

# ST0270 Formal Languages and Compilers Assignment 2

Sergio Ramírez Rico

September 12, 2024

## 1 Deadline

September 20, at 23:59.

## 2 Assignment

The assignment is to implement, in any programming language, the Cocke-Kasami-Younger (CKY) algorithm presented in Kozen 1997, Lecture 27.

Given a context-free grammar  $G = (N, \Sigma, P, S)$  in Chomsky normal form (CNF) and a string  $x \in \Sigma^*$ , the CKY algorithm decides whether or not  $x \in L(G)$ .

You may assume the grammar is in CNF and the capital letter 'S' is its initial symbol. Assume nonterminals are capital letters and terminals *are not* upper-case letters.

### 2.1 Input/Output

Your program should fulfill the following specifications.

#### Input

A *case* is a grammar in CNF and a list of strings to be analyzed. The input of the program is as follows.

- A line with a number  $n > 0$  indicating how many cases you will receive.
- For each case, two numbers,  $k$  and  $m$ , in a single line separated by a blank space. Here,  $k$  is the number of nonterminals ( $k = |N|$ ) and  $m$  is the number of strings to be analyzed.
- Then, your program should read  $k$  lines with the productions given in the following format:  
`<nonterminal> <derivation alternatives of the nonterminal separated by blank spaces>`
- Finally,  $m$  lines each one with a string to test.

#### Output

For each case, print  $m$  individual lines, one for each string in the input. Print 'yes' when a string is generated by the grammar  $G$ , print 'no' otherwise.

## 3 Assignment Submission

Solutions should be submitted using GitHub Classroom. You should follow the instructions below for this assignment.

1. It is allowed to work in groups of no more than two students.
2. Submitting is only allowed using the following GitHub Classroom links:

- Class 1587: <https://classroom.github.com/a/KWcGSSXE>
  - Class 1588: <https://classroom.github.com/a/FJ6JAoPJ>
3. Follow the input/output instructions. Do not print extra lines.
  4. A README.md file (Markdown format) in English is required. It must contain the following information:
    - Full names of group members.
    - Versions of the operating system, programming language, and tools used in your implementation.
    - Detailed instructions for running your implementation.
  5. Do not include unnecessary files or directories in the repository.

## References

Kozen, Dexter C. (1997). *Automata and Computability*. 1st. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387949070. DOI: <https://doi.org/10.1007/978-1-4612-1844-9>.

## Example I/O

Input	Output
3	yes
5 5	yes
S AB BA SS AC BD	yes
C SB	no
D SA	no
A a	yes
B b	yes
aabbab	no
aabb	yes
ab	no
aa	yes
b	yes
4 3	no
S AB AC SS	no
C SB	no
A a	
B b	
abab	
aaabbbbaabbab	
aabab	
2 6	
S AS b	
A a	
ab	
aaaaaaaa	
aaaaaaaaaaab	
b	
bb	
abb	