CS143 Midterm
Spring 2023

- Please read all instructions (including these) carefully.
- There are 5 questions on the exam, some with multiple parts. You have 90 minutes to work on the exam.
- The exam is open note. You may use laptops, phones and e-readers to read electronic notes, but not for computation or access to the internet for any reason other than to access the class webpage.
- Please write your answers in the space provided on the exam, and clearly mark your solutions. Do not write on the back of exam pages or other pages.
- Solutions will be graded on correctness and clarity. Each problem has a relatively simple and straightforward solution. You may get as few as 0 points for a question if your solution is far more complicated than necessary. Partial solutions will be graded for partial credit.

NAME: _____

In accordance with both the letter and spirit of the Honor Code, I have neither given nor received assistance on this examination.

SIGNATURE: _____

Problem	Max points	Points
1	15	
2	15	
3	20	
4	25	
5	25	
TOTAL	100	

1. Regular Languages

(a) Construct a regex that recognizes binary numbers ($\Sigma = \{0, 1\}$) that are divisible by 2^n for a given n. You may use the shorthand 0^n or 1^n to respectively represent a sequence of n 0's or 1's. You may consider the empty string ε as equivalent to 0 and the answer may have leading zeros.

(b) Construct an NFA that recognizes strings in the alphabet $\Sigma = \{1, 2, 3\}$, where adding the digits together results in a number that is a multiple of 2. That is, "11" and "222" are valid strings, but not "23". The empty string epsilon is included in the language.

2. Semantic Actions

Consider the set of base-3 numbers over the digits $\Sigma = \{0, 1, 2\}$. Give a syntax directed translation (a set of CFG productions and associated semantic actions) that assigns to the root of your parse tree an attribute equal to the base-10 representation of the input string. For example, "201" should evaluate to $2 \times 3^2 + 0 \times 3 + 1 = 19$. The empty string ε should not be included in your grammar. You should not use global variables.

3. Context-Free Grammars

Give Context-Free Grammars (CFGs) that generate the following languages.

(a) $\{w \in (a|b)^* \mid$ the length of w is odd and the middle symbol is $a\}$.

(b) $\{a^i\ b^j\ c^k \mid i, j, k \geq 0$ and $i + j = k\}$ over the alphabet $\Sigma = \{a, b, c\}$, where a^i means i repetitions of a and where $a^0 = \varepsilon$.

4. Top-Down Parsing

Consider the following CFG over the language $\Sigma = \{a, b, (,), -, 0, 1\}$:

$S \to aR(V \mid b(Q$
$Q \to bV1 \mid RQ \mid 0)-$
$V \to 0V \mid)-$
$R \to aR \mid 1$

(a) Construct the LL(1) parsing table for this grammar.

Answer:

	a	b	(	)	$-$	0	1
S							
Q							
V							
R							

(b) Is this grammar LL(1)? Explain why or why not.

Answer:

(c) Show the sequence of stack, input, and actions that occur during an LL(1) parse of the string "$a1(a11)-$". The acceptable actions are: "out <production>", "match <terminal>", "accept", and "error". The stack and input portions of the first line of the table are filled out for you.

Answer:

Stack	Input	Action
S $	a1(a11)-	

(d) Is this string in the context-free language described by this grammar? Explain why or why not.

Answer:

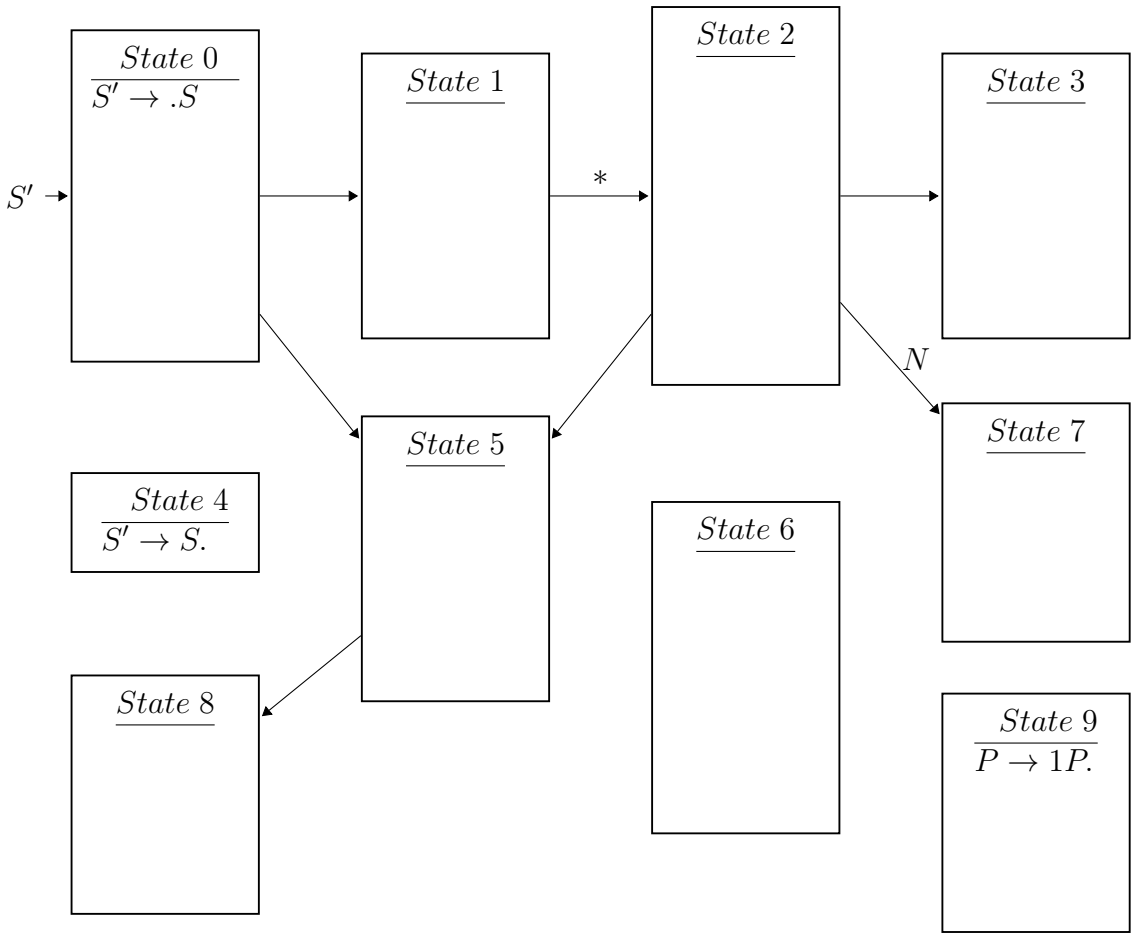
5. Bottom-Up Parsing

Consider the following grammar G over the alphabet $\{1, -, *\}$:

Production	Production Number
$S' \to S$	1
$S \to N * N$	2
$S \to N * P$	3
$N \to -P$	4
$P \to 1P$	5
$P \to 1$	6

You want to use an SLR(1) parser to parse strings defined by G. Notice that each production in G is associated with a production number.

(a) Provide the DFA edges and nodes of the LR(0) machine. A partial LR(0) DFA has already been provided. Specifically, you will need to write in missing node (state) labels, draw in missing edges (arrows). and complete all missing edge labels (for both provided and drawn-in edges).



(b) Fill in the SLR(1) parser table. The parser table includes both the action and goto tables. The follow sets of the nonterminal symbols are S, N, P.

$$\text{FOLLOW}(S) = \{\$\}$$
$$\text{FOLLOW}(N) = \{*, \$\}$$
$$\text{FOLLOW}(P) = \{*, \$\}$$

Each action table cell should only include si (shift), rj (reduce), or *accept* (accept), where i is a state number and j is a production number. Empty action table cells indicate error states. For instance, an acceptable action table entry would be $s1$. Each goto table cell should either contain a single state number or be empty. Solutions that do not follow this notation will not receive full credit.

state	action				goto			
	1	$-$	$*$	\$	S'	S	N	P
0								
1								
2								
3								
4								
5								
6								
7								
8								
9								