

Cây phân tích cú pháp (parse tree)

Bài tập 1

Giả sử ta có luật sinh sau trong văn phạm:

$\text{assign} \rightarrow \text{ID ASSIGN exp SEMI}$

Trên cây phân tích cú pháp (parse tree) sinh ra bởi ANTLR, nút ứng với luật này là đối tượng của lớp có tên là?

- ☐ assign
- ☐ Assign
- ☐ AssignContext

Chưa đúng, ký hiệu không kết thúc trong văn phạm viết trên ANTLR phải bắt đầu bằng ký tự chữ thường (do đó chữ a trong assign phải viết chữ thường) nhưng tên của lớp ứng với nó phải bắt đầu bằng chữ hoa

Chưa đúng, tên của lớp ứng với ký hiệu không kết thúc phải được nối với chữ Context

Chính xác

Solution

1. Wrong
2. Wrong
3. Correct Option

Bài tập 2

Cho luật sinh của văn phạm được viết như sau:

$\text{expp} \rightarrow \text{ADD term expp} \mid \epsilon$

Số nút con của nút ứng với expp là bao nhiêu?

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3

[Show Feedback](#)**Solution**

1. Correct
2. Incorrect
3. Incorrect
4. Correct

Bài tập 3

Cho luật sinh của một văn phạm được viết ở dạng EBNF như sau:

$\text{prog} \rightarrow \text{stmt}^+$

Số nút con của nút ứng với prog trên cây phân tích cú pháp là bao nhiêu?

- ☐ 0
- ☐ 1
- ☐ Bất cứ số dương (>0) nào.

Chưa đúng, dấu + sau stmt cho biết có thể có nhiều stmt nhưng ít nhất phải có 1. Do đó, không thể có trường hợp không có nút con

Đúng nhưng chưa chính xác. Nút ứng với prog có thể có 1 nút con nhưng cũng có thể có số nút con khác.

Chính xác.

Solution

1. Wrong
2. Wrong
3. Correct Option

Bài tập 4

Cho luật sinh của một văn phạm như sau:

$\text{assign} \rightarrow \text{ID ASSIGN exp SEMI}$

Giả sử biến **ctx** đang cất giữ nút ứng với assign trên cây phân tích cú pháp sinh ra bởi ANTLR. Hãy viết biểu thức để truy xuất nút ứng với **exp** bên vế phải?

- ☐ ctx.ExpContext
- ☐ exp()
- ☐ ctx.getChild(2)
- ☐ ctx.exp

Chưa đúng, ExpContext là kiểu của nút ứng với exp trên cây phân tích cú pháp.

Chưa đúng, exp() là một phương thức ảo (instance method) trong lớp AssignContext, do đó, để gọi nó cần phải có một đối tượng nhận (receiver object).

Đúng nhưng chưa tốt, vì ctx.getChild(2) trả về nút thứ ba trong các nút con của nút ứng với assign trên cây phân tích cú pháp (tức nút ứng với exp). Nhưng dùng ctx.getChild(2) sẽ làm chương trình khó đọc vì cần phải xem luật sinh để biết nút thứ ba ứng với ký hiệu văn phạm nào. Ngoài ra, nếu luật văn phạm bị thay đổi bằng việc thêm vào 1 ký hiệu nào đứng trước exp dẫn đến thay đổi thứ tự thì đoạn code này cần phải sửa lại cho thích hợp

Chính xác

Solution

1. Wrong
2. Wrong
3. Wrong
4. Correct Option

Bài tập 5

Cho luật sinh của một văn phạm như sau:

ifstmt \rightarrow IF exp THEN stmt ELSE stmt

Giả sử biến **ctx** đang cất giữ nút ứng với **ifstmt** trên cây phân tích cú pháp (parse tree). Hãy viết biểu thức để truy xuất nút ứng với **stmt** của vế else?

- ☐ ctx.stmt
- ☐ ctx.stmt(2)
- ☐ ctx.stmt(1)
- ☐ ctx.getChild(5)

Chưa đúng. Vì trên vế phải của luật sinh này có đến 2 ký hiệu stmt nên phương thức stmt sẽ trả về một danh sách các nút (trong trường hợp này thì danh sách có 2 nút), chứ không phải trả về nút ứng với stmt của vế else

Chưa đúng. Vì các nút trong danh sách được đánh số từ 0 nên stmt(2) sẽ ứng với nút ứng với ký hiệu stmt thứ ba, mà trong vế phải không có ký hiệu stmt thứ ba.

Chính xác

Đúng nhưng không tốt vì rất khó đọc để hiểu biểu thức này nhằm truy xuất đến nút ứng với stmt của vế else

Solution

1. Wrong
2. Wrong
3. Correct Option
4. Wrong

Bài tập 6

Cho luật sinh của văn phạm như sau:

$\text{prog} \rightarrow \text{stmt}^+$

Giả sử ctx là biến đang giữ nút ứng với prog. Hãy viết biểu thức để nhận được một danh sách tất cả các nút con **stmt** của nút ứng với **prog**?

- ☐ stmt
- ☐ ctx.stmt
- ☐ ctx.stmt(0)
- ☐ ctx.getChildren()

Chưa đúng vì stmt là một phương thức ảo (instance method) nên phải có đối tượng nhận (receiver object) trong biểu thức gọi của phương thức này

Chính xác

Chưa đúng vì biểu thức gọi này sẽ trả về nút con đầu tiên trong các nút con của nút ứng với prog.

Đúng nhưng không tốt vì biểu thức gọi này sẽ luôn trả về tất cả các nút con của ctx, trong trường hợp này cũng là các nút con stmt.

Solution

1. Wrong
2. Correct Option
3. Wrong

4. Wrong