Certainly, here are 50 multiple-choice questions (MCQs) related to Syntax Directed Translation in Compiler Design, along with their answers:

1. What is Syntax Directed Translation (SDT) in compiler design?

a. A technique for creating compiler front-ends

b. A method for generating intermediate code

c. A method for attaching semantic actions to grammar rules

d. A technique for optimizing the compiler's output

Answer: c

2. Which phase of a compiler typically handles Syntax Directed Translation?

a. Lexical Analysis

b. Syntax Analysis

c. Semantic Analysis

d. Code Generation

Answer: c

3. In a syntax-directed translation, what does the syntax tree represent?

a. Abstract Syntax

b. Concrete Syntax

c. Semantic Information

d. Lexical Tokens

Answer: a

4. In SDT, which of the following represents a production rule of a context-free grammar?

a. X → Y

b. X ::= Y

c. X => Y

d. X ::= Y | Z

Answer: b

5. What are semantic actions in SDT?

a. Actions that change the syntax of the language

b. Actions that generate intermediate code

c. Actions that attach meaning to grammar rules

d. Actions that check for syntax errors

Answer: b

6. Which of the following is NOT a common application of Syntax Directed Translation?

a. Error Handling

b. Code Optimization

c. Intermediate Code Generation

d. Syntax Analysis

Answer: d

7. What is an inherited attribute in SDT?

a. An attribute that is passed from parent to child nodes in the syntax tree

b. An attribute that is associated with a non-terminal

c. An attribute that is computed at a node and used by its children

d. An attribute that is only used for code optimization

Answer: a

8. What is a synthesized attribute in SDT?

a. An attribute that is passed from parent to child nodes in the syntax tree

b. An attribute that is associated with a non-terminal

c. An attribute that is computed at a node and used by its children

d. An attribute that is only used for code optimization

Answer: c

9. Which of the following is true regarding SDT and semantic rules?

a. Semantic rules cannot be attached to production rules in SDT.

b. Semantic rules are only used for syntax analysis.

c. Semantic rules can be attached to production rules in SDT.

d. Semantic rules are used to generate lexical tokens.

Answer: c

10. What is the purpose of a semantic rule in SDT?

a. To specify how attributes are computed or inherited for a production rule

b. To define the lexical structure of a language

c. To specify the grammar of a language

d. To identify syntax errors in the source code

Answer: a

11. In SDT, what is the attribute grammar?

a. A formalism used to specify the attributes and the rules for computing them

b. A grammar that defines the syntax of a programming language

c. A grammar that defines the lexical structure of a programming language

d. A formalism used to define the grammar of a programming language

Answer: a

12. In SDT, what is a syntax tree traversal order commonly used for attribute computation?

a. Preorder

b. Postorder

c. Inorder

d. Level order

Answer: b

13. What is the role of the attribute stack in SDT?

a. It is used to maintain attribute values during syntax tree traversal.

b. It is used to define grammar rules.

c. It is used to store intermediate code.

d. It is used to define lexical tokens.

Answer: a

14. In SDT, what is a bottom-up parsing technique used to evaluate attributes?

a. Postfix Evaluation

b. Preorder Traversal

c. LR Parsing

d. Top-down Parsing

Answer: a

15. In SDT, what is a top-down parsing technique used to evaluate attributes?

a. Postfix Evaluation

b. Preorder Traversal

c. LR Parsing

d. Recursive Descent Parsing

Answer: d

16. What is the main advantage of Syntax Directed Translation over other compiler design techniques?

a. It allows for faster code generation.

b. It simplifies the specification of semantics in the compiler.

c. It reduces the compilation time.

d. It makes lexical analysis more efficient.

Answer: b

17. In SDT, what is the purpose of a semantic rule associated with a production rule?

a. To specify the syntax of the language

b. To define the lexical structure of the language

c. To specify how attributes are computed and/or inherited for that production.

d. To create an abstract syntax tree for the input program.

Answer: c

18. Which of the following is NOT a phase of the compiler where SDT can be applied?

a. Lexical Analysis

b. Syntax Analysis

c. Semantic Analysis

d. Code Generation

Answer: a

19. What does an L-attribute grammar in SDT mean?

a. It is an attribute grammar where attributes are computed in a top-down manner.

b. It is an attribute grammar where attributes are computed in a bottom-up manner.

c. It is an attribute grammar that uses only synthesized attributes.

d. It is an attribute grammar that uses only inherited attributes.

Answer: b

20. What is the primary difference between S-attributes and L-attributes in SDT?

a. S-attributes are synthesized attributes, while L-attributes are inherited attributes.

b. S-attributes are inherited attributes, while L-attributes are synthesized attributes.

c. S-attributes are attributes computed in a top-down manner, while L-attributes are attributes computed in a bottom-up manner.

d. S-attributes are attributes associated with non-terminals, while L-attributes are associated with terminals.

Answer: a

21. What does S-attributed SDT stand for?

a. Syntax-Directed Translation with only Synthesized Attributes

b. Syntax-Directed Translation with only Inherited Attributes

c. Syntax-Directed Translation with both Synthesized and Inherited Attributes

d. Syntax-Directed Translation with no attributes

Answer: a

22. In SDT, what is a translation scheme?

a. A set of rules for the lexical analysis phase

b. A set of semantic rules associated with a context-free grammar.

c. A set of rules for error recovery.

d. A set of rules for code optimization.

Answer: b

23. Which attribute is associated with non-terminals in SDT for type checking?

a. Type Attribute

b. Syntax Attribute

c. Code Attribute

d. Token Attribute

Answer: a

24. In SDT, what is the primary purpose of attaching semantic actions to grammar rules?

a. To specify the syntax of the

language

b. To simplify the parsing process

c. To specify the computation of attributes and enforce semantic constraints.

d. To generate machine code directly from the source code.

Answer: c

25. What is the main purpose of syntax-directed translation in a compiler?

a. To generate code or perform other actions based on the syntax of the input program.

b. To parse the input program and create a syntax tree.

c. To perform lexical analysis of the source code.

d. To optimize the generated code.

Answer: a

26. What is the relationship between syntax trees and semantic actions in SDT?

a. Syntax trees are used to define the syntax of a language, while semantic actions are used for parsing.

b. Semantic actions are associated with nodes of the syntax tree and executed during traversal.

c. Semantic actions are used to create syntax trees.

d. Syntax trees and semantic actions are unrelated in SDT.

Answer: b

27. In SDT, what is the difference between a synthesized attribute and an inherited attribute?

a. Synthesized attributes provide information up the tree, while inherited attributes provide information down the tree.

b. Synthesized attributes are always computed at the root of the syntax tree.

c. Inherited attributes are always computed at the leaves of the syntax tree.

d. Synthesized attributes are associated with terminals, while inherited attributes are associated with non-terminals.

Answer: a

28. In SDT, when are semantic actions typically executed?

a. During syntax tree traversal

b. During lexical analysis

c. During syntax analysis

d. During code generation

Answer: a

29. What is a semantic attribute in the context of SDT?

a. An attribute that defines the lexical structure of a language.

b. An attribute associated with a grammar symbol that carries semantic information.

c. An attribute that is used to optimize the compiler's output.

d. An attribute that is only associated with terminals in the grammar.

Answer: b

30. In SDT, what is the main purpose of using attributes?

a. To specify the syntax of a programming language.

b. To compute and store information related to the program's semantics.

c. To generate code directly from the source code.

d. To define the grammar of a programming language.

Answer: b

31. In SDT, what is a semantic rule?

a. A rule that specifies how attributes are computed or inherited for a production.

b. A rule that defines the lexical structure of a language.

c. A rule that specifies the grammar of a language.

d. A rule that checks for syntax errors in the source code.

Answer: a

32. In SDT, what is an attribute?

a. A non-terminal symbol in the grammar.

b. A terminal symbol in the grammar.

c. A value associated with a grammar symbol or a non-terminal in a syntax tree.

d. A production rule in the grammar.

Answer: c

33. In SDT, what is the role of an attribute grammar?

a. To define the lexical structure of a language.

b. To specify the syntax of a programming language.

c. To formally specify how attributes are computed and inherited.

d. To define the grammar of a programming language.

Answer: c

34. Which phase of the compiler generates intermediate code in SDT?

a. Syntax Analysis

b. Code Generation

c. Lexical Analysis

d. Semantic Analysis

Answer: b

35. In SDT, what is the role of a syntax tree?

a. To represent the structure of a program based on its syntax.

b. To store intermediate code.

c. To define the grammar of a programming language.

d. To perform lexical analysis.

Answer: a

36. What is the primary goal of Syntax Directed Translation in compiler design?

a. To define the lexical structure of a language

b. To create a formal grammar for a programming language

c. To attach semantic actions to grammar rules for code generation and analysis.

d. To optimize the execution of the compiled code.

Answer: c

37. In SDT, what is the difference between syntax-directed definitions and syntax-directed translation schemes?

a. Syntax-directed definitions describe attribute values, while translation schemes specify computations.

b. Syntax-directed definitions are used for parsing, while translation schemes are used for code generation.

c. Syntax-directed definitions and translation schemes are synonymous.

d. Syntax-directed definitions specify grammar rules, while translation schemes specify lexical tokens.

Answer: a

38. In SDT, what is a syntax-directed definition?

a. A definition that specifies the grammar of a programming language.

b. A definition that associates attribute values with grammar symbols.

c. A definition that specifies the lexical structure of a language.

d. A definition that defines the grammar of a language.

Answer: b

39. In SDT, what is a syntax-directed translation scheme?

a. A set of rules for parsing a programming language.

b. A set of rules for code optimization.

c. A set of rules that specify attribute computations for a context-free grammar.

d. A set of rules for lexical analysis.

Answer: c

40. What is the main purpose of Syntax Directed Translation in compiler design?

a. To specify the syntax of a programming language.

b. To define how attributes are computed and associated with the syntax tree.

c. To optimize the execution of the compiled code.

d. To create a formal grammar for a programming language.

Answer: b

41. In SDT, which attributes are computed at a node and used by its children?

a. Inherited Attributes

b. Synthesized Attributes

c. Semantic Attributes

d. Terminal Attributes

Answer: b

42. In SDT, which attributes are computed at a node and used by its parent or siblings?

a. Inherited Attributes

b. Synthesized Attributes

c. Semantic Attributes

d. Terminal Attributes

Answer: a

43. In SDT, what is the primary use of inherited attributes?

a. To propagate information from parent to child nodes in the syntax tree.

b. To provide information about a node to its parent in the syntax tree.

c. To define the lexical structure of a language.

d. To check for syntax errors in the source code.

Answer: a

44. In SDT, what is the main purpose of synthesized attributes?

a. To create the abstract syntax tree for the input program.

b. To generate code for the input program.

c. To provide information about a node to its parent in the syntax tree.

d. To store the source code.

Answer: b

45. Which phase of a compiler is responsible for generating intermediate code in Syntax Directed Translation?

a. Syntax Analysis

b. Code Generation

c. Lexical Analysis

d. Semantic Analysis

Answer: b

46. What is the primary goal of attaching semantic actions in SDT?

a. To specify the syntax of a programming language.

b. To create a formal grammar for a programming language

.

c. To specify the computation and propagation of attributes in the syntax tree.

d. To define the lexical structure of a language.

Answer: c

47. In SDT, what is a semantic attribute's role in the translation process?

a. It carries information about the program's semantics.

b. It defines the syntax of the language.

c. It specifies the grammar of the language.

d. It is only used during code generation.

Answer: a

48. What is an example of an inherited attribute in SDT for a programming language?

a. Code Attribute

b. Type Attribute

c. Scope Attribute

d. Syntax Attribute

Answer: c

49. In SDT, what is the primary reason for using attribute grammars?

a. To provide a formal framework for specifying attribute computations.

b. To create the syntax tree.

c. To define the lexical structure of a language.

d. To check for syntax errors in the source code.

Answer: a

50. What is the primary difference between Syntax Directed Translation and abstract syntax trees (ASTs)?

a. Syntax Directed Translation includes both the syntax tree and semantic actions, while ASTs focus only on the tree structure.

b. Syntax Directed Translation and ASTs are the same thing and can be used interchangeably.

c. Syntax Directed Translation generates machine code, while ASTs do not.

d. ASTs are used only in lexical analysis, while Syntax Directed Translation is used in code generation.

Answer: a