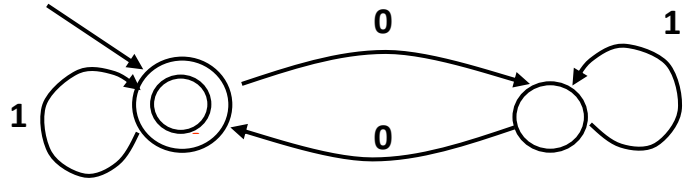
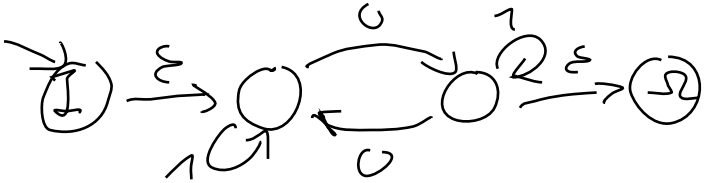
Formal Languages and Compilers Theory test July 20, 2020

1. Find a regular expression that represents the complement of the language represented by the following DFA:

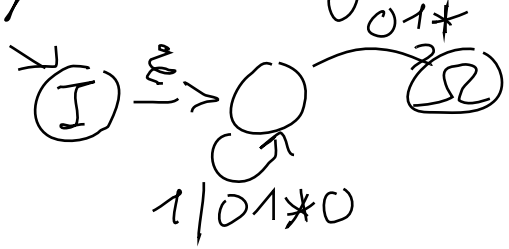


The DFA is already completely specified, so the DFA of the complement is:

Determine a corresponding regular expression:



⇒



⇒

The regular expression is:

(1|01*0)*01*

2. Write a CFG without epsilon transitions that generates the language { $a^{n+m+1} b^n c^m$ | n>=0, m>=0 }

$$a^{n+m+1} b^n c^m = \underbrace{a \underbrace{a^m \underbrace{a^n b^n}_{A} c^m}_{B}}_{S}$$

$S \to aB$

$B \to aBc \mid A$

$A \to aAb \mid \varepsilon$

Eliminate the epsilon-production:

$S \to aB \mid a$

$B \to aBc \mid A \mid ac$

$A \to aAb \mid ab$

Formal Languages and Compilers Theory test July 20, 2020

3. Given the following grammar, find NULLABLE(X) and FIRST(X) for each nonterminal X of the grammar, and then find the first row of the LR(1) parsing table for it.

1,2 S → (L) | A B
3,4 L → L , S | S
5,6 A → A a | ε
7,8 B → B b | b

0 S' → S

	NULLABLE	FIRST
S'	F	(a b
S	F	(a b
L	F	(a b
A	T	a
B	F	b

LR(1) automaton:

I0
S' → . S , $
S → . (L) , $
S → . A B , $
A → . A a , b/a
A → . , b/a

S → [ ]
(→ [ ]
A → [ ]

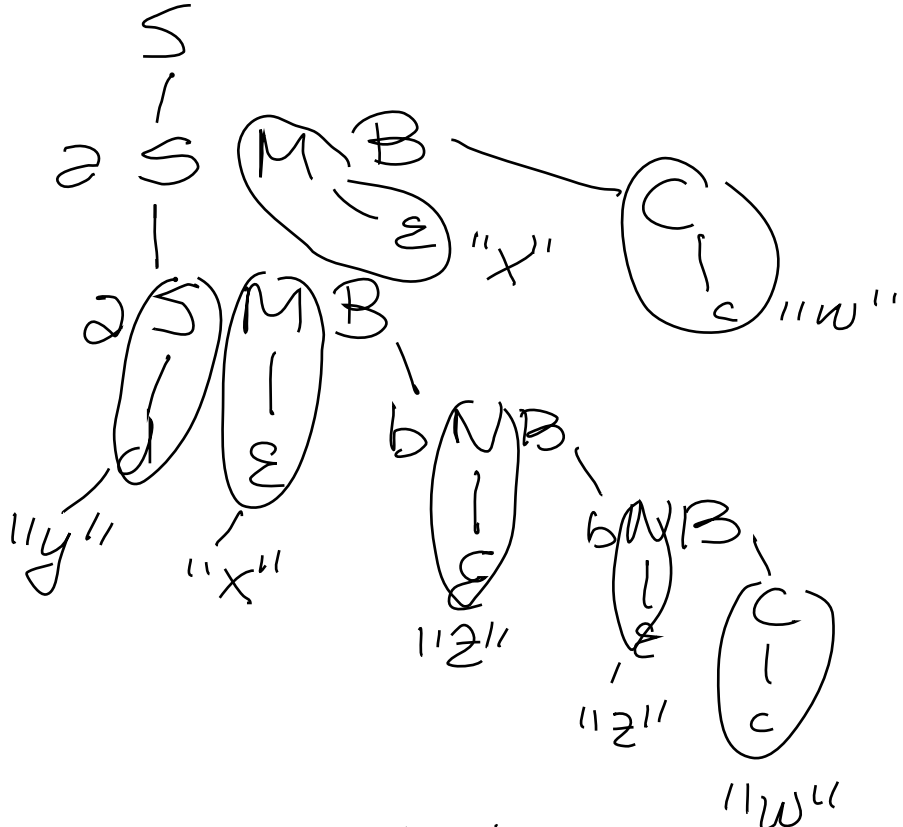
Parsing table:

	a	b	(	)	,	$	S'	S	L	A	B
I0	r6	r6	s2					1		3	

4. Transform the following SDT so that it can be implemented by a bottom-up parser and then indicate which string it will print when the input is aadbbcc.

S → a S {print “x”} B
S → d {print “y”}
B → b {print “z”} B
B → C
C → c {print “w”}

S → aSMB
M → ε {print "x"}
S → d {print "y"}
B → bNB
B → C
N → ε {print "z"}
C → c {print "w"}



⇒ y x z z w x w