Different Turing Machine Models

1. One way infinite tape

M = (Q, X, q₀, F, S) as before.
No successor ID is defined for q X w when S(q, x) = (q', x', L).

2. Two way infinite tape

what we defined previously.
Theorem: L is recognized by a Turing machine with a two-way infinite tape if and only if it is recognized by a TM with a one-way infinite tape.

Proof sketch:

\( \Leftarrow \) clear

\( \Rightarrow \)

Basic idea: Let \( M_2 \) be the machine with the two-way infinite tape that recognizes \( L \). We will construct a TM \( M \), with a one-way infinite tape that recognizes \( L \). \( M \) will have two "tracks", one representing the
cells of $M_2$'s tape to the right of, and including, the tape cell initially scanned, the other representing (in reverse order) the cells to the left of the initial cell.

\[ \ldots a_3 a_1 a_0 a_1 a_2 a_3 \ldots \]

$M_2$'s tape

\[
\begin{array}{c|cccc}
& a_0 & a_1 & a_2 & a_3 \\
\hline
\$ & a_1 & a_2 & a_3 & \ldots \\
\end{array}
\]

$M_1$'s tape

$M_1$ is constructed to simulate $M_2$ so that when the read head of $M_2$ is to the right of its initial
position, $M_1$ uses its "upper track," and when the read head of $M_2$ is to the left of its initial position, $M_1$ works on its lower track and moves in the direction opposite to which $M_2$ moves.

If the initial tape of $M_2$ is

| $\ldots$ | B | B | B | $a_0$ | $a_1$ | $a_2$ | $a_3$ | $\ldots$ |

Then the initial tape of $M_1$ will be

<table>
<thead>
<tr>
<th>$a_0$</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$\ldots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$$</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>$\ldots$</td>
</tr>
</tbody>
</table>

(The details of the construction are given in your text).
3. Multi-tape Turing Machines

On a single move, depending on the state of its finite control and the symbol scanned on each tape, the machine can:

1. Change state
2. Print a new symbol on each of the cells scanned by its
tape heads, and

(3) more each of its tape heads, one all
in the left or right or remain
stationary.

Initially, the input appears on the first
tape and the other tapes are blank.

Theorem: If a language $L$ is accepted
by a multitape Turing machine, it
is accepted by a single tape Turing
machine.

Proof sketch: If $L$ is accepted
by a machine $M$, with $K$ tapes,
we will construct a one-tape $T_M$. 
M2 with 2K tracks, two tracks for each of M1's tapes. One track records the contents of the corresponding track of M1; the other records the position of the read head for that track.

<table>
<thead>
<tr>
<th>head1</th>
<th>X</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>tape1</td>
<td>a1</td>
<td>a2</td>
<td>a3</td>
<td>a4</td>
</tr>
<tr>
<td>head2</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>tape2</td>
<td>b1</td>
<td>b2</td>
<td>b3</td>
<td>b4</td>
</tr>
<tr>
<td>head3</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tape3</td>
<td>c1</td>
<td>c2</td>
<td>c3</td>
<td>c4</td>
</tr>
</tbody>
</table>

This configuration of M2's tape indicates that M1 is scanning a2 on tape 1, b4 on tape 2, and c1 on tape 3. The procedure that M2 uses to simulate a move of M1.
is straightforward, but tedious to describe formally. (Again, see your text for details).

4. Multihead Turing Machines

A $K$-head Turing machine has some fixed number $K$ of heads. The heads are numbered 1 through $K$. A move of the TM depends on its current state and the symbol scanned by each of the $K$ read heads. In some more, the heads
may move independently left, right, or remain stationary.

Theorem: if L is accepted by some K-head TM M, it is accepted by a one head TM.

Proof ?

(5) Nondeterministic Turing Machines

Like basic one tape model, but

\[ S : Q \times X' \rightarrow 2^{Q \times X' \times \{L, R, N\}} \]

Theorem: if L is accepted by a nondeterministic Turing Machine, then L is accepted by some
deterministic Turing machine.

Proof Sketch:

Assume that $L$ is accepted by the nondeterministic Turing machine $M$.

For any state and tape symbol of $M$, there are a finite number of choices for the next move. Let $r$ be the maximum number of possible choices for any state-symbol pair.

Note that with each halting computation of $M$, it is possible to associate a string in $(1+2+\ldots+r)^*$ that uniquely determines that computation.
The deterministic machine $M_2$ will have three tapes. The first holds the input. The second is used to enumerate strings in $(1+2+\ldots+r)^*$. For each sequence generated on tape 2, $M_2$ will copy the input from tape 1 to tape 3 and use the sequence to determine the appropriate move of $M_1$. If $M_1$ enters an accepting state then $M_2$ will also. If there is an accepting computation of $M_1$, the sequence of moves in the computation will eventually be enumerated on tape 2 of $M_2$. 
Church's Thesis

Minsky: "Any process which could naturally be called an effective procedure can be realized by a Turing Machine."

Boolos & Jeffrey: "Any mechanical routine for symbol manipulation can be carried out in effect by some Turing machine."

Kleene: "Every function which would naturally be regarded as computable is computable by a Turing Machine."

Hartmanis: "If an alg. \( \Leftrightarrow \exists T. M. \)"