Komplexitätstheorie 2003
Problemset 2

December 4, 2003

Due: January 1, 2004

Regardless how many problems you solve, you can get at most 50 points for this problemset. 100 points are required in total (there are three problemsets).

Problem 1 – Transitivity of Log-Space Reductions

We already defined log-space reductions, i.e., reductions which are only allowed to use \( O(\log n) \) space. Note that such a reduction can produce a polynomially sized result on its output tape.

An important property of reductions in general is the transitivity. A reduction relation \( \leq \) is transitive if \( A \leq B \) and \( B \leq C \) implies \( A \leq C \) for any three computational problems \( A, B, C \).

[10] Prove that log-space reductions are indeed transitive.

Problem 2 – Satisfiability vs. Tautology

We proved that 3Sat is \( \text{NP} \)-complete. 3Sat is the problem of determining whether a given 3-CNF formula \( \phi \) has a satisfying assignment or not. Also, it has been shown that the problem of determining whether a given Boolean formula is a tautology is \( \text{coNP} \)-complete.

What about 3Taut?

[3] Determine the complexity of 3Taut, i.e., the problem of determining whether a given 3-CNF formula \( \phi \) is a tautology.

Problem 3 – Circuit Evaluation Variants

The computational problem CircuitEval has as instances Boolean circuits with constant gates, and-gates, or-gates and negation-gates. One of the gates is marked as output gate. The question to be answered, given such a circuit, is whether the output-gate evaluates to true or false (there are no input-gates, thus the value of the output-gate is constant).

We introduce the following two variants of CircuitEval.
• **MonotoneCircuitEval**: In this case, the instances are not allowed to contain negation-gates.

• **PlanarCircuitEval**: In this case, there must be an embedding of the circuit on the plane such that no wire crossings occur.

Prove that these two variants ([10] PlanarCircuitEval and [5] MonotoneCircuitEval) are P-complete with respect to log-space reductions.

**Problem 4 – Formula Evaluation**

A special case of a Boolean circuit is a Boolean formula. Formulas are allowed to contain the Boolean connectives \( \neg, \lor, \land \), the Boolean constants TRUE and FALSE and braces \( () \). FormulaEval denotes the problem of evaluating such a formula.

[5] The problem to be solved: Determine the complexity of deciding whether a given Boolean formula \( \phi \) evaluates to TRUE or FALSE.

**Problem 5 – HornSat**

The last problem set introduced the problem HornSat (the problem of deciding whether a set of Horn-clauses is satisfiable or not). In the last problem set, you were asked to prove that HornSat is in P. This time, you are asked to prove that HornSat is P-complete.

[5] Since we already have the membership in P, it remains to prove the P-hardness.

**Problem 6 – Fuzzy Logic**

The syntactical structure of Gödel logic is same as in the case of ordinary Boolean logic. However, the semantics are different. Given a formula \( \phi \) over a set of variables \( X = \{x_1, \ldots, x_n\} \), we define

• an assignment to the variables \( X \) as a function \( \tau : X \rightarrow [0,1] \).

• and the evaluation function \( m(\phi, \tau) \) where \( \phi \) is a formula over the variables \( X \) and \( \tau \) is an assignment to the variables in \( X \). We set

\[
\begin{align*}
- m(x_i, \tau) &= \tau(x_i) \text{ with } x_i \in X. \\
- m(\sigma \rightarrow \rho, \tau) &= \begin{cases} 
1 & : m(\sigma) \leq m(\rho) \\
m(\rho) & : \text{ otherwise}
\end{cases} \\
- m(\sigma \lor \rho, \tau) &= \max(m(\sigma), m(\rho)) \\
- m(\sigma \land \rho, \tau) &= \min(m(\sigma), m(\rho))
\end{align*}
\]

Using \( \neg \phi \) as shortcut for \( \phi \rightarrow 0 \) we find \( m(\neg \sigma, \tau) = \begin{cases} 
1 & : m(\sigma) = 0 \\
0 & : \text{ otherwise}
\end{cases} \)
We call the satisfiability problem in this logic \( \text{GödelSat} \).

- [10] Prove that the restricted version of \( \text{GödelSat} \) where we allow only assignments of the form \( \tau : X \to \{0, 1/2, 1\} \) is \( \text{NP} \)-hard.
- [2] Prove that the restricted version of \( \text{GödelSat} \) is in \( \text{NP} \).
- [10] Prove that general \( \text{GödelSat} \) is in \( \text{NP} \).

**Problem 7 – Coloring Variants**

Determine the complexity of the following coloring variants:

- [5] \( 2\text{COLORING}, \) the problem of determining whether a graph can be colored with at most two different colors.
- [5] \( 4\text{COLORING}, \) the problem of determining whether a graph can be colored with at most four different colors.
- [2] \( 2\text{COLORING} \) on trees.
- [3] \( 3\text{COLORING} \) on graphs with at most \( \log n \) edges, where \( n \) is the number of vertices.
- [5] Prove that the problem of computing an actual 3-coloring (if it exists) for a given graph can be done in \( \text{FP}^{\text{NP}} \), where \( \text{FP} \) denotes the class of functional problems which can be in polynomial time. Then \( \text{FP}^{\text{NP}} \) is the class of general polynomial time algorithms that can utilize an \( \text{NP} \)-oracle.

**Problem 8 – Tiling**

An instance \( \langle T, t, H, V, S \rangle \) of the TILING problem consists of a set of tiles \( T = \{t_1, \ldots, t_n\} \), a special tile \( t \in T \), two relationships \( H, V \subseteq T \times T \), and the size \( S \). \( H \) (\( V \)) is the horizontal (vertical) compatibility relationship.

Given such an instance, the question is whether there is a tiling function \( f : \{1, \ldots, n\} \times \{1, \ldots, n\} \to T \) such that

- \( f(1, 1) = t \)
- \( (f(i, j), f(i + 1, j)) \in H \) for all \( 1 \leq i < n \) and \( 1 \leq j \leq n \)
- \( (f(i, j), f(i, j + 1)) \in V \) for all \( 1 \leq i \leq n \) and \( 1 \leq j < n \)

The complexity of TILING depends drastically on the representation of the size \( S \).

- [10] Prove that TILING is \( \text{NP} \)-complete if \( S \) is given in unary notation.
- [10] Prove that TILING is \( \text{NEXP} \)-complete if \( S \) is in binary notation.