Graph Theory Research

Conor Shortt

March 2020

Contents

1 Introduction

This document contains the research I have done as part of my Graph Theory Project. Our project involves writing a program in Python to execute regular expressions on strings using an algorithm known as Thompson's construction, named after the well-known computer scientist Ken Thompson.
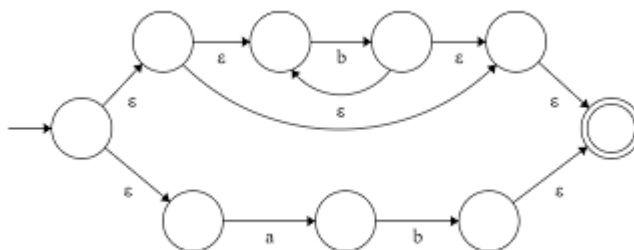

Figure 1: NFA Example for regex $a.b|b*$

1.1 GitHub Repository

· https://github.com/conorshortt123/graphtheory-project-repo

2 Thompson's Construction

Figure 2: Ken Thompson

[1][4][3] Thompson's construction algorithm, is a way of converting a regular expression into equivalent non-deterministic finite automaton. NFA's are used to match strings against regular expressions. This algorithm is credited to Ken Thompson but can also be called the McNaughton-Yamada-Thompson algorithm.

Regex and NFA's are two representations of formal languages. (A formal language in computer science consists of words whose letters are taken from an alphabet and are well formed according to a specific set of rules.) Text processors use regular expressions to describe advanced search patterns, NFA's are better suited for execution using a computer, therefore Thompson's algorithm has a practical interest in relation to executing regular expressions.
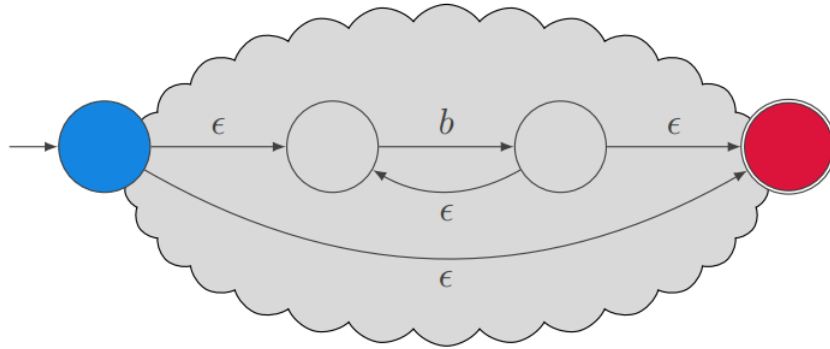
2.1 Example NFA's

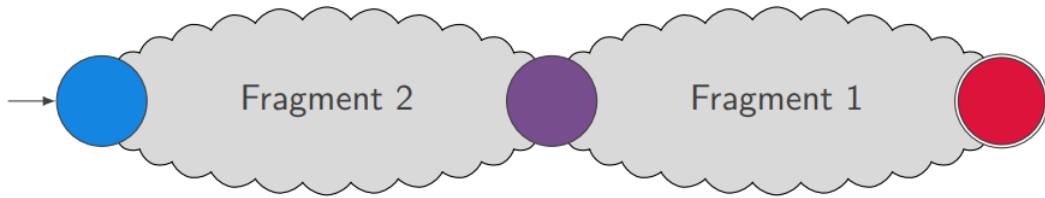Figure 3: NFA Example for regex $b*$

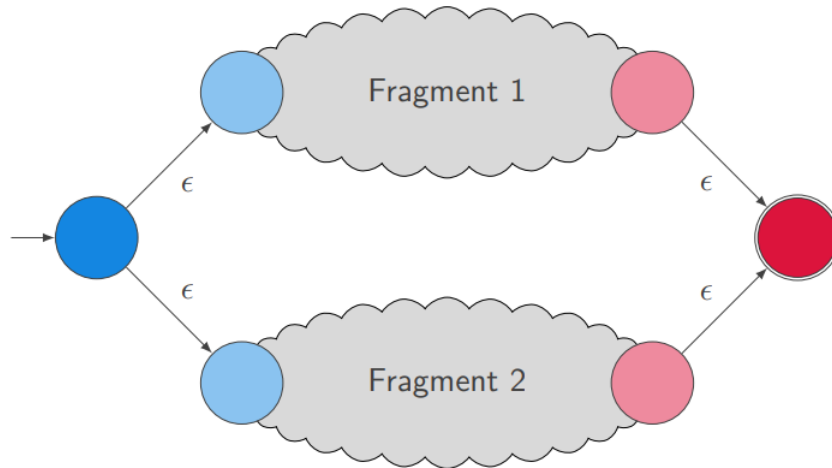Figure 4: NFA Example for a concatenation regex

Figure 5: NFA Example for a Union regex

These are examples of NFA fragments. NFA fragments can be combined together to form larger NFA's.

Figure 3:

As you can see in figure 3, this is an example of an NFA fragment for the Kleene star. The Kleene star is a unary operator, it is widely used for regular expressions, which is the context in which it was introduced by Stephen Kleene to characterize certain automata, where it means "zero or more".

Figure 4:

In figure 4 is an example of an NFA fragment for a concatenation regular expression. Concatenation combines fragments so the NFA can use multiple operators.

Figure 5:

In figure 5 is an example of an NFA fragment for a union regular expression. The union operator acts as an OR gate, which will accept one of two different inputs.

There are other operators such as the question mark operator, plus operator, caret operator etc. For the purposes of my particular project I included five operators $(*.|?+)$. The dot operator in my project is not the standard dot operator used in regular expression engines, it is generally used to match any character bar empty strings, but I used it for concatenation.

3 Regular Expression Operators

Match-self Operator
This operator matches the character itself, for example operator 'A' will match 'A' and only 'A'.

Concatenation Operator
Concatenates two fragments with each other. Generally, concatenation doesn't have a special operator character, you just put b after a to match the string "ab", but in my python program concatenation is done using the special character '.'

Kleene Star Operator
This operator repeats the preceding character zero, one, or many times. Therefore this operator will accept an empty string. The regular expression $a*$ will accept strings ", 'a', 'aaaaa' and so on.

Plus Operator
This operator repeats the preceding character like the Kleene star, but only one or more times, therefore this operator will not accept the empty string. [2]

References

[1] Ian McLoughlin. https://github.com/ianmcloughlin.

[2] MIT Common Operators. http://web.mit.edu/gnu/doc/html/regex$_3$.html.

[3] Graph Theory. https://learnonline.gmit.ie/course/view.php?id=1599.

[4] Wikipedia. https://en.wikipedia.org/wiki/thompson