DOKUZ EYLUL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME3203 THEORY OF COMPUTATION

ASSIGNMENT REPORT

CONTEXT FREE GRAMMAR

By

Sadullah Cihan

Özcan Elmacı

Lecturer

Res. Asst. Dr. Meltem Yıldırım Ekici

January, 2022

İZMİR

Page

CONTENTS

CHAPTER ONE

DESCRIPTION

In this assignment, it has been asked to check and implement whether the input string belongs to the given any Context Free Grammar that will be read from the “CFG.txt” file. Also, the parse tree and derivation of the string in the grammar should be implemented. Java programming language has been used to implement this assignment.

In order to solve the grammer, Chomsky Normal Form is required. In formal language theory, a context-free grammar, G, is said to be in Chomsky normal form (first described by Noam Chomsky)[1]. In order to convert a grammar to Chomsky Normal Form, the following steps must be applied:

- Eliminate the start symbol from right-hand sides
- Eliminate ε-rules
- Eliminate unit rules
- Eliminate right-hand sides with more than 2 nonterminals

After Chomsky Normal Form is applied, the CYK algorithm should be implemented to determine whether the input string is valid for the grammar. In computer science, the Cocke–Younger–Kasami algorithm (alternatively called CYK, or CKY) is a parsing algorithm for context-free grammars published by Itiroo Sakai in 1961 [2].

The standard version of CYK operates only on context-free grammars given in Chomsky normal form (CNF). However any context-free grammar may be transformed (after convention) to a CNF grammar expressing the same language (Sipser 1997)[3]. Briefly, Chomsky Normal Form and CYK algorithms have been implemented.

CHAPTER TWO

PSEUDO CODE

```
CREATE LIST row  {
                    add(NON-TERMINAL);
                    foreach elements
                                add(element);
}
CREATE LIST firstRow=S0;

DELETE EmptyString{

          foreach(i traverse the rows of grammar){

                    foreach(j traverse the elements of rows){

                                if (row.get(j) == "#" )

                                row.remove(j); // delete empty string

                                …

                    }

          }

}

REPLACE UnitRule{

          foreach(i traverse the rows of grammar){

                    foreach(j traverse the elements of rows){

                                String nonTerminalToBeSettled = row.get(j);

                                          if (NTToBeSettled.length == 1 &
                                          NTs.contains(NTToBeSettled))

                                …

                    }

          }

}

REPLACE right-hand sides with more than 2 nonterminals();
```

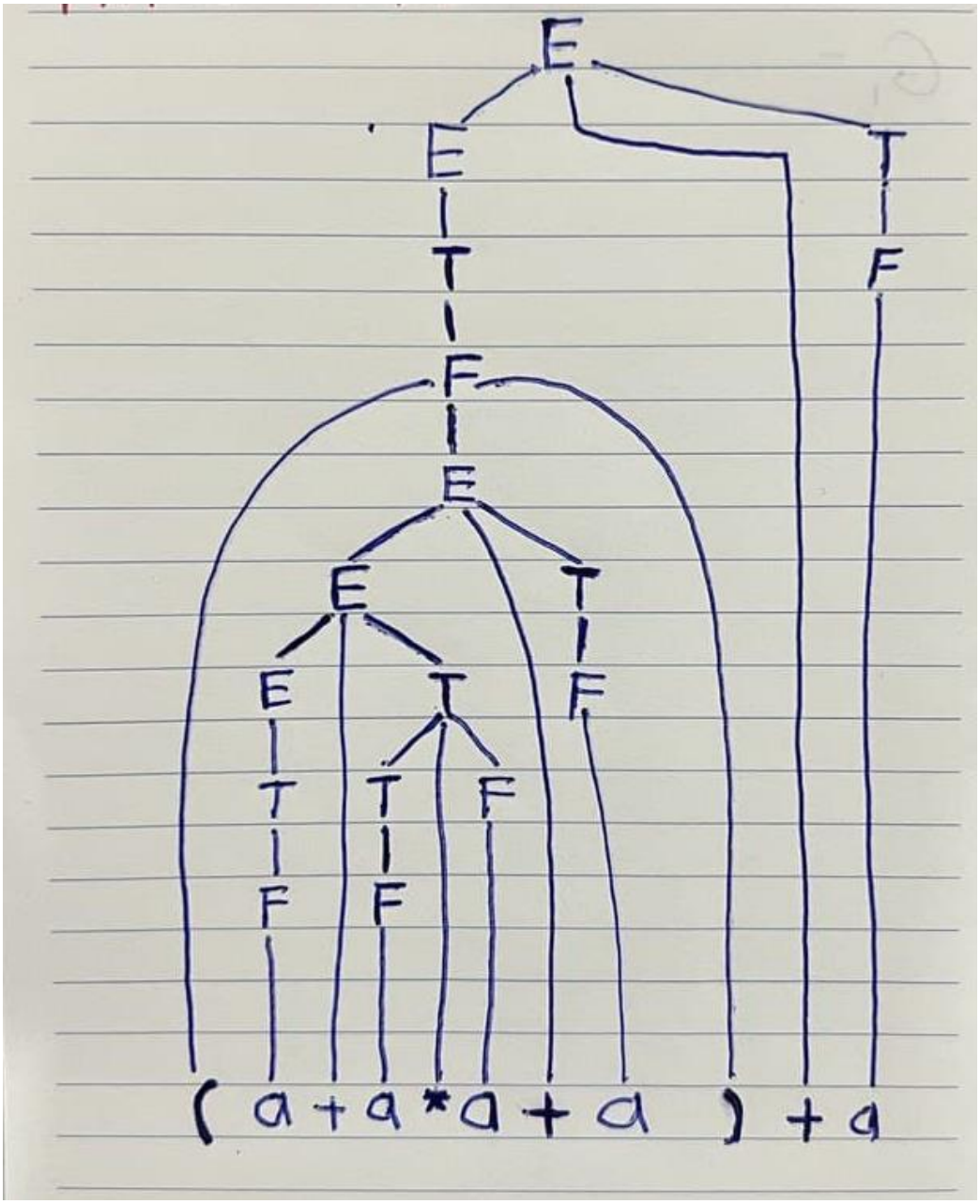
CYK Pseudo Code[3]:

let the input be a string I consisting of n characters: a_1 ... a_n.
let the grammar contain r nonterminal symbols R_1 ... R_r, with start symbol R_1.
let $P[n,n,r]$ be an array of booleans. Initialize all elements of P to false.

for each s = 1 to n
  for each unit production $R_v \rightarrow a_s$
    set $P[1,s,v]$ = true
for each l = 2 to n -- *Length of span*
  for each s = 1 to n-l+1 -- *Start of span*
    for each p = 1 to l-1 -- *Partition of span*
      for each production $R_a \rightarrow R_b\ R_c$
        if $P[p,s,b]$ and $P[l\text{-}p,s\text{+}p,c]$ **then** set $P[l,s,a]$ = true
if P[n,*1*,*1*] is true **then**
  I is member of language
else
  I is not member of language

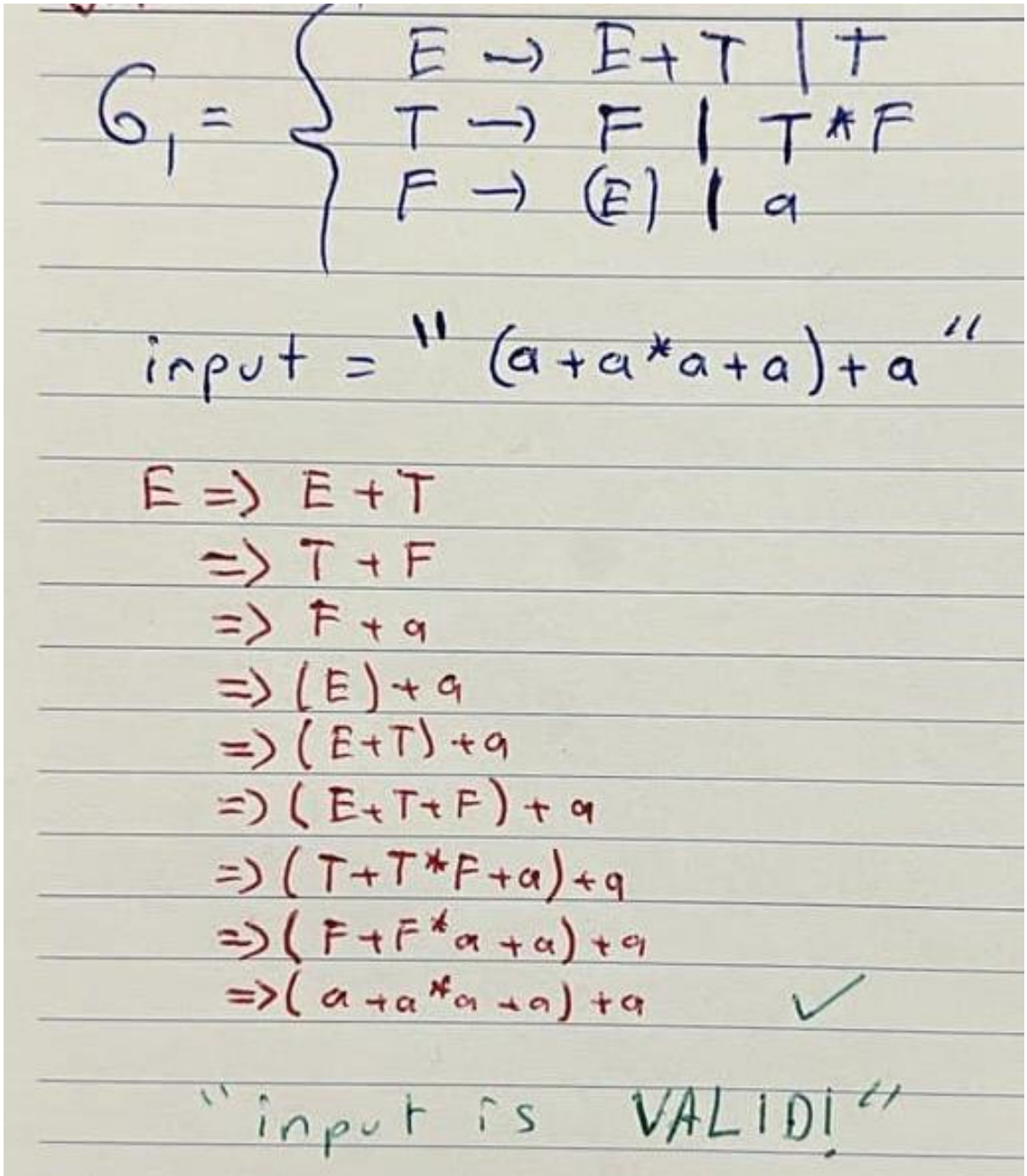
CHAPTER THREE

PARSING TREE



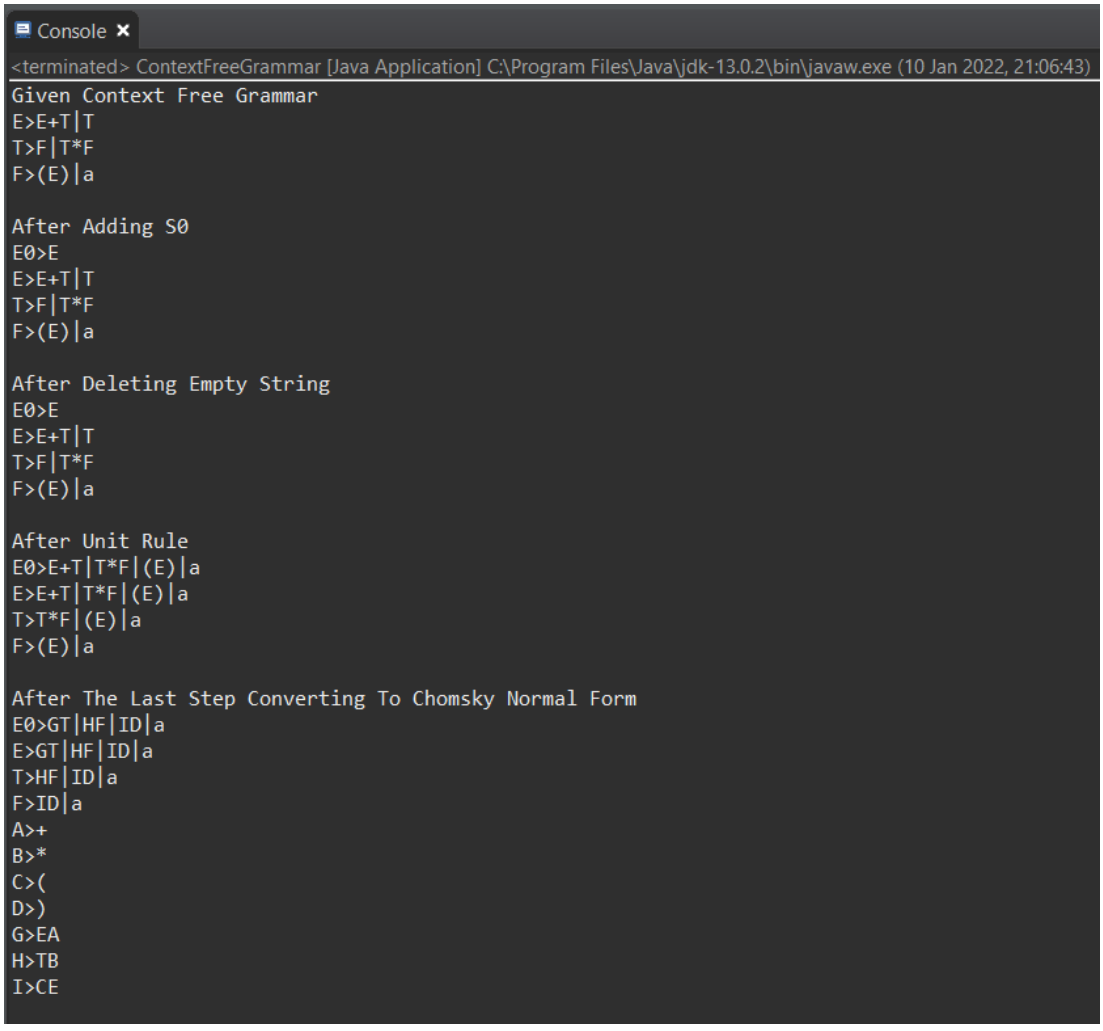
CHAPTER FOUR

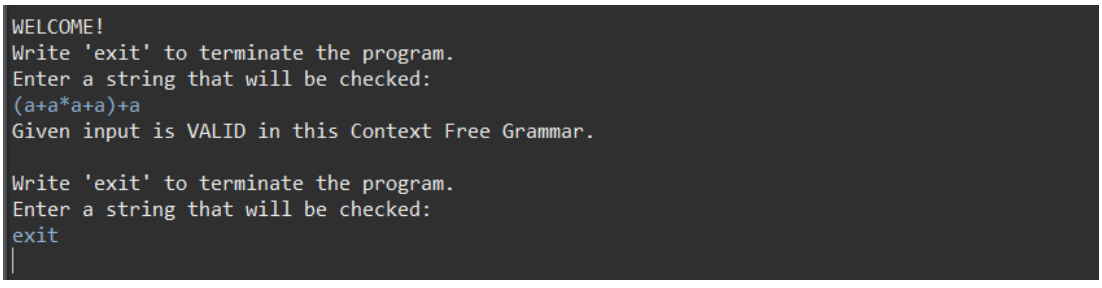
DERIVATION



CHAPTER FIVE

SCREENSHOTS OF THE PROGRAM





REFERENCES

1. Chomsky, Noam (1959). "On Certain Formal Properties of Grammars". Information and Control. 2 (2): 137–167. doi:10.1016/S0019-9958(59)90362-6. Here: Sect.6, p.152ff.
2. Grune, Dick (2008). Parsing techniques : a practical guide (2nd ed.). New York: Springer. p. 579. ISBN 978-0-387-20248-8.
3. N.D , https://en.wikipedia.org/wiki/CYK_algorithm#cite_note-1 , 10th January of 2022.