

Bacharelado em Ciência da Computação
Colegiado de Ciência da Computação
UNIOESTE Campus de Cascavel - CCET
Disciplina: Compiladores
Série: 4
Data: 19/03/2021

1ª Avaliação de Compiladores

Instruções:

- A Prova contém 8 questões, sendo 6 de múltipla escolha e 2 questões abertas.
- Você deve responder as questões e anotar suas respostas no gabarito do arquivo **gabarito.odt**.
- Para as questões abertas, você deve enviar a resolução por meio digital, fotografia ou figura.
- Cada questão objetiva tem apenas uma alternativa correta.
- **O que deve ser entregue** (via *tarefa específica da equipe de OAC na plataforma Teams*):
 1. PDF contendo o gabarito e folhas de resolução (gerado a partir de gabarito.odt).
 - Nomeie o arquivo como *gabarito_<Seu_Nome>.pdf*;

OBS: Não jogue fora suas soluções físicas após o encerramento da avaliação.

Boa prova!

Questões Objetivas

Questão 1 [10]– No processo de geração de um executável (linguagem de máquina) a partir de um programa fonte, escrito em linguagem de alto nível, o programa original passa por transformações e análises que são realizadas em diversas fases. De forma simplificada, pode-se dividi-las nas em oito (8) fases apresentadas, em ordem alfabética, a seguir:

- (A) Alocação de Registradores
- (B) Análise Léxica
- (C) Análise Sintática
- (D) Emissão de Código Assembly
- (E) Link Edição
- (F) Montagem
- (G) Seleção de Instruções
- (H) Verificação de Tipos e Símbolos

Assinale a alternativa correta:

- a[☐] A fase G está relacionada à fase de otimização de código.
- b[☐] A ordem correta de execução dessas fases é B, C, A, E, D, H, G, F.
- c[☐] As fases E, F, H fazem parte do *back-end* do compilador
- d[☐] As fases B, C e H fazem parte do *front-end* do compilador.
- e[☐] Nenhuma está correta.

Questão 2 [10] - Analise as seguintes afirmativas sobre a fase de análise (Front-End) de um compilador.

- I. O uso de uma variável de ponto flutuante para indexar um vetor causa um erro geralmente detectado na análise semântica.
- II. Parênteses desbalanceados são um erro geralmente detectado pela análise léxica já que essa fase lê o arquivo fonte e o traduz para uma sequência de tokens.
- III. A análise sintática tem como objetivo validar a gramática do programa, reconhecendo se a estrutura gramatical do código fonte está de acordo com as regras sintáticas da linguagem.
- IV. A sintaxe de uma linguagem de programação é normalmente dada pelas regras gramaticais de uma expressão regular.

A análise permite concluir que:

- a[☐] Apenas I. e II. estão corretas.
- b[☐] Apenas IV. está correta.
- c[☐] I. II. e III. estão corretas.
- d[☐] II. e IV. estão incorretas.
- e[☐] Todas estão incorretas.

Questão 3 [10] - A tarefa principal de um analisador léxico consiste em ler os caracteres da entrada do programa-fonte, agrupá-los em lexemas e gerar uma sequência de tokens. Sobre o analisador léxico, analise as assertivas abaixo:

- I. Além da identificação de lexemas, outras tarefas podem ser realizadas por esse analisador, tais como: remoção de comentários e espaços em branco e a associação de mensagens de erros às linhas do programa-fonte.
- II. A sequência de tokens produzida pelo analisador léxico é utilizada como entrada pelo módulo seguinte do compilador, o analisador semântico.
- III. Expressões regulares e geradores de analisadores léxicos são notações utilizadas para especificar os padrões de lexemas.
- IV. Na análise léxica, uma representação intermediária do tipo árvore é criada. Esta apresenta a estrutura gramatical da sequência de tokens.

Quais estão corretas?

- a[☐] I.
- b[☐] I. e III
- c[☐] II.
- d[☐] II. III e IV
- e[☐] I. e IV

Questão 4 [10] - Analise as seguintes afirmativas sobre os analisadores sintáticos Top-Down.

- I. São fáceis de implementar para linguagens cuidadosamente projetadas, porém geralmente exigem transformações em gramáticas originalmente apresentadas em BNF.
- II. Um dos principais problemas desse tipo de parser é a necessidade de retrocesso nas alternativas, o que pode ser resolvido com o uso de um parser recursivo preditivo.
- III. Para evitar os problemas do parser descendente recursivo, podemos realizar a análise top-down usando um parser preditivo não recursivo, ou parser preditivo tabular. O parser preditivo tabular usa uma tabela baseada nos conjuntos first e follow para decidir qual produção aplicar à entrada.

A análise permite concluir que:

- a[☐] Apenas I. está correta.
- b[☐] Apenas I. e II. estão corretas.
- c[☐] II. e III. estão corretas.
- d[☐] Nenhuma está correta.
- e[☐] Todas estão corretas.

Questão 5 [10] - Considere a gramática das expressões a seguir.

$S \rightarrow E$
 $E \rightarrow E + T$
 $E \rightarrow T$
 $T \rightarrow T * F$
 $T \rightarrow F$
 $F \rightarrow id$
 $F \rightarrow (E)$

Sobre essa gramática, considere as afirmativas a seguir:

- I. A gramática é LL(1).
- II. O operador + possui uma precedência igual ao operador *.
- III. Não é possível construir um analisador descendente recursivo para a gramática.
- IV. Os terminais + *) \$ pertencem ao conjunto follow de F.

Assinale a alternativa correta.

- a[] Apenas I. está correta.
- b[] Somente I. e II. estão corretas.
- c[] Somente III. e IV. estão corretas.
- d[] Somente II. e IV. estão corretas.
- e[] Todas estão corretas.

Questão 6 [1,0] - Considere a gramática livre de contexto G abaixo:

$X \rightarrow aZbXY \mid c$
 $Y \rightarrow dX \mid \epsilon$
 $Z \rightarrow e$

Considere ainda, a tabela LL(1) construída a partir da gramática G. Essa tabela possui duas produções distintas na célula (Y, d), gerando no analisador sintático uma dúvida na escolha da regra de produção a ser aplicada em determinado momento da análise.

	a	b	c	d	e	\$
X	$X \rightarrow aZbXY$		$X \rightarrow c$			
Y				$Y \rightarrow dX$ $Y \rightarrow \epsilon$		$Y \rightarrow \epsilon$
Z					$Z \rightarrow e$	

Considerando que a tabela corretamente construída, a existência de duas regras de produção distintas na célula (Y, d) resulta:

- a[] da ausência do símbolo de fim de cadeia (\$) nas regras de produção.
- b[] da presença de duas regras com um único terminal do lado direito da produção.
- c[] do uso incorreto da cadeia vazia nas regras de produção.
- d[] de um não-determinismo causado por uma ambiguidade na gramática.
- e[] de uma gramática não LL(1).

Questões Abertas

Questão 7 [2,0] - Considere os tokens:

- Números inteiros sem sinal
- Números reais sem sinal, com dígitos obrigatórios antes e após o "."
- Operadores aritméticos: +, -, /, *, ++
- Operadores relacionais: >, <, >= e <=
- Operador de atribuição: =
- Identificadores, para os quais são válidas apenas letras

a) Escreva as regras de reconhecimento, usando expressões regulares, para reconhecer os tokens acima.

b) Baseado nas expressões geradas, crie o autômato geral de reconhecimento.

Questão 8 [2,0] - Dada a gramática a seguir:

$S \rightarrow A)$

$A \rightarrow CB$

$B \rightarrow ;XB \mid \epsilon$

$C \rightarrow Y(X$

$X \rightarrow e$

$Y \rightarrow a$

a) Construa a tabela preditiva, descrevendo os passos realizados;

b) Usando a tabela, mostre passo-a-passo se a cadeia "a(e;e)" é aceita pela linguagem.