ICSI 409/509: Group HW

Due 2/12/24.

There are 3 parts to this HW.

1. 1 member will produce the grammars from the given language and convert it to the CNF format on paper: Members not comfortable with coding can attempt this section of the homework. 40%. – This Question is 40%, Question 2 is 40% Question 3 is 20%

2. 2|3 members will work on the implementation and preliminary testing. 60%. --This question is 60%, Question 1 is 20% Question 3 is 20%.

3. 2|3 members will work on the testing and documentation (your report and readme file) – (This must still be done and if a member is doesn't participate, you still provide documentation, but they get no grade for the HW). 50%. - This question is 50%, Question 2 is 30% and Question 1 is 20%

Assignment Description:

In this assignment, you will implement the CYK algorithm to determine whether a given string is in a CFG in CNF. The CYK algorithm is an algorithm that can efficiently decide whether a string can be generated by a CFG in CNF. The algorithm works by using a dynamic programming approach, filling in a matrix to determine if each substring of the input string can be generated by the CFG.

You will be provided with a set of grammars in CNF in the form of a set of productions. Your task is to implement the CYK algorithm to determine whether a given string can be generated by each of these grammars.

Assignment Requirements:

1. You should write a program that accepts a string and a set of productions, and then implements the CYK algorithm to determine whether the given string can be generated by the grammar defined by the productions.
2. Your program should be able to handle grammars in CNF. A grammar is in CNF if every production is either of the form A -> BC or A -> a, where A, B, and C are non-terminal symbols, and a is a terminal symbol.

Assignment Overview:

The CYK algorithm is a classical algorithm in computational linguistics and automata theory that is used for determining whether a string can be generated by a given context-free grammar. In this programming assignment, you will be implementing the CYK algorithm in your choice of programming language and using it to parse a set of input strings.

Task Description:

1. Implement the CYK algorithm in your chosen programming language. You should be able to read in a context-free grammar from a file and generate the corresponding parse table for a given input string.
2. Test your implementation on a set of input strings provided in the assignment. The input strings should include some that can be generated by the context-free grammar and some that cannot.
3. Write a report detailing your implementation and testing process. This report should include:

- An overview of your implementation, including a brief explanation of the CYK algorithm and any data structures used in your implementation.
- A description of the testing process and results, including a discussion of any issues or challenges you encountered.
- An analysis of the time and space complexity of your implementation, including any optimizations you made.

Evaluation:

Your implementation will be evaluated on the following criteria:

- Correctness: Your implementation should correctly determine whether a given string can be generated by a given context-free grammar.
- Efficiency: Your implementation should be efficient and not exceed reasonable memory or time limits.

- Clarity: Your code should be well-organized, easy to understand, and properly commented.
- Report quality: Your report should be well-written, clear, and concise.

Deliverables – This should be submitted in a zip file. 1 per group with the names of the student pair who solved each question indicated:

- Your implementation code, which should be properly commented and documented.
- A report detailing your implementation and testing process.
- Any input files used in the testing process.
- A README file with instructions on how to run your code and any dependencies required.

Additional Resources:

- The CYK algorithm is described in many textbooks on automata theory and computational linguistics. You may find these resources useful in developing your implementation and understanding the algorithm.
- There are also many open-source implementations of the CYK algorithm available online that you can reference for inspiration or guidance. However, please do not copy code directly from these sources, as that would be considered plagiarism.

Develop the Grammars for the following language, convert to CNF and then use the grammar as input into your code.

1. The grammar for the language of balanced parentheses.

2. The grammar for the language $\{a^i b^j c^k \mid i = j \text{ or } j = k\}$

3. The grammar for the language $\{x_1\#x_2\# \cdots \#x_k \mid k \geq 1, \text{each } x_i \in \{a, b\}^*, \text{and for some i and j, } x_i = x_j^R\}$. – Sipser textbook 2.6d

4. The grammar in Sipser Question 2.13.

Let G = (V,Σ, R,S) be the following grammar. V = {S, T, U}; Σ = {0, #}; and
R is the set of rules:

$S \rightarrow TT \mid U$
$T \rightarrow 0T \mid T0 \mid \#$
$U \rightarrow 0U00 \mid \#$