4. Construct npda's that accept the following languages on $\Sigma = \{a, b, c\}$.

4a. $L = \{a^n b^{2n} : n \geq 0\}$

The solution is obtained by letting each a put two markers on the stack, while each b consumes one.

$$\delta(q_0, \lambda, z) = \{(q_f, z)\},$$

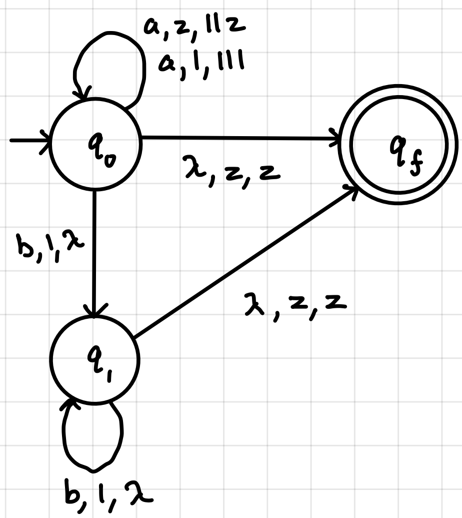
$$\delta(q_0, a, z) = \{(q_1, 11z)\},$$

$$\delta(q_0, a, 1) = \{(q_1, 111)\},$$

$$\delta(q_1, b, 1) = \{(q_1, \lambda)\},$$

$$\delta(q_1, \lambda, z) = \{(q_f, z)\}.$$

Professor Solution:

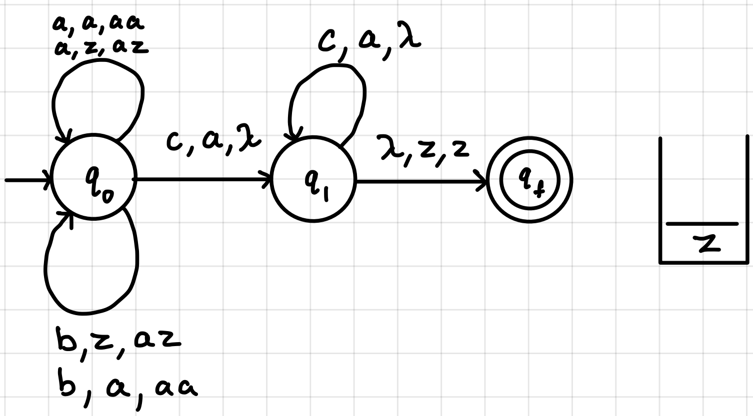


4c. $L = \{a^n b^m c^{n+m} : n \geq 0, m \geq 0\}$

$a^n b^m c^n c^m = \{\lambda, abcc, ac, bc \ldots\}$

Algorithm:

1. Read a, put token to be consumed by c
2. Read b, put token to be consumed by c
3. Read c, change state. For each c, pop a token.
4. Read λ, pop z and push z, move to the final state.



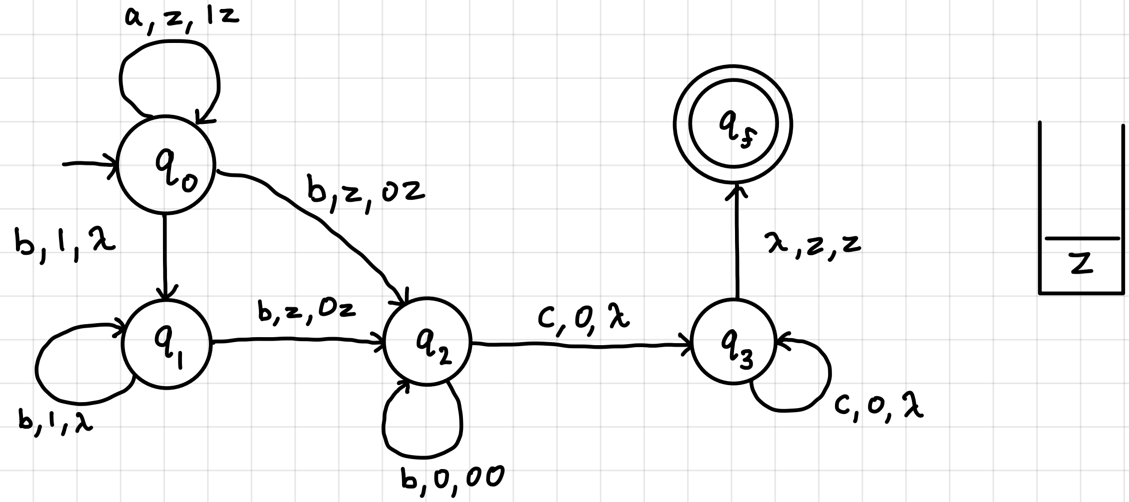
4d. $L = \{a^n b^{n+m} c^m : n \geq 0, m \geq 1\}$

$\underbrace{a^n b^n}\ \underbrace{b^m c^m}$

Algorithm:

1. Read a, put a token to be consumed by b
2. Read b, change state, for each b, pop a token until stack start symbol appears. Switch the state.
3. b puts token to be consumed by c
4. Read c, change state. For each c, pop a token.
5. Read λ, pop z and push z, move to the final state.

TG:



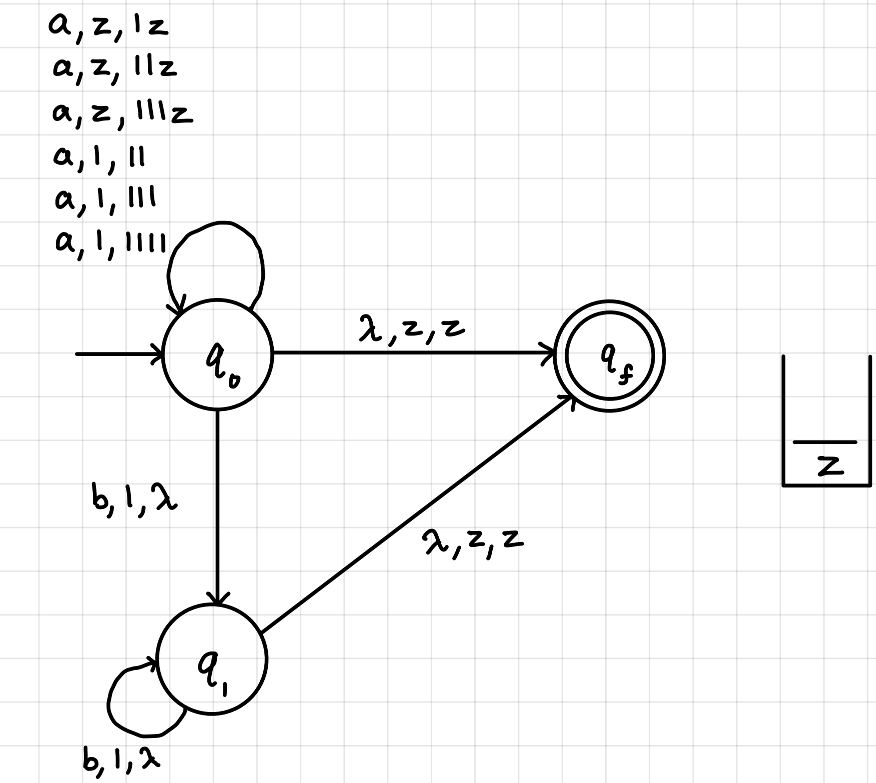
4f. $L = \{a^n b^m : n \le m \le 3n\}$

Algorithm:

For each a read, we put 1 or 2 or 3 tokens to be consumed by b.
When read b, change state and for each b, pop one token.

$ab \in L$	$aabb \in L$	
$abb \in L$	$aabbb$	$aabbbbb$
$abbb \in L$	$aabbbb$	$aabbbbbb$

TG:



2. Design a Turing machine with no more than three states that accepts the language $L(a(a + b)^*)$.Assume that $\Sigma = \{a, b\}$.Is it possible to do this with a two-state machine?

A three-state solution that scans the entire input is

$$\delta(q_0, a) = (q_1, a, R),$$

$$\delta(q_1, a) = \delta(q_1, b) = (q_1, a, R),$$

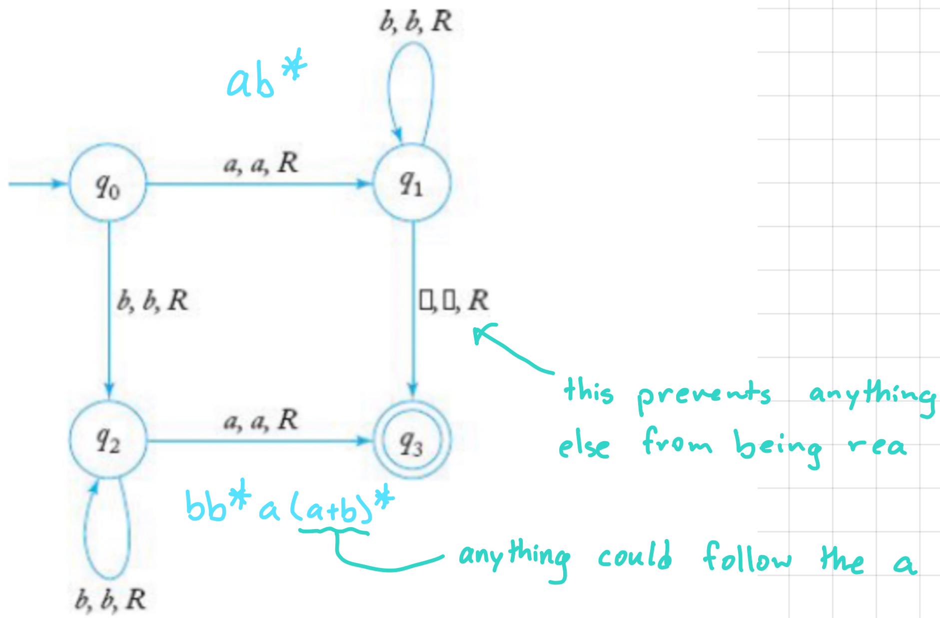
$$\delta(q_1, \square) = (q_2, \square, R), \qquad \text{with } F = \{q_2\}.$$

It is also possible to get a two-state solution by just examining the first symbol and ignoring the rest of the input, for example,

$$\delta(q_0, a) = (q_2, a, R).$$

Notice that in a Turing machine it is not necessary to examine the entire input before accepting it.

5. What language is accepted by the Turing machine whose transition graph is in the figure below?



$L = L(ab^* + bb^*a(a+b)^*)$

7. Construct Turing machines that will accept the following languages on {a,b}.

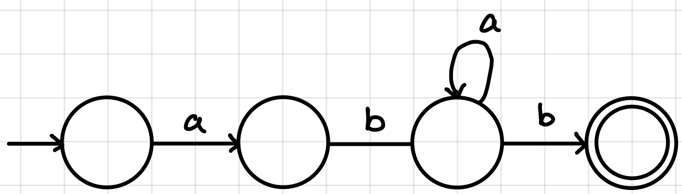
7a. $L = L(aba^*b)$

Book Solution:

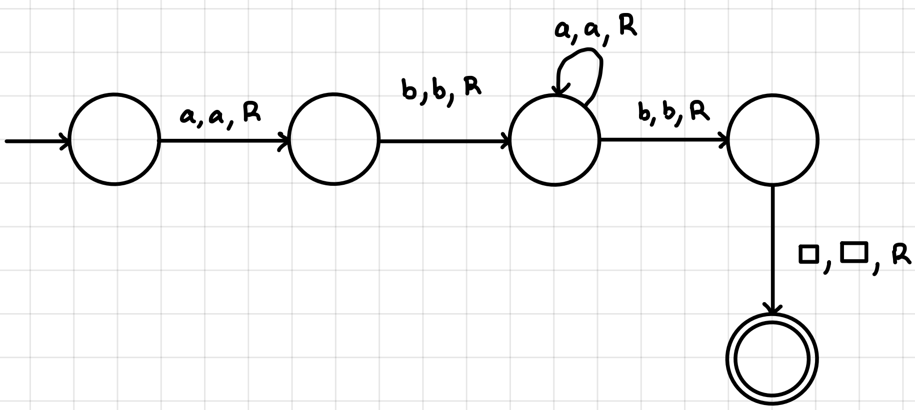
$$\delta(q_0, a) = (q_1, a, R),$$
$$\delta(q_1, b) = (q_2, b, R),$$
$$\delta(q_2, a) = (q_2, a, R),$$
$$\delta(q_2, b) = (q_3, b, R), \text{ with } F = \{q_3\}.$$

Professor

FA:



TM:

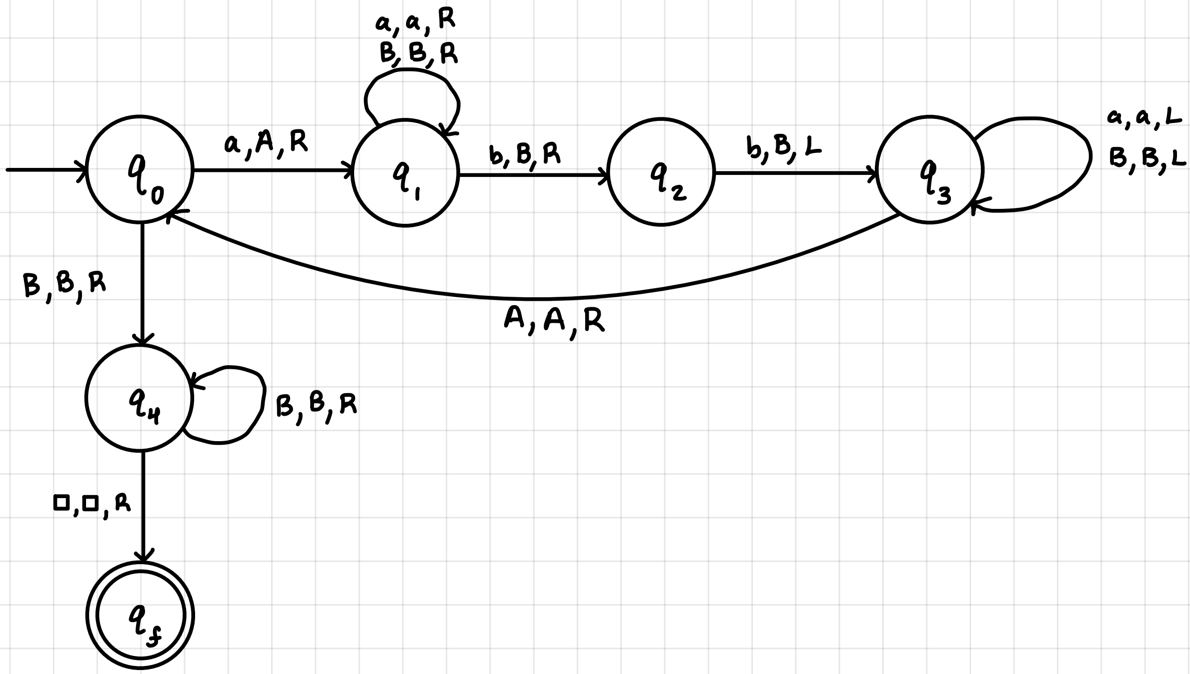


7h. $L = \{a^n b^{2n} : n \geq 1\}$

Algorithm:

1. For each a, turn it to A, move right to find 2 b's then turn them to BB.
2. Move back (left) until read A, move right to read another a or when read B move right all the way to read □.

TG:



1. Find context-sensitive grammars for the following languages.

1b. $L = \{a^n b^n a^{2n} : n \geq 1\}$

$S \to abaa \mid aAbaa$
$Ab \to bA$
$Aa \to Bbaaa$
$bB \to Bb$
$aB \to aa \mid aaA$

Test $n=2$: $a^2 b^2 a^4$

$S \Rightarrow a\underline{A}baa \Rightarrow ab\underline{Aa}a \Rightarrow ab\underline{Bbaaa}a \Rightarrow \underline{aB}bbaaaa \Rightarrow \underline{aa}bbaaaa$

TM for 3-integer adder

Algorithm: + + + $\overbrace{a * b}^{*}$ $\overbrace{a * b}^{*}$ $\overbrace{a * \square}^{0}$

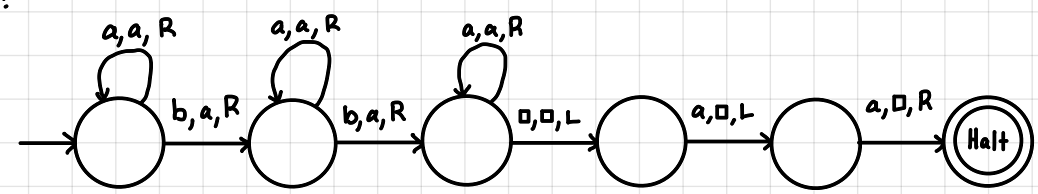
* { read a write a, move to R
read b write a, move to R, change state

* { read a write a, move to R
read b write a, move to R, change state

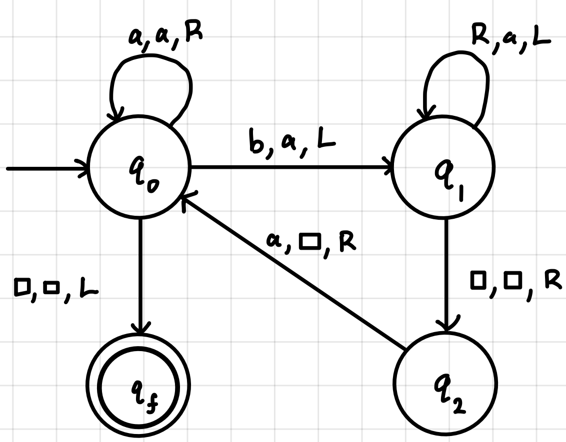
0 { read a write a, move to R
read □ write □, move to L, change state

read a, write □, move to L, change state
read a, write □, move to R, go to Halt state

TM:



TM for any # of int. adder



Input

	2	2		3	3	3		1		2	2	
□	a	a	b	a	a	a	b	a	b	a	a	□

⇓ TM

8
6 + 2
5 + 1 + 2
2 + 3 + 1 + 2 = 8

Output

□	□	□	□	a	a	a	a	a	a	a	a	□