BRAC UNIVERSITY

Department of Computer Science and Engineering
CSE420: Compiler Design
Final Examination, Fall 2014
Duration: 2.30 hours, Total Marks: 40
THERE ARE SIX (6) QUESTIONS. ANSWER ANY FOUR (4)

1. (a) Write the regular expression over the alphabet $\sum$ = {b, c}:

i) All strings containing at least 2c's [2+2=4]
ii) All strings of the form, $b^{2m}c^{3n}$, where m, $n \geq 1$

(b) Consider the following grammar for Boolean expression:

$E \rightarrow E \mid E$ $E \rightarrow (E)$ $E \rightarrow id$
$E \rightarrow E \,\&\, E$ $E \rightarrow true$
$E \rightarrow \sim E$ $E \rightarrow false$

i) Show that this grammar is ambiguous.
ii) Rewrite the grammar to remove the ambiguity. [2+2=4]

(c) Draw an NFA for $(x+y)^*px^*p(x+y)^*$ over the alphabet $\sum$ = {x, y, p}. [2]

2. (a) Convert the regular expression mm(m|n)*n over the alphabet $\sum$ = {m,n} directly to DFA.

(b) What is handle? Explain with an example.
(c) Using an example, describe how the DFA state minimization algorithm works? [7+2+1=10]

3. (a) Consider the following augmented grammar:

0. E' → E $
1. E → id
2. E → (E)
3. E → (id) E

i)Draw the LR (0) automation for this grammar. [4+3+1=8]
ii) Construct simple LR (SLR) parsing table.
iii) Is the grammar LR (0)? Why? Why not?

(b) Draw DAG for the following expression. [2]

a*(b+c) + (b+c)*d + a*(b^c) + d*(b^c)

4. (a) Consider the following grammar:

$E \rightarrow [L] \mid a$
$L \rightarrow ET$
$T \rightarrow ,L \mid \varepsilon$

i)Construct LL parsing table for the above grammar .
ii) Using predictive parsing algorithm, determine whether it is possible to parse the string [a]? Show the moves by predictive parser for this task. [4+3=7]

(b) Consider the following code fragment:
do i=1+1;
f=i*5000; while (a[i] < v);
Write the three address code and its quadruple representation. [3]

5. (a) Consider the following grammar: [3+4=7]
T → FT'
T' → +FT'
T' → ε
F → 1 | 2 | 3 | . . . | 9
i) Construct an SDD for the grammar.
ii) Using SDD constructed in (i) give an annotated parse tree for the expression: 2+3+4.

(b) Consider the augmented grammar: [3]
0. S → N
1. N → NB
2. N → B
3. B → 1
4. B → 0
Show a bottom-up parsing of the string 1101, using the LR parsing table below.

State	1	0	$	N	B
0	S3	S4		1	2
1	S3	S4	ACCEPT		5
2	R2	R2	R2		
3	R3	R3	R3		
4	R4	R4	R4		
5	R1	R1	R1		

6. (a) Write an Semantic Rules to generate 3-address code for the following grammar: [5+5 =10]
P → S
S → assign | if (B) S1 | S1 S2
B → B1 || B2 | B1 && B2 | id1 rel id2 | true | false
According to your SDD, write down the code that will be generated for the following expression.
if(x>200 && x!=y || x<100)
x=50;

(b) Given the following grammar, G =({S, A, B}, S, {a, b, x}) with the following productions:
1. S → A
2. S → xb
3. A → aAb
4. A → B
5. B → b
Computer the LR(1) items and the corresponding DFA and construct the parsing table.