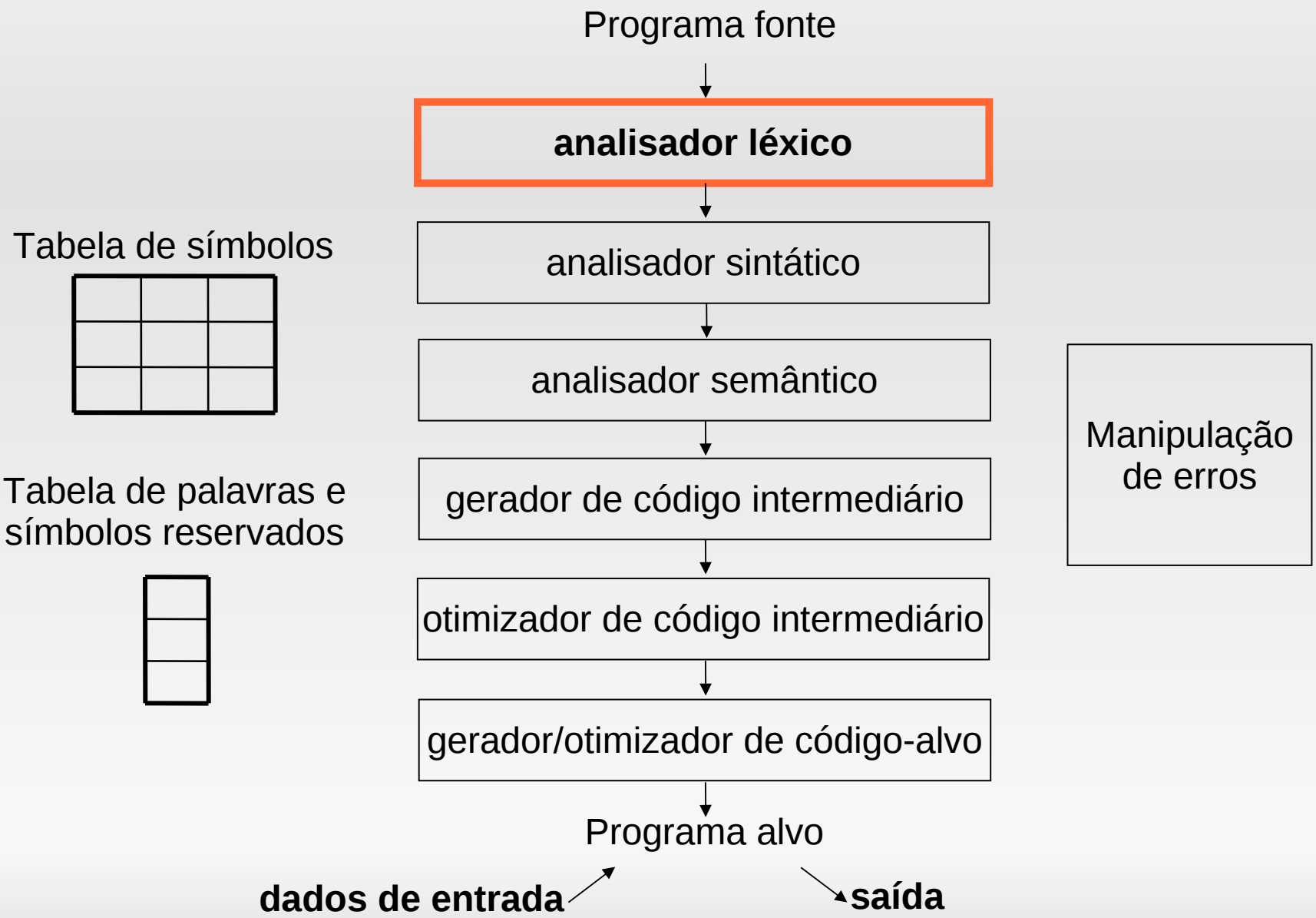
Construção de Compiladores

Análise Léxica

Profa. Helena Caseli
helenacaseli@dc.ufscar.br

Processo de Tradução

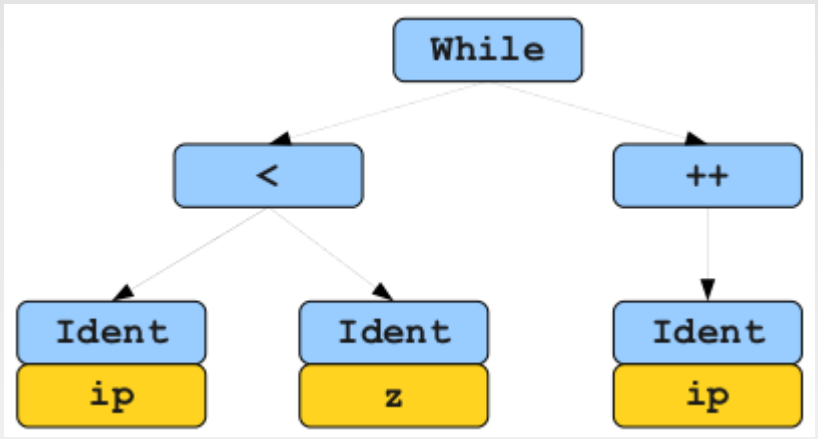


Análise Léxica

- O que é?
  - Etapa que lê o programa fonte como uma sequência de caracteres e os separa em *tokens*:
    - palavras reservadas: if, while etc.
    - identificadores
    - símbolos reservados: +, *, >=, <> etc.

Análise Léxica

- O que é?
  - Etapa que lê o programa fonte como uma sequência de caracteres e os separa em *tokens*

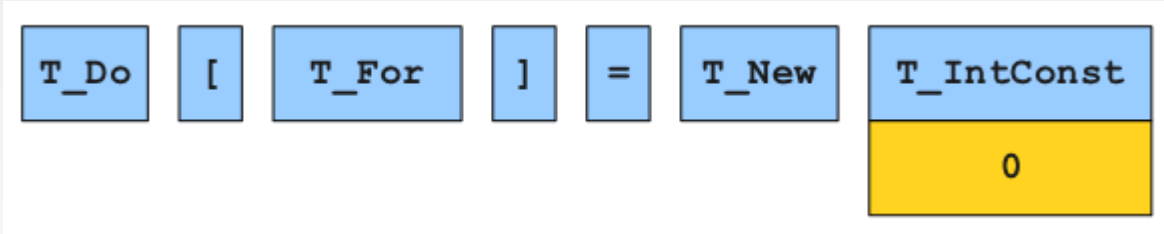


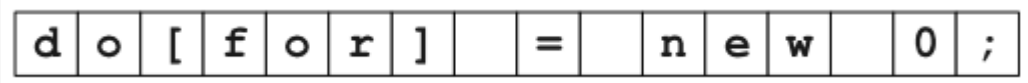

```
while (ip < z)
    ++ip;
```

http://www.stanford.edu/class/cs143/

Análise Léxica

- O que é?
  - Etapa que lê o programa fonte como uma sequência de caracteres e os separa em *tokens*






```
do[for] = new 0;
```

http://www.stanford.edu/class/cs143/

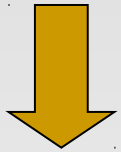
Análise Léxica

- O que é?
  - O *token*
    - Representa um determinado padrão de caracteres reconhecido pela varredura a partir do início à medida que esses caracteres são fornecidos
    - Possui várias informações associadas
      - categoria (identificador, símbolo, número, palavra reservada, etc.)
      - cadeia (valor ou lexema)
      - posição (linha e coluna em que ocorre no programa fonte)
    - Pode ter apenas um lexema (*token* simples) ou vários lexemas (*token* com argumentos)
      - Um lexema: palavras reservadas (if, then, else, while, etc.)
      - Vários lexemas: identificadores (var1, x, y, proc1, etc.)
        - todos *tokens* ” id”

Análise Léxica

- Exemplo

x:=x+y*2



<x,id> <:=,:=> <x,id> <+,op> <y,id> <*,op> <2,num>

cadeia categoria

- Como particionar o programa em *tokens*?

```
DO 5 I = 1,25
DO 5 I = 1.25
```

X

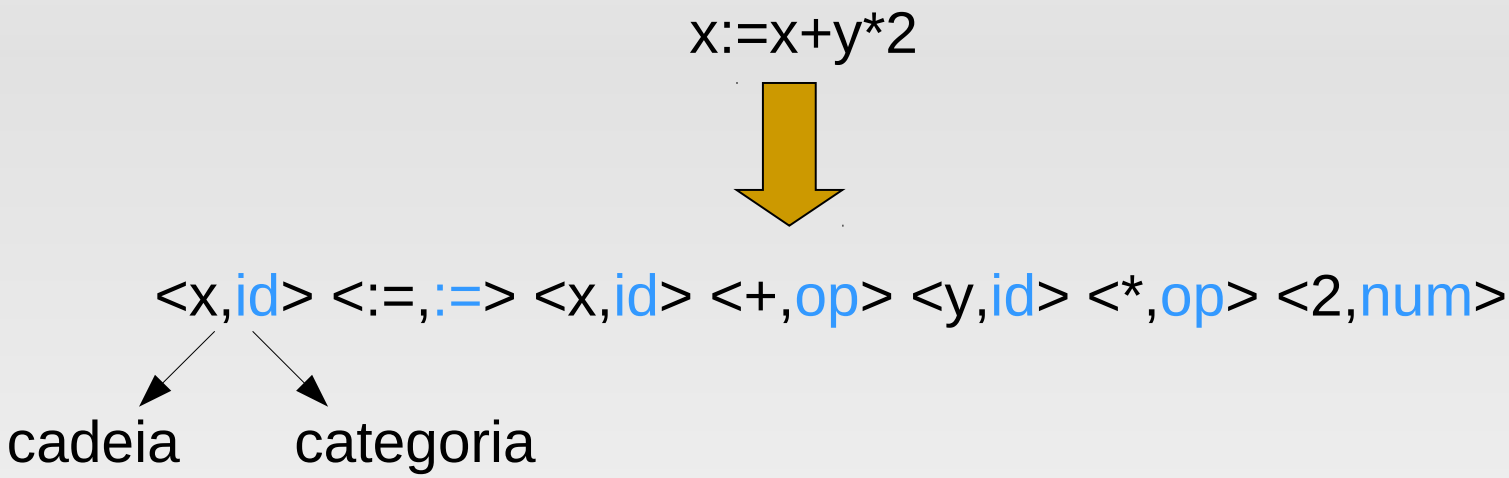
```
DO 5 I = 1,25
 DO5I  = 1.25
```

em FORTRAN espaço em branco é irrelevante

http://www.stanford.edu/class/cs143/

Análise Léxica

- Exemplo



- Como classificar cada *token* corretamente?

```
IF THEN THEN THEN = ELSE; ELSE ELSE = IF
```

Palavras reservadas podem ser usadas como identificador em algumas linguagens

http://www.stanford.edu/class/cs143/

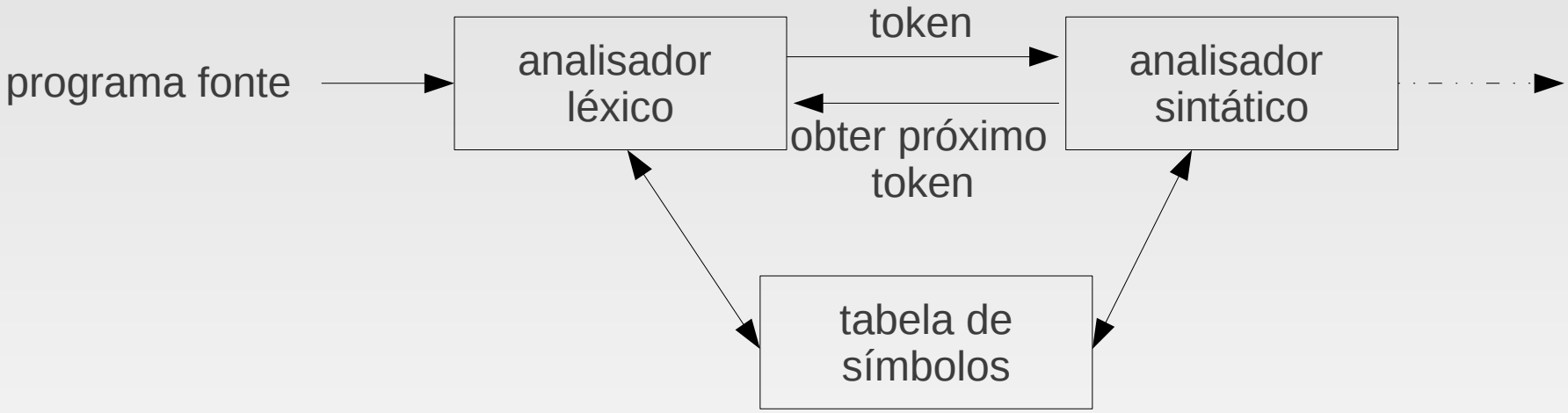
Análise Léxica

- Como é projetada?
  1. Escolha dos *tokens* (suas categorias)
     - Tipicamente: palavras e símbolos reservados, identificadores, números, etc.
     - Descartando comentários e espaços em branco
  2. Definição do conjunto de lexemas que cada categoria/*token* pode assumir
     - Combinação válida de caracteres na linguagem para formar os lexemas
     - ➔ **Expressões regulares**
  3. Projeto e implementação do algoritmo capaz de identificar e classificar corretamente os *tokens* desejados
     - ➔ **Autômatos**

http://www.stanford.edu/class/cs143/

Análise Léxica

- Como é executada?
  - Sob o comando do analisador sintático que requisita um *token* sempre que necessário



  - Motivos para dividir conceitualmente léxica de sintática
    - Simplificação, modularização e portabilidade
    - Diferentes notações garantem maior eficiência

Análise Léxica

- Como é executada?
  - Sob o comando do analisador sintático que requisita um *token* sempre que necessário
  - Exemplo em C:

```
typedef enum
{ IF, THEN, ELSE, PLUS, MINUS, NUM, ID, ...}
TokenType;

TokenType getToken(void);
```

  - A função `getToken` retorna o próximo *token* e computa todos os atributos do mesmo
  - ➔ **Expressões regulares** e **autômatos finitos**