



Chương trình dịch

Bài toán xây dựng bộ phân tích từ vựng ngôn ngữ VC

Nhóm 4

Vũ Trọng Thanh

Phạm Công Chính

Nguyễn Văn Thành

Nguyễn Ngọc Trường Sơn

Trần Vũ Toàn

1. Giới thiệu bài toán

VC là một biến thể của C. Nó có cả những đặc điểm chính của ngôn ngữ C và một vài tính năng của ngôn ngữ Java. Ngôn ngữ này được thiết kế để sử dụng trong chủ đề biên dịch tầng dưới, bao gồm một số kiểu nguyên thủy như các mảng một chiều, các cấu trúc điều khiển, các biểu thức, các câu lệnh phức hợp (ví dụ như các khối) và các hàm được lấy phần lớn từ ngôn ngữ C.

Bài toán “Xây dựng bộ phân tích từ vựng ngôn ngữ VC” được thực hiện với mục đích là nhận diện chính xác các từ vựng trong một file mã nguồn VC đồng thời báo lỗi nếu có

2. Input và Output của chương trình

2.1 File cấu hình ô tô mát

File vc.jflex bao gồm

- Trạng thái ban đầu
- Trạng thái kết thúc
- Danh sách các hàng
- Danh sách các cột
- Bảng chuyển
- Ánh xạ giữa trạng thái kết thúc và từ tố
- Các trạng thái mà phải trả lại kí tự đọc vào cuối cùng vào dòng vào

2.2 Input

File văn bản *.vc chứa chương trình VC

2.3 Output

File văn bản có dạng *.vctok chứa danh sách các từ vựng và kiểu của nó, mỗi từ vựng nằm trên một dòng theo định dạng <Từ vựng><dấu cách><Kiểu của từ vựng>

3. Giải quyết bài toán

- Source code Project: <https://github.com/thanhtlx/compiler>

3.1 Các bước thực hiện

- Suu tầm tất cả các luật từ vựng.
- Vẽ đồ thị chuyển cho từng luật một.
- Kết hợp các luật này thành một đồ thị chuyển duy nhất.
- Chuyển đồ thị này thành bảng.
- Thêm phần chương trình ở trên để thành bộ phân tích hoạt động được.

3.2 Xây dựng ô tô mát từ các luật từ vựng

- Xây dựng ô tô mát đoán nhận các danh định, các từ khóa
- Xây dựng ô tô mát đoán nhận các toán tử
- Xây dựng ô tô mát đoán nhận các ký tự phân cách
- Xây dựng ô tô mát đoán nhận các hằng
- Xây dựng ô tô mát đoán nhận các khoảng trắng, comment
- Kết hợp các ô tô mát trên

3.3 Xây dựng bảng chuyển

Bảng chuyển được xây dựng từ đồ thị chuyển. Chi tiết xin xem file vc.jflex đi kèm.

4. Cài đặt

4.1 Ngôn ngữ sử dụng

Java

4.2 Cấu trúc chương trình

Chương trình gồm có

- VCLexer: lớp chính của chương trình, được sử dụng để cài đặt bộ phân tích từ vựng

4.3 Lớp VCLexer

1. Constructor nhận vào 1 tham số là đối tượng file input.
2. Phương thức yylex(): phương thức yylex() tiến hành phân tích nội dung file đầu vào
3. Phương thức printTokens(): in ra toàn bộ token mà công cụ phân tích được

5. Một số chú ý đặc biệt

- Nếu ở dòng cuối cùng chương trình bắt gặp ký hiệu kết thúc file (EOF) trước khi có ký hiệu xuống dòng thì token cuối cùng sẽ báo là unterminated error. Điều này là do trong đặc tả VC đã ghi trong phần 4.1 là “Các dòng được kết thúc bởi các ký tự CR, LF, hoặc CR LF.” nên khi duyệt tới EOF token cuối cùng thì token cuối sẽ không đạt được trạng thái kết thúc.

6. Hướng dẫn chạy

- Biên dịch VCLexer.java bằng lệnh “javac VCLexer.java”. Yêu cầu cần phải có JDK 11 trở lên.
- Chạy chương trình bằng lệnh “java VCLexer “input file” > “output file”.
- Nếu chương trình hoạt động tốt, thì kết quả sẽ được in ra “output file”

7. Ví dụ

Với input là đoạn mã lệnh sau:

```
“int main(void){  
    print("hello world");  
    int sum = 0;  
  
    for(int i = 0; i < 5 ; i++) {  
        sum += i; // tính tổng  
    }  
    return 0;  
}  
“
```

Thì output của chương trình sẽ cho ra kết quả:
“

Token: int Type: Keyword

Token: main Type: Identifier

Token: (Type: Separator

Token: void Type: Keyword

Token:) Type: Separator

Token: { Type: Separator

Token: print Type: Identifier

Token: (Type: Separator

Token: "hello world" Type: String Literal

Token:) Type: Separator

Token: ; Type: Separator

Token: int Type: Keyword
Token: sum Type: Identifier
Token: = Type: Assignment Operator
Token: 0 Type: Integer Literal
Token: ; Type: Separator
Token: for Type: Keyword
Token: (Type: Separator
Token: int Type: Keyword
Token: i Type: Identifier
Token: = Type: Assignment Operator
Token: 0 Type: Integer Literal
Token: ; Type: Separator
Token: < Type: Relational Operator
Token: 5 Type: Integer Literal
Token: ; Type: Separator
Token: + Type: Arithmetic Operator
Token: + Type: Arithmetic Operator
Token:) Type: Separator
Token: { Type: Separator
Token: + Type: Arithmetic Operator
Token: = Type: Assignment Operator
Token: ; Type: Separator
Token: } Type: Separator
Token: return Type: Keyword
Token: 0 Type: Integer Literal
Token: ; Type: Separator
Token: } Type: Separator

”

8. Tổng kết

Trong suốt quá trình hoàn thiện bài tập lớn chúng em luôn được sự quan tâm, hướng dẫn và giúp đỡ tận tình của các thầy. Nhóm em xin được bày tỏ lòng biết ơn sâu sắc đến thầy đã tận tình giúp đỡ cho em suốt thời gian học tại trường. Dù dự án còn gặp nhiều sai sót, nhóm em rất hy vọng các thầy có thể góp ý và đưa ra giải pháp để chúng em có thể hoàn thiện sản phẩm trong tương lai.

Chúng em xin trân trọng cảm ơn!

Oki mình hiểu r cảm ơn Linh nha

Thật ra mấy hôm trc mình định hỏi Linh xem Cty có quan tâm tới ngày hội việc làm bên UET mình ko ý để mình xin hồ sơ tài trợ

//Gửi link page vs web

Nma mình hỏi thầy cô bảo hết hạn tài trợ r tiếc quá