Regular Expressions and Regular Languages

Regular Expressions

- Regular Expressions are an algebraic way to describe languages.
- Regular Expressions describe exactly the regular languages.
- If E is a regular expression, then L(E) is the regular language it defines.
- A regular expression is built up of simpler regular expressions (using defining rules)
- For each regular expression E, we can create a DFA A such that L(E) = L(A).
- For each a DFA A, we can create a regular expression E such that L(A) = L(E)

Regular Expressions - Definition

Regular expressions over alphabet Σ

	Reg. Expr. E	Language it denotes L(E)
Basis 1:	Φ	$\{\}$
Basis 2:	ϵ	$\{\varepsilon\}$
Basis 3:	$\mathbf{a} \in \Sigma$	$\{a\}$

Note:

$\{a\}$ is the language containing one string, and that string is of length 1.

Regular Expressions - Definition

Induction 1 – or : If E_1 and E_2 are regular expressions, then $\mathbf{E_1+E_2}$ is a regular expression, and $L(E_1+E_2) = L(E_1) \cup L(E_2)$.

Induction 2 – concatenation: If E_1 and E_2 are regular expressions, then $\mathbf{E_1E_2}$ is a regular expression, and $L(E_1E_2) = L(E_1)L(E_2)$ where $L(E_1)L(E_2)$ is the set of strings wx such that w is in $L(E_1)$ and x is in $L(E_2)$.

Induction 3 – Kleene Closure: If E is a regular expression, then **E*** is a regular expression, and $L(\mathbf{E^*}) = (L(E))^*$.

Induction 4 – Pranteheses: If E is a regular expression, then **(E)** is a regular expression, and $L(\ \mathbf{(E)}\ ) = L(E)$.

Regular Expressions - Parentheses

- Parentheses may be used wherever needed to influence the grouping of operators.
- We may remove parentheses by using precedence and associativity rules.

Operator	*Precedence*	*Associativity*
*	highest	
concatenation	next	left associative
+	lowest	left associative

ab^*+c means **$(a((b)^*))+(c)$**

Regular Expressions - Examples

Alphabet $\Sigma = \{0,1\}$

- $L(\mathbf{01}) = \{01\}$. $\quad L(\mathbf{01}) = L(\mathbf{0})\, L(\mathbf{1}) = \{0\}\{1\} = \{01\}$
- $L(\mathbf{01+0}) = \{01, 0\}$. $\quad L(\mathbf{01+0}) = L(\mathbf{01}) \cup L(\mathbf{0}) = (L(\mathbf{0})\, L(\mathbf{1})) \cup L(\mathbf{0})$

  $= (\{0\}\{1\}) \cup \{0\} = \{01\} \cup \{0\} = \{01,0\}$
- $L(\mathbf{0(1+0)}) = \{01, 00\}$.
  - Note order of precedence of operators.
- $L(\mathbf{0^*}) = \{\epsilon, 0, 00, 000, \dots\}$.
- $L(\mathbf{(0+10)^*(\epsilon+1)})$ = all strings of 0's and 1's without two consecutive 1's.
- $L(\mathbf{(0+1)(0+1)}) = \{00,01,10,11\}$
- $L(\mathbf{(0+1)^*})$ = all strings with 0 and 1, including the empty string

Regular Expressions -Examples

All strings of 0’s and 1’s starting with 0 and ending with 1

$0(0+1)^*1$

All strings of 0’s and 1’s with even number of 0’s

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

All strings of 0’s and 1’s with at least two consecutive 0’s

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

All strings of 0’s and 1’s without two consecutive 0’s

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

Regular Expressions	Regular Set
(0 + 10*)	
(0*10*)	
(0 + ε)(1 + ε)	
(a+b)*	
(a+b)*abb	
(11)*	
(aa)*(bb)*b	
(aa + ab + ba + bb)*	

Regular Expressions	Regular Set
(0 + 10*)	L = { 0, 1, 10, 100, 1000, 10000, … }
(0*10*)	
(0 + ε)(1 + ε)	
(a+b)*	
(a+b)*abb	
(11)*	
(aa)*(bb)*b	
(aa + ab + ba + bb)*	

Regular Expressions	Regular Set
(0 + 10*)	L = { 0, 1, 10, 100, 1000, 10000, … }
(0*10*)	L = {1, 01, 10, 010, 0010, …}
(0 + ε)(1 + ε)	
(a+b)*	
(a+b)*abb	
(11)*	
(aa)*(bb)*b	
(aa + ab + ba + bb)*	

Regular Expressions	Regular Set
$(0 + 10^*)$	$L = \{ 0, 1, 10, 100, 1000, 10000, \ldots \}$
(0^*10^*)	$L = \{1, 01, 10, 010, 0010, \ldots\}$
$(0 + \varepsilon)(1 + \varepsilon)$	$L = \{\varepsilon, 0, 1, 01\}$
$(a+b)^*$	
$(a+b)^*abb$	
$(11)^*$	
$(aa)^*(bb)^*b$	
$(aa + ab + ba + bb)^*$	

Regular Expressions	Regular Set
(0 + 10*)	L = { 0, 1, 10, 100, 1000, 10000, … }
(0*10*)	L = {1, 01, 10, 010, 0010, …}
(0 + ε)(1 + ε)	L = {ε, 0, 1, 01}
(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	
(11)*	
(aa)*(bb)*b	
(aa + ab + ba + bb)*	

Regular Expressions	Regular Set
$(0 + 10^*)$	L = { 0, 1, 10, 100, 1000, 10000, … }
(0^*10^*)	L = {1, 01, 10, 010, 0010, …}
$(0 + \varepsilon)(1 + \varepsilon)$	L = {ε, 0, 1, 01}
$(a+b)^*$	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, …………..}
$(11)^*$	
$(aa)^*(bb)^*b$	
$(aa + ab + ba + bb)^*$	

Regular Expressions	Regular Set
(0 + 10*)	L = { 0, 1, 10, 100, 1000, 10000, … }
(0*10*)	L = {1, 01, 10, 010, 0010, …}
(0 + ε)(1 + ε)	L = {ε, 0, 1, 01}
(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)*	
(aa)*(bb)*b	
(aa + ab + ba + bb)*	

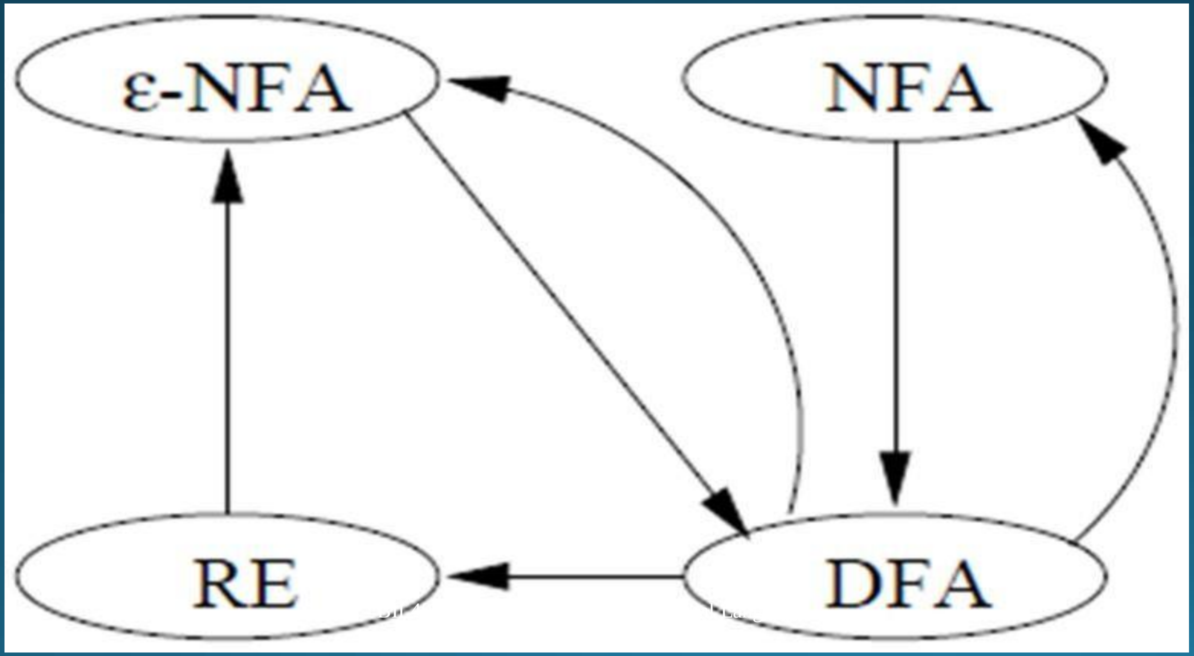
Regular Expressions	Regular Set
(0 + 10*)	L = { 0, 1, 10, 100, 1000, 10000, … }
(0*10*)	L = {1, 01, 10, 010, 0010, …}
(0 + ε)(1 + ε)	L = {ε, 0, 1, 01}
(a+b)*	Set of strings of a's and b's of any length including the null string. So L = { ε, a, b, aa , ab , bb , ba, aaa…….}
(a+b)*abb	Set of strings of 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	
(aa + ab + ba + bb)*	

Regular Expressions	Regular Set
(0 + 10*)	L = { 0, 1, 10, 100, 1000, 10000, … }
(0*10*)	L = {1, 01, 10, 010, 0010, …}
(0 + ε)(1 + ε)	L = {ε, 0, 1, 01}
(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)*	All Strings with even length, including null, so L = {aa, ab, ba, bb, aaab, aaba, aabb, …………..}

Can we construct a DFA/NFA for a regular expression?

Equivalence of FA's and Regular Expressions

- We have already shown that DFA's, NFA's, and ε-NFA's all are equivalent.
- To show FA's equivalent to regular expressions we need to establish that
  1. For every DFA A we can construct a regular expression R, s.t. L(R) = L(A).
  2. For every regular expression R there is a ε-NFA , DFA , s.t. L(A) = L(R).



Examples (Assignment)

Find regular expressions for the following languages over {0,1}:

– The set of all strings with an even number of 0's

– The set of all strings of even length

– The set of all strings that begin with 110

– The set of all strings containing exactly three 1's

– The set of all strings divisible by 2

– The set of strings where third last symbol is 1