

**03-60-214 COMPUTER LANGUAGES, GRAMMARS, AND
TRANSLATORS****MIDTERM EXAM (B) AND SOLUTION**

Section No	Name	Student Number	Marks

Question	Max	Score
1	10	
2	5	
3	5	
4	5	
5	5	
6	10	
7	10	
8	30	
9	20	
total	100	

For multiple choices, select all correct answers.

1. (10%) Fill in the blanks using the terms listed below.

The grammar of an XML document is described by DTD, which is similar to CFG, and more powerful than RE. One type of XML parser generates a DOM parser tree. Parsing is also called hierarchical analysis.

scanner, linear analysis, hierarchical analysis, lexical analysis, lex, regex, Jlex, javaCUP, Yacc, CFG, Parser, RE, DFA, NFA, token, parse tree, DTD, XML, DOM.

2. (5%) Many compilers are divided into two largely independent passes: a front end, responsible for analyzing source code, and a back end, responsible for generating target code. What is the most important motivation for this division into passes?

- (a) to provide multiple combinations of front ends and back ends in a compiler family;
- (b) to facilitate division of labor when a compiler is built by a large team of programmers;
- (c) to minimize memory requirements on modern machines;
- (d) to facilitate debugging the compiler.

Your answer: a.

3. (5%) Which of the following could be said as the advantages of NFA over DFA?

- (a) better computational power;
- (b) may be smaller than its deterministic counterpart;
- (c) may be easier to design;
- (d) can jump to any state in a transition;
- (e) may be better as a computational model, since it is easier to understand.

Your answer: (b), (c).

4. (5%) Find the shortest string that is not in the language represented by the regular expression $a^*(ab)^*b^*$. Your answer: ba.

5. (5%) Write a context free grammar for the strings that consist of equal number of a's followed by equal number of b's.

Your answer: $S \rightarrow aSb \mid \epsilon$.

6. (10%) Find a regular expression for language over the alphabet $\{0, 1\}$ in which all the strings contain at least two 0's.

Your Answer: $(0|1)^*0(0|1)^*0(0|1)^*$.

7. (10%) Describe as simply as possible in English (less than 15 words for each language) the language corresponding to the regular expression

(a) (5%) $((\epsilon|0)1^*)^*$ Your answer: any string of 0 and 1.

(b) (5%) $a^*b(a^*ba^*b)^*a^*$. Your answer: strings of a and b that have odd number of b's.

Solution: A string in the language can start and end with a or b, it has at least one b, and after the first b all the b's in the string appear in pairs. Any number of a's can appear any place in the string. Thus simply put, it is the set of strings over the alphabet $\{ a, b \}$ that contain an odd number of b's

8. (30%) Given the regular expression $(a|b)a(a|b)^*$.

- (a) (10%) Draw the corresponding NFA diagram using the Thompson construction;
- (b) (10%) Transform the NFA to DFA using subset construction. You need to write the derivation process and draw the resulting diagram;
- (c) (10%) Minimize DFA. You need to write the derivation process and draw the resulting diagram.

a)

b)

$\Sigma(\{0\}) = \{0, 1, 2\} \Rightarrow S$
 $\Sigma(\delta(S, a)) = \Sigma(\{3\}) = \{3, 5\} \Rightarrow A$
 $\Sigma(\delta(S, b)) = \Sigma(\{4\}) = \{4, 5\} \Rightarrow B$
 $\Sigma(\delta(A, a)) = \Sigma(\{6\}) = \{6, 7, 8, 9, 11\} \Rightarrow C$
 $\Sigma(\delta(A, b)) = \{5\}$
 $\Sigma(\delta(B, a)) = \Sigma(\{6\}) \Rightarrow C$
 $\Sigma(\delta(B, b)) = \{5\}$
 $\Sigma(\delta(C, a)) = \Sigma(\{10\}) = \{10, 11, 12, 13, 9\} \Rightarrow D$
 $\Sigma(\delta(C, b)) = \Sigma(\{11\}) = \{11, 12, 13, 7, 8, 9\} \Rightarrow D$
 $\Sigma(\delta(D, a)) = \Sigma(\{10\}) \Rightarrow D$
 $\Sigma(\delta(D, b)) = \Sigma(\{11\}) \Rightarrow E$
 $\Sigma(\delta(E, a)) = \Sigma(\{10\}) \Rightarrow D$
 $\Sigma(\delta(E, b)) = \Sigma(\{11\}) \Rightarrow E$

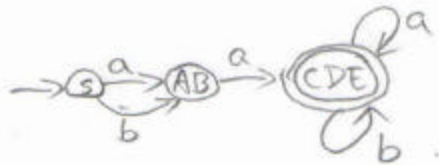
c). (SAB) (CDE)

$(S, a) \rightarrow A$

$(A, a) \rightarrow C$

s: A distinguishable,
the same for S, B.

(SAB) is partitioned into (S)(AB)

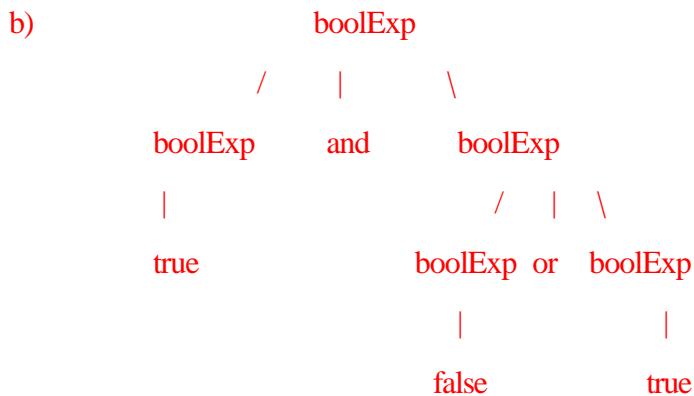


9. (20%) Consider the EBNF definition for Boolean expression:

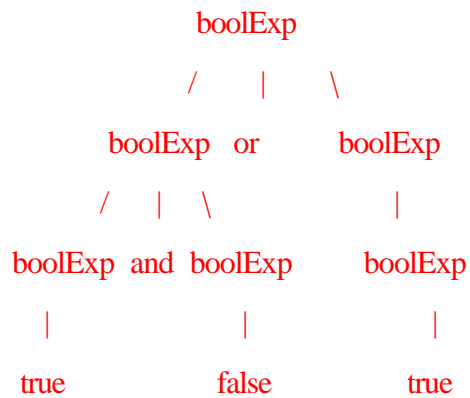
$\text{boolExp} \rightarrow \text{true} \mid \text{false} \mid \text{boolExp} \{ \text{or} \text{ boolExp} \} \mid \text{boolExp} \{ \text{and} \text{ boolExp} \}$

- (a) (5%) Rewrite (i.e., transform) the EBNF definition to context free grammar;
 (b) (5%) Construct a parse tree for the Boolean expression “**true and false or true**”;
 (c) (5%) Prove that the grammar is ambiguous;
 (d) (5%) Transform the grammar into an unambiguous one.

a) $\text{boolExp} \rightarrow \text{true} \mid \text{false} \mid \text{boolExp} \text{ or } \text{boolExp} \mid \text{boolExp} \text{ and } \text{boolExp}$



c) here is another parse tree



d) $\text{boolExp} \rightarrow \text{disjunction} \mid \text{disjunction and boolExp}$
 $\text{disjunction} \rightarrow \text{primitive} \mid \text{primitive or disjunction}$
 $\text{primitive} \rightarrow \text{true} \mid \text{false}$