Compiladores

Aula 03 – Análise Léxica (Implementação)

Edirlei Soares de Lima
<edirlei.lima@universidadeeuropeia.pt>

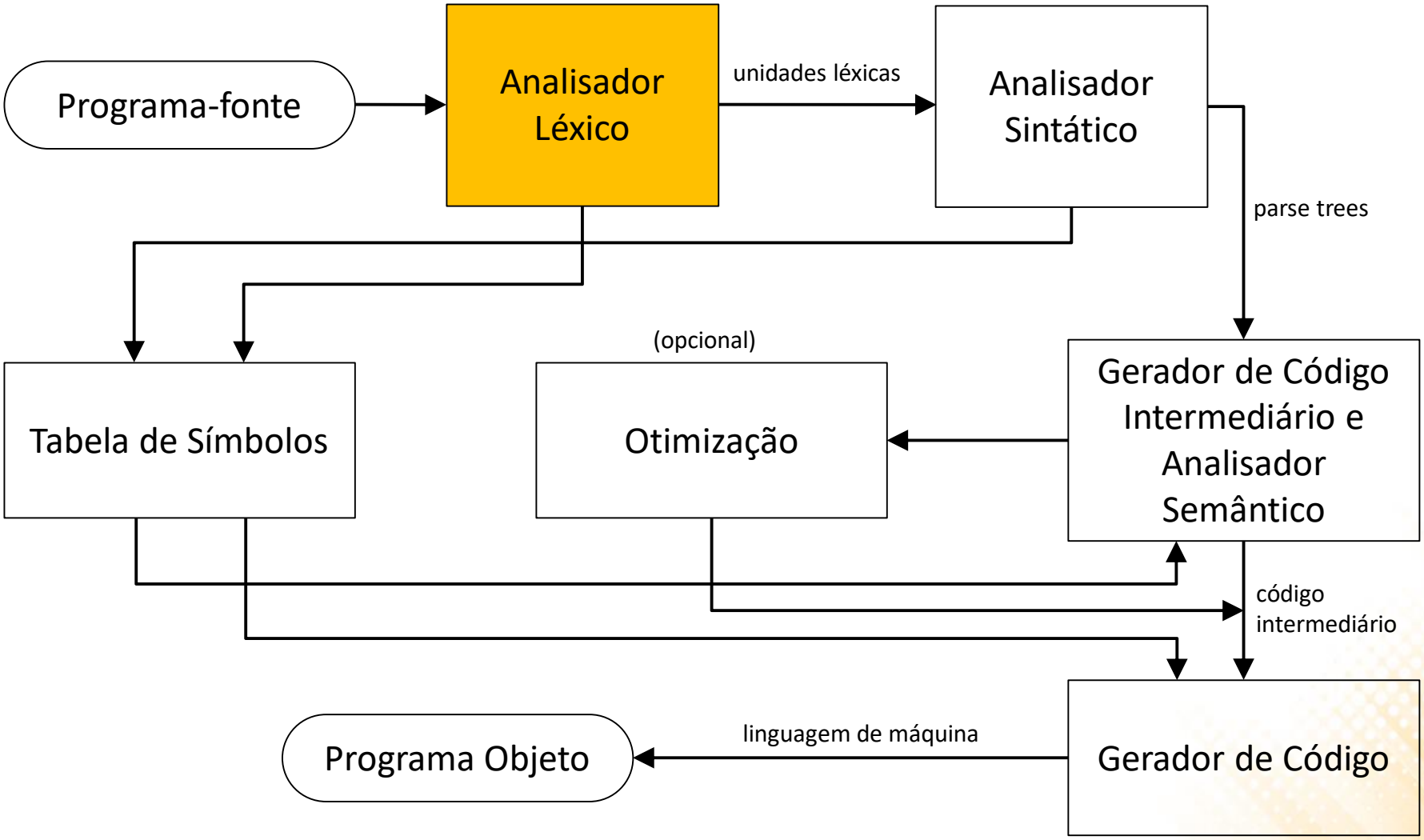
Análise Sintática

- A maioria dos compiladores separam a tarefa da **análise sintática** em duas partes distintas:
  - Análise Léxica;
  - Análise Sintática;

- O **analisador léxico** trata as construções de pequena escala da linguagem (nomes, literais numérico, símbolos...);

- O **analisador sintático** trata as construções de larga escala da linguagem (instruções, expressões, unidades do programa...);

Processo de Compilação



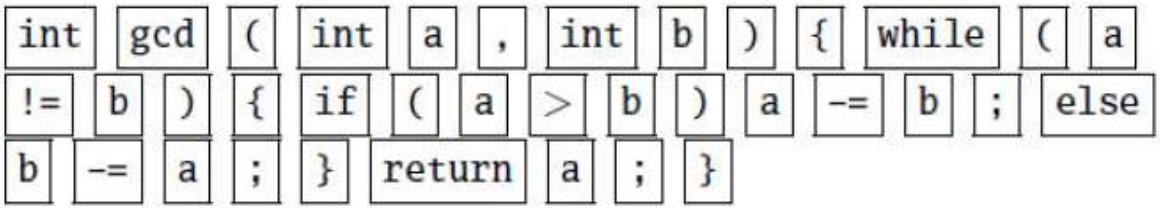
Análise Léxica

- Motivos para separar a análise léxica da análise sintática:
  - **Simplicidade**: as técnicas para a análise léxica são menos complexas do que as exigidas para a análise sintática;
  - **Eficiência**: a separação facilita a otimização do analisador léxico;
  - **Portabilidade**: o analisador léxico lê arquivos de entrada contendo o código fonte do programa, sendo assim ele é de certa forma dependente da plataforma. Já o analisador sintático pode ser independente da plataforma.

Análise Léxica

- Um analisador léxico é essencialmente um **verificador de padrões**.
- Um programa de entrada para um compilador é visto como uma **única cadeia de caracteres**.
- O analisador léxico **coleta os caracteres em agrupamentos lógicos** e atribuir códigos internos para os agrupamentos de acordo com a sua estrutura.

```
int gcd(int a, int b)
{
  while (a != b) {
    if (a > b) a -= b;
    else b -= a;
  }
  return a;
}
```



Descrevendo a Sintaxe

- Os agrupamentos de caracteres são chamados de **lexemas** e os códigos internos são chamados de **tokens**.

- Exemplo:

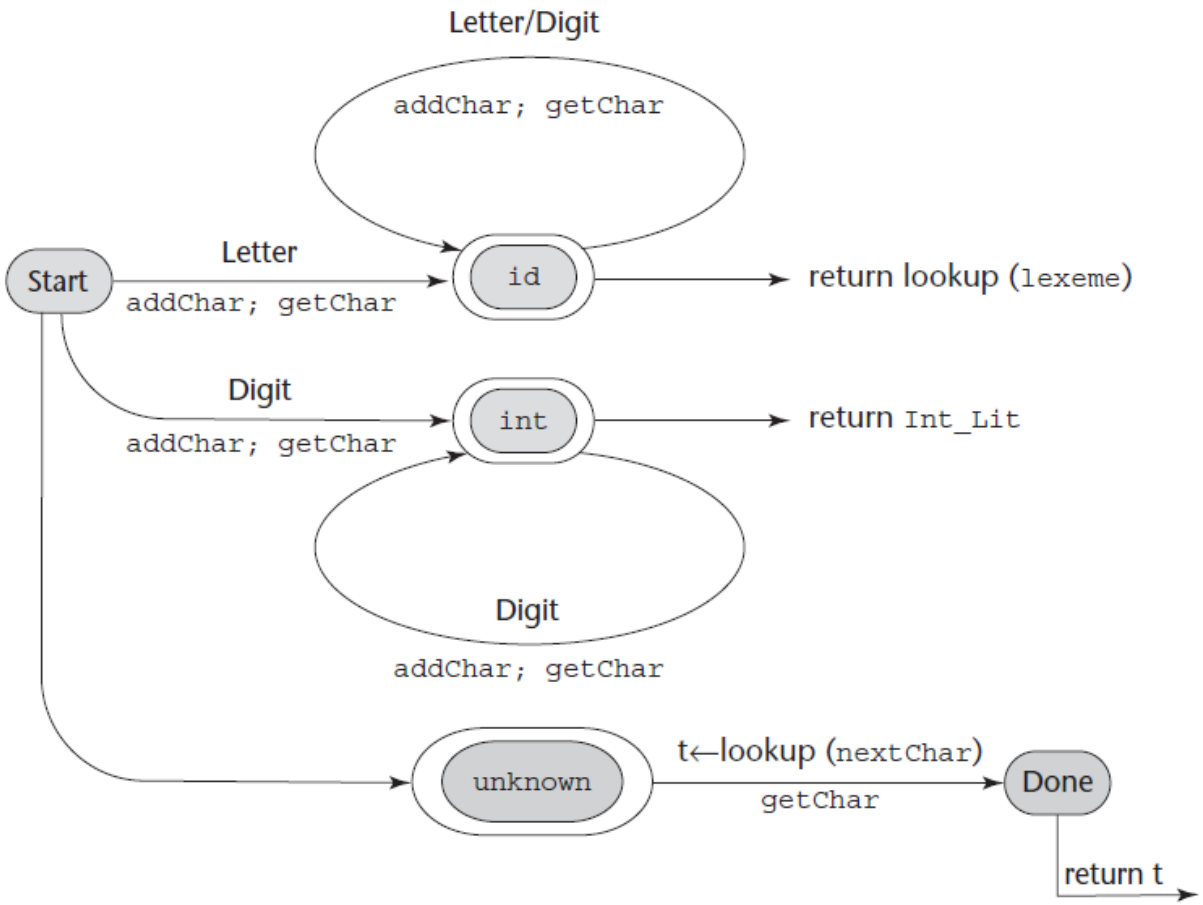
```
index = 2 * count + 17;
```

Lexemas	**Tokens**
index	IDENTIFICADOR
=	OP_ATRIBUICAO
2	INT_LITERAL
*	OP_MULTIPLICAÇÂO
cont	IDENTIFICADOR
+	OP_SOMA
17	INT_LITERAL
;	PONTO_E_VIRGULA

Análise Léxica

- Existem duas abordagens para se construir uma analisador léxico:
  - Escrever uma descrição formal dos padrões de símbolos da linguagem utilizando uma linguagem de descrição relacionada com **expressões regulares** e usar um software para gerar automaticamente o analisador. Exemplo: lex;
  - Construir **autômatos finitos** para descrever os padrões de símbolos da linguagem e escrever um programa para implementar e utilizar os autômatos.
    - Tabelas de transição ou diretamente em código.

Implementação Direta em Código



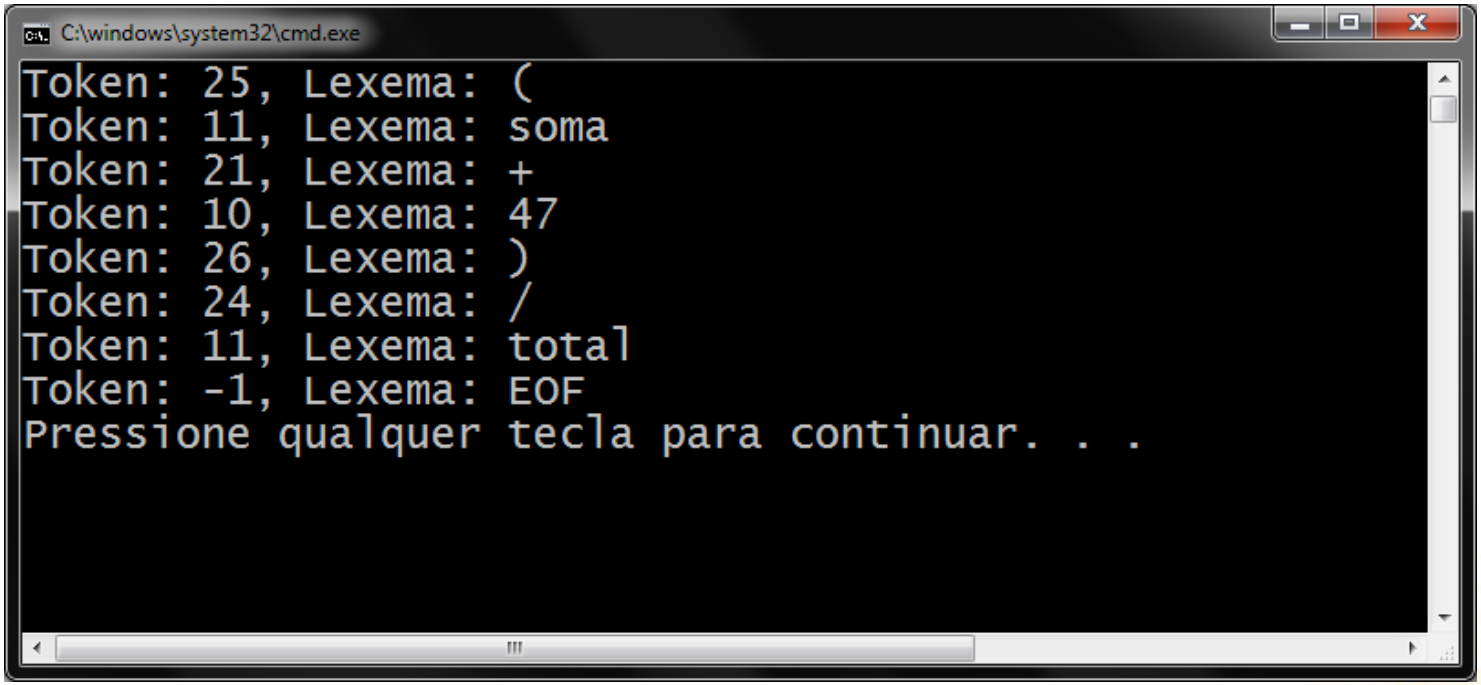
Implementação Direta em Código

```
...
*nextChar = getNonBlank(codefile, *nextChar);
switch (nextChar->type){
  case LETTER:
    tamLexema = addChar(nextLexeme, tamLexema, nextChar->value);
    *nextChar = getChar(codefile);
    while ((nextChar->type == LETTER) || (nextChar->type == DIGIT))
    {
      tamLexema = addChar(nextLexeme, tamLexema, nextChar->value);
      *nextChar = getChar(codefile);
    }
    nextToken = IDENT;
    break;
  case DIGIT:
    tamLexema = addChar(nextLexeme, tamLexema, nextChar->value);
    *nextChar = getChar(codefile);
    while (nextChar->type == DIGIT)
    {
      tamLexema = addChar(nextLexeme, tamLexema, nextChar->value);
      *nextChar = getChar(codefile);
    }
    nextToken = INT_LIT;
    break;
    ...
```

Implementação Direta em Código

- Entrada:

```
(soma + 47) / total
```



Exercício 01

1) Continue a implementação do analisador léxico desenvolvido em aula, acrescentando as seguintes características:

   a) Reconhecer e identificar números decimais (separados por ponto). Exemplo:

```
(soma + 87.2) / total
```

   b) Reconhecer e identificar expressões de atribuição. Exemplo:

```
resultado = (soma + 87.2) / total
```

   c) Reconhecer e identificar programas compostos por mais de uma expressão de atribuição (separadas por ponto e virgula). Exemplo:

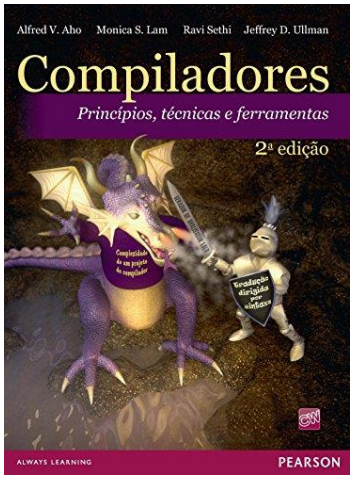
```
a = 8 + 42.75;
b = a * 0.8;
c = (a - b) / 2;
```

   d) Reconhecer e identificar declarações de variáveis (int e float). Exemplo:

```
int a;
float b;
```

Leitura Complementar

- Aho, A. V., Lam, M. S., Jeffrey, R. S. **Compiladores: Princípios, Técnicas e Ferramentas**. 2ª edição, Pearson, 2007. ISBN: 978-8588639249.
  - **Capítulo 3: Lexical Analysis**
  - **Capítulo 4: Syntax Analysis**



- Sebesta, R. W. **Conceitos de Linguagens de Programação**. 9ª edição Editora Bookman, 2011. ISBN: 978-8577807918.
  - **Capítulo 3: Descrevendo a Sintaxe e a Semântica**
  - **Capítulo 4: Análise Léxica e Sintática**