**ĐỀ THI GIỮA KÌ**

Lưu ý: Các câu này mình viết lại theo trí nhớ của mình nên nếu có sai sót mong mọi người thông cảm. Có gì thì có thể hỏi lại mình.

1. Hãy viết biểu diễn các token sau với điều kiện không quá 7 kí tự: =, +=, -=, \*=

Đáp án: [+ - \*]? =

1. Hãy viết mô tả token của một comment bằng ANTLR. Comment sẽ được trình biên dịch bỏ qua trong quá trình biên dịch. Nó bắt đầu và kết thúc bằng hai dấu nháy kép. Trong comment có thể chứa bất kì kí tự nào tuy nhiên không được tồn tại 2 dấu nháp kép liền nhau.

Đáp án: ‘ ”” ’ .\*? (~[””]\*) ‘ ”” ’ -> skip;

Làm lại:

COMMENT : '””' COMMENT\_BODY\*? '””' -> skip;

fragment COMMENT\_BODY: ~[“] | [“] ~[“] | [“] EOF;

1. Hãy viết văn phạm với đặc tả như sau: Có 3 loại toán tử, các toán tử này đều là toán tử 2 ngôi (Có 2 toán hạng), @, ^, ?

|  |  |  |
| --- | --- | --- |
| Toán tử | Độ ưu tiên | Tính kết hợp |
| @ | 3 | Kết hợp phải |
| ^ | 2 | Không kết hợp |
| ? | 1 | Kết hợp trái |

Token INT là các số nguyên integer. Nếu thêm dấu () ở bên ngoài biểu thức thì biểu thức đó có độ ưu tiên cao nhất.

Đáp án:

exp -> exp1 @ exp | exp1

exp1 -> exp2 ^ exp2 | exp2

exp2 -> exp2 ? exp3 | exp3

exp3 -> INT | (exp)

Sai thứ tự ưu tiên

1. Hãy viết mô tả văn phạm của ngôn ngữ sau: (Phần này nhớ không kĩ nên mọi người thông cảm).

Có một ngôn ngữ có cách khai báo kiểu array như sau: array [-1 .. 1,0 .. 1,1 .. 2] of type ;

Trong đó array là tên định danh của kiểu array. Type bao gồm các kiểu: integer, float và kiểu array.

Đáp án: Đáp án viết bằng biểu thức nhưng thầy yêu cầu viết bằng ANTLR, mọi người cố gắng chuyển nó qua ANTLR nha.

ARRAY, OF, COMMA, INT, FLOAT, SIGN, INTLIT, [, ]

array -> ARRAY dsmien OF type

dsmien -> “[” cacmien “]”

cacmien -> mien COMMA cacmien | mien

mien -> sn “..” sn

sn -> SIGN INTLIT| INTLIT

type -> INT | FLOAT | array

1. Giải quyết vấn đề đa thừa kế trong Python3. Ta có hình sau biểu diễn sự đa thừa kế của các lớp.

A

B

E

D

C

G

Đáp án: Đáp án này có ghi lại nhưng sau đó thầy nói nó không đúng vì vậy mọi người chỉ nên coi đây là tham khảo. Mọi người cần tìm hiểu kĩ lại những quy luật của đa thừa kế trong Python3. Nên nhớ Python2 và Python3 khác nhau rất nhiều.

L[A] = [A]

L[B] = B + [A]

* [B,A]

L[C] = C + L[B]

* [C,B,A]

L[D] = [D,A]

L[E] = [E,A]

L[F] = F + new(L[D],L[E])

new([D,A],[E,A])

F[D,E,A]

G = G + new(L(C),L(D),L(F))

[C,B,A],[D,A],[F,D,E,A]

1. Hãy viết output của biểu thức sau: 3 – 5 + 4\*6

Dấu cộng là hàm Add(toán hạng trái,toán hạng phải).

Dấu trừ là hàm Minus(toán hạng trái, toán hạng phải).

Dấu nhân là hàm Mul(toán hạng trái, toán hạng phải).

Số nguyên là hàm Lit(số nguyên)

Đáp án: Add(Minus(Lit(3),Lit(5)),Mul(Lit(4),Lit(6)))

1. Cách để hiện thực ngôn ngữ lập trình Python là gì ? (Câu này thầy không trả lời mà chỉ yêu cầu về nhà tìm hiểu.)

Gợi ý: Xem lại bài đầu tiên học về trình biên dịch và trình thông dịch. Hãy tìm hiểu xem Python là trình biên dịch hay trình thông dịch. Nếu là trình thông dịch thì là thông dịch theo dòng hay là trình thông dịch toàn chương trình. Phần này mình không nhớ rõ lắm. Mọi người cố gắng giải thích được nguyên nhân tại sao thì mới được tính điểm câu này.

1. Viết lập trình hàm: Các câu này thầy không sửa nha mọi người. Mọi người cẩn thận vì các câu này có thể có liên quan tới bài tập lớn đang làm hiện tại.
2. Cho một hàm có tên là HAM với tham số đầu vào là một danh sách và một số nguyên. Hàm đó có nhiệm vụ như sau:

List([1,2,3],3) => [(1,3),(2,3),(3,3)]

Hãy sửa lại ở chỗ vị trí x để biểu diễn hàm trên. Viết càng được nhiều cách thì càng được nhiều điểm.

def HAM(a: List(),b: Int)

comb(x)

return (x,b)

return

đệ quy:

def HAM(lst,b):

  if len(lst) == 0:

    return []

  return [(lst[0],b)] + HAM(lst[1:],b)

print(HAM(a,b))

list comprehen

[(x,b) for x in a]

Bậc cao

Comb = list(map( lambda x: (x,b) , a))

1. Hãy viết hàm để tính phép tính sau: Tính tích của các số nguyên trong 1 danh sách. Ví dụ: Ta có danh sách [1.5, 2, “b”, “cd”, 3]. Tích của các số nguyên là 2 x 3 = 6.

Def mul\_int( lst):  
 lst\_int = list(filter(lambda x: ,lst))  
 return x\*1 for x in lst\_int

from functools import reduce

def  mul\_int( lst):

  lst\_int = list(filter(lambda x: isinstance(x, int),lst))

  return reduce(lambda x, y: x\*y,lst\_int)

print(mul\_int( lst))

1. Viết các mô tả thích hợp (token và biểu thức chính quy) trên ANTLR để mô tả token INTLIT, biểu diễn một hằng nguyên thập phân trên ngôn ngữ C, INTLIT là một chuỗi các ký số từ 0 đến 9 nhưng không bắt đầu bằng ký tự số 0 trừ trường hợp bằng 0 (chỉ ghi một số 0). Ví dụ 1023, 900, 0 là các chuỗi hợp lệ ứng với token INTLIT trong khi các chuỗi sau không hợp lệ: 00 (không bắt đầu bằng số 0 nên chuỗi này sẽ tạo ra 2 token INTLIT(0), 0123 (không bắt đầu là số 0 nên chuỗi này sẽ tạo ra 2 token INLIT(0) và INTLIT (123), 123a4 (không có ký tự a nên sẽ tạo ra token INTLIT(123) và báo lỗi ở a). Sinh viên chỉ cần mô tả cho token INTLIT, không cần xử lý lỗi.

* INTLIT: [0] | ([1-9]+ [0]\*)+;
* INTLIT : [0] | [1-9]+ | [1-9]+[0-9]\*

1. Viết các mô tả thích hợp (token và biểu thức chính quy) trên ANTLR để mô tả token STRING, biểu diễn cho một hằng chuỗi trên ngôn ngữ X. STRING là một chuỗi bắt đầu bằng 2 dấu “, kết thúc cũng bằng dấu “. Và ở giữa là một chuỗi (có thể rỗng) các ký tự bất kỳ nhưng không được phép có 2 dấu “ liên tiếp hoặc 1 dấu “ nằm ở cuối. Ví dụ các chuỗi hợp lệ : “ “ “abcde “ f “ “, “ “ ??!’ abc” mdm” dfg “ “. Các chuối sai “ “ abc” “ “ “ (chỉ có “ “ abc “ “ là tạo thành STRING), “ “ abc” “ “ (chỉ có “ “ abc “ “ là tạo thành STRING).

* STRING : ‘ ““ ‘ (.)\*? ~[“”] & ~[ [(.)\*?][“] ] ‘ ”” ‘

1. Cho biểu thức có các phép toán nhị phân @, ^ và ? trong đó @ có độ ưu tiên cao nhất và có tính kết hợp phải, phép toán ^ có độ ưu tiên trung bình và không có tính kết hợp, phép toán ? có độ ưu tiên thấp nhất và có tính kết hợp trái. Toán hạng nguyên tố là số nguyên (INT) và dấu () cố thể được dùng để thay đổi độ ưu tiên và tính kết hợp. Hãy viết băn phạm cho biểu thức dùng dạng BNF?
2. Một kiểu trên ngôn ngữ TEST có thể là int, float hoặc kiểu array. Một kiểu array trên TEST bắt đầu bằng từ khóa array theo sau là một danh sách miền, kế đến là từ khóa of và cuối cùng là một kiểu. Một danh sách miền bắt đầu bằng dấu ‘[‘, kết thúc bằng một dấu ‘]’ và ở giữa là các miền cách nhau bằng dấu ‘ ,’. Phải có ít nhất một miền trong danh sách miền. Mỗi miền được viết bằng 2 số nguyên cách nhau bởi hai chấm liên tiếp ‘..’. Một số nguyên có thể là số dương (chỉ gồm các ký số) hoặc số âm (bắt đầu bằng dấu ‘-‘ theo sau là các ký số).

Array [1..3,-2..4,15..20] of array [-13..15] of int

Biểu diễn một array có 3 miền con với kiểu thành phần là một array có 1 miền với kiểu thành phần là int.

Hãy viết văn phạm và các token cần thiết cho một kiểu array trên TEST dùng ANTLR?

* array: ‘array’ daymien ‘of’ type
* daymien :‘[‘ mien (COMMA mien)\* ‘]’
* mien: INTLIT ‘..’ INTLIT
* INTLIT: [-]?[0-9]+
* COMMA: ‘,’
* type: INT | FLOAT | array
* INT: ‘int’
* FLOAT: ‘float’

II. Phần câu hỏi lý thuyết:

1. (a) Cho các lớp A, B (lớp con của A), C và visitor được khai báo như sau:

|  |  |
| --- | --- |
| Class A:  Def accept (self,v):  v. visitA(self)  Class B(A):  Def accept (self,v):  v. visitC(self)  Class C:  Def accept (self,v):  v. visitC(self) | Class Visitor:  Def visitA(self,x):  Return x.accept(self)  Def visitB(self,x):  pass  Def visitC(self,x):  pass |

Hỏi lệnh gọi x.accept(self) trong phương thức visitA của class Visitor có thể gọi (gián tiếp) đến các phương thức nào của class Visitor? Giải thích.

(b) Hãy giải thích giải pháp giải quyết “vấn đề kim cương” (diamond program) tên Python3 và minh họa cho giải thích trên thông qua xác định chuỗi tìm kiếm cha của đoạn mã Python.

|  |  |
| --- | --- |
| Class A:  Def foo(self):  Print (“A”)  Class B(A):  pass  Class C(B):  Pass  Class D(A):  Def foo(self):  Print (“D”) | Class E(A):  Pass  Class F(D,E):  Def foo(self):  Print (“F”)  Class G(C,F,D):  Pass  x=G()  x.foo() #1 |

1. (a) Thế nào là closure trên lập trình hàm? Cho một ví dụ để giải thích?

(b) Viết hàm subsum nhận vào một danh sách lst và trả về một danh sách mà mỗi phần tử là tổng của phần tử tương ứng trong lst với tất cả các phần tử đứng trước trong lst. Ví dụ subsum([1,3,4,5]) sẽ trả về [1,4,8,13] trong đó 1=1, 4=1+3, 8=1+3+4, và 13 = 1+3+4+15. Yêu cầu dùng cho một trong hai phương pháp: dùng đệ quy hoặc dùng hàm bậc cao. ?

* def subsum (sum:Int,lst:List[Int]) : List[Int] =lst match

case List() => false

case head :: tail

=>if(sum == null) sum = head

else sum = sum + head

subsum(sum,tail)

**Giải đề thi (đề ảnh)**

**Câu 1:**

INTLIT: ‘0’ | [1-9] [0-9]\*;

**Câu 2:**

STRING: ‘””’ STRING\_BODY\*? ‘””’;

Fragment STRING\_BODY: ~[“] | [“] ~[“] | [“] EOF;

**Câu 3:**

exp: exp ? exp1 | exp1;

exp1: exp2 ^ exp2 | exp2;

exp2: exp3 @ exp2 | exp3;

exp3: INT | ‘(‘ exp ‘)’;

**Câu 4:**

INTLIT: ‘0’ | [1-9] [0-9]\*;

type: ‘int’ | ‘float’ | ‘aray’;

int : INTLIT | ‘-‘ INTLIT

array: ‘array’ ‘[‘ dsmien ‘]’ ‘of’ type;

dsmien: mien ‘,’ dsmien | mien;

mien: int ‘..’ int;

**Câu 5:**

**a)**

**b)**

L(A) = [A,o]

L(B) = [B,A,o]

L(C) = [C,B,A,o]

L(D) = [D,A,o]

L(E) = [E,A,o]

L(F) = [F] + merge(L(D), L(E), [D,E]) = [F] + merge([D,A,o], [E,A,o], [D,E]) = [F,D,E,A,o]

L(G) = [G] + merge(L(C), L(F), L(D), [C,F,D]) = [G] + merge([C,B,A,o], [F,D,E,A,o], [D,A,o]) = [G,C,B,F,D,E,A,o]

**x = G()**

**x.foo() sẽ gọi hàm print("F") của class F**

**Câu 6:**

**a)**

CLOSURE LÀ SỰ KẾT HỢP ĐƠN GIẢN GIỮA MỘT HÀM VÀ MÔI TRƯỜNG NƠI HÀM ĐƯỢC KHAI BÁO VÀ VÀO LÚC HÀM ĐƯỢC CHẠY

**Vd:**

def g(x):

def inner(y):

return x\*y

return inner

**b)**

**Kiểu đệ quy:**

def subsum(lst):

if len(lst) == 0:

return []

elif len(lst) == 1:

return [lst[0]]

return subsum(lst[:-1]) + [subsum(lst[:-1])[-1] + lst[-1]]

**Kiểu hàm bậc cao:**

from functools import reduce

def subsum(lst):

if len(lst) == 0:

return []

return reduce(lambda x, y: x + [x[-1] + y],lst[1:], [lst[0]])