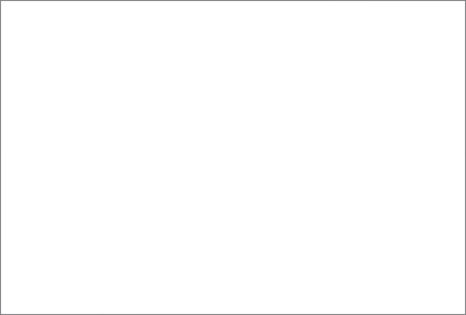
The SS language:

Lexical Analyzer

# Jinyang Li



# Spring 2017

Instructor: Suman Saha

# Introduction

The SS - Super Simple language is designed to satisfy a course project requirement with very human like syntax and easy understand programming structure. This is the lexical analyzer documentation.

# Direction To Use

Using flex to generate lexical analyzer source code.

The source file used for flex is lex.l

$lex lex.l

Using cc to generate lexical analyzer executable.

$cc lex.yy.c -o example -ll

You can generate analyzer on different platform using gcc or cc.

After obtain lexical analyzer, just run like:

$AnalyzerExecutableName <input file>

for example:

$ example a.c

Source code can be found in same zip folder.

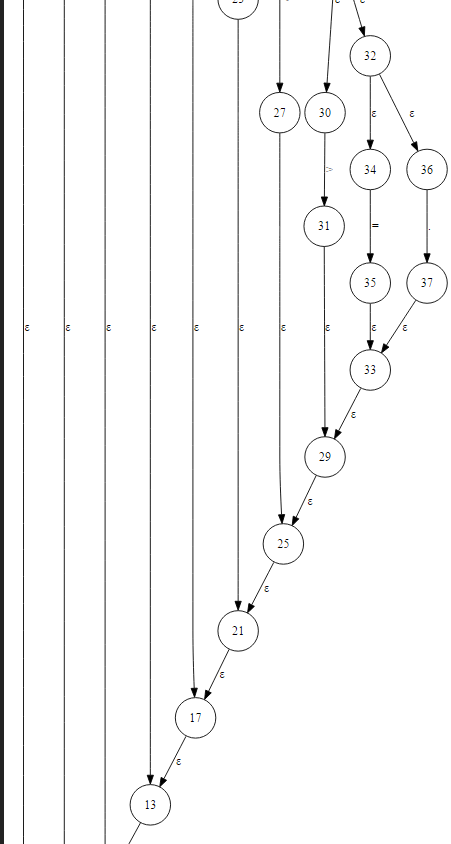
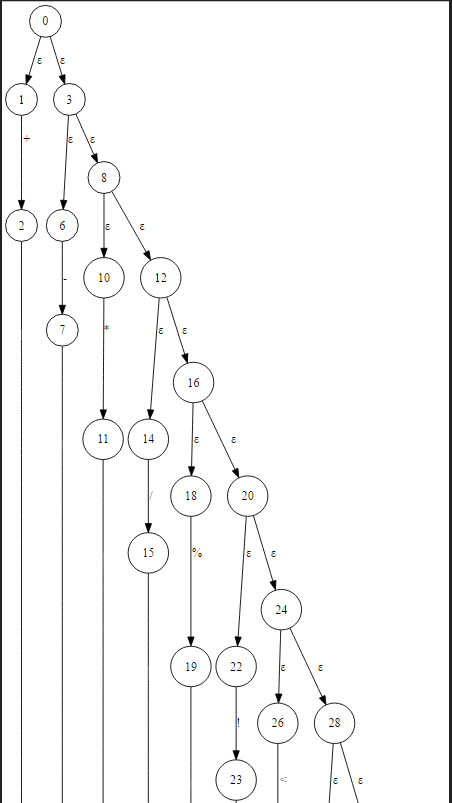
# Tokens

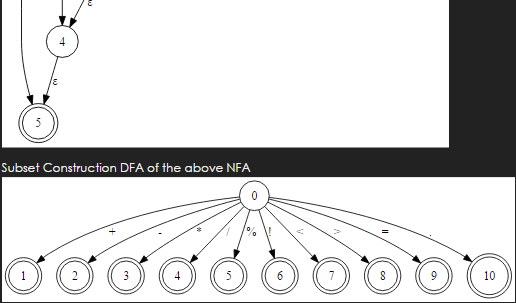
Operators - same meaning as Java

+ - \* / % = > < >= <=

Regular Expression:

[+\-\*\/%!<>=.]|==|[+][+]|—|>=|<=|!=

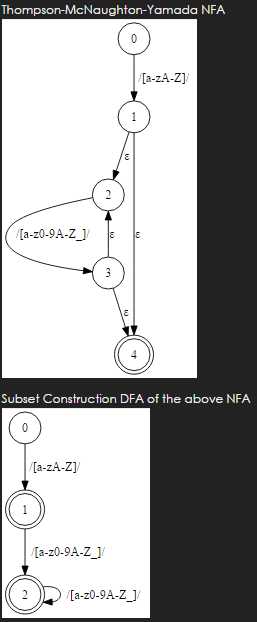




Identifier - beginning with lowercase character, then followed by any string

Regular Expression:

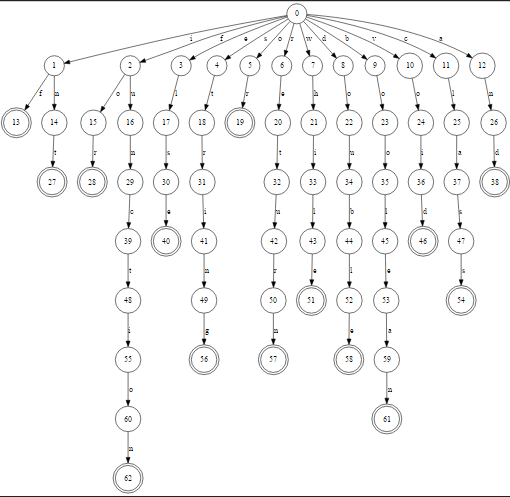
[a-zA-Z][a-z0-9A-Z\_]\*



Keywords -

Regular Expression:

if|else|for|while|int|double|void|string|class|boolean|function|return|and|or



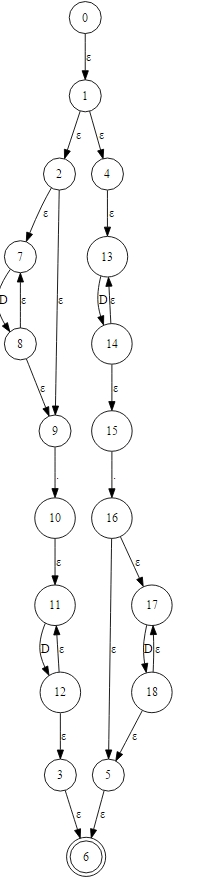
Int type:

[0-9]+



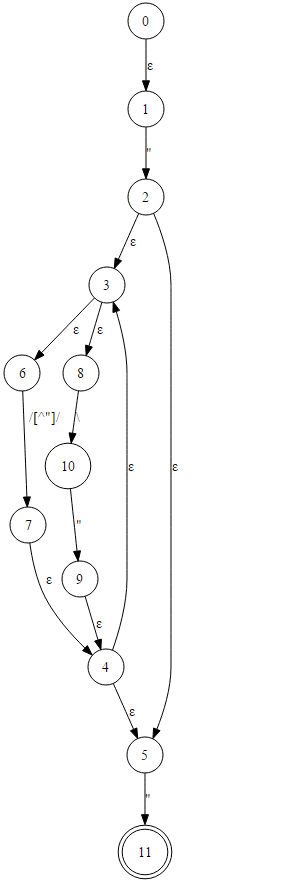
Double type

D\*.D+|D+.D\*



String type:

"([^"]|\\")\*"



Boolean type

True|False

