

# Algoritmos e Estruturas de Dados II

## Terceira Implementação

Bacharelado em Análise de Sistemas, DCT–UFMS, 1/9/2004

Neste trabalho, deve-se implementar um analisador léxico para algoritmos simples. Um **analisador léxico** varre o arquivo contendo um algoritmo da esquerda para a direita, agrupando os símbolos em ítems chamados **ítems léxicos**, que podem ser palavras reservadas (como **algoritmo**, **declare**, **leia**), identificadores (como **x**, **soma**, **quadrado**) ou constantes (como **6**, **2890** ou **O valor de x é**).

### 1 Criação da Tabela de Dispersão

Criar uma tabela de dispersão para armazenamento das seguintes **palavras reservadas**:

<b>algoritmo</b>	<b>declare</b>	<b>numero</b>	<b>caracter</b>	<b>leia</b>
<b>escreva</b>	<b>se</b>	<b>entao</b>	<b>senao</b>	<b>enquanto</b>
<b>faca</b>	<b>para</b>	<b>ate</b>	<b>fim</b>	<b>=</b>
<b>+</b>	<b>-</b>	<b>*</b>	<b>/</b>	<b>'</b>
<b>==</b>	<b>!=</b>	<b>&lt;</b>	<b>&gt;</b>	<b>&lt;=</b>
<b>&gt;=</b>		<b>,</b>		

### 2 Funções Básicas

Implementar as funções de busca, inserção, remoção e impressão da tabela de dispersão.

### 3 Análise Léxica

Implementar um analisador léxico para algoritmos simples. O analisador deve receber como entrada um arquivo contendo um algoritmo e deve analisar cada item léxico do arquivo, verificando se o item léxico é uma palavra reservada, um identificador ou uma constante, de acordo com a tabela de dispersão criada.

Exemplo:

*Entrada:* arquivo soma-vetor.alg

```
algoritmo
    declare x, y numero

    x = 100
    leia y
    enquanto x <= y faca
        x = x + y
    fim
    escreva 'Resultado =', x, y
fim
```

*Saída:* (no monitor)

algoritmo	palavra/símbolo reservada(o)
declare	palavra/símbolo reservada(o)
x	identificador
,	palavra/símbolo reservada(o)
y	identificador
numero	palavra/símbolo reservada(o)
x	identificador
=	palavra/símbolo reservada(o)
100	constante
leia	palavra/símbolo reservada(o)
y	identificador
enquanto	palavra/símbolo reservada(o)
x	identificador
<=	palavra/símbolo reservada(o)
y	identificador
faca	palavra/símbolo reservada(o)
x	identificador
=	palavra/símbolo reservada(o)
x	identificador
+	palavra/símbolo reservada(o)
y	identificador
fim	palavra/símbolo reservada(o)
escreva	palavra/símbolo reservada(o)
'	palavra/símbolo reservada(o)
Resultado =	constante
'	palavra/símbolo reservada(o)
,	palavra/símbolo reservada(o)
x	identificador
,	palavra/símbolo reservada(o)
y	identificador
fim	palavra/símbolo reservada(o)

*Observações:*

1. Note que o caractere branco (' ') **não** é um item léxico. Você deve fazer uma função que “consome” esses caracteres.
2. Defina se o seu analisador léxico é sensível ao caso ou não. Isto é, se a mesma palavra escrita com letras maiúsculas e com letras minúsculas são idênticas do ponto de vista do armazenamento, ou se são diferentes.
3. Note que um item léxico pode terminar devido a mais de um separador: espaço em branco ou um símbolo reservado.
4. A não ser que você queira incrementar seu analisador léxico, seu programa deve se preocupar apenas com algoritmos simples, com estruturas condicionais e estruturas de repetição, sem vetores, matrizes, apontadores ou funções.
5. Quando as chaves são seqüências de caracteres, em geral, a função de dispersão utilizando o método da divisão é obtida da seguinte forma. Seja  $x$  um item léxico com  $n$  caracteres. Então,

$$h(x) = \left( \sum_{i=1}^n \text{ascii}(x_i) \right) \bmod m.$$

Exemplo:

Seja  $x = abcd$  um item léxico do arquivo de entrada. Então, o valor do endereço-base será calculado da seguinte forma:

$$\begin{aligned} h(x) &= (\text{ascii}('a') + \text{ascii}('b') + \text{ascii}('c') + \text{ascii}('d')) \bmod m \\ &= (97 + 98 + 99 + 100) \bmod m. \end{aligned}$$

## 4 Entrega

O código-fonte de seu programa na **linguagem C padrão**, deve ser armazenado em um disquete e entregue na data de **22 de setembro**, durante a aula.