T-106.4200 Introduction to Compiling
Exam Oct. 22, 2014

Examiner: Jorma Tarhio

No written material is allowed in this exam. Submit at least one answer sheet, even if an empty one! Write on *each* answer sheet you submit the code of the course, the date, your name, and your student ID number.

You are given an extra point if you fill the (anonymous) course evaluation on the homepage of the course in December.

---

1. Answer shortly to the following questions:

   (a) What is dead code?

   (b) What is a lookahead symbol?

   (c) What is a token?

   (d) What is a static link?

   (e) Why a DFA is more often used for scanning instead of an NFA?

   (10 p)

2. (a) Which phases does a compiler normally consist of? What is the purpose of each phase – what is the input and output of the phase? How are the phases connected?

   (b) What are the pros and cons of multi-pass compiling?

   (7+3 p)

3. Consider the following regular expression:

$$(x\, y^* \mid (x\, z)^*\, y)$$

   (a) Make an NFA for the regular expression by using *Thompson's construction*. Number states starting from zero.

   (b) Build a DFA from the NFA. Name the states by capital letters starting from A.

   (6+6 p)

4. Transform the following grammar to a LL(1) grammar (if necessary, eliminate left recursion, and do left factoring):

$$E \rightarrow E[E] \mid E.\textbf{id} \mid \textbf{id} \mid \textbf{int}$$

   Give FIRST and FOLLOW sets for the new grammar. Construct a LL(1) parsing table. Is the grammar LL(1)? (5+6+2 p)

5. Roman numerals 1–18 are I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII, XIII, XIV, XV, XVI, XVII, and XVIII. Construct an attribute grammar which transforms a Roman numeral to a corresponding integer and which works for the given numerals I–XVIII. (The trivial solution containing a production for each numeral is not accepted.) (10 p)