Undergraduate Exam

CSE/304 Compiler Design

Mid-term Exam
Oct 27, 1998
Fall 98

Duration: 1 hour, 15 minutes.

Instructions:

Read the following carefully before answering any question.

1. Write your answers in the space provided.

2. Keep your answers brief and precise.

3. The exam consists of 5 questions, in 8 pages (including this page) for a total of 110 points.

4. You may attempt all questions. However, the maximum you can score in this exam is 100 points.

5. Doubts about the questions will be answered only in the first 15 minutes of the exam. So, read the questions carefully at the beginning of the exam.

6. Write your name and ID number in the space above.

7. The exam for undergraduate students is different from the exam for the graduate students. Make sure you are reading the correct question paper.

8. Make sure you have filled in your name and ID number in the space above.

9. Read the exam carefully before answering any question.

10. Good luck!

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>5</td>
</tr>
<tr>
<td>b.</td>
<td>5</td>
</tr>
<tr>
<td>c.</td>
<td>5</td>
</tr>
<tr>
<td>d.</td>
<td>5</td>
</tr>
<tr>
<td>e.</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>25</td>
</tr>
</tbody>
</table>

Consider the following two languages:

L1: strings over \{a, b, g\} such that they contain an even number of a's.

L2: strings over \{a, b, g\} such that the number of a's is a multiple of 3.

a. Write a regular expression corresponding to L1.

b. Write a regular expression corresponding to L2.

c. Draw an NFA that accepts L1.

d. Draw an NFA that accepts L2.

GOOD LUCK!
In text classes containing lower-case or upper-case English letters, numbers and common English punctuation symbols (quotations are), by convention, written by enclosing them between a pair of double quotes ("), there are no escape sequences within quotations and quotations can span multiple lines.

Write a Lex specification that copies a \texttt{cle} from stdin to stdout, replacing the double quotes in the beginning of each quotation to two backward quotes (\texttt{`}), and the double quotes that end each quotation to two single quotes (\texttt{'}). For instance, for the input text/:

\begin{verbatim}
He turned and said to me, "The answer is no!", and then he left.
\end{verbatim}

the output should be:

\begin{verbatim}
He turned and said to me, `The answer is no!', and then he left.
\end{verbatim}

Consider the context-free grammar $G$:

\begin{align*}
S & \rightarrow aSbS \\
& \rightarrow bSaS \\
\end{align*}

(a) (6 points) Show that $G$ is ambiguous by drawing two different parse trees for the string $abab$.

(b) (6 points) Write down two right-most derivations of the string $abab$. Each derivation will correspond to one of the two parse trees of part (a).
Compute FIRST and FOLLOW.

Is $G_{3}$ LL? Justify.

Hint: You need not build the SLR parsing table!

Is $G_{3}$ SLR? Justify.

Hint: You need not build the SLR parsing table!

Consider the following simplified grammar, $G_{4}$ of a formatting language:

$$
T, ! T \mathit{Exp} \ T, ! T \mathit{sqrt} \ T, ! T \mathit{frac} \ T, ! T, ! T \mathit{id} \ a, \ldots
$$

When grammar $G_{4}$ is specified in bison/yacc, we get 6 shift/reduce conflicts. In the output produced by bison/yacc, one of the states with conflicts is reported as:

```
state 10
T - T TOK_exp T /rule 1/ TOK_exp T /rule 1/
T - T TOK_exp shift/
T TOK_exp /rule 1/ reduce using rule 1/tok
T TOK_exp /rule 1/ reduce using rule 1/tok
T - T TOK_exp /rule 1/ default reduce using rule 1/tok
```

What entries in the parsing table (e.g., what shifts and reductions) do the above conflicts correspond to?

**Total:** 5 points

**Points for partial credit:**

- 1 point for specifying the LL or SLR status of $G_{4}$.
- 1 point for specifying the number of shift/reduce conflicts.
- 2 points for identifying the shift/reduce conflicts.
- 1 point for explaining why the conflicts occur.

**Points for full credit:**

- 5 points for specifying the correct states and terminals in the parsing table.
The shift/reduce conflicts can be eliminated by specifying the precedence and associativity of the operators, i.e., exp, sqrt, and frac, in G. The rules for precedence and associativity are:

- exp is right associative,
- frac is left associative and
- sqrt is nonassociative.

exp has highest precedence, followed by sqrt, followed by frac, which has the lowest precedence.

Specify, in the form of yacc declarations, the operator precedence and associativity information for G. How will the shift/reduce conflicts in state 10 (from part (a) of this question) be resolved if the precedence and associativity you defined in part (b) are made available to the parser? 

Let G be an LL(1) grammar with $T$ as the set of terminal symbols and $N$ as the set of nonterminal symbols. a. (6 points) Estimate the size of LL(1) parsing table for G.

b. (6 points) Estimate the size of the parse tree produced by G for a string of length n. You may assume that G has no -productions and each production in G is of the form A → B C or A → a where a is a terminal symbol.

END OF EXAM