



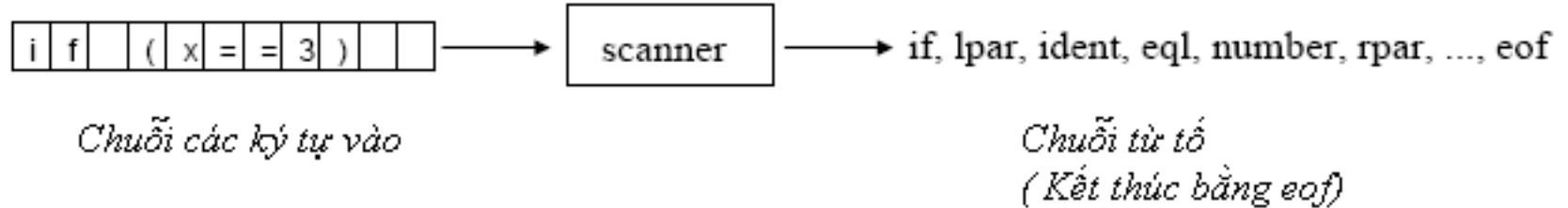
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Bài 5

Bộ phân tích từ vựng

Nhiệm vụ của bộ phân tích từ vựng

- Phát hiện các từ tố



- Bỏ qua các ký tự không cần thiết
 - Khoảng trống
 - Dấu tab
 - Ký tự xuống dòng (CR,LF)
 - Chú thích

Từ tổ có cấu trúc cú pháp

```
ident =    letter {letter | digit}.  
number =  digit {digit}.  
if =      "i" "f".  
eq =      "=" "=".  
...
```

- Tại sao không xử lý các luật này trong giai đoạn phân tích cú pháp ?

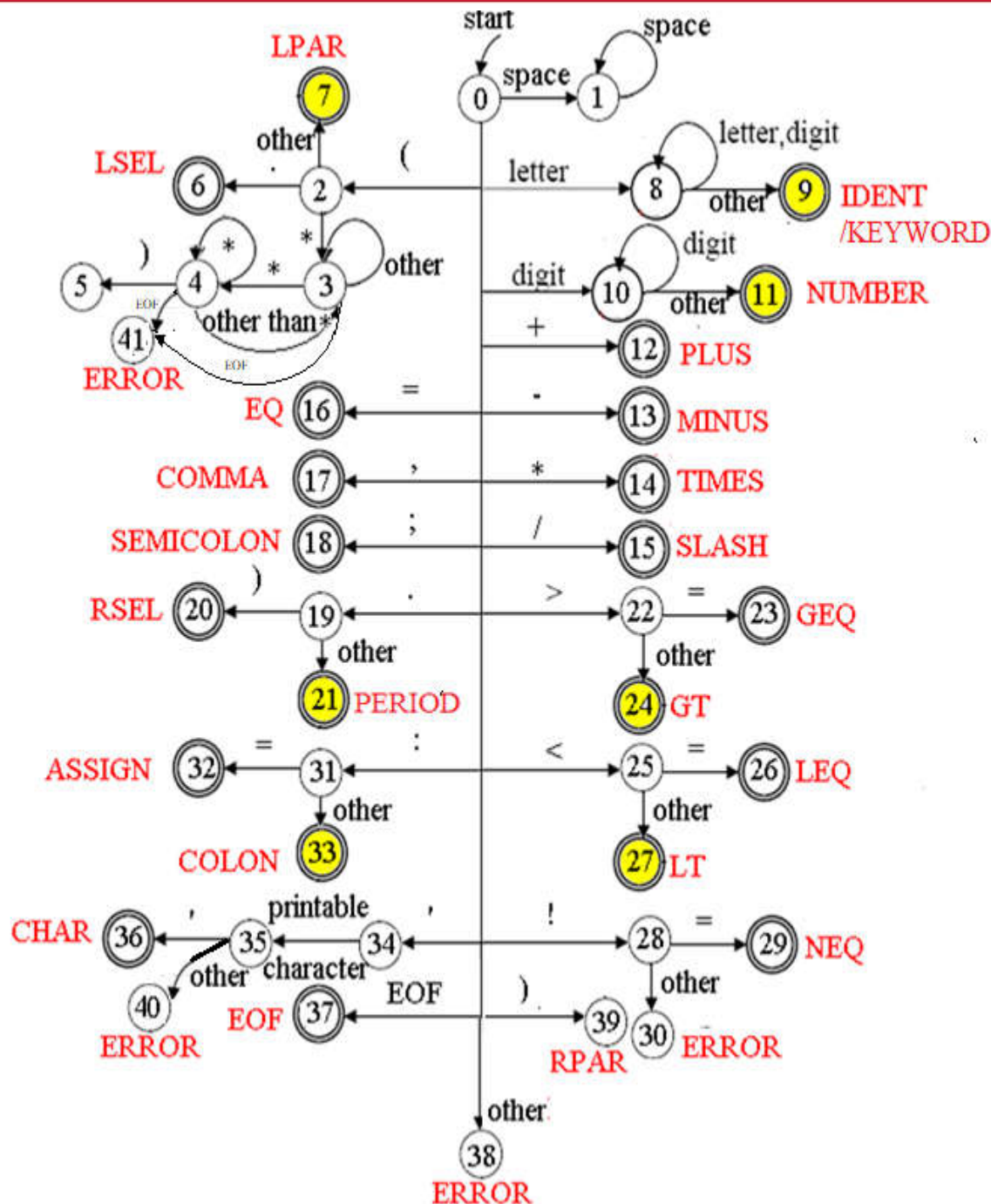
Xử lý các luật từ vựng trong bộ phân tích cú pháp ?

- Làm cho bộ phân tích cú pháp trở nên quá phức tạp
 - Phân biệt tên và từ khoá
 - Phải có những luật phức tạp để xử lý chuỗi các ký tự không cần thiết (khoảng trống, tab, chú thích)

Các từ tổ của KPL

- Số nguyên
- Định danh
- Từ khóa: begin,end, if,then, while, do, call, const, var, procedure, program,type, function,of,integer,char,else,for, to,array
- Hằng ký tự
- Dấu phép toán:
 - số học
+ - */
 - so sánh
= != < > <= >=
- Dấu phân cách
() . : ; (.)
- Dấu phép gán :=

Ôtômat hữu hạn của bộ phân tích từ vựng KPL



Mỗi khi đoán nhận được 1 từ tố, ô tô mat hữu hạn lại quay về trạng thái 0.

Với những ký tự không đoán nhận được, cần thông báo lỗi.

Nếu ô tô mat đến những trạng thái màu vàng, ký tự hiện hành đã là ký tự đầu của token tiếp theo

Cài đặt bộ phân tích từ vựng dựa trên ô tômat

```
state = 0;  
currentChar = getCurrentChar;  
token = getToken();  
while ( token!=EOF)  
{  
    state =0;  
    token = getToken();  
}
```

Đoán nhận từ tổ

```
switch (state)
{
case 0 : currentChar = getCurrentChar();
    switch (currentChar)
    {
        case space
            state = 1;
        case lpar
            state = 2;
        case letter
            state = 8;
        case digit
            state = 10;
        case plus
            state = 12;
```


Đoán nhận từ tổ (tiếp theo)

case 1:

```
while (current Char== space) // skip blanks  
    currentChar = getCurrentChar();
```

```
state =0;
```

case 2:

```
currentChar = getCurrentChar();
```

```
switch (currentChar)
```

```
{
```

```
case period
```

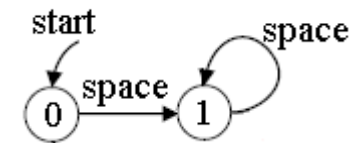
```
state = 6; // token lsel
```

```
case times
```

```
state =3; //skip comment
```

```
else
```

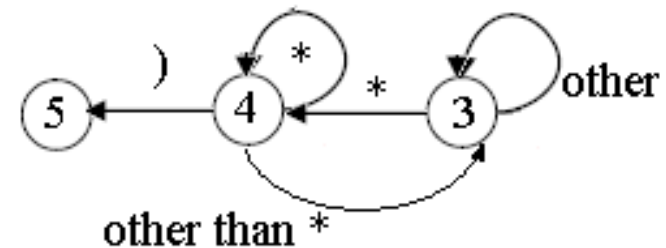
```
state =7; // token lpar
```



Đoán nhận từ tổ (tiếp theo)

case 3: // skip comment

```
currentChar = getCurrentChar();  
while (currentChar != times)  
{state = 3;  
currentChar = getCurrentChar();}  
state = 4;
```



case 4:

```
currentChar = getCurrentChar();  
while (currentChar == times)  
{state = 4;  
currentChar = getCurrentChar();}
```

If (currentChar == lpar) state = 5; else state = 3;

Đoán nhận từ tổ (tiếp theo)

case 9:

```
if (checkKeyword (token) == TK_IDENT)
```

```
install_ident();// save to symbol table
```

```
else
```

```
return checkKeyword(token);
```

.....

Các thông tin trong bảng ký hiệu

- Thông tin của định danh
 - Tên: xâu ký tự
 - Thuộc tính: tên kiểu, tên biến, tên thủ tục, tên hằng. . .
 - Kiểu dữ liệu
 - Phạm vi sử dụng
 - Địa chỉ vùng nhớ, kích cỡ vùng nhớ
 - . . .
- Với các số, thông tin về giá trị sẽ được lưu trữ

Cấu trúc dữ liệu

enum {

TK_NONE, TK_IDENT, TK_NUMBER, TK_CHAR, TK_EOF,

KW_PROGRAM, KW_CONST, KW_TYPE, KW_VAR,

KW_INTEGER, KW_CHAR, KW_ARRAY, KW_OF,

KW_FUNCTION, KW_PROCEDURE,

KW_BEGIN, KW_END, KW_CALL,

KW_IF, KW_THEN, KW_ELSE,

KW_WHILE, KW_DO, KW_FOR, KW_TO,

SB_SEMICOLON, SB_COLON, SB_PERIOD, SB_COMMA,

SB_ASSIGN, SB_EQ, SB_NEQ, SB_LT, SB_LE, SB_GT, SB_GE,

SB_PLUS, SB_MINUS, SB_TIMES, SB_SLASH,

SB_LPAR, SB_RPAR, SB_LSEL, SB_RSEL

Xử lý định danh / từ khoá

- Lập danh mục từ khóa, có thể dùng mảng
- Nếu số lượng từ khóa nhiều có thể phân phối bộ nhớ động
- Lập một hàm trả ra một từ khóa hoặc định danh

Lưu ý:

- Quan tâm đến việc phân biệt chữ hoa/chữ thường
- Xử lý dấu _
- Độ dài số hợp lý để tránh lỗi khi chuyển từ ký tự sang số
- Không dừng chương trình khi gặp lỗi
- Nếu dùng ‘\’, ‘\\’ để biểu diễn các hằng ‘ và \ thì xử lý như thế nào?
- Độ dài tối đa cho định danh có thể là bao nhiêu?