

CS 5000: Theory of Computation

Assignment 6: Cocke-Younger-Kasami Algorithm

Vladimir Kulyukin
Department of Computer Science
Utah State University

Learning Objectives

1. Chomsky Normal Form
2. Cocke-Younger-Kasami Algorithm
3. Dynamic Programming
4. CFG Parsing

Problem 1 (2 points)

Convert the CFG below into CNF

$S \rightarrow ABC$
 $C \rightarrow BaB \mid c$
 $B \rightarrow b \mid bb$
 $A \rightarrow a$

Problem 2 (8 points)

Implement the CYK algorithm. To make your job more interesting and less tedious, I have made a few auxiliary classes available for you at this repo - <https://github.com/VKEDCO/java/tree/master/cyk/org.vkedco.cs.cyk2k>. The CNFG.java class is an implementation of a CNF grammar. CNFProductionRHS.java is a helper class used by CNFG.java to process the right-hand sides of CNF productions. DTable.java is the 2D $D[i,j]$ table we covered in Lecture 12. CYK.java is my partial implementation of the CYK algorithm. It is partial in the sense that I removed a dozen lines or so of my code from the `isInCFL()` method that implements the actual algorithm. This part is for you to fill in. CYK.java also defines a few sample CNF grammars and shows you how you can test them. Below is the output I got in my Java console after running the tests with my CYK

implementation. You may use my implementation or choose to implement everything on your own. If you do, please implement the same tests and another test that tests your CNF grammar from question 1 on various strings. Before you work on your implementation you may want to read pp. 137-141 in the document 'Ch06_PropertiesOfCFLanguagesCYKAlgorithm.pdf' in your Box folder.

===== Test 1 =====

A -> B A

A -> a

B -> C C

B -> b

C -> A B

C -> a

S -> A B

S -> B C

Input string: baaba

Result = true

===== Test 2 =====

A -> B A

A -> a

B -> C C

B -> b

C -> A B

C -> a

S -> A B

S -> B C

Input string: baaa

Result = true

===== Test 3 =====

A -> B A

A -> a

B -> C C

B -> b

C -> A B

C -> a

S -> A B

S -> B C

Input string: baba

Result = false

===== Test 4 =====

A -> B A

A -> a

B -> C C

B -> b

C -> A B

C -> a

S -> A B

S -> B C

Input string: baaabab

Result = false

===== Test 5 =====

A -> B A

A -> a

B -> C C

B -> b

C -> A B

C -> a

S -> A B

S -> B C

Input string: aabb

Result = true