

Đại học Bách khoa TP.HCM  
Khoa Khoa học & Kỹ thuật Máy tính

# Nhiệm vụ 3

Bộ kiểm tra tĩnh

Tác giả

TS Nguyễn Hứa Phùng

Ngày 15 tháng 11 năm 2020

---

**Trang 2****Nội dung**

|  |   |
|--|---|
| 1 Đặc điểm kỹ thuật                                      | 2 |
| 2 Bộ kiểm tra tĩnh                                       | 2 |
| 2.1 Biến / Hàm / Tham số được khai báo lại: . . . . .    | 3 |
| 2.2 Định danh / Chức năng không được khai báo: . . . . . | 3 |
| 2.3 Loại Không thể Suy ra: . . . . .                     | 3 |
| 2.4 Loại Không khớp trong câu lệnh: . . . . .            | 4 |
| 2.5 Loại Không khớp trong Biểu thức: . . . . .           | 5 |
| 2.6 Không có điểm vào: . . . . .                         | 5 |
| 3 lần nộp  | 5 |
| 4 Đạo văn  | 5 |
| 5 Nhật ký thay đổi                                       | 6 |

---

**Trang 3****Nhiệm vụ 3****phiên bản 1.0**

Sau khi hoàn thành nhiệm vụ này, bạn sẽ có thể

- giải thích các nguyên tắc cách trình biên dịch có thể kiểm tra một số ràng buộc ngữ nghĩa như khả năng tương thích kiểu, ràng buộc phạm vi, ... và
- viết một chương trình Python trung bình (300-500LOC) để thực hiện điều đó.

**1 Đặc điểm kỹ thuật**

Trong bài tập này, bạn được yêu cầu viết một trình kiểm tra tĩnh cho một chương trình được viết bằng BKIT. Để hoàn thành bài tập này, bạn cần:

- Đọc kỹ đặc tả của ngôn ngữ BKIT
- Tải xuống và giải nén tệp Assign3.zip
- Nếu bạn tự tin về Bài tập 2 của mình, hãy sao chép BKIT.g4 của bạn vào src / main / bkit / parser và ASTGeneration.py của bạn vào src / main / bkit / astgen và bạn có thể kiểm tra Bài tập của mình -ment 3 sử dụng đầu vào BKIT giống như ba bài kiểm tra đầu tiên (400-402).
- Mặt khác (nếu bạn không hoàn thành Bài tập 2 hoặc bạn không tự tin vào

Bài tập 2), đừng lo lắng, chỉ cần nhập AST làm đầu vào cho bài kiểm tra của bạn (như bài kiểm tra 403-405).

- Sửa đổi StaticCheck.py trong src / main // checker để triển khai trình kiểm tra tĩnh và sửa đổi CheckSuite.py trong src / test để triển khai 100 testcase để kiểm tra mã của bạn.

## 2 Bộ kiểm tra tĩnh

Trình kiểm tra tĩnh đóng một vai trò quan trọng trong các trình biên dịch hiện đại. Nó kiểm tra trong quá trình biên dịch thời gian nếu một chương trình tuân theo các ràng buộc ngữ nghĩa theo ngôn ngữ cụ thể sự. Trong bài tập này, bạn được yêu cầu triển khai trình kiểm tra tĩnh cho ngôn ngữ BKIT.

Đầu vào của bộ kiểm tra nằm trong AST của chương trình BKIT, tức là đầu ra của nhiệm vụ

2. Đầu ra của trình kiểm tra là không có gì nếu đầu vào được kiểm tra là chính xác, nếu không, là lỗi thông báo được phát hành và trình kiểm tra tĩnh sẽ dừng ngay lập tức

Đối với mỗi lỗi ngữ nghĩa, sinh viên nên ném ngoại lệ tương ứng được đưa ra trong StaticError.py bên trong thư mục src / main / bkit / checker / để đảm bảo rằng nó sẽ được in ra

2

---

## Trang 4

giống như mong đợi. Mọi test-case đều có nhiều nhất một loại lỗi. Các ràng buộc về ngữ nghĩa yêu cầu kiểm tra trong nhiệm vụ này như sau.

### 2.1 Biến / Hàm / Tham số được khai báo lại:

Mã định danh phải được khai báo trước khi sử dụng. Tuy nhiên, khai báo phải là duy nhất trong phạm vi của nó. Nếu không, Redeclareds ngoại lệ (<kind>, <identifier>) sẽ được phát hành, trong đó <kind> là loại <identifier> (Biến / Hàm / Tham số) trong decla thứ hai-khẩu phần ăn. Phạm vi của một định danh (biến, tham số) được mô tả không chính thức như trong Phần 5 của đặc điểm kỹ thuật BKIT. Tất cả các khai báo hàm đều nằm trong phạm vi toàn cục và phạm vi của chúng là toàn bộ chương trình. Điều đó có nghĩa là một hàm có thể được gọi trước khi khai báo nó. Tất cả function tên là duy nhất có nghĩa là một tên hàm không thể giống với bất kỳ tên chung nào khác biến, tên hàm định sẵn hoặc các tên hàm khác.

### 2.2 Định danh / Chức năng không được khai báo:

Ngoại lệ "Chưa khai báo (<kind>, <identifier>) được phát hành khi có một <identifier> được sử dụng nhưng không thể tìm thấy khai báo của nó. Giá trị nhận dạng có thể là một biến hoặc tham số hoặc chức năng. Loại Hàm được sử dụng khi mã định danh được sử dụng làm tên hàm của một gọi hàm nhưng không có một hàm nào được khai báo trong chương trình. Định danh loại

được sử dụng trong các trường hợp khác.

### 2.3 Loại Không thể Suy ra:

BKIT không yêu cầu khai báo kiểu biến hoặc hàm nhưng có thể suy ra các loại của chúng. Nếu một số nhận dạng được sử dụng nhưng kiểu của nó vẫn chưa được suy ra, ngoại lệ `TypeCannotBeInferred` (<câu lệnh>) sẽ được phát hành. Để suy ra loại của một biến hoặc một chức năng, BKIT đọc chương trình từ đầu đến cuối và áp dụng như sau quy tắc:

- Khi một biến được khởi tạo trong phần khai báo của nó, kiểu của biến cũng là loại chữ được khởi tạo.
- Loại mã định danh (biến, tham số hoặc hàm) phải được suy ra trong sự xuất hiện đầu tiên của việc sử dụng số nhận dạng và không thể thay đổi. Nếu loại của nó không thể được suy ra trong lần sử dụng đầu tiên, câu lệnh trong cùng chứa lần sử dụng đầu tiên của định danh được gửi với ngoại lệ.
- Nếu một biểu thức có thể được suy ra cho một số kiểu nhưng một số thành phần của nó không thể được suy ra cho bất kỳ kiểu nào, câu lệnh trong cùng chứa thành phần kiểu chưa được giải quyết-nent sẽ được liên kết với ngoại lệ. Ví dụ, biểu thức ở bên phải phía bên tay của câu lệnh  $y = a + \text{foo}(x)$  có thể được suy ra để nhập `int` làm kết quả `of +` thuộc kiểu `int` và `y`, `a` và kiểu trả về của `foo` cũng có thể được suy ra là kiểu `int`,

3

---

## Trang 5

nhưng chúng ta không thể suy ra kiểu của `x`, khi đó ngoại lệ được đưa ra với phép gán tuyên bố.

- Một câu lệnh gọi đến một hàm chưa được giải quyết kiểu hợp lệ khi tất cả các kiểu tham số của nó có thể được suy ra bởi các kiểu đối số tương ứng và kiểu trả về của nó có thể là `VoidType`. Nếu tồn tại ít nhất một tham số kiểu chưa được giải quyết, ngoại lệ được nêu ra với câu lệnh cuộc gọi. Lưu ý rằng nếu số lượng các đối số không giống với số lượng đối số, ngoại lệ có liên quan trong Phần 2,4 được nâng lên.
- Một lời gọi hàm đến một hàm loại chưa được giải quyết là hợp lệ nếu tất cả các loại tham số của nó và loại trả lại có thể được giải quyết. Nếu không, câu lệnh trong cùng chứa lệnh gọi hàm được liên kết với ngoại lệ. Lưu ý rằng nếu số lượng các đối số không giống với số lượng đối số, ngoại lệ có liên quan trong Phần

2,5 được nâng lên.

- Kiểu của cả hai vế của một phép gán phải giống nhau (tức là cùng kiểu vô hướng hoặc cùng một loại phân tử và cùng một danh sách các kích thước cho kiểu mảng) để nếu một bên đã giải quyết kiểu của nó, mặt còn lại có thể được suy ra cùng kiểu. Nếu cả hai bên không thể được giải quyết các loại của họ, ngoại lệ được nêu ra với nhiệm vụ.
- Đối với mỗi câu lệnh, tất cả các biến xuất hiện trong câu lệnh phải có kiểu được giải quyết nếu không, câu lệnh trong cùng chứa biến kiểu chưa được giải quyết sẽ là liên quan đến ngoại lệ được nêu ra.

#### 2.4 Loại Không khớp trong Tuyên bố:

Một câu lệnh phải tuân theo các quy tắc kiểu tương ứng cho các câu lệnh, nếu không thìception `TypeMismatchInStatement` (<câu lệnh>) được phát hành.

Các quy tắc loại cho các câu lệnh như sau:

- Kiểu của biểu thức điều kiện trong câu lệnh `if` phải là kiểu boolean.
- Loại biến chỉ số, biểu thức 1 và biểu thức 3 trong câu lệnh `for` phải là số nguyên trong khi kiểu biểu thức 2 là boolean.
- Loại biểu thức điều kiện trong câu lệnh `do while` và `while` phải là kiểu boolean.
- Đối với một bài tập, phần bên tay trái (LHS) có thể thuộc bất kỳ kiểu nào ngoại trừ `VoidType`. Phía bên tay phải (RHS) là cùng loại với bên tay phải của LHS.
- Đối với một câu lệnh gọi <tên phương thức> (<args>), callee phải có `VoidType` là loại trả lại. Số lượng đối số và số lượng tham số phải là tương tự. Ngoài ra, loại của mỗi đối số phải giống với đối số tương ứng tham số.

- Đối với câu lệnh `return`, nếu kiểu trả về của hàm kèm theo là `VoidType`, thì biểu thức trong câu lệnh trả về phải trống. Nếu không, loại lợi nhuận biểu thức phải giống với kiểu trả về của hàm.

#### 2.5 Loại Không khớp trong Biểu thức:

Một biểu thức phải tuân theo các quy tắc kiểu cho các biểu thức, nếu không thì ngoại lệ `TypeMismatchInExpression` (<expression>) được phát hành.

Các quy tắc kiểu cho biểu thức như sau:

- Đối với mảng lập chỉ mục E [E1] ... [En], E phải thuộc kiểu mảng có n kích thước và E1 ... En phải là số nguyên.
- Đối với một biểu thức nhị phân và một ngôi, các quy tắc kiểu được mô tả trong đặc tả BKIT-cation.
- Đối với một lệnh gọi hàm <tên hàm> (<các đơn vị>), số lượng các tham số thực tế phải giống với tham số chính thức của hàm tương ứng.  
Kiểu của mỗi đối số phải giống với kiểu của tham số tương ứng.

## 2.6 Không có Điểm vào:

Phải có một hàm có tên chính trong chương trình BKIT. Nếu không, ngoại lệ-  
tion NoEntryPoint () được phát hành.

## 3 lần nộp

Bài tập này yêu cầu bạn gửi 2 tệp: StaticCheck.py chứa lớp Static-  
Trình kiểm tra với kiểm tra phương pháp nhập và CheckSuite.py chứa 100 (200 cho quà tặng  
sinh viên) tù đưng thử nghiệm.

Tệp StaticCheck.py và CheckSuite.py phải được gửi trong "Bài tập 3  
Nộp hồ sơ".

Thời hạn được thông báo trên trang web của khóa học và đó cũng là nơi bạn PHẢI  
gửi mã của bạn.

## 4 Đạo văn

- Bạn phải tự mình hoàn thành bài tập và không để công việc của bạn bị  
một người nào khác.

Nếu bạn vi phạm bất kỳ yêu cầu nào, bạn sẽ bị trừng phạt bởi nội quy của trường đại học vì tội đạo văn.

## 5 Nhặt ký thay đổi

