Pen and Paper

1. Write a grammar for the Doty simulator CFG syntax.

$S \to V(\to)K$
$K- > U;|U;S$
$U- > U(|)U|V|T|\varepsilon|UU$
$T- >< anystring >$
$V- >< anysymbol >$
$|,; \neq w_i|w_0...w_i \in anystring$
$|,; \notin anysymbol$

Code

Create a command-line program that takes two positional arguments. The first argument should be a path to a text file containing a grammar in Doty CFG syntax. The second argument should be a path to a text file containing a string.

The program need not produce any output apart from a return code. If the provided grammar can parse the input string, the program should return 0. If the provided grammar cannot parse the input string, the program should return 1.

Use one of the following languages: C++, Python, C, Elixir, Elm, Go, Haskell, Java, OCaml, Racket, Scheme, Rust, or Typescript.

2. Write a parser for the Doty CFG language. You may use any technique for parsing this language.

3. Implement the algorithm in Theorem 7.16 to determine if an input string can be generated by the input grammar. You may not use a parsing library. You may assume that the input grammar is in Chomsky Normal Form.