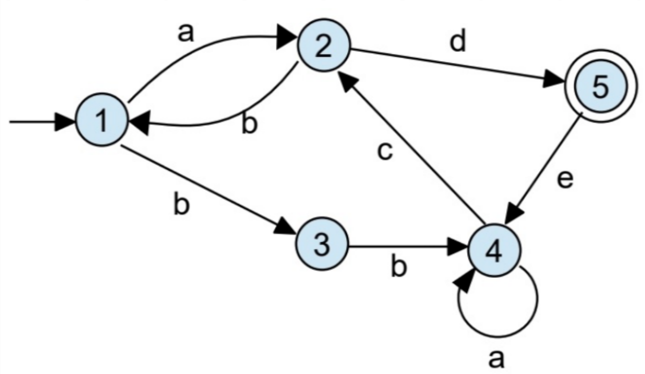
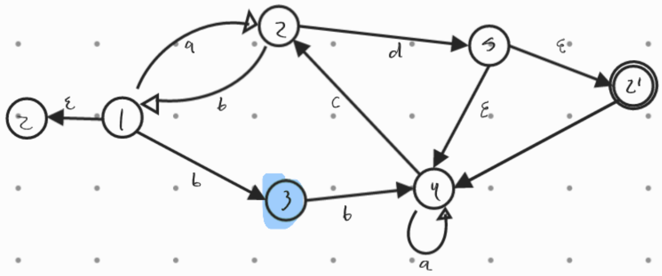
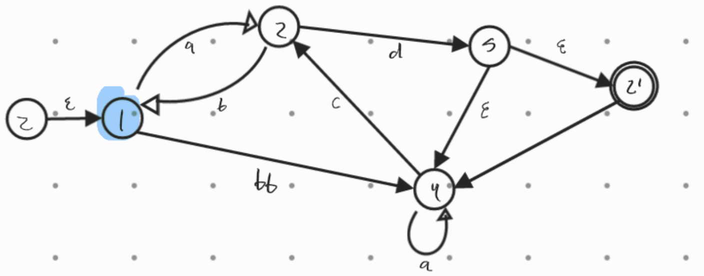
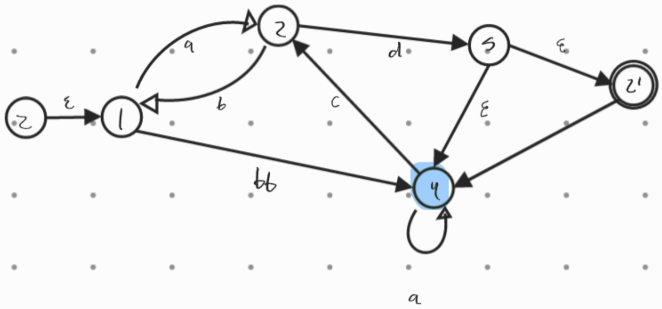
Exercise 8.1

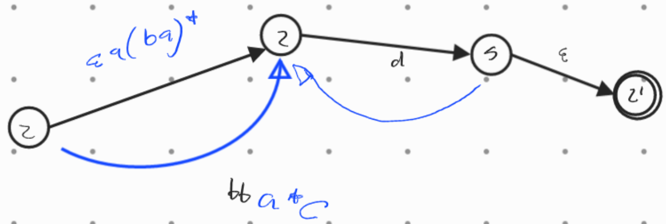
Use elimination procedure 9.11 to convert this e-NFA into an equivalent regular expression.

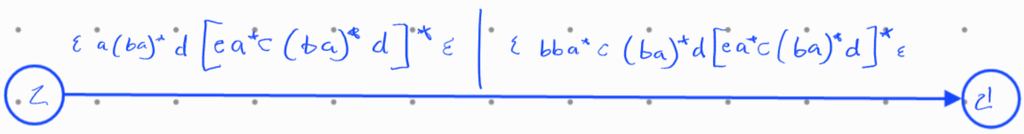












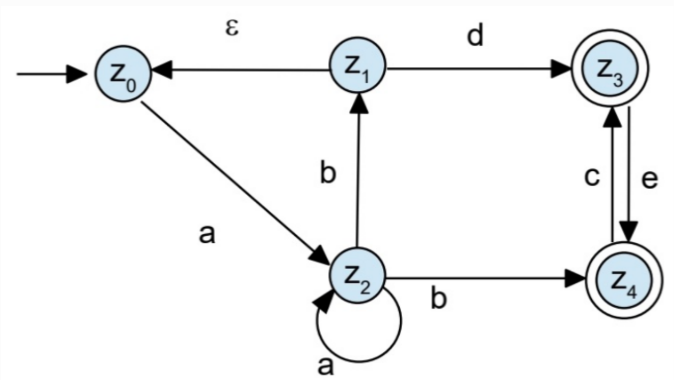
ε(a(ba)*d | bba*c(ba)*d) (ea*c(ba)*d)* ε

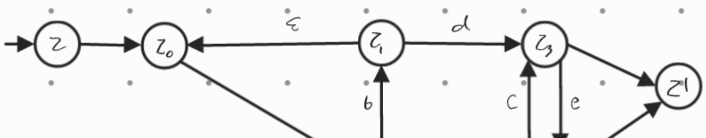
Exercise 8.2 - obligatory (8 points)

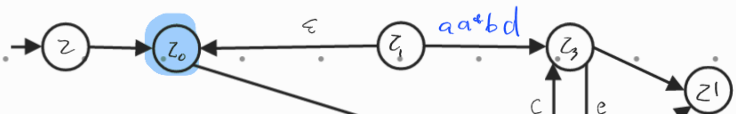
Convert this e-NFA into an equivalent regular expression.

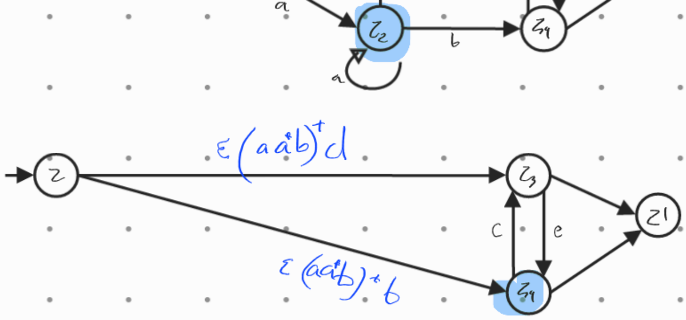
Show all important intermediate steps, so that the approach can be understood.

Hint: In order to keep notation clear, you can introduce abbreviations for complex regular expressions.









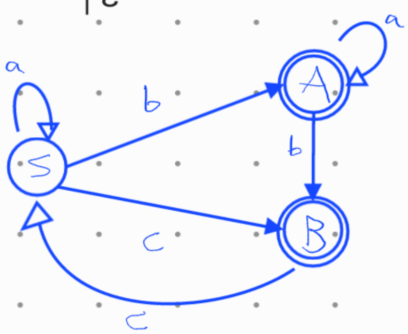
$\varepsilon(aa^*b)^+d(e(ce)^*)^* \mid (aa^*b)^+ a^+b(c(ec)^+)^*$

ε(aa*b)^+ d(e(ce)*)*ε | ε(aa*b)^+ b(c(ec)*)*ε

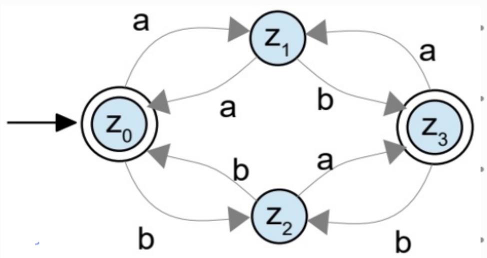
Exercise 8.3 - obligatory (4 points)

a) Specify an e-NFA that accepts the language of the following Chomsky type-3 grammar:

S → bA
| aS
| cB
A → bB
| aA
| ε
B → cS
| ε



b) Specify a Chomsky type-3 grammar that generates the language of the following DFA:



z0 = S, z1 = A, z2 = B, z3 = C
S → aA
| bB
| ε
A → aS
| bC
B → aC

| bS
C → aA
| bB
| ε

Exercise 8.4
Let ξ = {0,1}. Which of the following languages are regular, which are not? Give a brief explanation.
(1) La = { 0^k11(00)m| k > 0, m ≥ 0}
R = 0^+11(00)* Language is regular.

(2) Lb = { w ε Σ* | |w|0 is odd } (number of 0 symbols is odd)

(3) Lc = {w1,...., wn} (finite language, finite set of strings)

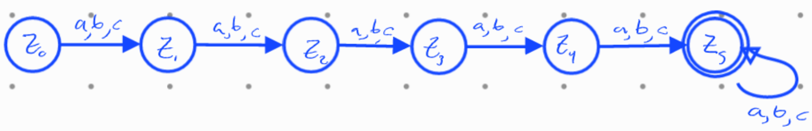
(4) Ld = {w ε Σ* | w contains 11 or 000 as a substring}

(5) Le = Σ* \ {01101, 101}

(6) Lf = {w ε Σ* | w contains 00, but not 000 as a substring}

Exercise 8.5 - obligatory (4 points)
Let Σ = {a, b, c}. Which of the following languages are regular? Give a short justification in each case.
(1) L1 = { (cc)^kab | k > 0} υ {a^n(bc)^m| n ≥ 0, m ≥ 0} not regular , |S| < P
P = 5
cccccccccab aaaaabcbcbcbcbc
= abc

(2) L2 = {w ε Σ* | |w| > 4 } (strings longer than 4 symbols) regular



(3) L3 = { c^k a^k| k ≥ 0 } not regular

S = c^p a^p
let p = xyz Pumping lemma
case 2: xy^2z ε L , but it contains more a's than b's
P=7 ; S = cccccccaaaaaaa

case1: y is in the c part
$\overset{x}{cc}\ \overset{y}{cccc}\ \overset{z}{caaaaaaa}$

xy^2z

case 1 cc cccccccc caaaaaaa

$11 \neq 7$

case2: y is in the b part
$\overset{x}{ccccccccaa}\ \overset{y}{aaaa}\ \overset{z}{a}$

case2: y is in the a and b parts
$\overset{x}{ccccc}\ \overset{y}{ccaa}\ \overset{z}{aaaaa}$

(4) L4 = { a^nw | n ≥ 3, wεΣ*} Π { u(bc)^k| uεΣ*, k ≥ 1} regular

$aaaa^*(abc)^* \cap (abc)^*(bc)^+$

L = { {}, {a}, {b}, {c}, {ab}, {ac}, {bc}, {abc} }

Exercise 8.6
Let L = { a^nbc^k| k ≥ n }.
a) (difficult!) Prove that language L is not regular. Hint: Use the pumping lemma.

$a^n\ b\ c^k$

$\underbrace{aa}_{x}\underbrace{aa\ aaa}_{y}\ \underbrace{b\ ccccccc}_{z}$

xy^2z aaaaaa aaaa ab ccccccc

$\underbrace{aaaa\ aaa\ b\ cc}_{x}\underbrace{ccc}_{y}\underbrace{cc}_{z}$

aaaaaa abcc cccc ccccc

$\underbrace{aaaa}_{x}\ \underbrace{aaa\ b}_{y}\ \underbrace{ccccccc}_{z}$

b) (not so difficult) Show that L is a context-free language.

Exercise 8.7 - obligatory (4 points)

A push-down automaton P =(Z, Σ, Γ, Λ, z0, k0, E) is defined as following:

- Set of states Z = { z0, z1, ze }
- Input alphabet Σ = { 0,1 }
- Stack alphabet Γ = { k0, x }
- z0 is start state
- k0 is initial stack symbol
- accept states E = { ze }
- Transition relation Λ:

(1) z0, k0→0 z0, k0 x
(2) z0, x→0 z0, xx
(3) z0, x→1 z1, x
(4) z1, x→0 z1,ε
(5) z1, k0 →ε ze, ε

a) Which of the following strings are accepted by P?

(1) ε

(2) 00100 accepted
state stack input next operation
(z0, k0, 00100) // push X on stack
(z0, k0X, 0100) // pop X and push XX
(z0, k0XX, 100) // pop X and push X, change state to z1
(z1, k0XX, 00) // pop X
(z1, k0X, 0) // pop X
(z1, k0, e) // change to final state
(ze, e, e) //

(3) 0010
state stack input next operation
(z0, k0, 0010) // push X
(z0, k0X, 010) // pop X and push XX
(z0, k0XX, 10) // pop X and push X, change to state z1
(z1, k0XX, 0) // pop X
(z1, k0X, e) // can't change to state z(e), k0 is not on top of stack

(4) 0100
state stack input next operation
(z0, k0, 0100) // push X
(z0, k0X, 100) // pop X and push X, change state to z1
(z0, k0X, 00) // pop X

(z1, k0, 0) // there's no X to pop, can't change state to z0

b) Which language does the PDA accept?
L = { 0^n(1)0^n) | n > 0}