KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI

MIDSEMESTER EXAMINATION, 2018/2019

BSc (COMPUTER SCIENCE)

FOURTH YEAR

CSM 484 INTRODUCTION TO COMPILERS **DURATION:1 HOUR**

1. a) Find the phrase structure grammar to generate the set $\{0^m 1^n \mid$ m, n are nonnegative integers. [5 Marks]

   b) Differentiate between lexical analysis and syntactic analysis/parsing. [5 Marks]

   lexer parser

   c) Let V = {S, A, B, a, b} and T = {a, b}. Find the language generated by the grammar

   G = { V, T, S, P} when the set P of productions consists of: [5 Marks]

   i) $S \rightarrow AB$, $A \rightarrow ab$, $B \rightarrow bb$.

   ii) $S \rightarrow AB$, $S \rightarrow aA$, $A \rightarrow a$, $B \rightarrow ba$.

   iii) $S \rightarrow AB$, $S \rightarrow AA$, $A \rightarrow aB$, $A \rightarrow ab$, $B \rightarrow b$.

   iv) $S \rightarrow AA$, $S \rightarrow B$, $A \rightarrow aaA$, $A \rightarrow aa$, $B \rightarrow bB$, $B \rightarrow b$.

   v) $S \rightarrow AB$, $A \rightarrow aAb$, $B \rightarrow bBa$, $A \rightarrow \lambda$

   r*

2. a) Prove that $(r^*)^* = r^*$. [2 Marks]

   b) Show that the following regular expressions are equivalent: [5 Marks]

   i) $a(a+b)^*$

   ii) $a((a+\lambda)(b+\lambda))^*$

   $(aa+ab)^*$

   c) Draw state diagrams for 2(b) (i) and (ii). [6 Marks]

   d) Differentiate between DFA and NFA. [2 Marks]

   unique input from one state another

J. K. PANFORD

COMPUTER SCIENCE IV

CSM 484 INTRODUCTION TO COMPILERS

Time Allowed: 1 Hours

Instruction: Answer all questions

1. Find a deterministic finite-state automaton (state diagram) that recognizes each of the following sets.

   a) $\{0^*\}$ b) $\{1.00^*\}$ c) $\{1^n \mid n = 3,4,5,\ldots\}$

2. Fully parenthesize each of the following expressions and show the partial relationships between the sub-expressions.

   a) $k = \sim b - s \uparrow c * g + f \uparrow q * v$

   b) $w = \sim q \uparrow m - p / k \uparrow c - v * a$

3. Give the step-by-step generation of machine code (in assembly language) of each of the following algebraic/infix strings using 1-address instruction format.

   a) $c = k \div g \uparrow (-(m / q - 2)) + f * v \uparrow w$

   b) $w = \sim(\sim q \uparrow m) + b / c - t * s \uparrow a$

END OF SECOND SEMESTER EXAMINATION 2013-2014

BSc. COMPUTER SCIENCE IV

CSM 484 INTRODUCTION TO COMPILERS

April 2014 **[100 Marks]** **Time Allowed: 2½ Hours**

Instructions: Answer ALL FOUR questions in the answer booklet provided. Each question carries 25marks!

---

1. a) Using a suitable diagram explain the phases of the compilation process.

   b) Describe the types of grammars and give one example under each.

   intermediate

   c) Give 2 advantages of immediate representation (IR) of code.

   d) Let G_1 and G_2 be context-free grammars, generating the languages $L(G_1)$ and $L(G_2)$ respectively. Show that there is a context-free grammar generating each of the following sets.

   i) $L(G_1) \cup L(G_2)$ ii) $L(G_1)L(G_2)$ iii) $L(G_1)^*$

2. a) Let V be an alphabet, and let A and B be subsets of V^* with $A \subseteq B$. Show that $A^* \subseteq B^*$.

   b) Determine whether the string 11101 is in each of the following regular sets.

   i) $\{11\}^*\{01\}^*$ ii) $\{111, 000\}\{00, 01\}$ iii) $\{11\}\{1\}^*\{01\}$

   c) Let V be an alphabet, and let A and B be subsets of V^*, show that:

   $$|AB| \leq |A||B|.$$

   d) A context-free grammar is ambiguous if there is a word in L(G) with two derivations that produce different derivation trees, considered as ordered, rooted trees. Show that the grammar G={V,T, S, P} with V={0,S}, T={0}, starting state S, and Productions P={S→0S, S→S0, and S →0} is ambiguous by constructing two different derivation trees for 0^3.

3. a) Give the step-by-step generation of machine code of the following algebraic/infix strings.

   i) $m = -c \uparrow b * q + a + b/r$
   ii) $p = s * b \uparrow c - w/r * d + e$

   b) Give the step-by-step generation of the algebraic string of each of the following Reverse Polish strings, if possible.

   i) x y g ↑ a b c * e / + f − e / * + = ⇒
   ii) q w m ↑ a / g s * + ⇒

   c) List 2 advantages of a byte-code over an object code.

↑ = 4
× / = 3
~ = 2
+ − = 1
= = 0
(= 10
) = 10

4. a) Fully parenthesize each of the following using the PARENTHESIZING ALGORITHM step-by-step.

   i) $m = -c \uparrow b * q + a + b/r$
   ii) $p = s * b \uparrow c - w/r * d + e$

   b) Draw syntax diagrams for each of the following:

   i) *For* statement
   ii) *If* statement
   iii) *While* statement
   iv) *Try/catch* statement

   c) Explain how a deterministic FSA can be transformed into non-deterministic FSA.

COMPUTER SCIENCE IV

April 2013 **[100 Marks]** **Time Allowed: 2½ Hours**

Instructions: Answer ALL FOUR questions in the answer booklet provided. Each question carries 25 marks.

1. a) Differentiate between each of the following: [10 Marks - 2½ marks each]

   i) Lexer and Parser
   ii) Symbol table and Lexicons
   iii) Deterministic and Nondeterministic Finite State Automata
   iv) Top-down and Bottom-up parsing

   b) i) Construct a phrase-structure grammar that generates all signed decimal numbers, consisting of a sign, either + or - ; a nonnegative integer; and a decimal fraction that is either the empty string or a decimal point followed by a positive integer, where initial zeros in an integer are allowed. [5 Marks]

   ii) Give the Backus-Naur form (BNF) of the grammar in question 1b(i). [5 Marks]

   iii) Construct a derivation tree (syntactic parse tree) for -31.4 in the grammar in 1b(i) using:

   α) top-down parsing [2½ Marks]

   β) bottom-up parsing [2½ Marks]

2. a) Let V be an alphabet, and let A and B be subsets of V* with $A \subseteq B$. Show that $A^* \subseteq B^*$. [5 Marks]

   b) Determine whether the string 11101 is in each of the following (regular) sets. [6 Marks – 2 marks each]

   i) $\{0, 1\}^*$ ii) $\{1\}^*\{0\}^*\{1\}^*$ iii) $\{111\}^*\{0\}^*\{1\}$

   c) Find a deterministic finite state automaton (state diagram) that recognizes each of the following sets. [6 Marks – 2 marks each]

   i) $\{0\}$ ii) $\{1, 00\}$ iii) $\{1^n \mid n = 2, 3, 4, \ldots\}$

   d) A context-free grammar is ambiguous if there is a word in L(G) with two derivations that produce different derivation trees, considered as ordered, rooted trees. Show that the grammar G = (V, T, S, P) with V = {0, S}, T = {0}, starting state S, and production $P = \{S \to 0S, S \to S0,$ and $S \to 0\}$ is ambiguous by constructing two different derivation trees

[8 Marks]

3. a) Using the parenthesizing algorithm, fully parenthesize each of the following; and show the spatial relationship between the expression and the subexpressions:

i) $w = -b \uparrow c + d - f/(-m \times k) \uparrow -c$ [6 Marks]

ii) $-h + w/s - k \uparrow c - t \times d \uparrow m$ [5 Marks]

b) Using 1-address instruction format convert each of the following algebraic strings into machine (code in assembly language).

i) $w = -s \uparrow g / c \times b - (-d \uparrow -m) + 2$ [7 Marks]

ii) $d - k/h + (-t \uparrow b) \uparrow (d/e \times f) - k$ [7 Marks]

4. a) Convert each of the following infix strings into Reverse Polish strings: [10 Marks – 5 marks each

i) $p = -g \uparrow c + r - f/(-m \times k) \uparrow -c$

ii) $w - k/h + (-t \uparrow r) \uparrow (d/e \times s) - k$

b) Draw a syntax diagram, in C/C++ or Java, for each of the following:

i) statement [4 Marks]

ii) for statement [2 Marks]

iii) while statement [2 Marks]

iv) try/catch statement [2 Marks]

c) Suppose that A and B are finite subsets of V*, where V is an alphabet. Is it necessary true that $|AB| = |BA|$? Explain. [5 Marks]

$A = \{a\}$ $B = \{b\}$

$AB = \{ab\}$

$BA = \{ba\}$

END OF SECOND SEMESTER EXAMINATION, 2011

CSM 484 Introduction to Compilers

April, 2011 [100 Marks] Time Allowed: 3 Hou

Instruction: Answer __ALL__ questions in the answer booklet provided. All questions carry equal mar

1. a) i. Differentiate between TOP- DOWN and BOTTOM-UP parsing. [1 Marks]

   ii. List the phases of the compilation process, and briefly explain the phases. [4 Marks]

   b) Give the BOTTOM-UP parsing, if possible, of the following strings using the syntactic de in APPENDIX A. [20 Marks – 5 Marks each]

   i. QZ = W + (LB – KT) / DIX + SAM

   ii. 45 IF U / S – DN + RAT .GT. 47.5 * Q THEN 80

   iii. AZ ↑ (A + CD / BAT) + ELS * SLE

   iv. ABC + EF * (CAT + KIJ) / 50.75

2. a) Find a deterministic finite-state automaton (state diagram) that recognizes each of the follo [9 Marks – 3 Marks each

   α) {0} β) {1,00} γ) $\{1^n \mid n = 2,3,4,...\}$

   b) Draw a detailed syntax diagram in Java for each of the following: [12 Marks – 3 Mar

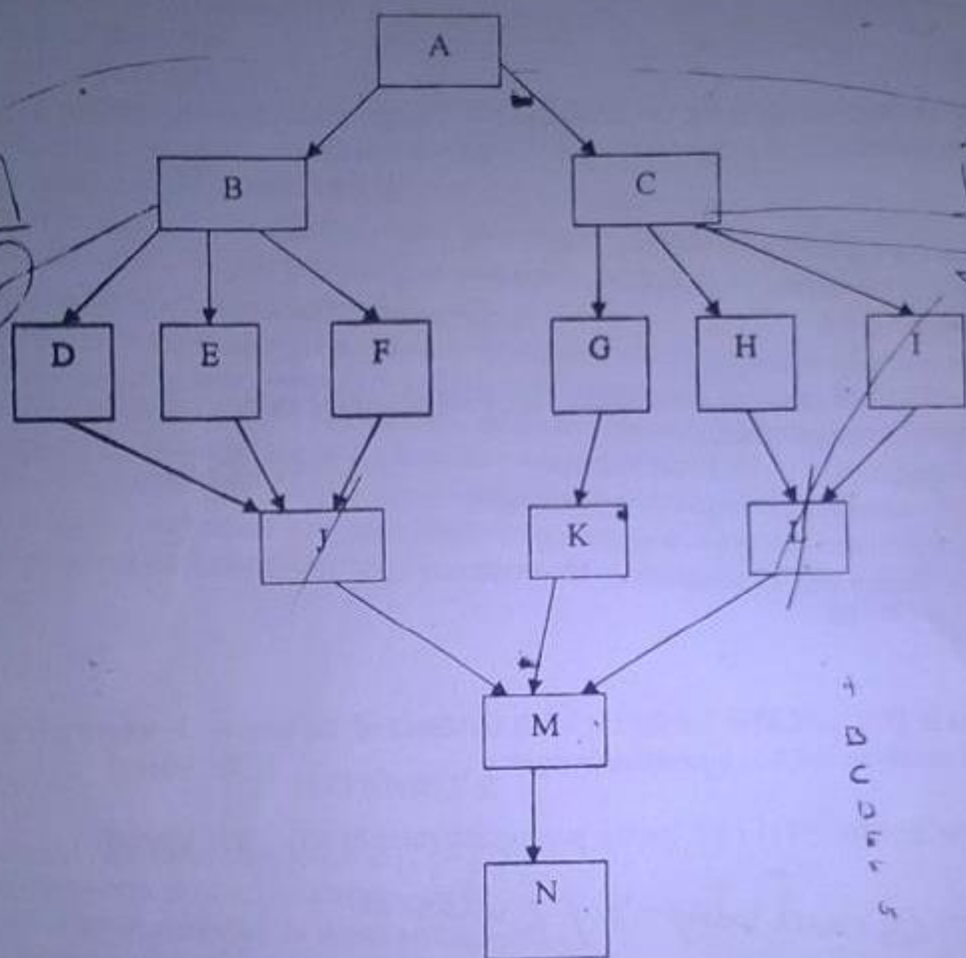
   i) *try/catch* statement

   ii) *if* statement

   iii) *for* statement

   iv) *method* declaration

c) Optimize the following blocks of code using the *dominator tree* algorithm. A, B, C, . . . , N are statement blocks. [4 Marks]



3. a) Determine whether the string 11101 is in each of the following sets. [5 Marks]

i) $\{0, 1\}^*$ ii) $\{1\}^*\{0\}^*\{1\}^*$ iii) $\{11\}^*\{01\}^*$

iv) $\{11\}^*\{1\}^*\{01\}$ v) $\{111\}^+\{0\}^*\{1\}$

b) Give the step-by-step generation of REVERSE POLISH STRING of each of the following algebraic strings: [10 Marks – 5 Marks each]

i. $q = a + b / c + d * e - g \uparrow r$
ii. $w = - z \uparrow b * c + d - r - g$

c) Give the step-by-step generation of the algebraic string of each of the following Reverse Polish st if possible:

[10 Marks – 5 Marks each]

i. x y g + a b c * e / + f – e / * + =

ii. e f c ↑ a + d r / + *

4. a) Fully parenthesize the following strings using the PARENTHESIZING ALGORITHM step-by-step, and show the spatial relationship between the expression and the sub-expressions:

[10 Marks – 5 Marks each]

i. n = - q ↑ r * s + a + b / g

ii. p = s + b * c + w / r * d + e

b) Give the step-by-step generation of machine code of the following algebraic strings using the rules in APPENDIX B and an appropriate algorithm:

[10 Marks – 5 Marks each]

i. w = a * b * c + z / y ↑ t

ii. d = -q ↑ v + r * s + a / b - m

c) i) Give the production rules in BNF or EBNF for the set of all fractions of the form a / b, where a is a signed integer in decimal notation and b is a positive integer. [2½ Marks]

ii) Construct a syntactic parse tree for +311 / 17 for the production rules in c(i). [2½ Marks]

<fraction> ::= <signed Integer> / <integer>

<signed Integer> ::= <sign><Integer>

<sign> ::= + | – | ⊔

<Integer> ::= <digit> | <digit><Integer>

<digit> ::= 0 | 1 | ... | 9

+311 / 17

APPENDIX A

<null> ::= b (b ≡ blank or space)
<sign>::= + | - | <null>
<prefix operator>::= <sign>
<alphabetic character>::= A | B |C| ...|Z
<zero digit>::= 0
<non zero digit>::= 1 | 2 | 3 | ...| 9
<digit>::= <zero digit> | <non zero digit>
<replacement sign>::= =
<relational operator>::= .EQ. | .GT. | .LE. | .GE. | .NE. | .LT.
<line number>::= {<digit>}^{2} <non zero digit> | {<non zero digit>}^{3} | <non zero digit> {<digit>}^{2}
<constant>::= <sign> {<digit>}| <constant> .{<digit>} | <constant> E {- | +}^{1}{<digit>}$^{2}_{1}$
<variable>::= {<alphabetic character>}^{4}
<term>::= <constant> | <variable> | (<expression>) | {<prefix operator>}^{1}<term>
<involution factor>::= <term> | <term> ↑ <term>
<multiplying factor>::= <involution factor> | <multiplying factor> { * | / }^{1} <involution factor>
<expression>::= <multiplying factor> | <expression> { + | - }^{1} <multiplying factor>
<if statement>::= IF <expression><relational operator><expression> THEN <line number>
<arithmetic statement>::= <variable><replacement sign><expression>
<referenced statement>::= <line number><if statement> | <line number><arithmetic statement>

APPENDIX B

<operator in input string>::= + | - | * | / | ↑
<unary operator in input string>::= ~ | -
<replacement operator in input string>::= =
<temporary storage location>::= {[α1 | α2 | α3 | α4 | α5 | α6 | α7 | α8 | α9 | α10]}^{1}
<operator in generated instruction>::= ADD | SUB | MUL | DIV | EXP
<operand::= a | b | c | ...| z
<operand1>::= <operand>
<operand2>::= <operand>

PRODUCTION RULES (ALGEBRAIC STRING TO MACHINE CODE)

a) <operand1><operator in input string><operand2>
→ LDA <operand1>
<operator in generated instruction><operand2>
STOR <temporary storage location>

DEPARTMENT OF COMPUTER SCIENCE

MIDSEMESTER EXAMINATION 2014

CSM 484 INTRODUCTION TO COMPILERS Time Allowed: 45 Minutes

1. a) Find a deterministic FSA that recognizes each of the following sets.

ii) $\{0\}$ ii) $\{1, 00\}$ iii) $\{1^n \mid n = 2, 3, 4, \ldots\}$

b) Let V be an alphabet, and let A and B be subset of V^* with $A \subseteq B$. show that $A^* \subseteq B^*$.

c) Show that the grammar $G = \{V, T, S, P\}$ with $V = \{0, S\}$, $T = \{0\}$, starting state S, and productions $S \rightarrow 0S$, and $S \rightarrow 0$ is ambiguous by constructing two different derivation or parse trees for 0^3.

2. a) Fully parenthesize each of the following using the parenthesizing algorithm:

i ~~iii)~~ $k = -b \uparrow m + q * s - w / f$

ii ~~iii)~~ $t - w \uparrow s - d / q \uparrow - g * b$

b) Convert 2(a) (i) and (ii) into Reverse Polish notation.

c) Write a method in C++ or Java that accepts an input algebraic string and generate a syntactic parse tree.