**CS332 Module 1**

* What is a language?
  + A set of *strings.*
    - Strings means a set of symbols.
* Formal System:
  + Small # of things (symbols)
  + Small # of operators (concatenation, set ups – union, intersection, not)
  + Complexity through iteration (repetition).
* Symbol: a single distinguishable mark.
* Alphabet: a finite set of *symbols*.
* String: a sequence of symbols (finite, from same set (Σ) ).
* Language: a *set* of strings.
* **An alphabet, Σ**, is a finite set of symbols.
  + **|Σ| represents the size of alphabet Σ**.
  + If Σ = {a, b, c}, then |Σ| = 3.
* Terminals are the symbols in the alphabet (i.e. {a, b}.
  + Non-terminals can be represented as a capital letter. They are like placeholders in the grammar????? What is the point/meaning of the non-terminals/capital letters?
* A string, u, is a sequence of symbols from some Σ.
  + Typically use u, v, and w.
  + **λ** represents **the empty string***: a string of length zero having no symbols*.
  + The length of string u, |u|, is the number of symbols that make it up.
  + If u = abbca, then |u| = 5; |λ| = 0. • Σ\* represents all possible strings that can be made from Σ.
* A **language, L**, is a (potentially infinite) set of strings.
  + Languages have no inherent meaning.
  + Languages are not about communication. Some are used for communication.
  + Languages are not about programming. Some are used for programming.
* **BNF** (Backus-Naur Form): ?????
  + BNF uses *production rules* composed of a left-hand side (LHS), and a right-hand side (RHS), each containing symbols.
  + The symbols on left hand side (LHS) can be replaced by those on the right-hand side (RHS).
  + *Production rules* take the form LHS 🡪 RHS (some authors use LHS := RHS).
  + **Example**: If you have a rule that says A 🡪 aCa, and the string abAca, you can use the rule to replace the “A” in the string with “aCa” – result is abaCaca (underlined for visibility)

Text

Description automatically generated

* The rules to construct a language are called its “**Grammar**”, **G**.
  + For example, if we say the language is composed of strings with every a coming in groups of three.
  + We’re assuming there is a pattern or structure to the language but a set of random strings is still a language thus grammar only applies to non-random languages.
  + **The grammar has to be able to derive ALL and ONLY strings in the language**.
* We use **S for the “start symbol”** which all grammars must start with… WHAT IS S???
  + EX: Here is how to define G for a language. Assuming Σ = {a, b}. ???EXPLAIN IN COMMON LANGUAGE THIS LANGUAGE’S GRAMMAR???
    - S 🡪 AB
    - A 🡪 aA
    - A 🡪 a
    - B 🡪 bB
    - B 🡪 b
* The language and alphabet are different things. Alphabet = symbols (terminals) and the language is the possible strings made using the alphabet and following a grammar (assuming it has a grammar).
* Table

  Description automatically generatedGrammar and language talk about the same thing but grammar is like the rules and language is the stuff that the grammar describes.

Text

Description automatically generatedText

Description automatically generatedTable

Description automatically generated