SECTION 3.1

#1. b, ab, ba, bb, aab, aba, abb, baa, bab, bba, bbb

#2. Yes.

#3 (a) see page 12 for detailed solution

(b) $(\underline{1} + \underline{0}\underline{1})^* (\underline{0} + \underline{1}^*) . \underline{\lambda}^*$

(c) $(\underline{1} + \underline{0}\underline{1})^* (\underline{0} + \underline{\lambda}) . \underline{\lambda}$

#4 $\underline{a}\underline{a}\underline{a}\underline{a}^* . (\underline{b}\underline{b})^*$

#5 $(\underline{a}\underline{a})^*(\underline{b}\underline{b})^* + \underline{a}(\underline{a}\underline{a})^*\underline{b}(\underline{b}\underline{b})^*$

#6 (a) $\underline{a}\underline{a}\underline{a}\underline{a}\underline{a}^* . (\underline{\lambda} + \underline{b} + \underline{b}\underline{b} + \underline{b}\underline{b}\underline{b})$

(b) $(\underline{\lambda} + \underline{a} + \underline{a}\underline{a} + \underline{a}\underline{a}\underline{a}) . (\underline{\lambda} + \underline{b} + \underline{b}\underline{b} + \underline{b}\underline{b}\underline{b})$

#7 (a) $\{\lambda\}$ (b) ϕ

#8 Set of all strings that consists of a "b" surrounded by an even no. of a's on both sides or an odd number of a's on both sides.

$\{a^m b a^n : m - n \equiv 0 \pmod 2\}$

#9 $(\underline{b}\underline{a} + \underline{a})^* . \underline{b} . (\underline{b} + \underline{a})^*$

#10 We split $L = \{a^n b^m : n.m \geq 3, n \geq 1 \ \& \ m \geq 1\}$ into three pieces according to the conditions

$n \geq 3, m = 1$ to get $\{a^n b : n \geq 3\}$

$n \geq 2, m = 2$ " $\{a^n bb : n \geq 2\}$

$n \geq 1, m \geq 3$ " $\{a^n b^m : n \geq 1, m \geq 3\}$

Then answer is $\underline{a}\underline{a}\underline{a}\underline{a}^*\underline{b} + aaa^*bb + \underline{a}\underline{a}^*\underline{b}\underline{b}\underline{b}\underline{b}^*$

SECTION 3.1 (Number 3 redone)

#3 (a) Let $R_1 = (\underline{1} + \underline{01})^* \cdot (\underline{0} + \underline{1}^*)$ and
$R_2 = (\underline{1} + \underline{01})^* \cdot (\underline{0} + \lambda)$
be the expression from Example 3.6.

Clearly $L(R_2) \subseteq L(R_1)$ because $\underline{0} + \underline{\lambda} \subseteq \underline{0} + \underline{1}^*$
$(\lambda \in \underline{1}^*)$.

Now let $\varphi \in L(R_1)$. Then
$\varphi = \alpha . \beta$ where $\alpha \in L((\underline{1} + \underline{01})^*)$ and
$\beta = 0$ or $\beta \in L(\underline{1}^*)$

But if $\beta = 0$, then $\varphi \in L(R_2) = L((\underline{1} + \underline{01})^* \cdot (\underline{0} + \lambda))$
And if $\beta \in L(\underline{1}^*)$, then $\beta = 1^n$ for some
$n \geq 0$. So

$$\varphi = \alpha . 1^n \quad \text{with} \quad \alpha \in L((\underline{1} + \underline{01})^*)$$
= 1's & (01)'s followed by n 1's
$$\in L\ (\underline{1} + \underline{01})^* = L((\underline{1} + \underline{01})^* \cdot \underline{\lambda})$$
$$\subseteq L\ (\underline{1} + \underline{01})^* \cdot (\underline{0} + \underline{\lambda})$$

$\therefore L(R_1) \subseteq L(R_2)$. Hence $L(R_1) = L(R_2)$

(b) $(\underline{01} + 1)^* \cdot (\underline{\lambda} + \underline{0})$ and $(\underline{1} + \underline{01})^* \cdot (\underline{\lambda} + \underline{0} + \underline{1})$
are two other expressions which are
equivalent to $(\underline{1} + \underline{01})(\underline{0} + \underline{1}^*)$.

#12 $L = \underline{a}\,\underline{b}\underline{b}\underline{b}\underline{b}^*(\underline{a}+\underline{b})(\underline{a}+\underline{b})^*$

$L =$ set of all strings which consists of an even no. of a's followed by an odd no. of b's

$= \{a^{2n}b^{2m+1} : n \geq 0, m \geq 0\}$

$L^c =$ set of all strings of the form $a^n b^m$ with n odd or m even, or of strings with a "b" in front of an "a"

$= \{a^{2n+1}b^m : n \geq 0, m \geq 0\} \cup \{a^n b^{2m} : n \geq 0, m \geq 0\}$
$\cup$ { anything . ba . anything }

An expression for L^c is now easily seen to be

$$\underline{a}(\underline{a}\underline{a})^*\underline{b}^* + \underline{a}^*(\underline{b}\underline{b})^* + (\underline{a}+\underline{b})^*.\underline{b}\underline{a}.(\underline{a}+\underline{b})^*$$

#13. $\underline{a}\underline{a}(\underline{a}+\underline{b})^*\underline{a}\underline{a} + \underline{a}\underline{b}(\underline{a}+\underline{b})^*\underline{a}\underline{b} + \underline{b}\underline{a}(\underline{a}+\underline{b})^*\underline{b}\underline{a} + \underline{b}\underline{b}(\underline{a}+\underline{b})^*\underline{b}\underline{b}$.

#14. A silly question. The answer is $(\underline{a}+\underline{b})^*$.

#15. $(\underline{1} + \underline{0}\underline{1})^*.\underline{0}\underline{0}.(\underline{1}+\underline{1}\underline{0})^*$

#16 (a) $(\underline{b}+\underline{c})^*\underline{a}(\underline{b}+\underline{c})^*$

(b) Look at the four cases: no a's, one a, two a's and three a's. With these cases we get:
$(b+c)^* + (b+c)^*.a.(b+c)^* + (b+c)^*.a.(b+c)^*.a.(b+c)^*$
$+ (b+c)^*.a.(b+c)^*.a.(b+c)^*.a.(b+c)^*$

(c) Look at six cases: ...a...b...c..., ...a...c...b..., ...b...a...c..., ...b...c...a..., ...c...a...b..., ...c...b...a... .
Now insert $(a+b+c)^*$ for the dots.

Section 3.1   exactly 2 00's
$(1+01)^*.(000+00(1+10)^*.100).(1+10)^*$

#17. (a) $(0+1)^*.01$ (f) $(0+\lambda)(1+00+000)^*(0+\lambda)$

(b) $(\lambda+0+1)+(0+1)^*.(00+10+11)$

(c) $(1^*01^*01^*)^*+1^*$ is one answer

$(1+01^*0)^*$ is another answer

(d) $(0+1)^*(000+00(0+1)^*.00)(0+1)^*$

(e) $(01+1)^*(\lambda+0+00+000+00.(10+1)^*100)(10+1)^*$

#20 (a) Clearly $L(r_1^*)\subseteq L((r_1^*)^*)$. Now let $\varphi\in L((r_1^*)^*)$. Then

$\varphi =$ a string of strings from r_1^*

= a string of strings of strings from r_1

= a string of strings from r_1

$\in L(r_1^*)$

$\therefore\ L((r_1^*)^*)\subseteq L(r_1^*)$

Thus $L((r_1^*)^*)=L(r_1^*)$ i.e. $(r_1^*)^*\equiv r_1^*$

(b) Again clearly $L((r_1+r_2)^*)\subseteq L(r_1^*(r_1+r_2)^*)$ because $\lambda\in L(r_1^*)$.

Now let $\varphi\in L(r_1^*(r_1+r_2)^*)$. Then

$\varphi = \alpha.\beta$ with $\alpha\in L(r_1^*)$ & $\beta\in L((r_1+r_2)^*)$

$= \alpha_1\alpha_2\cdots\alpha_m.\beta_1\beta_2\cdots\beta_n$ with the α_i's in $L(r_1)$ and β_j's in $L(r_1+r_2)$

$= \alpha_1\alpha_2\cdots\alpha_m\beta_1\cdots\beta_n$ with the α_i's and β_j's in $L(r_1+r_2)$

$\in L((r_1+r_2)^*)$

$\therefore\ L(r_1^*(r_1+r_2)^*)\subseteq L((r_1+r_2)^*)$.

So $L(r_1^*(r_1+r_2)^*)=L((r_1+r_2)^*)$ i.e. $r_1^*(r_1+r_2)^*\equiv(r_1+r_2)^*$.

#20 (c) First observe that $L(r_1^* r_2^*) \supseteq L(r_1 + r_2)$ bec. $\lambda \in L(r_1^*)$ and $\lambda \in L(r_2^*)$. So $L((r_1^* r_2^*)^*) \supseteq L((r_1+r_2)^*)$

Now let $\varphi \in L((r_1^* r_2^*)^*)$. Then

$\varphi =$ a string of things which are made of a string from r_1^* followed by a string from r_2^*

$=$ a string of things which are made up of strings of things in r_1 followed by strings of things in r_2.

$=$ a string of things from either r_1 or r_2

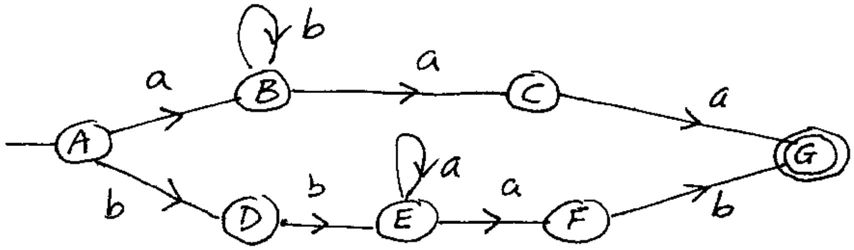
$\in L((r_1 + r_2)^*)$.

So $L((r_1^* r_2^*)^*) \subseteq L((r_1+r_2)^*)$. Thus $L((r_1^* r_2^*)^*) = L((r_1+r_2)^*)$ i.e. $(r_1 + r_2)^* \equiv (r_1^* r_2^*)^*$

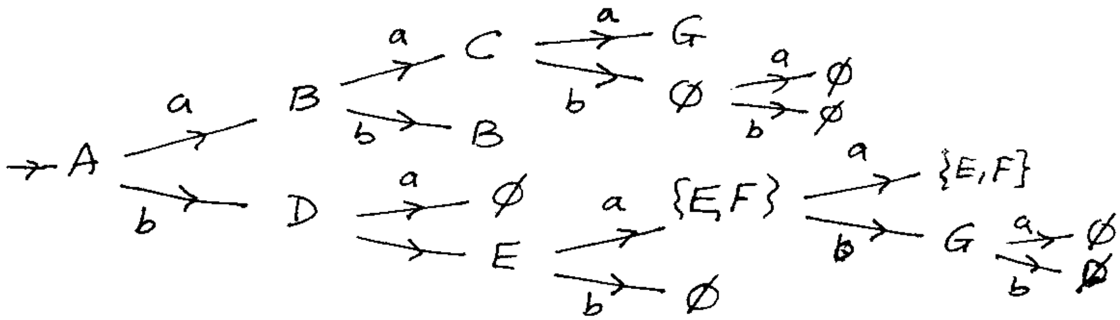
d) false. $(\underline{a} \cdot \underline{b})^* \neq \underline{a}^* \cdot \underline{b}^*$ because $abab \in (\underline{a} \cdot \underline{b})^*$ but $abab \notin \underline{a}^* \cdot \underline{b}^*$.

SECTION 3.2 p.87

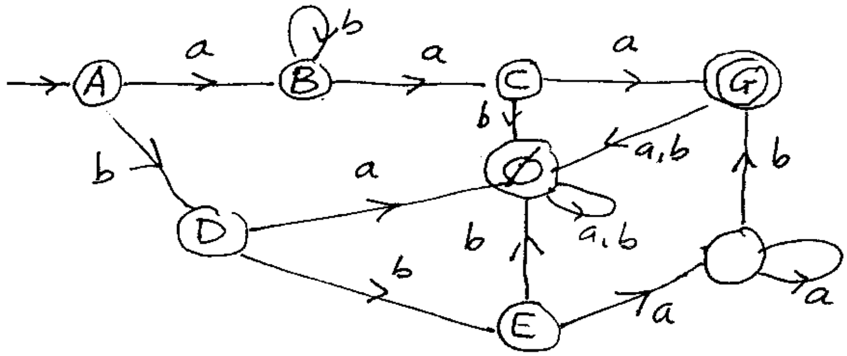
#1 $M =$



#2. (a) First convert M into a dfa M_D

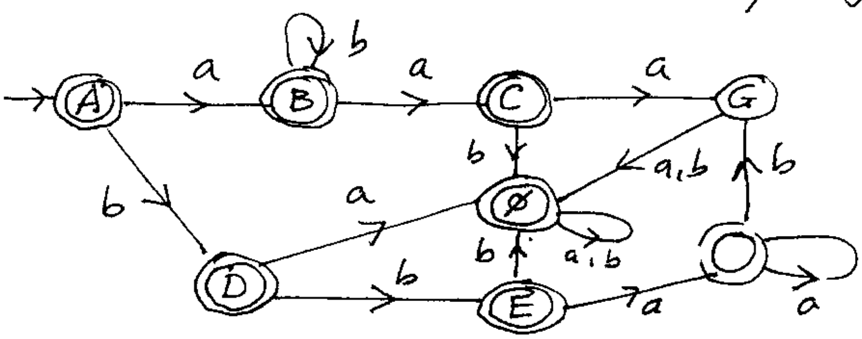


$M_D =$

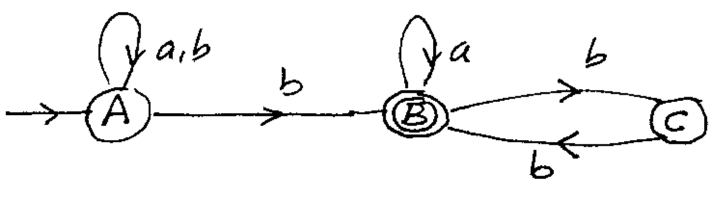


(b) Then switch accepting & non-accepting states

$M_D^c =$



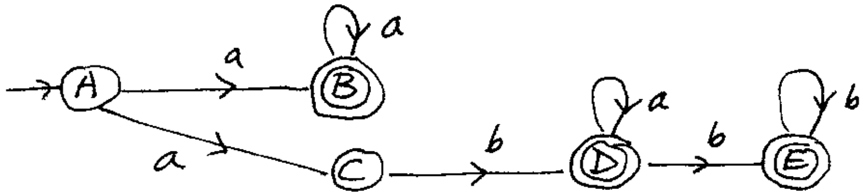
#3.



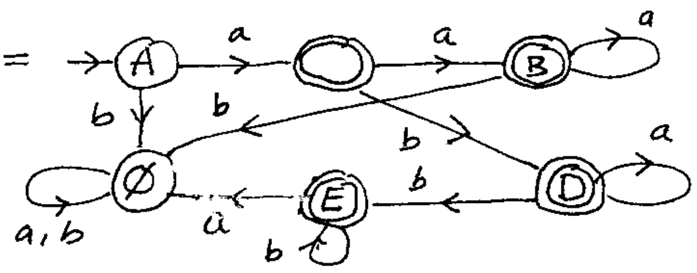
Section 3.2 p. 87

#4. First find an nfa and then convert it into a dfa

(a) M =
nfa:

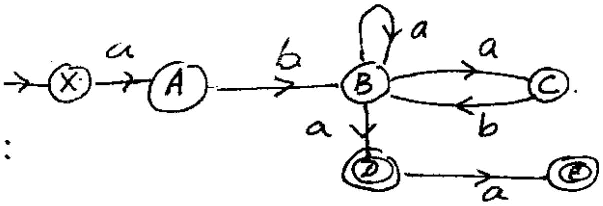


M_D =

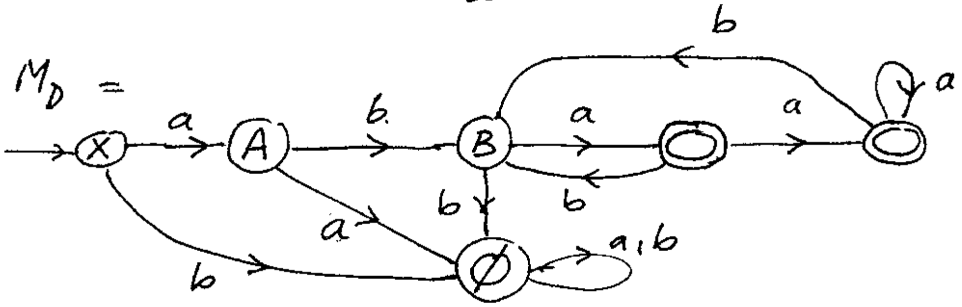


= dfa

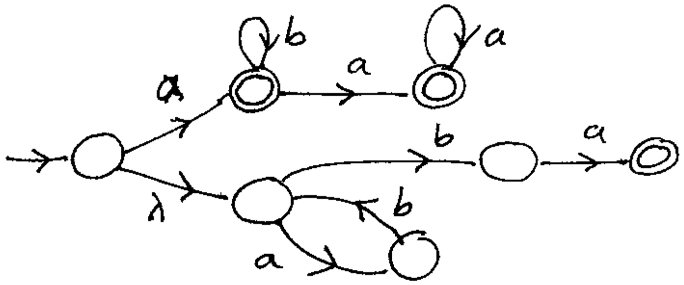
(b) M =
nfa:



dfa: M_D =

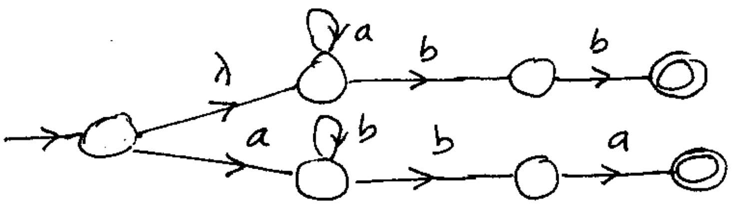


#5. (a) nfa,
M =



Now convert M into a dfa M_D.

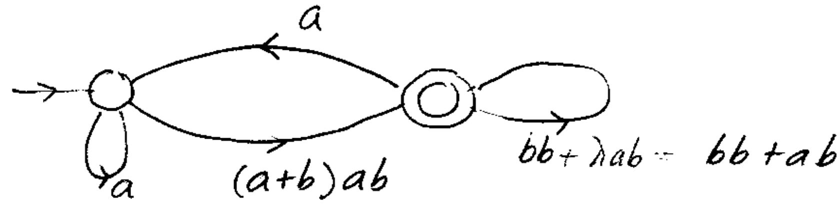
#7 nfa M =



Now convert M into a dfa M_D and then minimize M_D by using the Partition Algorithm.

SECTION 3.2 p. 88

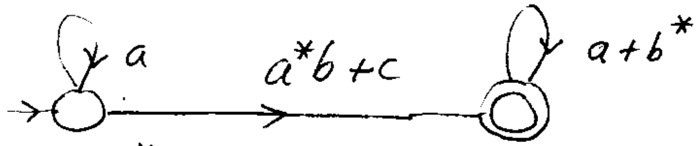
#8 (a)



(b) $\underline{a}^* \cdot \underline{(a+b)ab} \cdot \left((\underline{bb+ab}) + \underline{a} \cdot \underbrace{a^*(a+b)ab} \right)^*$

$\uparrow r_1{}^*$ $\uparrow r_2$ $\uparrow r_4$ $\uparrow r_3$ $\uparrow r_1^* \cdot r_2$

#9

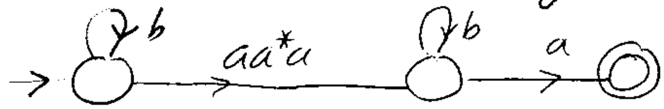


$L(M) = \underline{a}^* \cdot \underbrace{(a^*b+c)} \cdot (\underbrace{a+b^*})^*$

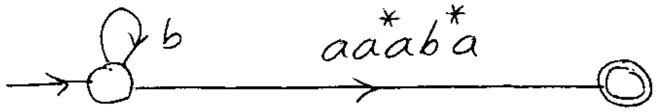
$\uparrow r_1{}^*$ $\uparrow r_2$ $\uparrow r_4$

#10 (a) Since this is an easy example, you can instantly see that $L(M) = b^*aa^*ab^*a$. But let's see how the algorithm proceeds

M_1



M_2



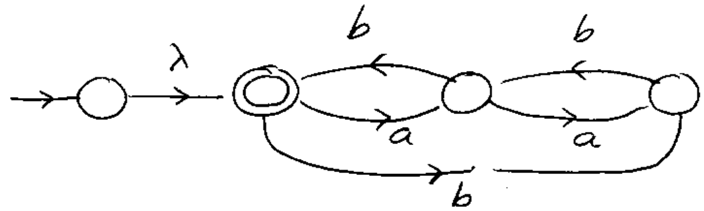
$L(M) = \underbrace{b}^* \cdot \underbrace{aa^*ab^*a} \cdot (\emptyset)^* \quad \leftarrow r_1^* r_2 (\underbrace{r_4}_{\emptyset} + \underbrace{r_3 r_1^* r_2}_{=\emptyset})^*$ bec. $r_3 = \emptyset$

$\uparrow r_1{}^*$ $\uparrow r_2$ $\uparrow r_4 + r_3 r_1^* r_2$

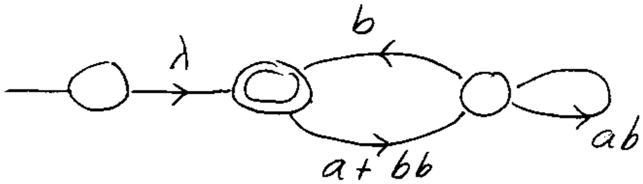
$= b^*aa^*ab^*a \cdot \lambda = b^*aa^*ab^*a$.

SECTION 3.2 p. 89

10(b) M_0 :



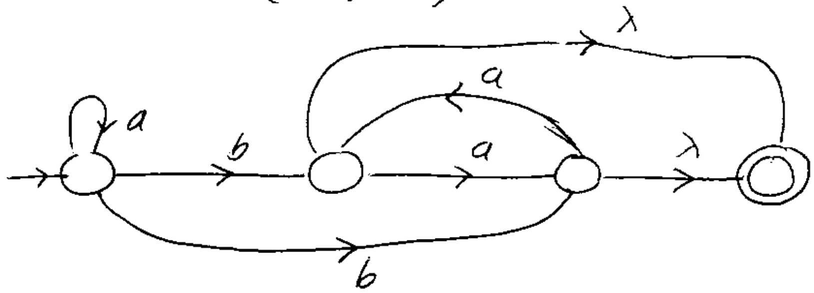
M_1 :



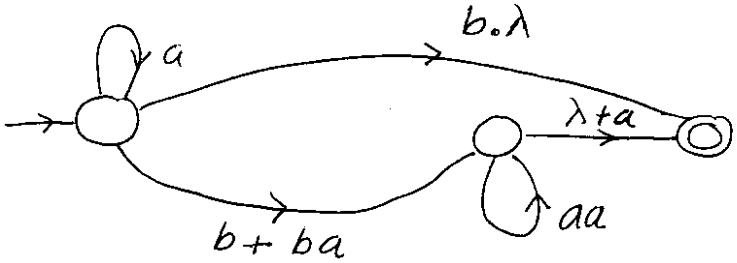
M_2 :

$$L(M) = \lambda.\left((a+bb).(ab)^*.b\right)^*$$

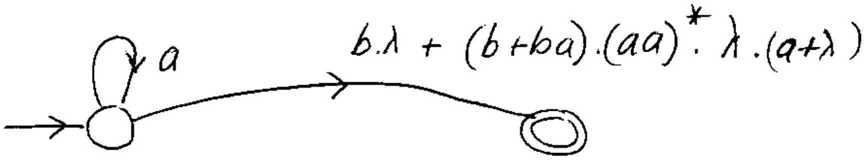
(c) M_0 :



M_1 :



M_2 :



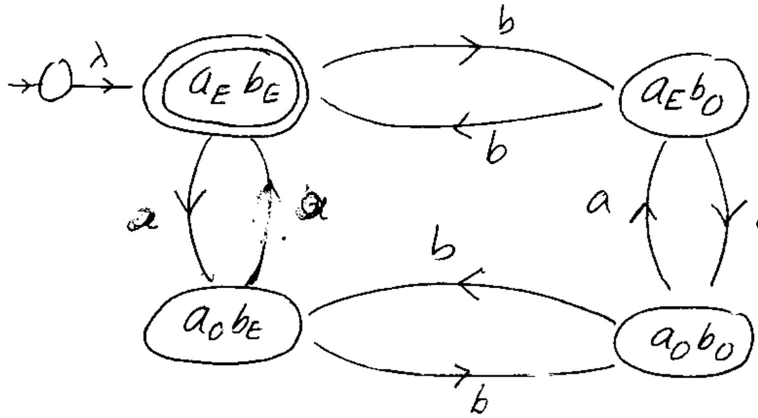
$$L(M) = a^*.\left(b + (b+ba).(aa)^*.\lambda.(a+\lambda)\right)$$

SECTION 3.2 p. 89

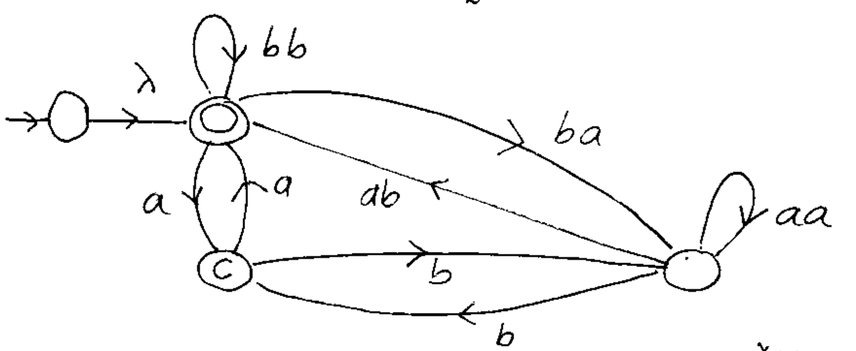
#13 (a) First find an nfa and then find the regular expression from your nfa.

E = even
O = odd

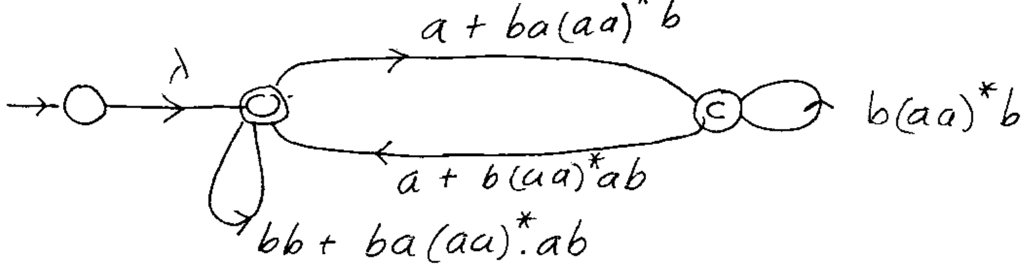
M_0:



M_1:

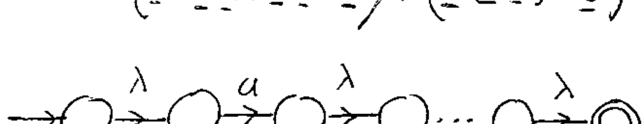


M_2:

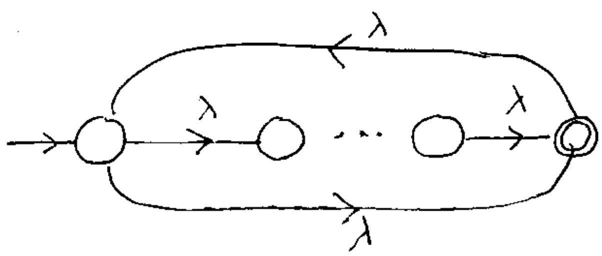


$$L(M) = \lambda \cdot \left(bb + ba.(aa)^*.ab + \left(a + ba(aa)^*b\right).\left(b(aa)^*.b\right)^*.\left(a + b(aa)^*.ab\right)\right)^*$$

#18 (a) $L(a\emptyset)$:



(b) $L(\emptyset^*)$

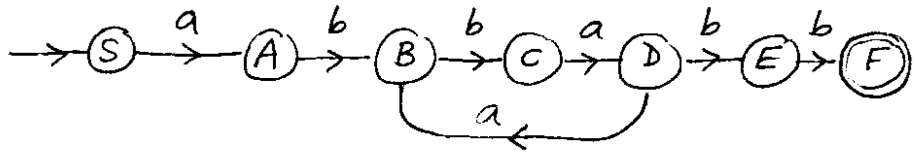


.... = no connection.
= dead wire

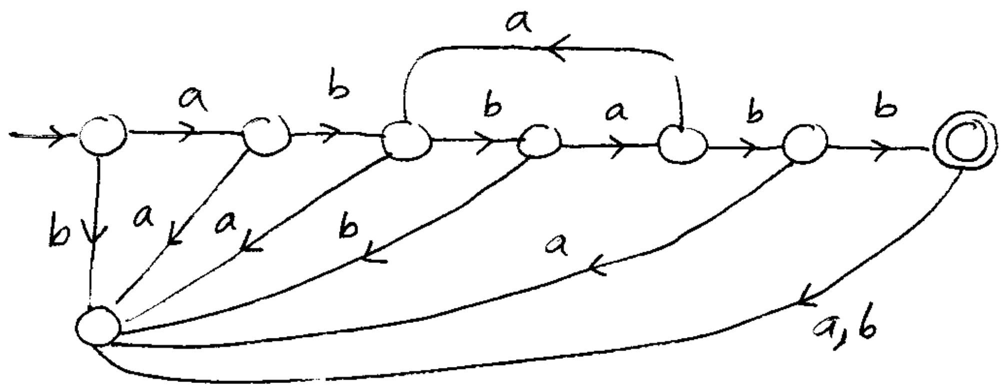
SECTION 3.3 p. 96

#1. First find an nfa that accepts L(G) and then convert your nfa into a dfa.

nfa :



dfa :



#2. $S \to aA$, $A \to aA$, $A \to B$, $B \to abB$, $B \to aB$, $B \to \lambda$.

#3. The simplest thing to do is to find L(G) in Exercise 1 and then find a left-linear grammar for L(G).
$L(G) = abba.(aba)^* bb$
Left-Lin. Grammar is $S \to Abb$, $A \to Aaba | abba$

#4. (a) RLG: $S \to aaA$, $A \to aA | B$, $B \to bB | bbb$
(b) LLG: $S \to Bbbb$, $B \to Bb | A$, $A \to aa | Aa$

#6. RLG: $S \to aaB$, $B \to bB$, $B \to ab$, $B \to abS$
$S \to \lambda$.

Note: After you see the scheme, you will then realize that the 3rd production is not needed. So a better ans. is: $S \to aaB | \lambda$, $B \to abS | bB$.

SECTION 3.3 p. 99

\#7. Let $L_i = \{\varphi \in \{a,b\}^* : \varphi \text{ has exactly } i \text{ a's}\}$ for $i = 0, 1, 2$ & 3. Find regular grammars G_i for L_i and then find a grammar G which gives the union.

G: $S \to S_0/S_1/S_2/S_3$, $S_0 \to bS_0/\lambda$,
$S_1 \to bS_1/aA$, $A \to bA/\lambda$,
$S_2 \to bS_2/aB$, $B \to bB/aC$, $C \to bC/\lambda$
$S_3 \to bS_3/aD$, $D \to bD/aE$, $E \to bE/aF$
$F \to bF/\lambda$.

\#10. $S \to Aab$, $A \to Ab$, $A \to aa$, $A \to Saa$, $S \to \lambda$
As in exercise 5 you don't really need the 3rd production.

\#11 Let $L_1 = \{a^n b^m : n \text{ \& } m \text{ are even}\}$
$L_2 = \{a^n b^m : n \text{ \& } m \text{ are odd}\}$
Then $L = L_1 \cup L_2$. Find Regular grammars for L_1 & L_2 & then do the union thing:
$S \to S_1/S_2$ $S_2 \to aaS_2/aB$, $B \to bbB/b$
$S_1 \to aaS_1/A$, $A \to bbA/\lambda$

\#12 $S \to aA/bB/\lambda$, $A \to aS/bC$, $B \to bS/aC$, $C \to bA/aB$.

\#13 <u>Hint</u>: Find the corresponding nfa then convert.
(a) $S \to \lambda/bB/aD$, $B \to bS/aC$, $C \to aB/bD$
$D \to aS/bC$