Notes 1-31

Chapter 3: Syntax

1. Fortran- imperative language

* Algorithm with specific sequence of instructions

1. ALGOL- first language with a formal description 🡪 too formal and no one could understand it

Two things you want when trying to learn a language (programing language components):

1. Syntax- rules of language; how we write a sentence in the language
2. Semantics- the meaning of the sentences

* All language are just a set of strings of characters from an alphabet
  + Strings are just sentences which hold syntax?

Vocab:

* Lexeme- smallest recognizable unit of a language (smallest syntactical unit)
  + Ex: 1 in variable = 1
* Token- syntactic category that forms a class of lexemes
  + Ex: if (a >=b) {

b = b + 93;

a = a – 3;

}

* + - Lexemes: if, (, a, >=, b, …
    - Tokens: keyword, left parenthesis, identifier, comparison, identifier, …
* A language can be formally defined in 2 ways:

1. Language generator- used to generate sentences in the language
2. Language recognizer: recognize if a sentence is legal or not
   * Language L with an alphabet
   * Construct a mechanism R (recognition device) to determine if a sentence S belongs to L.
   * Accept or reject S is its job
   * Ex: Finite State Automation

Language Generator:

* Will always produce legal sentence
* Ex: regular expression

Given a (a\* U b\*) b 🡪 1. Output an a; 2. Either output 0 or more a’s or 0 or more b’s; 3. Output b