CHƯƠNG TRÌNH DỊCH

Bài 5: Sinh Tự Động Bộ PTTV

Nội dung

1. Giải bài tập của các buổi trước
2. Giới thiệu về LEX
3. CsLex – phiên bản LEX cho C#

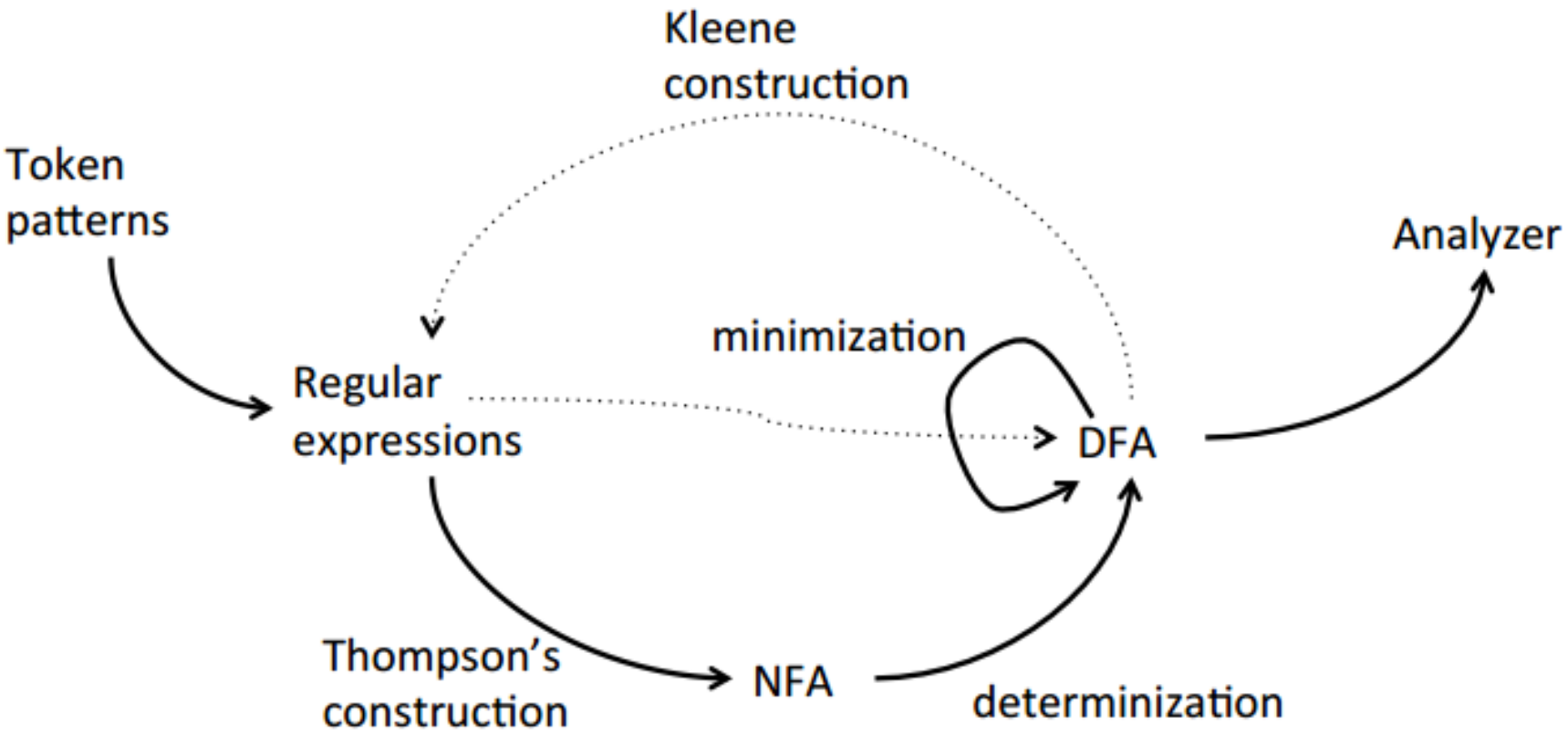
Phần 1

Giải bài tập của các buổi trước

Phần 2

Giới thiệu về LEX

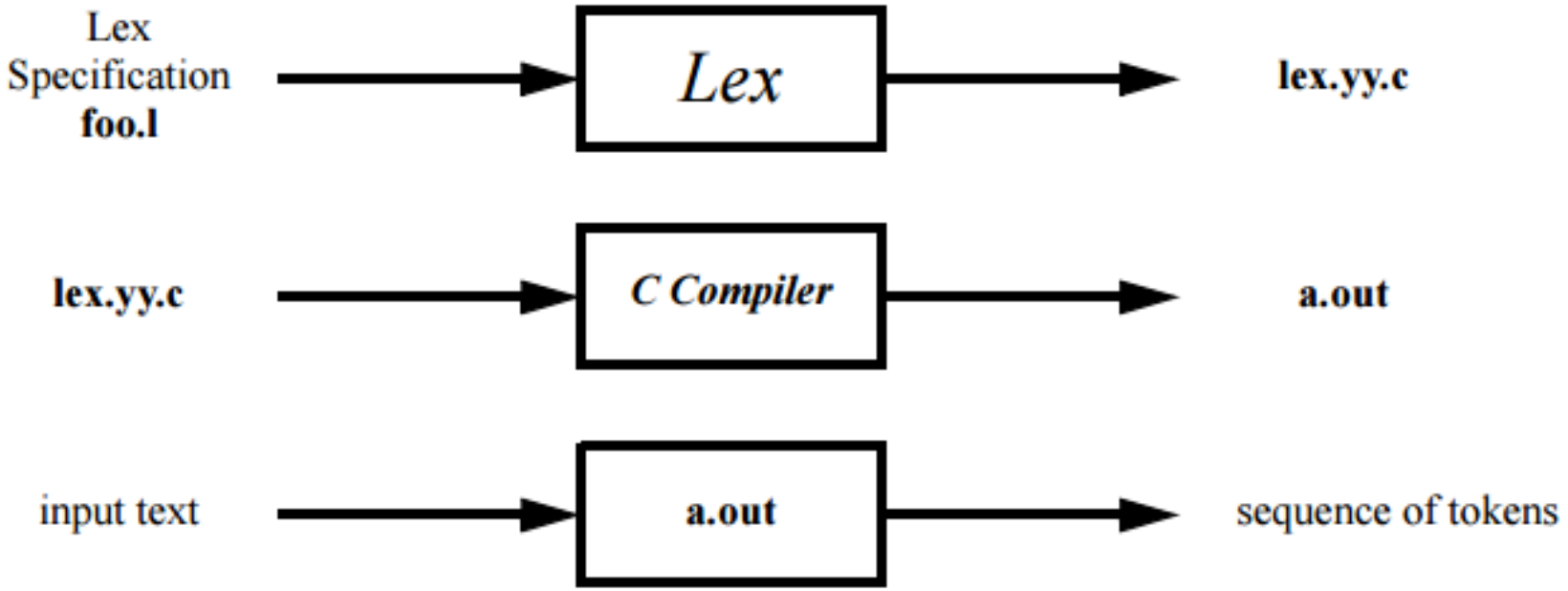
Từ patterns đến scanner



Các bước để tạo một bộ PTTV

- Các bước để tạo một bộ PTTV:
  1. Định nghĩa từ loại ở dạng các RE
  2. Chuyển các RE thành một NFA duy nhất
  3. Chuyển NFA thành DFA
  4. Tối ưu hóa DFA
  5. Viết mã xử lý DFA
  6. Xử lý các tình huống nhập nhằng hoặc đặc biệt
- Tự viết bộ PTTV (ad-hoc analyser): tự làm tất cả các bước trên
- Sử dụng LEX: làm bước 1 và 6, LEX làm các bước còn lại

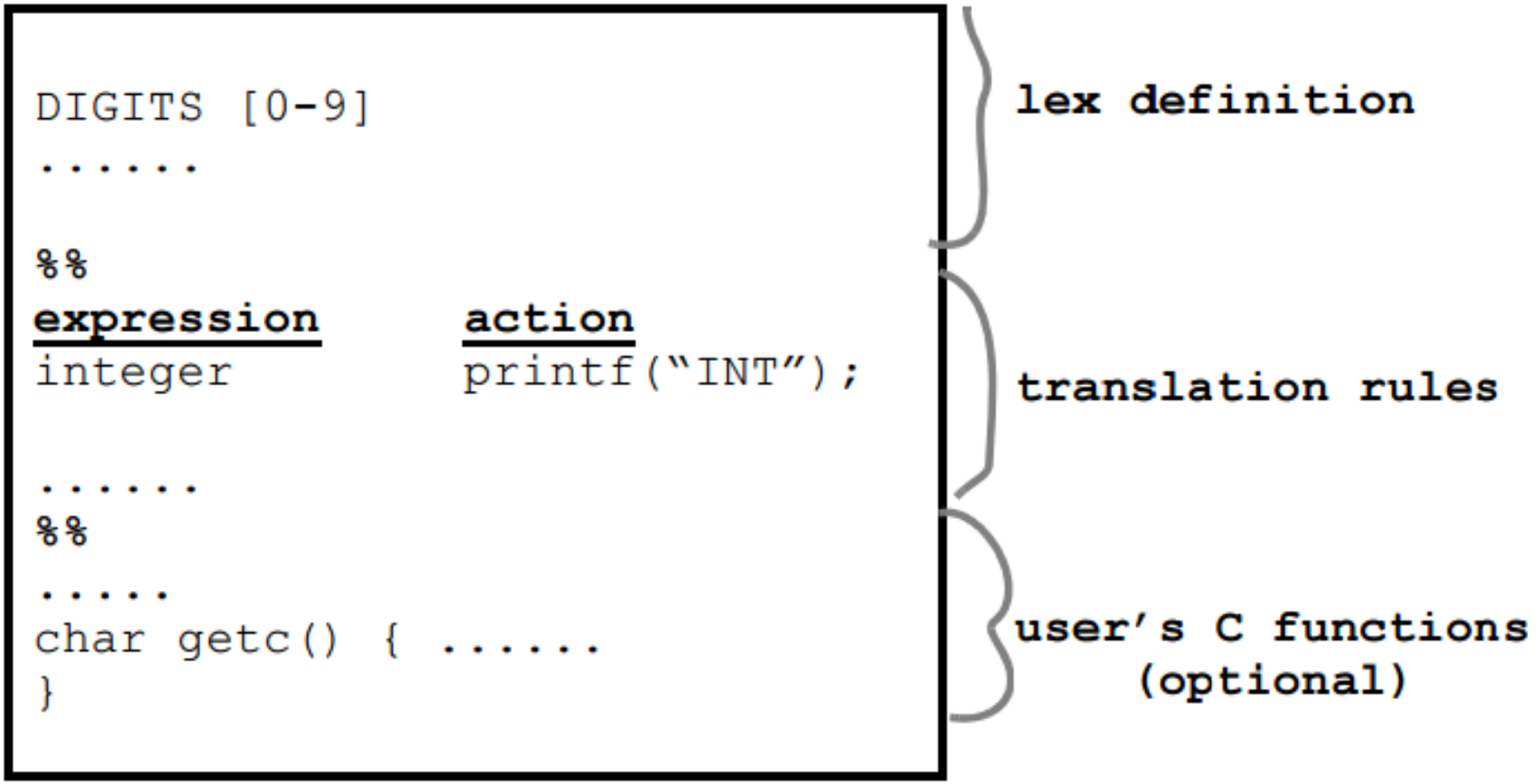
LEX: cách làm việc



Implementation of Lex:

Lex Spec -> NFA -> DFA -> Transition Tables + Actions -> yylex()

LEX: đầu vào



expression *is a regular expression ;* ***action*** *is a piece of C program;*

Ví dụ về LEX

- Cho ngôn ngữ X có văn phạm như sau:

```
stmt → if expr then stmt else stmt
     | if expr then stmt | ε
expr → term relop term | term
term → id | num
```

- Các từ loại:

```
if    → if
then  → then
else  → else
relop → < | <= | = | <> | > | >=
id    → letter (letter | digit)*
num   → digit+ (. digit+)? (E (+ | -)? digit+)?
delim → blank | tab | newline
ws    → delim+
```

Ví dụ về LEX

```
%{
/* đoạn mã định nghĩa các hằng: LT, LE, EQ, NE, GT, GE, IF, THEN,
ELSE, ID, NUMBER, RELOP */
#define LT 0
#define LE 1
…
}%
/* định nghĩa chính quy */
delim    [\t\n]
ws       {delim}+
letter   [A - Za -  z]
digit    [0 - 9]
id       {letter}({letter}| {digit})*
number   {digit}+(\.{digit}+)?(E[+\-]?{digit}+)?
```

Ví dụ về LEX

```
%%
/* xử lý khi gặp các ký hiệu */
{ws}        {/* Không có action, không có return */}
if          {return(IF); }
then        {return(THEN); }
else        {return(ELSE); }
{id}        {yylval = install_id( ); return(ID) }
{number}    {yylval = install_num( ); return(NUMBER) }
"<"         {yylval = LT; return(RELOP) }
"<="        {yylval = LE; return(RELOP) }
"="         {yylval = EQ; return(RELOP) }
"<>"        {yylval = NE; return(RELOP) }
">"         {yylval = GT; return(RELOP) }
">="        {yylval = GE; return(RELOP) }
```

Ví dụ về LEX

```
%%
/* các thủ tục bổ sung, viết bằng ngôn ngữ C */

/* Thủ tục phụ cài id vào trong bảng ký hiệu */
install_id ( ) {
    …
}

/* Thủ tục phụ cài một số vào trong bảng ký hiệu */
install_num ( ) {
   …
}
```

Phần 3

CsLex – phiên bản LEX cho C#

CsLex

- Cũng tương tự như Lex, nhưng sử dụng ngôn ngữ C# để làm việc
- Phiên bản gần nhất có thể tìm thấy trên GitHub, yêu cầu lớp về tìm hiểu để có thể làm bài thực hành
- Cấu trúc file input tương tự như Lex, nhưng có điều chỉnh cho phù hợp với C#

```
user code
%%
CsLex directives
%%
regular expression rules
```

KIỂM TRA

(45 phút)

Kiểm tra

- *Cho ngôn ngữ L có các từ loại sau:*
  - *Tên phép toán: ADD, SUB, MUL, DIV, MOD*
  - *Địa chỉ bộ nhớ: số nguyên không dấu nằm trong cặp ngoặc vuông (ví dụ: [15], [1000], [0],...)*
  - *Tên thanh ghi: EAX, EBX, ECX, EDX*
  - *Tên biến: các chữ cái tiếng Anh viết liền, không trùng với tên phép toán hoặc tên thanh ghi*
- *Yêu cầu:*
  1. *Viết biểu thức chính quy cho các từ loại của L*
  2. *Xây dựng các đồ thị chuyển cho từng từ loại*
  3. *Kết hợp các đồ thị chuyển ở câu 2 thành đồ thị chuyển duy nhất đoán nhận ngôn ngữ L*