03-60-214 Computer Languages, grammars, and translators

Midterm exam (A) 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	

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

   Scanning is also called linear analysis_ or __lexical analysis_____. _JLex_____ and _Lex__ are examples of scanner generators. The primary inputs to scanner generators are ___RE____. When you run the scanner, the primary outputs are _Tokens.

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

2. (5%) Nondeterminism of a finite automaton can be viewed as which of the following?

(a) when the machine is in a given state and reads the next input symbol, we know exactly what the next state will be;

(b) a kind of parallel computation wherein several processes can be running concurrently ;

(c) Several choices may exist for the next state at any point;

(d) an NFA, when running on an input string, it has a state with multiple ways to proceed.

Your answer: __b, c, d___.

3. (5%) If k is the number of states of the NFA, it is always possible to create an equivalent DFA with at most how many states?

(a) k^2

(b) 2k

(c) 2^k

(d) k+ 5

Your answer: ___c _____.

4. (5%) True or false: $(01)^* 0 = 0(10)^*$ Your answer is:__true_____.

5. (5%) Giving the following automaton.

(a) (2%) True or false: It is an infinite automaton because it can recognize or reject strings of any length. Your answer: __false_____.

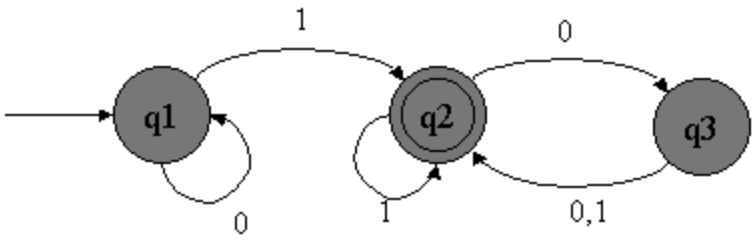
(b) (3%) Which string(s) can be recognized by the automata? Your answer is ___i, iii ___.

i. 0101000000

ii. 0111111000

iii. 0000010000

iv. 1010000000



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%) (0|1)*0(0|1)(1|0) Your answer:_any string over {0,1} with the last third digit is 0.

   (b) (5%) $a^*b(a^*ba^*b)^*a^*$. Your answer:__any string over {a,b} with 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

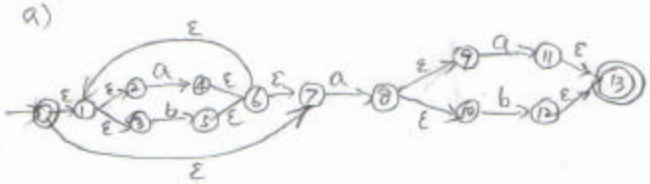
8. (30%) Given the regular expression (a|b)*a(a|b).

   (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) $\varepsilon(0) = \{0,1,2,3,7\} \Rightarrow S$

$\varepsilon(\delta(S,a)) = \varepsilon(\{4,8\}) = \{4,6,7,1,2,3,8,9,10\} \Rightarrow A$

$\varepsilon(\delta(S,b)) = \varepsilon(\{5\}) = \{5,6,7,1,2,3\} \Rightarrow B$

$\varepsilon(\delta(A,a)) = \varepsilon(\{4,8,11\}) = \{4,6,1,2,3,11,13,7,8,9,10\} \Rightarrow C$

$\varepsilon(\delta(A,b)) = \varepsilon(\{5,12\}) = \{5,6,7,1,2,3,12,13\} \Rightarrow D$

$\varepsilon(\delta(B,a)) = \varepsilon(\{4,8\}) \Rightarrow A$

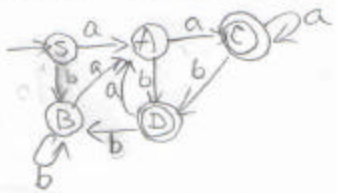
$\varepsilon(\delta(B,b)) = \varepsilon(\{5\}) \Rightarrow B$

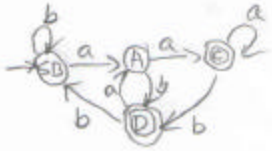
$\varepsilon(\delta(C,a)) = \varepsilon(\{4,8,11\}) \Rightarrow C$

$\varepsilon(\delta(C,b)) = \varepsilon(\{5,12\}) \Rightarrow D$

$\delta(\varepsilon(D,a)) = \varepsilon(\{4,8\}) \Rightarrow A$

$\delta(\varepsilon(D,b)) = \varepsilon(\{5\}) \Rightarrow B$





c) $\pi_1 = (SAB)(CD)$

$(A,a) \to C$, $(S,a) \to A$ } A and S are distinguishable.

$\pi_2 = (SB)(A)(CD)$

$(S,a) \to A$, $(B,a) \to A$, $(S,b) \to B$, $(B,b) \to B$ } S and B are indistinguishable

(CD) are distinguishable.

9. (20%) Consider the EBNF definition for if statement in miniJava:

stmt → ifStmt| other

IfStmt → if "(" Bool ")" stmt [else stmt]

Bool → true|false

(a) (5%) Rewrite (i.e., transform) the EBNF definition to context free grammar;

(b) (5%) Construct a parse tree for the if statement "if (true) if (false) S1 else S2";

(c) (5%) Prove that the grammar is ambiguous;

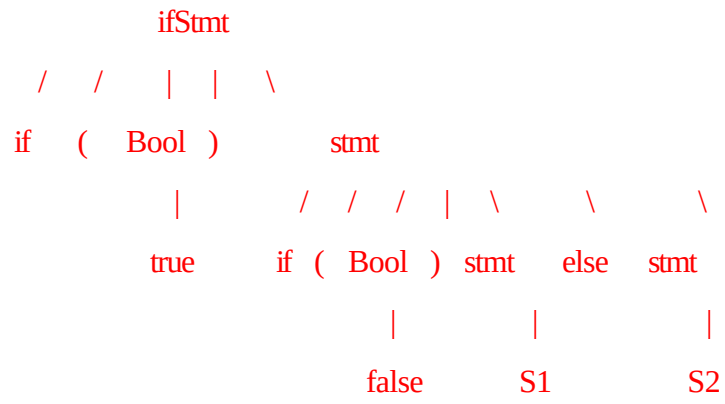
(d) (5%) Transform the grammar into an unambiguous one.

a) stmt→ifStmt|other

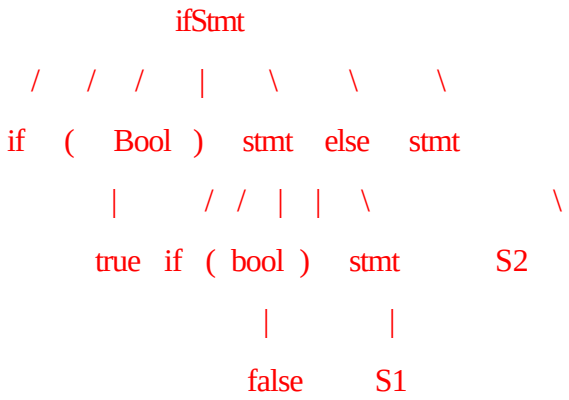
ifStmt→if (Bool) stmt | if (Bool) stmt else stmt

Bool→true|false

b)



c) here is another parse tree:



d) stmt →mStmt |uStmt

mStmt→if (bool) mStmt else mStmt | other

uStmt→ if (bool) stmt | if (bool) mStmt uStmt