

Spring 2016
CSCI 2670 Theory of Computing
Programming Assignment 3: Context-Free
Languages

General Information

Deadline: 11:00am Tuesday March 26

Worth: 50 pts

The Assignment

The purpose of this assignment is to introduce you to implementing string generation using context-free grammars. Your grade will be based on the correctness of the program and how well you have documented your program. Please utilize good programming practices and write code in Java that is efficient and well commented.

Note: This assignment is not a group project and everybody should work on it individually.

Problems

1. (30 points) Program a random sentence generator that reads a context-free grammar from an input file and prints out one or more random sentences. It should take as its first argument, a path to the file containing the grammar. If a second numeric argument is present, it should generate as many sentences as the number in the second argument indicates. You should name your program as “**randCFL**”.

Use the input file provided with this assignment labeled “*grammar*” to test your generator. You should get sentences (some that may not make any

sense) like,

Jim ate cheese.
big Jim ate green cheese.
big cheese ate Jim.
green Jim ate green big Jim.

The format of the example *grammar* file is as follows:

```
P      N  
P      A P  
START P V P.  
A      big  
A      green  
N      cheese  
N      Jim  
V      ate
```

This corresponds to the following context-free grammar, (V, Σ, R, S) where

$V = \{S, A, P, N, V\}$

$\Sigma = \{Jim, big, green, cheese, ate\}$

```
 $R = \{$   $P \rightarrow N$   
         $P \rightarrow A P$   
         $S \rightarrow P V P.$   
         $A \rightarrow big$   
         $A \rightarrow green$   
         $N \rightarrow cheese$   
         $N \rightarrow Jim$   
         $V \rightarrow ate\}$ 
```

Note that a line in the input file will consist of a non-terminal symbol followed by a space or a tab, and then a sequence of any number of non-terminal and terminal symbols. The grammar's start symbol is *START*.

2. (20 points) Give your program an option “-t” that makes it produce elaborate derivations instead of just strings. Please use the left-most derivation to produce the string. In other words, when your program is invoked as,

randCFL -t grammar 1

instead of simply printing out a sentence say, big cheese ate Jim, it should

print the following derivation:

```
START => P V P. => A P V P. => big P V P.  
=> big N V P. => big cheese V P.  
=> big cheese ate P. => big cheese ate Jim.
```

Notice that this includes extra information on how the string was generated. If you think that the grammar is ambiguous and multiple leftmost derivations may exist, just pick one and show it.

What and how to hand it in

You'll submit the *typed* code in Java. Please be sure to include all files needed to compile the program.

With the program, you should also submit a README file that clearly explains how to compile the program, execute it, and shows some example test runs including the random sentences generated by your program for the given *grammar* file. Please include your name and student ID in the README file. Programs without a README file will not be graded.

You should ensure that your program compiles and runs correctly on *Nike*.

Please submit your code and README file as a single *zipped* file via eLC by the deadline. Assignments that are **late** but within a day of the deadline will be penalized 33% of the total number of points. Assignments submitted later than one day will not be accepted.