CS 352, Jens Palsberg   Aug 29, 2000

Homework #1
Regular Languages

Question #1

Write regular expressions for each of the following:

a) String over the alphabet {a,b,c} with an odd number of a's

Solution: (b|c)*.a. (b|c)* . ((b|c)*.a.(b|c)*.a.(b|c)*)*

b) Binary numbers multiple of 2 and representing a decimal number greater than or equal to 8

Solution: We assume that there might be a leading zero

0*.1 .(0|1)* .(0|1).(0|1).0

c) Binary numbers greater than 110011

Solution: We assume that there might be a leading zero.

0*.1.1.0.1.(0|1).(0|1) |
0*.1.1.1.(0|1).(0|1).(0|1) |
0*.1.(0|1)* .(0|1). (0|1). (0|1). (0|1). (0|1).(0|1)

d) Strings of the kind EPX where E is an integer number, P is a lowercase letter from the alphabet and X is an integer greater than 3 and less than 13. Examples: 143a6, 555b12, etc.

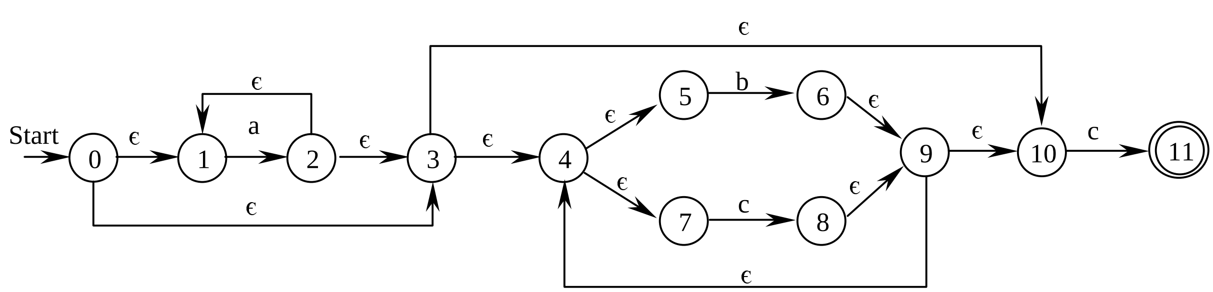
Solution: 0|((- ?).[1-9].[0-9]*).[a-z].((1.[0-2])|[3-9])

Question #2.

Convert the following regular expressions to nondeterministic finite automata.

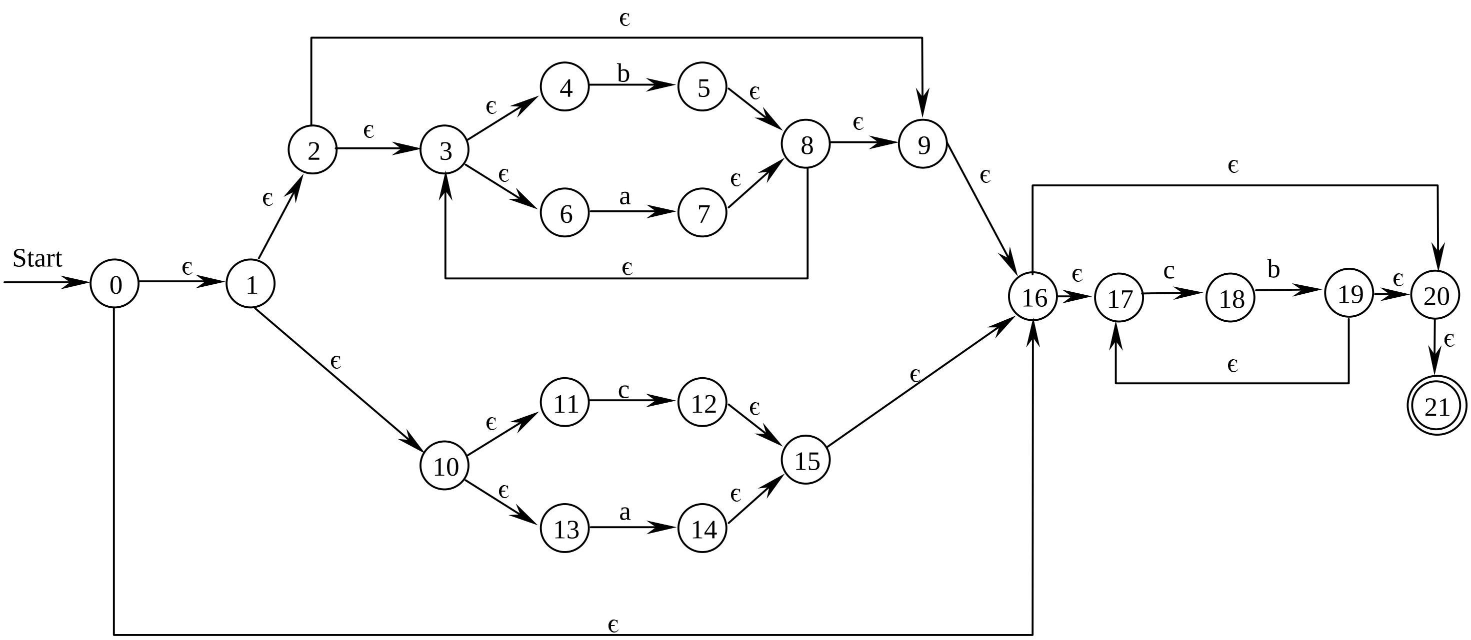
a) a* (b | c)* c

Solution:



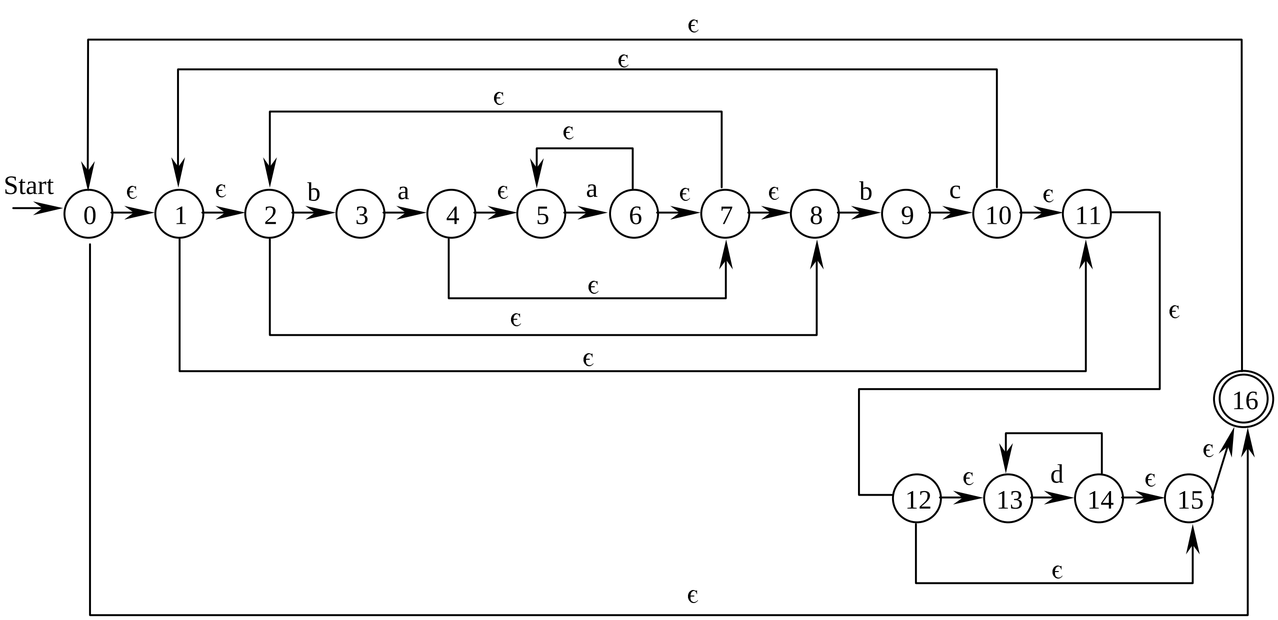
b) ((b|a)*|(c|a))*(cb)*

Solution:



c) (((baa*)*bc)*d*)*

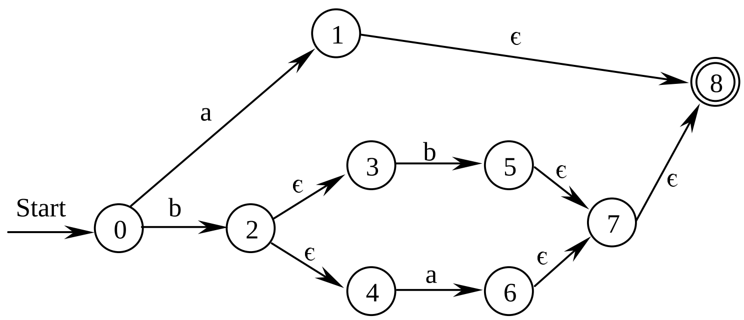
Solution:



Question #3

Convert the following NFA to DFA. Show each **closure** and **edge** in the process. For a) and b) show the **state transition table.**

a)

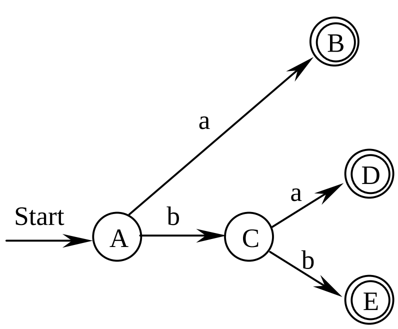


Solution:

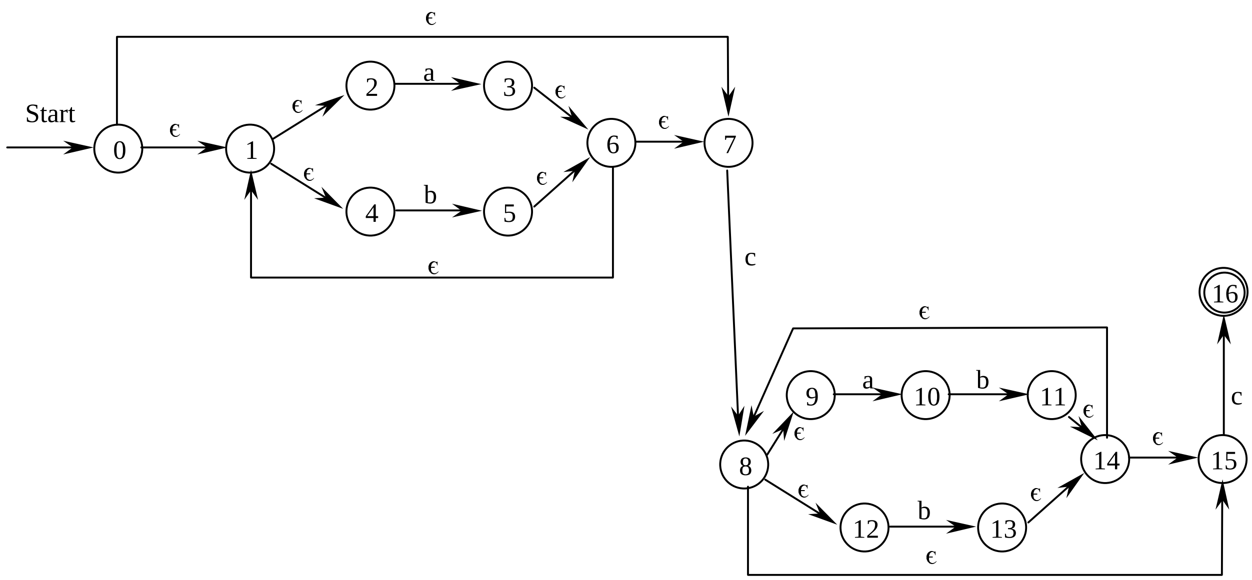
Closure (0)={0}=A
DFAedge(A,a)=closure(edge(A,a))=closure({1})={1,8}=B-accepting
DFAedge(A,b)=closure(edge(A,b))=closure({2})={2,3,4}=C
DFAedge(B,a)=closure(edge(B,a))=closure({})={}
DFAedge(B,b)=closure(edge(B,b))=closure({})={}
DFAedge(C,a)=closure(edge(C,a))=closure({6})={6,7,8}=D-accepting
DFAedge(C,b)=closure(edge(C,b))=closure({5})={5,7,8}=E-accepting
DFAedge(D,a)=closure(edge(D,a))=closure({})={}
DFAedge(D,b)=closure(edge(D,b))=closure({})={}
DFAedge(E,a)=closure(edge(E,a))=closure({})={}
DFAedge(E,b)=closure(edge(E,b))=closure({})={}

<u>State transition table</u>

State	a	b
A	B	C
B	-	-
C	D	E
D	-	-
E	-	-



b)



Solution:

Closure (0)={0,1,2,4,7}=A

DFAedge(A,a)=closure(edge(A,a))=closure({3})={1,2,3,4,6,7}=B

DFAedge(A,b)=closure(edge(A,b))=closure({5})={1,2,4,5,6,7}=C

DFAedge(A,c)=closure(edge(A,c))=closure({8})={8,9,12,15}=D

DFAedge(B,a)=closure(edge(B,a))=closure({3})={1,2,3,4,6,7}=B

DFAedge(B,b)=closure(edge(B,b))=closure({5})={1,2,4,5,6,7}=C

DFAedge(B,c)=closure(edge(B,c))=closure({8})={8,9,12,15}=D

DFAedge(C,a)=closure(edge(C,a))=closure({3})={1,2,3,4,6,7}=B

DFAedge(C,b)=closure(edge(C,b))=closure({5})={1,2,4,5,6,7}=C

DFAedge(C,c)=closure(edge(C,c))=closure({8})={8,9,12,15}=D

DFAedge(D,a)=closure(edge(D,a))=closure({10})={10}=E

DFAedge(D,b)=closure(edge(D,b))=closure({13})={8,9,12,13,14,15}=F

DFAedge(D,c)=closure(edge(D,c))=closure({16})={16}=G-accepting

DFAedge(E,a)=closure(edge(E,a))=closure({})={}

DFAedge(E,b)=closure(edge(E,b))=closure({11})={8,9,11,12,14,15}=H

DFAedge(E,c)=closure(edge(E,c))=closure({})={}

DFAedge(F,a)=closure(edge(F,a))=closure({10})={10}=E

DFAedge(F,b)=closure(edge(F,b))=closure({13})={8,9,12,13,14,15}=F

DFAedge(F,c)=closure(edge(F,c))=closure({16})={ 16}=G-accepting

DFAedge(G,a)=closure(edge(G,a))=closure({})={}

DFAedge(G,b)=closure(edge(G,b))=closure({})={}

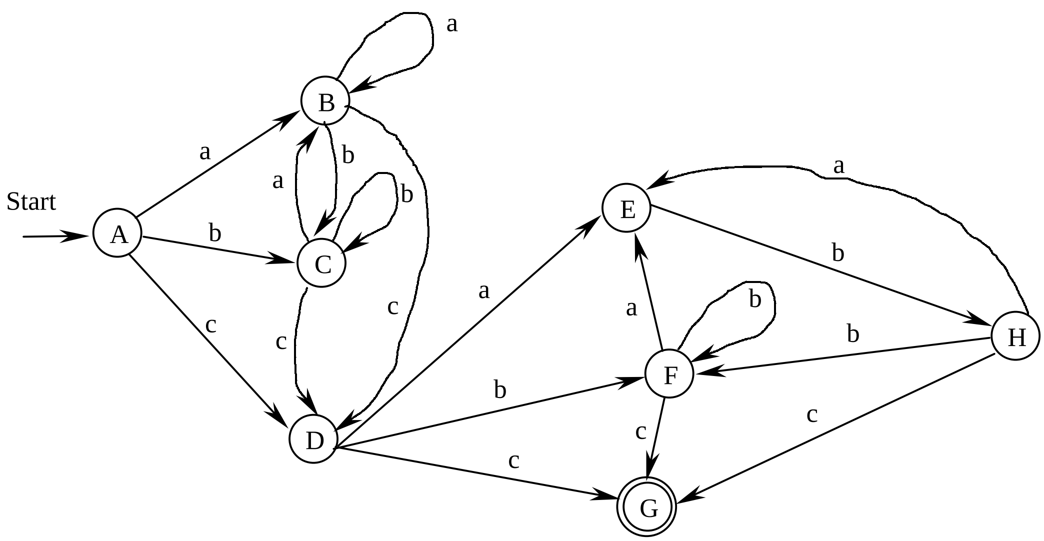
DFAedge(G,c)=closure(edge(G,c))=closure({})={}

DFAedge(H,a)=closure(edge(H,a))=closure({10})={10}=E

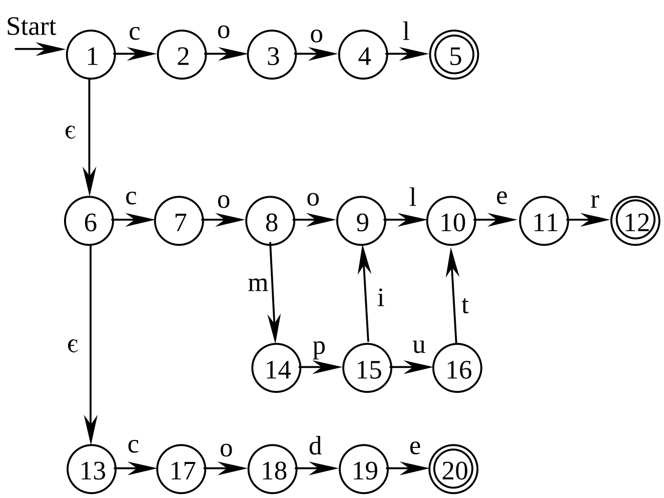
DFAedge(H,b)=closure(edge(H,b))=closure({13})={8,9,12,13,14,15}=F
DFAedge(H,c)=closure(edge(H,c))=closure({16})={ 16}=G-accepting

State transition table

State/Input	a	b	c
A	B	C	D
B	B	C	D
C	B	C	D
D	E	F	G
E	-	H	-
F	E	F	G
G	-	-	-
H	E	F	G



c)



Solution:

Closure(1)={1,6,13}=A
DFAedge(A,c)=closure(edge(A,c))=closure({2,7,17})={2,7,17}=B
DFAedge(B,o)=closure(edge(B,o))=closure({3,8,18})={3,8,18}=C
DFAedge(C,o)=closure(edge(C,o))=closure({4,9})={4,9}=D
DFAedge(C,m)=closure(edge(C,m))=closure({14})={14}=E
DFAedge(C,d)=closure(edge(C,d))=closure({19})={19}=F
DFAedge(D,l)=closure(edge(D,l))=closure({5,10})={5,10}=G-accepting
DFAedge(E,p)=closure(edge(E,p))=closure({15})={15}=H
DFAedge(F,e)=closure(edge(F,e))=closure({20})={20}=I-accepting
DFAedge(G,e)=closure(edge(G,e))=closure({11})={11}=J
DFAedge(H,i)=closure(edge(H,i))=closure({9})={9}=K
DFAedge(H,u)=closure(edge(H,u))=closure({16})={16}=L
DFAedge(J,r)=closure(edge(J,r))=closure({12})={12}=M-accepting
DFAedge(K,l)=closure(edge(K,l))=closure({10})={10}=N
DFAedge(L,t)=closure(edge(L,t))=closure({10})={10}=N
DFAedge(N,e)=closure(edge(N,e))=closure({11})={11}=J