CS 410-PROJECT2 DESIGN REPORT
FALL2022

1. Introduction

In the project, implementing a program which will be converted CFGs to CNFs is aimed. The input will be a text file that contains the information of terminals, non-terminals, rules and start state. The output, the CNF version of CFG, will have the same format with input file.

In the section 2, the tools that will be used to design and create this project are introduced.

2. Tools Used

In the project, the tools are used in below.

- Java 8: Used as the software language.
- IntelliJ IDEA Ultimate 2019: Used as the Integrated Development Environment.
- Unified Modeling Language (UML): Used in creating diagram.
- Java Util Package: Used to create ArrayList objects to manipulate converting NFA to DFA.
- Java Io Package: Used to read and file.
- Creatly: Used website to create diagram.

3. Software Design

In this section, algorithm is explained and activity diagram is introduced.

The algorithm contains functions to make the algorithm simplified. The list of the functions are below:

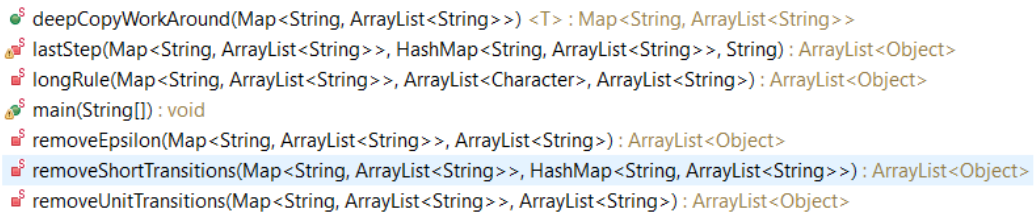

Figure 1: List of functions of the CFG to CNF converter

deepCopyWorkAround: It is used to deep copy the map objects. It was used on almost every step of the algorithm.

lastStep: The algorithm contains 4 steps. It is the last step of the algorithm. It checks whether there is a nonused rule or not.

longRule: It is the first step of the algorithm. The function makes the long rule shorter with new rule definitions.

removeEpsilon: It is the second step of the algorithm. The function removes epsilons on the rules.

removeShortTransitions: It is included in third step of the algorithm. The function removes short transitions after removing unit transitions.

removeUnitTransitions: It is third step of the algorithm. The function removes unit transitions.

The algorithm start with reading the grammar file and add the ArrayLists and strings which are usedLetters, alphabet, rules and start necessarily. After that, dict map object created to keep the strings that were converted on the conversion steps. For every step, the new objects are created because of returning multiple objects for every function on the algorithm. The first function, which is named longRule, used to removing large rule. For example, if the rule is S: XYZT, ZT will be converted to A, and YA converted to B and the result will be XB. The second algorithm, which is named removeEpsilon, used to remove epsilons on the rules. For example, if the rules are A: BCD | e and X: ABC, the epsilon on the A will be removed and rules of X will be converted to X: ABC | BC. The third function, which is named removeShortTransitions, used to form the short transitions to proper transition. For example, if the rules are A:B and B: AB, it is not proper for Chomsky Normal form. It should be converted to A: AB. The last step on the function helps to remove unit transitions with dictionary object. The last function is lastStep. LastStep function is used to remove nonused transitions.

Finally, after converting successfully, the rules, usedLetters, alphabet and start will be printed on console with the pattern that is used on grammar file.

NON-TERMINAL // syntactic variables that denote sets of string
S
F
TERMINAL // components of the sentences generated using a grammar, it can represented with small letters or numbers.
0
1
RULES // set of recursive rules used to generate pattern of strings
S:00S
S:11F
F:00F
F:e
START // start variable of the grammar
S

Example 1: Example of CFG, the form will be same on the converted CNF