

Giả sử cần phải định nghĩa một hàm có tên **foo**, với hai thông số x kiểu Int và y kiểu Float, nhưng khi gọi hàm foo thì có thể gọi chỉ với 1 thông số là một số nguyên (ví dụ như List(1.2f, 3.0f, 1.2f).map(foo(3))). Hãy viết định nghĩa của hàm foo trên Scala? Chú ý tuân theo qui tắc viết sau để đảm bảo so trùng đúng:

- viết từ chữ def đến trước dấu = (không viết dấu = và thân hàm)
- chỉ có 1 khoảng trắng giữa chữ def và tên hàm, không có khoảng trắng nào khác trong định nghĩa
- tên hàm, tên, thứ tự và kiểu các thông số đúng theo đề bài đã nêu
- không cần viết kiểu trả về

Trả lời:

Gọi **dist(x,lst)** là hàm phân phối x với mỗi phần tử của danh sách lst để tạo một danh sách có các phần tử là các cặp gồm x và 1 phần tử của danh sách lst. Ví dụ dist(2,[3,4,5]) sẽ trả về [(2,3), (2,4), (2,5)].

Hãy điền vào chỗ trống trong THÂN của hàm dist?

def dist(x,lst):
return list()

Chú ý tuân thủ qui tắc viết để đảm bảo so trùng đúng:

- KHÔNG có khoảng trắng, CHỈ dùng khoảng trắng khi phải phân cách keyword và danh hiệu, khi đó cũng chỉ dùng 1 khoảng trắng.
- Sử dụng biến theo trình tự y,z,t

Given the production of dowhile statement as follows:

dowstmt \rightarrow DO stmt WHILE exp SM

where **dowstmt** represents a dowhile statement; exp is a non-terminal to represent an expression; stmt is a non-terminal to represent a statement; tokens DO, WHILE, SM represent keywords do, while, and ';', respectively.

Let node DOWSTMT in AST represent the above dowhile statement, how many children node of DOWSTMT are there?

Trả lời:

Gọi **dist(x:Int,lst:List[Int])** là hàm phân phối x với mỗi phần tử của danh sách lst để tạo một danh sách có các phần tử là các cặp gồm x và 1 phần tử của danh sách lst. Ví dụ `dist(2,List(3,4,5))` sẽ trả về `List((2,3), (2,4), (2,5))`.

Hãy viết THÂN của định nghĩa hàm dist?

`def dist(x:Int,lst:List[Int]) =`

Chú ý tuân thủ qui tắc viết để đảm bảo so trùng đúng:

- chỉ viết THÂN hàm tức từ sau dấu `=`
- không có khoảng trắng ở đầu và cuối
- NGẮN nhất (cố dùng simplifying) có thể

Trả lời: `lst.map((x,_))`

Cho văn phạm có các luật sinh sau:

`vardecls → vardecl vardecls | vardecl`

`vardecl → type ids`

`type → INTTYPE | FLOATTYPE`

`ids → ID COMMA ids | ID`

Một visitor sinh bởi ANTLR cho văn phạm sẽ có phương thức visit nào trong các phương thức dưới đây?

Chọn một:

- ☐ a. visitVardeclContext
- ☒ b. visitIds
- ☐ c. visittype
- ☐ d. visitID

Cho luật sinh của phát biểu if như sau:

`ifstmt → IF exp THEN stmt ELSE stmt`

Cho các khai báo cấu trúc dữ liệu của AST hiện có như sau:

`trait Stmt // cấu trúc lưu trữ cho phát biểu`

`trait Exp // cấu trúc lưu trữ cho biểu thức`

Chọn khai báo thích hợp cho cấu trúc dữ liệu của nút IfStmt biểu diễn cho phát biểu If trên AST?

Chọn một:

- ☐ a. `case class IfStmt(s:Stmt) extends Stmt`
- ☐ b. `case class IfStmt(e:Exp,s1:Stmt,s2:Stmt) extends Exp`
- ☐ c. `case class IfStmt(e:Exp,s:Stmt) extends Stmt`
- ☒ d. `case class IfStmt(e:Exp,s1:Stmt,s2:Stmt) extends Stmt`

Gọi zip là hàm nhận vào hai danh sách có số phần tử như nhau và trả về một danh sách các phần tử là kết hợp từng cặp phần tử của hai danh sách. Ví dụ: `zip(List(1,2,3),List(4,5,6))` sẽ trả về `List((1,4),(2,5),(3,6))`. Cho một phần của định nghĩa hàm zip như sau:

```
def zip(l1:List[Int],l2:List[Int]):List[(Int,Int)] = (l1,l2) match {  
  case (nil,nil) => nil // nil tương đương List()  
  -----  
}
```

Hãy chọn đáp án thích hợp nhất cho dòng để trống?

Chọn một:

- ☒ a. `case (h1::t1,h2::t2) => (h1,h2)::zip(t1,t2)`
- ☐ b. `case (h1::t1,h2::t2) => (h1::zip(t1),h2::zip(t2))`
- ☐ c. `case h::tail => h::zip(l1,l2)`
- ☐ d. `case (h1::t1,h2::t2) => (h1,h2)::(t1,t2)`

Cho cấu trúc dữ liệu của AST được khai báo như sau:

trait Exp

case class Add(e1:Exp,e2:Exp) extends Exp // cho phép cộng 2 ngôi

case class Minus(e1:Exp,e2:Exp) extends Exp // cho phép trừ 2 ngôi

case class Mul(e1:Exp,e2:Exp) extends Exp // cho phép nhân

case class Lit(i:Int) extends Exp

Hãy viết AST thích hợp của biểu thức $(10 - 20) * 5 + 7$ (độ ưu tiên và kết hợp như thông lệ)?

Qui ước viết:

- Viết đúng cả chữ thường và hoa, đúng thứ tự toán hạng (trái hoặc phải), trình tự tính toán

- Không viết khoảng trắng

= Không viết dư thừa

Trả lời:

Cho các hàm được định nghĩa như sau:

val inc = (x:Int) => x + 1

val sqr = (x:Int) => x * x

Hãy viết lệnh gán cho biến `sqrinc` dựa trên hàm `inc` và `sqr` để `sqrinc` có tác dụng như một hàm có thân là $(x + 1) * (x + 1)$? Chú ý đảm bảo tuân theo qui tắc viết để so trùng đúng:

- bắt đầu bằng **val `sqrinc` =**
- luôn cách nhau bằng 1 khoảng trắng
- không có khoảng trắng ở đầu và cuối

Trả lời:

Cho cấu trúc dữ liệu của AST được khai báo như sau:

```
trait Exp
```

```
case class Add(e1:Exp,e2:Exp) extends Exp // cho phép cộng 2 ngôi
```

```
case class Minus(e1:Exp,e2:Exp) extends Exp // cho phép trừ 2 ngôi
```

```
case class Mul(e1:Exp,e2:Exp) extends Exp // cho phép nhân
```

```
case class Lit(i:Int) extends Exp
```

Hãy chọn AST thích hợp của biểu thức $10 - 20 * 5 + 7$ (độ ưu tiên và kết hợp như thông lệ)?

Chọn một:

- ☐ a. `Mul(Minus(Lit(10),Lit(20)),Add(Lit(5),Lit(7)))`
- ☐ b. `BinExp("+",BinExp("-",Lit(10),BinExp(""*,Lit(20),Lit(5))),Lit(7))`
- ☐ c. `Minus(Lit(10),Add(Mul(Lit(20),Lit(5)),Lit(7)))`
- ☒ d. `Add(Minus(Lit(10),Mul(Lit(20),Lit(5))),Lit(7))`

Cho luật sinh của phát biểu if như sau:

`ifstmt` \rightarrow `IF exp THEN stmt ELSE stmt`

trong đó **ifstmt** là ký hiệu không kết thúc đại diện cho phát biểu if; các token **IF**, **THEN**, **ELSE** ứng với các từ khoá if, then, else; **exp** là ký hiệu không kết thúc đại diện cho một biểu thức; **stmt** là ký hiệu không kết thúc đại diện cho một phát biểu.

Nút trên cây phân tích cú pháp (parse tree) sinh bởi ANTLR ứng với ifstmt sẽ là đối tượng của lớp nào?

Chọn một:

- ☒ a. `IfstmtContext`
- ☐ b. `ifstmt`
- ☐ c. `Stmt`
- ☐ d. `Ifstmt`

Cho luật sinh của phát biểu if như sau:

$\text{ifstmt} \rightarrow \text{IF exp THEN stmt ELSE stmt}$

trong đó **ifstmt** là ký hiệu không kết thúc đại diện cho một phát biểu if; các token **IF**, **THEN**, **ELSE** ứng với các từ khoá if, then, else; **exp** là ký hiệu không kết thúc đại diện cho một biểu thức; **stmt** là ký hiệu không kết thúc đại diện cho một phát biểu bất kỳ.

Trên AST, nút IFSTMT được dùng diễn tả cho phát biểu if trên, hỏi nút IFSTMT sẽ có bao nhiêu nút con?

Chọn một:

- ☐ a. 2
- ☐ b. 6
- ☒ c. 3
- ☐ d. 1

Given the production of if statement as follows:

$\text{ifstmt} \rightarrow \text{IF exp THEN stmt ELSE stmt}$

where ifstmt is a non-terminal representing if statement; tokens IF, THEN, ELSE represent keywords if, then, else, respectively; exp is a non-terminal representing an expression; stmt is a non-terminal representing a statement.

In AST, let node IFSTMT represent the above if statement, how many children nodes of IFSTMT are there?

Trả lời:

Cho văn phạm có các luật sinh sau:

$\text{program} \rightarrow \text{vardecls}$

$\text{vardecls} \rightarrow \text{vardecl vardecls} \mid \text{vardecl}$

$\text{vardecl} \rightarrow \text{type ids}$

$\text{type} \rightarrow \text{INTTYPE} \mid \text{FLOATTYPE}$

$\text{ids} \rightarrow \text{ID COMMA ids} \mid \text{ID}$

Và AST tương ứng với văn phạm trên được định nghĩa như sau:

```
trait AST
case class Program(decls:List[VarDecl]) extends AST
case class VarDecl(typ:Type,id:List[String]) extends AST
trait Type extends AST
object IntType extends Type
object FloatType extends Type
```

Đối tượng IntType được tạo ra trong phương thức nào?

Chọn một:

- ☐ a. visitIds
- ☐ b. visitVarDecl
- ☐ c. visitINTTYPE
- ☒ d. visitType

Cho luật sinh của biểu thức nhị phân exp như sau:

$\text{exp} \rightarrow \text{operand (RELOP operand)}^*$

trong đó exp là ký hiệu không kết thúc đại diện cho một biểu thức; token **RELOP** đại diện cho một phép toán nhị phân; **operand** là ký hiệu không kết thúc đại diện cho một toán hạng.

Giả sử ctx là biến đang cất giữ nút ứng với exp, để truy xuất nút con operand đầu tiên trong vế phải thì cần phải viết như thế nào?

Chọn một:

- ☐ a. ctx.operand
- ☐ b. ctx.operand()
- ☐ c. ctx.OperandContext()
- ☒ d. ctx.operand(0)