VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)

IBRAHIMBAGH, HYDERABAD-31

Department of Computer Science and Engineering

BE-VII SEMESTER II-Internal Examination

Name of the COURSE: COMPILER CONSTRUCTION

Subject Reference Code: UI21PC630CS

Open Book Test

Max Marks: 30 Date: 16/05/2024

Time: 2:30PM to 4:30PM

Note: Answer any four questions
(4 X 7.5 = 30M)

Q. No.	Question	Marks	BTL (1/2 /3/4 /5/6)	Mapped	
				CO	PO
1	Define Synthasized attributes and inheritted attributes $S \to L \,.\, L \mid L$ $L \to L\ B \mid B$ $B \to 0 \mid 1$ Design an L-attributed SDD to compute S.val, the decimal-number value of an input string. For example, the translation of string 101.101 should be the decimal number 5.625. Hint: use an inherited attribute L. Side that tells which side of the decimal point a bit is on.	7.5	3	3	1,2,3
2	Suppose that we have a production A →BCD. Each of the four non-terminals A, B, C, and D have two attributes: s is a synthesized attribute, and i is an inherited attribute. For each of the sets of rules below, tell whether. (i) the rules are consistent with an S-attributed definition (ii) the rules are consistent with an L-attributed definition, and (iii) whether the rules are consistent with any evaluation order at all? (iv) Draw dependency graph for each set of rules a) A.s = B.i + C.s. b) A.s = B.i + C.s and D.i = A.i + B.s. c) A.s = B.s + D.s. d) A.s = D.i, B.i = A.s + C.s, C.i = B.s, and D.i = B.i + C.i.	7.5	3	3	1,2,3

3	a) In this task we are given a Java-like language where methods can have locally defined methods. Furthermore, it is possible to declare variables and methods at the other most program level. That is supposed to work as usual in languages with static scoping. The following is a program in this language.
	`{`
	`    class C {`
	`        void m1 () {`
	`            void f() {};`
	
	`            f ();`
	`        }`
	`        void m2 () {`
	`            int i;`
	`            void g() {`
	`                int j;`
	`                j = i;`
	`            };`
	`            i = 1;`
	`            rC.m1();`
	`        };`
	`    };`
	
	`    C rc ;`
	`    void main () {`
	`        rc =  new C{} ; rC.m2{};`
	`    }`
	`}`
	Draw the call stack in the situation where the activation record for **f** is on top of the stack for the first time. Draw the stack including variables, access-links, and control-links, but without access-links for methods which are directly declared in a class.
		4.5	3	4	1,2,3

	b) Write a small program that will produce different values depending on which kind of variable scoping mechanism is used, static or dynamic. Explain your answer.	3	3	4	1,2,3

4	a) Consider the following object-oriented program in Java style:
	`class A { int a,b; };`
	`class B extends A { int c,d1,d2,d3,d4,d5,d6,d7,d8,d9; };`
	`...`
	`static void f(A x) { x.a = 1; }`
	`static void g(B x) { x.c = 2; }`
	`static void h() { A p = new A(); f(p); g(p); }`
	`static void k() { B p = new B(); f(p); g(p); }`
	`static void main() { h(); k(); }`
	Explain the run-time structure of values of type A and B. Indicate a constraint on the layout of these structures needed to support inheritance.
		4.5	3	4	1,2,3

	b) Explain how it is possible to “leak memory” using a reference counting garbage collector and describe any technique that might be used to address this problem.	3	3	4	1,2,3

5	Define basci block and control flow graph.Discuss the algorithm to construct basic blocks from the given three address code
	`for (i=0; i<n; i++)`
	`    for (j=0; j<n; j++)`
	`        c[i][j] = 0.0;`
	`for (i=0; i<n; i++)`
	`    for (j=0; j<n; j++)`
	`        for (k=0; k<n; k++)`
	`            c[i][j] = c[i][j] + a[i][k]*b[k][j];`
	Construct the control flow graph for the above code
		7.5	3	5	1,2,3

6	`void quicksort(int m, int n)`
	`    /* recursively sorts a[m] through a[n] */`
	`{`
	`    int i, j;`
	`    int v, x;`
	`    if (n <= m) return;`
	`    /* fragment begins here */`
	`    i = m-1; j = n; v = a[n];`
	`    while (1) {`
	`        do i = i+1; while (a[i] < v);`
	`        do j = j-1; while (a[j] > v);`
	`        if (i >= j) break;`
	`        x = a[i]; a[i] = a[j]; a[j] = x; /* swap a[i], a[j] */`
	`    }`
	`    x = a[i]; a[i] = a[n]; a[n] = x; /* swap a[i], a[n] */`
	`    /* fragment ends here */`
	`    quicksort(m,j); quicksort(i+1,n);`
	`}`
	Explain machine independent optimization and convert the above code into three address code and optimize the generated three address code.
		7.5	3	5	1,2,3