

EXAM

Please read all instructions, including these, carefully

- There are 7 questions on the exam, with multiple parts. You have 3 hours to work on the exam.
- The exam is open book, open notes.
- Please write your final answers in the space provided on the exam. 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 given partial credit.

NAME : _____

Problem	Max points	Points
1	15	
2	10	
3	10	
4	10	
5	25	
6	15	
7	15	
TOTAL	100	

1. Consider the following augmented CFG for prefix expressions.

$$\begin{aligned} E' &\rightarrow E \\ E &\rightarrow + E E \mid - E E \mid \text{id} \end{aligned}$$

- (a) Compute the following closures of items.

i) $I_a = \text{closure}(\{E' \rightarrow \bullet E\})$

ii) $I_b = \text{closure}(\{E \rightarrow + \bullet EE\})$

iii) $I_c = \text{closure}(\{E \rightarrow +E \bullet E\})$

iv) $I_d = \text{closure}(\{E \rightarrow \text{id} \bullet\})$

(b) Show the following entries of the SLR(1) parsing table M , where I_b and I_d indicate the states corresponding to the items sets computed in (a).

i. $M[I_b, E]$

ii. $M[I_d, +]$

iii. $M[I_d, \$]$

2. Consider extending the LR(1) case to the general k lookahead symbols. Given an LR(k) grammar with N nonterminals and T terminals, how many columns would we have in a corresponding LR(k) action/goto table? State your assumptions and show your work for full credit.

3. Consider the following grammar G .

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow \text{id} \mid \text{id} () \mid \text{id} (L) \\ L &\rightarrow E ; L \mid E \end{aligned}$$

- (a) Give an LL(1) grammar that generates the same language as G .
- (b) Prove that your grammar is LL(1) by whatever method you like. (Hint: This does not require an exhaustive construction.) Don't just restate the definition of LL(1).

4. Here is an attribute grammar G for arithmetic expressions using multiplication, unary -, and unary +.

$$\begin{aligned} S &\rightarrow E \{ \text{if } (E.\text{sign} == \text{POS}) \text{ print("result is positive");} \\ &\quad \text{else print("result is negative"); } \} \\ E &\rightarrow \text{unsigned_int} \\ &\quad | + E \\ &\quad | - E \\ &\quad | E * E \end{aligned}$$

- (a) Add semantic actions to compute an attribute $E.\text{sign}$ for non-terminal E to record whether the arithmetic value of E is positive or negative. The attribute sign can have two values, either POS or NEG. To get you started, the first rule is provided.

- (b) Show the parse tree for input $2 * - 3$ (where 2 and 3 are “unsigned_ints”). Indicate at each node what the values of associated attributes are.

- (c) Is $E.\text{sign}$ an inherited attribute or a synthesized attribute?

5. Consider the following grammar G :

$$S \rightarrow Sa \mid bL$$

$$L \rightarrow ScL \mid Sd \mid a$$

(a) Remove left recursion from G . The result is called G' .

(b) Left factor G' . The result is G'' .

(c) Fill in the table below with the FIRST and FOLLOW sets for the non-terminals in grammar G'' (note that the number of rows may be more than you need):

Non-terminal	<i>First</i>	<i>Follow</i>

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

Non-terminal	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	$\$$

(e) Using the parse table in (d), show a top-down parse of the string *bbacbada*.

Stack (with top at left)	Input	Action

6. Consider the following simple context free grammars:

G_1 :

$S \rightarrow Aa$

$A \rightarrow \varepsilon$

$A \rightarrow bAb$

G_2 :

$S \rightarrow Aa$

$A \rightarrow \varepsilon$

$A \rightarrow Abb$

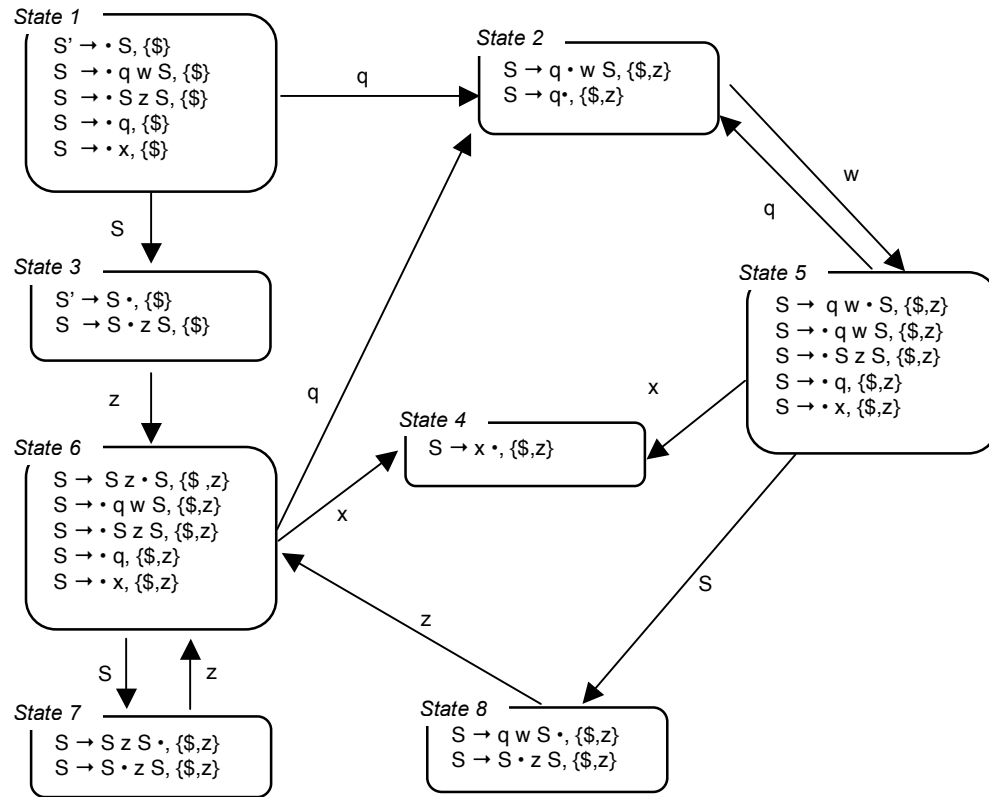
Note that the grammars generate the same language: strings consisting of even numbers of b 's (including 0 of them), followed by an a .

(a) Attempt to show a shift-reduce parse of the string $bbbba$ for a parser for grammar G_1 . Show the contents of the stack, the input, and the actions (i.e., shift, reduce, error, accept). You don't need to create a parse table; just use your knowledge of the grammar and how the parser works. Be sure to indicate any conflicts and explain why they are conflicts. Is G_1 LR(1)? Is G_1 LR(0)?

(b) Attempt to show a shift-reduce parse of the string $bbbba$ for a parser for grammar G_2 . Show the contents of the stack, the input, and the actions (i.e., shift, reduce, error, accept). You don't need to create a parse table; just use your knowledge of the grammar and how the parser works. Be sure to indicate any conflicts and explain why they are conflicts. Is G_2 LR(1)? Is G_2 LR(0)?

- (c) Indicate whether G_1 and G_2 are LL(1) and explain your answer. You don't need to construct the parse tables, but may argue from other properties.
- (d) Of the language classes we have discussed, which is the *smallest* category into which $L(G_1)$ fits? Justify your answer.

7. Consider the following GOTO graph of an LALR(1) parser. Notice the state numbers and refer to them in the answers to the following questions.



(a) Which states of the parser have conflicts? What kind of conflicts are they, and under what circumstances do they arise?

(b) Describe the possible contents of the stack when the parser is in state 5.

(c) Give a string that could remain in the input when the parser is in state 5 that would lead to a successful parse.