CSE-304 Compiler Design

Fall '98

Mid-term Exam

Oct 27, 1998.

Duration: 1 hours, 15 minutes.

ID Number : _____

Name : _____

INSTRUCTIONS

Read the following carefully before answering any question.

- MAKE SURE YOU HAVE FILLED IN YOUR NAME AND ID NUMBER IN THE SPACE ABOVE.
- The exam for undergraduate students is different from the exam for the graduate students. Make sure you are reading the correct question paper.
- 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.
- Write your answers in the space provided.
- Keep your answers brief and precise.
- The exam consists of 5 questions, in 8 pages (including this page) for a total of **110** points.
  You may attempt all questions. However, the maximum you can score in this exam is **100** points.
  Each of questions 1, 2, and 5 are on a single page.
  Questions 3 and 4 have multiple parts and are over two pages each.

GOOD LUCK!

Question	Points
1.	
2.	
3.	
4.	
5.	
Total	

Undergraduate Exam

1. **[Total: 28 points]** Consider the following two languages:
   L_1: strings over $\{a, b\}$ such that they contain even number of a's.
   L_2: strings over $\{a, b\}$ such that the number of a's is a multiple of 3.

   a. (7 points) Write a regular expression corresponding to L_1.

   b. (7 points) Write a regular expression corresponding to L_2.

   c. (7 points) Draw an NFA that accepts L_1.

   d. (7 points) Draw an NFA that accepts L_2.

2. **[20 points]** In text files (plain ASCII files 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 (**"**). Since text files are read only by humans, there are no escape sequences within quotations and quotations can span multiple lines.

   Write a Lex specification that copies a file from `stdin` to `stdout`, replacing the double quotes in the beginning of each quotation to two backward quotes (i.e., ‘‘) and the double quotes that end each quotation to two single quotes (i.e., ’’).

   For instance, for the input text:

   ```
   He turned and said to me, "The answer is no!", and then he left.
   ```

   the output should be:

   ```
   He turned and said to me, ‘‘The answer is no!’’, and then he left.
   ```


3. **[Total: 25 points]** Consider the context-free grammar G_3:

   $$\begin{aligned} S &\longrightarrow aSbS \\ S &\longrightarrow bSaS \\ S &\longrightarrow \epsilon \end{aligned}$$

   a. (6 points) Show that G_3 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)).

c. (5 points) Compute $FIRST(S)$ and $FOLLOW(S)$.

d. (4 points) Is G_3 $LL(1)$? Justify.

e. (4 points) Is G_3 $SLR(1)$? Justify.
[**Hint:** You need not build the SLR(1) parsing table!]

4. [**Total: 25 points**] Consider the following simplified grammar, G_4, of a formatting language for math equations:

$$
\begin{array}{lcll}
T & \longrightarrow & T \texttt{ exp } T & \text{(Exponent)} \\
T & \longrightarrow & \texttt{sqrt } T & \text{(Square root)} \\
T & \longrightarrow & T \texttt{ frac } T & \text{(Fraction)} \\
T & \longrightarrow & (\ T\ ) & \\
T & \longrightarrow & \texttt{id} &
\end{array}
$$

a. (8 points) When grammar G_4 is specified in bison/yacc, we get 6 shift/reduce conflicts. In the `.output` file produced by bison/yacc, one of the states with conflicts is reported as:

```
state 10

    T  ->  T . TOK_exp T   (rule 1)
    T  ->  T TOK_exp T .   (rule 1)
    T  ->  T . TOK_frac T   (rule 3)

    TOK_exp  shift, and go to state 7
    TOK_frac shift, and go to state 8

    TOK_exp  [reduce using rule 1 (t)]
    TOK_frac [reduce using rule 1 (t)]
    $default reduce using rule 1 (t)
```

What entries in the parsing table (i.e., what state and terminal symbols) do the above conflicts correspond to?

b. (10 points) The shift/reduce conflicts can be eliminated by specifying the precedence and associativity information for the operators, i.e., `exp`, `sqrt` and `frac`, in G_4. 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/bison declarations, the operator precedence and associativity information for G_4.

c. (7 points) How will the shift/reduce conflicts in `state 10` (from part (a) of this question) be resolved by bison/yacc if the precedences and associativity you defined in part (b) were added to the bison/yacc specifications?

Undergraduate Exam

5. **[Total 12 points]** Let G_5 be an LL(1) grammar with Σ as the set of terminal symbols and Π as the set of nonterminal symbols.

a. (6 points) Estimate the size of LL(1) parsing table for G_5.

b. (6 points) Estimate the size of the parse tree produced by G_5 for a string of length n. You may assume that G_5 has no ϵ-productions, and each production in G_5 is of the form $A \longrightarrow B\ C$ (where B and C are nonterminals) or $A \longrightarrow a$ (where a is a terminal).

END OF EXAM