Automata Finals Project

Made by Group 4 from BCS31

Members:

Estelle Marie Dasal

Vea Annamika De Guzman

Eisley Kirk Maglabe

Renbert Rosas

Table of Contents

Introduction

This User Manual is written as a guide for the usage of BCS31’s Group 4’s Automata Theory and Formal Languages Final Project, a program made for the completion of the academic requirement of creating said program to simulate and evaluate strings for two (2) given regular expressions by the professor, Ms. Jennylinde Manaois.

Key Features

The key features of this website are:

- Validation and simulation of inputted strings
- Display the context-free grammar (CFG) and pushdown automata (PDA) of the given regular expressions (RegEx)

Requirements

The requirements for running this website are as follows:

- Internet connection – for accessing the website online and to download a copy of the repository for offline purposes
- Any computer (preferred) or mobile device
- Browser

Website Access

QR code for the website

QR code for the GitHub repository

You can scan the above QR codes to access the project website (on the left) and the GitHub repository (on the right) of the project. Or you can choose to access it using the following links:

- Website: https://yahan-vdg.github.io/AutomataFinalsProject/
- GitHub repository: https://github.com/Yahan-VDG/AutomataFinalsProject

About the Project

This project is developed in a collaboration between the group members. The RegEx that the website has are the following:

- (bab+bbb)(a*b*)(a*+b*)(ba)*(aba)(bab+aba)*bb(a+b)*(bab+aba)(a+b)* with the alphabet of a and b. This RegEx will be referred to as Expression #1 onward.
- (1+0)*1*0*(101+01+000)(1+0)*(101+00)*(111+00+101)(1+0)* with the alphabet of 0 and 1. This RegEx will be referred to as Expression #2 onward.

The specifications of this program were to create a correct deterministic finite automaton (DFA), pushdown automata (PDA), context-free grammar (CFG) for the RegEx's given.

Language Used

This website is hosted in GitHub Pages and is written using JavaScript, Cascading Style Sheets (CSS), and Hypertext Markup Language (HTML).

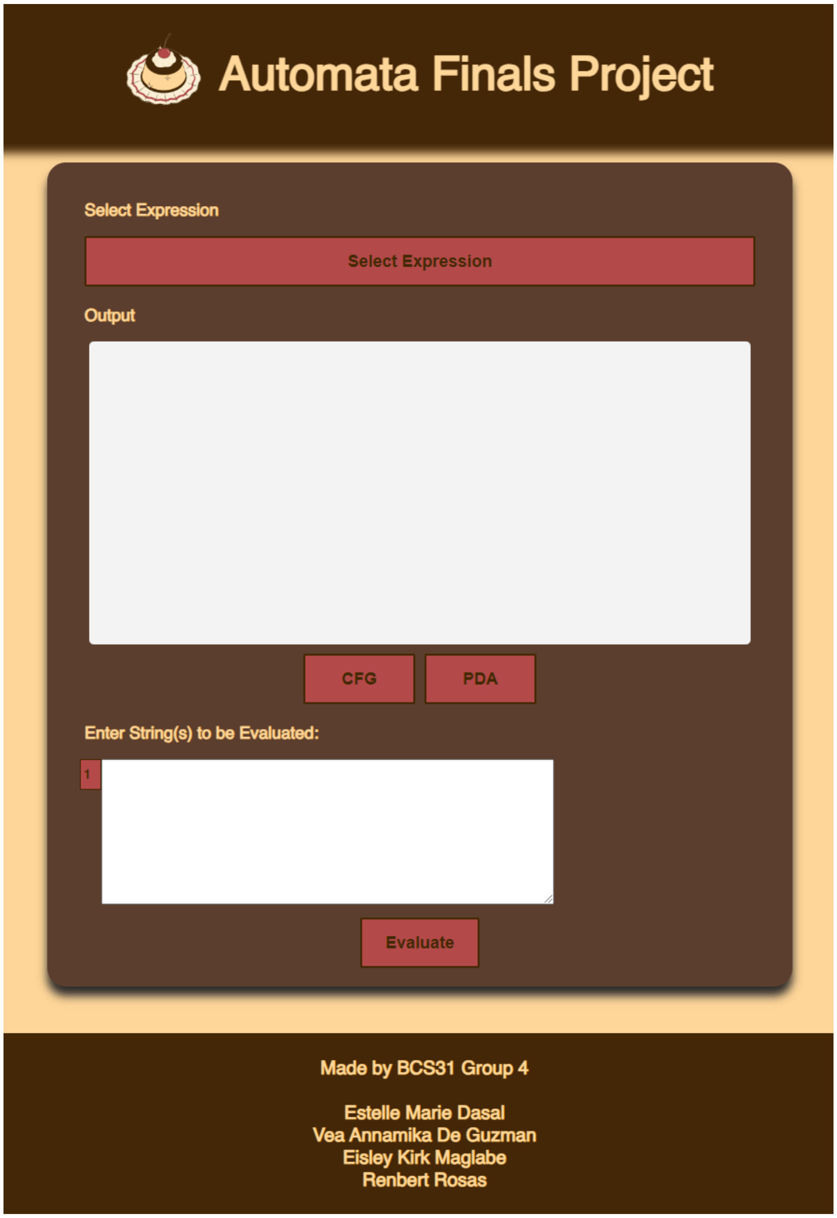
Program Features

The program will allow users to test multiple strings at once for the two RegEx's. They can then see the DFA for the RegEx they have chosen. The users can also view the CFG and PDA of the RegEx by pressing the corresponding buttons on the website. Once the users have entered the strings they wish to test on the RegEx, the users are to click the Evaluate button in order to see whether the string entered is valid or invalid for that

RegEx. They can also see an animation on the DFA of their entered string if they choose to click on the Simulate buttons on the right side of the input box.

How to Use

Website Proper



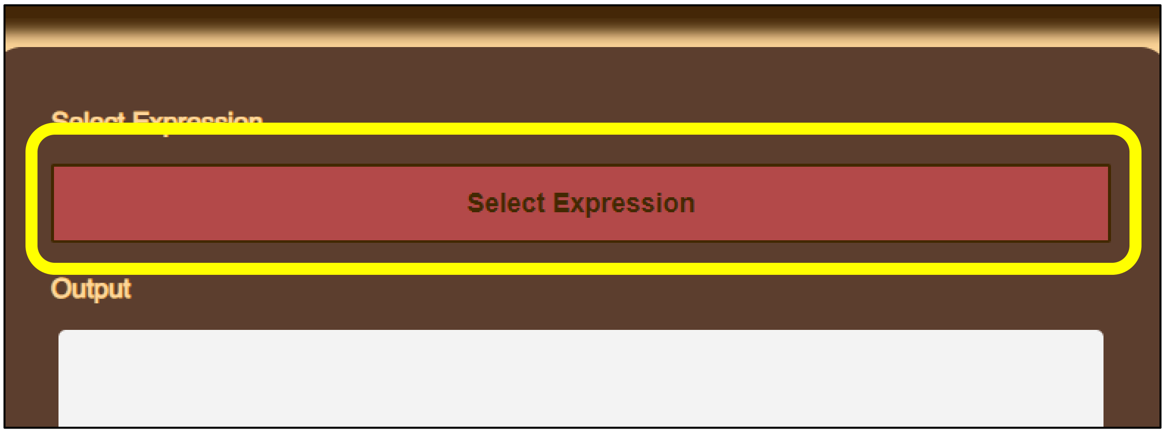
Made by BCS31 Group 4

Estelle Marie Dasal
Vea Annamika De Guzman
Eisley Kirk Maglabe
Renbert Rosas

Pictured on the previous page is the website's appearance. The logo and the website name are shown on top then the body – the heart of the program itself is shown. Lastly, the credits are on the footer of the website.

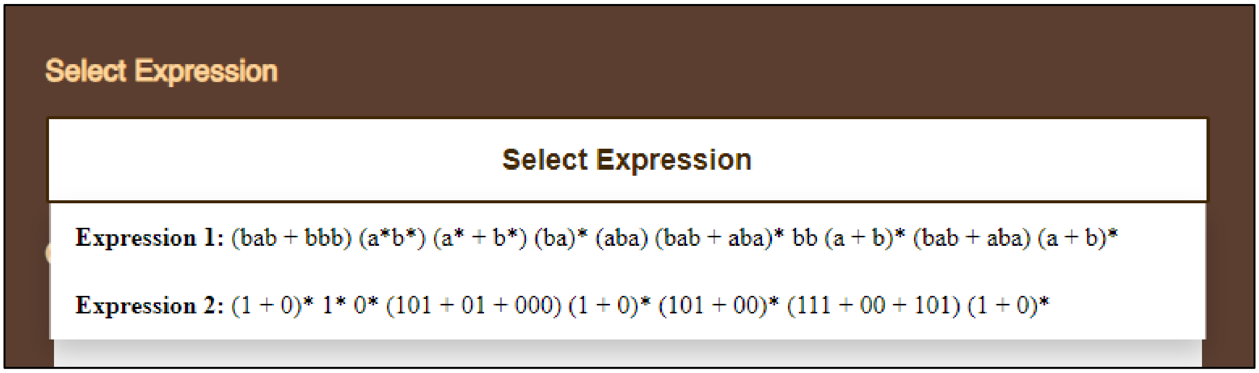
RegEx Selection

To select a regular expression to test on, the users must click on the "Select Expression" button below the header of the website.



Select Expression button boxed

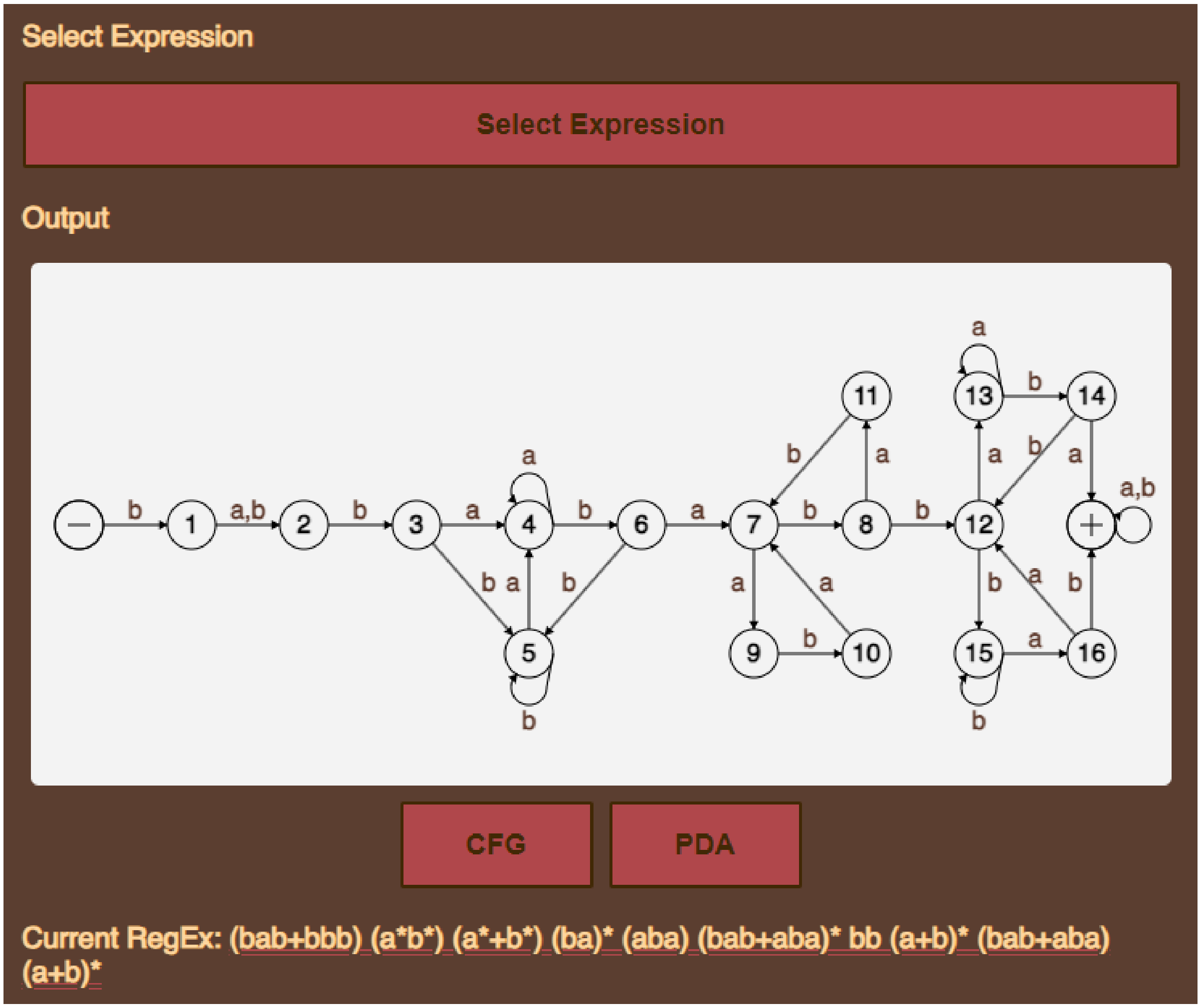
Next, a dropdown list of the expressions will be displayed. Once the user has clicked on which expression to select, the corresponding DFA will be shown (see page 9 for an example) and the expression the user has selected will be shown below the CFG and PDA buttons.



Expression 1: (bab + bbb) (a*b*) (a* + b*) (ba)* (aba) (bab + aba)* bb (a + b)* (bab + aba) (a + b)*

Expression 2: (1 + 0)* 1* 0* (101 + 01 + 000) (1 + 0)* (101 + 00)* (111 + 00 + 101) (1 + 0)*

Expressions dropdown list

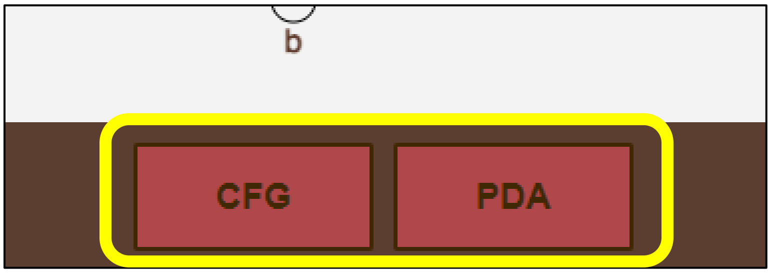


Current RegEx: (bab+bbb) (a*b*) (a*+b*) (ba)* (aba) (bab+aba)* bb (a+b)* (bab+aba) (a+b)*

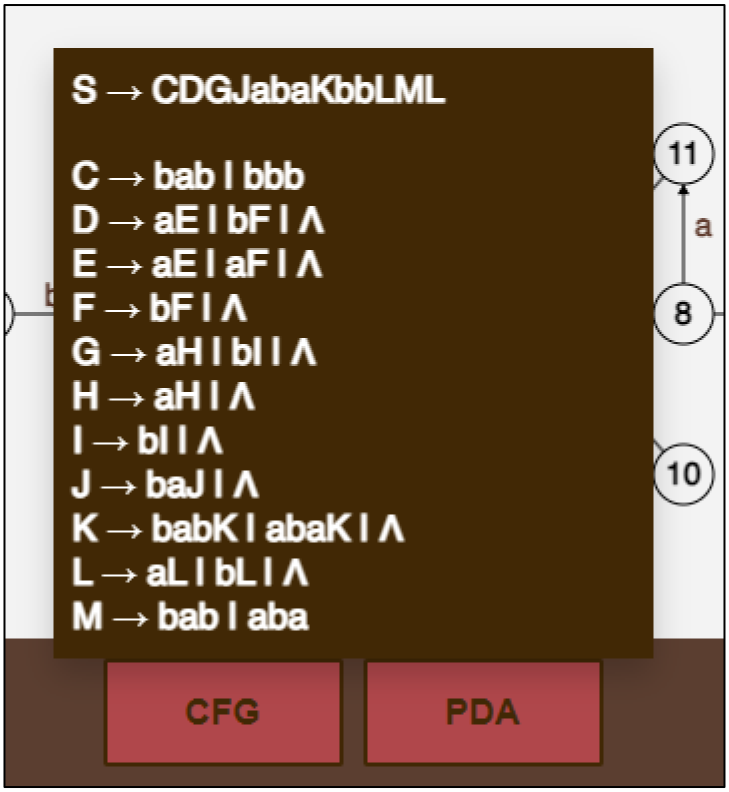
DFA output for Expression #1

CFG and PDA

To see the CFG and PDA of the chosen RegEx, the user must click on the corresponding buttons of which they want to see.



CFG and PDA buttons boxed



S → CDGJabaKbbLML

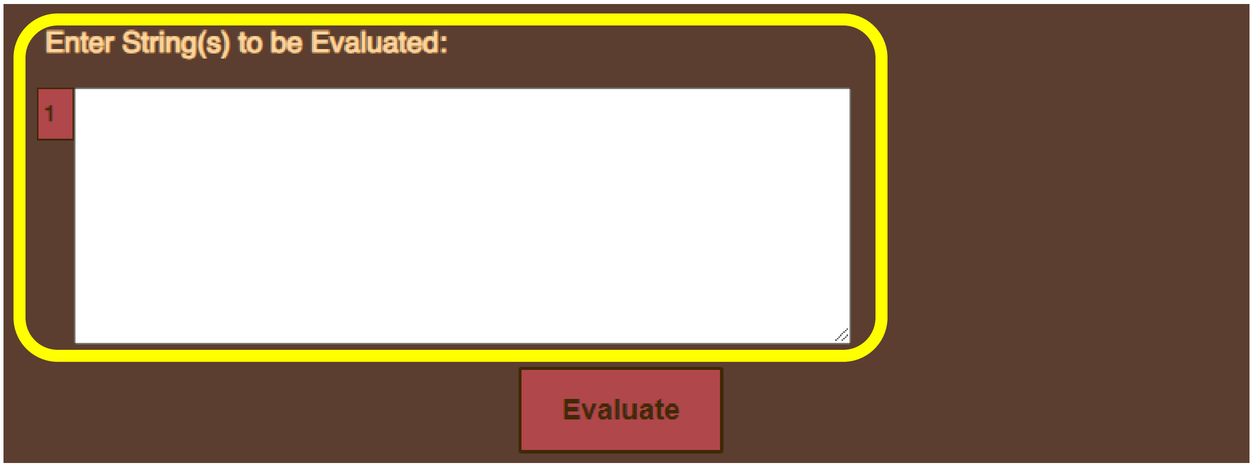
C → bab | bbb
D → aE | bF | Λ
E → aE | aF | Λ
F → bF | Λ
G → aH | bI | Λ
H → aH | Λ
I → bI | Λ
J → baJ | Λ
K → babK | abaK | Λ
L → aL | bL | Λ
M → bab | aba

CFG for Expression #1

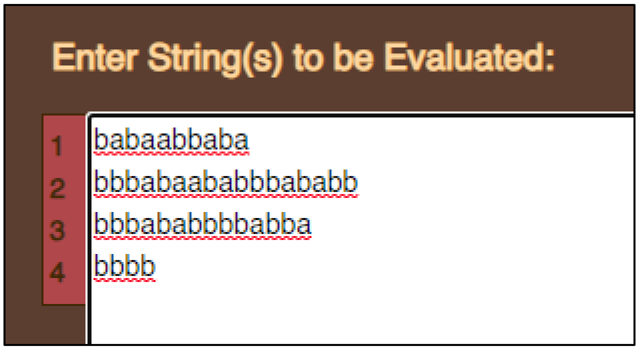
Click outside of the CFG or PDA pop-up to close the pop-up. The users can choose to switch between Expression #1 and #2 to see the CFG and/or PDA for both.

String Validation

To test the list of strings the user wishes to test on their chosen RegEx, the user must enter it in the input box on the bottom of the website.



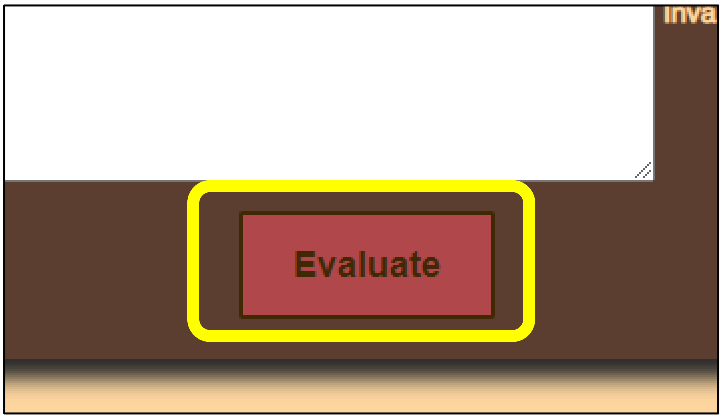
String input box boxed



```
babaabbaba
bbbabaababbbababb
bbbababbbbabba
bbbb
```

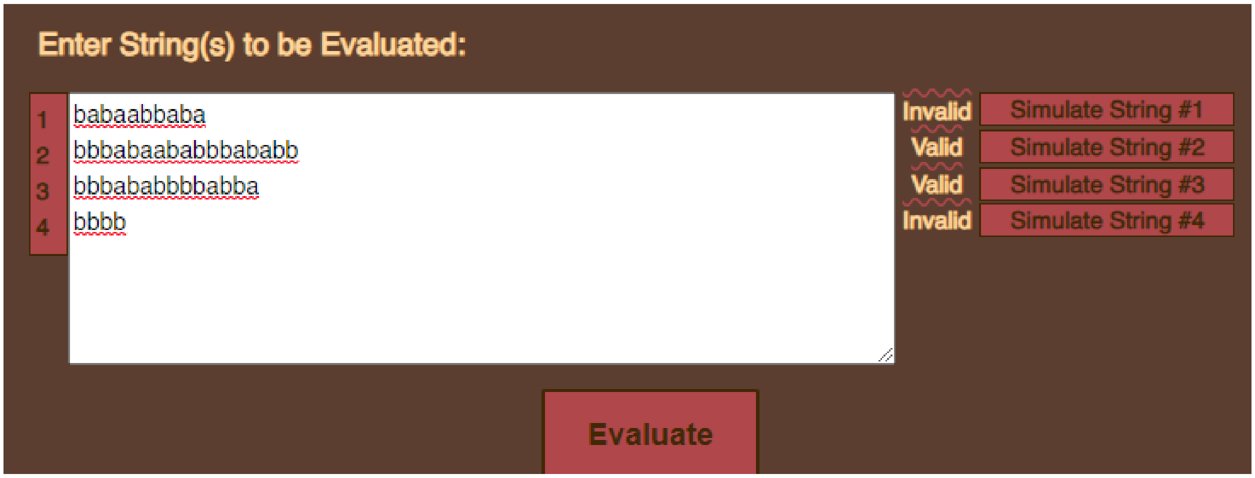
Example strings entered.

A number list on the left will show how many strings the user has entered. Additionally, the space on the right of the input box is intentional. It is for the displaying of the validation result and simulation buttons once the user has pressed the Evaluate button.



Evaluate button boxed

After clicking on the Evaluate button, the user will now see the results of the validation and a button to click on to see the animated simulation of the string on the DFA.

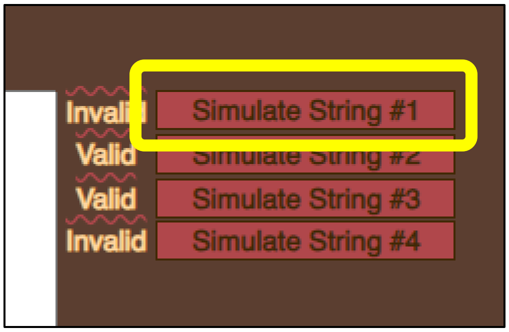

Enter String(s) to be Evaluated:

1 babaabbaba — Invalid
2 bbbabaababbbababb — Valid
3 bbbababbbbabba — Valid
4 bbbb — Invalid

Results from the example strings

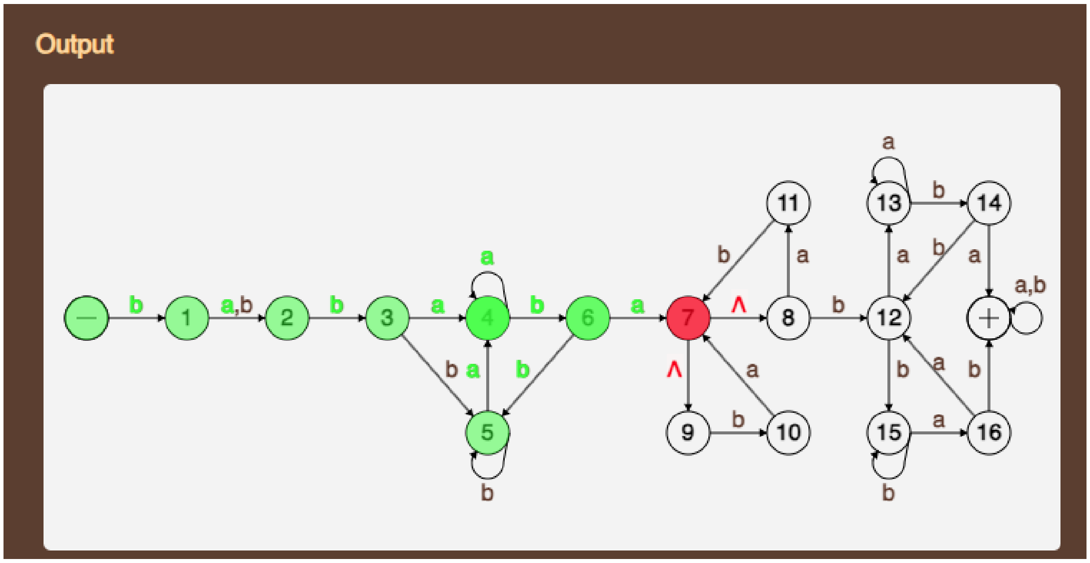
String Animation

Once the user has entered their list of strings and clicked on the Evaluate button to see the validation result and simulate buttons. With the help of the number list on the left side of the input box, the user can determine which string they want to see animated. The simulate buttons are named correspondingly to the number of strings entered. Click on the simulate button to see the string animation on the DFA.



Simulate string button boxed

The animation is a simple highlight of the current state and symbol. Valid symbols are highlighted green and invalid symbols are highlighted red.



String animation

Credits

The group members are from the block of BCS31, batch 2022 – 2023, in the course of Computer Science from De La Salle University – Dasmariñas. The members are:

- Estelle Marie Dasal – the group leader
- Vea Annamika De Guzman
- Eisley Kirk Maglabe
- Renbert Rosas

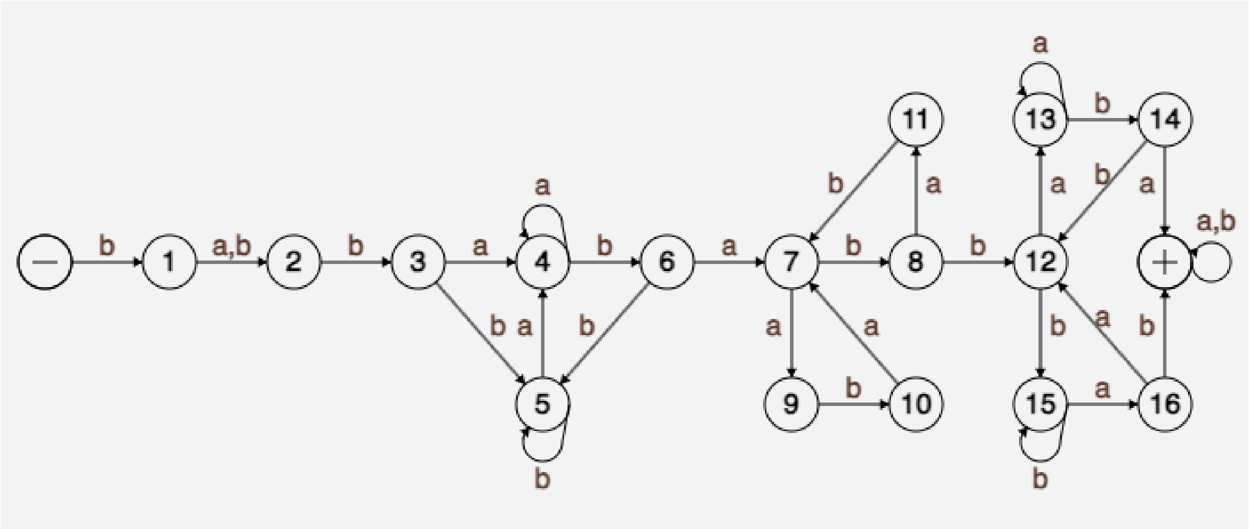
Contributions

- DFA and CFG – De Guzman
- JavaScript code for the DFA – Rosas
- PDA – Dasal
- UI – Maglabe
- RegEx validation code and string animation code – Dasal and Rosas

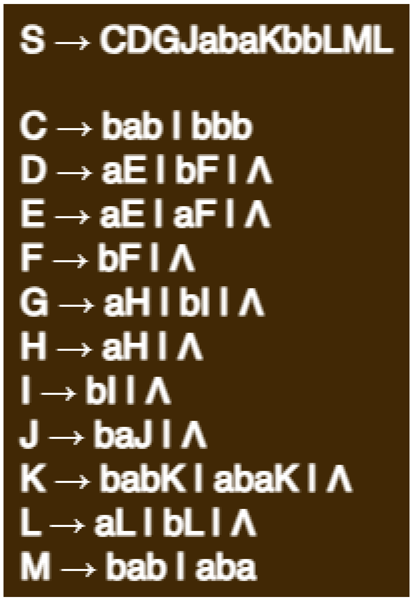
Documentation

Expression 1 (a & b)

Deterministic Finite Automaton



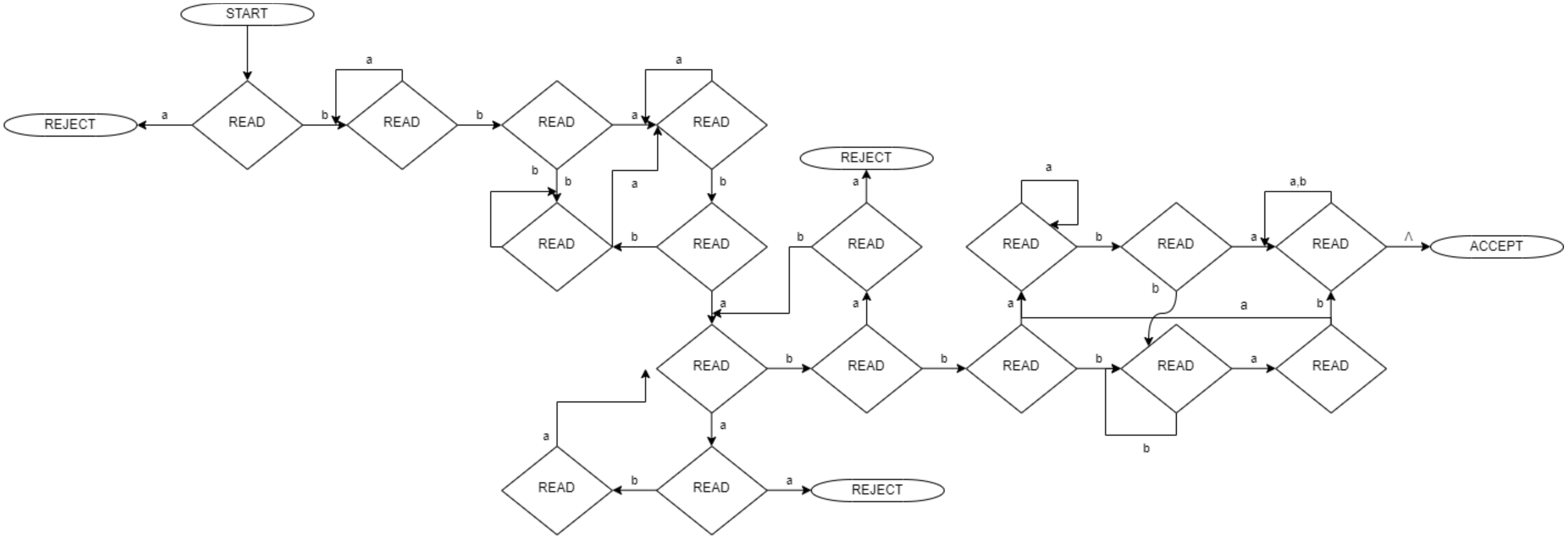
Context-Free Grammar



S → CDGJabaKbbLML

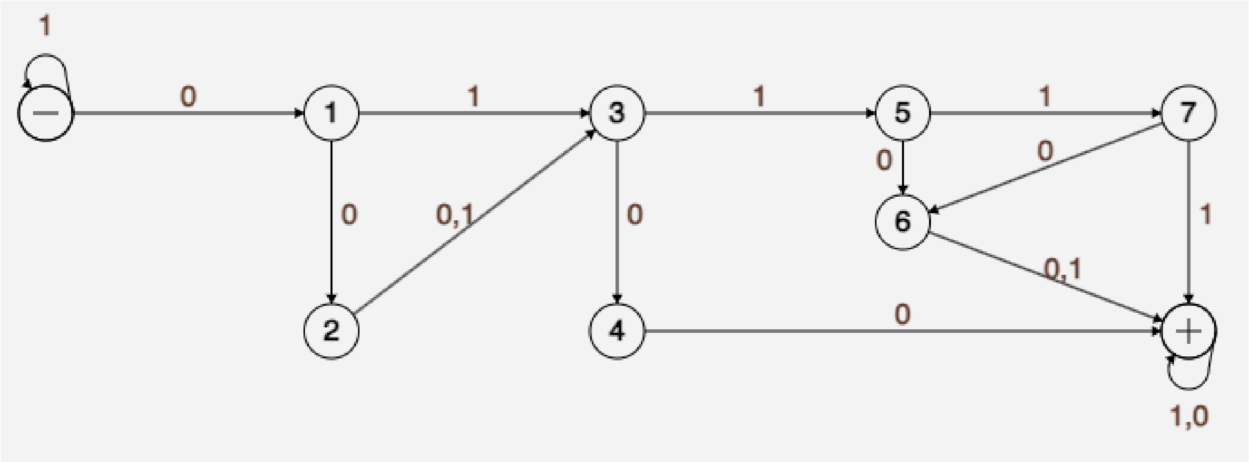
C → bab | bbb
D → aE | bF | Λ
E → aE | aF | Λ
F → bF | Λ
G → aH | bI | Λ
H → aH | Λ
I → bI | Λ
J → baJ | Λ
K → babK | abaK | Λ
L → aL | bL | Λ
M → bab | aba

Pushdown Automaton

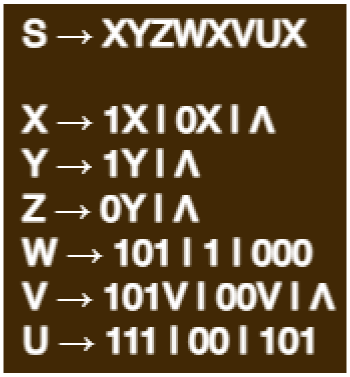


Expression 2 (0 & 1)

Deterministic Finite Automaton



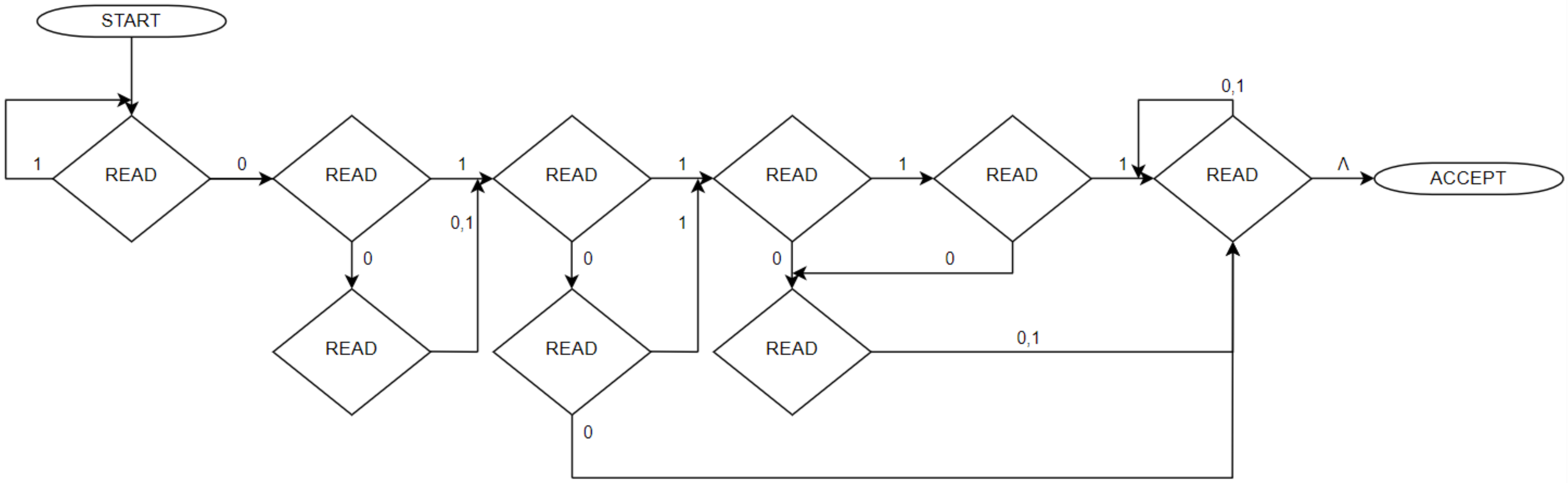
Context-Free Grammar



S → XYZWXVUX

X → 1X | 0X | Λ
Y → 1Y | Λ
Z → 0Y | Λ
W → 101 | 1 | 000
V → 101V | 00V | Λ
U → 111 | 00 | 101

Pushdown Automata


START
READ
READ
READ
READ
READ
READ
ACCEPT
READ
READ
READ
1
0
1
1
1
1
0,1
Λ
0
0,1
0
1
0
0
0
0,1
0