

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUY NHƠN

ĐỒ ĐĂNG KHOA

**ĐỘ TƯƠNG TỰ VỀ HÀNH VI CỦA CHƯƠNG TRÌNH VÀ LÀM
THỰC NGHIỆM.**

Chuyên ngành: Khoa học máy tính
Mã số: 60 48 01 01

TÓM TẮT LUẬN VĂN THẠC SĨ.....

Bình Định - Năm 2018

Công trình được hoàn thành tại
TRƯỜNG ĐẠI HỌC QUY NHƠN

Người hướng dẫn: **Ts. PHẠM VĂN VIỆT**

Phản biện 1:
Phản biện 2:

Luận văn được bảo vệ tại Hội đồng đánh giá luận văn thạc sĩ chuyên ngành Khoa học máy tính, ngày ... tháng ... năm 2018 tại Trường Đại học Quy Nhơn.

Có thể tìm hiểu luận văn tại:

-, Trường Đại học Quy Nhơn

Mục lục

LỜI CAM ĐOAN	5
LỜI CẢM ƠN	6
DANH MỤC CÁC KÝ HIỆU	7
DANH MỤC CÁC BẢNG, BIỂU, SƠ ĐỒ, HÌNH VẼ	8
Chương 1 GIỚI THIỆU	9
I Lý do chọn đề tài, ngữ cảnh bài toán	9
II Những nghiên cứu có liên quan	10
Chương 2 KIẾN THỨC CƠ SỞ	11
I Một số khái niệm, định nghĩa	11
II Kỹ thuật lập trình trên C sharp	11
III Một số kỹ thuật sinh testcase	11
1 Kỹ thuật Dynamic symbolic execution (DSE)	11
Chương 3 ĐỘ TƯƠNG TỰ HÀNH VI CỦA CHƯƠNG TRÌNH	12
I Một số khái niệm, định nghĩa	12
1 Hành vi của chương trình (Program Execution)	12
2 Tương đương hành vi (Behavioral Equivalence)	12
3 Sự khác biệt về hành vi (Behavioral Difference)	13
4 Tương tự hành vi (Behavioral Similarity)	13
II Một số phép đo độ tương tự hành vi dựa trên testcase	14
1 Kỹ Thuật Random Sampling (RS)	14
2 Kỹ Thuật Single Program Symbol Execution (SSE)	14
III So sánh các phép đo và phương pháp kết hợp	14
1 Kỹ Thuật Paired Program Symbolic Execution (PSE)	14
Chương 4 CÀI ĐẶT CÁC PHÉP ĐO VÀ KẾT QUẢ THỰC NGHIỆM	15
I Cài đặt các phép đo độ tương tự hành vi	15
II Thực nghiệm và đánh giá kết quả các phép đo	15
KẾT LUẬN VÀ KIẾN NGHỊ	16

LỜI CAM ĐOAN

Tôi xin cam đoan: Luận văn này là công trình nghiên cứu thực sự của cá nhân, được thực hiện dưới sự hướng dẫn khoa học của Tiến sĩ PHẠM VĂN VIỆT.

Các số liệu, những kết luận nghiên cứu được trình bày trong luận văn này trung thực và chưa từng được công bố dưới bất cứ hình thức nào.

Tôi xin chịu trách nhiệm về nghiên cứu của mình.

LỜI CẢM ƠN

Tôi xin chân thành cảm ơn sự hướng dẫn, chỉ dạy và giúp đỡ tận tình của các thầy cô giảng dạy sau đại học - Trường đại học Quy Nhơn.

Đặc biệt, tôi cảm ơn thầy Ts.Phạm Văn Việt, giảng viên bộ môn Công nghệ phần mềm, khoa Công nghệ thông tin, Trường Đại học Quy Nhơn đã tận tình hướng dẫn truyền đạt những kiến thức và kinh nghiệm quý báu để giúp tôi có đầy đủ kiến thức và nghị lực hoàn thành luận văn này.

Và tôi xin cảm ơn bạn bè, đồng nghiệp và những người thân trong gia đình đã tin yêu, động viên giúp tôi thêm nghị lực trong quá trình học tập và nghiên cứu.

Mặc dù đã cố gắng rất nhiều trong việc thực hiện luận văn, song với thời gian có hạn, nên luận văn không thể tránh khỏi những thiếu sót và chưa hoàn chỉnh. Tôi rất mong nhận được ý kiến đóng góp của quý Thầy Cô và các bạn.

Một lần nữa, tôi xin chân thành cảm ơn!

HỌC VIÊN

Đỗ Đăng Khoa

DANH MỤC CÁC KÝ HIỆU

DANH MỤC CÁC BẢNG, BIỂU, SƠ ĐỒ, HÌNH VẼ

Chương 1

GIỚI THIỆU

I Lý do chọn đề tài, ngữ cảnh bài toán

Trong những năm gần đây, xu hướng đào tạo lập trình viên nói riêng và công nghệ phần mềm nói chung đang ngày càng trở nên phổ biến. Các trường đại học và một số trung tâm đào tạo đã cho ra đời nhiều chương trình đào tạo phong phú về nội dung, đa dạng về hình thức và thu hút được nhiều sự quan tâm.

Trong mỗi khóa học, số lượng học viên thường có hàng trăm, hàng ngàn người tham gia, nhưng chỉ một vài giáo viên giảng dạy. Trong đó, chất lượng việc quản lý, truyền đạt nội dung của giáo viên và mức độ hiểu biết, nắm bắt nội dung chương trình học của học viên là yêu cầu tất cả các khóa học cần đạt được. Để đạt được điều đó, một yêu cầu bắt buộc đó là giáo viên phải đọc và hiểu tất cả các đoạn code của học sinh, nhưng công việc này lại tốn quá nhiều thời gian. Nếu bỏ qua hoặc trì hoãn các công việc như vậy thì người dạy không thể theo dõi được quá trình học tập của người học. Về phía người học, yêu cầu đặt ra là phải tiến bộ theo thời gian, nắm vững lý thuyết và thành thạo kỹ năng lập trình. Những không phải lúc nào gặp khó khăn người học đều có sự hỗ trợ kịp thời từ giảng viên. Họ có thể nhờ sự giúp đỡ từ đồng nghiệp, bạn bè, nhưng những người được nhờ hiups đỡ chưa chắc đã đủ trình độ, kinh nghiệm hoặc thời gian để ngồi bên cạnh giúp đỡ người học khi cần.

Để giảm bớt những khó khăn nêu trên, một công cụ hỗ trợ quá trình giảng dạy và học tập của giáo viên và học sinh hiệu quả hơn, tiết kiệm thời gian hơn đó là một công cụ tự động hóa (có thể một phần) việc đánh giá kết quả lập trình của học sinh, cũng như hỗ trợ theo dõi sự tiến bộ của học sinh. Công cụ tự động hóa này sẽ tính toán, định lượng tỷ lệ chính xác sự tương tự về hành vi giữa chương trình của người học và chương trình của người dạy đưa ra trước đó. Dựa trên kết quả, giáo viên sẽ đánh giá được kỹ năng lập trình của học sinh, sự giống nhau giữa hai chương trình càng cao thì tỷ lệ độ tương tự càng cao, điểm số càng cao. Nếu điểm số thấp, người học có thể quay lại kiểm tra để viết mã chương trình đúng hơn, hạn chế được nguy cơ tìm ẩn trong cách viết chương trình của người học.

Những đánh giá này có thể thực hiện được nếu ta đo được độ tương tự giữa các chương trình có độ chính xác cao. Đề tài “Độ tương tự về hành vi của các chương trình và làm thực nghiệm” với mục đích sẽ giải quyết các vấn đề nêu trên.

II Những nghiên cứu có liên quan

Chương 2

KIẾN THỨC CƠ SỞ

I Một số khái niệm, định nghĩa

II Kỹ thuật lập trình trên C sharp

III Một số kỹ thuật sinh testcase

1 Kỹ thuật Dynamic symbolic execution (DSE)

Là một kỹ thuật thu thập các ràng buộc từ các nhánh của chương trình và phủ nhận một phần các ràng buộc để tạo ra dữ liệu đầu vào của chương trình và có độ phủ cao

Chương 3

ĐỘ TƯƠNG TỰ HÀNH VI CỦA CHƯƠNG TRÌNH

I Một số khái niệm, định nghĩa

Để định lượng hai chương trình tương tự nhau, chúng ta định nghĩa các khái niệm về hành vi chương trình và các định nghĩa liên quan đến sự tương tự của hai chương trình, và ví dụ minh họa cho các định nghĩa.

1 Hành vi của chương trình (Program Execution)

Định nghĩa 1 *Hành vi chương trình P là thực hiện hàm: $P \times I \rightarrow O$. Với giá trị đầu vào $i \in I$, giá trị đầu ra $o \in O$. Trong đó I là miền các giá trị đầu vào của chương trình P và O là tập hợp các giá trị đầu ra của chương trình P .*

2 Tương đương hành vi (Behavioral Equivalence)

Định nghĩa 2 *Hai chương trình P_1 và P_2 có cùng một miền các giá trị đầu vào I và tương đương về hành vi nếu $exec(P_1; I) = exec(P_2; I)$, với $\forall i \in I \ exec(P_1; i) = exec(P_2; i)$.*

Ví dụ:

Chương trình 3.1: Tính y, sử dụng hàm switch...case

```
public static int TinhY(int x)
{
    y = 0;
```

```

switch (x) {
  case 1: y += 4; break;
  case 2: y *= 2; break;
  default: y = y * y;
}
return y;
}

```

Chương trình 3.2: Tính y, sử dụng hàm If...else

```

public static int TinhY(int x)
{
  y = 0;
  if (x == 1)
    y += 4;
  else if (x == 2)
    y *= 2;
  else
    y = y * y;
  return y;
}

```

Ví dụ trên cho chúng ta thấy 2 chương trình là tương đương nhau về hành vi. Hai chương trình có giá trị đầu vào là như nhau (cùng kiểu **Int**). Chương trình đầu tiên sử dụng hàm **switch...case**, chương trình tiếp theo sử dụng hàm **if...else** để kiểm tra giá trị đầu vào, tuy cú pháp khác nhau nhưng cách xử lý và trả về kết quả **y** là như nhau.

3 Sự khác biệt về hành vi (Behavioral Difference)

Định nghĩa 3 Hai chương trình P_1 và P_2 có cùng một miền các giá trị đầu vào I và khác biệt về hành vi nếu $exec(P_1; I) \neq exec(P_2; I)$, $\exists i \in I \text{ } exec(P_1; i) \neq exec(P_2; i)$.

4 Tương tự hành vi (Behavioral Similarity)

Định nghĩa 4 Hai chương trình P_1 và P_2 được gọi là tương tự hành vi khi là hai chương trình có cùng miền giá trị đầu vào I và $|I_s| \neq |I|$, trong đó $I_s \subseteq I$, nếu $exec(P_1, I_s) = exec(P_2, I_s)$, và $\forall j \in I \setminus I_s, exec(P_1; j) \neq exec(P_2; j)$.

Ví dụ BS:

Ví dụ cụ thể

II Một số phép đo độ tương tự hành vi dựa trên testcase

Để tính sự tương đồng về hành vi giữa hai chương trình, chúng ta có thể đo bằng cách tính tỷ lệ đầu vào và đầu ra trên cả hai chương trình trên cùng toàn bộ miền giá trị đầu vào. Trong đó một phương pháp tiếp cận đơn giản đó là liệt kê tất cả các dữ liệu trong miền đầu vào và chạy từng đầu vào trên cả hai chương trình để so sánh các kết quả đầu ra. Nhưng việc này sẽ không thực tế hoặc không liên quan đến các chương trình với một miền đầu vào lớn hoặc vô hạn.

Để đo độ tương tự hành vi được chính xác hơn, chúng tôi chạy các dữ liệu đầu vào đại diện thay vì tất cả các dữ liệu đầu vào cho các chương trình. Bằng cách thống nhất lấy mẫu một phần dữ liệu đầu vào từ miền đầu vào, độ tương tự về hành vi sẽ được tính dựa trên tỷ lệ các đầu vào mẫu trên các đầu ra.

Dựa trên DSE, chúng ta có thể tạo ra được dữ liệu đầu vào thử nghiệm có độ bao phủ cao, và sử dụng trong các kỹ thuật đo độ tương tự.

1 Kỹ Thuật Random Sampling (RS)

Định nghĩa

Hai chương trình P_1 và P_2 là hai chương trình có cùng miền giá trị đầu vào I và I_s là tập con các giá trị đầu vào của tập I , và I_a là tập con các giá trị đầu vào của tập I_s , trong đó $\forall i \in I_a$, sao cho $exec(P_1, i) = exec(P_2, i)$ và $\forall j \in I_s \setminus I_a$, $exec(P_1, j) \neq exec(P_2, j)$. Độ đo của kỹ thuật của RS sẽ là $M_{RS}(P_1, P_2) = |I_a|/|I_s|$

2 Kỹ Thuật Single Program Symbol Execution (SSE)

III So sánh các phép đo và phương pháp kết hợp

1 Kỹ Thuật Paired Program Symbolic Execution (PSE)

Chương 4

CÀI ĐẶT CÁC PHÉP ĐO VÀ KẾT QUẢ THỰC NGHIỆM

I Cài đặt các phép đo độ tương tự hành vi

II Thực nghiệm và đánh giá kết quả các phép đo

KẾT LUẬN VÀ KIẾN NGHỊ

PHỤ LỤC

QUYẾT ĐỊNH GIAO LUẬN VĂN

Tài liệu tham khảo