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 9, 2 april 2012

7. Turing Machines
7.7. Nondeterministic Turing Machines
7.8. Universal Turing Machines
8. Recursively Enumerable Languages

7.7. Nondeterministic Turing Machines

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:

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

Nondeterministic Turing machine.

There may be more than one move for a state-symbol pair.

Same notation:

$$wpax \vdash_T yqbz \qquad wpax \vdash_T^* yqbz$$

A string x is accepted by T if

$$q_0\Delta x \vdash_T^* wh_ay$$

for some strings $w, y \in (\Gamma \cup \{\Delta\})^*$.

NTM useful for producing output,
but not for computing function.

Example 7.28. The Set of Composite Natural Numbers.

NB → *G2* → *NB* → *G2* → *PB* → *M* → *PB* → *Equal*

Take $x = 1^{15}$

Theorem 7.31.
For every nondeterministich TM $T = (Q, \Sigma, \Gamma, q_0, \delta)$,
there is an ordinary (deterministic) TM $T_1 = (Q_1, \Sigma, \Gamma_1, q_1, \delta_1)$
with $L(T_1) = L(T)$.

Proof. . .

7.8. Universal Turing Machines

Definition 7.32. Universal Turing Machines

A *universal* Turing machine is a Turing machine T_u that works as follows. It is assumed to receive an input string of the form $e(T)e(z)$, where
- T is an arbitrary TM,
- z is a string over the input alphabet of T,
- and e is an encoding function whose values are strings in $\{0, 1\}^*$.

The computation performed by T_u on this input string satisfies these two properties:
1. T_u accepts the string $e(T)e(z)$ if and only if T accepts z.
2. If T accepts z and produces output y, then T_u produces output $e(y)$.

Crucial features of any encoding function e:

1. It should be possible to decide algorithmically, for any string $w \in \{0,1\}^*$, whether w is a legitimate value of e.
2. A string w should represent at most one Turing machine, or at most one string z.
3. If $w = e(T)$ or $w = e(z)$, there should be an algorithm for *decoding* w.

Assumptions:

1. Names of the states are irrelevant.

2. Tape alphabet Γ of every Turing machine T is subset of infinite set $\mathcal{S} = \{a_1, a_2, a_3, \ldots\}$, where $a_1 = \Delta$.

Definition 7.33. An Encoding Function

Assign numbers to each state:
$n(h_a) = 1$, $n(h_r) = 2$, $n(q_0) = 3$, $n(q) \geq 4$ for other $q \in Q$.

Assign numbers to each tape symbol:
$n(a_i) = i$.

Assign numbers to each tape head direction:
$n(R) = 1$, $n(L) = 2$, $n(S) = 3$.

Definition 7.33. An Encoding Function (continued)

For each move m of T of the form $\delta(p, \sigma) = (q, \tau, D)$

$$e(m) = 1^{n(p)}01^{n(\sigma)}01^{n(q)}01^{n(\tau)}01^{n(D)}0$$

We list the moves of T in some order as $m_1, m_2, \ldots, m_k$, and we define

$$e(T) = e(m_1)0e(m_2)0 \ldots 0e(m_k)0$$

If $z = z_1 z_2 \ldots z_j$ is a string, where each $z_i \in \mathcal{S}$,

$$e(z) = 01^{n(z_1)}01^{n(z_2)}0 \ldots 01^{n(z_j)}0$$

Example 7.34. A Sample Encoding of a TM

Theorem 7.36. Let $E = \{e(T) \mid T$ is a Turing machine$\}$. Then for every $x \in \{0,1\}^*$, $x \in E$ if and only if all these conditions are satisfied:

1. x matches the regular expression

$$(11^*0)^5 0((11^*0)^5 0)^*$$

so that it can be viewed as a sequence of one or more 5-tuples.

...

Theorem 7.36. Let $E = \{e(T) \mid T$ is a Turing machine$\}$. Then for every $x \in \{0,1\}^*$, $x \in E$ if and only if all these conditions are satisfied:

1. x matches the regular expression $(11^*0)^5 0((11^*0)^5 0)^*$ so that it can be viewed as a sequence of one or more 5-tuples.

2. No two substrings of x representing 5-tuples can have the same first two parts (no move can appear twice, and there can't be two different moves for a given combination of state and tape symbol).

3. None of the 5-tuples can have first part 1 or 11 (there can be no moves from a halting state).

4. The last part of each 5-tuple must be 1, 11, or 111 (it must represent a direction).

8. Recursively Enumerable Languages

8.1. Recursively Enumerable and Recursive

Definition 8.1. Accepting a Language and Deciding a Language

A Turing machine T with input alphabet Σ accepts a language $L \subseteq \Sigma^*$,
if $L(T) = L$.

T *decides* L,
if T computes the characteristic function $\chi_L : \Sigma^* \to \{0, 1\}$

A language L is *recursively enumerable*,
if there is a TM that accepts L,

and L is *recursive*,
if there is a TM that decides L.

Theorem 8.2.
Every recursive language is recursively enumerable.

Proof. . .

Theorem 8.3.
If $L \subseteq \Sigma^*$ is accepted by a TM T that halts on every input string, then L is recursive.

Proof. . .

Morgen publicatie Huiswerkopgave 3

Volgende week maandag 9 april 2012: geen college (tweede paasdag)

Volgende week dinsdag 10 april 2012: 1 uur hoorcollege, 1 uur werkcollege