



PARADIGMAS E LINGUAGENS DE PROGRAMAÇÃO

ENGENHARIA DA COMPUTAÇÃO – UFC/SOBRAL

Prof. Danilo Alves

`danilo.alves@alu.ufc.br`

INTRODUÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Sintaxe de uma LP: a **forma** de suas expressões, sentenças e unidades de programas
- Semântica de uma LP: o **significado** dessas expressões, sentenças e unidades de programas
- Sintaxe e semântica fornecem uma definição (concisa e compreensível) da linguagem
 - Quem são os usuários de uma definição de linguagem?
 - Desenvolvedores de linguagem
 - Implementadores
 - Programadores (os usuários da linguagem)
- Exemplo da sintaxe **while** da linguagem **Java**

Dificuldade de escrever uma LP

SINTAXE E SEMÂNTICA



UNIVERSIDADE
FEDERAL DO CEARÁ

- while (expressão_booleana) sentença para executar 10 vezes

int i=0;

while i < 10 i++; Erro de sintaxe

while (i < 10) i++; Correto

Quando o valor atual da expressão booleana for verdadeiro, a sentença dentro da estrutura é executada.

Caso contrário, o controle continua após a construção while

O controle retorna implicitamente para a expressão booleana para repetir o processo

while (i <= 10) i++; Erro de semântica

while (i < 10) i++; Correto



- Implementadores devem ser capazes de determinar como as expressões, sentenças e unidades de programas de uma linguagem são formadas, e os efeitos pretendidos quando executadas
- Usuários da linguagem devem ser capazes de determinar como codificar soluções de software ao procurarem em um manual de referência da linguagem

SINTAXE E SEMÂNTICA



UNIVERSIDADE
FEDERAL DO CEARÁ

- Sintaxe: conjunto de regras que determinam quais construções são corretas
- Semântica: descrição de como as construções da linguagem devem ser interpretadas e executadas
- Em Pascal: `a:=b`

Sintaxe: comando de atribuição correto

Semântica: substituir o valor de `a` pelo valor de `b`

- Verificar se existem erros:

```
1. int j=0, conta, V[10]; float i@;  
2. conta = '0'  
3. for (j=0, j<10; j++  
4. {  
5.   V[j] = conta++;  
6. }
```

O compilador tem a responsabilidade de reportar ERROS!

➤ É necessário algum recurso para identificá-los

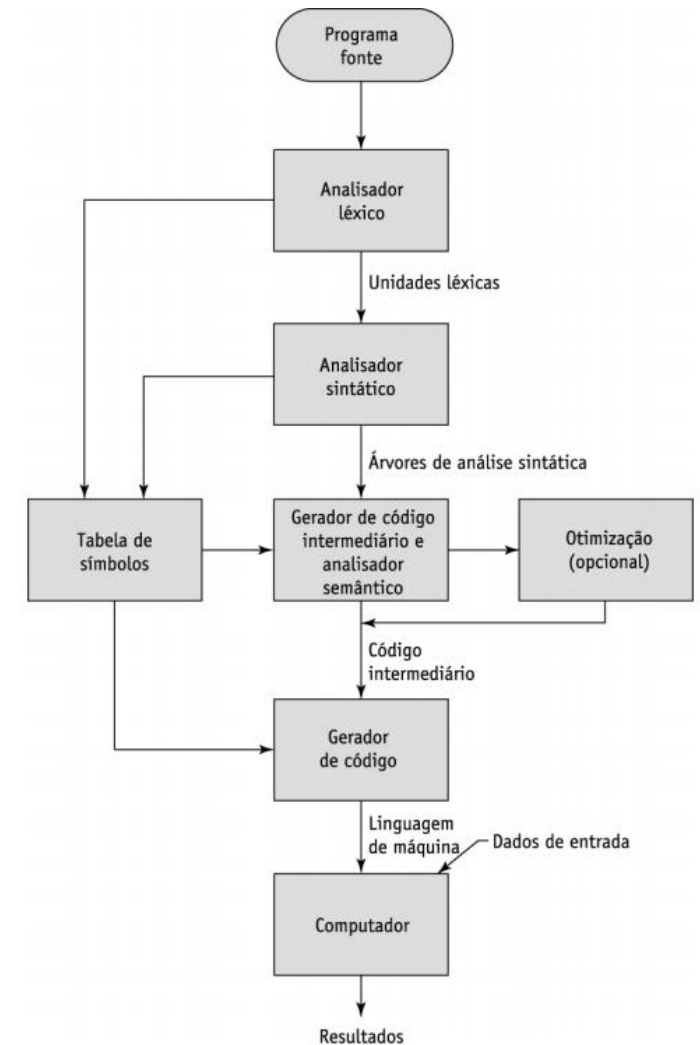
O que diferencia os tipos de ERROS?

ANÁLISE DE COMPILAÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Esses erros são verificados e diferenciados durante a fase de análise da compilação



ANÁLISE DE COMPILAÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Esses erros são verificados e diferenciados durante a fase de análise da compilação
- **léxica:** palavras (tokens) do programa
 - i, j, for, =, (, <, int, ++, conta, V[]
 - Erro: i@
- **sintática:** combinação de tokens que formam o programa
 - comando_for for (expr1; expr2; expr3) {comandos}
 - Erros: ; for(j=0,...
- **semântica:** adequação do uso
 - Tipos semelhantes em comandos (atribuição, por exemplo), uso de identificadores declarados
 - Erro: conta = '0'

ESTRUTURAS DA COMPILAÇÃO

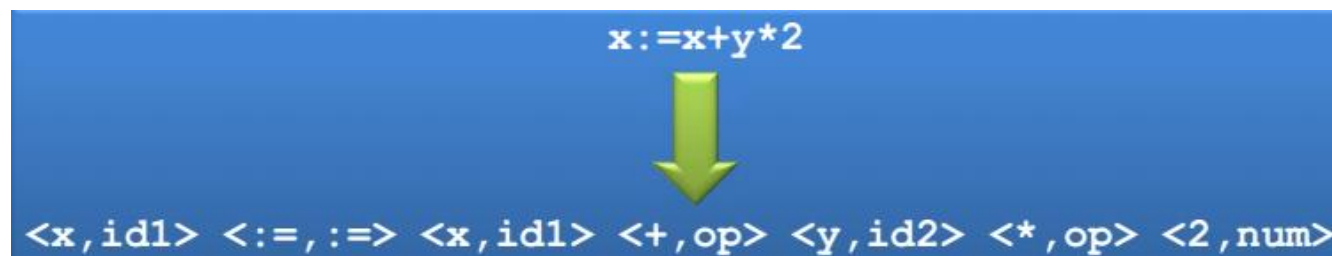


UNIVERSIDADE
FEDERAL DO CEARÁ

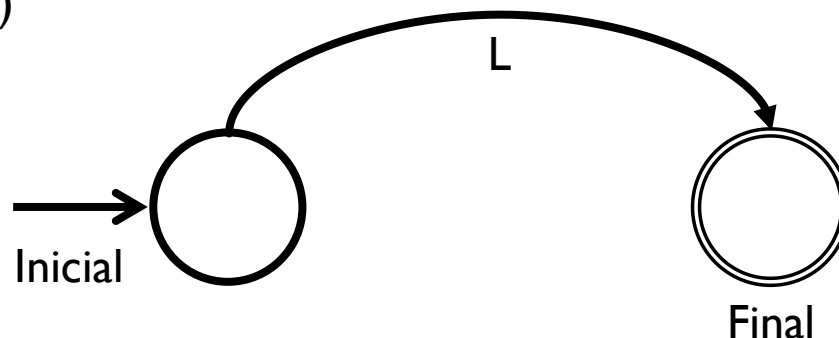
- Como diferenciar as palavras e símbolos reservados (for, int, float, =) de identificadores definidos pelo usuário?
 - Tabela de palavras e símbolos reservados

identificador	Classe	Tipo	Valor	...
j	var	int	0	
fat	function	-	-	
...	

- Análise léxica: atua no reconhecimento e classificação dos tokens (palavras)
 - Expressões regulares e autômatos finito



- AF é uma máquina de estados finitos. Formada por um conjunto de estados (um estado inicial e um ou mais estados finais)

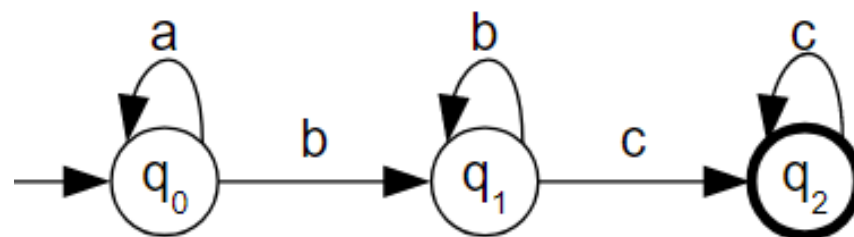
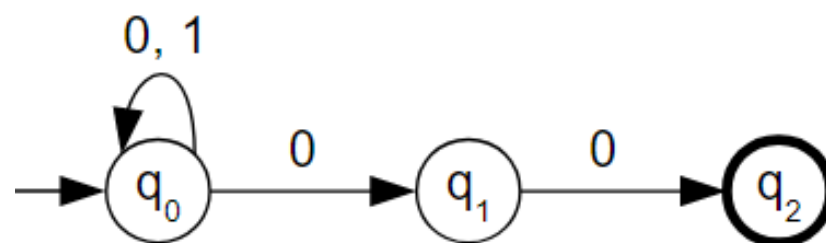


ANÁLISE LÉXICA



UNIVERSIDADE
FEDERAL DO CEARÁ

■ Autômatos

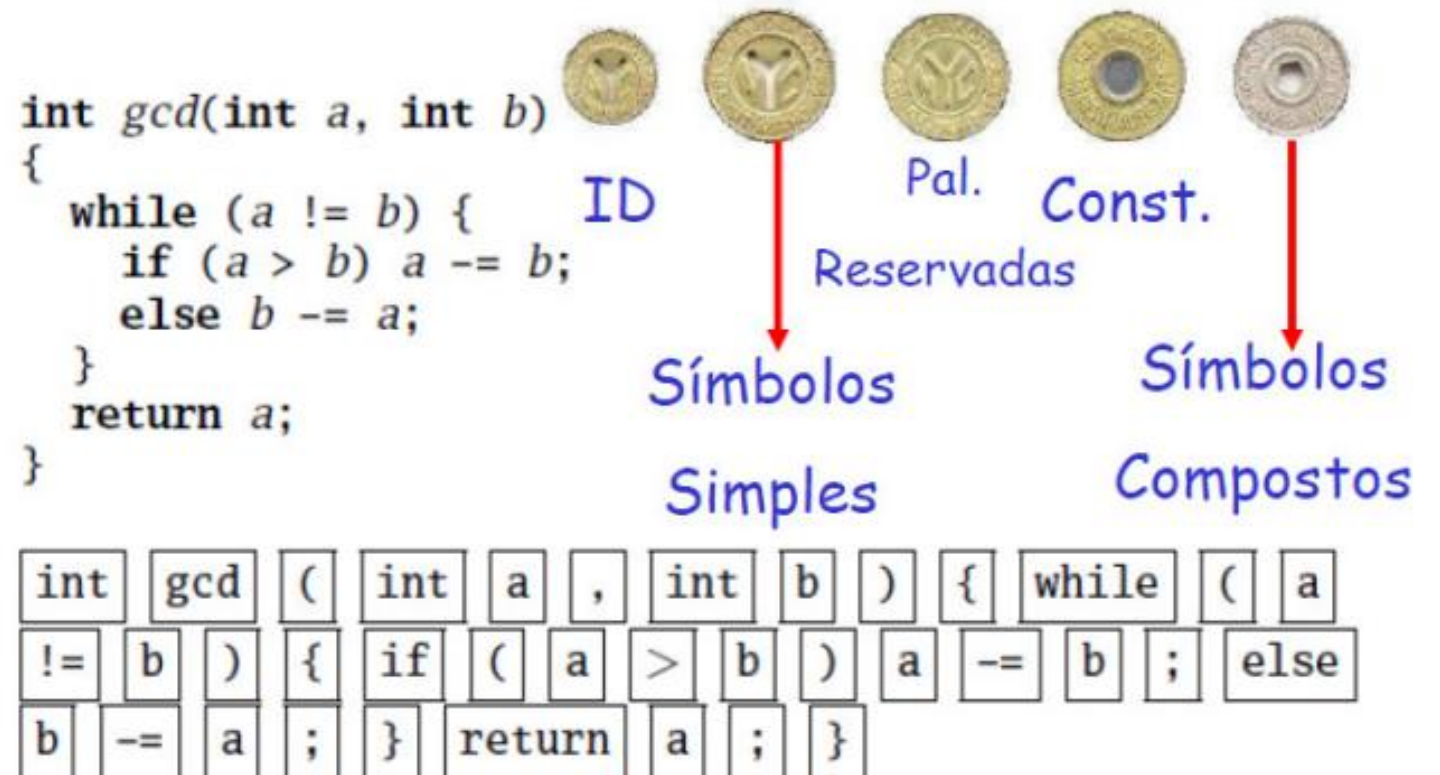


ANÁLISE LÉXICA



UNIVERSIDADE
FEDERAL DO CEARÁ

- **Análise léxica:** A análise léxica (AL) é responsável por ler o código fonte e separá-lo em partes significativas agrupando os caracteres em **lexemas** e produzir uma sequência de símbolos léxicos conhecidos como **tokens**.



- Exemplos de tokens que podem ser reconhecidos em uma linguagem de programação como C

palavras reservadas

if else while do

operadores relacionais

< > <= >= == !=

operadores aritméticos

+ * / -

operadores lógicos

&& || & | !

operador de atribuição

=

delimitadores

;,

caracteres especiais

() [] {}

ANALISADOR SINTÁTICO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Analisador sintático: Verificação da formação do programa
- Gramáticas livres de contexto

`x:=x+y*2`



`<x,id1> <:=, :=> <x,id1> <+,op> <y,id2> <*,op> <2,num>`

`<x, id1> <:=, :=> <x, id1> <+, op> <y, id2> <*, op> <2, num>`



`expressao → id1 := id1 op id2 op num`

ANALISADOR SEMÂNTICO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Analisador semântico: verificação do uso adequado
- A partir da gramática, verifica se os identificadores estão sendo usados de acordo com o tipo declarado

CÓDIGO INTERMEDIÁRIO



UNIVERSIDADE
FEDERAL DO CEARÁ

■ Geração de código intermediário

