(G)$  - language generated by grammar G, denoted  $L(g)$  is the set of all strings that satisfy these conditions
- $V^*$  - closure set of V and  $V^+$
- T0 language - All grammars, All Turing computable languages
- T1 language - Context Sensitive Grammars (CSG)
- T2 language - Context Free Grammars (CFG), push-down automata
- T3 language - Regular Grammars, finite-state automata
- context-sensitive language - a language that can be defined by a context-sensitive grammar
- context-free language - a language that can be defined by a CFG
- regular language - language that can be defined as
- finite state automata - Regular languages that can be recognized by fsm's, includes a finite set of states, a set of transitions between states, an input, and output
- push-down automata - follows finite state automata, but includes a stack. Each cycle, machine can push/pop something off the stack
- regular expression - shorthand for abbreviating a pattern or set of patterns
- Atom - A symbol from the alphabet is a RE: a (represents matching the symbol)
- Concatenation - two RE adjacent to one another is a RE: ab (represents one RE followed by the other RE)
- Alternation - two RE separated by a vertical line is a RE: a|b (represents selection of one RE or the other RE)
- Kleene star - RE followed by an asterisk is a RE:  $a^*$  (represents zero or more copies of the RE)
- Parenthesization - RE in parenthesis is a RE: (a) (represents grouping and scope for other operators)
- Lexical** - defines the syntax
- Syntax - Defines how tokens are combined into sentences
- Semantics - Defines the logical meaning of sentences
- Parser - To see if the string of tokens is derivable according to the syntax of the language
- Scanner - To see if we can recognize all terminals in the string (Program)
- top-down parse - Two approaches ( 1.) **TDFB top-down with full backup**: This is a depth-first expansion of nonterminals starting with the leftmost nonterminal in the expansion of goal, 2.) **Recursive Descent (RD)**: allows no backup:uses binary-values and recursive functions to recognize substrings that correspond to the expansion of a specific nonterminal.
- bottom-up parse - *when creating the interior of a parse tree of productions that link S to x, if we fill the interior of the tree from the terminal leaves of the tree to the root S*
- Programming language - a notational system intended to facilitate human-machine interaction
- imperative language - C/C++,Java,Fortran
- declarative language - Prolog
- functional language - ML,CAML,LISP,HOPE
- rule-based language - OPS5, clips and soar
- event-driven language - "virtually all object-oriented programming languages are event-driven"
- parallel language - MPI/OpenMP
- BNF - Backus Naur Form - a language for writing programming language grammars
- CFL - Context Free Language - Requires a nonterminal on the left, than any amount of nonterminals ending with a terminal
- CNF - Chomsky Normal Form - *is a form for productions in which each production is written as  $A \rightarrow BC$  or  $A \rightarrow a$*
- CYK - Cocke-Younger-Kasami - is a parsing approach that will parse string x in a number of steps to  $|x|^3$  This approach requires the CFG to be in Chomsky Normal Form
- dynamic programming - *solves a problem by breaking it into subproblems and simplifying many of the sub problems so that they can all be solved at once. This allows complex problems to be solved quickly.*
- Token - Tokens or lexical units which define the basic "words" or symbols of the language
- Terminal - *the elementary symbols of the language defined by a formal grammar, that replace groups of syntactic variables according to production rules*
- Non-terminal - *symbols aka syntactic variables that are replaced by groups of their counterpart according to the production rules*
- Identifier - *the name supplied for a variable, type, function, or label in a program*
- reserved word - *a word in a programming language that is a reserved for use as part of the language and may not be used as an identifier*
- Expression - *any valid unit of code that resolves to a value*
- Conditional - *a programming language command for handling decisions often using (if/else)*
- Production - *a rewrite rule of a grammar specifying a symbol substitution can be recursively performed to generate new symbol sequences*

## Symbols

- ::= denotes "is defined as" or "maybe replaced by"
- | denotes "or"
- { } denotes "items repeated zero or more times"
- Letter denotes "non-terminal"
- A denotes "Terminal"
- $x^*$  denotes "zero or more x"