Exercise 8.2.1

Show the IDs of the Turing machine of figure 8.9 if the input tape contains the following:

(a) 00

$q_0 00 \mapsto X q_1 0 \mapsto X 0 q_1 B$ without accepting.

(b) 000111

$q_0 000111 \vdash X q_1 00111 \vdash X 0 q_1 0111 \vdash X 00 q_1 111 \vdash X 0 q_2 0 Y 11 \vdash X q_2 00 Y 11 \vdash q_2 X 00 Y 11 \vdash X q_0 00 Y 11 \vdash XX q_1 0 Y 11 \vdash XX 0 q_1 Y 11 \vdash XX 0 Y q_1 11 \vdash XX 0 q_2 YY 1 \vdash XX q_2 0 YY 1 \vdash X q_2 X 0 YY 1 \vdash XX q_0 0 YY 1 \vdash XXX q_1 YY 1 \vdash XXXY q_1 Y 1 \vdash XXXYY q_1 1 \vdash XXXY q_2 YY \vdash XXX q_2 YYY \vdash XX q_2 XYYY \vdash XXX q_0 YYY \vdash XXXY q_3 YY \vdash XXXYY q_3 Y \vdash XXXYYY q_3 B \vdash XXXYYYB q_4 B$ and the machine halts in accepting state q_4.

(c) 00111

$q_0 00111 \vdash X q_1 0111 \vdash X 0 q_1 111 \vdash X q_2 0 Y 11 \vdash q_2 X 0 Y 11 \vdash X q_0 0 Y 11 \vdash XX q_1 Y 11 \vdash XXY q_1 11 \vdash XX q_2 YY 1 \vdash X q_2 XYY 1 \vdash q_2 XXYY 1 \vdash X q_0 XYY 1 \vdash XX q_0 YY 1 \vdash XXY q_3 Y 1 \vdash XXYY q_3 1$ and the machine halts in non-accepting state q_3.

Exercise 8.2.2

Design Turing machines for the following languages:

(a) The set of strings with an equal number of 0s and 1s.

We need states that search right for 0s and left for 1s. Also, we need to find the beginning and end of the string and, if no zeros are found, make sure that no ones are found either.

State	0	1	X	B
$\rightarrow q_0$	(q_1, X, R)	(q_2, X, R)	(q_0, X, R)	(q_4, B, L)
q_1	$(q_1, 0, R)$	(q_3, X, L)	(q_1, X, R)	(q_5, B, L)
q_2	(q_3, X, L)	$(q_2, 1, R)$	(q_2, X, R)	(q_5, B, L)
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	(q_3, X, L)	(q_0, B, R)
$*q_4$	–	–	–	–
q_5	–	–	–	–

The components of the Turing machine are $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, q_0 is the start state, q_4 is the accepting state, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, X, B\}$, the blank is B, and δ is given above.

(b) $\{a^n b^n c^n \mid n \geq 1\}$.

This will be much like the machine given in figure 8.9.

State	a	b	c	X	Y	Z	B
q_0	(q_1, X, R)	–	–	–	(q_5, Y, R)	–	–
q_1	(q_1, a, R)	(q_2, Y, R)	–	–	(q_1, Y, R)	–	–
q_2	–	(q_2, b, R)	(q_3, Z, L)	–	–	(q_2, Z, R)	–
q_3	(q_3, a, L)	(q_3, b, L)	–	(q_0, X, R)	(q_3, Y, L)	(q_3, Z, L)	–
q_5	–	–	–	–	(q_5, Y, R)	(q_6, Z, R)	–
q_6	–	–	–	–	–	(q_6, Z, R)	(q_π, B, R)
q_π	–	–	–	–	–	–	–

The components of the Turing machine are $Q = \{q_0, q_1, q_2, q_3, q_5, q_6, q_\pi\}$, q_0 is the starting state, q_π is the accepting state, $\Sigma = \{a, b, c\}$, $\Gamma = \{a, b, c, X, Y, Z, B\}$, the blank is B, and δ is given above.

(c) $\{ww^R \mid w \in (0+1)^*\}$

Here we need states that search for the left and right ends of the string, depending on whether a 0 or 1 was found at the beginning, then backtracking to see if a 0 or 1 is at the end.

State	0	1	B
q_0	(q_1,B,R)	(q_2,B,R)	(q_π,B,L)
q_1	$(q_1,0,R)$	$(q_1,1,R)$	(q_3,B,L)
q_2	$(q_2,0,D)$	$(q_2,1,D)$	(q_4,B,L)
q_3	(q_5,B,L)	–	–
q_4	–	(q_5,B,L)	–
q_5	$(q_5,0,L)$	$(q_5,1,L)$	(q_0,B,R)
q_π	–	–	–

The components of the Turing machine are $Q = \{q_0, q_1, q_2, q_3, q_5, q_\pi\}$, q_0 is the starting state, q_π is the accepting state, $\Sigma = \{0,1\}$, $\Gamma = \{0,1,B\}$, the blank is B, and δ is given above.

Exercise 8.4.1

Informally but clearly describe multi-tape Turing machines that accept each of the languages of Exercise 8.2.2. try to make each of your turing machines run in time proportional to the input length.

(a) The set of strings with an equal number of 0s and 1s. On the second tape, build a string that contains all of the 0s together and all of the 1s together. This can be done by writing from the inside out. Write the first character we see onto the second tape. Then, as we scan the first tape, if we see a 0, move the second tape head to the left until a blank is found, then write a 0. If we see a 1, move the second tape head to the right until a blank is found and then write a 1. While moving the second head, the first tape head is stationary. When the head on the first tape encounters a blank, then the second tape will contain a string of the form 0^n1^m. The machine will now ignore the first tape head and process the second. Move the second tape head to the left until a blank is found. This is the beginning of the string 0^n1^m. Make a subroutine call to the Turing machine in figure 8.9. If that machine accepts, then the first machine accepts.

(b) $\{a^nb^nc^n \mid n \geq 1\}$. Have four tapes. As long as the first head reads a, write that to the second tape and move the first two heads right, keeping the third and fourth stationary. When b is found, write that to the third head, move the first and third right but keep the second and fourth stationary. When c is found, write that to the fourth head, move the first and fourth to the right, keep the second and third stationary. When a blank is found, stop.

Now the second, third, and fourth tapes have all a, b, or c and all are at the right end of those strings. Then if each tape has an a, b, and c, respectively, move left. If one reaches a blank while another still reads a, b, or c, halt and fail. If all three heads read blank simultaneously, stop and accept. Having at least one of each character will be handled as the tape is scanned initially, as will reading the characters out of sequence. In those cases, halt and fail.

(c) $\{ww^R \mid w \in (0+1)^*\}$ Using two tapes, scan the first all the way to the first blank. Now start scanning left, writing to the second tape, moving the second head to the right. When the blank at the left appears (the beginning of the original string), move the second head all the way back to the left end. Now the second tape contains the reversal of the first tape. Compare character by character (i.e., $\delta(q,(0,0)) = (q,(0,0),R)$ but $\delta(q,(0,1)) = -$).

Oops, what about strings of odd length? As the string is being copied in reverse, use a *third* tape (the flag extension would work just as well) to alternately store 1 or 0 and move right (on the first character copied, write 1, then write 0 when the second character is copied,...). When the final character is copied, back the third tape up one character and if it is a 1, halt and fail. If a zero, ignore the third tape and compare the first two character by character as above. This would work with a flag by initializing it to 0, then toggling it with each character copied.

Exercise 8.4.5

Consider a nondeterministic TM whose tape is infinite in both directions. At some time, the tape is completely blank, except for one cell which holds the symbol \$. The head is currently at some blank cell and the state is q.

(a) Write transitions that will enable the machine to enter state p. That is, find the money.

State	B	\$
q	$\{(q_1, B, R), (q_2, B, L)\}$	$(p, \$, L)$
q_1	(q_1, B, R)	$(p, \$, R)$
q_2	(q_2, B, L)	$(p, \$, L)$
p	–	–

Why states q_1, q_2? So we don't get an exponential explosion in the number of machines. From the beginning, simply clone the machine, one searches left, one searches right. There's no sense in being wasteful.

(b) Suppose the TM were deterministic instead. How would you enable it to find the money and enter state p?

Write a non-blank, non-\$ character, say, π, to the starting spot. Move right until a blank or \$ is found. If found, go to p. If not, write π and move left until a blank or \$ is found. If found, go to p. If not, write π and move right until a blank or \$ is found. Repeat.