CSE211 – Formal Languages and Automata Theory

U1L13 – Regular Expressions Part 2

Dr. P. Saravanan
School of Computing
SASTRA Deemed University

Agenda

- Recap of previous class
- Finding RE: Examples
- Applications of RE
- Regular Expression an Automata
- Defining language by RE
- Precedence of operators in RE
- Some equalities

Recap: Examples

1. $01^* = \{0, 01, 011, 0111, \ldots..\}$
2. $(01^*)(01) = \{001, 0101, 01101, 011101, \ldots..\}$
3. $(0+1)^*$
4. $(0+1)^*01(0+1)^*$
5. $((0+1)(0+1)+(0+1)(0+1)(0+1))^*$
6. $((0+1)(0+1))^*+((0+1)(0+1)(0+1))^*$
7. $(1+01+001)^*(\varepsilon+0+00)$

Some more Examples

- Construct a RE over $\Sigma = \{0,1\}$ that represents
  - All strings that have two consecutive 0s.

$$(0+1)^*00(0+1)^*$$

  - All strings except those with two consecutive 0s.

$$(1^*01)^*1^* + (1^*01)^*1^*0$$

  - All strings with an even number of 0s.

$$(1^*01^*01^*)^*$$

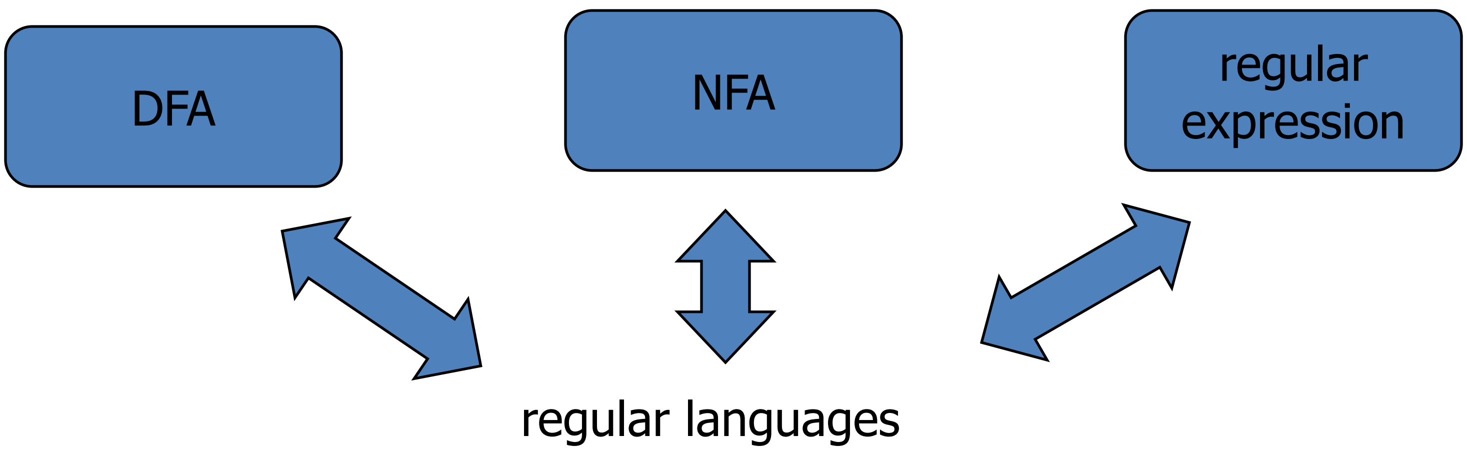
Applications of RE

- Used as commands for finding strings in Web browsers or text-formatting systems (such as UNIX grep commands)
- Used as lexical analyzer generator (such Lex or Flex)
- A lexical analyzer breaks source programs into "tokens" (keywords, identifiers, signs, …)

Regular Expression an Automata

- Main theorem for regular languages

A language is regular if and only if it is the language of some DFA



Defining a Language by RE

- A regular expression **(**RE**)** *E* and its corresponding language *L*(*E*) are defined recursively in the following way ---
  - Constants ε and ϕ are RE's defining languages $\{\varepsilon\}$ and ϕ, respectively, which are expressed as $L(\varepsilon) = \{\varepsilon\}$, $L(\phi) = \phi$
  - If *a* is a symbol, then **a** is an RE defining the language $\{a\}$ which may be expressed as $L(\mathbf{a}) = \{a\}$
  - A variable like *L* (capitalized and italic) represents any language.

Defining a Language by RE

- Given two RE's E and F, then we have the following more complicated RE's.
- *Union* —
  - $E + F$ is an RE such that $L(E + F) = L(E) \cup L(F)$
- *Concatenation* —
  - EF is an RE such that $L(EF) = L(E)L(F)$
- *Closure* —
  - E^* is an RE such that $L(E^*) = (L(E))^*$
- *Parenthesization* —
  - (E) is an RE such that $L((E)) = L(E)$

Defining a Language by RE

- An RE defining a language of strings of alternating 0’s and 1’s is one of the two below:
  - $(01)^* + (10)^* + 0(10)^* + 1(01)^*$
  - $(\varepsilon + 1)(01)^*(\varepsilon + 0)$

Precedence of RE operators

- Highest "*" (closure)
- Next "." (concatenation) (left to right)
- Last "+" (union) (left to right)
- Use parentheses anywhere to resolve ambiguity
- The following are three ways to interpret the RE **01^*** + **1** if parentheses are not used:
  - **$(0(1^*)) + 1$** by precedence above;
  - **$(01)^* + 1$** (another meaning);
  - **$0(1^* + 1)$** (a third meaning).

Some equalities (*R* is an RE)

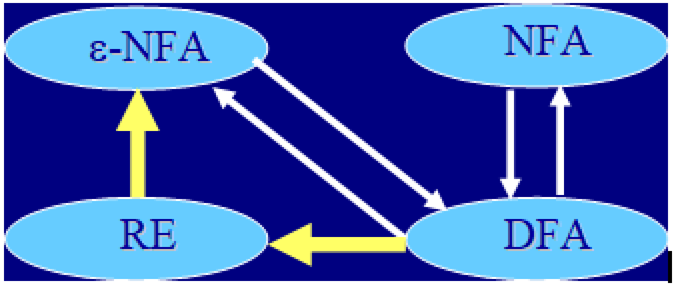
- $\phi R = R\,\phi = \phi$ (so ϕ = *annihilator* for concatenation);
- $\phi + R = R + \phi = R$ (so ϕ = *identity* for union);
- $\varepsilon\, R = R\, \varepsilon = R$ (so ε = *identity* for concatenation);
- $(\varepsilon + \mathbf{a})^* = \mathbf{a}^* = (\mathbf{a} + \varepsilon)^*$
- $(\varepsilon + \mathbf{a})\mathbf{a}^* = (\varepsilon\, \mathbf{a}^* + \mathbf{aa}^*) = \mathbf{a}^* + \mathbf{aa}^* = \mathbf{a}^*$
- $\mathbf{a}^*(\varepsilon + \mathbf{a}) = (\mathbf{a}^*\, \varepsilon + \mathbf{a}^*\mathbf{a}) = \mathbf{a}^* + \mathbf{a}^*\mathbf{a} = \mathbf{a}^*$

RE Examples

(a+b)*	Set of strings of a's and b's of any length including the null string. So L = { ε, a, b, aa , ab , bb , ba, aaa…….}
(a+b)*abb	Set of strings of a's and b's ending with the string abb. So L = {abb, aabb, babb, aaabb, ababb, …………..}
(11)*	Set consisting of even number of 1's including empty string, So L= {ε, 11, 1111, 111111, ……….}
(aa)*(bb)*b	Set of strings consisting of even number of a's followed by odd number of b's , so L = {b, aab, aabbb, aabbbbb, aaaab, aaaabbb, …………..}
(aa + ab + ba + bb)*	String of a's and b's of even length can be obtained by concatenating any combination of the strings aa, ab, ba and bb including null, so L = {aa, ab, ba, bb, aaab, aaba, …………..}

FA's & RE's

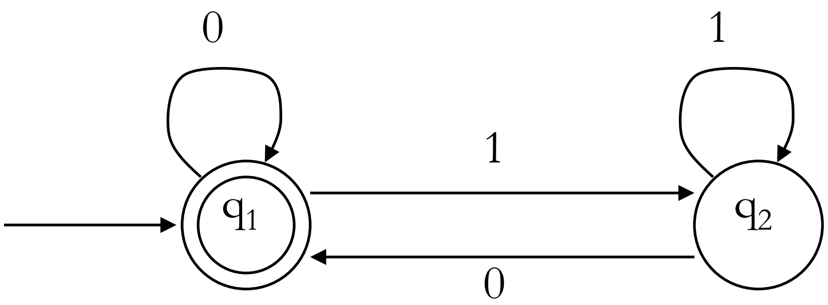
- Theorems :
  - Every language defined by a DFA is also defined by an RE.
  - Every language defined by an RE is also defined by an e-NFA.
- **Relations of theorems**

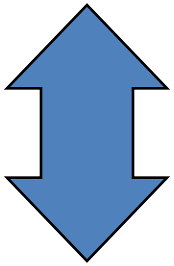


Equivalence Relations among DFA's, NFA's, e-NFA's, and RE's.

Construction of RE: Example

- Construct a regular expression for this DFA:





RE=?

Summary

- Applications of RE
- Regular Expression an Automata
- Defining language by RE
- Precedence of operators in RE
- Some equalities

References

- John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory*, Languages, and Computation, Pearson, 3rd Edition, 2011.
- Peter Linz, An Introduction to Formal Languages and Automata, Jones and Bartle Learning International, United Kingdom, 6th Edition, 2016.

Next Class:

DFA to RE Conversion

THANK YOU.