Compiler Design Solved MCQs- Part 2

MCQs

Multiple Choice Questions

Compiler Design Solved MCQs- Part 2

The output of lexical analyzer is

- A set of regular expressions
- Syntax tree
- Set of tokens
- Strings of character

If conversion from one type to another type is done automatically by the compiler then,it is called

- Implicit conversion
- coercions
- both a and b
- None of the above

Syntax directed translation scheme is desirable because

- It is based on the syntax
- Its description is independent of any implementation
- It is easy to modify
- All of these

A top down parser generates

- Right most derivation
- Right most derivation in reverse
- Left most derivation
- Left most derivation in reverse

Macro-processors are _____

- Hardware

- Compiler
- Registers
- None of the above

Inherited attribute is a natural choice in

- Keeping track of variable declaration
- Checking for the correct use of L values and R values
- Both A and B
- None of these

Concept which can be used to identify loops is

- Dominators
- Reducible graphs
- Depth first ordering
- All of these

Reduction in strength means

- Replacing run time computation by compile time computation
- Removing loop invariant computation
- Removing common sub expression
- Replacing a costly operation by a relatively cheaper one

---

the term environment in programming language semantics is said as

- function that maps a name to value held there
- Function that maps a name to storage location
- The functin that maps a storage location to the value held there
- None of the above

A self relocating program is one which

- cannot be made to execute in any area of storage other than the designated for it at the time of its coding or translation
- Consists of a program and relevant information for its relocation
- Can itself perform the relocation of its address sensitive protions
- All of the above

The lexical analyzer takes_____as input and produces a stream of_____as output.

- Source program,tokens
- Token,source program
- Either A and B
- None of the above

Intermediate code generation phase gets input from

- Lexical analyzer
- Syntax analyzer
- Semantic analyzer
- Error handling

A grammar is meaningless

- If terminal set and non terminal set are not disjoint
- If left hand side of a production is a single terminal
- If left hand side of a production has no non terminal
- All of these

The optimization technique which is typically applied on loops is

- Removal of invariant computation
- Peephole optimization
- Constant folding
- All of these

resolution is externally defined symbols is performed by

- linker
- loader
- compiler
- assembler

Which of the following is used for grouping of characters into tokens?

- Parser
- Code optimization
- Code generator
- Lexical analyzer

Whether a given pattern constitutes a token or not depends on the

- Source language
- Target language
- Compiler
- All of these

A optimizing compiler

- Is optimized to occupy less space
- Is optimized to take less time for execution
- Optimized the code
- None of the above.

Which of the following symbols table implementation is based on the property of locality of reference?

- Hash table
- Search tree

- Self organizing list
- Linear list

Three address code involves

- Exactly 3 address
- At most most 3 address
- No unary operators
- None of these

- A
- B
- C
- D

A compiler is

- A program that place program into memory and prepares them for execution
- A program that automates the translation of assembly language into machine language
- program that accepts program written in high level language and produces an object program
- A program that appears to execute a source program as if it were machine language

---

In a bottom up evaluation of a syntax direction definition ,inherited attributes can

- Always be evaluated
- Be evaluated only if the definition is L -attributed
- Be evaluated only if the definition has synthesized attributes
- None of the above

Which of the following actions an operator precedence parser may

take to recover from an error ?

- Insert symbols onto the stack
- Delete symbols from the stack
- Insert or delete symbols from the input
- All of the above

DAG representation of a basic block allows

- Automatic detection of local common sub expressions
- Automatic detection of induction variables
- Automatic detection of loop variant
- None of the above

Recursive descent parsing is an example

- Top down parsing
- Bottom up parsing
- Predictive parsing
- None of the above

The graph that shows basic blocks and their successor relationship is called

- DAG
- Flow chart
- Control graph
- Hamiltonian graph

Generation of intermediate code based on a abstract machine model is useful in compilers because

- it makes implementation of lexical analysis and syntax analysis easier
- syntax directed translation can be written for intermediate code generation.

- It enhances the portability of the front end of the compiler
- it is not possible to generate code for real machines directly from high level language programs

---

Advantage of panic mode of error recovery is that

- It is simple ti implement
- It never gets into an infinite loop
- Both A and B
- None of these

In operator precedence parsing , precedence relations are defoned

- For all pair of non terminals
- For all pair of terminals
- To delimit the handle
- Only for a certain pair of terminals

An intermediate code form is

- Postfix notation
- Syntax trees
- Three address code
- All of these

Code can be optimized at

- Source from user
- Target code
- Intermediate code
- All of the above

Consider the program given below, in a block-structured pseudo-language with lexical scoping and nesting of procedures permitted.

Program main;
Var ...
Procedure A1;
Var ...
Call A2;
End A1
Procedure A2;
Var ...
Procedure A21;
Var ...
Call A1;
End A21
Call A21;
End A2
Call A1;
End main.
Consider the calling chain: Main -> A1 -> A2 -> A21 -> A1
The correct set of activation records along with their access links is given by

- A
- B
- C
- D

Pee hole optimization

- Loop optimization
- Local optimization
- Constant folding
- Data flow analysis

In which way(s) a macroprocessor for assembly language can be implemented ?

- Independent two-pass processor
- Independent one-pass processor

- Expand macrocalls and substitute arguments
- All of the above

A compiler for a high level language that runs on one machine and produce code for different machine is called

- Optimizing compiler
- One pass compiler
- Cross compiler
- Multipass compiler

Which of the following is used for grouping of characters into tokens (in a computer)

- A parser
- Code optimizer
- Code generator
- Scanner

Local and loop optimization in turn provide motivation for

- Data flow analysis
- Constant folding
- Pee hole optimization
- DFA and constant folding

A simple two-pass assembler does which of the following the first phase

- It allocates space for the literals.
- It computes the total length of the program.
- It builds the symbol table for the symbols and their values.
- All of above

---

the translator is best described as

- Application software
- A system software
- A hardware component
- All of the above

Which of the following is the most powerful parser?

- SLR
- LALR
- Canonical LR
- Operator precedence

Which of the following are language processors?

- Assembler
- Compilers
- interpreters
- All of these

Relocating bits used by relocating loader are specified by

- Relocating loader itself
- Linker
- Assembler
- Macro processor

Synthesized attribute can be easily simulated by a

- LL grammar
- Ambiguous grammar
- LR grammar
- None of the above

A compiler that runs on one machine and produces code for a different machine is called

- Cross compilation
- One pass compilation
- Two pass compilation
- None of the above

_____or scanning is the process where the stream of characters making up the source program is read from left to right and grouped into tokens.

- Lexical analysis
- Diversion
- Modeling
- None of the above

Input to code generator

- Source code
- Intermediate code
- Target code
- All of the above

In analysis the compilation PL/I program the description Create of more optimal matrix is associated with

- Assembly and output
- code generation
- Syntax analysis
- machine independent optimization

A bottom up parser generates

- right most derivation
- right most derivation in reverse
- left most derivation
- left most derivation in reverse

In an absolute loading scheme which loader function is accomplished by assembler ?

- re-allocation
- allocation
- linking
- loading