

CSE-504 Compiler Design

Spring '08

Mid-term Exam

March 13, 2008.

Duration: 1 hour, 20 minutes.

Name: _____

USB ID Number: _____

INSTRUCTIONS

Read the following carefully before answering any question.

- Make sure you have filled in your name and USB ID number in the space above.
- Write your answers in the space provided; Keep your answers brief and precise.
- The exam consists of 4 questions, in 9 pages (including this page) for a total of **30** points.

Question 1 has two pages

Question 2 is on a single page.

Question 3 has three pages.

Question 4 has two pages.

GOOD LUCK!

Question	Max.	Score
1.	8	
2.	4	
3.	10	
4.	8	
Total:	30	

1. **[Total: 8 points]** Consider the following three languages:

L_1 : strings over $\{a, b, c\}$ that contain an even number of a 's. (*Note:* zero is an even number.)

L_2 : strings over $\{a, b, c\}$ that have ab as a substring.

Note: A substring of string s is obtained by deleting zero or more symbols from the beginning and end of s ; for instance **banana**, **nan** and ϵ are substrings of **banana**

L_3 : strings over $\{a, b, c\}$ that do not have ab as a substring.

a. (1 point) Write a regular expression corresponding to L_1 .

b. (1 point) Write a regular expression corresponding to L_2 .

c. (1.5 points) Draw a (possibly Nondeterministic) Finite Automaton that accepts L_1 .

d. (1.5 points) Draw a (possibly Nondeterministic) Finite Automaton that accepts L_2 .

[CONTINUED ON NEXT PAGE]

[CONTINUED FROM PREVIOUS PAGE]

f. (1.5 points) Write a regular expression corresponding to L_3 .

e. (1.5 points) Draw a (possibly Nondeterministic) Finite Automaton that accepts L_3 .

2. [4 points]

Consider the following Lex specification:

```

digit  [0-9]
alpha  [a-zA-Z]
%%
"class"          {return{CLASS};}
{alpha}({alpha}|{digit})* {return(ID);}
"return"         {return(RETURN);}
{digit}+         {return{INT};}
[ \t\n]          {}
.                {return(UNKNOWN);}
--.*             {return(COMMENT);}
%%

```

For each of the following inputs, write the token stream that will be generated by the above Lex specification.

Input	Token Stream
class class1	
\$-123.0e+6	
class --class1	
return return21	

3. **[Total: 10 points]** Consider the context-free grammar G_3 with start symbol S :

$$S \longrightarrow T U \mathbf{f}$$

$$T \longrightarrow \mathbf{a} T \mid \epsilon$$

$$U \longrightarrow \mathbf{a} \mathbf{b} U \mid \epsilon$$

- a. **(3 points)** Compute *FIRST* and *FOLLOW* sets for the three nonterminal symbols, S , T and U .

- b. **(2 points)** Is G_3 *LL(1)*? Justify.

[CONTINUED ON NEXT PAGE]

[CONTINUED FROM PREVIOUS PAGE]

c. (2 points) Show that G_3 is not $SLR(1)$.

[CONTINUED ON NEXT PAGE]

[CONTINUED FROM PREVIOUS PAGE]

- d. (3 points)** A small modification to G_3 will make it $SLR(1)$. Suggest such a modification and show that the resultant grammar is $SLR(1)$.

4. **[Total: 8 points]** Consider the following simplified grammar, G_4 , of a formatting language for math equations:

$T \longrightarrow T \text{ exp } T$	(Exponent)
$T \longrightarrow \text{sqrt } T$	(Square root)
$T \longrightarrow T \text{ frac } T$	(Fraction)
$T \longrightarrow (T)$	
$T \longrightarrow \text{id}$	

- a. **(2 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. (3 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 following examples (strings in the language specified by G_4 , and the corresponding math formulas) illustrate the associativity and precedence of the three operators:

String	Formula	
<code>a exp b frac n</code>	$\frac{a^b}{n}$	
<code>a exp n exp b</code>	a^{n^b}	equivalent to $a^{(n^b)}$
<code>sqrt a frac n</code>	$\frac{\sqrt{a}}{n}$	
<code>sqrt a exp n</code>	$\sqrt{a^n}$	
<code>a frac b frac c</code>	$\frac{a}{\frac{b}{c}}$	equivalent to $\frac{a}{b/c}$

Specify, in the form of yacc/bison declarations, the operator precedence and associativity information for G_4 that is consistent with the above examples.

- c. (3 points)** How will the shift/reduce conflicts in **state 10** be resolved by bison/yacc if the precedences and associativity you defined in part (b) were added to the bison/yacc specifications?

END OF EXAM