

CSE-504 Compiler Design

Spring '12

Mid-term Exam

March 14, 2012.

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 7 pages (including this page) for a total of **30** points.

Question 1 is on a single page.

Question 2 has two pages

Question 3 is on a single page.

Question 4 has two pages.

Question	Max.	Score
1.	8	
2.	8	
3.	6	
4.	8	
Total:	30	

1. [Total: 8 points]

L_1 : strings over $\{a, b, c\}$ such that there are even number of b 's between every pair of a 's. (*Note*: zero is an even number.)

L_2 : strings over $\{a, b, c\}$ such that all a 's occur before any b . For example, $acacbbcb$ is in L_2 , but $acacbbab$ is not in L_2 .

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

L_4 : strings over $\{a, b, c\}$ where the total number of a 's and b 's is an even number.

L_5 : strings over $\{a, b, c\}$ where the total number of a 's and b 's is divisible by 3.

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

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

(c) [2 points] Is L_3 regular? Justify.

(d) [2 points] Is L_4 regular? Justify.

(e) [2 points] Is L_5 regular? Justify.

2. [Total: 8 points] For this question, consider grammar G_2 with start symbol S given below:

$$S \rightarrow A \mathbf{a} A \mathbf{b}$$

$$S \rightarrow B \mathbf{b} B \mathbf{a}$$

$$A \rightarrow \mathbf{c}$$

$$A \rightarrow \epsilon$$

$$B \rightarrow \mathbf{d}$$

$$B \rightarrow \epsilon$$

(a) [2 points] Compute *FIRST* and *FOLLOW* of each of the following symbols.

Symbol	<i>FIRST</i>	<i>FOLLOW</i>
S		
A		
B		

(b) [3 points] Is G_2 an $LL(1)$ grammar? Justify.

(c) [3 points] Is G_2 an $SLR(1)$ grammar? Justify.

3. [Total: 6 points] Consider the following Bison/Yacc parser specification:

```
%token a b c
%%
```

```
S : S a S
   | S b S
   | c;
```

When compiled using Bison/Yacc, 4 shift/reduce conflicts are reported. The following is a fragment of the .output file that indicates these four conflicts:

state 6	state 7
<pre> 1 S: S . a S 1 S a S . 2 S . b S a shift, and go to state 4 b shift, and go to state 5 a [reduce using rule 1 (S)] b [reduce using rule 1 (S)] \$default reduce using rule 1 (S)</pre>	<pre> 1 S: S . a S 2 S . b S 2 S b S . a shift, and go to state 4 b shift, and go to state 5 a [reduce using rule 2 (S)] b [reduce using rule 2 (S)] \$default reduce using rule 2 (S)</pre>

Now, consider the addition of associativity and precedence declarations for tokens **a** and **b** to the Bison/Yacc specification such that the shift/reduce conflicts are eliminated as follows. In both states 6 and 7, the shift/reduce conflict on symbol **b** is resolved as “shift”; all other conflicts are resolved as “reduce”.

(a) Which symbol, “a” or “b”, has higher precedence? Why?

(b) What is the associativity of “a”? Why?

(c) What is the associativity of “b”? Why?

4. [Total: 8 points] Consider the following syntax directed definition written using the attribute grammar notation:

$$\begin{array}{lll}
 S & \rightarrow & A B C \quad \{S.v = B.y + C.z; \quad B.x = A.w\} \\
 A & \rightarrow & \mathbf{a} \quad \{A.w = 1\} \\
 B & \rightarrow & \mathbf{b} B \quad \{B_1.x = B.x + 1; \quad B.y = B_1.y + B.x\} \\
 B & \rightarrow & \epsilon \quad \{B.y = B.x\} \\
 C & \rightarrow & \mathbf{c} \quad \{C.z = 3\}
 \end{array}$$

- (a) [2 points] For each of the attributes, state whether the attribute is synthesized or inherited.

Attribute	Synthesized/Inherited
$S.v$	
$A.w$	
$B.x$	
$B.y$	
$C.z$	

- (b) [3 points] What is the value of attribute $S.v$ for sentence **abbc**?

- (c) [3 points] Show an encoding of the above syntax-directed definition in Yacc/Bison. For full credit, your Yacc/Bison specification should evaluate the attributes at the correct time (i.e. the value of an attribute must be computed before it is used).