CPS 140 - Mathematical Foundations of CS
Dr. Susan Rodger
Section: FA and Regular Expressions (Ch. 2.3) (handout)

Review Regular Expressions

Method to represent strings in a language

$+$	union (or)
$\circ$	concatenation (AND) (can omit)
$*$	star-closure (repeat 0 or more times)

Example:

$(a+b)^* \circ a \circ (a+b)^* = (a+b)^* a (a+b)^*$

Closure Properties

A set is closed over an operation if

$L_1, L_2 \in$ class
L_1 op $L_2 = L_3$
$\Rightarrow L_3 \in$ class

Example

L_1={x | x is a positive even integer}

L is closed under

addition?
multiplication?
subtraction?
division?

Example

L_2={x | x is a positive odd integer}

L is closed under

addition?
multiplication?
subtraction?
division?

Closure of Regular Languages

Theorem 2.3.1 If L_1 and L_2 are regular languages, then

$L_1 \cup L_2$
$L_1 L_2$
L_1^*
$\bar{L}_1$
$L_1 \cap L_2$

are regular languages.

Proof(sketch)

Union $M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$, $M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2)$

Construct M, L(M)=L(M_1)∪L(M_2)

Concatenation $M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$, $M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2)$

Construct M, L(M)=L(M_1)∘L(M_2)

Kleene Star

$M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$

Construct M, L(M)=L(M_1)*

Complementation:

$M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$

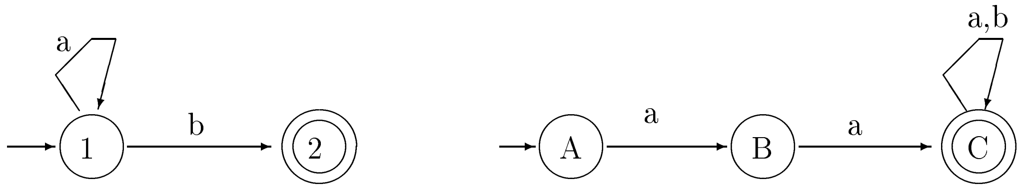
Construct M, L(M)=$L(\bar{M}_1)$

Intersection

$M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$, $M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2)$

Construct M, L(M)=L(M_1)∩L(M_2)

Example:



Regular languages are closed under

reversal	L^R
difference	L_1-L_2
right quotient	L_1/L_2

Right quotient

Def: $L_1/L_2 = \{x | xy \in L_1 \text{ for some } y \in L_2\}$

Example:

$L_1 = \{a^*b^* \cup b^*a^*\}$
$L_2 = \{b^n | n \text{ is even}, n > 0\}$
$L_1/L_2 =$

Equivalence of DFA and R.E.

Definition A language L is regular if it can be described by a regular expression.

Theorem 2.3.3 A language is regular if and only if it is accepted by a finite automaton.

- Proof Part 1 ($\Rightarrow$):

  Let r be a R.E., then $\exists$ NFA M s.t. L(M)=L(r).

  $\emptyset$

  $\{\lambda\}$

  $\{a\}$

  Suppose r and s are R.E.

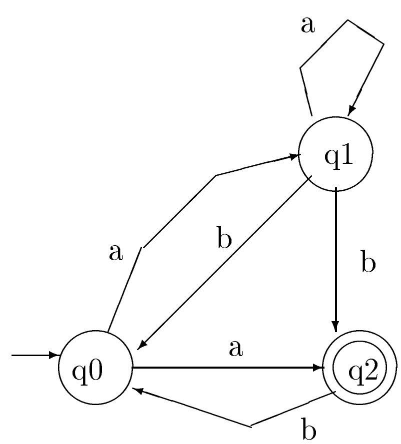
  1. r+s
  2. r$\circ$s
  3. r^*

Example

$ab^* + a$

- Proof Part 2 ($\Leftarrow$):
  Given an NFA M $\exists$ R.E. r s.t. L(M)=L(r).

Example:



Grammar G=(V,Σ,R,S)

V	variables (nonterminals)
Σ	terminals
R	rules (productions)
S	start symbol

Right-linear grammar:

all productions of form
$A \rightarrow xB$
$A \rightarrow x$
where A,B $\in$ V, x $\in \Sigma^*$

Left-linear grammar:

all productions of form
$A \rightarrow Bx$
$A \rightarrow x$
where A,B $\in$ V, x $\in \Sigma^*$

Definition:

A regular grammar is a right-linear or left-linear grammar.

Example 1:

G=({S},{a,b},R,S), R=
S $\rightarrow$ abS
S $\rightarrow$ λ
S $\rightarrow$ Sab

Example 2:

G=({S,B},{a,b},R,S), R=
S $\rightarrow$ aB | bS | λ
B $\rightarrow$ aS | bB

Theorem: L is a regular language iff $\exists$ regular grammar G s.t. L=L(G).

Outline of proof:

($\Longleftarrow$) Given a regular grammar G
Construct NFA M
Show L(G)=L(M)
($\Longrightarrow$) Given a regular language
$\exists$ DFA M s.t. L=L(M)
Construct reg. grammar G
Show L(G) = L(M)

Proof of Theorem:

($\Longleftarrow$) Given a regular grammar G
G=(V,Σ,R,S)
V=$\{V_0, V_1, \ldots, V_y\}$
Σ=$\{v_o, v_1, \ldots, v_z\}$
S=V_0
Assume G is right-linear
(left-linear case similar).
Construct NFA M s.t. L(G)=L(M)
If w$\in$L(G), w=$v_1 v_2 \ldots v_k$

M=(V,Σ,δ,V_0,F)
V_0 is the start (initial) state
For each production, $V_i \rightarrow aV_j$,

For each production, $V_i \rightarrow a$,

Show L(G)=L(M)

Thus, given R.G. G,

L(G) is regular

($\Longrightarrow$) Given a regular language L

$\exists$ DFA M s.t. L=L(M)

M=(K,Σ,δ,q_0, F)

K=$\{q_0, q_1, \ldots, q_n\}$

$\Sigma = \{a_1, a_2, \ldots, a_m\}$

Construct R.G. G s.t. L(G) = L(M)

G=(K,Σ,R,q_0)

if $\delta(q_i, a_j)$=q_k then

if q_k $\in$F then

Show w $\in$L(M) $\Longleftrightarrow$ w $\in$ L(G)

Thus, L(G)=L(M).

QED.

Example

G=({S,B},{a,b},R,S), R=

S $\rightarrow$ aB | bS | λ

B $\rightarrow$ aS | bB

Example: