Computer Science Comprehensive Exam—Spring 2001
Compiler Construction

Instructions: Do *not* put your name on the exam, please answer all the questions directly on the exam itself. Answer **all** the questions; you have 60 minutes. Explain answers as fully as possible, give examples or define terms, if appropriate.

1. Answer the following two questions.

   (a) What is the relationship between the set of languages recognized by LL(1) parsers and the set of languages recognized by SLR parsers?

   (b) What is the relationship between the set of languages recognized by LR(1) parsers and the set of languages recognized by LALR(1) parsers?

2. Consider the algorithm to compute CLOSE[I] for set of LR(1) items I. Suppose $A \rightarrow \alpha \bullet X\beta$, z is in I and $X \rightarrow \gamma$. Which item or items (if any) would be added to CLOSE[I]?

3. Consider the following augmented grammar:

$$
\begin{array}{lcl}
E' & \rightarrow & E\,\$ \\
E & \rightarrow & -\,E \\
E & \rightarrow & (\,E\,) \\
E & \rightarrow & V\,T \\
T & \rightarrow & -\,E \\
T & \rightarrow & \epsilon \\
V & \rightarrow & \mathbf{id}\,L \\
L & \rightarrow & (\,E\,) \\
L & \rightarrow & \epsilon
\end{array}
$$

(a) Compute nullable, FIRST, and FOLLOW for all nonterminals of the grammar.

	nullable	FIRST	FOLLOW
E'			
E			
T			
V			
L			

(b) For all productions, compute the FIRST of the right-hand side, or the FOLLOW of the left-hand side, as appropriate.

	$A \rightarrow \alpha$	FIRST(α)	FOLLOW(A)
1	$E' \rightarrow E\,\$$		
2	$E \rightarrow -\,E$		
3	$E \rightarrow (\,E\,)$		
4	$E \rightarrow V\,T$		
5	$T \rightarrow -\,E$		
6	$T \rightarrow \epsilon$		
7	$V \rightarrow \mathbf{id}\,L$		
8	$L \rightarrow (\,E\,)$		
9	$L \rightarrow \epsilon$		

(c) Complete the *entire* LL(1) parse table below.

	-	(	)	**id**	$
E'					
E					
T					
V					
L					

4. Describe how to implement non-local variable access in typical block-structured, statically-scoped programming languages.

5. Using Sethi-Ullman register allocation, generate code for the following expression:

$$(a + b) * (e - (c * d))$$

First, draw a tree for the given expression and label each node with the number of registers it needs during evaluation. Then give the generated code, use instructions like $r_1 \leftarrow M[a]$ (load a into register 1), $r_2 \leftarrow r_3 \times r_4$, $r_1 \leftarrow r_3 - r_2$, etc.