Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik, TU Dresden
Prof. Dr. F. Baader

Nöthnitzer Str. 46
01187 Dresden
Tel.: 0351/463–38237
Tel.: 0351/463–39171

3. Exercises for the Course „Complexity and Logic“

Exercise 11:

In the lecture, a Turing machine was sketched that accepts the following language:

$$L := \{\#\widehat{n}\#\widehat{m}\#\widehat{n+m}\# \mid n, m > 0\}$$

Work out the details for Part 2 of this TM, which checks that the sum is correct. For simplicity, you can assume that the input has the required form, and that $\widehat{n}$, $\widehat{m}$, and $\widehat{n+m}$ all have the same number of bits (padded with leading 0's).
Describe the TM in graphical form (as used for Part 1 of the same TM in the lecture).

Exercise 12:

We want to construct a Turing machine that implements the algorithm for graph reachability given in the lecture. If the input graph consists of the nodes $\{1, \ldots, n\}$, then the input to the Turing machine is given as a word

$$m_1 \cdots m_{n^2} \# u_1 \cdots u_k \# v_1 \cdots v_k$$

where

- $m_1 \ldots m_{n^2} \in \{0,1\}^*$ is obtained by concatenating all rows of the adjacency matrix of the input graph (as described in the lecture);
- $u_1 \cdots u_k \in \{0,1\}^*$ is a binary representation of the start node u;
- $u_1 \cdots u_k \in \{0,1\}^*$ is a binary representation of the destination node v;

Assume that the algorithm implements the graph reachability algorithm by marking edges (instead of nodes): it simply replaces entries $0, 1$ of the adjacency matrix with marked variants $0', 1'$. Describe how the Turing machine can solve the following subtasks:

(a) determine the number of nodes in the input graph;

(b) given a node i in binary representation, find a node j that is a direct successor of i such that the edge from i to j is unmarked (if such a node exists).

Describe how the Turing machine works using these subtasks as blackbox modules. Do not give details of the transition function.

Exercise 13:

Let Σ be a finite alphabet. Describe a Turing machine M that enumerates all (finite) words over Σ such that shorter words are enumerated before longer ones. More precisely, M has a special state q_{out} that is visited after one word has been written on the tape and before the next word is produced. Give a high-level description of the TM, without all the details of the transition function.
Can you also build a TM that enumerates all (finite) words over the infinite alphabet $\mathbb{N}$? How?

Exercise 14:

A *2-sided Turing machine (2TM)* is a Turing machine with a two-side infinite tape. A 2TM starts with its head on the left-most symbol of the input. Both to the left and right of the input, the tape is filled with blanks. Acceptance is defined as in the case of regular TMs.
Prove that 2TMs accept the same languages as standard TMs.