

Compiler Construction Assignment

1. What Is bootstrapping?

bootstrapping is the process of writing a compiler (or assembler) in the source programming language which it intends to compile. Applying this technique leads to a self-hosting compiler. Many compilers for many programming languages are bootstrapped, including compilers for BASIC, ALGOL, C etc

2. Relation between pattern lexeme and token.

Token: Token is a sequence of characters that can be treated as a single logical entity.

Typical tokens are,

1) Identifiers 2) keywords 3) operators 4) special symbols 5) constants

Pattern: A set of strings in the input for which the same token is produced as output.

This set of strings is described by a rule called a pattern associated with the token.

Lexeme: A lexeme is a sequence of characters in the source program that is matched by the pattern for a token.

3. Differentiate LR and LL parsers

LR parsers	LL parsers
1. These are bottom up parsers	These are top down parsers
2. LR parsers begin at the target string and try to arrive back at the start symbol.	LL parsers begin at the start symbol and try to apply productions to arrive at the target string
3. Ex: SLR, LALR, CLR	Ex: LL(1), LL(0)

4. Eliminate Left recursion from :

$G: A \rightarrow A\alpha/\beta$

$A \rightarrow A\alpha\alpha$

$A \rightarrow A\alpha\alpha\alpha$

$A \rightarrow A\alpha^*$

$A \rightarrow \beta\alpha^*$

$A \rightarrow \beta A^1$

$A^1 \rightarrow \alpha A^1/\epsilon$

Thus after eliminating the left recursion grammar is :

$A \rightarrow \beta A^1$

$A^1 \rightarrow \alpha A^1/\epsilon$

5. In which phase of compilation symbol table is created and when is it used?

symbol table is created in lexical analysis phase of the compiler.

The symbol table is accessed by most phases of a compiler, beginning with the lexical analysis to optimization.

The phases that use symbol table are:

- a. Lexical Analysis phase
 - b. Syntactic Analysis phase
 - c. Semantic Analysis phase
 - d. Intermediate Code Generation phase
 - e. Code Optimization Phase
-

6. What is Attribute in Grammar?

An attribute grammar is a formal way to define attributes for the productions of a formal grammar, associating these attributes to values. The evaluation occurs in the nodes of the abstract syntax tree, when the language is processed by some parser or compiler.

7. What are various storage allocation strategies?

Various storage allocation strategies are:

- a. Static Allocation
 - b. Stack Allocation
 - c. Heap Allocation
-

8. What is type coercion?

type coercion is the process of implicit type casting in which a variable's data type is automatically converted into another without any changes manually.

An example would be the conversion of an integer value into a floating point value or its textual representation as a string.

9. What is peephole optimization?

peephole optimization is a kind of optimization performed over a very small set of instructions in a segment of generated code. The set is called a "peephole" or a "window". It works by recognising sets of instructions that can be replaced by shorter or faster sets of instructions

10. How to overcome errors in parsing?

Using any of the strategies we can overcome errors in parsing

- a. Panic mode
 - b. Error Productions
 - c. Global Corrections
 - d. Phase recovery mode
-