Fundamentele Informatica 3

voorjaar 2012

`http://www.liacs.nl/home/rvvliet/fi3/`

Rudy van Vliet
kamer 124 Snellius, tel. 071-527 5777
rvvliet(at)liacs.nl

college 7, 19 maart 2012

7. Turing Machines
7.1. A General Model of Computation
7.2. Turing Machines as Language Acceptors
7.3. Turing Machines That Compute Partial Functions
7.4. Combining Turing Machines

7.2. Turing Machines as Language Acceptors

Example 7.3. A TM Accepting a Regular Language

$L = \{a, b\}^*\{ab\}\{a, b\}^* \cup \{a, b\}^*\{ba\}$

First a finite automaton, then a Turing machine

Example 7.5. A TM Accepting $XX = \{xx \mid x \in \{a, b\}^*\}$

7.1. A General Model of Computation

Definition 7.1. Turing machines

A Turing machine (TM) is a 5-tuple $T = (Q, \Sigma, \Gamma, q_0, \delta)$, where

Q is a finite set of states. The two *halt* states h_a and h_r are not elements of Q.

Σ, the input alphabet, and Γ, the tape alphabet, are both finite sets, with $\Sigma \subseteq \Gamma$. The *blank* symbol Δ is not an element of Γ.

q_0, the initial state, is an element of Q.

δ is the transition function: . . .

Definition 7.1. Turing machines

A Turing machine (TM) is a 5-tuple $T = (Q, \Sigma, \Gamma, q_0, \delta)$, where

Q is a finite set of states. The two *halt* states h_a and h_r are not elements of Q.

Σ, the input alphabet, and Γ, the tape alphabet, are both finite sets, with $\Sigma \subseteq \Gamma$. The *blank* symbol Δ is not an element of Γ.

q_0, the initial state, is an element of Q.

δ is the transition function:

$$\delta : Q \times (\Gamma \cup \{\Delta\}) \rightarrow (Q \cup \{h_a, h_r\}) \times (\Gamma \cup \{\Delta\}) \times \{R, L, S\}$$

Interpretation of

$$\delta(p, X) = (q, Y, D)$$

If q is h_a or h_r, the move causes T to halt

What if $D = L$ and T is on square 0?

Normally, TM starts with

- input string starting in square 1 and all other squares blank,
- and its tape head on square 0.

Tape always contains finite number of non-blanks.

Notation:

configuration. . .

Notation:

description of tape contents: $x\underline{\sigma}y$ or $x\underline{y}$

$x\underline{y} = x\underline{y}\Delta = x\underline{y}\Delta\Delta$

if $y = \Lambda$, then $x\underline{\Delta}$

configuration $xqy = xqy\Delta = xqy\Delta\Delta$

if $y = \Lambda$, then $xq\Delta$

move: $xqy \vdash_T zrw \quad xqy \vdash_T^* zrw$

$xqy \vdash zrw \quad xqy \vdash^* zrw$

example: configuration $aabqa\Delta a$ and $\delta(q, a) = (r, \Delta, L)$

initial configuration corresponding to input x: $q_0\Delta x$

In the third edition of the book, a configuration is denoted as $(q, x\underline{y})$ or $(q, x\underline{\sigma}y)$ instead of xqy or $xq\sigma y$.
This old notation is also allowed for Fundamentele Informatica 3.

Definition 7.2. Acceptance by a TM

If $T = (Q, \Sigma, \Gamma, q_0, \delta)$ is a TM and $x \in \Sigma^*$,
x is accepted by T if

$$q_0 \Delta x \vdash_T^* w h_a y$$

for some strings $w, y \in (\Gamma \cup \{\Delta\})^*$
(i.e., if, starting in the initial configuration corresponding to input x, T eventually halts in the accepting state).

N.B.: sequence of moves leading to h_a is unique

A language $L \subseteq \Sigma^*$ is accepted by T if $T = L(T)$, where

$$L(T) = \{x \in \Sigma^* \mid \ x \text{ is accepted by } T\ \}$$

Example 7.7. Accepting $L = \{a^i b a^j \mid 0 \leq i < j\}$

To illustrate that a Turing machine T may run forever for an input that is not in $L(T)$. No problem!

7.3. Turing Machines That Compute Partial Functions

Definition 7.9. A Turing Machine Computing a Function

Let $T = (Q, \Sigma, \Gamma, q_0, \delta)$ be a Turing machine, k a natural number, and f a partial function on $(\Sigma^*)^k$ with values in Γ^*. We say that T computes f if for every $(x_1, x_2, \ldots, x_k)$ in the domain of f,

$$q_0 \Delta x_1 \Delta x_2 \Delta \ldots \Delta x_k \vdash_T^* h_a \Delta f(x_1, x_2, \ldots, x_k)$$

and no other input that is a k-tuple of strings is accepted by T.

A partial function $f : (\Sigma^*)^k \to \Gamma^*$ is Turing-computable, or simply computable, if there is a TM that computes f.

Example 7.10. The Reverse of a String

$$
\begin{array}{l}
\underline{\Delta}\, a\ a\ b\ a\ b \\
\Delta\, A\, a\ b\ a\ b \\
\Delta\, A\, a\ b\ a\, A \\
\Delta\, B\, a\ b\ a\, A \\
\Delta\, B\, A\, b\ a\, A \\
\Delta\, B\, A\, b\, A\, A \\
\Delta\, B\, A\, b\, A\, A \\
\Delta\, B\, A\, B\, A\, A \\
\underline{\Delta}\, b\ a\ b\ a\ a
\end{array}
$$

Study this example yourself.

Example 7.12. The Quotient and Remainder Mod 2

Example 7.14. The Characteristic Function of a Set

$$\chi_L(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L \end{cases}$$

Example 7.14. The Characteristic Function of a Set

$$\chi_L(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L \end{cases}$$

From computing χ_L to accepting L

From accepting L to computing χ_L

7.4. Combining Turing Machines

Many notations

Example 7.17. Finding the Next Blank or the Previous Blank

NB

PB

Example 7.18. Copying a String

Copy: from $\underline{\underline{\Delta}}x$ to $\underline{\underline{\Delta}}x\Delta x$

Example 7.20. Inserting and Deleting a Symbol

Delete: from $x\underline{\sigma}y$ to $x\underline{y}$

Insert(σ): from $x\underline{y}$ to $x\underline{\sigma}y$

N.B.: y does not contain blanks

Example 7.21. Erasing the Tape
from the current position to the right

Example 7.24. Comparing Two Strings

Equal: accept $\underline{\Delta} x \Delta y$ if $x = y$,
and reject if $x \neq y$

Example 7.25. Accepting the Language of Palindromes

$$Copy \rightarrow NB \rightarrow R \rightarrow PB \rightarrow Equal$$