Project Proposal

Random Sentence Generation from a CFG

Bryce Manley

Contents

Description

This Program will take a CFG that includes words and non-terminal symbol (N.T.S). The program will take this CFG and create a pseudorandom sentence based on this CFG.

Expected Inputs

The program will take a txt file that will represent a CFG and will include the words that could be used in the random sentence generation.

Examples of the precise input file layout can be seen in the test cases section, but the rules are as follows:

Take for example the CFG
A-> {the, a} N
N->{policeman , firefighter , paramedic}

1. Begin with the “name” of the N.T.S in this case “A” followed by ->
2. Include all words in brackets separated by commas
3. The N.T.S symbol should be separated by commas
4. Only one rule per line

Expected Outputs

Based on input parameters the program will either print a random generated sentence to terminal or to a txt file.

Intended Programming Language

This Project will be completed in Java. As there are no libraries that I will need to import; I am free to pick the language I am most comfortable with which for me is Java.

Program Design

The user will execute the program with two arguments:

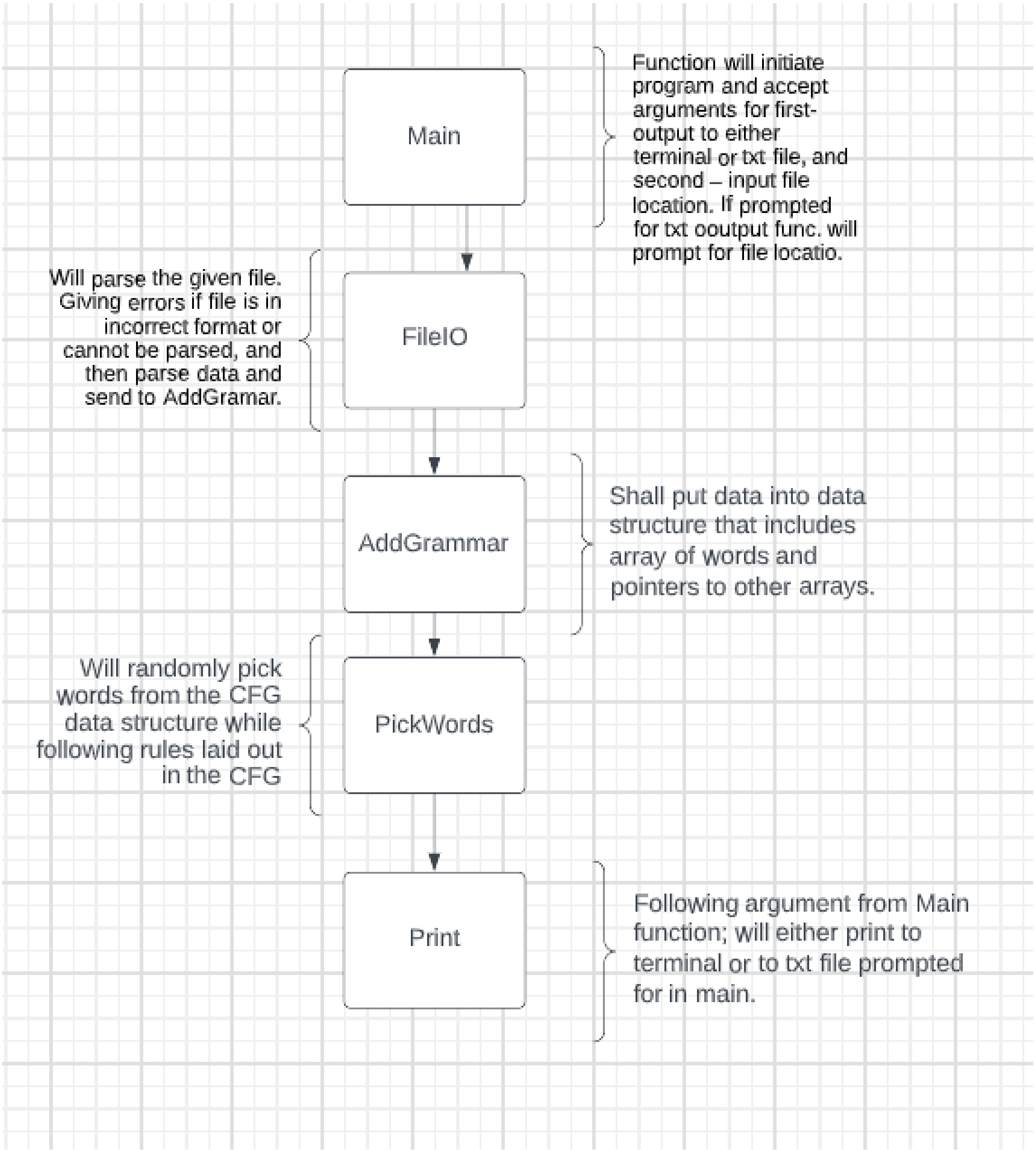
1. the expected output -either t for terminal output or x for txt file output.
2. The input txt file that will be formatted similar to the example bellow.

Take for example the CFG
A-> {the}a, N
N->{policeman , firefighter , paramedic}

The program will create an object that includes "the" and "a" as strings and will include a pointer to the N object . The N object will include the strings policeman firefighter and paramedic.

When executed the program will iterate through the data structure and randomly pick "the" or "a" then follow the pointer to the N array and randomly pick policeman firefighter or paramedic.

Figure 1: Program Execution Block Diagram



Function will initiate program and accept arguments for first-output to either terminal or txt file, and second – input file location. If prompted for txt ooutput func. will prompt for file locatio.

Will parse the given file. Giving errors if file is in incorrect format or cannot be parsed, and then parse data and send to AddGramar.

Shall put data into data structure that includes array of words and pointers to other arrays.

Will randomly pick words from the CFG data structure while following rules laid out in the CFG

Following argument from Main function; will either print to terminal or to txt file prompted for in main.

Test Cases

These Test cases will show the expected input and a possible output of the given CFG. The first two cases are more of edge cases as the most efficient way to build the grammar is to have all the N.T.S first and words after that.

Test Case 1:

Input:
A-> {the ,a}N
N->{policeman , firefighter , paramedic} V
V-> {ran, sprinted, helped}

Output:
The firefighter helped

Test Case 2:
Input:
S-> A,N,V,A,N
A-> {the ,a}
N->{policeman, firefighter, paramedic, guy, person}
V-> {assisted, helped}
Output
The guy assisted the firefighter

Test Case 3:

Input:

S-> A,BV,A,V
A-> {I, he, the Pope}
N->{policeman, firefighter, paramedic, guy, person}
BV-> {is, was}
V->{nice, kind, hateful, mean}
Output:
The pope is mean.

Test Case 4:
S->{}N, pilot
Output:
Error: Rules are called that do not exist

Test Case 5:
Input:
S- A,BV,A,V
A- {I, he, the Pope}
N->{policeman, firefighter, paramedic, guy, person}
BV-> {is, was}
V->{nice, kind, hateful, mean}

Output:
Error: Lines in CFG do not have “->”

Foreseen Issues I want to take note of:

I can’t have a pointer to a grammar that doesn’t exist yet

Potential solutions:

Include separate array in the same rule object that includes all the grammars that need to be pointed to. After we have read everything then go through the array and make sure everything exists and if it does point to it. I think this would actually be a good use for a HashMap by using the rules as Keys?? Would love some feedback on that idea.

Need to keep an eye out for infinite loops in the case where CFG’s are calling each other. I should throw an error in that case

Pointers don’t exist in java:

use references - not an actual issue.