

# ATIVIDADE 04/10/2023

## COMPILADORES



FADERGS  
CÊNCIA DA COMPUTAÇÃO

Rafael Leindecker Nunes de Souza

# ANÁLISE LÉXICA

- **Definição e Exemplos**

Análise léxica é o processo de dividir uma sequência de caracteres em unidades significativas, chamadas de tokens. Esses tokens são então usados pelo analisador sintático para analisar a estrutura do programa.

Por exemplo, a seguinte sequência de caracteres:

```
int main() {  
    // O código aqui  
}
```

seria dividida em tokens da seguinte forma:

**int, main, (, ), {, }, //, O, código, aqui, }**

Os tokens podem ser divididos em duas categorias principais:

Tokens léxicos são tokens que representam elementos da linguagem de programação, como identificadores, operadores, palavras-chave e constantes.

Tokens de pontuação são tokens que representam elementos de pontuação, como vírgulas, pontos e vírgulas, colchetes, chaves e parênteses.

- **Especificação dos tokens:**

A especificação dos tokens é um processo de definir as regras que determinam quais sequências de caracteres são tokens válidos. Essas regras geralmente são expressas em forma de expressões regulares.

Por exemplo, a seguinte expressão regular especifica os tokens léxicos de uma linguagem de programação que usa identificadores, operadores e números:

`[a-zA-Z_][a-zA-Z0-9_]* | [+*/%] | [0-9]+`

Essa expressão regular especifica que os identificadores podem ser formados por qualquer combinação de caracteres alfabéticos, numéricos ou sublinhados. Os operadores podem ser qualquer um dos quatro símbolos matemáticos básicos, e os números podem ser qualquer sequência de dígitos.

- **Reconhecimento dos tokens**

O reconhecimento de tokens é o processo de identificar os tokens em uma sequência de caracteres. Isso é feito usando um analisador léxico, que é um programa que implementa as regras de especificação dos tokens.

Os analisadores léxicos geralmente são implementados usando um algoritmo chamado escaner. O scanner varre a sequência de caracteres caractere a caractere, usando as regras de especificação dos tokens para identificar os tokens.

- **Conceituar token**

Token é uma sequência de caracteres que tem um significado especial em uma linguagem de programação. Os tokens podem ser divididos em duas categorias principais:

Tokens léxicos são tokens que representam elementos da linguagem de programação, como identificadores, operadores, palavras-chave e constantes.

Tokens de pontuação são tokens que representam elementos de pontuação, como vírgulas, pontos e vírgulas, colchetes, chaves e parênteses.

Por exemplo, a seguinte sequência de caracteres:

```
int main() {  
    // O código aqui  
}
```

- possui os seguintes tokens:

**Tokens léxicos:**

**int**

**main**

**(**

**)**

**{**

**;**

**//**

Tokens de pontuação:

**,**

**.**

}

## **2. Identificar o melhor formalismo para especificar um token**

O melhor formalismo para especificar um token é a expressão regular. Expressões regulares são uma forma compacta e poderosa de descrever sequências de caracteres. Elas são usadas em uma variedade de aplicações, incluindo análise léxica, processamento de linguagem natural e segurança da informação.

Por exemplo, a seguinte expressão regular especifica os tokens léxicos de uma linguagem de programação que usa identificadores, operadores e números:

**`[a-zA-Z_][a-zA-Z0-9_]* | [+*/%] | [0-9]+`**

Essa expressão regular especifica que os identificadores podem ser formados por qualquer combinação de caracteres alfabéticos, numéricos ou sublinhados. Os operadores podem ser qualquer um dos quatro símbolos matemáticos básicos, e os números podem ser qualquer sequência de dígitos.

Outros formalismos que podem ser usados para especificar tokens incluem:

Lista de palavras-chave: Listas de palavras-chave são um conjunto de palavras que têm um significado especial em uma linguagem de programação.

Tabelas de símbolos: Tabelas de símbolos são estruturas de dados que mapeiam nomes de tokens para seus significados.

## **3. Analisar as estratégias para reconhecimento dos tokens**

Existem duas estratégias principais para reconhecimento dos tokens:

Analizador léxico descendente: O analisador léxico descendente varre a sequência de caracteres caractere a caractere, usando as regras de especificação dos tokens para identificar os tokens.

Analizador léxico ascendente: O analisador léxico ascendente começa com um token inicial e, em seguida, usa as regras de especificação dos tokens para identificar os tokens subsequentes.

O analisador léxico descendente é o método mais comumente usados. Ele é eficiente e fácil de implementar. O analisador léxico ascendente é menos comumente usado, mas pode ser mais eficiente em casos de certas linguagens de programação.