Solutions for the 2nd middle term exam 21.4. 2005

1. The CYK-table:

a	a	b	b	c
S, D, C_a				
D	S, D, C_a			
S	S, A	C_b		
S, A	X	$\emptyset$	C_b	
S	$\emptyset$	$\emptyset$	S, E	S, C, C_c

   Thus $aabbc$ belongs to language. (Notice the same language can be described as $\{a^n b^m c^l | n = m \vee m = l\}$.

   For totally correct answer 10 p, detailed evaluation criteria will be informed later.

2. It was enough to give the correct subclass for each grammar.

   a) G_1 describes all integers (with possible beginning zeroes), and you can give a regular expression $(0 \cup 1 \cup ... \cup 9)(0 \cup 1 \cup ... \cup 9)^*$ or a right-linear grammar $S \rightarrow 0S|1S|...|9S|0|1|...|9$. 2p

   G_2 is the same language as in task 1. It is ambiguous, e.g. string $aabbcc$ has two parse trees in the given grammar. We can prove that the language is in fact inherently ambiguous but this was not required. 3p

   G_3 can be described as $\{a^n b^m | n \leq m\}$. This can be solved easily by a deterministic pushdown automaton, so the language is at least deterministic. However, if you perform $LL(1)$ test, you see it is also $LL(1)$. $FIRST(\{aSb\}FOLLOW(S)) \cap FIRST(\{B\}FOLLOW(S)) = \{a\} \cap \{b, \epsilon\} = \emptyset$ and $FIRST(\{bB\}FOLLOW(B)) \cap FIRST(\{\epsilon\}FOLLOW(B)) = \{b\} \cap \{\epsilon\} = \emptyset$. 3p

   b) Parsing methods:
   Linear languages: deterministic finite automaton 1p
   $LL(1)$-languages: recursive parser 1p
   Deterministic languages: deterministic pushdown automaton (or $LR(1)$ parser) 1p
   Unambiguous (non-deterministic) and inherently ambiguous languages: CYK-algorithms 1p

3. Let's assume that the input is given the least significant bit on left (easier to solve). Idea: change the second character to 1, if it was 0 or $<$. If it was 1, then scan to right and change all 1s to 0's until you meet 0 or $<$, which is changed to 1.

   The machine described by transition functions:

$\delta(q_1, 0) = (q_2, 0, R)$
$\delta(q_1, 1) = (q_2, 1, R)$
$\delta(q_2, 0) = (q_3, 1, R)$
$\delta(q_2, <) = (q_3, 1, R)$
$\delta(q_2, 1) = (q_2, 0, R)$

Task: Draw the machine!

For fully correct answer 10 points. Detailed evaluation criteria will be informed later.

4. a) See e.g. lecture material. The main point was that we have no computational means to test run-time properties of computer programs. Simulating the program doesn't help, because the program may not halt or we have to test all possible inputs, the set of which is typically infinite. 2p

   b)Semantic properties: i) and iv), syntactic ii) and iii). 4 points, 1p/each.

   c) i) is partially solvable: you can e.g. construct a machine which recognizes such c_M, but does not halt in "No"-cases, or show that the complement problem is totally unsolvable. 2p

   iv) is totally unsolvable: we should check all possible inputs, which will never finish. You can also prove it easily by contradiction method. Or reason that because the complement problem is partially solvable (recursively enumerable language), the current problem must be totally unsolvable (otherwise the problem would be solvable, which is known to be false). 2p