# CS606 Final Term Preparation:

## Solved and verified by Masters

### Subscribe our YouTube channel for more solved papers

Paper#1

Cs606 Current Paper Final Term Spring 2022

Almost Mcqs are from junaid file 35.

1. The shift action_____ a terminal on the stack.

**Pushes (page 73)**

2. The regular expressions a*|b* and (a|b)* describe the_____set of strings.

**Different**

3. Yacc contains built-in support for handling ambiguous grammars resulting in_____ conflicts.

**Shift-reduce**

Shift action

Yacc contains built-in support for handling ambiguous grammars resulting in shift-reduce conflicts. By default these conflicts are solved by performing the _____.

4. For convenience, lexical analyzers should read the complete ___ program into memory.

**Input**

5. The code generator uses_____ for keeping track of register.

**Two data structures**

register. The code generator uses two data structures for keeping track of register usage:

6. Address descriptor is the location where the_____ for a variable can found.

**Current value**

register. The code generator uses two data structures for keeping track of register usage:

1. Register descriptor - register status (empty, inuse) and contents (one or more "values")
2. Address descriptor - the location (or locations) where the current value for a variable can be found (register, stack, memory)

7. When constructing an LR(1) parser we record for each item exactly in which context it appears, which resolves many conflicts present in_ parsers based on FOLLOW sets.

**SRL(1)**

8. The order in which the DAG is traversed can lead to_____ code.

**Better**

The order in which the DAG is traversed can lead to better code.

9. . Three-address codes are often implemented as a_____

**Set of quadruples**

Three-address codes are often implemented as a *set of quadruples*.

10. Start symbol start with_____

**S**

11. ___is the process of discovering a derivation for some sentence of a language.

**Parsing**

12. LALR(1) leads to tables that have _____fewer states than LR(1)

**10 times**

Masters

These are called the *LALR(1) states*. LALR(1) stands for LookAhead LR(1). This leads to tables that have 10 times fewer states than LR(1).



13. Token are stored in_____.

**Stack**

which usually belongs to a later phase of compilation. A symbol table maps symbols (that is, names) to values. The quotation in your question is about tokens, not symbols. Tokens are not usually stored in a symbol table, and there is no reason to store numbers in a symbol table. Typically a parser will have a stack, and will push whole tokens onto the stack, and pop them off again as needed.

14. Values used to compute____ attributes flow bottom-up in the parse tree.

**Synthesized**

and from constants are called *synthesized attributes*. Values used to compute synthesized attributes flow bottom-up in the parse tree.

15. Back patching is especially suited for_____.

**bottom-up parsers.**

Backpatching is especially suited for bottom-up parsers.

## Subjective

1. Top down parser may pick the wrong production and In that case it will need to backtrack, can every grammar back track inn such situation.

(Lecture12)

2. Which information we can get from the second section of the YACC file? Most important

https://www.cs.utexas.edu/users/novak/yaccpaper.htm#:~:text=The%20next%20several%20sections%20describe,the%20preparation%20of%20lexical%20analyzers.

The next several sections describe the basic process of preparing a Yacc specification; Section 1 describes the preparation of grammar rules, Section 2 **the preparation of the user supplied actions associated with these rules**, and Section 3 the preparation of lexical analyzers.

Q: How inherited attributes can be used in syntax directed definition?

**Implementation of Syntax-directed Translation**

The easiest way to implement syntax-directed definitions is to use two passes: construct a syntax tree for the input in the first pass and then walk the tree in depth-first order evaluating attributes and emitting code. We would like to use only one pass if possible. The problem in generating three-address code in one pass is that we may not know the labels that the control must go to when we generate jump statements. However, by using a technique called *back-patching*, we can generate code in one pass.

Q: **How the following elementary programming construct play an important role of changing the flow of control in program Label GOTO**

code in one pass is that we may not know the labels that the control must go to when we generate jump statements. However, by using a technique called *back-patching*, we can generate code in one pass.

As we generate code, we will generate the jumps (conditional or unconditional) with targets temporarily left unspecified. Each such statement will be put on a list of goto statements that have targets missing. We will fill the labels when the proper label can be determined; this is the backpatching step. Backpatching is especially suited for bottom-up parsers.

**Q: What can be tracked in address descriptor?**

Address descriptor keep the track of register usage.

CS606 paper #2

Q: Translate the regular expression “r*” in simple words. (2 Marks)

means match any character in regular expressions. * means **zero or more occurrences of the SINGLE regex preceding it**. 27-Jan-2017

d ( $r^*$) are regular expressions corresponding to languages $L_r \cup L_s$ , $L_r L_s$ and $L_r{}^*$ .

1. What are two properties of reduced flow graphs?(rep) (2 Marks)

**Reducible flow graph**

**Intuition**: Produced by exclusive use of flow of control statements such as if-then-else, while-do, continue, break etc.

**Statistics:** Even programs written using goto statements are almost always reducible.

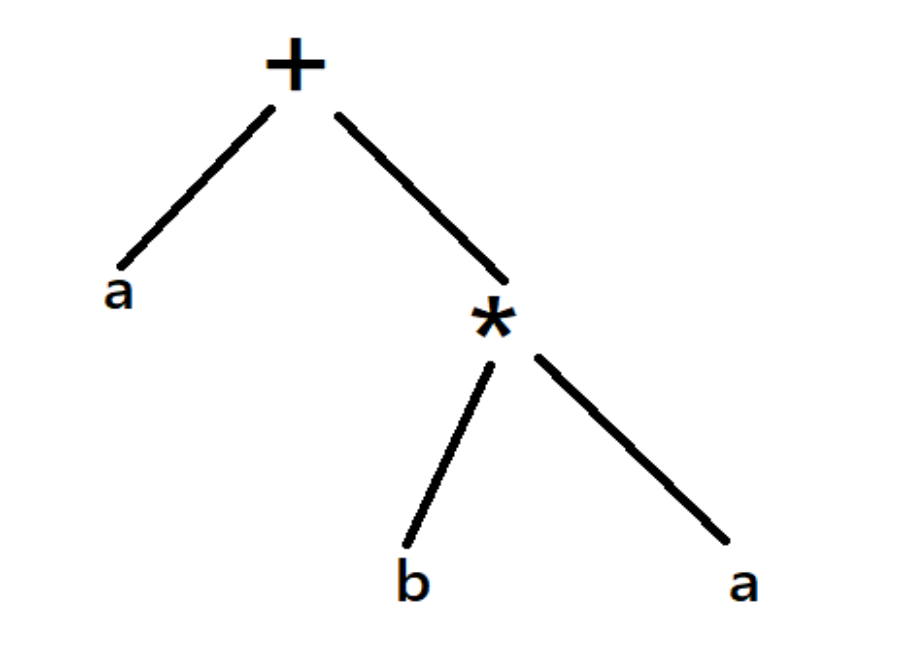
**Main property:** there are no jumps in the middle of loops from outside; the only entry to a loop is through its header.

**Definition:** A flow graph G is reducible if and only if we can partition the edges into two disjoint groups, often called the forward edges and back edges with properties:- forward edges from an acyclic graph in which every node can be reached from the initial node of G.- the back edges consist only of edges whose heads dominate their tails.

2. Which information we can get from the second section of YACC file? (2 Marks) (rep)
3. **In main memory, what type of information is stored in data segment? (2 Marks)**

In memory the code generated by the intermediate Representation is stored. It include the source file, which contain the tokenized symbol, variable, objects and other parse data. This code is very near to the machine language.

## Q: Consider the following a + (b*a) you are required to translate it into an AST. (3 Marks)




The syntax is "abstract" in not representing every detail appearing in the real syntax. For instance, grouping parentheses are implicit in the tree structure

**Q: How do we construct an LR parsing table for a grammar**? (3 Marks)

## LR Parser

← Prev Next –

LR parsing is one type of bottom up parsing. It is used to parse the large class of grammars.

In the LR parsing, "L" stands for left-to-right scanning of the input.

"R" stands for constructing a right most derivation in reverse.

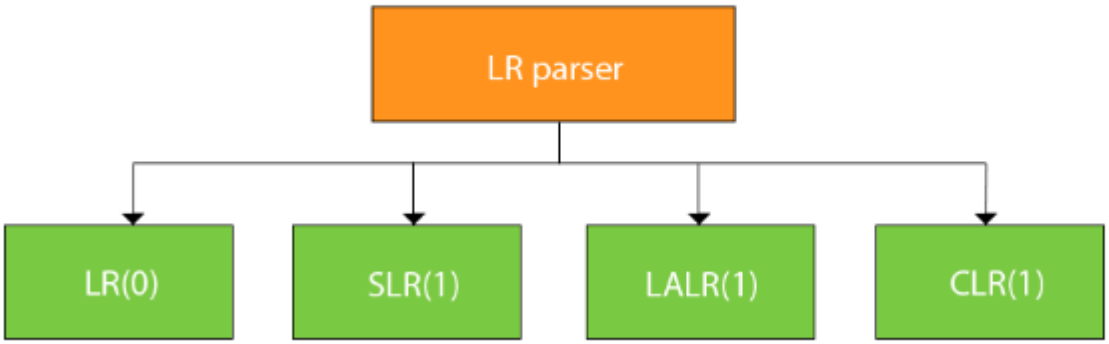
"K" is the number of input symbols of the look ahead used to make number of parsing decision.

Various steps involved in the LR (1) Parsing:

- For the given input string write a context free grammar.
- Check the ambiguity of the grammar.
- Add Augment production in the given grammar.
- Create Canonical collection of LR (0) items.
- Draw a data flow diagram (DFA).
- Construct a LR (1) parsing table.

Solved and verified by Masters

LR parsing is divided into four parts: LR (0) parsing, SLR parsing, CLR parsing and LALR parsing.




## LR(0) Table

- If a state is going to some other state on a terminal then it correspond to a shift move.
- If a state is going to some other state on a variable then it correspond to go to move.
- If a state contain the final item in the particular row then write the reduce node completely.

| States | Action | | | Go to | |
|---|---|---|---|---|---|
| | a | b | $ | A | S |
| $I_0$ | S3 | S4 | | 2 | 1 |
| $I_1$ | | | accept | | |
| $I_2$ | S3 | S4 | | 5 | |
| $I_3$ | S3 | S4 | | 6 | |
| $I_4$ | r3 | r3 | r3 | | |
| $I_5$ | r1 | r1 | r1 | | |
| $I_6$ | r2 | r2 | r2 | | |

**Explanation:**

- I0 on S is going to I1 so write it as 1.
- I0 on A is going to I2 so write it as 2.
- I2 on A is going to I5 so write it as 5.
- I3 on A is going to I6 so write it as 6.
- I0, I2and I3on a are going to I3 so write it as S3 which means that shift 3.
- I0, I2 and I3 on b are going to I4 so write it as S4 which means that shift 4.
- I4, I5 and I6 all states contains the final item because they contain • in the right most end. So rate the production as production number.

For any queries feel free to ask rimshazaibabbasi@gmail.com or comment below

**Q: Develop an algorithm to generate LALR (1) DFA.** (5 Marks)

Here is the algorithm to generate LALR(1) DFA.

1. Repeat until all states have distinct code
2. choose two distinct states with same core
3. merge states by creating a new one with the union of all the items point edges from predecessors to new state
4. new state points to all the previous successors

CS606 Paper #3

## 2. How a flow graph can be a Reducible Flow Graph?

A control flow graph G is said to be reducible **if the removal of its back edges leads to an acyclic graph where each node can be reached from the initial node of G**. In this case, all the edges of this acyclic graph are called forward edges.

## 3. What is LR (1) parsing and discuss main advantage?

The LR(1) parser is **a deterministic automaton and as such its operation is based on static state transition tables**. These codify the grammar of the language it recognizes and are typically called "parsing tables". The parsing tables of the LR(1) parser are parameterized with a lookahead terminal.

- The LR parsing method is the most general non-backtrackingshift-reduce parsing method known.
- LR parsers can parse a strictly larger class of grammars than (top-down) predictive parsers.
- LR parsers can usually recognize all programming language construct that can be specified by context-free grammars.
- LR parsers detect errors fast.
- Drawback: it is too much work to construct an LR parser by hand.
- Fortunately, we can use an LR parser generator such as YACC.

Advantages of LR parsing :

**It is able to detect syntactic errors**. It is an efficient non-backtracking shift reducing parsing method.

LR parsers **can parse a strictly larger class of grammars than (top-down) predictive parsers**. LR parsers can usually recognize all programming language construct that can be specified by context-free grammars. LR parsers detect errors fast.

## Q: Characteristics of Live variables?

### Other Uses for Live Variables

Compilers use liveness in many contexts other than finding uninitialized variables.

- Live-variable information plays a critical role in global register allocation (see Section 13.4). The register allocator need not keep values in registers unless they are live; when a value makes the transition from being live to being not live, the allocator can reuse its register for another purpose.
- Live-variable information is used to improve the ssa construction; a value does not need a $\phi$-function in any block where it is not live. Using live information in this way can significantly reduce the number of $\phi$-functions that the compiler must insert when building the ssa form of a program.
- The compiler can use live information to discover useless store operations. At an operation that stores $v$ to memory, if $v$ is not live then the store is useless. This simple technique works well for unambiguous scalar variables—that is, variables known by only

## 5. Which information we get from 1st section of YACC File? (Repeated)

## 6. Context free parsing context sensitive parsing?

**Context-sensitive grammars are more general than context-free grammars**, in the sense that there are languages that can be described by a CSG but not by a context-free grammar. Context-sensitive grammars are less general (in the same sense) than unrestricted grammars.

# Paper#4

## MERA AAJ CS606 KA PAPER (29-08-2020)

Zyada mcqs past waly thy, moazz or Zain ki files me se.

Subject me se,

### 1. Discuss properties of reduce flow?

**Reducible flow graph**

**Intuition**: Produced by exclusive use of flow of control statements such as if-then-else, while-do, continue, break etc.

**Statistics:** Even programs written using goto statements are almost always reducible.

**Main property:** there are no jumps in the middle of loops from outside; the only entry to a loop is through its header.

**Definition:** A flow graph G is reducible if and only if we can partition the edges into two disjoint groups, often called the forward edges and back edges with properties:- forward edges from an acyclic graph in which every node can be reached from the initial node of G.- the back edges consist only of edges whose heads dominate their tails.

**Q: Name the IR closest to program code?**

An intermediate representation (**IR**) is the data structure or **code** used internally by a compiler or virtual machine to represent **source code**.

Q: Why we break source code to machine language?

Assemblers are used to translate a program written in a low-level assembly language into a machine code (object code) file **so it can be used and executed by the computer**. Once assembled, the program file can be used again and again without re-assembly.

### Q: What does the compiler need to consider when applying optimization?

The compiler needs to minimize or maximize some attributes in an executable program. It is better to minimize the time taken to execute the program. Also to minimize the occupied memory.

### Q: Define NFA?

Nonfinite automata is automata theory with finite states. It has an initial state and transition state. It is used to automate the things. It does not tell the exact location to which the machine is rotated. It can be converted to dfa.

Q: **In Main memory, what type of information is stored in data segement?**

In memory the code generated by the intermediate Representation is stored. It include the source file, which contain the tokenized symbol, variable, objects and other parse data. This code is very near to the machine language.

**Q: Identify whether the grammer is LR(1) and LL(1)?**

S->Aa|Bb

A->c

B->c

The grammer is LR( 1 )grammer. It gives the output.

**Q: Characteristics of Live variables?**

Live variable are the variable in an object that can be used for further use. Live-variable information plays a critical role in global register allocation. The characteristics of live variable are; They help in calculation of each variable in the compiler. Live-variable information is used to improve the ssa construction. The compiler can use live information to discover useless store operations

Masters

**Q: Algorithm to generate LALR(1) DFA?**

Algorithm to generate LALR1 DFA is below:

1. Repeat until all the states have distinct code.
2. Choose two distinct states with same core.
3. Merge these states by a new state having union of all items.
4. New state points all previous states.

Q: Which information we get from 3rd section of YACC File? (repeat)

**Q: How can complier transformation improve program?**

Compiler transformation improves program, by reducing its unnecessary attributes. The main requirement of a program is to reduce the run time of

program. Another less necessary is to occupy less memory on program. The garbage collection should work perfect way.

**Q: LL Parse table?**

Q: **For each instruction, show which variable are live variable are live immediately after instruction execution (5marks)** (page 137) Lec 44

1. A=7
2. B=A+2
3. C=A+B
4. D=C+B
5. B=C+B

**Solution:**

**Live= {2, 7}**

A=7

Live= {A, 2}

B=A+2

Live = {A,B}

C=A+B

Live= {A, C, B}

D=C+B

Live= {A, C, D}

B=C+B

Live = {A, B, C, D}

**Q: what is non deterministic fa?**

Answer: Nonfinite automata is automata theory with finite states. It has an initial state and transition state. It is used to automate the things. It does not tell the exact location to which the machine is stated. It can be converted to dfa.

Masters

**Q: Productions were given and canonical collection set was asked to build using productions. Lecture 24, 25 (Page 80)**

**Q: Action table and goto table were given we were asked to show in stack and input buffer when we use garbage collection for memory management?**

It is an automatic memory management tool. We use the garbage collection for memory management to reduce the extra memory occupied by the program at run time and the objects that are no longer exist in program. It allocate and deallocate the memory to a program.

**Q: If on right hand side of production there are k symbols then how many positions are there for place holder?**

If the right hand side of production there are k symbols then the position on the placeholder will be k+1.

**Q:Production was given to determine whether this is of LR (1) or LL(1) ?**

**Q:Code was given and we have to find basic blocks?Page 132 Lecture 43**

Masters

*Algorithm*: partition 3-address statements into basic blocks:

1. Determine the set of leaders – the first statements of basic blocks. The rules are:

   - The first statement is a *leader*
   - Any statement that is the target of a conditional or unconditional goto is a leader
   - Any statement that immediately follows a goto or conditional goto is a leader

2. For each leader, its basic block consists of the leader and all statements up to but not including the next leader or the end of the program

**Q: use of address descriptor in code generation algorithm of 5 marks**

An address descriptor is used **to store the location where current value of the name can be found at run time**.

A register descriptor is used **to keep track of what is currently in each registers**. The register descriptors show that initially all the registers are empty. An address descriptor stores the location where the current value of the name can be found at run time.

https://www.sathyabama.ac.in › sites › files › Unit-5 ▾ PDF

UNIT – V - Compiler Design – SCS1303 - Sathyabama

Address descriptor : **Values of the names (identifiers) used in the program might be stored at different locations while in execution**. Address descriptors are used to keep track of memory locations where the values of identifiers are stored.

**Q: Third section of YACC represents what kind of information?**

**Q: How inherited attribute are different synthesized with respect to grammar attribute?**

Inherited attributes are different then synthesized with respect to the following main reasons.

1. On the base of direction of flow value.
2. On the base of parsing tree. (Syn follows bottom up, while inherited follows top down parsing technique.)

**(**Attributes are distinguished based on the direction of value flow. Attributes of a node whose values are defined wholly in terms of attributes of node's children and from constants are called synthesized attributes. Values used to compute synthesized attributes flow bottom-up in the parse tree.

Attributes whose values are defined in terms of a node's own attributes, node's siblings and node's parent are called inherited attributes. Values flow top-down and laterally in the parse tree. The following attributed tree shows

the inherited and synthesized attributes for the input signed binary number -101. )

**Q: How context free parsing is different from context sensitive parsing?**

**Context Free parsing:**

It is a formal parsing in which each context is in form of Vàw. Where V in a non-terminal and w may be both non terminal and terminal, it's value may be zero.

**Context sensitive language** is a formal context grammar in which each left hand side and right hand side of production rules is surrounded by the context in case of terminal and non-terminal.

**Q: Brief note on Reducible Flow Graphs?**

**Q: What are basic operation of shift reduce parser to parser a CFG? Explain activities how they performed?**

Shift moves ► one place to the right which shifts a terminal to the left string

E + (► int) ⇒ E + (int ►)

In the reduce action, the parser applies an inverse production at the right end of the left string. If E → E + (E) is a production, then

E + ( E+(E)►) ⇒ E + ( E ►)

The basic operation of shift reduce parser in CFG are.

One symbol look ahead and indicates either the value should be shift or reduce. In case of shift a symbol pushes to the stack that is increased by one and look ahead for next symbol. It that symbol needed to reduce then the stack comes to its on position.

Paper#6

**CS606 FINAL TERM PAPER ON AUGUST 20, 2022**

aoa 80% mcqs were past papers.
Q Which information we get from the second YACC file?

Masters

Q Which type of information is store in memory of code segment.

Paper#7

**CS606 FINAL TERM PAPER ON March 2022**

my today papr at 11:00 AM

Q: Third section of YACC represents what kind of information?
Q: How inherited attribute are different synthesized with respect to grammar attribute?
Q: How context free parsing is different from context sensitive parsing?
Q: For each instruction, show which variable are live variable are live immediately after instruction execution what are basic operation of shift reduce parser to parser a CFG ? Explain activities how they performed?
**Q: LL () table was give. Show content of stack and input buffer for parse of string "ab" where s is start symbol? Lecture 23 page 77**
Objective was conceptual.. Some objective from moaz file.

Paper#8

**CS606 FINAL TERM PAPER ON sep 2022**

Assalm o ALaikum
What does the compiler need to consider when applying optimization?

Define NFA?

In Main memory, what type of information is stored in data segement?

Identify whether the grammer is LR(1) and LL(1)?

S->Aa|Bb

A->c

B->c

Characteristics of Live variables?

Algorithm to generate LALR(1) DFA?

Which information we get from 3rd section of YACC File?

Masters

How can complier transformation improve program?

LL Parse table ?

INT and BOOl main sy bhi tha 5 marks ka ? lecture 39

For each instruction, show which variable are live variable are live imidiately after instruction execution

(5marks)

1. A=7
2. B=A+2
3. C=A+B

4.D=C+B

5.B=C+B

## Paper#8

**CS606 FINAL TERM PAPER ON March 2022**

Assalamualaikum.

CS606 exam held today:

1. A tree was given, In-order traversal was required.
2. What are characteristics of Live Variables?
3. What does the compiler need to consider when applying optimization?

4. In main memory what type of information is stored in data segment? (repeated)
5. How can compiler transformation improve program? (repeated)
6. What is the use of Address descriptor in Code Generation algorithm? (repeated)
7. A table with left column filled with terms, and right column filled with meanings of terms was given. It was required to match the term in the left column to its meaning on the right.

Masters

| | word | Stack | Handle | Action |
|---|---|---|---|---|
| 9 | × | Expr – Term ▶ | - none - | *shift* |
| 10 | id | Expr – Term × ▶ | - none - | *shift* |
| 11 | $ | Expr – Term × id▶ | <Factor → id,5> | *reduce* |
| 12 | $ | Expr–Term × Factor▶ | <Term→Term×Factor,5> | *reduce* |
| 13 | $ | Expr – Term ▶ | <Expr → Expr – Term,3> | *reduce* |
| 14 | $ | Expr ▶ | <Goal → Expr,1> | *reduce* |
| 15 | $ | Goal | - none - | ***accept*** |

## Paper#10

### CS606 FINAL TERM PAPER ON March 2022

Q: How Lex different from Flex? (2marks) mid

Q:For each instruction, show which variable are live variable are live immediately after instruction execution (5marks) repeat

1. A=7
2. B=A+2
3. C=A+B

4.D=C+B

5.B=C+B

Q: write algorithm LALR(1) parser (5 marks) repeat

Q: find semantic rules from given expressions.(5 Marks)

Masters

| *Production* | *Semantic Rule* |
|---|---|
| E → E1 + E2 | E.nptr = mknode('+', E1.nptr, E2.nptr) |
| E → E1 * E2 | E.nptr = mknode('*', E1.nptr, E2.nptr) |
| E → – E1 | E.nptr = mknode('–', E1.nptr) |
| E → ( E1 ) | E.nptr = E1.nptr |
| E → num | E.nptr = mkleaf('num', num.val) |

## Paper#11

**CS606 FINAL TERM PAPER ON March 2022**

**My todays Paper of 4:30pm**

**Mostly mcq's was from saher objective file.**

**Q:** Give an example of function preventing transformation?(2)

Example: As the relationship t 4:=4*j surely holds after such an assignment to t 4 in Fig. and t4 is not changed elsewhere in the inner loop around B3, it follows that just after the statement j:=j -1 the relationship t4:= 4*j-4 must hold. We may therefore replace the assignment t 4:= 4*j by t4:= t4-4.

https://edurev.in › ... › Compiler Design

Function Preserving Transformations | Study Compiler Design

**Q:** What is the purpose of using Live variable?(2)

Live-variable information **plays a critical role in global register allocation** (see Section 13.4). The register allocator need not keep values in registers unless they are live; when a value makes the transition from being live to being not live, the allocator can reuse its register for another purpose.

https://www.sciencedirect.com › topics › computer-science

Live Variable - an overview | ScienceDirect Topics

**Q:** Identify whether the following grammer is LR(1) or LL(1)?(2)

S à Aca|Bcb

Aàc

Bàc

**Q:** What are the basic blocks for the following 3-addresses code segment? (3)

(Code was given) (repeat)

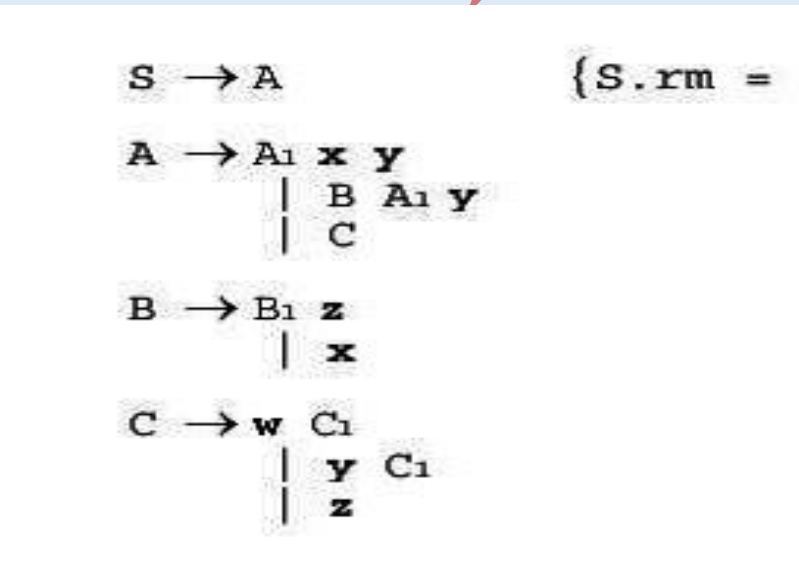
**Q:** How context free parsing is different from context-Sensitive Switching? (3) (repeat)

**Q:** How inherited attribute can be used Syntax-directed definition? (3) (repeat)

**Q:** Consider the following ACTION/Goto table (5)

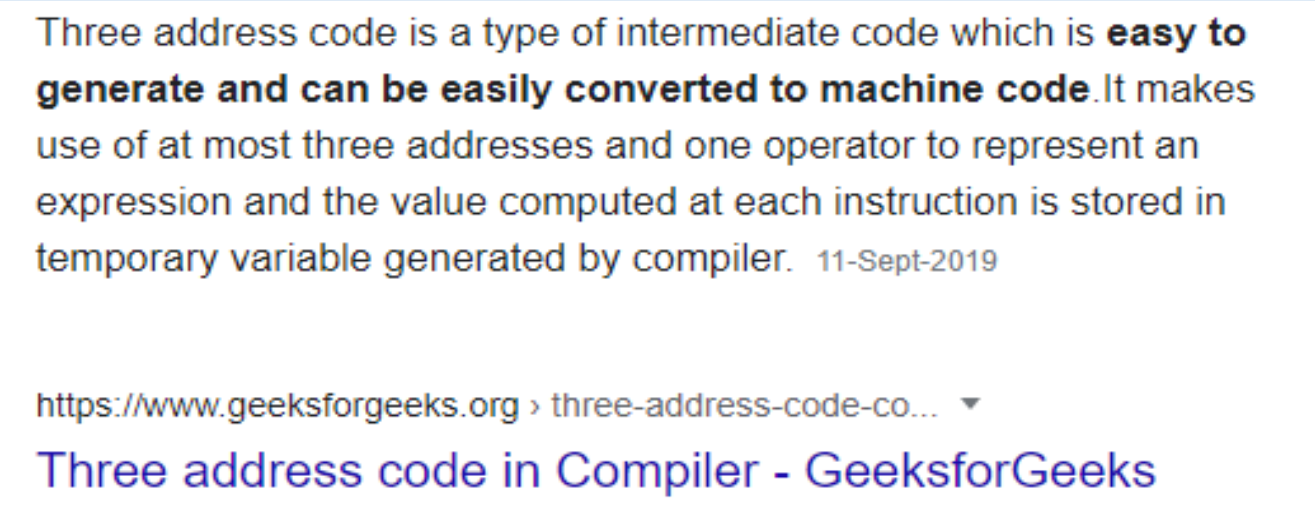
*Table was given and the statement was (Show the contents of the stack and input buffer for the shift-reduce parse of input "a" assuming state 0 is the start state)

**Q:** Add semantic rules to the following grammar to compute the attribute rm, whose value is the rightmost terminal in the string we parsed. For example, if the string parsed were zxyxy, S.rm would be y.(5)




Masters

**Q:** Source language expression can be break down into simple parts using three address code. you are required to give any two reasons of doing so(5)




Masters

## Paper#12

**CS606 FINAL TERM PAPER ON March 2022**

Aslam-O-Alikum. MCQ's are 30% from past papers.

Subjective Questions are:

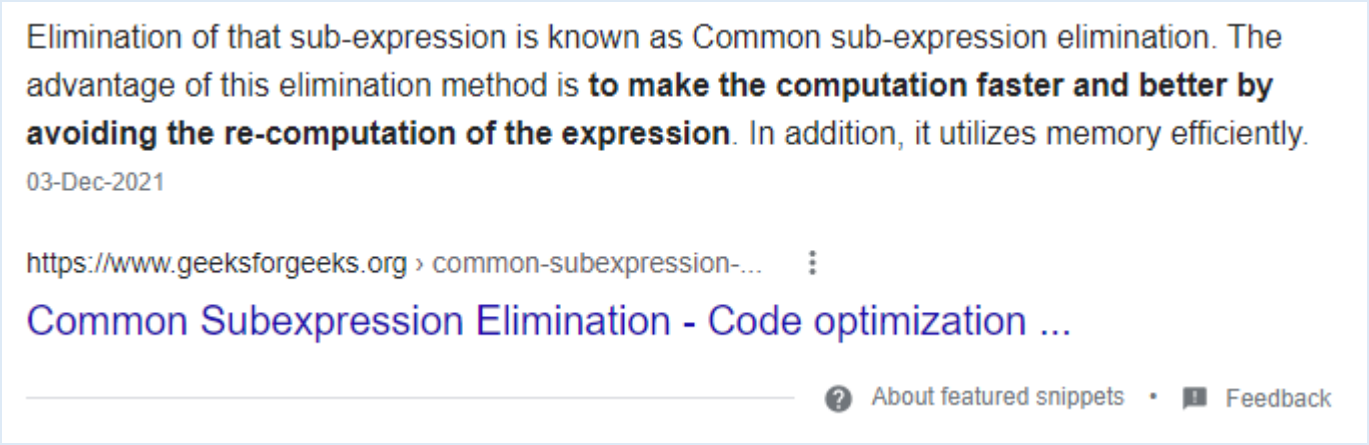
**1)** S ->{SX} | a 2 Marks

X -> ε| + SY | Yb

Y -> ε | - SXc

Find Non- Terminals and Terminals

2) why use common sub-expression. 2 Marks


Elimination of that sub-expression is known as Common sub-expression elimination. The advantage of this elimination method is **to make the computation faster and better by avoiding the re-computation of the expression**. In addition, it utilizes memory efficiently.
03-Dec-2021

https://www.geeksforgeeks.org › common-subexpression-...

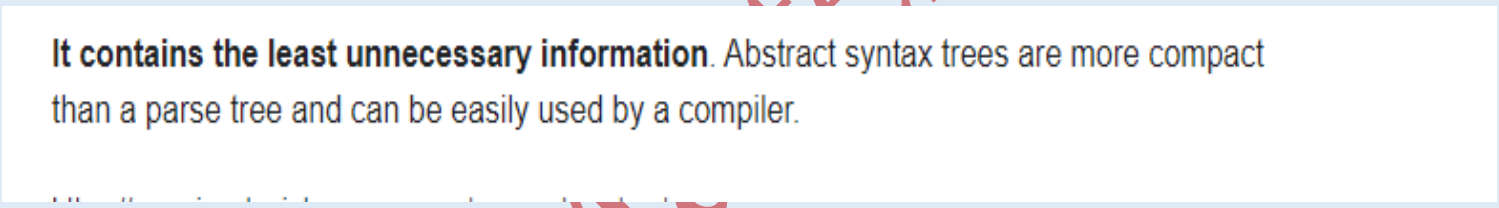
Common Subexpression Elimination - Code optimization ...

About featured snippets • Feedback

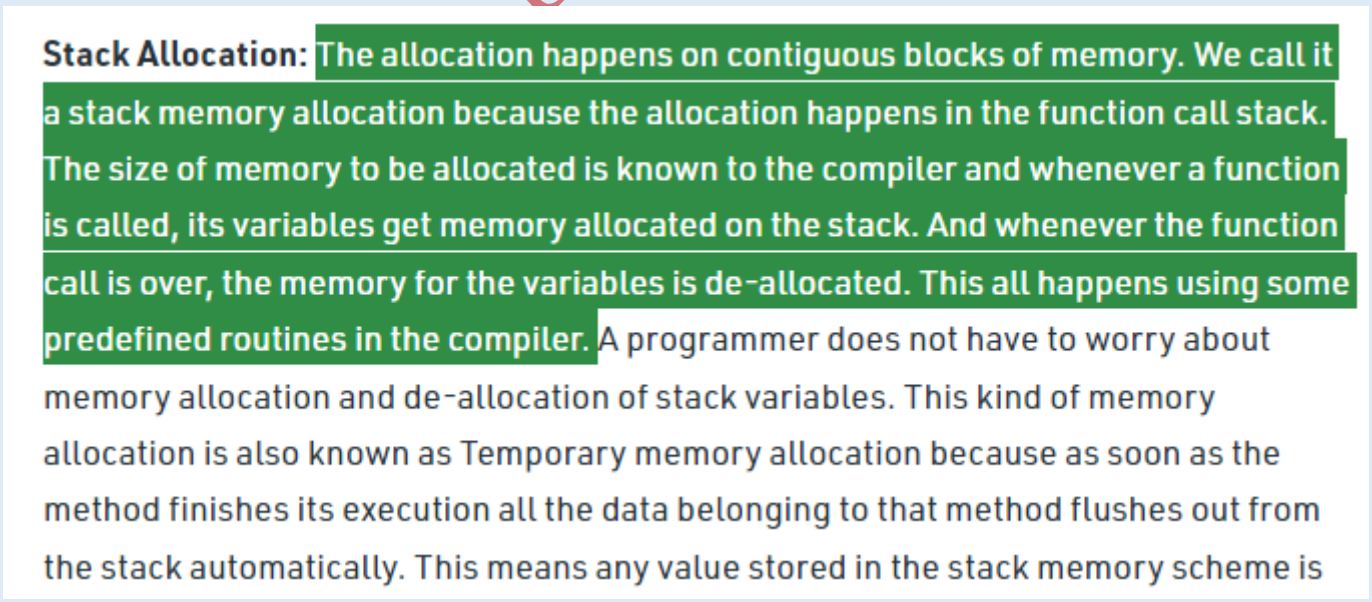
3) Q: Which Information we can get the second section of YACC file. 2 Marks (repeat)

Q: In main memory, what type of information store in data segment. 2 Marks (repeat)

4) One Question conceptual about get rid. Microsoft use parsing technique and it hava face problem. we will help about alternative solution. e.t.c 3 Marks


**It contains the least unnecessary information**. Abstract syntax trees are more compact than a parse tree and can be easily used by a compiler.

5) When compiler use stack for allocation at run time for storing variable. 3 Marks


**Stack Allocation:** The allocation happens on contiguous blocks of memory. We call it a stack memory allocation because the allocation happens in the function call stack. The size of memory to be allocated is known to the compiler and whenever a function is called, its variables get memory allocated on the stack. And whenever the function call is over, the memory for the variables is de-allocated. This all happens using some predefined routines in the compiler. A programmer does not have to worry about memory allocation and de-allocation of stack variables. This kind of memory allocation is also known as Temporary memory allocation because as soon as the method finishes its execution all the data belonging to that method flushes out from the stack automatically. This means any value stored in the stack memory scheme is

6) Write Basic block 3-address code segment. 3 Marks (repeat)

Solved and verified by Masters

1) a = 1

2) b = 2

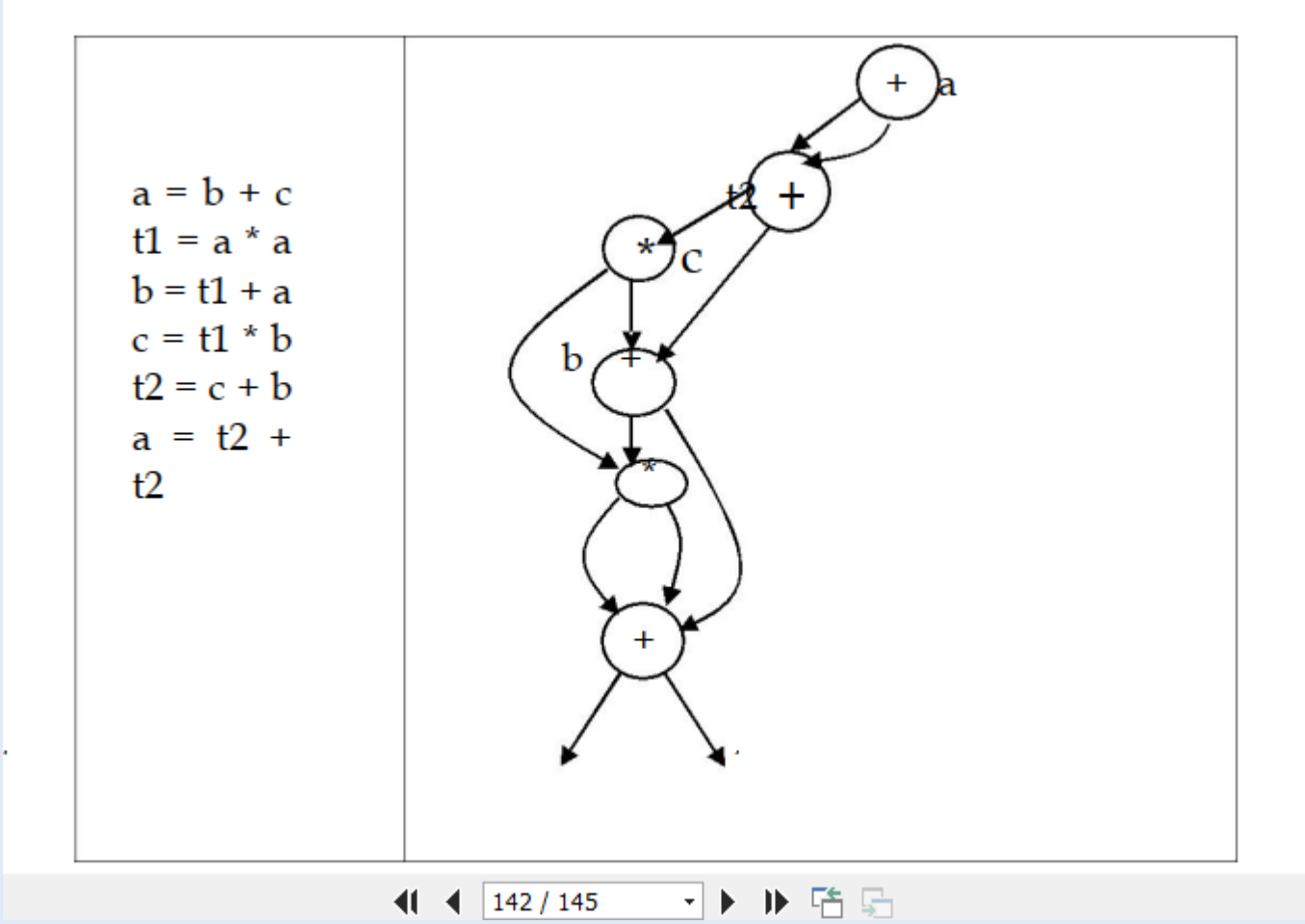
3) c = a + b

4) d = c - a

**10)** One Question about Lable and Goto Simply describe about uses of those 2 control flow statement. Marks 5

**11)** DAG representation of Basic Block Marks 5

t1 := a + b t2 := c + d t3 := e - t2 t4 := t1 - t3




**12)** Ye question kafi lengthy tha lekan ye parse tree k topic main sy he tha. or parse tree he bnana tha 5 Marks

Best of Luck. :-)

Paper#14

**CS606 FINAL TERM PAPER ON March 2022**

For any queries feel free to ask rimshazaibabbasi@gmail.com or comment below

***Complete paper***

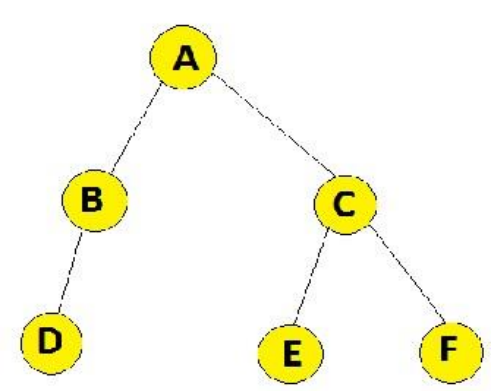
Paper was very easy and short mostly from Past papers.

**MCQ 30 – 35 from Past papers**

## Subjective Part

**A code statement was given have to identify LR (1) or LL (1) (2marks)**

**Traverse the given tree in postorder. (2marks)**




Masters

**Why compiler uses multiple instruction representation? 3 Marks**

The instruction set, also called ISA (instruction set architecture), is part of a computer that pertains to programming, which is more or less machine language. The instruction set **provides commands to the processor, to tell it what it needs to do**. 05-Feb-2021

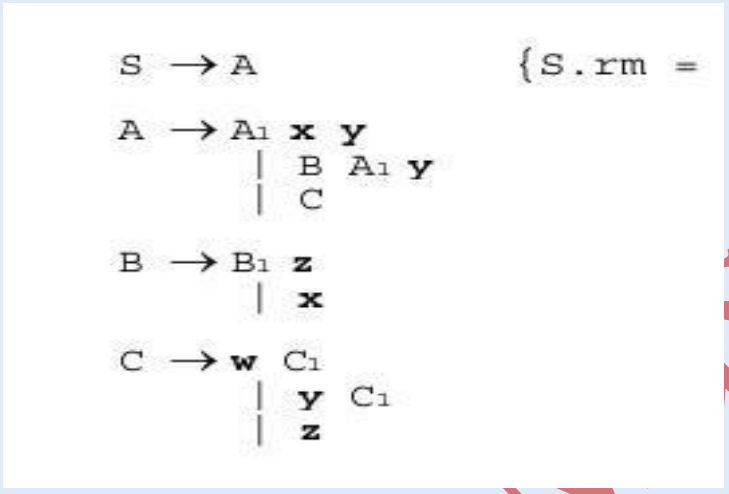
**Which information we can get from the first section of YACC file? (3 Marks)**

**Differentiate attribute grammar and syntax-directed grammar. (3marks)**

**What is reducible flow graph? (3 marks)**

**Add semantic rules to the following grammar to compute the attribute rm, whose value is the rightmost terminal in the string we parsed. For example, if the string parsed were zxyxy, S.rm would be y.**

**(5marks)**




S.nptr = A.nptr

S.nptr = mknode ('rm', y.val)

A.nptr = mknode('*', A1.nptr, x.val, y.val)

A.nptr = mknode('*', B.nptr, A1.val, y.val)

B.nptr = mknode('*', B1.nptr, z.val)

B.nptr = mkleaf('*', x.val)

**A grammar was given we have to find First and Follow (5 marks)**
**An expression was given we have to draw syntax tree. (5marks)**
**Role of GOTO and Label (5 marks)**

## Some important Objective MCQs shared by students:

1. The regular expressions a*|b* and (a|b)* describe the _____set of strings.

Same

**Different**

Onto

2. This MCQ was asked 2 times in the paper

S --> a | B

B --> Bb | E

The non-terminal _____ is left recursive.

**Answer: B**

**3. A lexical analyzer generator automatically constructs a _____ that recognizes tokens.**

**FA**

PDA

DP

None of the given

4) The following two items A -> P • Q B -> P • Q can co-exist in an _____ item set.

**LR**

LS

LT

PR

Masters

**5) Three-address codes are often implemented as a _____.**

**Set of quadruples Page no: 104**

**6) Attributes of a node, whose, values are defined wholly in terms of attributes of node's children and from constants are called _____.**

**Synthesized attributes Page no: 92**

**7) The notation _____ instructs YACC to push a computed attribute value on the stack. Select correct option:**

**Answer: $$ Page no: 98**

**8) What does following statement represent? x[i] = y** Answer: indexed assignment Page no: 107

**9) S --> A B A --> e | aA B --> e | bB - FIRST(S) contains ___ elements**

**Answer: 3**

**Page no: 46**

**First{S} = {e,a,b}**

**10) Dotted items (T a •b) record which part of a token has already been matched. Integer ? ([0-9])+ • this is a _____ item.**

**Answer: Reduced**

**11) A _____ is a top down parser.**

**Answer: Predictive Parsing Page no: 46**

12) Intermediate Representation (IR) stores the value of its operand in _____

**Registers Page no: 10**

Memory

Hard disk

Secondary storage

Masters

**13) In compilation process Hierarchical analysis is also called Select correct option:**

Parsing

**Syntax analysis**

Both Parsing and Syntax analysis

None of given

**14)Consider the grammar A --> B C D**

B --> h B | epsilon

C --> C g | g | C h | i

D --> A B | epsilon

Follow of C is _____ .

h **g,**

**h, i,**

**$** g, i

g

**15). In PASCAL _____ represent the inequality test.**

:=

=

**<>**

None of the given

**16)** In DFA minimization we construct one _____ for each group of states from the initial DFA.

**State Page no: 25**

NFA

PDA

None of given

**17)** Intermediate Representation (IR) stores the value of its operand in

Masters

**Registers**

Memory

Hard disk

Secondary storage

**18) Yacc contains built-in support for handling ambiguous grammars resulting in shift-reduce conflicts. By default these conflicts are solved by performing the _____**

Shift action

Reduce action

Shift and reduce actions

De-allocation of memory

**19) When generating a lexical analyzer from a _____ description, the item sets (states) are constructed by two types of "moves": character moves and e moves.**

Character

Grammar

**Token (Page 18)**

Sentence

**20)** The notation _____ instructs YACC to push a computed attribute value on the stack.

**$$ (Page 98)**

&&

## --

**21) Performing common sub expression elimination on a dependency graph requires the identification of nodes with the same operator and operands. When using a hash table (with a hash function based on operator and operands) all_____ nodes can be identified in linear time.**

**Common**

Uncommon

Next

Previous

Masters

**22)Linear IRs resembles pseudo-code for same _____.**

Automated Machine
Mechanical machines
Token machines
**Abstract machine (Page 100)**

**23)**A lexical analyzer generator automatically constructs a _____ that recognizes tokens.

**FA (Page 18)**
PDA
DP
None of the given

**24)** The following two items A -> P • Q B -> P • Q can co-exist in an _____ item set.

**LR**
LS
LT
PR

**25) Simple code generation considers one AST node at a time. If the target is a** register machine, **the code can be generated in one _____ traversal of the AST, possibly introducing temporaries when running out of registers.**

**Depth-first**
Breadth-first
Top-Down
Bottom-Up

Masters

**26) Simple code generation considers one AST node at a time. If the target is a *stack machine*, the code can be generated in one _____ traversal of the AST, possibly introducing temporaries when running out of registers.**

**Depth-first**

Breadth-first

Top-Down

Bottom-Up

**27) In compilers, linear analysis is also called _____ a. Lexical Analysis**

b. Scanning

c. **Lexical Analysis and Scanning**

d. None of the given

**28) The parser generator YACC can handle _____ grammar.**

a. LL(1)

b. LT(1)

c. LS(1)

d. LF(1)

**29)** A linker combines multiple object files into a _____ executable object.

**a. Single**

b. Double

c. Triple

d. Quadruple

Masters

## 30) Hybrid IRs combine elements of _____

a. Graphical (Structural)
b. Linear IRs
c. **Both Graphical (Structural) and linear**
d. Non-Linear IRs

Most Important Lectures:

Lecture 45
Lecture 44
Lecture 43
Lecture 42
Lecture 38
Lecture 39
Lecture 30
Lecture 27

Masters

| Solved by: | Sahar (well wisher) | Class | BSCS 6th Semester |
|---|---|---|---|
| Subject | CS606 (COMPILER CONSTRUCTION) | Solution Type: | Final Term Solved MCQZ and Quizez including Papers of Year : 2013,2012,2011,2010,2009… 2006 |
| Institute: | Virtual University of Pakistan | | |

# Final term QUIZEZ
# Year 2013

**1)**

The regular expressions a*|b* and (a|b)* describe the _____set of strings.

Same

**Different**

Onto

**2)**

A canonical collection of sets of items for an augmented grammar, C is constructed as -----

For each set / in C and each grammar symbol X where goto (C, X) is empty and not in C add the set goto (C, X) to C.

**The first set in C is the closure of {[S' --> .S]}, where S' is starting symbol of original grammar and S is the starting non-terminal of augmented grammar. Page no : 72**

The first set in C is the closure of {[S' --> .S]}, where S is starting symbol of original grammar and S' is the starting non-terminal of original grammar.

**3)**

S --> a | B

B --> Bb | E The non-terminal _____ is left recursive.

**Answer: B**

**4)**

Consider the following grammar, S --> aTUe T --> Tbc/b U --> d And suppose that string "abbcde" can be parsed bottom-up by the following reduction steps:

(i) aTbcde
(ii) aTde
(iii) aTUe
(iv) S So, what can be a handle from the following?

**The (a) in (aTUe) not confirm**

The (a) in (aTUe)

The (d) in (aTUe)

**5)**

In an attribute grammar each production rule(N--> a) has a corresponding attribute evaluation rule that describes how to compute the values of the _____attributes of each particular node N in the AST.

Select correct option:

1. **Synthesized page no : 92**
2. Complete
3. Free
4. Bounded

**6)**

Consider the following grammar, S --> aTUe T --> Tbc/b U --> d And suppose that string “abbcde” can be parsed bottom-up by the following reduction steps: (i) aTbcde (ii) aTde (iii) aTUe (iv) S So, what can be a handle from the following?

Select correct option:

1. **The whole string, (aTUe) Page no : 68**
2. The whole string, (aTbcde)
3. The whole string, (aTde)
4. None of the given

**7)**

When constructing an LR(1) parser we record for each item exactly in which context it appears, which resolves many conflicts present in _____parsers based on FOLLOW sets.

Select correct option:

1. SLR(1)
2. LRS(1)
3. RLS(1)
4. **None of the given**

**8)**

The _____translation statements can be conveniently specified in YACC

Select correct option:

1. **Syntax-directed Page no : 120**
2. Image-directed
3. Sign-directed
4. None of the given.

**9)**

We use ----- to mark the bottom of the stack and also the right end of the input when considering the Stack implementation of Shift-Reduce Parsing.

Select correct option:

1. Epsilon

2. #
3. **$ Page no : 65**
4. None of the given

**10)**

Grammars with LL(1) conflicts can be made LL(1) by applying left-factoring, substitution, and left-recursion removal. Left-factoring takes care of _____conflicts.

Select correct option:

1. FIRST/FIRST
2. FIRST/SECOND
3. SECOND/FIRST
4. **None of the given**

**11)**

When generating a lexical analyzer from a token description, the item sets (states) are constructed by two types of “moves”: character moves and ____ moves.

Select correct option:

1. **E (empty string) Page no : 18**
2. #
3. @
4. none of given

**12)**

Bottom-up parsers handle a _____ class of grammars.

Select correct option:

1. **large Page no : 63**
2. small
3. medium
4. none of the given

**13)**

Let a grammar G = (Vn, Vt, P, S) is modified by adding a unit production S’--> S to the grammar and now starting non-terminals becomes S’ and grammar becomes G’ = (Vn U {S’}, Vt, P U{S’ --> S}, S’). The Grammar G’ is called the ----------

Select correct option:

1. **Augmented Grammar Page no : 76**
2. Lesser Grammar
3. Anonymous Grammar
4. none of given

**14)**

The LR (1) items are used as the states of a finite automaton (FA) that maintains information about the parsing stack and progress of a shift-reduce parser.

Select correct option:

1. **True** **Page no: 74**
2. false

Back patching to translate flow-of-control statements in ____ pass.

Select correct option:

1. **one** **Page no : 111**
2. two
3. three
4. all of the given

**15)**

Consider the following grammar, S --> aTUe T --> Tbc/b U --> d And suppose that string “abbcde” can be parsed bottom-up by the following reduction steps: (i) aTbcde (ii) aTde (iii) aTUe (iv) S So what can be a handle from the following?

Select correct option:

1. The second (b) in (abbcde)
2. The first (b) in (abbcde)
3. The substring (cd) in (abbcde)
4. None of the given

**16)**

S --> A | xb
A --> aAb | x This grammar contains a reduce-reduce conflict.
Select correct option:

**True**
False

**17)**

Yacc contains built-in support for handling ambiguous grammars resulting in _____ conflicts.
Select correct option:

**Shift-reduce**
Shift-Shift
Shift-second
None of the given

**18)**

A lexical analyzer generator automatically constructs a _____ that recognizes tokens.
Select correct option:

**FA**
PDA
DP
None of the given

**19)**

When generating a lexical analyzer from a token description, the item sets (states) are constructed by two types of “moves”: character moves and ____ moves.
Select correct option:

**E (empty string)** **Repeated**
#
@
None of the given

**20)**
Attributes whose values are defined in terms of a node’s own attributes, node’s siblings and node’s parent are called _____ .
Select correct option:

**Inherited attributes** **Page no : 92**
Physical attributes
Logical attributes
Un-synthesized attributes

**21)**
Register allocation by graph coloring uses a register interference graph. _____ nodes in the graph are joined by an edge when the live ranges of the values they represent overlap.
Select correct option:

**Two** **page no : 136**
Three
Four
Five

**22)**
The following two items A -> P • Q B -> P • Q can co-exist in an _____ item set.
Select correct option:

**LR**
LS
LT
PR

**23)**

Goto L statement represent

Select correct option:

**Unconditional jump** **Page no : 107**

**24)**

Three-address codes are often implemented as a _____.

Select correct option:

**Set of quadruples** **Page no : 104**

**25)**

Attributes of a node whose values are defined wholly in terms of attributes of node's children and from constants are called _____.

**Synthesized attributes Page no : 92**

**26)**

The error handling mechanism of the yacc parser generator pushes the input stream back when inserting 'missing' tokens.

Select correct option:

**Answer: True**

**27)**

The notation _____ instructs YACC to push a computed attribute value on the stack.

Select correct option:

**Answer: $$ Page no : 98**

**28)**

Flow of values used to calculate synthesized attributes in the parse tree is:

Select correct option:

**Answer: Bottom-up Page no: 92**

**29)**

Following statement represents: if x relop y goto L

**Answer: conditional loop Page no : 107**

**30)**

What does following statement represent? x[i] = y

**Answer: indexed assignment Page no : 107**

**31)**

**S --> A B**

**A --> e | aA**

**B --> e | bB - FIRST(S) contains ___ elements**

**Answer: 3 Page no : 46**

**First{S}={e,a,b}**

**32)**

**Dotted items (T→a •b) record which part of a token has already been matched. Integer? ([0-9])+ • This is a _____ item.**

**Answer: Extended Page no : 73**

**33)**

**Parser generator for the grammar LALR (1) is:**

**Answer: YACC, Bison, CUP Page no: 88**

**34)**

**A _____ is a top down parser.**

**Answer: Predictive Parsing Page no: 46**

**35)**

**A lexical analyzer transforms a stream of tokens. The tokens are stored into symbol table for further processing by the parser.**

**Answer: True Page no: 99**

## FINAL TERM QUIZEZ

## YEAR 2012+previous

Question # 1 of 10 ( Start time: 06:31:04 PM ) Total Marks: 1
Parser always gives a tree like structure as output
Select correct option:
**True Page no : 8**

False

Question # 2 of 10 ( Start time: 06:32:19 PM ) Total Marks: 1
Lexer and scanner are two different phases of compiler
Select correct option:

True
**False Page no : 25**

Question # 3 of 10 ( Start time: 06:33:49 PM ) Total Marks: 1
Intermediate Representation (IR) stores the value of its operand in _____ .
Select correct option:

**Registers Page no : 10**
Memory
Hard disk
Secondary storage

Question # 4 of 10 ( Start time: 06:35:17 PM ) Total Marks: 1
In Flex specification file different sections are separated by _____ .
Select correct option:

**%%  Page no: 26**
&&
##
\\

Question # 7 of 10 ( Start time: 06:38:21 PM ) Total Marks: 1
_____ phase which supports macro substitution and conditional compilation.
Select correct option:

**Semantic**
Syntax
Preprocessing
None of given

Question # 9 of 10 ( Start time: 06:40:30 PM ) Total Marks: 1
Flex is an automated tool that is used to get the minimized DFA (scanner).
Select correct option:

True
**False  Page no: 26**

Question # 10 of 10 ( Start time: 06:41:58 PM ) Total Marks: 1
_____tree in which each node represents an operator and children of the node represent the operands.
Select correct option:

**Abstract syntax  Page no : 100**
Concrete syntax
Parse
None of the given

Question # 2 of 10 ( Start time: 06:47:42 PM ) Total Marks: 1
In _____certain checks are performed to ensure that components of a program fit together meaningfully.
Select correct option:

Linear analysis
Hierarchical analysis

**Semantic analysis** **Page no : 33**
None of given

Question # 3 of 10 ( Start time: 06:48:58 PM ) Total Marks: 1
In compilation process Hierarchical analysis is also called
Select correct option:

Parsing
**Syntax analysis**
Both Parsing and Syntax analysis
None of given

Question # 5 of 10 ( Start time: 06:51:46 PM ) Total Marks: 1
Which of the following statement is true about Two pass compiler.
Select correct option:

Front End depends upon Back End
**Back End depends upon Frond End** **page no : 5**
Both are independent of each other
None of the given

Question # 8 of 10 ( Start time: 06:55:41 PM ) Total Marks: 1
_____ algorithm is used in DFA minimization.
Select correct option:

James's
Robert's
**Hopcroft's** **Page no : 19**
None of given

Question # 9 of 10 ( Start time: 06:56:23 PM ) Total Marks: 1
Ambiguity can easily be handled by Top-down Parser
Select correct option:

**True**
False

LR parsers can handle _____ grammars.

**Left-recursive** **Page no: 163**

file-recursive

End-recursive

Start-recursive

_____ convert the reloadable machine code into absolute machine code by linking library and reloadable object files.

Assembler

**Loader/link-editor**

Compiler

Preprocessor

Consider the grammar A --> B C D

B --> h B | epsilon

C --> C g | g | C h | i

D --> A B | epsilon

Follow of B is _____ .

h

**g, h, i, $**

g, i

g

Consider the grammar A --> B C D

B --> h B | epsilon

C --> C g | g | C h | i

D --> A B | epsilon

Follow of C is _____ .

**g, h, i, $** **Page no : 47**

g, h, $

h, i, $

h, g, $

An important component of semantic analysis is _____ .

code checking

**type checking page no : 6**

flush checking

None of the given

In PASCAL _____ represent the inequality test.

:=

=

<>
None of the given

Lexical Analyzer generator _____ is written in Java.

Flex

**Jlex Page no : 26**

Complex

None of given

_____avoid hardware stalls and interlocks.

Register allocation

**Instruction scheduling Page no : 10**

Instruction selection

None of given

Consider the following grammar,

A --> B C D

B --> h B | episilon

C --> C g | g | C h | i

D --> A B | episilon

First of A is _____ .

**h, g, i**

g

h

None of the given

Recursive _____ parsing is done for LL(1) grammar.

**Decent Page no : 47**

Ascent

Forward

Backward

One of the core tasks of compiler is to generate fast and compact executable code.

**True**

False

Left factoring of a grammar is done to save the parser from back tracking.

**True Page no:61**

False

Responsibility of _____ is to produce fast and compact code.

**Instruction selection**

Register allocation

Instruction scheduling

**None of given Page no: 9**

_____ algorithm is used in DFA minimization.

James's

Robert's
**Hopcroft's** **Page no:25**
None of given

Compilers are sometimes classified as.

Single pass

Multi pass

Load and go

**All of the given**

In multi pass compiler during the first pass it gathers information about _____ .
Select correct option:
Declaration
Bindings
Static information
**None of the given ****

Question # 9 of 10 ( Start time: 06:40:30 PM ) Total Marks: 1
Flex is an automated tool that is used to get the minimized DFA (scanner).
Select correct option:
True
**False**

In compilation process Hierarchical analysis is also called
Select correct option:
Parsing
Syntax analysis
**Both Parsing and Syntax analysis**
None of given

For each language to make LL(1) grammar, we take two steps, 1st is removing left recurrence and 2nd is applying fin sequence.

True
**False**

_____is evaluated to yield a value.

Command
**Expression**
Declaration
Declaration and Command

LR parsers can handle _____ grammars.
**Left-recursive** **page no : 63**
file-recursive
End-recursive
Start-recursive

Optimal registers allocation is an NP-hard problem.

True

**False** **Page no : 10**

Parser takes tokens from scanner and tries to generate _____ .

Binary Search tree

Parse tree

**Syntax trace** **Page no : 6**

None of the given

Front end of two pass compiler takes_____ as input.

**Source code** **Page no: 5**

Intermediate Representation (IR)

Machine Code

None of the Given

In DFA minimization we construct one _____ for each group of states from the initial DFA.

**State** **Page no : 25**

NFA

PDA

None of given

In Three-pass compiler _____ is used for code improvement or optimization.

Front End

**Middle End** **Page no : 10**

Back End

Both Front end and Back end

_____ of a two-pass compiler is consists of Instruction selection, Register allocation and Instruction scheduling.

**Back end** **Page no : 9**

Front end

Start

None of given

NFA is easy to implement as compared to DFA.

True

**False** **Page no : 19**

We can get an LL(1) grammar by _____ .

Removing left recurrence

Applying left factoring

**Removing left recurrence and Applying left factoring**

None of the given

Parser always gives a tree like structure as output

**True** **page no : 37**

False

Intermediate Representation (IR) stores the value of its operand in

**Registers**
Memory
Hard disk
Secondary storage

In Back End module of compiler, optimal register allocation uses_____ .
O(log n)
O(n log n)
**N P-Complete Page no : 10**
None of the given

**CS 606 Quizez**

**Can a DFA simulate NFA?**

Yes

No

**Sometimes**

Depend upon nfa

**_____ phase which supports macro substitution and conditional compilation.**

**Semantic**

Syntax

Preprocessing

None

**Which of the statement is true about Regular Languages?**

Regular Languages are the most popular for specifying tokens.

Regular Languages are based on simple and useful theory.

Regular Languages are easy to understand.

**All of the given**

**Lexer and scanner are two different phases of compiler**

True

**False Page no :13**

**Lexical Analyzer generator _____ is written in Java.**

Flex

**Jlex Page no :26**

Complex

None of the given

**In a transition table cells of the table contain the _____ state.**

Reject state

**Next state** **Page no 18**

Previous state

None of the given

**The transition graph for an NFA that recognizes the language ( a | b)*abb will have following set of states.**

{0}

{0,1}

{0,1,2}

**{0,1,2,3}** **not sure**

**Front end of two pass compiler takes_____ as input.**

**Source code**

Intermediate representation

Machine code

None

**Functions of Lexical analyzer are?**

Removing white space

Removing constants, identifiers and keywords

Removing comments

**All of the given**

**Question # 1 of 10 ( Start time: 07:25:59 PM ) Total Marks: 1**

**Front-end of a two pass compiler is consists of Scanner.**

**Select correct option:**

**True**

**False**

**Question # 2 of 10 ( Start time: 07:26:40 PM ) Total Marks: 1**

**LL(1) parsing is called non-predictive parsing.**

**Select correct option:**

**True**

**False**

**Question # 3 of 10 ( Start time: 07:28:09 PM ) Total Marks: 1**

**Recursive _____ parsing is done for LL(1) grammar.**

**Select correct option:**

**Backward**

**Forward**

**Ascent**

**Decent**

**Question # 4 of 10 ( Start time: 07:29:35 PM ) Total Marks: 1**

**In predictive parsing table the rows are _____ .**

**Select correct option:**

**Non-terminals**

**Terminals**

**Both non-terminal and terminals**

**None of the given**

**Question # 5 of 10 ( Start time: 07:30:38 PM ) Total Marks: 1**

**We can get an LL(1) grammar by _____ .**

**Select correct option:**

**Removing left recurrence**

**Applying left factoring**

**Removing left recurrence and Applying left factoring**

**None of the given**

**Question # 6 of 10 ( Start time: 07:31:48 PM ) Total Marks: 1**

**_____ of a two-pass compiler is consists of Instruction selection, Register allocation and Instruction scheduling.**

**Select correct option:**

**Backend**

**Frontend**

**Start**

**Question # 7 of 10 ( Start time: 07:32:26 PM ) Total Marks: 1**

**Consider the grammar**

**A --> B C D**

**B --> h B | epsilon**

**C --> C g | g | C h | i**

**D --> A B | epsilon First of C is _____ .**

**Select correct option:**

**g, I look down for reference**

**g**

**h i**

**i**

**Quiz Start Time: 07:25 PM**

**Time Left 81**

**sec(s)**

**Question # 8 of 10 ( Start time: 07:33:12 PM ) Total Marks: 1**

**Alternative of the backtrack in parser is Look ahead symbol in _____ .**

**Select correct option:**

**input**

**output**

**input and output**

**none**

**Question # 9 of 10 ( Start time: 07:34:09 PM ) Total Marks: 1**

**AST summarizes the grammatical structure with the details of derivations.**

**Select correct option:**

**True**

**False**

**Quiz Start Time: 07:25 PM**

**Time Left 81**

**sec(s)**

**Question # 10 of 10 ( Start time: 07:35:06 PM ) Total Marks: 1**

**One of the core tasks of compiler is to generate fast and compact executable code.**

**Select correct option:**

**True**

False

**Left factoring is enough to make LL1 grammar**

True

**False**

**In LL1() parsing algorithm _____ contains a sequence of grammar symbols.**

**Stack**

**Link list**

**Array**

**None**