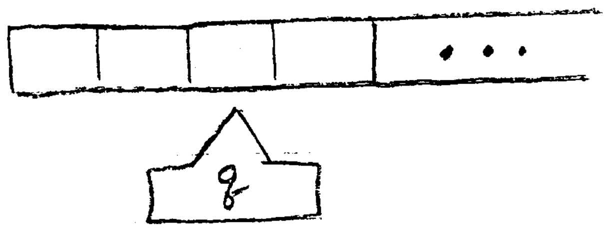
Different Turing Machine Models

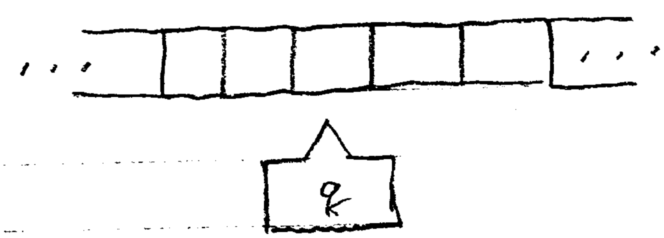
1. One way infinite tape



$M = (Q, X, q_0, F, \delta)$ as before.

No successor ID is defined for $q\,x\,w$ when $\delta(q, x) = (q', x', L)$.

2. two way infinite tape



What we defined previously.

<u>Theorem</u>: 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:

($\Leftarrow$) <u>clear</u>

($\Rightarrow$)

Basic idea: Let M_2 be the machine with the two-way infinite tape that recognizes L. We will construct a TM M_1 with a one way infinite tape that recognizes L. M_1 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 the initial cell.

...	a_{-3}	a_{-1}	a_{-1}	a_0	a_1	a_2	a_3	...

M_2's tape

a_0	a_1	a_2	a_3	...
\$	a_{-1}	a_{-2}	a_{-3}	...

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

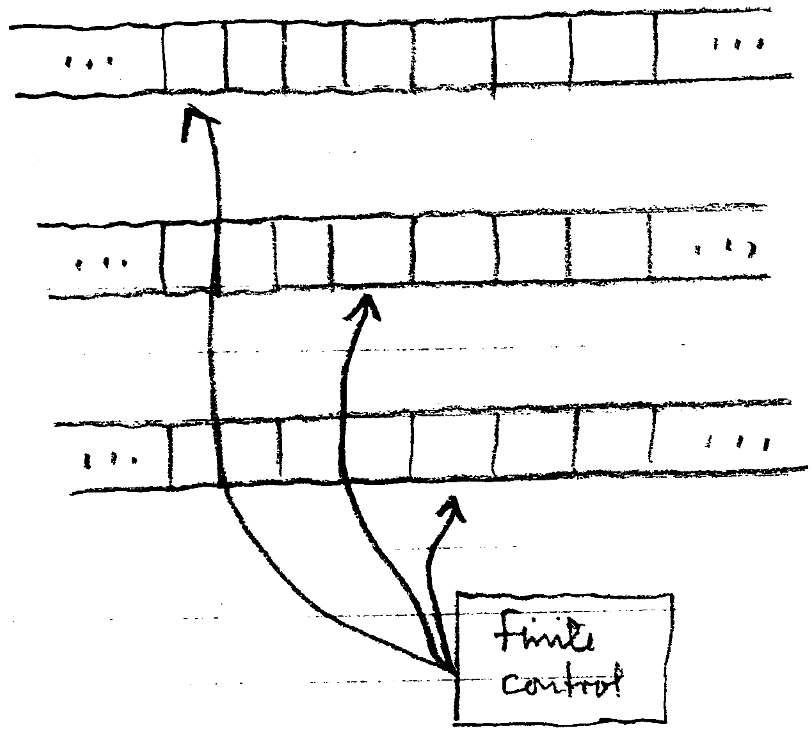
...	B	B	B	a_0	a_1	a_2	a_3	...

then the initial tape of M_1 will be

a_0	a_1	a_2	a_3	...
\$	B	B	B	...

(The details of the construction are given in your text).

3. Multitape 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) move each of its tape heads, one all to 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_1 with K tapes, we will construct a one tape TM

M_2 with $2K$ tracks, two tracks for each of M_1's tapes. One Track records the contents of the corresponding track of M_1; the other records the position of the read head for that track.

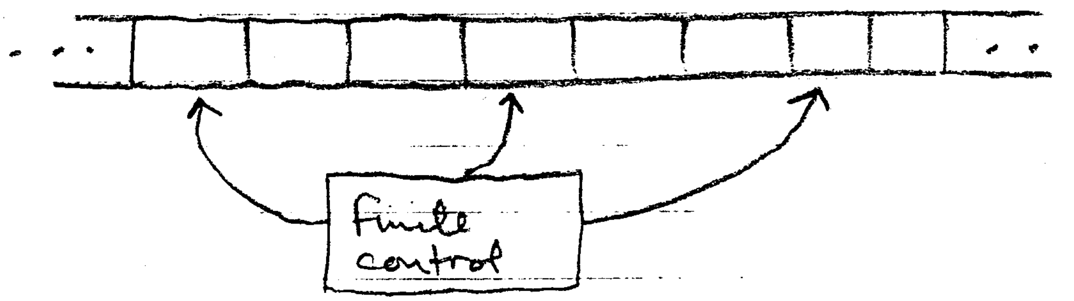
head 1		X			
tape 1	a_1	a_2	a_3	a_4	a_5
head 2				X	
tape 2	b_1	b_2	b_3	b_4	b_5
head 3	X				
tape 3	c_1	c_2	c_3	c_4	c_5

...

This configuration of M_2's tape indicates that M_1 is scanning a_2 on tape 1, b_4 on tape 2, and c_1 on tape 3. The procedure that M_2 uses to simulate a move of M_1

is straight forward, 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 one move, the heads may move independently left, right, or remain stationary.

Theorem: If L is accepted by some K-head TM M_1, it is accepted by a one head TM.

Proof ?

(5) Nondeterministic Turing Machines

Like basic one tape model, but

$$\delta : Q \times X' \longrightarrow 2^{Q \times X' \times \{L, R, N\}}.$$

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_1. For any state and tape symbol of M_1, 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_1, it is possible to associate a string in $(1+2+\dots r)^*$ 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 : "$\exists$ an alg. $\Longleftrightarrow$ $\exists$ T.M."