Resolução da prova de Compiladores

1. (2,0) A respeito das fases do compilador responda:
2. Qual o produto da Análise Sintática?

R: Ela agrupa os tokens fornecidos pela analise léxica em estruturas sintáticas, construindo a árvore sintática correspondente. Também tem como produto a detecção de erros sintáticos.

1. O que é um erro semântico? Cite um exemplo.

R: É um erro relacionado ao contexto da linguagem, um exemplo: Considerando o seguinte trecho de código em linguagem C, “x = 2;”, Se a variável x não tiver sido declarada ou até mesmo tiver sido declarada com um tipo char o compilador retornará um erro semântico.

1. Descreva o funcionamento do SCANNIG.

O analisador léxico separa o conjunto de caracteres que representa o programa fonte em entidades ou tokens símbolos básicos da linguagem.

1. Quais os dois princípios fundamentais da compilação?

R: O primeiro diz que o compilador deve preservar o significado do programa a ser compilado. O segundo princípio fala que o compilador deve melhorar o programa de entrada de alguma forma perceptível.

1. Qual a diferença entre um compilador e um interpretador. Cite vantagens e desvantagens de cada um.

R: O compilador analisa o código fonte a fim de encontrar erros e ao fim gera um código executável pela máquina, já o interpretador apenas interpreta o código fonte e executa suas instruções.

1. Sabe-se que o objetivo do compilador é transformar um programa fonte em um programa alvo. Sendo assim, para que serve o código intermediário em um compilador.

R: Serve para ter um reaproveitamento das análises que foram feitas até então, ou seja, reaproveita as fazes e consequentemente possibilita o código ter uma maior portabilidade.

1. (1,0) A respeito das linguagens responda:
2. O que é uma Linguagem Formal?

R: Uma linguagem é considerada formal quando pode ser representada através de um sistema com sustentação matemática.

1. Para que serve uma Gramática?

R: Uma gramática define uma estrutura sobre um alfabeto de forma a permitir que apenas determinadas combinações de símbolos sejam consideradas sentenças.

1. De acordo com a Chomsky, como as gramáticas podem ser classificadas?

R: Gramáticas com estrutura de fase, Gramática sensível ao contexto, Gramática livre de contexto e Gramáticas Regulares.

1. (1,0) Como podemos classificar a gramática abaixo. Qual a linguagem denotada por ela?

G = ({S, A, B}, {a, b}, P, S)

P: S AB

A aA | a

B bB | b

R: É uma gramática livre de contexto. Seja G = (T, N, P, S), L(G) = {W | W pertence a T+ ^ w = (anbm)}

1. (1,0) Identifique os **lexemas** e os **tokens** do seguinte trecho de código:

if A > B then max:=A

R:

|  |  |
| --- | --- |
| Lexemas | tokens |
| If | Palavra reservada, if |
| A | Id1, A |
| > | Símbolo maior que, > |
| B | Id2, B |
| Then | Palavra reservada, Then |
| Max | Variável1, Max |
| := | Atrib, := |
| A | Id3, A |

1. (2,0) Construa um Diagrama de Transição (autômato finito) para:
2. Identificadores em Pascal (Letra seguida por Letras ou Dígitos);

[a-z A-Z]

[a-z A-Z 0-9]

1. Números Reais
2. Operadores relacionais
3. Operadores Matemáticos
4. Integre todos os diagramas descritos em um único diagrama.
5. (3,0) Baseado no diagrama de transições criado na questão anterior, codifique um Analisador Léxico (em Lex) que identifique:
6. Espaços em branco, tabulações, parágrafos e não retorne nada.
7. Identificadores e números reais (sem sinal) e retorne o lexema e o token.
8. Operadores Relacionais e retorne o operador e o token.
9. Operadores Matemáticos e retorne operador e o token.

espaço (“ “ | “/t” | “/n”)

num ([0-9])

letra ([a-zA-Z])

opeR (“<” | “=” | “<” | “>=” | “<=” | “<>”)

oprM (“+” | “-“ | “\*” | “/”)

%%

{espaço} {}

{letra} ({letra} | {num})**\*** {printf(“Lexema = %s”, yytext, “Token = Identificador“);}

{num}**+**,{num}**+**  {printf(“Lexema = %s”, yytext, “Token = Numero Real”);}

{opeR} {printf(“Token = Operador Relacional, %s”, yytext);}

{opeM} {printf(“Token = Operador Matemático, %s”, yytext);}

.\* {printf(“ERRO!”);}

%%

Main () {

yylex();

}