DOKUZ EYLUL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME 3203 THEORY OF COMPUTATION

CFG to CNF Conversion Homework

DR.ÖĞR.ÜYESİ ÖZLEM AKTAŞ

DR. MELTEM YILDIRIM EKİCİ

by

YASSER EL HASAN – 2019510006

ULAŞ CAN YUSUFOĞLU – 2018510070

IZMIR

26.12.2022

DESCRIPTION

The given Java code reads an input file and converts it to Chomsky Normal Form using the CFG_CNF class. **A** context-free grammar is in Chomsky Normal Form if every production is of the form **A** → **BC** or **A** → **a**, where a denotes a terminal and **A, B, C** denote variables where neither **B** nor **C** is the start variable. In addition, there is a production **S** → **ε** if and only if ε belongs to the language.

The **CFG_CNF** class contains various methods that perform the different steps of the conversion process, such as eliminating ε-productions, eliminating single variables, and replacing two terminal variables with a new variable.

The main procedure first creates a **ReadInput** object to read the input file, and a **CFG_CNF** object to perform the conversion. It sets the input and line count for the converter, and stores the input in a map.

Next, the **main** procedure prints the original context-free grammar and calls the **eliminateEpsilon** method from the **CFG_CNF** class to eliminate ε-productions. It then calls the **removeDuplicateKeyValue** method to remove duplicate key-value pairs, and the **eliminateSingleVariable** method to eliminate single variables in every production.

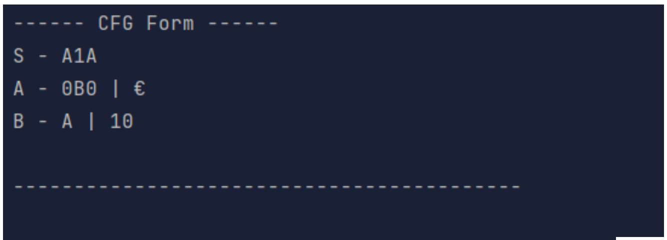
After that, the **main** procedure calls the **assignVariable** method to assign new variables to terminal productions, and the **removeThreeTerminal** method to replace two terminal variables with a new variable.

Finally, the main procedure prints the resulting Chomsky Normal Form grammar.

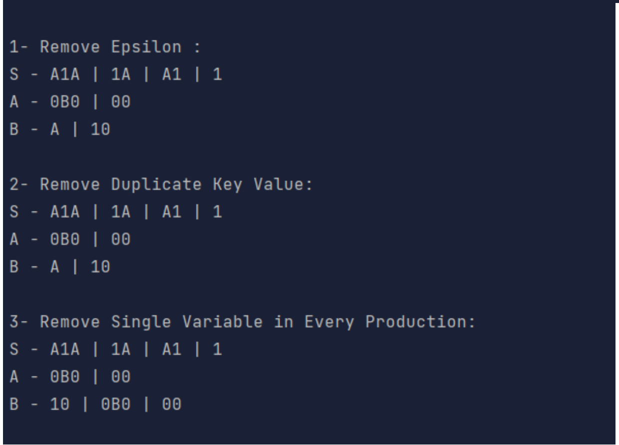
PSEUDOCODE

```
Main() {
    // Read input file and create ReadInput and CFG_CNF objects
    input = ReadInput()
    converter = CFG_CNF()
    // Set input and line count for converter
    converter.setInput(input.getInput())
    converter.setLineCount(input.getLineCount())
    // Store input in map
    cfg = input.getCfg()
    // Print original context-free grammar
    print("Original context-free grammar:")
    input.printCfg(cfg)

    // Eliminate ε-productions
    cfg = converter.eliminateEpsilon(cfg)

    // Remove duplicate key-value pairs
    cfg = converter.removeDuplicateKeyValue(cfg)

    // Eliminate single variables
    cfg = converter.eliminateSingleVariable(cfg)
    // Assign new variables to terminal productions
    cfg = converter.assignVariable(cfg)
    // Replace two terminal variables with a new variable
    cfg = converter.removeThreeTerminal(cfg)
    // Print resulting Chomsky Normal Form grammar
    print("\nChomsky Normal Form:")
    input.printCfg(cfg)
}
```

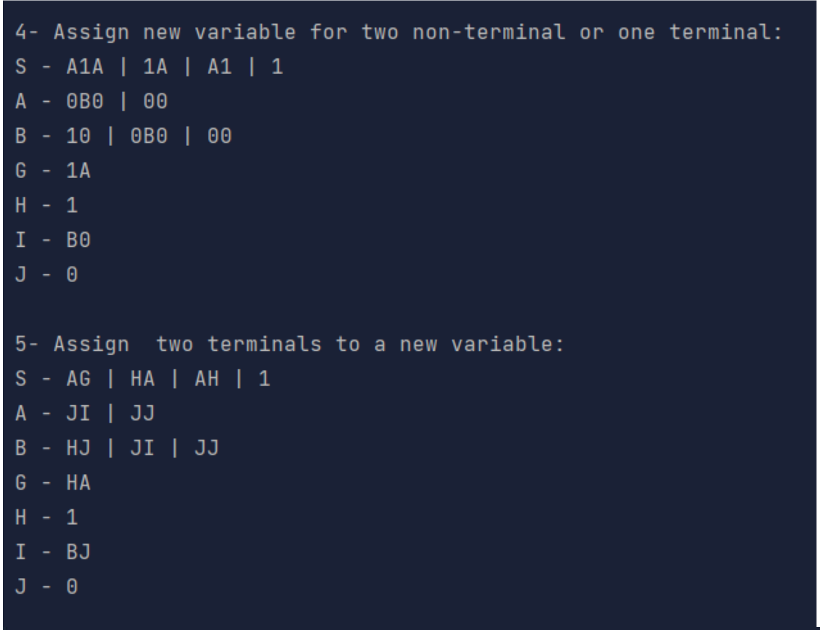
SCREENSHOOTS



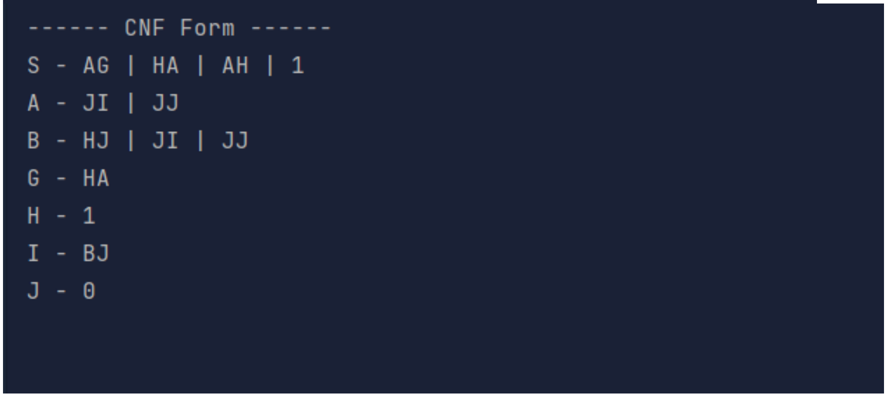
```
------ CFG Form ------
S - A1A
A - 0B0 | €
B - A | 10

------------------------------------------
```



```
1- Remove Epsilon :
S - A1A | 1A | A1 | 1
A - 0B0 | 00
B - A | 10

2- Remove Duplicate Key Value:
S - A1A | 1A | A1 | 1
A - 0B0 | 00
B - A | 10

3- Remove Single Variable in Every Production:
S - A1A | 1A | A1 | 1
A - 0B0 | 00
B - 10 | 0B0 | 00
```



```
4- Assign new variable for two non-terminal or one terminal:
S - A1A | 1A | A1 | 1
A - 0B0 | 00
B - 10 | 0B0 | 00
G - 1A
H - 1
I - B0
J - 0

5- Assign  two terminals to a new variable:
S - AG | HA | AH | 1
A - JI | JJ
B - HJ | JI | JJ
G - HA
H - 1
I - BJ
J - 0
```



```
------ CNF Form ------
S - AG | HA | AH | 1
A - JI | JJ
B - HJ | JI | JJ
G - HA
H - 1
I - BJ
J - 0
```