

# CS410 - Automata Theory and Formal Languages

-

## Project 2 Design Report

### Introduction

This project aims to create a program that converts a context-free grammar to its equivalent Chomsky normal form. The program will get CFG as an input with the help of a pre-designed .txt file and it will print an equivalent CNF of it with the same design.

### Problem Definition

Context Free Grammar can be defined by four tuples which are:

**1-N is a set of non-terminal symbols.**

**2-T is a set of terminals.**

**3-P is a set of rules.**

**4-S is the start symbol.**

To check if a CFG is in CNF all production rules must satisfy one of the following conditions.

**1-Only the start symbol generates epsilon. (ex: if S is start  $S \rightarrow (\epsilon)$ )**

**2-A non terminal generates 2 non terminals. (ex:  $A \rightarrow BC$ )**

**3-A non terminal generates a terminal. (ex:  $A \rightarrow b$ )**

Since all CFG's have an equivalent CNF of it, it is possible to convert every CFG input to equivalent CNF of it.

### Tools

JAVA : Programming language for implementation

Eclipse: Integrated development environment for JAVA

### Usage of The program

To use the java application CFG to CNF Converter follow these steps:

- Compile the file:  
`javac Selami_Karakas_S018705.java`
- Run the program with giving your path to input file as an argument:  
`java Selami_Karakas_S018705 <Path to Input File>`  
**Example:** `java Selami_Karakas_S018705 /home/selami/CS410/G1.txt`  
**Note:** Please be sure that your path does not include any whitespaces.

### Method

To create the program following structures and functions needed:

- A method that reads input files and contains its tuples in arraylists.
- A method that checks whether there is any start terminal on the right hand side of the production rules.
- A method that returns a free char that is not used in CFG from the alphabet.
- A method that returns the index of a production rule if it contains epsilon.

- A method that returns the index of a production rule if it is Unit Production.
- A method that returns the index of a production rule if it has a terminal with some terminals or non terminals on the right hand side.
- A method that returns the index of a production rule if it has more than two terminals on the right hand side.

## **Implementation Steps Of Conversion**

In this section the conversion function is explained step by step with an example. The input example is at the end of this report.

## **Input Output Format**

Here is the format mentioned in a few parts.

NON-TERMINAL

S

F

TERMINAL

0

1

RULES

S:00S

S:11F

F:00F

F:e

START

S