ST0270 Formal Languages and Compilers

2024-2
Assignment 2
Context-Free Grammar
Teacher: *Oscar Eduardo García Quintero*

1 Deadline

The deadline date and time are available in the Interactiva mailbox.

2 Assignment

The assignment is to implement the Cocke-Kasami-Younger (CKY) algorithm presented in [Kozen, Lecture 27] (1).

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 Additional Requirement

Each group must propose a context-free grammar in Chomsky normal form (CNF) that generates a specific set of words. The grammar must be tested in your **input file**, and the words that it generates must be clearly indicated in your final report. Make sure to:

- Propose a valid CNF grammar.
- Specify the set of words the grammar should generate.
- Provide examples of words that **should** be generated by the grammar (success cases).
- Provide examples of words that **should not** be generated by the grammar (failure cases).
- Include the proposed grammar as part of the **input file** cases for your CKY algorithm.

4 Delivery

1. You must deliver the implementation compressed in a zip file on Interactiva by the deadline.
2. It is allowed to work in groups of no more than two students.
3. 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.
4. Do not include unnecessary files or directories in the repository.

References

[1] Kozen, Dexter C. *Automata and Computability*. Springer, Third printing, 1997 [2012]. Undergraduate Texts in Computer Science. https://doi.org/10.1007/978-1-4612-1844-9.

Input in `input.txt`:

```
3
5 5
S AB BA SS AC BD
C SB
D SA
A a
B b
aabbab
aabb
ab
aa
b
4 3
S AB AC SS
C SB
A a
B b
abab
aaabbbaabbab
aabab
2 6
S AS b
A a
ab
aaaaaaaa
aaaaaaaaaaab
b
bb
abb
```

Output

```
yes
yes
yes
no
no
yes
yes
no
yes
no
yes
yes
no
no
```