Unidades léxicas (*tokens*)

- Relembrando...
  - Função do analisador léxico é...
    - Ler programa fonte (seqüência de caracteres)
    - Coletar *tokens* (palavras de uma Linguagem Regular)
  - *Token* representado internamente por 3 informações
    - Classe
      - » Identificador, cte. numérica, palavra reservada, operador, etc.
    - Valor (depende da classe)
      - » Identificador: ele mesmo ou seu índice na Tabela de Símbolos
      - » Cte. numérica: o próprio número ou seu índice na Tabela.
      - » Palavra reservada: sem valor associado
    - Posição
      - » Linha e coluna onde o *token* ocorre no texto do programa
      - » (posterior uso para indicação de local de erro!)

Unidades léxicas (*tokens*)

- Relembrando...
  - Valor determina 2 grupos de *tokens*:
    - Simples: sem valor associado
      - » Palavras reservadas, operadores, delimitadores
    - Com argumento: com valor associado (definidos pelo programador)
      - » Identificador, cte. numérica, string

- Exemplo inicial: "while I < 100 do I := J + I;"
  - Análise léxica produz seguinte seqüência de *tokens*

    [whi,] [id, 7] [<,] [cte, 13] [do,] [id, 7] [:=,] [id, 12] [+,] [id, 7] [;,]

  - Omitindo posição
  - [classeToken,índiceTabela] para identificadores e ctes. numéricas
  - Palavras reservadas abreviadas
  - Delimitadores e operadores: classe é o próprio valor do *token*

  - (para simplificar, Tradutores geralmente associam um inteiro a cada classe)

Especificação

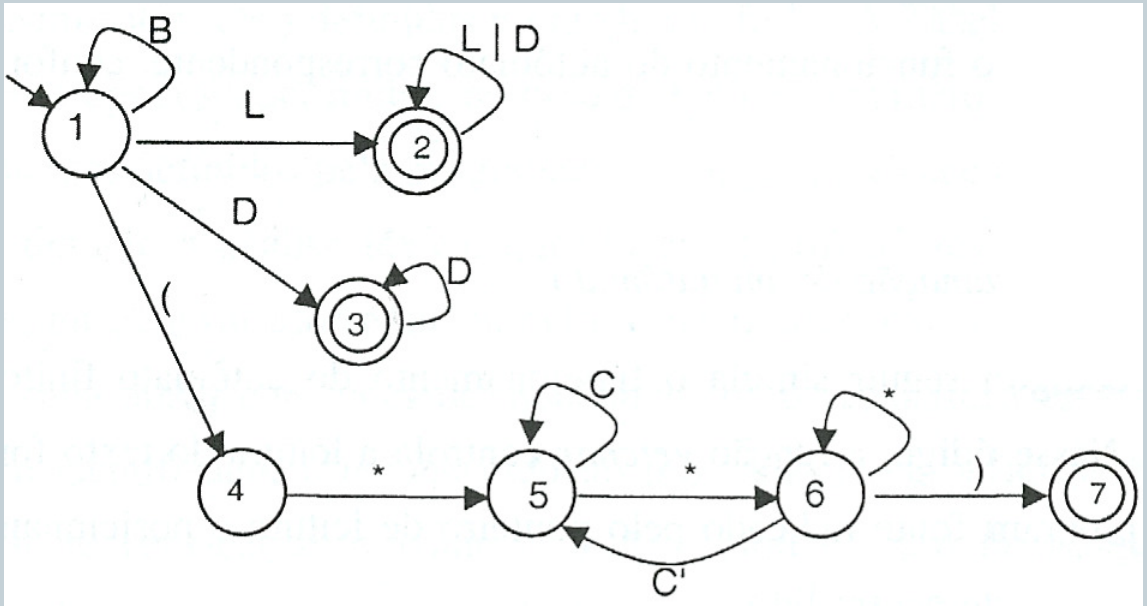
- Especificação de Analisadores Léxicos:
  - Formalismos para definição de Linguagens Regulares
    - Autômatos Finitos
    - Expressões Regulares
    - Gramáticas Regulares
  - Formalismo deve descrever toda a Linguagem, ou seja, todo o conjunto de *tokens*
    - Além disso, levar em conta caracteres que aparecem entre os *tokens*...
      - ...e que devem ser ignorados (branco, comentários, etc.)
  - Tratamento de palavras reservadas
    - Caso particular de identificador
    - Não previstas explicitamente na especificação
      - Tabela separada, consultada sempre que um identificador é encontrado
      - Se "identificador" na tabela, então é palavra reservada

Especificação

- Exemplo
  - L é o conjunto dos seguintes *tokens*:
    - Identificadores
      - » Letra seguida de combinação opcional de letras e/ou dígitos
    - Inteiros
      - » Um ou mais dígitos
    - Comentários
      - » Delimitados por (* *)
    - (ignorar espaços em branco)
  - AF (especificando analisador léxico) para reconhecer L...

Especificação

- No AF (um estado final para cada classe de *token*):
  - B representa espaço em branco
  - L, letra
  - D, dígito
  - C qualquer caractere menos *, C', qualquer um menos * e)



- AF para reconhecer L (Price & Toscani, 2005) -

Implementação

- Podemos codificar o AF:

```
letter: set of (a .. z);
digit: set of (0 .. 9);
ident := null;      /* inicialização */
number := null;
begin
      car := getchar;
      while car = ' ' do  car := getchar;
      if car in letter
      then  while car in (letter or digit) do
                begin
                      ident := ident || car;
                      car := getchar
                end
      else   if car in digit
             then   while car in digit do
                      begin
                            number := number || car;
                            car := getchar
                      end
              else  if car in delimiter
                      then ....
      ....
end
```

- Trecho de código para o AF (Price & Toscani, 2005) -

Implementação

- Ou
  - Construir especificação com ER
  - E utilizar um gerador de analisador léxico...
    - para gerar o código correspondente
    - (ao AF que reconhece a linguagem definida pela ER)
- Por exemplo, no lex/flex...