Algorithm Theory

Georg Moser Mircea Dan Hernest

Institute of Computer Science @ UIBK

Summer 2007

Turing Machines

A (1-string) Turing machine (TM) M is a quadruple (K, Σ, δ, s)

1. K finite set of states
2. Σ finite alphabet (disjoint from K) contains always $\sqcup, \triangleright$
3. δ is the transition function

$$\delta \colon K \times \Sigma \to (K \cup \{\mathrm{h}, \mathit{yes}, \mathit{no}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$$

4. s is the initial state

Restriction: If $\delta(p, \triangleright) = (p, \rho, D)$, then $\rho = \triangleright$ and $D = \rightarrow$

Input x of M is written next to $\triangleright$.

- if M reaches h, it is halting
- the output y is the string of M at halting $\quad \mathrm{M}(x) = y$
- if it reaches *yes*, it accepts $\quad \mathrm{M}(x) = \mathit{yes}$
- if it reaches *no* it rejects $\quad \mathrm{M}(x) = \mathit{no}$

Example: TM for binary successor

$p \in K$	$\sigma \in \Sigma$	$\delta(p, \sigma)$
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	$\triangleright$	$(h, \triangleright, \rightarrow)$

- Question : What is the output of this TM on the input 11011?
- Anwer : 11100
- Question : What happens on input 1111?

Turing Machine for palindromes

$p \in K$	$\sigma \in \Sigma$	$\delta(p, \sigma)$
s	0	$(q_0, \triangleright, \rightarrow)$
s	1	$(q_1, \triangleright, \rightarrow)$
s	$\triangleright$	$(s, \triangleright, \rightarrow)$
s	$\sqcup$	$(yes, \sqcup, -)$
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	$\sqcup$	$(q'_1, \sqcup, \leftarrow)$

$p \in K$	$\sigma \in \Sigma$	$\delta(p, \sigma)$
q'_0	0	$(q, \sqcup, \leftarrow)$
q'_0	1	$(no, 1, -)$
q'_0	$\triangleright$	$(yes, \triangleright, \rightarrow)$
q'_1	0	$(no, 1, -)$
q'_1	1	$(q, \sqcup, \leftarrow)$
q'_1	$\triangleright$	$(yes, \triangleright, \rightarrow)$
q	0	$(q, 0, \leftarrow)$
q	1	$(q, 1, \leftarrow)$
q	$\triangleright$	$(s, \triangleright, \rightarrow)$

Configuration

- a configuration of M is (q, w, u)
  1. q is the state
  2. w is the string to the left of the cursor, including the symbol scanned by the cursor
  3. u is the string to the right

  consider M on 0010:

  $(s, \triangleright, 0010)$ $(q_0, \triangleright\triangleright 010\sqcup, \epsilon)$ $(q_0', \triangleright\triangleright 010, \sqcup)$ $(q, \triangleright\triangleright 01, \sqcup\sqcup)$

- $(q, w, u) \xrightarrow{M} (q', w', u')$ one step from (q, w, u) to (q', w', u')
  yields in one step
- $(q, w, u) \xrightarrow{M^k} (q', w', u')$ k steps yield (q', w', u') from (q, w, u)
  yields in k steps
- $(q, w, u) \xrightarrow{M^*} (q', w', u')$ exists k, $(q, w, u) \xrightarrow{M^k} (q', w', u')$
  yields

Turing Machines as Algorithms

$\mathrm{L} \subseteq (\Sigma - \{\triangleright, \sqcup\})^*$ be a language, M a TM with

1. $\forall$ strings $x \in (\Sigma - \{\triangleright, \sqcup\})^*$: if $x \in \mathrm{L}$, then $M(x) = \textit{yes}$, and if $x \notin \mathrm{L}$, then $M(x) = \textit{no}$
2. M decides L; L is called recursive (or decidable)

M be a TM with

1. $\forall$ strings $x \in (\Sigma - \{\triangleright, \sqcup\})^*$: if $x \in \mathrm{L}$, then $M(x) = \textit{yes}$, and if $x \notin \mathrm{L}$, then M does not terminate
2. M accepts L; L is called recursive enumerable (or semi-decidable)

Fact If L is recursive, then it is recursive enumerable

$f : (\Sigma - \{\triangleright, \sqcup\})^* \to \Sigma^*$, M be a TM (over Σ), with

1. $\forall$ strings $x \in (\Sigma - \{\triangleright, \sqcup\})^*$: $M(x) = f(x)$
2. M computes f; f is called recursive

Many-string Turing Machine

Representation

To solve a problem by a TM, we have to fix a representation of an instance of the problem as string.

- ➡ All reasonable representations are polynomially related
  Unary coding is not reasonable
- ➡ Convention: Numbers will always be represented in binary.

Definition

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

1. K, Σ as before.
2. δ is a function
$$\delta \colon K \times \Sigma^k \to (K \cup \{\mathrm{h}, \mathit{yes}, \mathit{no}\}) \times (\Sigma \times \{\leftarrow, \rightarrow, -\})^k$$
3. If M computes a function, the output is written on the last string.

Example: TM for palindromes (faster version)

$p \in K$	$\sigma_1 \in \Sigma$	$\sigma_2 \in \Sigma$	$\delta(p, \sigma_1, \sigma_2)$
s	0	$\sqcup$	$(s, 0, \rightarrow, 0, \rightarrow)$
s	1	$\sqcup$	$(s, 1, \rightarrow, 1, \rightarrow)$
s	$\rhd$	$\rhd$	$(s, \rhd, \rightarrow, \rhd, \rightarrow)$
s	$\sqcup$	$\sqcup$	$(q, \sqcup, \leftarrow, \sqcup, -)$
q	0	$\sqcup$	$(q, 0, \leftarrow, \sqcup, -)$
q	1	$\sqcup$	$(q, 1, \leftarrow, \sqcup, -)$
q	$\rhd$	$\sqcup$	$(p, \rhd, \rightarrow, \sqcup, \leftarrow)$
p	0	0	$(p, 1, \rightarrow, \sqcup, \leftarrow)$
p	1	1	$(p, 1, \rightarrow, \sqcup, \leftarrow)$
p	0	1	$(\mathit{no}, 1, -, \sqcup, -)$
p	1	0	$(\mathit{no}, 1, -, \sqcup, -)$
p	$\sqcup$	$\rhd$	$(\mathit{yes}, \sqcup, -, \rhd, \rightarrow)$

Configuration (cont'd)

- A configuration of a k-string TMis

$$(q, w_1, u_1, \ldots, w_k, u_k)$$

  1. q is the state
  2. $w_i u_i$ is the i^{th} string
  3. the cursor points to the last symbol in w_i

- $(q, w_1, u_1, \ldots, w_k, u_k) \xrightarrow{\mathrm{M}} (q', w'_1, u'_1, \ldots, w'_k, u'_k)$ one step from $(q, w_1, u_1, \ldots, w_k, u_k)$ to $(q', w'_1, u'_1, \ldots, w'_k, u'_k)$

- $(q, w_1, u_1, \ldots, w_k, u_k) \xrightarrow{\mathrm{M}^l} (q', w'_1, u'_1, \ldots, w'_k, u'_k)$ l steps

- $(q, w_1, u_1, \ldots, w_k, u_k) \xrightarrow{\mathrm{M}^*} (q', w'_1, u'_1, \ldots, w'_k, u'_k)$ yields

Consider the 2-string TM M for palindromes:

$$(s, \triangleright, 0010, \triangleright, \epsilon) \xrightarrow{\mathrm{M}} (s, \triangleright 0, 010, \triangleright \sqcup, \epsilon) \xrightarrow{\mathrm{M}^*} (q, \triangleright 0010, \sqcup, \triangleright 0010, \sqcup)$$

Time Complexity

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

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

  then the time required by M is t $(H \in \{h, yes, no\})$

- M operates within time $f(n)$ if, for any input string x, the time required is at most $f(|x|)$

- $\mathrm{L} \in \textbf{TIME}(f(n))$
  if L is decided by a multi-string M operating in time $f(n)$

Example: Palindromes

The 1-string Turing machines M for palindromes operates in $\lceil \frac{n}{2} \rceil$ stages and

- ➡ first stage: $2n + 1$ steps to compare 1st and last symbol
- ➡ second stage: $2(n - 2) + 1$ steps to compare 1st and last symbol
- ➡ ...

In total

$$(2n+1) + (2n-3) + \cdots = \frac{(n+1)(n+2)}{2} = f(n)$$

Fact

1. The language L of all palindromes is in $\mathbf{TIME}(f(n)) = \mathbf{TIME}(\mathcal{O}(n^2))$
2. And with the 2-string Turing machine in $\mathbf{TIME}(\mathcal{O}(n))$

Theorem

Given any k-string Turing machine M operating within time $f(n)$, we can construct a Turing machine M′ operating within time $\mathcal{O}(f(n)^2)$ and $\mathrm{M}(x) = \mathrm{M}'(x)$.

Proof

Define $\mathrm{M}' = (K', \Sigma', \delta', s)$ $\qquad \Sigma' = \Sigma \cup \underline{\Sigma} \cup \{\triangleright', \triangleleft, \triangleleft'\}$

Configuration

$$(q, w_1, u_1, \ldots, w_k, u_k)$$

in M, becomes

$$(q, \triangleright, w'_1 u_1 \triangleleft w'_2 u_2 \triangleleft \cdots w'_k u_k \triangleleft \triangleleft)$$

w'_i equals w_i with $\triangleright$ replaced by $\triangleright'$ and last symbol σ_i by $\underline{\sigma_i}$.

Initial phase.

1. Shift input to the right, precede with $\triangleright'$
2. Write $\triangleleft(\triangleright'\triangleleft)^{k-1}\triangleleft$ after the input

Simulation a move in M

1. M′ scans its string and "remembers" the k currently scanned symbols in M; remembering is done, by introducing new states that represent the states of M and the scanned symbols
2. M′ scans its string again and simulates one step of M
3. If one of the strings is extended in M, M′ has to move everything after it to the right
   - ➡ Replace currently scanned $\triangleleft$ by $\triangleleft'$
   - ➡ go to the right side (marked by $\triangleleft\triangleleft$)
   - ➡ move everything to the right
   - ➡ when we reach $\triangleleft'$, replace by $\sqcup\triangleleft$

Fact

total length of the string of M′ is never more than $k(f(|x|)+1)+1$

Simulating a move at most costs

$$(4k(f(|x|)+1)+4 \text{ steps}) + (3k(f(|x|)+1)+3 \text{ steps}) \cdot k$$

In total, the (worst-case) time-complexity is

$$\mathcal{O}(k^2 f(|x|)^2) = \mathcal{O}(f(|x|)^2)$$

□