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

1. Exercises for the Course „Complexity and Logic“

Exercise 1:

Consider the algorithm for graph reachability presented in the lecture. Prove that the algorithm returns "yes" if and only if the node v is reachable from the node u.

Note: since it follows from the complexity analysis given in the lecture that the algorithm terminates, the latter can be assumed without proof.

Exercise 2:

Determine which of the following statements are correct. Explain your decision.

(a) $2n = O(n)$, $2n = o(n)$, $1 = o(n)$;

(b) $n = O(n \cdot \log n)$, $n \cdot \log n = o(n^2)$;

(c) $2^{2^n} = O(2^n)$, $2^{2n} = o(2^n)$;

(d) $f(n) = o(g(n))$ implies $f(n) = O(g(n))$ and $g(n) \neq O(f(n))$;

(e) Show that the following two functions are a counterexample against the converse of (d):

- $f(n) = 0.5 \cdot n$;
- $g(n) = \begin{cases} 2^n & \text{if } n \text{ even} \\ n & \text{if } n \text{ odd} \end{cases}$

Exercise 3:

It was sketched in the lecture how to decide the satisfiability of Horn formulas in linear time by introducing a counter for each implication.

Work out the details of this idea by writing a linear-time algorithm for the satisfiability of Horn formulas. Write the algorithm in pseudo-code notation (as in the lecture).

Exercise 4:

Prove that the graph reachability algorithm is a special case of the algorithm for deciding satisfiability of Horn formulas. Hint: introduce one variable for each node of the graph.