T-106.4200 Introduction to Compiling
Exam Dec. 15, 2010

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 not later than on Jan. 2.

---

1. Answer shortly to the following questions:

   (a) What does the letter R in LR(1) mean?

   (b) What is a token?

   (c) What is an LR(0) item?

   (d) What is dead code?

   (e) What is bottom-up parsing?

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

(12 p)

2. Consider the following grammar:

$$
\begin{array}{lcl}
L & \rightarrow & L\,;\,S \\
L & \rightarrow & S \\
S & \rightarrow & B \\
S & \rightarrow & \text{if } B \text{ then } S \text{ else } S \text{ fi} \\
S & \rightarrow & \text{if } B \text{ then } S \text{ fi} \\
B & \rightarrow & \text{id} = \text{num}
\end{array}
$$

   Transform the grammar to the LL(1) form.

(8 p)

3. (a) Consider the following grammar: $\{P \rightarrow P \texttt{=>} P \mid P \texttt{ and } P \mid \texttt{not } P \mid (P) \mid \texttt{atom}\}$. Show that the grammar is ambiguous.

(3 p)

   (b) Its LR parsing table is given below. Remove the parse conflicts by assuming that `and` is left associative, `=>` is right associative, and the precedence of the operators (higher first) is `not`, `=>`, and `and`.

	=>	and	not	(	)	atom	$	P
0			s2	s3		s4		1
1	s5	s6					acc	
2			s2	s3		s4		7
3			s2	s3		s4		8
4	r5	r5			r5		r5	
5			s2	s3		s4		9
6			s2	s3		s4		10
7	r3/s5	r3/s6			r3		r3	
8	s5	s6			s11			
9	r1/s5	r1/s6			r1		r1	
10	r2/s5	r2/s6			r2		r2	
11	r4	r4			r4		r4	

(8 p)

P.T.O.

4. Consider the piece of code below. It has been written in a Basic dialect.

```
uusi:
INPUT luku
IF luku<1 THEN END
PRINT luku;"=";
k=2: GOSUB test
k=3
WHILE k<=SQR(luku): GOSUB test: k=k+2: WEND
PRINT luku: GOTO uusi
test:
WHILE luku/k=INT(luku/k) AND luku>k
    luku=luku/k: PRINT k;"*";
WEND
RETURN
END
```

Give a context-free grammar, which describes the syntactic structures of the example. Note that end-of-line is a separator in Basic. (9 p)

5. Explain the aim and the methods of error recovery in a parser.

(9 p)

6. Design a problem of your own on compilers and answer to it. (You cannot get any points for too easy a question.) (6 p)