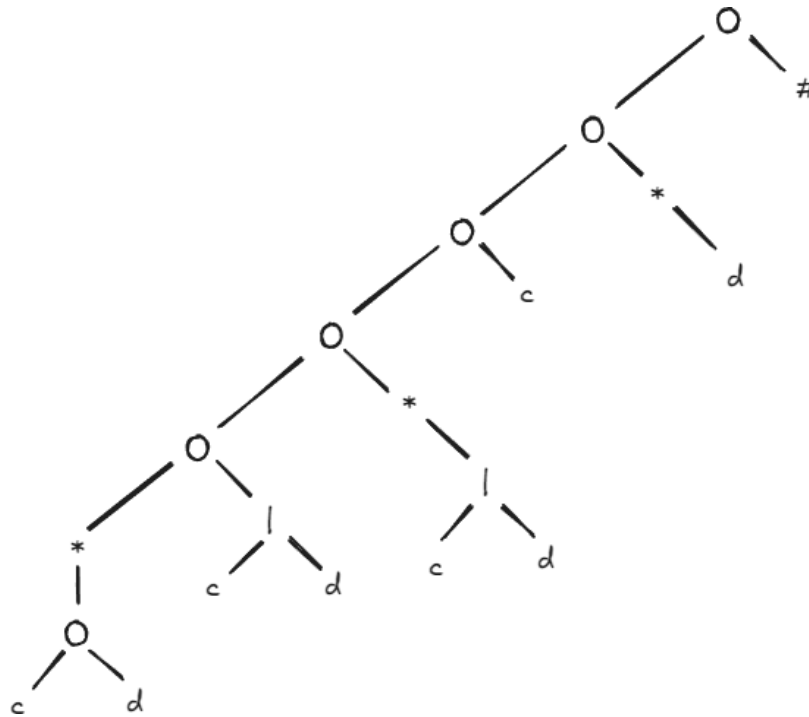


3) Considere a árvore a seguir. Marque a alternativa correta.



- (a) A expressão regular é $(cd)^*(c|d)(c|d)^*cd^*$ e o AFD resultante possui 8 estados e 2 estados finais.
 (b) A expressão regular é $(cd)^*(c|d)(c|d)^*cd^*$ e o AFD resultante possui 7 estados e 3 estados finais.
 (c) A expressão regular é $(cd)^*(c|d)(c|d)^*cd^*$ e o AFD resultante possui 8 estados e 4 estados finais.
 (d) A expressão regular é $c(d)^*(c|d)(c|d)^*cd^*$ e o AFD resultante possui 8 estados e 3 estados finais.
 (e) A expressão regular é $c(d)^*(c|d)(c|d)^*cd^*$ e o AFD resultante possui 7 estados e 4 estados finais.

4) Ainda sobre a árvore da questão anterior, julgue as afirmativas a seguir:

I - O conjunto lastpos com maior soma possível é $\{3,4,5,6\}$.

II - O conjunto followpos que foi o mais repetido é $\{8,9\}$.

III - A minimização do autômato resultante o deixaria com apenas 1 estado final.

IV - No autômato resultante um dos estados finais do é representado por $\{5,6,7,8,9\}$.

V - No autômato resultante o estado inicial é representado por $\{2,5,6,7\}$.

Quais estão corretas? (a) I, II e III (b) I, III e IV (c) I e IV (d) III e V (e) I, III, IV e V

Considere o contexto abaixo para responder às questões 5 e 6:

Você é um desenvolvedor de software trabalhando em um compilador customizado para uma nova linguagem de programação chamada Arcadia. A linguagem possui características únicas, como tipagem dinâmica, suporte a tipos genéricos e um sistema de inferência de tipos avançado. Sua tarefa é garantir que o compilador esteja corretamente implementando a análise léxica, sintática e semântica, assegurando que os programas escritos em Arcadia estejam livres de erros e inconsistências em todos esses níveis.

5) Arcadia permite que identificadores e palavras-chave utilizem caracteres Unicode, oferecendo suporte para diversos alfabetos e símbolos. Durante a fase de análise léxica, o compilador encontra uma sequência de caracteres que pode ser interpretada tanto como um identificador quanto como uma palavra-chave, dependendo do contexto. Como o analisador léxico de Arcadia deve lidar com essa ambiguidade?

- (a) Priorizar a sequência como palavra-chave se ela corresponder a uma das palavras reservadas da linguagem.
 (b) Tratar todas as sequências como identificadores, independentemente de serem palavras-chave.
 (c) Solicitar ao programador que esclareça, no código, se a sequência é uma palavra-chave ou identificador.
 (d) Ignorar a sequência e seguir para o próximo token.
 (e) Ignorar a sequência e seguir para a análise sintática onde os devidos tratamentos serão feitos.

6) Arcadia possui uma sintaxe flexível que permite expressões de ordem superior e uma combinação de estruturas de controle em uma única linha de código. Durante a análise sintática, o compilador encontra uma expressão que pode ser analisada de múltiplas maneiras, dependendo de como os operadores são agrupados. Qual estratégia seria mais apropriada para garantir uma análise sintática correta e consistente em Arcadia?

- (a) Ignorar a ambiguidade e escolher a primeira interpretação possível.
- (b) Deixar a decisão para a fase de análise semântica, onde a lógica do programa será verificada.
- (c) Solicitar ao programador que sempre utilize parênteses para eliminar qualquer ambiguidade.
- (d) Deixar a decisão para a fase de análise semântica, onde a sintaxe e sentido do programa serão verificados.
- (e) Seguir uma ordem de precedência fixa para operadores, garantindo consistência na análise.

Considere o contexto abaixo para responder à questão 7:

Você é um desenvolvedor de software trabalhando em um compilador customizado para uma nova linguagem de programação chamada Arcadia. A linguagem possui características únicas, como tipagem dinâmica, suporte a tipos genéricos e um sistema de inferência de tipos avançado. Sua tarefa é garantir que o compilador esteja corretamente implementando a análise semântica, assegurando que os programas escritos em Arcadia estejam livres de erros semânticos e inconsistências de tipos.

7) Durante a análise semântica em Arcadia, o compilador verifica se as operações realizadas em variáveis e expressões são válidas em termos de tipos. Qual das seguintes tarefas é principalmente realizada durante a análise semântica?

- (a) Geração de código intermediário a partir da árvore sintática.
- (b) Verificação de tipos para garantir que as operações entre variáveis sejam consistentes.
- (c) Verificação de sintaxe para garantir que as operações entre variáveis sejam consistentes.
- (d) Identificação de tokens no código-fonte.
- (e) Construção da árvore sintática a partir da análise gramatical.

8) Usando a tradução das Figuras 6.43 e 6.46, traduza a expressão a seguir. Assuma que o endereço da primeira instrução é 100 e escreva as instruções emitidas (incluindo os remendos). Lembrando que o atributo nextlist do comando S representa todos os desvios condicionais e incondicionais para a instrução seguinte ao comando S. Por fim, marque a alternativa correta:

$a == b \ \&\& \ (c == d \ || \ e == f)$

- (a) A instrução *if e == f* possui um *goto* para 105.
- (b) A instrução *if a == b* possui um *goto* para 103.
- (c) A instrução *if c == d* possui um *goto* vazio na linha 102, seguido por um *goto 104* na linha 103.
- (d) A instrução *if e == f* possui um *goto* vazio na linha 104, seguido por um *goto 106* na linha 105.
- (e) A instrução *if a == b* possui um *goto* 102 na linha 100, seguido por um *goto 103* na linha 101.

9) Para a SDD, a árvore de sintaxe de **$c - 7 + a$** possui quantas folhas e quanto nós?

Produção	Regras Semânticas
$E \rightarrow E1 + T$	$E.node = \text{new Node} ('+', E1.node, T.node)$
$E \rightarrow E1 - T$	$E.node = \text{new Node} ('-', E1.node, T.node)$
$E \rightarrow T$	$E.node = T.node$
$T \rightarrow (E)$	$T.node = E.node$
$T \rightarrow id$	$T.node = \text{new Leaf} (id, id.entry)$
$T \rightarrow num$	$T.node = \text{new Leaf} (num, num.val)$

- (a) 2 folhas e 3 nós. (b) 3 folhas e 3 nós. (c) 2 folhas e 2 nós.
- (d) 3 folhas e 2 nós. (e) 3 folhas e 1 nó.

10) A alternativa que contém as quádruplas ou triplas corretas para traduzir a expressão aritmética $a + -(b + c)$ é:

(a) Quádrupla:

	op	arg1	arg2	res
0	+	b	c	t1
1	minus	t1		t2
2	+	a	t2	t1

(b) Quádrupla:

	op	arg1	arg2	res
0	+	b	c	t1
1	minus	t1		t2
2	-	a	t2	t3

(c) Quádrupla:

	op	arg1	arg2	res
0	+	b	c	t1
1	-	t1		t2
2	+	a	t2	t3

(d) Tripla:

	op	arg1	arg2
0	+	b	c
1	minus	(0)	
2	-	a	(1)

(e) Tripla:

	op	arg1	arg2
0	+	b	c
1	minus	(0)	
2	+	a	(1)

11) Qual é a principal função do analisador léxico em um compilador?

- (a) Gerar código intermediário.
- (b) Verificar a sintaxe do programa.
- (c) Realizar otimizações de código.
- (d) Montar o programa para a linguagem de máquina.
- (e) Identificar e classificar tokens.

12) Qual das seguintes alternativas descreve corretamente um autômato finito determinístico (AFD)?

- (a) Possui um número infinito de estados.
- (b) Permite transições vazias (ϵ -transições).
- (c) Cada estado tem exatamente uma transição para cada símbolo do alfabeto.
- (d) Não possui estados de aceitação.
- (e) Pode ter mais de um estado inicial.

13) O que é uma árvore de derivação (parse tree)?

- (a) Uma árvore que representa a estrutura sintática de um programa.
- (b) Uma representação linear das operações aritméticas.
- (c) Uma estrutura de dados usada para a análise léxica.
- (d) Um grafo que mostra as dependências de dados.
- (e) Um método para otimização de código.

14) Qual das seguintes afirmações sobre expressões regulares (ERs) é verdadeira?

- (a) ERs podem expressar qualquer linguagem livre de contexto.
- (b) ERs são usadas principalmente na análise semântica.
- (c) ERs são úteis apenas para a correspondência exata de strings.
- (d) ERs são usadas para descrever os padrões de tokens na análise léxica.
- (e) ERs não podem ser convertidas em autômatos finitos.

15) Qual é a principal vantagem de converter um autômato finito não-determinístico (AFN) em um autômato finito determinístico (AFD)?

- (a) Aumenta o número de transições.
- (b) Simplifica a construção de expressões regulares.
- (c) Facilita a implementação da análise léxica.
- (d) Permite a inclusão de ϵ -transições.
- (e) Reduz o número de estados no autômato.

16) Durante a análise léxica, como são geralmente identificados os erros?

- (a) Quando o analisador léxico não consegue corresponder uma sequência de caracteres a um token válido.
- (b) Quando o código-fonte contém erros semânticos.
- (c) Quando uma variável não é inicializada.
- (d) Quando o código-fonte tem erros de lógica.
- (e) Quando o analisador sintático não reconhece uma construção gramatical.

17) Qual das seguintes alternativas descreve corretamente um remendo (backpatching)?

- (a) Uma técnica para otimização de código em tempo de execução.
- (b) Um método para lidar com desvios e rótulos durante a geração de código intermediário.
- (c) Um processo de validação semântica de variáveis.
- (d) Um tipo de transformação de código fonte em linguagem de máquina.
- (e) Um processo de análise de dependência de dados.

18) O que define um analisador LR(0)?

- (a) Usa análise top-down com backtracking.
- (b) Não requer uma tabela de parsing.
- (c) Baseia-se em estados e transições para decidir ações.
- (d) Sempre resolve conflitos de parsing automaticamente.
- (e) Funciona melhor para gramáticas LL(1).

19) Qual das seguintes alternativas descreve corretamente atributos sintetizados?

- (a) São usados apenas em análise léxica.
- (b) Podem ser herdados de nós filhos.
- (c) São propagados apenas de filhos para o pai na árvore de derivação.
- (d) São propagados de pai para filho.
- (e) Não são utilizados em árvores de derivação.

20) Qual das seguintes estruturas é usada para representar expressões com operações compartilhadas de forma eficiente?

- (a) Árvore de derivação.
- (b) DAG (grafo acíclico dirigido).
- (c) Quádruplas.
- (d) Tabela de símbolos.
- (e) Pilha de controle.