Turing Machine

Guan-Shieng Huang

Sep. 27, 2006

Turing Machine

K

$\triangleright 0111000a \cdots 01bb \square \square \square \square \cdots \cdots$

Definition of TMs

A Turing Machine is a quadruple $M = (K, \Sigma, \delta, s)$, where

1. K is a finite set of states; (line numbers)

2. Σ is a finite set of symbols including $\sqcup$ and $\triangleright$; (alphabet)

3. $\delta : K \times \Sigma \rightarrow (K \cup \{\text{h}, \text{"yes"}, \text{"no"}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$, a transition function; (instructions)

4. $s \in K$, the initial state. (starting point)

- h: halt, "yes":accept, "no": reject
  (terminate the execution)

- →: move right, ←: move left, −: stay
  (move the head)

- ⊔: blank, ▷: the boundary symbol

- $\delta(q, \sigma) = (p, \rho, D)$

  While reading σ at line q, go to line p and write out ρ on the tape. Move the head according to the direction of D.

- $\delta(q, \rhd) = (p, \rho, \rightarrow)$, to avoid crash.

Example 2.1

$p \in K,$	$\sigma \in \Sigma$	$\delta(p,\sigma)$
$s,$	0	$(s,0,\rightarrow)$
$s,$	1	$(s,1,\rightarrow)$
$s,$	⊔	$(q,⊔,\leftarrow)$
$s,$	▷	$(s,▷,\rightarrow)$
$q,$	0	$(q_0,⊔,\rightarrow)$
$q,$	1	$(q_1,⊔,\rightarrow)$
$-q,$	⊔	$(q,⊔,-)$
$q,$	▷	$(h,▷,\rightarrow)$
$q_0,$	0	$(s,0,\leftarrow)$
$q_0,$	1	$(s,0,\leftarrow)$
$q_0,$	⊔	$(s,0,\leftarrow)$
$-q_0,$	▷	$(h,▷,\rightarrow)$
$q_1,$	0	$(s,1,\leftarrow)$
$q_1,$	1	$(s,1,\leftarrow)$
$q_1,$	⊔	$(s,1,\leftarrow)$
$-q_1,$	▷	$(h,▷,\rightarrow)$

— halt

0.	$s,$	▷010
1.	$s,$	▷010
2.	$s,$	▷010
3.	$s,$	▷010
4.	$s,$	▷010⊔
5.	$q,$	▷010⊔
6.	$q_0,$	▷01 ⊔ ⊔
7.	$s,$	▷01⊔0
8.	$q,$	▷01 ⊔ 0
9.	$q_1,$	▷0 ⊔ ⊔0
10.	$s,$	▷0⊔10
11.	$q,$	▷0 ⊔ 10
12.	$q_0,$	▷ ⊔ ⊔10
13.	$s,$	▷⊔010
14.	$q,$	▷ ⊔ 010
15.	$h,$	▷⊔010

Figure 2.1. Turing machine and computation.

Remark

x: input of M

$$M(x) = \begin{cases} \text{``yes''} \\ \text{``no''} \\ y \text{ if } M \text{ entered } h \\ \nearrow \text{ if } M \text{ never terminates} \end{cases}$$

Example 2.2

$(n)_2 \rightarrow (n+1)_2$ if no overflow happens.

$p \in K, \quad \sigma \in \Sigma$		$\delta(p, \sigma)$
$s,$	0	$(s, 0, \rightarrow)$
$s,$	1	$(s, 1, \rightarrow)$
$s,$	$\sqcup$	$(q, \sqcup, \leftarrow)$
$s,$	$\triangleright$	$(s, \triangleright, \rightarrow)$
$q,$	0	$(h, 1, -)$
$q,$	1	$(q, 0, \leftarrow)$
$q,$	$\triangleright$	$(h, \triangleright, \rightarrow)$

Figure 2.2. Turing machine for binary successor.

Example 2.3 — Palindrome

$p \in K,\quad \sigma \in \Sigma$		$\delta(p,\sigma)$
s	0	$(q_0, \triangleright, \rightarrow)$
s	1	$(q_1, \triangleright, \rightarrow)$
s	$\triangleright$	$(s, \triangleright, \rightarrow)$
s	$\sqcup$	$(\text{``yes''}, \sqcup, -)$
q_0	0	$(q_0, 0, \rightarrow)$
q_0	1	$(q_0, 1, \rightarrow)$
q_0	$\sqcup$	$(q_0', \sqcup, \leftarrow)$
q_1	0	$(q_1, 0, \rightarrow)$
q_1	1	$(q_1, 1, \rightarrow)$
q_1	$\sqcup$	$(q_1', \sqcup, \leftarrow)$

$p \in K,\quad \sigma \in \Sigma$		$\delta(p,\sigma)$
q_0'	0	$(q, \sqcup, \leftarrow)$
q_0'	1	$(\text{``no''}, 1, -)$
q_0'	$\triangleright$	$(\text{``yes''}, \sqcup, \rightarrow)$
q_1'	0	$(\text{``no''}, 1, -)$
q_1'	1	$(q, \sqcup, \leftarrow)$
q_1'	$\triangleright$	$(\text{``yes''}, \triangleright, \rightarrow)$
q	0	$(q, 0, \leftarrow)$
q	1	$(q, 1, \leftarrow)$
q	$\triangleright$	$(s, \triangleright, \rightarrow)$

Figure 2.3. Turing machine for palindromes.

Turing Machines as Algorithms

- $L \subseteq (\Sigma - \{\sqcup, \triangleright\})^*$, a language

- A TM M decides L if for all string x,
$$\begin{cases} x \in L \Rightarrow M(x) = \text{``yes''} \\ x \notin L \Rightarrow M(x) = \text{``no''}. \end{cases}$$

- A TM M accepts L if for all string x,
$$\begin{cases} x \in L \Rightarrow M(x) = \text{``yes''} \\ x \notin L \Rightarrow M(x) = \nearrow . \end{cases}$$

k-string Turing Machines

A k-string Turing machine is a quadruple (K, Σ, δ, s) where

1. K, Σ, s are exactly as in ordinary Turing machines;

2. $\delta : K \times \Sigma^k \to (K \cup \{\text{h}, \text{``yes''}, \text{``no''}\}) \times (\Sigma \times \{\leftarrow, \rightarrow, -\})^k$;

An Example

$p \in K,$	$\sigma_1 \in \Sigma$	$\sigma_2 \in \Sigma$	$\delta(p, \sigma_1, \sigma_2)$
$s,$	0	⊔	$(s, 0, \rightarrow, 0, \rightarrow)$
$s,$	1	⊔	$(s, 1, \rightarrow, 1, \rightarrow)$
$s,$	▷	▷	$(s, \triangleright, \rightarrow, \triangleright, \rightarrow)$
$s,$	⊔	⊔	$(q, \sqcup, \leftarrow, \sqcup, -)$
$q,$	0	⊔	$(q, 0, \leftarrow, \sqcup, -)$
$q,$	1	⊔	$(q, 1, \leftarrow, \sqcup, -)$
$q,$	▷	⊔	$(p, \triangleright, \rightarrow, \sqcup, \leftarrow)$
$p,$	0	0	$(p, 0, \rightarrow, \sqcup, \leftarrow)$
$p,$	1	1	$(p, 1, \rightarrow, \sqcup, \leftarrow)$
$p,$	0	1	$(\text{"no"}, 0, -, 1, -)$
$p,$	1	0	$(\text{"no"}, 1, -, 0, -)$
$p,$	⊔	▷	$(\text{"yes"}, \sqcup, -, \triangleright, \rightarrow)$

Figure 2.5. 2-string Turing machine for palindromes.

1. If for a k-string Turing machine M and input x we have

$$(s, \triangleright, x, \triangleright, \epsilon, \ldots, \triangleright, \epsilon) \xrightarrow{M^t} (H, w_1, u_1, \ldots, w_k, u_k)$$

   for some $H \in \{\text{h}, \text{"yes"}, \text{"no"}\}$, then the time required by M on input x is t.

2. If for any input string x of length $|x|$, M terminates on input x within time $f(|x|)$, we say $f(n)$ is a time bound for M.

(worst case analysis)

TIME($f(n)$): the set of all languages that can be decided by TMs in time $f(n)$.

Theorem 2.1

Given any k-string TM M operating within time $f(n)$, we can construct a TM M' operating within time $O(f(n)^2)$ and such that, for any input x, $M(x) = M'(x)$.

(by simulation)

Space Bounds

A k-string TM with input and output is an ordinary k-string TM s.t.

1. the first tape is read-only;
   (Input cannot be modified.)

2. the last tape is write-only.
   (Output cannot be wound back.)

Space Bound for TM

Suppose that, for a k-string TM M and input x,

$$(s, \triangleright, x, \ldots, \triangleright, \epsilon) \xrightarrow{M^*} (H, w_1, u_1, \ldots, w_k, u_k)$$

where $H \in \{\text{h}, \text{"yes"}, \text{"no"}\}$ is a halting state.

1. The space required by M on input x is $\sum_{i=1}^{k} |w_i u_i|$.

2. If M is a machine with input and output, then the space required by M on input x is $\sum_{i=2}^{k-1} |w_i u_i|$.

15

1. We say that Turing machine M operates within space bound $f(n)$ if, for any input x, M requires space at most $f(|x|)$.

2. A language L is in the space complexity class $\mathrm{SPACE}(f(n))$ if there is a TM with I/O that decides L and operates within space bound $f(n)$.

3. Define $\mathcal{L} = \mathrm{SPACE}(\lg(n))$.

Random Access Machines

Input: $(i_1, i_2, \ldots, i_n)$

Output: r_0 (accumulator)

Memory: $r_0, r_1, r_2, \ldots$ (infinite memory)

k: program counter

Three address modes: (for x)

1. j: direct;

2. $\uparrow j$: indirect;

3. $= j$: immediate.

(arbitrary large number)

| Instruction	Operand	Semantics
READ	j	$r_0 := i_j$
READ	$\uparrow j$	$r_0 := i_{r_j}$
STORE	j	$r_j := r_0$
STORE	$\uparrow j$	$r_{r_j} := r_0$
LOAD	x	$r_0 := x$
ADD	x	$r_0 := r_0 + x$
SUB	x	$r_0 := r_0 - x$
HALF		$r_0 := \lfloor \frac{r_0}{2} \rfloor$
JUMP	j	$\kappa := j$
JPOS	j	**if** $r_0 > 0$ **then** $\kappa := j$
JZERO	j	**if** $r_0 = 0$ **then** $\kappa := j$
JNEG	j	**if** $r_0 < 0$ **then** $\kappa := j$
HALT		$\kappa := 0$

Theorem 2.5

If a RAM program Π computes a function ϕ in time $f(n)$, then there is a 7-string TM which computes ϕ in time $O(f(n)^3)$.

(by simulation)

Nondeterministic Machines

A nondeterministic TM is a quadruple $N = (K, \Sigma, \Delta, s)$, where

1. K, Σ, s are as in ordinary TM;

2. $\Delta \subseteq (K \times \Sigma) \times [(K \cup \{h, \text{``yes''}, \text{``no''}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}]$.
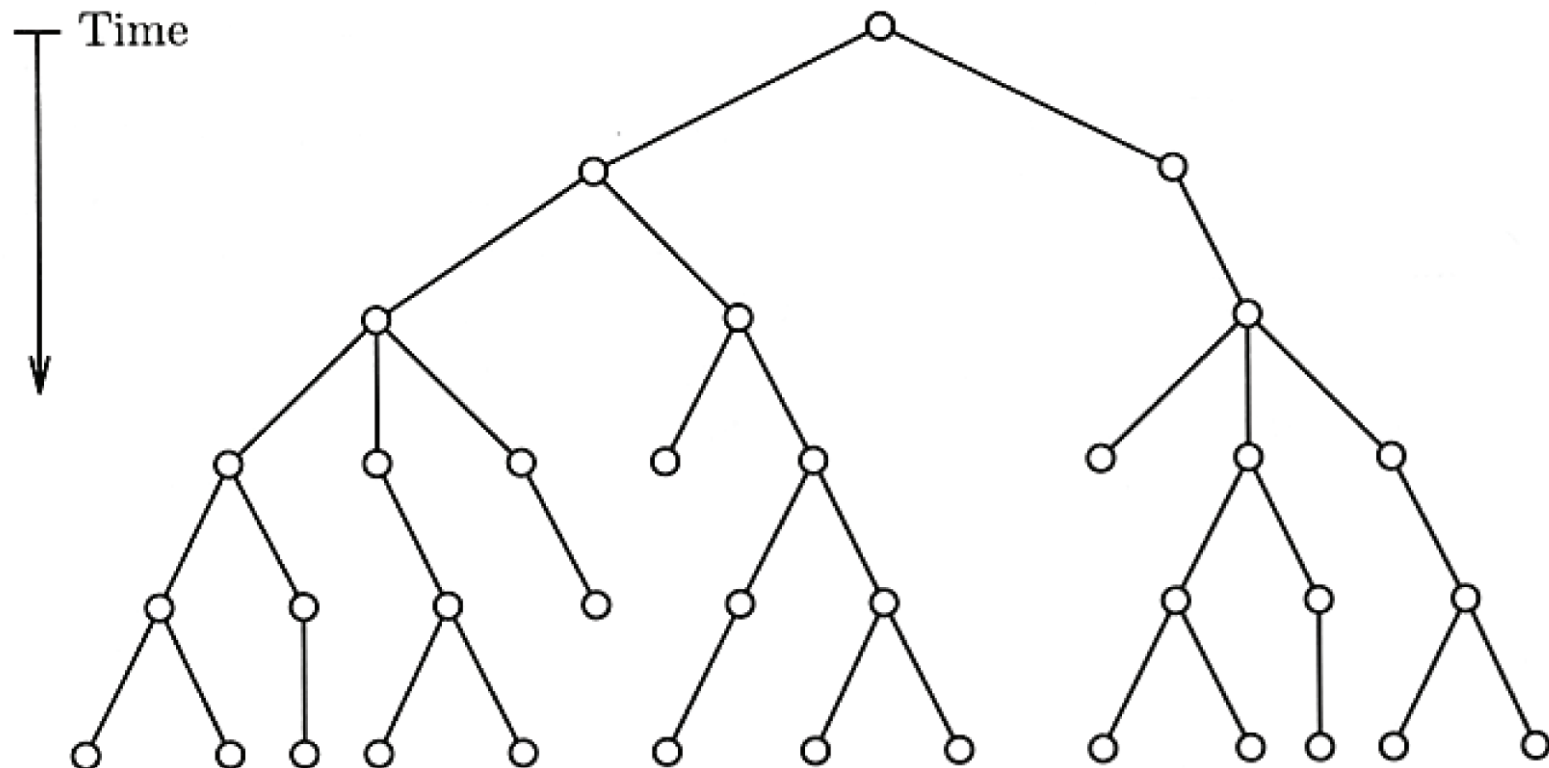
Figure 2-9. Nondeterministic computation.