



UNIVERSITY OF AMSTERDAM

## ASSIGNMENT 2

---

# Lexicographic Analysis

---

February 17, 2021

*Students:*

René Kok

13671146

Aram Mutlu

13574116

*Lecturer:*

Dhr. dr. C.U. Grelck

*Course:*

Compiler Construction

*Course code:*

5062COMP6Y

## 1 Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## 2 Assignment

### 2.1 Thompson's Construction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

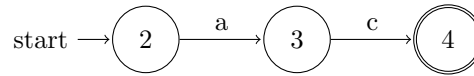


Figure 1: Step 1 creating NFA for  $(ac) \rightarrow (ac|ab)^*$

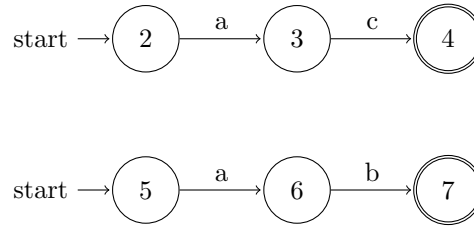


Figure 2: Step 2 creating NFA for  $(ab) \rightarrow (ac|ab)^*$

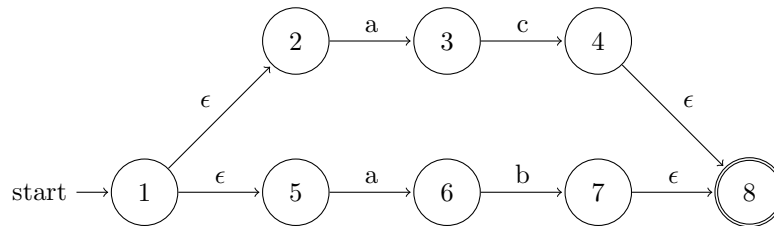


Figure 3: Step 3 creating NFA for  $(ac|ab) \rightarrow (ac|ab)^*$

### 2.2 Subset Construction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

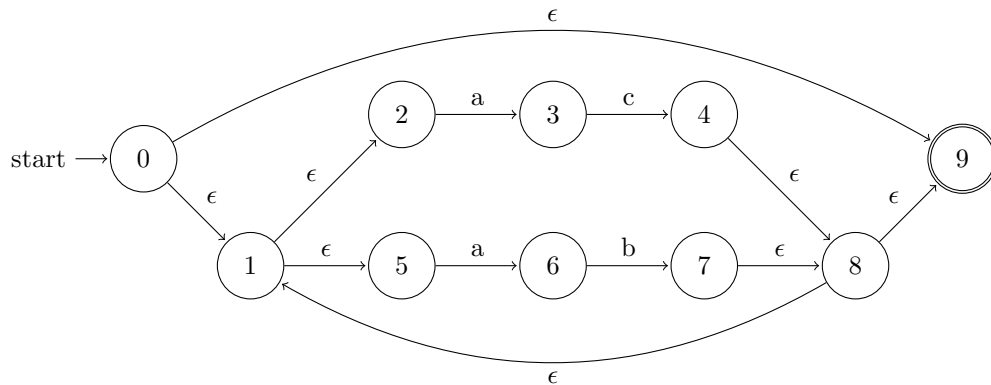
Figure 4: Final non-deterministic finite automaton for  $(ac|ab)^*$ 

Table 1: Subset construction from NFA

State	E-closures
{0}	{0, 1, 2, 5, 8, 9}
{1}	{1, 2, 5, 8, 9}
{2}	{2}
{3}	{3}
{4}	{4, 8, 9, 1, 2, 5}
{5}	{5}
{6}	{6}
{7}	{7, 8, 9, 1, 2, 5}
{8}	{8, 9, 1, 2, 5}
{9}	{9}

## 2.3 Hopcroft's Algorithm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## 2.4 Direct-coded Scanner

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend,

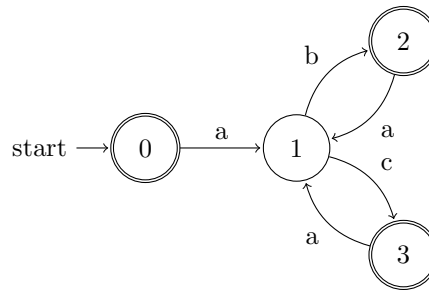


Figure 5: Deterministic finite automaton for  $(ac|ab)^*$

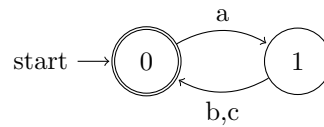


Figure 6: Minimized deterministic finite automaton for  $(ac|ab)^*$

sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```

#include <iostream>
using namespace std;
class Phone {
public:
    double cost;
    int slots;
};
int main() {
    Phone Y6;
    Phone Y7;

    Y6.cost = 100.0;
    Y6.slots = 2;

    Y7.cost = 200.0;
    Y7.slots = 2;
    cout << "Cost_of_Huawei_Y6:" << Y6.cost << endl;
    cout << "Cost_of_Huawei_Y7:" << Y7.cost << endl;

    cout << "Number_of_card_slots_for_Huawei_Y6:" << Y6.slots << endl;
    cout << "Number_of_card_slots_for_Huawei_Y7:" << Y7.slots << endl;

    return 0;
}

```