Jackson Warren

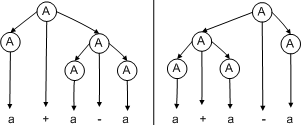
CSCI 330

Prof. Seif

10/1/21

Homework #1 – CSCI 330

Context Free Grammars (CFGs) define a formal language, where a formal language is defined as able to work under a defined and strict set of rules. CFGs are designed for context free languages using recursive rules to generate patterned strings. To create a CFG, one must: have a starting symbol, apply production rules to the left-hand side while replacing this with the right-hand side of the production, and repeating this process of selecting nonterminal symbols in the string and replacing them with that right-hand side till no nonterminal values remain. This can also be modeled using a parse tree. Each node in the tree represents a symbol, while the edges are the applications of said production rules. The end nodes, or leaves, are the resulting possible ways that this result could have formed. It is important to note, many CFGs are not a single route to the expected result. The parse tree shows this visually, representing the possible combinations that could occur to create the results with terminal symbols. For example, CFGs can create the string “a + a – a” in two different ways, as shown below (Fig 1.0).



(Fig 1.0)

CFGs are compatible with almost all languages, only excluding the languages that do not use regular expressions in their syntax.

References:

A. (2020, January 15). *What are Context Free Grammars?* FreeCodeCamp.Org. https://www.freecodecamp.org/news/context-free-grammar/

*Context Free Grammars | Brilliant Math & Science Wiki*. (n.d.). Context Free Grammars. Retrieved September 28, 2021, from https://brilliant.org/wiki/context-free-grammars/#relationship-with-other-computation-models