Assignment 2 – Lexical Analyser

SEG2106 – Software Construction

Question 1 – Regular Expressions (30 points – 6 points each)

Find regular expressions that define the following languages:

1. XML opening and closing tags where the opening tags may include attributes. The form of such tags is given by the following examples: <name-x attribute-1="some string" attribute-2="xyz"> or <funny laugh123="go" s-t-o-p=";>"/> or </name> . The alphabet is composed of the sets alpha, digit and {-,<,>,/, ", =, ; }.

   Regular expression to find the opening tag:

   (<\s*[\w\-]+\s*([\w\-]+="[^"]+"\s*)*\/?>)

   Regular expression to find the closing tag:

   (<\/\s*[\w\-]+\s*>)

   Putting the two together:

   (<\s*[\w\-]+\s*([\w-]+=".+")*\s*\/?>)|(<\/\s*[\w\-]+\s*>)

2. All strings over the alphabet {a, b} that do not contain the substring aaa.
   a{0,2}(b+a{0,2}b*)*

3. All strings over the alphabet {a, b} for which the number of "a" is a multiple of 3 (including zero).

   (b*(ab*ab*a)*)*

4. All binary numbers greater than 10111.

   We assume that there might be a leading zero.

   (0*11(0|1)(0|1)(0|1)**) |**

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

5. All strings of the kind EPX where E is an integer number, P is a lowercase letter and X is an integer greater than 3 and less than 13. Examples: 143a6, 555b12, etc.

   To match the integer:

   (0|((-?)[1-9][0-9]*))

   To match the lower case letter:

   [a-z]

   To match an integer greater than 3 and less than 13:

   ((1[0-2])|[3-9])

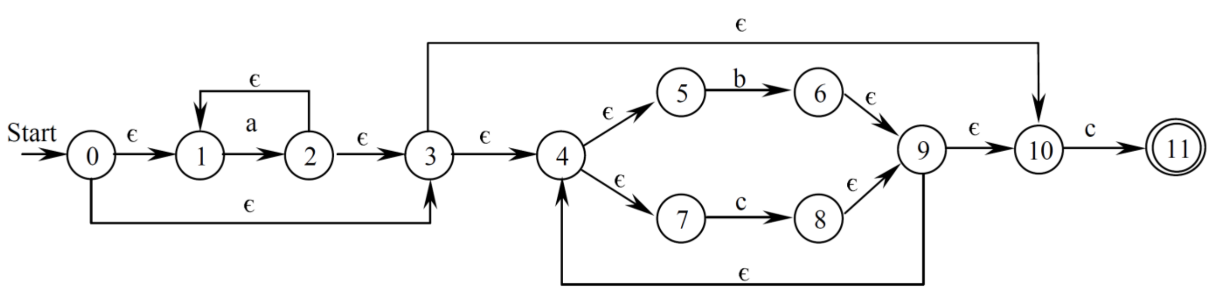
   Putting it all together:

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

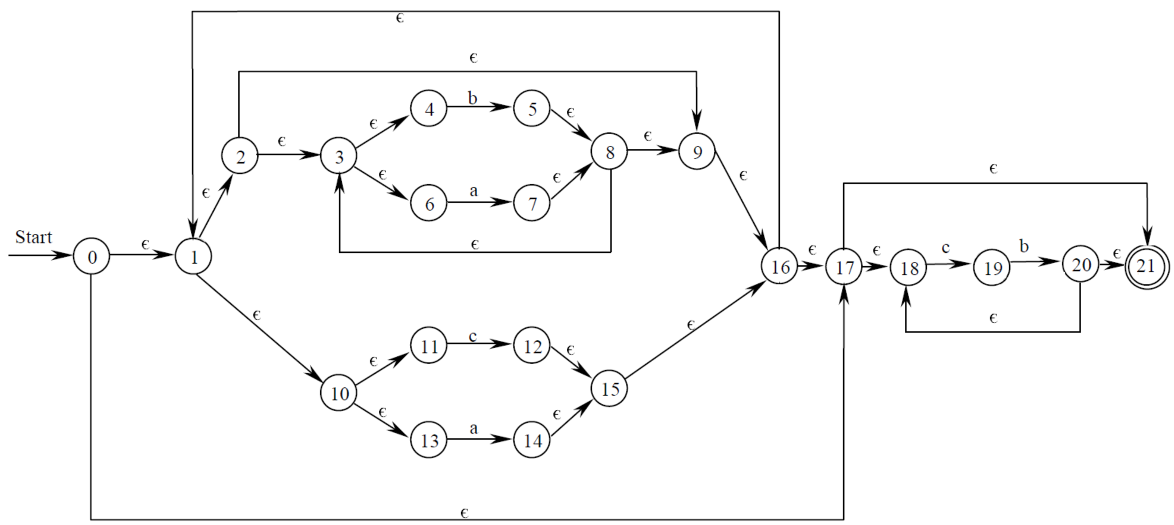
Question 2 – Non-Deterministic Finite Automata (30 points – 10 points each)

Convert the following regular expressions to Non-deterministic Finite Automata (NFA):

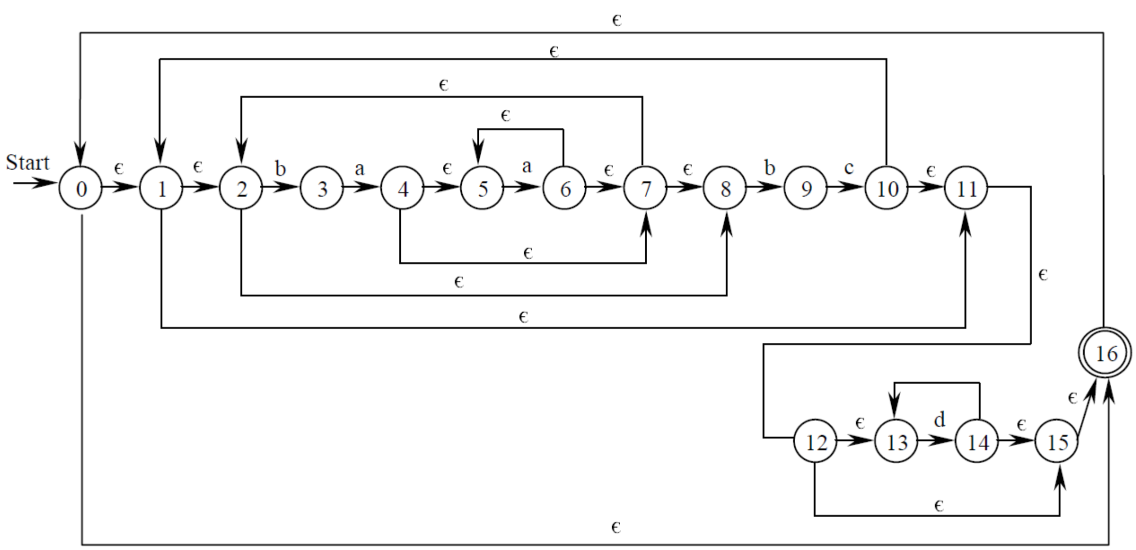
a) a* (b | c)* c



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



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



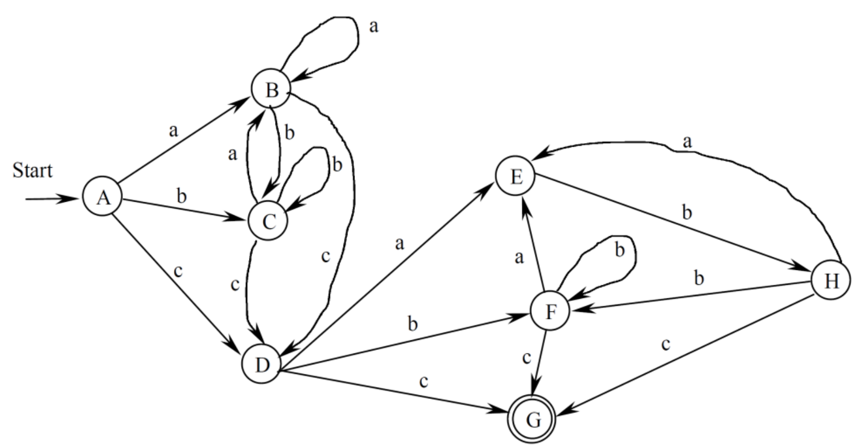
Question 3 – NFA to DFA Conversion (40 points – 20 points each)

Convert the following Non-deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA) using the subset construction algorithm. Show every ε-closure(s), ε-closure(T), and move(T,a) calculation (as we have done in class).

a)

ε-closure (0)={0,1,2,4,7}=A
ε-closure(move(A,a))=ε-closure({3})={1,2,3,4,6,7}=B

ε-closure(move(A,b))=ε-closure({5})={1,2,4,5,6,7}=C
ε-closure(move(A,c))=ε-closure({8})={8,9,12,15}=D
ε-closure(move(B,a))=ε-closure({3})={1,2,3,4,6,7}=B
ε-closure(move(B,b))=ε-closure({5})={1,2,4,5,6,7}=C
ε-closure(move(B,c))=ε-closure({8})={8,9,12,15}=D
ε-closure(move(C,a))=ε-closure({3})={1,2,3,4,6,7}=B
ε-closure(move(C,b))=ε-closure({5})={1,2,4,5,6,7}=C
ε-closure(move(C,c))=ε-closure({8})={8,9,12,15}=D
ε-closure(move(D,a))=ε-closure({10})={10}=E
ε-closure(move(D,b))=ε-closure({13})={8,9,12,13,14,15}=F
ε-closure(move(D,c))=ε-closure({16})={16}=G-accepting
ε-closure(move(E,a))=ε-closure({})={}
ε-closure(move(E,b))=ε-closure({11})={8,9,11,12,14,15}=H
ε-closure(move(E,c))=ε-closure({})={}
ε-closure(move(F,a))=ε-closure({10})={10}=E
ε-closure(move(F,b))=ε-closure({13})={8,9,12,13,14,15}=F
ε-closure(move(F,c))=ε-closure({16})={ 16}=G-accepting
ε-closure(move(G,a))=ε-closure({})={}
ε-closure(move(G,b))=ε-closure({})={}
ε-closure(move(G,c))=ε-closure({})={}
ε-closure(move(H,a))=ε-closure({10})={10}=E
ε-closure(move(H,b))=ε-closure({13})={8,9,12,13,14,15}=F
ε-closure(move(H,c))=ε-closure({16})={ 16}=G-accepting



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

b)

ε-closure(1)={1,6,13}=A

ε-closure(move(A,c))=ε-closure({2,7,17})={2,7,17}=B

ε-closure(move(B,o))=ε-closure({3,8,18})={3,8,18}=C

ε-closure(move(C,o))=ε-closure({4,9})={4,9}=D

ε-closure(move(C,m))=ε-closure({14})={14}=E

ε-closure(move(C,d))=ε-closure({19})={19}=F

ε-closure(move(D,l))=ε-closure({5,10})={5,10}=G-accepting

ε-closure(move(E,p))=ε-closure({15})={15}=H

ε-closure(move(F,e))=ε-closure({20})={20}=I-accepting

ε-closure(move(G,e))=ε-closure({11})={11}=J

ε-closure(move(H,i))=ε-closure({9})={9}=K

ε-closure(move(H,u))=ε-closure({16})={16}=L

ε-closure(move(J,r))=ε-closure({12})={12}=M-accepting

ε-closure(move(K,l))=ε-closure({10})={10}=N

ε-closure(move(L,t))=ε-closure({10})={10}=N

ε-closure(move(N,e))=ε-closure({11})={11}=J