Midterm Review

CSCE 322

Name: _____

Instructions

Please solve the problems presented below. **Show your work to receive full credit; just an answer is not enough**. **No Approximations**.

Question 1 (12 points)

(a) Describe in English the language defined by the regular expression `^([1-9][0-9]*)?[13579]$`.

(b) Write an unambiguous context-free grammar that generates the same language.

(c) Using your grammar from part (b), give a derivation of the string `20140213`.

Question 2 (10 points)

Consider this top-down grammar for `if` statements:

$$
\begin{aligned}
S &\rightarrow \text{if (expression) } S \\
S &\rightarrow \text{if (expression) } S \text{ else } S \\
S &\rightarrow \text{other}
\end{aligned}
$$

Give two parse trees for the expression `if ( expr1 ) if( expr2 ) f(); else g();` that prove this grammar is ambiguous.

Question 3 (18 points)

Consider the following CFG for octal numbers.

$$
\begin{aligned}
O &\rightarrow N\ O \\
O &\rightarrow \epsilon \\
N &\rightarrow \texttt{0}|\texttt{1}|\texttt{2}|\texttt{3}|\texttt{4}|\texttt{5}|\texttt{6}|\texttt{7}
\end{aligned}
$$

Augment this grammar with attribute rules that will accumulate the value of the number into a `val` attribute of the root of the parse tree.

Question 4 (24 points)

For the regular expression `^(-)?[^0][0-9]*('.'[0-9]+)?$`, determine which of the following inputs will match

(a) `3.14159`

(b) `-2`

(c) `F.75`

(d) `0.7071`

(e) `.5`

(f)

(g) `->`

(h) `F80000`

Question 5 (24 points)

Consider this top-down grammar

$$\begin{aligned} S &\rightarrow aS \\ S &\rightarrow aSbS \\ S &\rightarrow \epsilon \end{aligned}$$

(a) Provide the parse tree for the input `aab`

(b) Is this language ambiguous? If so, provide an alternate parse tree for `aab` to prove it. If not, why not?

Question 6 (26 points)

Consider the following CFG for binary numbers.

$$
\begin{array}{lll}
B & \to Z & \\
 & \triangleright\ B\texttt{.twice = false} & \\
B & \to N\ M & \\
Z & \to \texttt{0} & \\
N & \to \texttt{1} & \\
M & \to Z\ M & \\
M & \to N\ M & \\
M & \to \epsilon &
\end{array}
$$

(a) Augment this grammar with attribute rules that will accumulate `true` into a `twice` attribute of the root of the parse tree if the string contains at least twice as many 1s as 0s, and `false` otherwise.

(b) Is your attribute grammar S-attributed?

Question 7 (15 points)

(a) What is the input to a scanner?

(b) What is the input to a parser?

(c) What is the input to a semantic analyzer?

Question 8 (18 points)

Consider the following CFG for hexidecimal numbers

$$
\begin{array}{rl}
H & \rightarrow N\ NS \\
NS & \rightarrow N\ NS \\
NS & \rightarrow \epsilon \\
N & \rightarrow \texttt{0} \\
 & \quad \triangleright N\texttt{.value = 0} \\
N & \rightarrow \texttt{1} \\
 & \quad \triangleright N\texttt{.value = 1} \\
N & \rightarrow \texttt{2} \\
 & \quad \triangleright N\texttt{.value = 2} \\
N & \rightarrow \texttt{3} \\
 & \quad \triangleright N\texttt{.value = 3} \\
N & \rightarrow \texttt{4} \\
 & \quad \triangleright N\texttt{.value = 4} \\
N & \rightarrow \texttt{5} \\
 & \quad \triangleright N\texttt{.value = 5} \\
N & \rightarrow \texttt{6} \\
 & \quad \triangleright N\texttt{.value = 6} \\
N & \rightarrow \texttt{7} \\
 & \quad \triangleright N\texttt{.value = 7} \\
N & \rightarrow \texttt{8} \\
 & \quad \triangleright N\texttt{.value = 8} \\
N & \rightarrow \texttt{9} \\
 & \quad \triangleright N\texttt{.value = 9} \\
N & \rightarrow \texttt{a} \\
 & \quad \triangleright N\texttt{.value = 10} \\
N & \rightarrow \texttt{b} \\
 & \quad \triangleright N\texttt{.value = 11} \\
N & \rightarrow \texttt{c} \\
 & \quad \triangleright N\texttt{.value = 12} \\
N & \rightarrow \texttt{d} \\
 & \quad \triangleright N\texttt{.value = 13} \\
N & \rightarrow \texttt{e} \\
 & \quad \triangleright N\texttt{.value = 14} \\
N & \rightarrow \texttt{f} \\
 & \quad \triangleright N\texttt{.value = 15}
\end{array}
$$

Augment this grammar with attribute rules that will accumulate the Decimal representation `dec` into the root of the parse tree.

Question 9 (12 points)

For the regular expression `^/'*'[^*/]*'*'/$`, determine which of the following inputs will match

(a) `/* This is a Java comment */`

(b) `// This is also a Java comment`

(c) `-- This is a Haskell comment`

(d) `% This is a Prolog comment`

(e) `/* Is *this* a Java comment? */`

(f)

(g) `/* What is the value of array[0]? */`

(h) `/* Is this a /* comment */ within a comment? */`

(i) `/* What is the value of x^2? */`

(j) `/* How much is $Texas ? */`

Question 10 (12 points)
Consider this grammar

$$
\begin{array}{rl}
A & \rightarrow I \texttt{ = } E \\
I & \rightarrow \texttt{a} \mid \texttt{b} \mid \texttt{c} \\
E & \rightarrow E \texttt{ + } E \\
E & \rightarrow E \texttt{ * } E \\
E & \rightarrow \texttt{(} E \texttt{)} \\
E & \rightarrow I
\end{array}
$$

(a) Provide the parse tree for the input `a = b + c * a`

(b) Is this language ambiguous? If so, provide an alternate parse tree for `a = b + c * a` to prove it. If not, why not?

Question 11 (12 points)

Consider the following CFG for a list of numerals.

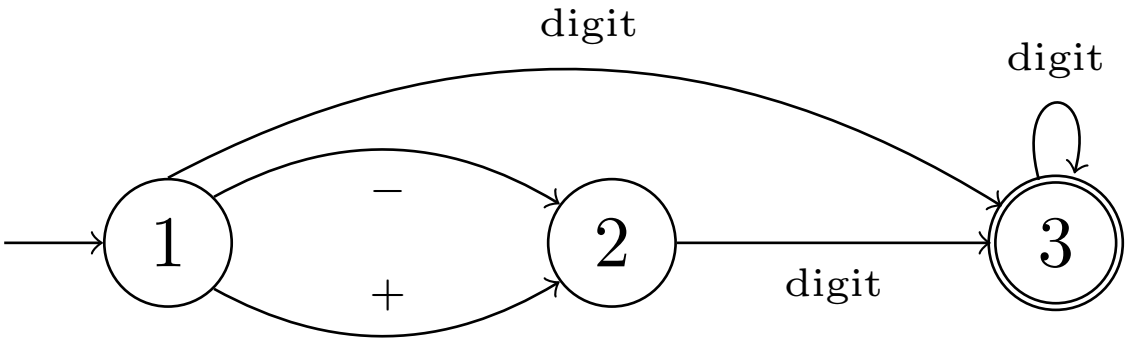
$L \rightarrow \epsilon$
  $\triangleright$ L.avg = 0
$L \rightarrow N\ L$
$N \rightarrow$ 0
  $\triangleright$ N.value = 0
$N \rightarrow$ 1
  $\triangleright$ N.value = 1
$N \rightarrow$ 2
  $\triangleright$ N.value = 2
$N \rightarrow$ 3
  $\triangleright$ N.value = 3
$N \rightarrow$ 4
  $\triangleright$ N.value = 4
$N \rightarrow$ 5
  $\triangleright$ N.value = 5
$N \rightarrow$ 6
  $\triangleright$ N.value = 6
$N \rightarrow$ 7
  $\triangleright$ N.value = 7
$N \rightarrow$ 8
  $\triangleright$ N.value = 8
$N \rightarrow$ 9
  $\triangleright$ N.value = 9

(a) Augment this grammar with attribute rules that will accumulate the average of the list into an `avg` attribute at the root of the parse tree. Hint: The first number in a list of five numbers contributes 20% of its value to the average; the average of the last four numbers accounts for the other 80% of the overall average.

(b) Is your attribute grammar S-attributed?

Question 12 (12 points)

Create the scanner table for this finite state automata that describes optionally signed integers,



Question 13 (12 points)

(a) Provide two examples of imperative languages

(b) Provide two examples of functional languages

(c) Provide one example of logic languages

Question 14 (12 points)

Consider this grammar

$$
\begin{array}{rl}
A & \rightarrow I \texttt{ = } E \mid I \texttt{ = } E \texttt{ + } E \\
I & \rightarrow \texttt{a} \mid \texttt{b} \mid \texttt{c} \\
E & \rightarrow E \texttt{ * } E \\
E & \rightarrow \texttt{(} E \texttt{)} \\
E & \rightarrow I
\end{array}
$$

(a) Provide the parse tree for the input `a = b + c * a`

(b) Is this language ambiguous? If so, provide an alternate parse tree for `a = b + c * a` to prove it. If not, why not?

Question 15 (12 points)

Consider the following CFG for a list of numerals.

$$
\begin{array}{rl}
L & \to N\ LT \\
LT & \to \epsilon \\
LT & \to \texttt{,}\ N\ LT \\
N & \to \texttt{0} \\
 & \quad \triangleright N\texttt{.value = 0} \\
N & \to \texttt{1} \\
 & \quad \triangleright N\texttt{.value = 1} \\
N & \to \texttt{2} \\
 & \quad \triangleright N\texttt{.value = 2} \\
N & \to \texttt{3} \\
 & \quad \triangleright N\texttt{.value = 3} \\
N & \to \texttt{4} \\
 & \quad \triangleright N\texttt{.value = 4} \\
N & \to \texttt{5} \\
 & \quad \triangleright N\texttt{.value = 5} \\
N & \to \texttt{6} \\
 & \quad \triangleright N\texttt{.value = 6} \\
N & \to \texttt{7} \\
 & \quad \triangleright N\texttt{.value = 7} \\
N & \to \texttt{8} \\
 & \quad \triangleright N\texttt{.value = 8} \\
N & \to \texttt{9} \\
 & \quad \triangleright N\texttt{.value = 9}
\end{array}
$$

(a) Augment this grammar with attribute rules that will accumulate the maximum of the list into a `max` attribute at the root of the parse tree.

(b) Is your attribute grammar S-attributed?

Question 16 (20 points)

For the regular expression `^(<[^>]*>[A-Za-z0-9+=/]*</[^>]*>)+$`, determine which of the following inputs will match

(a) `<b>HELLO</b>`

(b) `<pre>HELLO`

(c) `<b>5>4</b>`

(d) `<b>1+1=2</b>`

(e) `<b>5-2=3</b>`

(f)

(g) `<b>1</pre>`

(h) `<></>`

(i) `<b>OK</b></pre>`

(j) `<pre><b>OK</b></pre>`

Question 17 (12 points)

Consider this grammar

$$
\begin{array}{rl}
S & \rightarrow \texttt{a}\ S \\
S & \rightarrow S1\ S \\
S & \rightarrow \epsilon \\
S1 & \rightarrow \texttt{a}\ S1\ S1\ \texttt{b} \\
S1 & \rightarrow \epsilon
\end{array}
$$

(a) Provide the parse tree for the input `aab`

(b) Is this language ambiguous? If so, provide an alternate parse tree for `aab` to prove it. If not, why not?

Question 18 (24 points)

Consider the following CFG for a list of vowels.

$$
\begin{array}{lll}
L & \to \epsilon \\
 & \quad \triangleright\ L\texttt{.all = false} \\
L & \to N\ L \\
N & \to \texttt{A} \\
N & \to \texttt{E} \\
N & \to \texttt{I} \\
N & \to \texttt{O} \\
N & \to \texttt{U}
\end{array}
$$

(a) Augment this grammar with attribute rules that will accumulate the value `true` into an `all` attribute at the root of the parse tree if the list contains at least one of each of the possible vowels (A,E,I,O,U), and `false` otherwise.

(b) Is your attribute grammar S-attributed?

Question 19 (24 points)

Create the scanner table for this finite state automata that describes amounts of US dollars.