

FINAL EXAM

Please read all instructions, including these, carefully.

- There are 8 questions on the exam with multiple parts. You have a maximum of 3 hours to work on the exam.
- The exam is open book, open notes.
- Please write your answers in the space provided on the exam, and clearly mark your solutions. You may use the backs of the exam pages as scratch paper, or use additional pages (available at the front of the room).

Each problem has a straightforward solution. Solutions will be graded on correctness and clarity. Partial solutions will be graded for partial credit.

NAME _____

Problem	Max points	Points
1	15	
2	15	
3	15	
4	10	
5	10	
6	10	
7	15	
8	10	
TOTAL	100	

(1) Consider the following grammar:

$$A' \rightarrow A$$

$$A \rightarrow xA \mid yA \mid y$$

(a) Fill in the LL(1) parse table for this grammar.

	x	y	$\$$
A'			
A			

(b) Explain why this grammar is not LL(1).

(c) Transform the grammar into a grammar that is LL(1).

(d) Give the parse table for the grammar created in (c). If you created additional non-terminals in your transformation, you will have to add rows to the table.

	x	y	$\$$
A'			
A			

(2) In this question you will determine if the (original) grammar in the previous question is SLR(1). To answer, construct the following:

(a) The canonical collection of LR(0) items

(b) The SLR(1) parse table

(c) Is the grammar SLR(1)? Explain your answer.

- (3) For the same grammar as in the previous two questions, construct the following:
- (a) The canonical collection of LR(1) items

(b) The LR(1) parse table

(c) Is the grammar LR(1)? Explain your answer.

(4) (a) Give an *unambiguous* grammar that is not LR(1).

(b) Explain why the following grammar is LL(1) but not SLR(1):

$$\begin{aligned} X &\rightarrow Y a Y b \mid Z b Z a \\ Y &\rightarrow \epsilon \\ Z &\rightarrow \epsilon \end{aligned}$$

(5) Consider the following grammar \square

$$\begin{aligned}
 S &\rightarrow A \\
 A &\rightarrow A+A \mid B++ \\
 B &\rightarrow y
 \end{aligned}$$

(a) Draw the parse tree for the input “ $y + + + y + +$ ”

(b) Complete the table below tracing an LR(1) parse of the input above. The *Stack* column shows the stack (with the top at right), the *Input* column shows the not-yet-processed input, and the *Action* column shows whether the parser performs a shift action, a reduce action, or accepts the input. In the case of a reduce action, indicate which production is used. (NOTE: you do not need to construct the parse table; just use your knowledge of how the parser works and the parse tree given in the previous part.)

Stack (with top at right)	Input	Action

(6) Consider the following grammar :

- 1 $S \rightarrow [SX]$
- 2 | a
- 3 $X \rightarrow \epsilon$
- 4 | $+ SY$
- 5 | Yb
- 6 $Y \rightarrow \epsilon$
- 7 | $- SXc$

(a) Fill in the table below with the First and Follow sets for the non-terminals in this grammar:

	<i>First</i>	<i>Follow</i>
S		
X		
Y		

(b) Using the parse table for the grammar above, show a top-down parse of the string $[a+a-ac]$. The numbers in the cells refer to productions, as numbered in the grammar.

	a	b	c	+	-	[]	\$
S	2					1		
X		5	3	4	5		3	
Y		6	6		7		6	

Stack (with top at left)	Input	Action

- (7) The following context-free grammar describes part of the syntax of a Pascal-like language. Terminals are given in capitals and terminals in lower case. VAR represents a variable name and CONST represents a constant. The productions for ASSN-STMT are not given, as they do not affect the question.

```
1 PROGRAM    →  PROCEDURE STMT-LIST
2 STMT-LIST  →  STMT STMT-LIST
3            |  STMT
4 STMT       →  do VAR = CONST to CONST begin STMT-LIST end
5            |  ASSN-STMT
```

- (a) Show the parse tree for the following:

```
Procedure
  do i = 1 to 100 begin
    ASSN-STMT
    ASSN-STMT
  end
ASSN-STMT
```

- (b) Write an attribute grammar that computes the number of executed statements for a program conforming to this grammar. Use the fact that the number of times a loop executes can be computed from the two constants that specify the range of the loop variable. You may assume that the lower bound of the range is less than or equal to the upper bound.

PROGRAM \rightarrow PROCEDURE STMT-LIST	
STMT-LIST ₀ \rightarrow STMT STMT-LIST ₁	
STMT-LIST \rightarrow STMT	
STMT \rightarrow do VAR = CONST ₁ to CONST ₂ begin STMT-LIST end	
STMT \rightarrow ASSN-STMT	

- (c) Annotate the parse tree for the program fragment given in part (a) with the computed attribute values. (Annotate the parse tree you drew on the previous page, rather than re-draw it here.)
- (d) For each attribute used in your attribute grammar, indicate whether it is *synthesized* or *inherited*.

(8) Consider the following grammar:

$$\begin{aligned} S &\rightarrow aS \mid Ab \\ A &\rightarrow XY\zeta \mid \varepsilon \\ X &\rightarrow cS \mid \varepsilon \\ Y &\rightarrow dS \mid \varepsilon \\ \zeta &\rightarrow eS \end{aligned}$$

(a) Give a leftmost derivation of the string $aebb$.

(b) Explain why it is that if we add the production $X \rightarrow bS$, the grammar is no longer LL(1).