compiler Design Assignment

CS21B2011

G. Hemanth.

(1) which data structure in a compiler is used for managing information about variables and their attributes?

Ans:- In a compiler, the data structure is responsible for managing info about variables and their attributes is a symbol table.

(2) The output of a lexical analyzer is

(a) A parse tree (b) Intermediate code.

(c) Machine code (d) stream of tokens.

Ans:- (d) stream of tokens.

(3) No. of tokens in a FORTRAN statement DO 10 I = 1.25 is:-

(a) 3 (b) 4 (c) 5 (d) None of these.

Ans:- There are 3 tokens because it's 1.25 and not 1,25.

→Answer would be 5 if it would have comma 1,25.

∴ (A) 3.

(4) which of the following machine model can be used in a necessary and sufficient sense for lexical analysis?

(1) Deterministic PDA (2) Finite Automata

(3) Non Deterministic FA (4) Turing machine

Ans:- Finite Automata.

CS21B2011
G. Hemanth

⑤ In a compiler, key words are recognised during?

Ans:- Keywords of a language are recognized during lexical analysis phase.

⑥ The number of Tokens in the C statement:-

```
printf("i = %d, &i = %x", i, &i);
```

(a) 3 (b) 26 (c) 10 (d) 21

Ans:- (c) 10

∵ There are 10 tokens in this statement.

⑦ Match elements from Group-1 to Group-2

Group-1	Group-2
P. Regular Expression	1. Syntax Analysis
Q. Push Down Automata	2. Code generation
R. Dataflow Analysis	3. Lexical analysis
S. Register Allocation	4. Code optimisation.

Ans:- P-3 :- RE is used in Lexical analysis.
Q-1 :- PDA is used for Syntax Analysis.
R-4 :- performed during code optimisation.
S-2 :- part of code generation.

⑧ Type checking is normally done during?

Ans:- Type checking is normally done during the Syntax Directed Translation (or) Semantic Analysis.

CS21B2011
G.Hemanth

⑨ Arrange given compilation process in correct order:-
(i) Linking (ii) Assembly (iii) Compiling (iv) pre-processing

(a) iv → iii → ii → i (b) i → iii → ii → iv
(c) iv → iii → i → ii (d) i → ii → iii → iv

Ans:- (a) iv → iii → ii → i

⑩ In some languages, an identifier is permitted to be a letter followed by any number of letters and digits. If L and D denote the sets of letters and digits, How do you define identifier?

Ans:- L: {a-z, A-Z} D: {0-9}

Then Identifier ⇒ L(L+D)*

⑪ A lexical analyzer uses following patterns to recognize three tokens, T_1, T_2, T_3 over {a, b, c}. T_1: a?(b|c)* a T_2: b?(a|c)* b T_3: c?(b|a)*c. Note that "x"? means 0,1 occurence of symbol x. if string bbaacabc is processed by analyzer which is the tokens it outputs?

(a) $T_1 T_2 T_3$ (b) $T_1 T_1 T_3$
(c) $T_2 T_1 T_3$ (d) $T_3 T_3$

Ans:- (d) $T_3 T_3$. Given string : bbaacabc. Longest matching prefix is 'bbaac' ⟶ T_3 Remaining part 'abc' can be made by T_3.

CS21B2011
G. Hemanth

(12) Consider the following C-code segment,

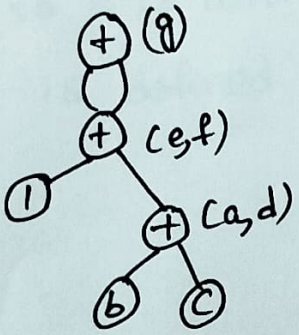
a=b+c; e=a+1; d=b+c; f=d+1; g=e+f;

For this, the number of nodes in DAG is?

Ans:- Here a and b are same as both have b+c. So f and e are also same

∴ Total nodes = 6 (a, b, c, e, 1, g)

The DAG is,



(13) A top-down parser generats:-

(a) Rightmost derivation (b) Rightmost dervatin in Reverse.

(c) Leftmost derivation (d) Leftmost dervation in Reverse.

Ans:- (c) Top-down parser is LL parser which means left to right parsing; leftmost derivation. Bottom up parser is LR parser with Left to Right Parsing performing Rightmost derivation in reverse order.

(14) Consider the grammer:

S → aABld

B → b

No. of reduction steps taken by bottom up parser while parsing aaadbb is

(a) 5 (b) 6 (c) 7 (d) 8

Ans:- (c) 7,

CS21B2011
G. Hemanth

(15) which of the following statements is False?

(a) CFG can be used to specify both lexical & syntax rules.

(b) Type checking is done before parsing.

(c) both a and b.

(d) None of the above.

Ans:- (b) is false.

→ Type checking is done on parsing tree generated by Parser, so b is wrong.

(16) A grammer is defined as:-

$A \to BC$
$B \to x/Bx$
$C \to B/D$
$D \to y/\epsilon y$
$E \to z$

Non terminal alphabet of the grammer is:-

(a) {A, B, C, D, E} (b) {B, C, D, E} (c) {A, B, C, D, E, x, y, z}

(d) {x, y, z}.

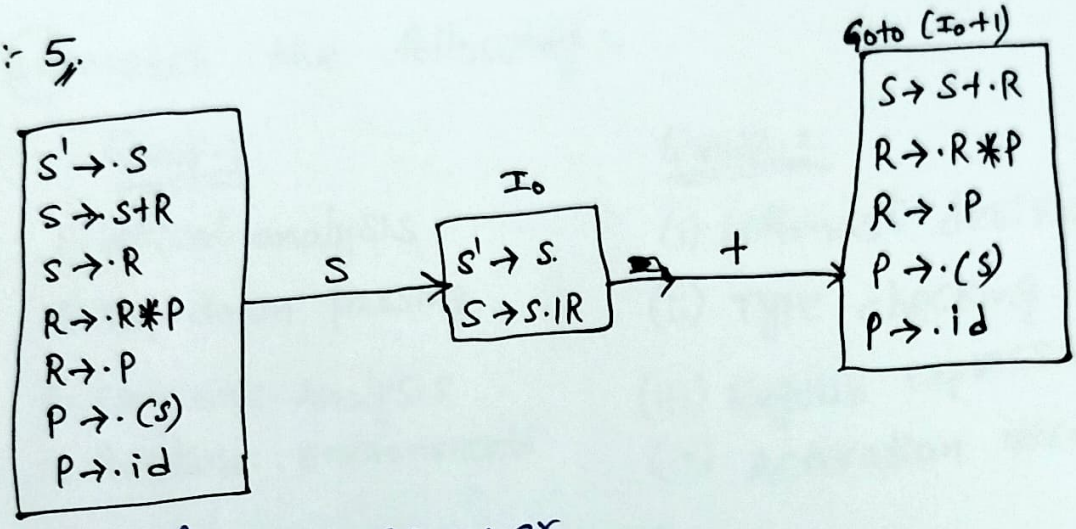
Ans:- (a) {A, B, C, D, E}. This set contains all the symbols from which strings of long can be derived by applying production rules. These are present on L.H.S of the production rules.

(17) consider the augmented grammer with {+, *, (,), id} as terminals and if,

$S' \to S$ $S \to S+R/R$ $R \to R*P/P$ $P \to (S)/id$

If I_0 is the set of two LR(0) items {[$S' \to .S$], [$S \to .S+R$]} then goto (closure(I_0, +)) contains exactly ____ Items.

Ans:- 5.



(18) consider following grammer,

$S \to aSB/d \quad B \to b$

The number of reduction steps taken by a bottom-up parser while accepting the string aaadbbb is ____

Ans:- Given aaadbbb

(i) $S \to aSB$ (ii) $aSB \to aaSBB$

(iii) $aaSBB \to aaaSBBB$ (iv) $aaaSBBB \to aaadBBB$

(v) $aaadbbB \to aaadbBB$ (vi) $aaadbBB \to aaadbB$

(vii) $aaadbbB \to aaadbbb$

$\therefore$ 7 steps.

(19) consider the grammer given here,

$S \to Aa \quad A \to BD \quad B \to b/\epsilon \quad D \to d/\epsilon$

Let a, b, d and $ be indexed as,

a:3, b:2, d:1, $:0

compute follow set of B and write the index values of symbols in follow set in descending order.

Ans:- Initially, Follow(B) = First(D) = {d, ε} − {ε} = {d}

put ε in Prod-2, Follow(B) = Follow(B) ∪ Follow(A) = {d, a}

$\therefore$ The number = 31 (d, a)

CS21B2011
G. Hemanth

(20) Match the following :-

Group-1	Group-2
P. lexical analysis	(i) Leftmost derivation
Q. Top down parsing	(ii) Type checking.
R. Semantic Analysis	(iii) Regular Expressions
S. Runtime environments	(iv) Activation records.

Ans:- P-iii, Q-i, R-ii, S-iv

(21) Among simple LR (SLR), canonical LR, and look-ahead LR (LALR), which of the following pairs identify the method that is very easy to implement and the method that is most powerful.

(a) SLR, LALR (b) canonical LR, LALR

(c) SLR, canonical LR (d) LALR, canonical LR.

Ans: power : LR(0) < SLR(1) < LALR < CLR(1)

complexity :-

LR(0) < SLR(1) < LALR(1) < CLR(1)

∴ SLR, Canonical LR ⇒ option C.

(22) Match the following :-

Group-1	Group-2
P. Lexical analysis	1. Graph coloring.
Q. Parsing	2. DFA minimization.
R. Register Allocation	3. post-order traversal.
S. Expression Evaluation	4. production tree.

Ans:- P-2; Q-4, R-1; S-3.

(23) which of the following is true about shift-Reduce parsing?

(a) variable prefixes appear only at the buttom of the stack and not inside

(b) viable prefixes appear only at the top of the stack and not inside.

(c) the stack contains only a set of viable prefixes.

(d) the stack never contains vaible prefixes

Ans:- (c)

→ the handle is always on top of stack and a viable prefix will never extend past R.H.S of handle (i.e TOS)

(24) consider grammer defined by

$S \to T*P \quad T \to U \mid T*U \quad P \to Q+P \mid Q \quad Q \to id \quad U \to id$

which is true?

(a) + is left, * is right associative.

(b) + is right, * is left associative.

(c) Both are right associative.

(d) Both are left associative.

Ans:- In $T \to U \mid T*U$, $T \to T*U$ is left recursive.

$P \to Q+P$, $P \to Q+P$ is right recursive.

$\therefore$ * is left, + is right

option - (b) //

CS21B2011
G.Hemanth

(25) consider the following grammer along with translation rules.

$S \rightarrow S_1 \# T \{S.Val = S_1.Val * T.Val\}$

$S \rightarrow T \{S.Val = T.Val\}$

$T \rightarrow T_1 \% R \{T.Val = T_1.Val \div R.Val\}$

$T \rightarrow R \{T.Val = R.Val\}$

$R \rightarrow id \{R.Val = id.Val\}$.

Here # and % are operators and id is a token that represent a token that represent an integer and id.Val represents corresponding integer value. Now, find S.Val for expression 20#10%5#8%2%2

Ans:- $20\#10\%5\#8\%2\%2 = 20*(10\div5)*(8\div2)\div2)$

$= 80$ //

(26) consider the following code segment

$x = u - t;\ y = x * v;\ x = y + w;\ y = t - z;\ y = x * y;$

The minimum no. of total variables required to convert the above code segment to static assignment form.

Ans:- In static Assignment

$T_1 = u - t$

$T_2 = T_1 * v$

$T_3 = T_2 + w$

$T_4 = t - z$

$T_5 = T_3 * T_4$

$\therefore$ Total $= T_1 - T_5 + (u, t, v, w, z)$

$= 10$

(27) The purpose of using intermediate cod in compilers is to ——?

Ans:- It increases the chance of reusing the machine-independent code optimizer in other compilers.

CS21B2011
G. Hemanth

(28) In the grammer, what is associativity of + and *

$X \to X+Y \mid Y \quad Y \to Z*Y \mid Z \quad Z \to id$

Ans:- $X \to X+Y$: left Recursive → +: left Associative.

$Y \to Z*Y$: Right Recursive → *: right Associative.

(29) A linker is given object modules for a set of programs that were compiled seperately. what information needed to be included in an object module?

Ans:- Names and locations of all external symbols defined in the object module.

(30) Generation of intermediate code based on an abstract machine model is useful in compilers because?

Ans:- It enhances the portability of front-end of compiler.

(31) consider the following SDTS,

$S \to aA$ {print 1} $\quad A \to Sb$ {print 3}

$S \to a$ {print 2}

using SDTS, find output printed by bottom-up parser for input aab

Ans:- $S \to aA$ print 1

$\to asb$ print 3

$\to aab$ print 2

$\therefore$ Answer = 231 $\quad \therefore$ 132 //

(32) consider the block given below;

$a = b+c \quad e = d-b$

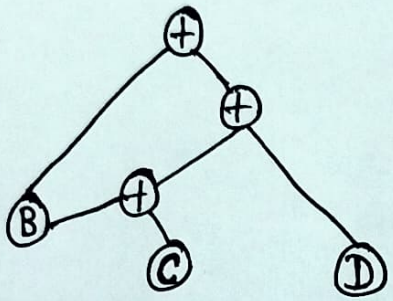
$c = a+d \quad a = e+b$

$d = b+c$

CS21B2011
G. Hemanth

The nodes and edges in min DAG of above block.

Ans:-



∴ 6 Nodes & 6 Edges.

(33) which languages necessarily need heap in the runtime?

Ans:- Those that allow dynamic data structures.

(34) Some code optimizations are carried out on the intermediate code because?

Ans:- they enhance the portability of the compiler to other target processors.

(35) which pase of the compiler is responsible for memory allocation & deallocation for variables.

(a) hexical Analysis (b) syntax Analysis

(c) semantic Analysis (d) code generation.

Ans:- Semantic Analysis.

(36) Incompatable types work with the

(a) syntax tree (b) semantic Analyser

(c) code optimiser (d) Lexical Analyser.

Ans:- (b) semantic Analyser.

CS21B2011
G. Hemanth

(37) which is NOT intermediate representation of source program?

(a) Three Address code (b) AST

(c) CFG (d) symbol Table.

Ans:- (d) symbol.

(38) substitution of values for names is done in:-

(A) Local optimization (B) Loop optimization.

(C) constant Folding (d) strength reduction.

Ans:- (C) → constant Folding.

(39) which class of statement usually produces no executable code when compiled?

(A) Declaration (B) Assignment statement

(c) I/o statements (D) structural statements.

Ans:- (A) → Declaration.

(40) consider two modules M_1 & M_2. If M_1 contains reference to a function defined in M_2 the reference will be resolved at:-

(A) Edit time (B) Compile time

(c) Link time (D) Load time

Ans:- (C) → link time.

→ Each module is compiled seperately and linked together to make the executable. We can do this using gcc.//