CSc 6340 Introduction to Compilers
Fall 2004, Exam #3 (Take Home) – INDIVIDUAL WORK - NO COLLABORATION
Due: 14 December 2004 (Tuesday)

1. Consider the following syntax-directed semantic specification:

production $A \to \beta$	semantic rule (w.r.t. $A \to \beta$)
$S \to B+$	$[v_S =]print(v_B)$
$S \to C-$	$[v_S =]print(v_C)$
$B \to n$	$v_B = identity(v_n)$
$B \to B_1, n$	$v_B = max(v_{B_1}, v_n)$
$C \to n$	$v_C = identity(v_n)$
$C \to C_1, n$	$v_C = min(v_{C_1}, v_n)$

   (a) Give an attributed parse tree for the source string $5, 2, 3-$ assuming the token n is lexically matched by any string of one or more digits. Evaluate the attributes in the attributed parse tree through syntax-directed evaluation.

   (b) What is the purpose of this syntax-directed semantic specification?

2. Consider the following syntax-directed semantic specification with underlying grammar G2, in which semantic functions are written and evaluated as Java expressions:

production $A \to \beta$	semantic rule (w.r.t. $A \to \beta$)
$S \to aA$	`System.out.println("dogs " +` v_A `+ ".")`
$S \to aAb$	`System.out.println("cats " +` v_A `+ ".")`
$A \to \epsilon$	$v_A =$ `"scratch"`
$A \to baA_1$	$v_A =$ `"chase; cats " +` v_{A_1}

   Give an attributed parse tree for the source string *ababa*, evaluate the attributes in the attributed parse tree through syntax-directed evaluation, and show the output produced.

3. In a game of Paper/Rock/Scissors, two players p_1 and p_2 simultaneously select one of $\{paper, rock, scissors\}$, and the winner is determined by the following function:

$$winner(p_1, p_2) = \begin{cases} paper, & if \ \{p_1, p_2\} = \{paper, rock\}; \\ rock, & if \ \{p_1, p_2\} = \{rock, scissors\}; \\ scissors, & if \ \{p_1, p_2\} = \{scissors, paper\}; \\ tie, & if \ p_1 = p_2. \end{cases}$$

   From the following underlying CFG that generates Paper/Rock/Scissors games, construct an S-attributed translation scheme that prints the winner of a game.

$$G \to (P, P)$$
$$P \to paper \mid rock \mid scissors$$

4. From the following underlying CFG which generates binary numbers, construct an S-attributed translation scheme which has semantic rules embedded after the right-hand sides of the productions and computes the value (val_N) of a binary number (N).

$$N \to NB \mid B$$
$$B \to 0 \mid 1$$

5. Construct from the following underlying CFG an S-attributed attribute grammar which computes the value of a decimal literal (D) consisting of an integer literal (I), an optional fraction (F), and an optional exponent (E). Assume that the terminal symbol d has an attribute val_d which gives the value of the digit d.

production $A \to \beta$	semantic rule (w.r.t. $A \to \beta$)
$D \to IFE$	
$I \to d$	
$I \to Id$	
$F \to \epsilon$	
$F \to .I$	
$E \to \epsilon$	
$E \to eSI$	
$S \to \epsilon$	
$S \to +$	
$S \to -$	