

Slide 1

Métodos de Implementação:

- **Compilação:** Programas são traduzidos para linguagem de máquina. Inclui Sistema JIT.

Programa-fonte → Analisador Léxico → Analisador Sintático →

Gerador de Código Intermediário & Analisador Semântico →

Gerador de Código → Computador → Resultado.

- **Interpretação Pura:** Programas são interpretados por outros programas (Interpretadores).

Programa-fonte → Interpretador → Resultado.

- **Híbrido:** Compromisso entre compiladores e interpretadores puros.

Programa-fonte → Analisador Léxico → Analisador Sintático →

Gerador de Código Intermediário → Interpretador → Resultado.

Sintaxe e Semântica:

- **Sintaxe:** Descrição precisa das expressões, sentenças e unidades do programa.
 - As regras sintáticas definem quais sentenças formadas a partir do alfabeto fazem parte da linguagem.
- **Semântica:** Significado dessas expressões, sentenças e unidades do programa.
 - A Semântica deve seguir a Sintaxe, ou seja, a aparência de uma sentença deve sugerir o que ela faz.

- **Lexema:** Palavra (ou parte dela) que serve de base ao sentido que ela expressa.

Os lexemas são divididos em grupos:

- **Token:** Nome do grupo de lexemas ou categoria

Lexema	Token
valor	identificador
=	sinal-igualdade
2	literal-inteiro
*	operador-multiplicação
quantidade	identificador
+	operador-adição
15	literal-inteiro
;	ponto-virgula

- **Reconhecedores:** lê as strings e decide se fazem parte ou não da linguagem. Exemplo: Análise Sintática.
- **Geradores:** Dispositivo que gera sentenças de uma linguagem.

Fundamentos da BNF:

- **LHS:** É a abstração definida;
- **RHS:** É a definição da abstração;
- **Regra ou Produção:** É a representação completa (LHS ou RHS).
- **Símbolos não-terminais:** Abstrações;
- **Símbolos terminais:** Lexemas e Tokens;

Slide 2

Gramáticas ambíguas

- Uma gramática é ambígua quando sua árvore sintática(parse tree) pode correr mais de 1 caminho.

Gramática ambígua

`<expressao> -> <expressao> + <expressao> | const`

Gramática não ambígua

`<expressao> -> <expressao> + const | const`

BNF Estendida (EBNF)

- **Partes Opcionais** são colocadas entre parênteses.
- **Partes Alternativas** são colocadas dentro de parênteses e separadas por barra.
- **Repetições** são colocadas entre chaves

Gramática de Atributos

ENTENDI NADA

Semântica

Semântica Operacional: Descreve o significado de uma sentença especificando os efeitos de executá-la em uma máquina.

Semântica denotacional: Método formais mais rigoroso para descrever o significado de programas.

Define objetos matemáticos para cada entidade da linguagem e então define uma função que mapeia instâncias das entidades para instâncias dos objetos matemáticos correspondentes.

Semântica axiomática: É baseada na lógica matemática. Ela especifica o que pode ser provado sobre o programa.

Axiomas são definidos para cada tipo de declaração da linguagem para permitir transformações de expressões lógicas em expressões lógicas mais formais. As expressões lógicas são chamadas de asserções (*assertions*).

Slide 3

- A parte de análise sintática de um processador de linguagem quase sempre consiste em duas partes:
 - Parte de baixo-nível (analisador léxico)
 - Parte de alto-nível (analisador de sintaxe ou parsing)
- Razões para separar análise léxica e sintática:
 - Simplicidade (Torna mais simples as análises)
 - Eficiência (Permite Otimizar o analisador léxico)
 - Portabilidade

Análise Léxica

- É um identificador de padrões para caracteres de uma string
- identifica **lexemas** e os associa aos **tokens**.
- Abordagens para construir um analisador léxico:
 - Tabela gerada a partir de uma descrição formal;
 - Diagrama de estados que descreve tokens;
 - Os 2 Juntos.

‘Basicamente, identifica o que as coisas são
(exemplo: ‘**A**’ é um identificador / ‘**+**’ é um operador)’

Parsing ou Analisador de Sintaxe

- Objetivos do Parser:
 - Encontrar todos os erros de sintaxe;

- produzir um diagnóstico dos erros;
 - produzir uma árvore de análise sintática (ou seja, vai analisar se o programa está feito de forma que faz sentido).
-
- Duas categorias de Parser:
 - **Top Down:** Produz a 'parser tree' começando pela raiz (Derivação à esquerda)
 - **Bottom Up:** Produz a 'parser tree' começando pelas folhas (Derivação à direita)

Slide 4

Linguagens Imperativas:

- Variável é abstração de uma célula da memória
- Características das variáveis:
 - Nome
 - Endereço
 - Tipo
 - Valor
 - Tempo de vida (vinculação)
 - Escopo

Nomes

- Pode ser chamado de identificador
- **Tamanho:** cadeia de caracteres usada para identificar alguma entidade do programa
- **Caracteres Especiais:** Em algumas linguagens deve-se iniciar o nome de uma variável com um certo caractere especial (às vezes, desse caractere define o tipo da variável).
- **Case Sensitive:** Caixa alta ou baixa muda alguma coisa?
- **Palavras Especiais:** Em algumas linguagens, existem palavras que não podem ser usadas como nome (Palavras-Reservadas).

Endereço

- Endereço da memória no qual a variável é associada
- Uma variável pode ter diferentes endereços em diferentes lugares do programa.
- Duas variáveis podem acessar o mesmo endereço = Aliases

Tipo e Valor

- **Tipo:** Determina o intervalo de valores que a variável pode armazenar .
- **Valor:** Conteúdo da variável.

Vinculação e Tempo de Vida

- **Vinculação:** É a associação entre um atributo e uma entidade (exemplo: variável e seu tipo)
- **Tempo de vida:** Tempo entre a alocação e desalocação da memória.
- Principais tempos que a vinculação pode ocorrer:
 - Tempo de Compilação
 - Tempo de Execução

Slide 5

Vinculação

- **Vinculação Estática:** Ocorre em tempo de Compilação.
- **Vinculação Dinâmica:** Ocorre em tempo de Execução (ou seja, pode ser alterada durante a execução).
 - Vantagem: flexibilidade na programação;
 - Desvantagem: dificulta na hora de achar erros.

Categorias de Variáveis:

- **Variáveis Estáticas**
- **Variáveis Dinâmicas de Pilha**
- **Variáveis Dinâmicas de Heap**
- **Variáveis Dinâmicas de Geap**

Escopo

- Área onde a variável é visível (ser visível significa que é possível 'chamar' ou 'utilizar' a variável).
- Tipos de Escopo:
 - **Escopo Estático**
 - **Escopo Dinâmico**

ATENÇÃO:

Saber fazer:

- Parser Tree (árvore do ngc lá)
- Derivar a linguagem (fazer tipo uma BNF lá)

PERGUNTAS

1) Quais são os 3 métodos de implementação de um código?

R: - Compilação (código fonte → analisador léxico → analisador sintático → gerador de código intermediário e analisador semântico → gerador de código → computador → resultado).

- Interpretação (código-fonte → interpretador → resultado).
- Híbrido (código-fonte → analisador léxico → analisador sintático → gerador de código intermediário → interpretador → resultado).

2) Qual a diferença entre Sintaxe e Semântica?

R: Sintaxe define quais palavras fazem parte da Linguagem, Semântica define o que estas palavras significam ou fazem.

3) O que é Lexema e Token?

R: Palavra que serve de base para o sentido que ela expressa. Lexema é a interpretação de palavras (ou parte delas).

Token é o nome ou categoria do lexema.

4) Descreva os fundamentos da BNF.

R: - LHS: Nome da abstração.

- RHS: o que a abstração significa.
- Regra ou Produção: Representação completa (LHS e RHS).
- Símbolos não-terminais: abstrações.
- Símbolos Terminais: Lexemas ou Tokens.

5) Dê um exemplo de gramática ambígua.

R: A gramática ambígua é aquela que pode seguir mais de uma parser tree.

Exemplo de gramática ambígua: $\langle \text{expressão} \rangle = \langle \text{var} \rangle + \langle \text{var} \rangle \mid \langle \text{var} \rangle$

Exemplo de gramática não-ambígua: $\langle \text{expressão} \rangle = \langle \text{var} \rangle + \text{const} \mid \langle \text{var} \rangle$

6) Descreva os três tipos de semântica.

R: - Operacional: Descreve o significado de uma sentença especificando seus efeitos na máquina.

Denotacional: Descreve o significado de uma sentença definindo objetos matemáticos para cada entidade da linguagem.

- ☐ Axiomática: Mostra o que pode ser provado sobre o programa, baseado na lógica matemática.

7) Quais são as razões para se separar o analisador léxico e sintático?

R: - simplicidade

- poder reutilizar o analisador sintático (portabilidade)
- eficiência

8) Quais são as três abordagens para se construir um analisador léxico?

R: - Diagrama de estados que descreve Tokens.

- Tabela Gerada a partir de uma descrição formal
- Os 2 juntos.

9) O que o Parsing faz?

R: Encontra os erros de sintaxe e produz uma parser tree.

10) Quais são as seis características de uma variável em linguagens imperativas?

R: - Nome

- Endereço
- Tipo
- Valor
- Vinculação e Tempo de Vida
- Escopo

11) Quais são os principais tempos em que a vinculação pode ocorrer?

R: - Tempo de Compilação (vinculação estática).

- Tempo de execução (vinculação dinâmica).

12) Quais são os quatro tipos de variáveis?

R: - Variável Estática.

- Variável dinâmica de pilha.
- Variável dinâmica de heap.
- Variável dinâmica de geap.

13) O que é escopo e quais seus tipos?

R: - Escopo de uma variável é a área em que esta variável é visível para o código, existem escopos estáticos e dinâmicos.