T-79.148 **Autumn 2002**
Introduction to Theoretical Computer Science
Tutorial 12, 4–5 December
Problems

This is the last set of tutorial problems. The course exam is on Tuesday 17 December, 9–12 a.m. Remember to register for the exam via TOPI, and note that all your Regis-assignments must be completed by the exam date — otherwise your exam paper will not be graded (really!). Please fill in and return also the computerised course feedback form available via the course's web page. If you submit the form by 22 December, you earn an extra exam point that counts towards increasing your grade in case you pass the course. (Further instructions available on the course's web page.)

Homework problems:

1. You are being offered the following programming assignment:

   > *Intel Septium code optimisation*
   > A large company producing embedded systems software would like to have a code optimiser that, given as input any machine language program for the new Intel Septium processor, will output the smallest machine language program that is functionally equivalent to the given one (i.e. has the same input-output behaviour).

   Your comments? Under what conditions would you accept the assignment? Justify your answer.

2. Consider application programs running under some given operating system. Let us say that a program P is *dangerous*, if it on some input modifies the operating system's system files. A *general purpose virus tester* is a program that receives as input an arbitrary application program text P, analyses it and returns output "DANGER", if the program is dangerous, and "OK" otherwise. Show that if any dangerous programs exist at all, then general-purpose virus testing is impossible.

3. Design unrestricted grammars (general rewriting systems) that generate the following languages:

   (a) $\{w \in \{a, b, c\}^* \mid w$ contains equally many a's, b's and c's$\}$,

   (b) $\{ww \mid w \in \{a, b\}^*\}$.

Demonstration problems:

4. Prove, without appealing to Rice's theorem, that the following problem is undecidable:

   Given a Turing machine M; does M accept the empty string?

5. Show that all context-sensitive languages can be recognised by linear-bounded automata. (Make use of the fact that in applying the grammar's production rules, the length of the sentential form under consideration can never decrease, except in the special case of the empty string.) Deduce from this result the fact that all context-sensitive languages are recursive.