Free University of Bozen-Bolzano – Faculty of Computer Science
Master of Science in Computer Science
Theory of Computing – A.Y. 2015/2016
Final exam – 20/6/2016 – 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 [2+2+2 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 over the unary alphabet $\Sigma = \{1\}$ is recursively enumerable.

(b) Determine whether the following problem is decidable: Given the encoding $\mathcal{E}(M)$ of a TM M, decide whether there is a string in $\mathcal{L}(M)$ that starts and ends with the same symbol.

(c) Let M_3 be a 3-tape TM, and let M_1 be the result of converting M_3 into an ordinary (single-tape) TM. Sketch using two to three sentences the main idea underlying the conversion. How are the running times of M_3 and M_1 related to each other?

Problem 1.2 [4+2 points] Consider a TM M_d that computes the proper subtraction of two numbers represented in *binary*. More precisely, the tape initially contains $\mathfrak{b}\overline{n}_1\mathfrak{b}\overline{n}_2$, where $\overline{n}_i$ denotes the number n_i represented in binary, with the least significant digit on the right. M_d starts its computation with the head on the blank preceding the first symbol of $\overline{n}_1$ and when it enters the final state, the tape contains $\overline{d}$, where $d = n_1 \dot{-} n_2$. Note that, when M_d enters the final state, its head may be in any position of the tape.
E.g.: when the tape initially contains $101\mathfrak{b}11$, then M_d computes 10.

(a) Construct M_d. [*Hint*: M_d repeatedly decrements n_2 and n_1, until $n_1 = 0$ or $n_2 = 0$. When this happens, M_d deletes what remains of $\overline{n}_2$, and what remains of $\overline{n}_1$ represents the result.]

(b) Show the sequence of IDs of M_d on the input string $101\mathfrak{b}11$.

Problem 1.3 [6 points] Consider a language L over $\{0, 1\}$ for which there exists a TM M_e with tape alphabet $\{0, 1, \texttt{\#}\}$ that, when started on the empty tape, outputs on the 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.
[*Hint*: Show how to construct a TM M that decides L, by comparing the strings produced by M_e with the input string for M. Find an appropriate criterion to stop the computation of M.]

Problem 1.4 [3+3 points] Let f and g be one-argument primitive recursive functions. Show that the following functions are primitive recursive:

(a) $f_1(x) = \begin{cases} 1 + f(x), & \text{if } g(i) > g(i+1), \text{ for all } i \text{ with } 0 \leqslant i \leqslant x \\ 1, & \text{otherwise} \end{cases}$

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

Problem 1.5 [2+4 points] Let p be a total predicate with $n+1$ arguments, and $f(\vec{x}, y)$ the $n+1$-argument function defined from $p(\vec{x}, z)$ by *bounded minimization*.

(a) Provide the formal definition of $f(\vec{x}, y)$. Describe the intuitive meaning of $f(\vec{x}, y)$.

(b) Show that, when p is a primitive recursive predicate, then f is a primitive recursive function.
[*Hint*: Exploit the fact that a bounded product and a bounded sum of a primitive recursive function is primitive recursive.]