Regular Expression

Course Code:CSC3220 Course Title: Compiler Design

Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:		**Week No:**		**Semester:**	
Lecturer:	*Md Masum Billah, billah.masumcu@aiub.edu*				

Lecture Outline

1. Definition of a Regular Expression
2. Rules of a Regular Expression
3. Examples
4. Exercises

Objectives and Outcomes

Objectives:

- Understand the basic concept of Regular expression
- Understand the regular expression algorithm

Outcome:

- Students should be able to design the nondeterministic finite automate from regular expression.
- Students should be able to know the applications of a regular expression.

Regular Expression

Definition: A sequence of symbols and characters expressing a string or pattern to be searched for within a longer piece of text.

Another words to say a regular expression is a method used in programming for pattern matching. Regular expressions provide a flexible and concise means to match strings of text.

The regular expressions are built recursively out of smaller regular expressions, using some rules.
Each regular expression r denotes a language L(r), which is also defined recursively from the languages denoted by r ' s subexpressions.

Regular Expression

The regular expressions are built recursively out of smaller regular expressions, using some rules.

Each regular expression r denotes a language L(r), which is also defined recursively from the languages denoted by r ' s subexpressions.

Rules of Regular Expression

Here are the rules that define the regular expressions over some alphabet £ and the languages that those expressions denote.

- Basis
- Induction
- Precedence

Rules of Regular Expression

BASIS: There are two rules that form the basis:

- E is a regular expression, and *L(E)* is {E}, that is, the language whose sole member is the empty string.

- If *a* is a symbol in E, then **a** is a regular expression, and L**(a) =** *{a},* that is, the language with one string, of length one, with *a* in its one position. Here italics is used for symbols, and boldface for their corresponding regular expression.

Rules of Regular Expression

INDUCTION: There are four parts to the induction. Suppose r and *s* are regular expressions denoting languages *L(r)* and *L(s),* respectively.

- (r)|(s) is a regular expression denoting the language *L(r)* U *L(s).*
- (r)(s) is a regular expression denoting the language *L(r)L(s).*
- (r)* is a regular expression denoting (L(r))*.
- (r) is a regular expression denoting *L(r).*The last rule says that we can add additional pairs of parentheses around expressions without changing the language they denote.

Example of a Regular expression

Let E = {a, b}.

- 1. The regular expression **a|b** denotes the language *{a, b}.*
- 2. **(a|b)(a|b)** denotes *{aa, ab, ba, bb},* the language of all strings of length two over the alphabet E.
- Another regular expression for the same language is **aa|ab|ba|bb.**
- 3. **a*** denotes the language consisting of all strings of zero or more a's, that is, { E, *a, aa, aaa, . . . }.*

Example of a Regular expression

Let E = {a, b}.

- 4. **(a|b)*** denotes the set of all strings consisting of zero or more instances of *a* or *b,* that is, all strings of a's and b's: {E *,a, b,aa, ab, ba, bb,aaa,...}.*

- Another regular expression for the same language is **(a*b*)*.**

- **a|a*b** denotes the language *{a, b, ab, aab, aaab,...}*, that is, the string *a* and all strings consisting of zero or more a's and ending in *b.*

Operations of a Regular expression

Operations:

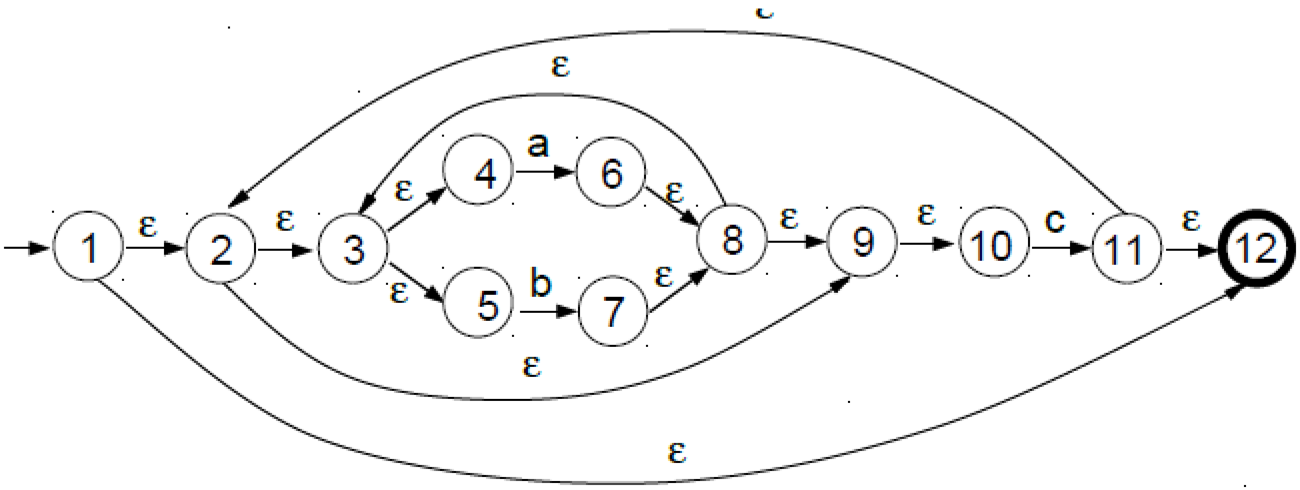
The various operations on languages are:

- Union of two languages L and M is written as
  L U M = {s | s is in L or s is in M}
- Concatenation of two languages L and M is written as
  LM = {st | s is in L and t is in M}
- The Kleene Closure of a language L is written as
  L* = Zero or more occurrence of language L.

Regular Expression To NFA

Outline the NFA generated by the construction of Thompson relevant to the following regular expression:

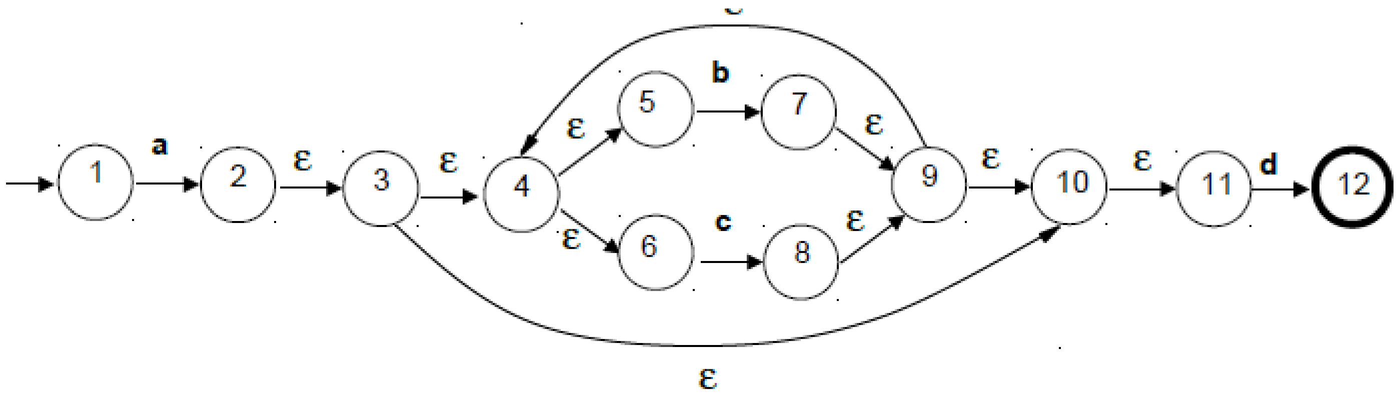
Example: ((a | b)*c)*



Regular Expression To NFA

By means of the construction of Thompson, outline the NFA relevant to the following regular expression:

Example: a (b | c)*d



Regular Expression To NFA

By means of the construction of Thompson, outline the NFA relevant to the following regular expression:

Example: a (b | c)*d

Class Exercises

1. (aUb)*abc

1. (abUbc(abUc)*)*

Lecture References

A. Aho, R. Sethi and J. Ullman, ***Compilers: Principles, Techniques and Tools*** *(*The Dragon Book*),* [Second Edition]

References

1. A. Aho, R. Sethi and J. Ullman, ***Compilers: Principles, Techniques and Tools****(*The Dragon Book*)*, [Second Edition]

2. Principles of Compiler Design (2nd Revised Edition 2009) A. A. Puntambekar

3. Basics of Compiler Design Torben Mogensen