Let $x = e + E$. We have to watch out that our regular expression doesn't match ".". So for the non-exponent part:

$$\underbrace{(a+m+\lambda)}_{\text{matches } \pm \text{ or nothing}} \quad \underbrace{\underbrace{[dd^*(p+\lambda)d^*}_{\text{matches } 0,\ 0.0 \text{ but not } .0 \text{ or } .} + \underbrace{pdd^*]}_{\text{matches } .0 \text{ but not } .}}_{\text{matches } 0,\ 0.0,\ .0 \text{ but not } .}$$

Now for the exponent:

$$\underbrace{x}_{\text{matches } e \text{ or } E} \quad \underbrace{(a+m+\lambda)}_{\text{matches } \pm \text{ or nothing}} \quad \underbrace{dd^*}_{\text{matches 1 or more digits}}$$

Thus we get

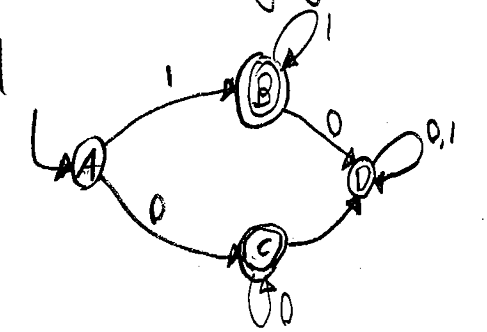
$$(a+m+\lambda)[dd^*(p+\lambda)d^* + pdd^*]x(a+m+\lambda)dd^*$$

$x = e + E$

$\boxed{3.24}$ 15 pts

Consider the language L over $\{0,1\}$ accepted by

M



also $11^* + 00^*$

Now consider the strings

$x_1 = 0$
$x_2 = 1$

these are distinguishable (0 or 1 distinguishes them). Thus by Lemma 3.1

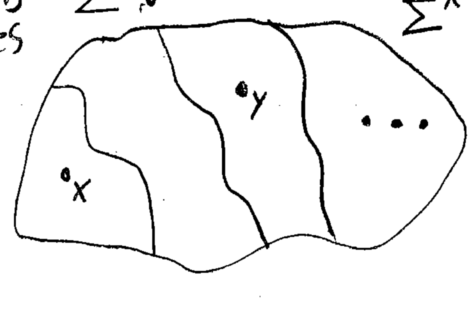
$$\delta^*(A, x_1) \neq \delta^*(A, x_2)$$

i.e. the machine must be in diffrent states after read X_1 and X_2. Since $x_1 \in L$ and $x_2 \in L$, $\delta^*(A, x_1)$ and $\delta^*(A, x_2)$ must be accepting states, thus any FA that recognizes L must have at least two accepting states.

[In general: if L contains two distinguishable words (distinguishable with respect to L), then any FA recognizing L must have at least two states.]

The equivalance relation I_L partitions Σ^*: if any two strings from diffrent classes are in L, then any FA that accepts L must have two or more accept states.

Σ^*



15pt

$\boxed{3.25}$ Let L be the language accepted by the FA.

Looking at the FA in Fig. 3.11d, consider the shortest string that will get you into each of the FA's four states:

$x_1 = \Lambda$ } land in accept state
$x_2 = 0$

$x_3 = 01$ } land in reject state
$x_4 = 00$

Of these x_1 and x_2 are accepted by the FA and x_3 and x_4 are not accepted.

Thus x_1 or x_2 can be distinguished from x_3 or x_4 by Λ. $x_1\Lambda, x_2\Lambda \in L$ but $x_3\Lambda, x_4\Lambda \notin L$.

Within these two classes, 0 distinguishes x_1 from x_2:

$$x_1 0 = 0 \in L$$
$$x_2 0 = 00 \notin L$$

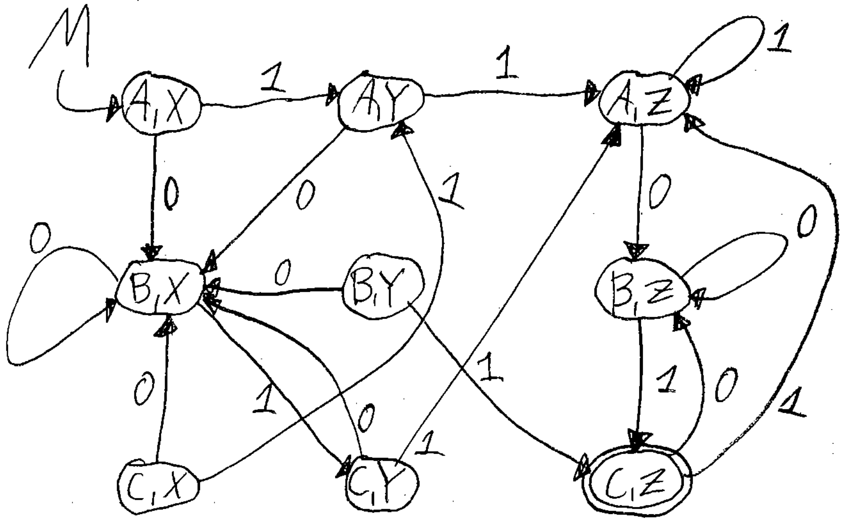
and 0 also distinguishes x_3 and x_4

$$x_3 0 = 010 \in L$$
$$x_4 0 = 000 \notin L$$

Thus, by Thm 3.2, any FA that accepts L must have at least four states.

(b) Using the <u>cross-product construction</u>, we will produce an FA that has states coresponding to pairs of states from the original FAs:

M

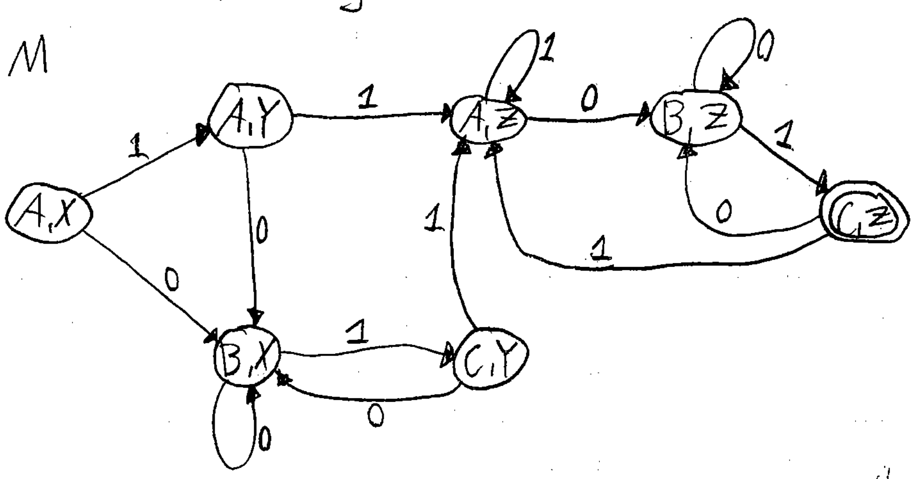


all states have 2 arrows leaving ☑

Since we want to recognized $L_1 \cap L_2$ the final states of M corespond to the states were M_1 <u>and</u> M_2 accept. This is state (C,Z).

Note that there are no arrows into states (C,X) and (B,Y). Thus these states can be removed. This gives:

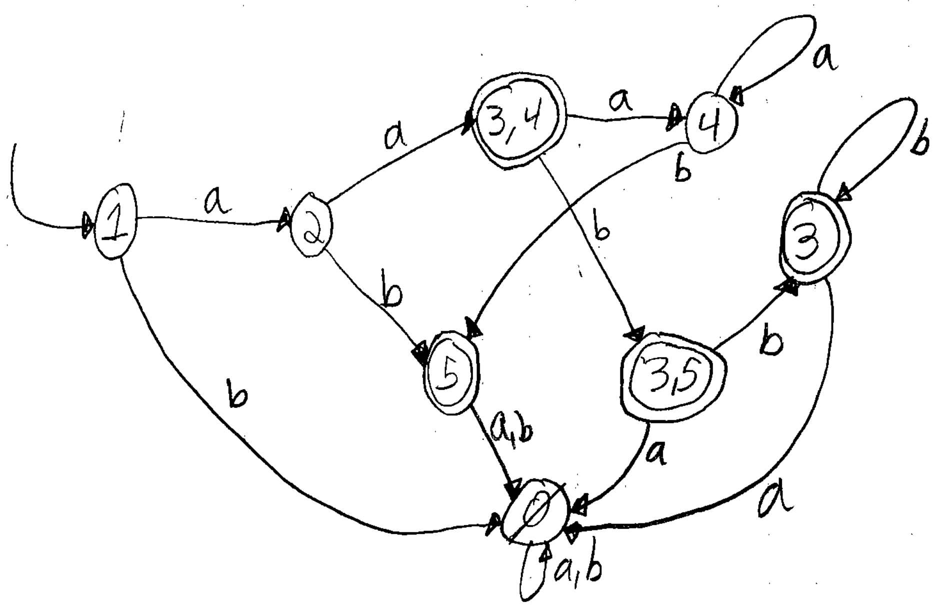
M



cross prod con

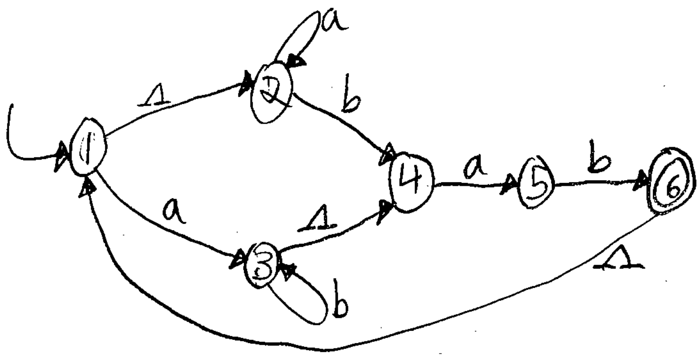
4.10f   15 pts

Using the subset construction, we will produce an FA that has states that correspond to subsets of states of the original NFA.



$\boxed{\text{Problem 6}}$ 20pts

The Λ-NFA is



Why are states always labeled starting with 1 and not 0?

The table is

q	$\delta^*(q,\Lambda^*)$	$\delta^*(q,\Lambda^* a)$	$\delta^*(q,\Lambda^* b)$	$\delta^*(q,\Lambda^* a \Lambda^*)$	$\delta^*(q,\Lambda^* b \Lambda^*)$
1	1, 2	2, 3	4	2, 3, 4	4
2	2	2	4	2	4
3	3, 4	5	3	5	3, 4
4	4	5	$\emptyset$	5	$\emptyset$
5	5	$\emptyset$	6	$\emptyset$	1, 2, 6
6	1, 2, 6	2, 3	4	2, 3, 4	4

new edges given here

Now we removed the transitions and add the ones given by the last two columns: