Problem 3.2 (b,d) (3 pts each)
Part b:
Use TM $M_1$ from Example 3.9 on input: "1#1"
$q_11#1 \rightarrow xq_3#1 \rightarrow x\#q_51\rightarrow xq_6\#x \rightarrow q_7x\#x \rightarrow xq_1\#x \rightarrow x\#q_8x \rightarrow x\#xq_8 \uparrow \rightarrow x\#x \uparrow q_{accept}$

Part d:
Use TM $M_1$ from Example 3.9 on input: "10#11"
$q_{1}10#11 \rightarrow xq_{3}0#11 \rightarrow x0q_{3}#11 \rightarrow x0q_{5}11 \rightarrow x0q_{6}#x1 \rightarrow x0q_{6}#x1 \rightarrow xq_{7}0#x1 \rightarrow q_{7}x0#x1 \rightarrow xq_{1}0#x1 \rightarrow xxq_{2}#x1 \rightarrow xx\#q_{4}x1 \rightarrow xx\#xq_{4}1 \rightarrow xx\#x1 q_{reject}$

Problem 3.8 (b,c) (4 pts each)
(describe how the tape is changed and how the head moves)
Part b: $\{w \mid w$ contains twice as many 0s as 1s\}
$M = \text{"On input } w:\$
1. Scan the tape and mark the first 1 which has not been marked. If no unmarked 1s are found, go to stage 5. Otherwise move the head back to the start of the tape.
2. Scan the tape and mark the first 0 that has not been marked. If no unmarked 0 remains, then reject.
3. Scan the tape and mark the next 0 that is unmarked. If no unmarked 0 remains, then reject.
4. Move the head back to the front of the tape and return to stage 1.
5. Move the head back to the start and scan the tape to see if any 0 remains unmarked. If there is an unmarked 0 reject, otherwise accept.”

Part c: $\{w \mid w$ does not contain twice as many 0s as 1s\}
$M = \text{"On input } w:\$
1. Scan the tape and mark the first 1 which has not been marked. If no unmarked 1s are found, go to stage 5. Otherwise move the head back to the start of the tape.
2. Scan the tape and mark the first 0 that has not been marked. If no unmarked 0 remains, then accept.
3. Scan the tape and mark the next 0 that is unmarked. If no unmarked 0 remains, then accept.
4. Move the head back to the front of the tape and return to stage 1.
5. Move the head back to the start and scan the tape to see if any 0 remains unmarked. If there is an unmarked 0 accept, otherwise reject.”

Problem 3.15 (b,d) (4 pts each)
Show the collection of decidable languages is closed under:
Part b: concatenation
For any two decidable languages $L_1$ & $L_2$ let $M_1$ & $M_2$ be the Turing machines that decide them.
Construct the TM $M$ that decides the concatenation of $L_1$ and $L_2$.
$M = \text{"On input } w:\$
1. Let $i = 0$;
2. Let $x$ be the prefix of $w$ with $i$ symbols and $y$ be the rest of $w$, i.e., $w = xy$:
3. Run $M_1$ on $x$. If it rejects then go to stage 5. Otherwise if it accepts, then continue.
4. Run $M_2$ on $y$. If it accepts, then accept.
5. Let $i = i+1$; if $i>|w|$ then reject. Otherwise, go to stage 2.”
If there exists a way to split the input $w$ into two substrings where $M_1$ accepts the first part and $M_2$ accepts the second part, $w$ will belong to the concatenation of $L_1$ & $L_2$ and $M$ will accept $w$ after a finite number of steps.

**Part d: complementation**

For any decidable language $L$, let $M_1$ be the Turing machine that decides it. Construct the TM $M$ that decides the complement of $L$.

$M = \text{"On input } w:\$

1. Run $M_1$ on $w$. If $M_1$ accepts then reject; if $M_1$ rejects then accept.

Since $M$ does the opposite of whatever $M_1$ does, $M$ decides the complement of $L$.

**Problem 3.16 (c,d) (4 pts each)**

Show the collection of Turing-recognizable languages is close under:

**Part c: star**

For any Turing-recognizable language $L$, let $M_1$ be the TM that recognizes it. Construct TM $M$ that recognizes the start of $L$.

$M = \text{"On input } w:\$

1. If $w = \epsilon$, run $M_1$ on $w$. If $M_1$ accepts, accept.
2. Nondeterministically split $w$ into non-empty pieces so that $w = w_1 w_2 w_3 \ldots w_n$.
3. Run $M_1$ on $w_i$ for all $i$. If $M_1$ accepts all pieces, then accept. If it halts on any and rejects, then reject.”

If there is a way to split $w$ into substrings such that $M_1$ accepts all the substrings, $w$ must belong to the star of $L$ and $M$ will accept $w$ after a finite number of steps.

**Part d: intersection**

For any two Turing-recognizable languages $L_1$ and $L_2$, let $M_1$ and $M_2$ be the TMs that recognize them. Construct TM $M$ that recognizes the intersection of $L_1$ and $L_2$.

$M = \text{"On input } w:\$

1. Run $M_1$ on $w$. If it rejects, then reject. Otherwise continue.
2. Run $M_2$ on $w$. If it rejects, then reject. If it accepts, then accept.

If both $M_1$ and $M_2$ accept input $w$, $w$ belongs to the intersection of $L_1$ and $L_2$ and $M$ will accept it after a finite number of steps.