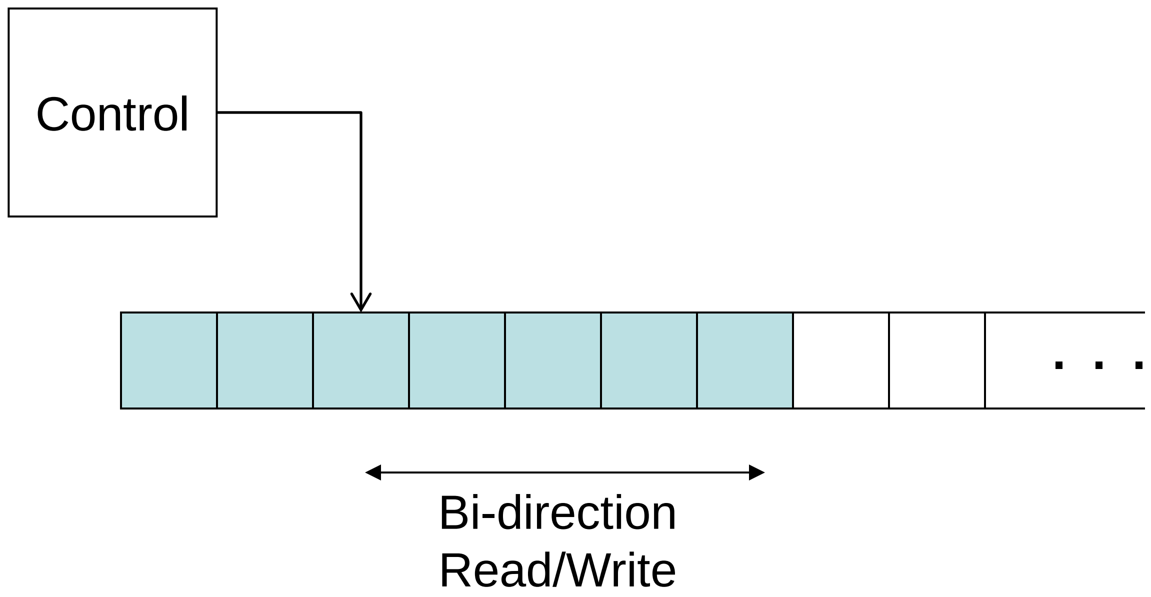
Chapter 3: The Church-Turing Thesis

Turing Machine (TM)



Turing machine is a much more powerful model, proposed by Alan Turing in 1936.

Church/Turing Thesis

Anything that an algorithm can compute can be computed by a Turing machine and vice-versa.

Turing Machine (cont'd)

- The differences between finite automata and Turing machines
  - A Turing machine can both write on the tape and read from it,
  - The read-write head can move both to the left and to the right,
  - The tape is infinite,
  - The special states for rejecting and accepting take effects immediately.

Turing Machine Definition

A deterministic Turing machine is a 7-tuple:

$(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where Q, Σ, Γ are finite sets, and

1. Q: set of states
2. Σ: input alphabet not containing the blank symbol ⊔
3. Γ: tape alphabet, including Σ and the blank symbol
4. δ: $Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function
5. q_0: $q_0 \in Q$ is the start state,
6. q_{accept}: $q_{accept} \in Q$ is the accept state
7. q_{reject}: $q_{reject} \in Q$ is the reject state

Non-Deterministic Turing Machine

A non-deterministic Turing machine is a 7-tuple:
$(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where Q, Σ, Γ are finite sets, and

1. Q: set of states
2. Σ: input alphabet not containing the blank symbol $\sqcup$
3. Γ: tape alphabet, including Σ and the blank symbol
4. $\delta: Q \times \Gamma \longrightarrow P(Q \times \Gamma \times \{L, R\})$ is the transition function
5. q_0: $q_0 \in Q$ is the start state,
6. q_{accept}: $q_{accept} \in Q$ is the accept state
7. q_{reject}: $q_{reject} \in Q$ is the reject state

TM for {w#w| w∈{0, 1}* }

M = " On input string *w*:

1.Check whether the string contains exactly one #. If no, *reject*;

2.Check whether the string contains only 0's and 1's besides the #. If there are other symbols, *reject*;

3.Zig-zag across the tape to corresponding positions on either side of the # symbol to check whether these positions contain the same symbol. If they do not, *reject*;

4.Cross off symbols as they are checked to keep track of which symbols correspond;

5.When all symbols to the left of the # have been crossed off, check for any remaining symbols to the right of the #. If any symbols remain, *reject*; otherwise, *accept*."

TM for $\{0^n1^n \mid n > 0\}$

M= “on input string w:

1. Look for 0’s from the left end of the tape. Only blank cells are allowed to pass.
2. If 0 found, change it to x and move right, else reject
3. Scan (to the right) passing 0’s and y’s until reach 1
4. If 1 found, change it to y and move left, else reject.
5. Move left passing y’s and 0’s
6. If x found move right
7. If 0 found, loop back to step 2.
8. If 0 not found, scan to the right passing y’s and accept; otherwise reject.”

TM for $\{0^n1^n2^n \mid n > 0\}$

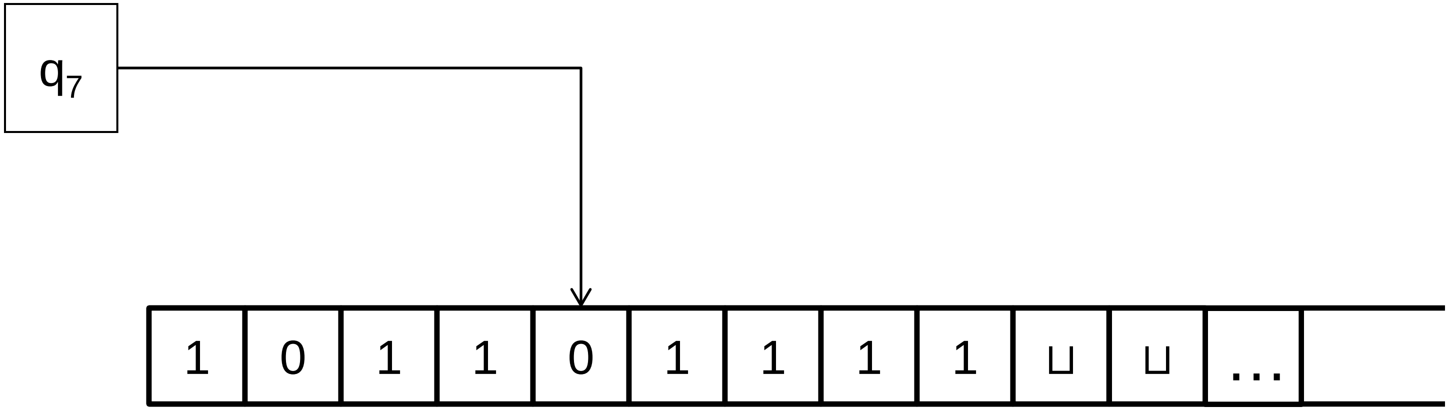
M = “on input string w:

1. Look for 0’s from the left end of the tape. Only blank cells are allowed to pass.
2. If 0 found, change it to x and move right, else reject
3. Scan to the right passing 0’s and y’s until reach 1
4. If 1 found, change it to y and move right, else reject.
5. Scan to the right passing 1’s and z’s until reach 2
6. If 2 found, change it to z and move left, else reject.
7. Scan to the left passing z’s, 1’s, y’s, and 0’s,
8. If x found move right
9. If 0 found, loop back to step 2.
10. If 0 not found, scan to the right passing only y’s and z’s and accept. Otherwise reject.

Configurations

- A TM configuration is a setting of three items: current state, current tape contents, and the current head location, i.e. uqv, where u and v are substrings and q is a state.
- Configuration C_1 yields configuration C_2 if the Turing machine can legally go from C_1 to C_2 in a single step.
  - Suppose that a, b, and c in Γ, u and v in Γ^* and states q_i and q_j. uaq_ibv and uq_jacv are two configurations.
    - uaq_ibv yields uq_jacv if $\delta(q_i, b) = (q_j, c, L)$.
      - handles the case where the TM moves leftward.
    - For a rightward move, uaq_ibv yields $uacq_jv$ if $\delta(q_i, b) = (q_j, c, R)$.

Configurations (cont'd)



A Turing machine with configuration $1011q_701111$

Configurations: Special Cases

- Special Cases occur when the head is at the left-hand end or the right-hand end of the configurations.
  - For the left-hand end, the configurations $q_i bv$ yields $q_j cv$ if the transition is left-moving: $\delta(q_i, b) = (q_j, c, L)$.
  - For the right-hand end, the configuration uaq_i is equivalent to $uaq_i\sqcup$ because we assume that blanks follow the part of the tape represented in the configuration. Thus we can handle this case as before, with the head no longer at the right-hand end.

Configurations (cont'd)

- The start configuration of M on input w is the configuration q_0w, which indicates that the machine is in the start state q_0 with its head at the leftmost position on the tape.
- In an accepting configuration, the state of the configuration is $q_{accept.}$
- In a rejecting configuration, the state of the configuration is $q_{reject.}$
- Accepting and rejecting configurations are halting configurations and do not yield further configurations.

L(M): the language of TM M

- A TM M accepts input w if a sequence of configurations $C_1, C_2, \ldots, C_k$ exists, where
  1. C_1 is the start configuration of M on input w,
  2. Each C_i yields C_{i+1}, and
  3. C_k is an accepting configuration.
- The collection of strings that M accepts is the language of M, denoted L(M).

Turing decidable/recognizable

- Turing-recognizable
  - A language is Turing-recognizable if some Turing machine recognizes it.
  - i.e., TM M recognizes language L if L = {w | M accepts w}.
  - *Note*: 3 outcomes possible, either TM accepts, rejects, or loops on a string.

Turing decidable/recognizable (cont'd)

- Turing-decidable:
  - A language is Turing-decidable if some Turing machine decides it.
  - TM M decides L if
    - (i) w $\in$ L, M accepts it.
    - (ii) w $\notin$ L, M rejects it.

Note: Every decidable language is Turing-recognizable but certain Turing-recognizable language are not decidable.

TM for $\{0^{2^n} \mid n \geq 0\}$

- TM M= "on input string *w*:
  1. Sweep left to right across the tape, crossing off every other 0,
  2. If in step 1, the tape contained a single 0, accept,
  3. If in step 1, the tape contained more than a single 0 and the number of 0s was odd, reject,
  4. Return the head to the left-hand end of the tape,
  5. Go to step 1."

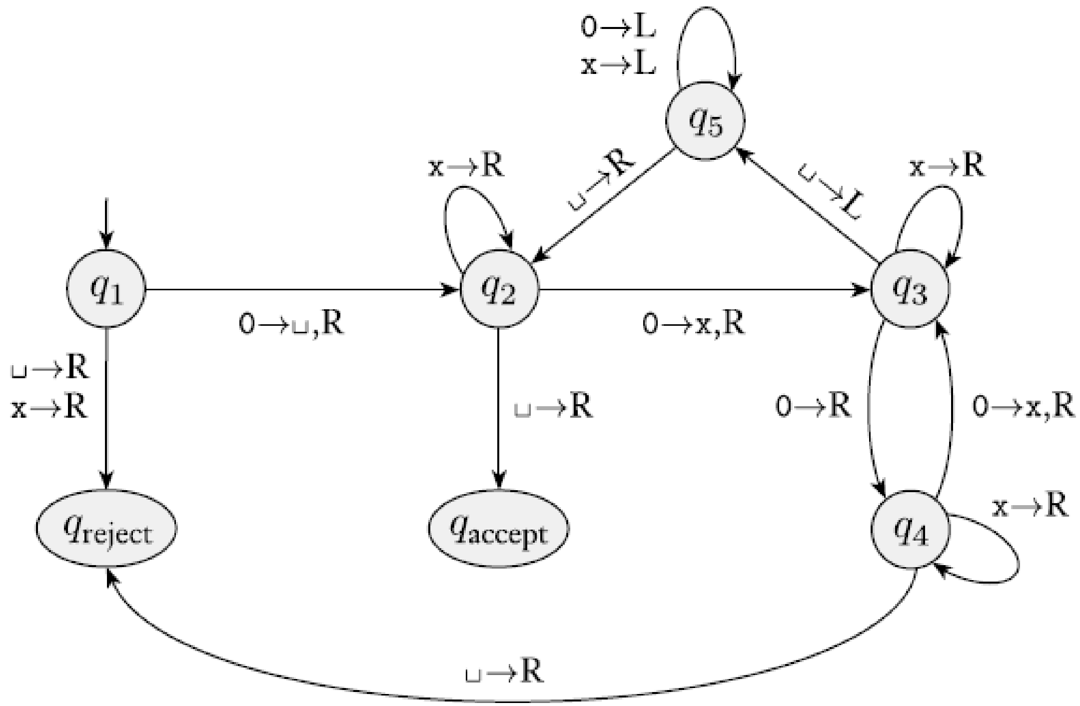


FIGURE 3.8
State diagram for Turing machine M_2

$q_1\mathtt{0000}$
$\sqcup q_2\mathtt{000}$
$\sqcup\mathtt{x}q_3\mathtt{00}$
$\sqcup\mathtt{x0}q_4\mathtt{0}$
$\sqcup\mathtt{x0x}q_3\sqcup$
$\sqcup\mathtt{x0}q_5\mathtt{x}\sqcup$
$\sqcup\mathtt{x}q_5\mathtt{0x}\sqcup$
$\sqcup q_5\mathtt{x0x}\sqcup$
$q_5\sqcup\mathtt{x0x}\sqcup$
$\sqcup q_2\mathtt{x0x}\sqcup$
$\sqcup\mathtt{x}q_2\mathtt{0x}\sqcup$
$\sqcup\mathtt{xx}q_3\mathtt{x}\sqcup$
$\sqcup\mathtt{xxx}q_3\sqcup$
$\sqcup\mathtt{xx}q_5\mathtt{x}\sqcup$
$\sqcup\mathtt{x}q_5\mathtt{xx}\sqcup$
$\sqcup q_5\mathtt{xxx}\sqcup$
$q_5\sqcup\mathtt{xxx}\sqcup$
$\sqcup q_2\mathtt{xxx}\sqcup$
$\sqcup\mathtt{x}q_2\mathtt{xx}\sqcup$
$\sqcup\mathtt{xx}q_2\mathtt{x}\sqcup$
$\sqcup\mathtt{xxx}q_2\sqcup$
$\sqcup\mathtt{xxx}\sqcup q_{\text{accept}}$

The figures are taken from the book *Introduction to Theory of Computation, Michael Sipser,* page 172.

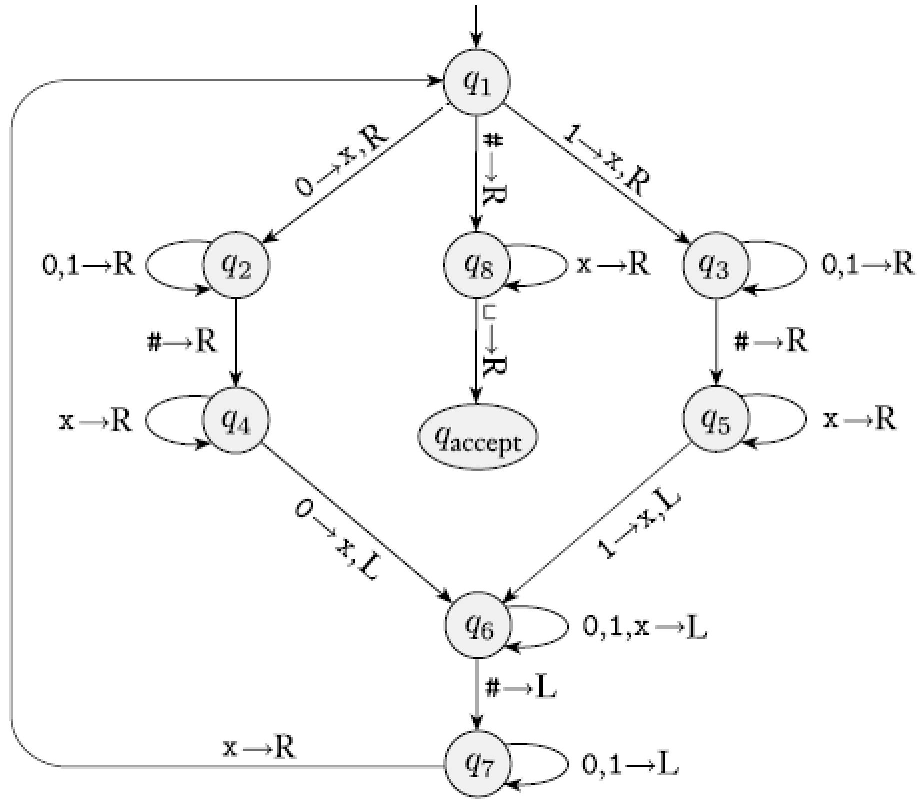


FIGURE 3.10
State diagram for Turing machine M_1

The figures are taken from the book *Introduction to Theory of Computation, Michael Sipser,* page 173.

Variants of TM models

- Variants of TM model (not more powerful than basic TM model)
  - Multitape Turing machine
    - Like ordinary TM but with several tapes.
    - Every multitape Turing machine has an equivalent single-tape Turing machine

Variants of TM models (cont'd)

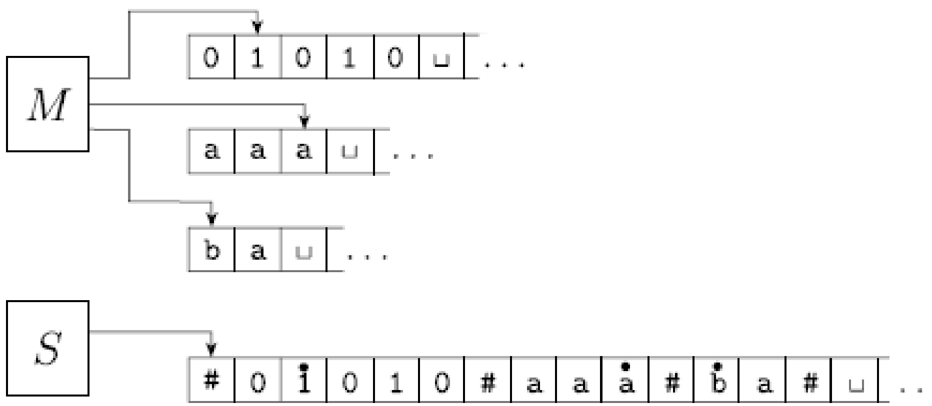


FIGURE 3.14
Representing three tapes with one

This figure is taken from the book *Introduction to Theory of Computation, Michael Sipser,* page 177.

Variants of TM models (cont'd)

- Nondeterministic Turing machine
  - Every nondeterministic Turing machine has a deterministic Turing machine

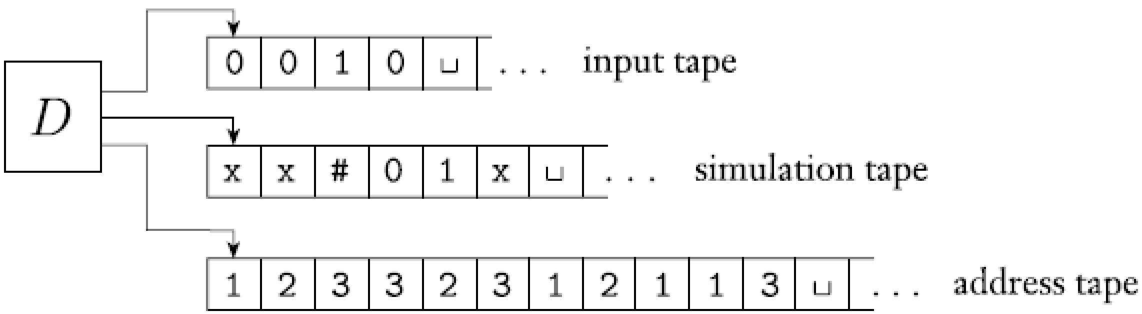


FIGURE 3.17
Deterministic TM D simulating nondeterministic TM N

This figure is taken from the book *Introduction to Theory of Computation, Michael Sipser,* page 179.