Free University of Bozen-Bolzano – Faculty of Computer Science
Master of Science in Computer Science
Theory of Computing – A.Y. 2009/2010
Final exam – 5/2/2010 – Part 1
Time: 90 minutes

This is a closed book exam: the only resources allowed are blank paper, pens, and your head. Explain your reasoning. Write clearly, in the sense of logic, language and legibility. The clarity of your explanations affects your grade. Write your name and ID on every solution sheet. Good luck!

Problem 1.1 [6 points] You must give a brief explanation of your answer to receive full credit.

(a) Decide whether the following statement is TRUE or FALSE: Every language contained in a recursive language is recursive.

(b) Let M_2 be a two-tape (deterministic) TM, and let M be the result of converting M_2 into a one-tape TM. Sketch using two to three sentences the main idea underlying the conversion. How are the running times of M_2 and M related to each other?

(c) Decide whether the following statement is TRUE or FALSE: There exist more non-deterministic TMs than deterministic TMs.

Problem 1.2 [6 points] Consider the language $L = \{nw \mid n \in \{0,1\}^*, w \in \{a,b\}^*$, and n represents in binary the number of b's in $w\}$.

E.g.: $\texttt{0} \in L$, $\texttt{0aaa} \in L$, $\texttt{10babaa} \in L$, $\texttt{10babba} \notin L$, $\texttt{babaa10} \notin L$, $\texttt{10} \notin L$.

(a) Construct a TM M that accepts L.

(b) Show the sequence of IDs of M on the input strings "`10bab`" and "`10baa`".

Problem 1.3 [6 points] Consider a language L over $\{0,1\}$ for which there exists a TM M_e over $\{0,1,\texttt{\#}\}$ that outputs on its tape all strings of L *in lexicographic order*, separating each string from the next by a `#`. A string $w \in L$ is considered to be output by M_e, as soon as the `#` following w is written, and from that moment onward w is not touched anymore by M_e. Show that L is recursive.

Problem 1.4 [6 points] Let f and g be primitive recursive functions. Show that the following functions are primitive recursive:

(a) $f_1(x) = \begin{cases} f(x), & \text{if } g(i) < g(i+1), \text{ for some } 0 \leq i \leq x \\ 1, & \text{otherwise} \end{cases}$

(b) $f_2(x) = \begin{cases} f(x), & \text{if } x = 0 \text{ or } x = 1 \\ g(x-1) \cdot f_2(x-1) + f_2(x-2), & \text{if } x \geq 2 \end{cases}$

Problem 1.5 [6 points] Let p be a total predicate with $n+1$ arguments, and $f(\vec{x})$ the n-argument function defined from $p(\vec{x}, y)$ by *(unbounded) minimization*.

(a) Provide the formal definition of the function $f(\vec{x})$ and describe its meaning. When is $f(\vec{x})$ undefined?

(b) Show that, if $p(\vec{x}, y)$ is a total Turing computable predicate, then also $f(\vec{x})$ is Turing computable.