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 10a, dinsdag 10 april 2012

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 $\chi_L : \Sigma^* \rightarrow \{0, 1\}$

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...

Theorem 8.4. If L_1 and L_2 are both recursively enumerable languages over Σ, then $L_1 \cup L_2$ and $L_1 \cap L_2$ are also recursively enumerable.

Proof...

Exercise 8.2. Consider modifying the proof of Theorem 8.4 by executing the two TMs sequentially instead of simultaneously.
Given TMs T_1 and T_2 accepting L_1 and L_2, respectively, and an input string x, we start by making a second copy of x.
We execute T_1 on the second copy; if and when this computation stops, the tape is erased except for the original input, and T_2 is executed on it.

a. Is this approach feasible for accepting $L_1 \cup L_2$, thereby showing that the union of recursively enumerable languages is recursively enumerable? Why or why not?

b. Is this approach feasible for accepting $L_1 \cap L_2$, thereby showing that the intersection of recursively enumerable languages is recursively enumerable? Why or why not?

Theorem 8.5. If L_1 and L_2 are both recursive languages over Σ, then $L_1 \cup L_2$ and $L_1 \cap L_2$ are also recursive.

Proof. Exercise 8.1, done at last week's exercise class.

Theorem 8.6. If L is a recursive language over Σ, then its complement L' is also recursive.

Proof. . .

Theorem 8.7. If L is a recursively enumerable language, and its complement L' is also recursively enumerable, then L is recursive (and therefore, by Theorem 8.6, L' is recursive).

Proof. . .

8.2. Enumerating a Language

Definition 8.8. A TM Enumerating a Language

Let T be a k-tape Turing machine for some $k \geq 1$, and let $L \subseteq \Sigma^*$. We say T enumerates L if it operates such that the following conditions are satisfied.

1. The tape head on the first tape never moves to the left, and no nonblank symbol printed on tape 1 is subsequently modified or erased.

2. For every $x \in L$, there is some point during the operation of T when tape 1 has contents

$$x_1\#x_2\#\ldots\#x_n\#x\#$$

for some $n \geq 0$, where the strings $x_1, x_2, \ldots, x_n$ are also elements of L and $x_1, x_2, \ldots, x_n, x$ are all distinct. If L is finite, then nothing is printed after the # following the last element of L.

Theorem 8.9. For every language $L \subseteq \Sigma^*$,
• L is recursively enumerable
if and only if there is a TM enumerating L,
• and L is recursive if and only if there is a TM that enumerates the strings in L in canonical order (see Section 1.4).

In other words:
1. If there is a TM that accepts L, then there is a TM that enumerates L.
2. If there is a TM that enumerates L, then there is a TM that accepts L.
3. If there is a TM that decides L, then there is a TM that enumerates L in canonical order.
4. If there is a TM that enumerates L in canonical order, then there is a TM that decides L.

Proof. . .