

	<b>VIETTEL AI RACE</b>	TD064
	<b>KIỂM THỬ DÒNG ĐIỀU KHIỂN</b>	Lần ban hành: 1

Trong chương này, chúng ta sẽ tìm hiểu chi tiết về phương pháp kiểm thử dòng điều khiển (control flow testing) nhằm phát hiện các lỗi tiềm ẩn bên trong chương trình/đơn vị chương trình cần kiểm thử. Các lỗi này thường khó phát hiện bởi các kỹ thuật kiểm thử chức năng hay kiểm thử hộp đen được trình bày trong chương 5. Để áp dụng phương pháp này, chúng ta cần phân tích mã nguồn và xây dựng các ca kiểm thử ứng với các dòng điều khiển của chương trình/đơn vị chương trình. Các độ đo hay tiêu chí kiểm thử cho phương pháp này cũng sẽ được giới thiệu.

## 1. Kiểm thử hộp trắng

Kiểm thử hộp trắng sử dụng các chiến lược cụ thể và sử dụng mã nguồn của chương trình/đơn vị phần mềm cần kiểm thử nhằm kiểm tra xem chương trình/đơn vị phần mềm có thực hiện đúng so với thiết kế và đặc tả hay không. Trong khi các phương pháp kiểm thử hộp đen hay kiểm thử chức năng chỉ cho phép phát hiện các lỗi/khiếm khuyết có thể quan sát được, kiểm thử hộp trắng cho phép phát hiện các lỗi/khiếm khuyết tiềm ẩn bên trong chương trình/đơn vị phần mềm. Các lỗi này thường khó phát hiện bởi các phương pháp kiểm thử hộp đen. Khác với các phương pháp kiểm thử hộp đen nơi mà các ca kiểm thử được sinh ra từ đặc tả của hệ thống, các ca kiểm thử trong các phương pháp kiểm thử hộp trắng được sinh ra từ mã nguồn. Kiểm thử hộp đen và kiểm thử hộp trắng không thể thay thế cho nhau mà chúng cần được sử dụng kết hợp với nhau trong một quy trình kiểm thử thống nhất nhằm đảm bảo chất lượng phần mềm. Tuy nhiên, để áp dụng các phương pháp kiểm thử hộp trắng, người kiểm thử không chỉ cần hiểu rõ giải thuật mà còn cần có các kỹ năng và kiến thức tốt về ngôn ngữ lập trình được dùng để phát triển phần mềm, nhằm hiểu rõ mã nguồn của chương trình/đơn vị phần mềm cần kiểm thử. Do vậy, việc áp dụng các phương pháp kiểm thử hộp trắng thường tốn thời gian và công sức nhất là khi chương trình/đơn vị phần mềm có kích thước lớn. Vì lý do này, các phương pháp kiểm thử hộp trắng chủ yếu được sử dụng cho kiểm thử đơn vị [D.95].

Hai phương pháp được sử dụng trong kiểm thử hộp trắng là kiểm thử dòng điều khiển (control flow testing) và kiểm thử dòng dữ liệu (data flow testing). Phương pháp kiểm thử dòng điều khiển tập trung kiểm thử tính đúng đắn của các giải thuật sử dụng trong các chương trình/đơn vị phần mềm. Phương pháp kiểm thử dòng dữ liệu tập trung kiểm thử tính đúng đắn của việc sử dụng các biến dữ liệu sử dụng trong chương trình/đơn vị phần mềm. Trong chương này, chúng ta sẽ tìm hiểu chi tiết về phương pháp kiểm thử dòng điều khiển. Phương pháp kiểm thử dòng dữ liệu sẽ được giới thiệu trong chương 7.

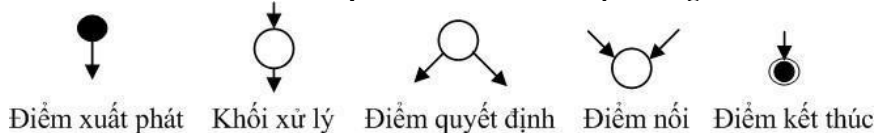
	<b>VIETTEL AI RACE</b>	TD064
	<b>KIỂM THỬ DÒNG ĐIỀU KHIỂN</b>	Lần ban hành: 1

## 2. Đồ thị dòng điều khiển

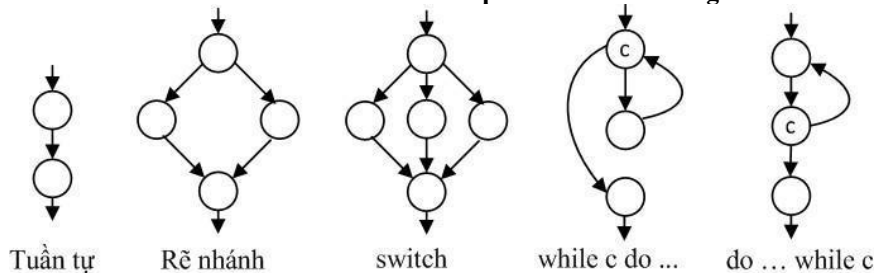
Phương pháp kiểm thử dòng điều khiển dựa trên khái niệm đồ thị dòng điều khiển (control flow graph). Đồ thị này được xây dựng từ mã nguồn của chương trình/đơn vị chương trình. Đồ thị dòng điều khiển là một đồ thị có hướng gồm các đỉnh tương ứng với các câu lệnh/nhóm câu lệnh và các cạnh là các dòng điều khiển giữa các câu lệnh/nhóm câu lệnh. Nếu  $i$  và  $j$  là các đỉnh của đồ thị dòng điều khiển thì tồn tại một cạnh từ  $i$  đến  $j$  nếu lệnh tương ứng với  $j$  có thể được thực hiện ngay sau lệnh tương ứng với  $i$ .

Xây dựng một đồ thị dòng điều khiển từ một chương trình/đơn vị chương trình khá đơn giản. Hình 6.1 mô tả các thành phần cơ bản của đồ thị dòng điều khiển bao gồm điểm bắt đầu của đơn vị chương trình, khối xử lý chứa các câu lệnh khai báo hoặc tính toán, điểm quyết định ứng với các câu lệnh điều kiện trong các khối lệnh rẽ nhánh hoặc lặp, điểm nối ứng với các câu lệnh ngay sau các lệnh rẽ nhánh, và điểm kết thúc ứng với điểm kết thúc của đơn vị chương trình. Các cấu trúc điều khiển phổ biến của chương trình được mô tả trong Hình 6.2. Chúng ta sẽ sử dụng các thành phần cơ bản và các cấu trúc phổ biến này để dễ dàng xây dựng đồ thị dòng điều khiển cho mọi đơn vị chương trình viết bằng mọi ngôn ngữ lập trình.

Hình 6.1: Các thành phần cơ bản của đồ thị chương trình.



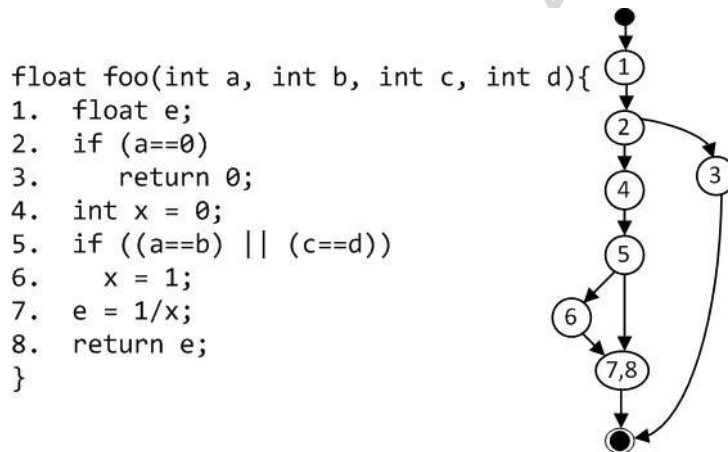
Hình 6.2: Các cấu trúc điều khiển phổ biến của chương trình.



Chúng ta thử xem cách dựng đồ thị dòng điều khiển cho đơn vị chương trình có mã nguồn bằng ngôn ngữ C như Hình 6.3. Chúng ta đánh số các dòng lệnh của đơn vị chương trình và lấy số này làm đỉnh của đồ thị. Điểm xuất phát của đơn vị chương trình ứng với câu lệnh khai báo hàm foo. Đỉnh 1 ứng với câu lệnh khai báo biến  $e$ . Các đỉnh 2 và 3 ứng với câu lệnh if. Đỉnh 4 ứng với câu lệnh khai báo biến  $x$  trong khi các đỉnh 5 và 6 ứng với câu lệnh if. Đỉnh 7,8 đại diện cho hai câu lệnh 7 và 8. Trong trường hợp này, chúng ta không tách riêng thành hai đỉnh vì đây là hai câu lệnh tuần tự nên chúng ta ghép chúng thành một đỉnh nhằm tối thiểu số đỉnh của đồ thị dòng điều khiển. Với cách làm này, chúng ta xây dựng được đồ thị dòng điều khiển với số đỉnh

	VIETTEL AI RACE	TD064
	KIỂM THỬ DÒNG ĐIỀU KHIỂN	Lần ban hành: 1

nhỏ nhất. Chúng ta sẽ sử dụng đồ thị này để phân tích và sinh các ca kiểm thử nên đồ thị càng ít đỉnh thì độ phức tạp của thuật toán phân tích càng nhỏ.



Hình 6.3: Mã nguồn của hàm foo và đồ thị dòng điều khiển của nó.

### 3. Các độ đo kiểm thử

Kiểm thử chức năng (kiểm thử hộp đen) có hạn chế là chúng ta không biết có thừa hay thiếu các ca kiểm thử hay không so với chương trình cài đặt và thiếu thừa ở mức độ nào. Độ đo kiểm thử là một công cụ giúp ta đo mức độ bao phủ chương trình của một tập ca kiểm thử cho trước. Mức độ bao phủ của một bộ kiểm thử (tập các ca kiểm thử) được đo bằng tỷ lệ các thành phần thực sự được kiểm thử so với tổng thể sau khi đã thực hiện các ca kiểm thử. Thành phần liên quan có thể là câu lệnh, điểm quyết định, điều kiện con, đường thi hành hay là sự kết hợp của chúng. Độ bao phủ càng lớn thì độ tin cậy của bộ kiểm thử càng cao. Độ đo này giúp chúng ta kiểm soát và quản lý quá trình kiểm thử tốt hơn. Mục tiêu của chúng ta là kiểm thử với số ca kiểm thử tối thiểu nhưng đạt được độ bao phủ tối đa. Có rất nhiều độ đo kiểm thử đang được sử dụng hiện nay, dưới đây là ba độ đo kiểm thử đang được sử dụng phổ biến nhất trong thực tế [Lee03].

**Độ đo kiểm thử cấp 1 ( $C_1$ ):** mỗi câu lệnh được thực hiện ít nhất một lần sau khi chạy các ca kiểm thử (test cases). Ví dụ, với hàm foo có mã nguồn như trong Hình 6.3, ta chỉ cần hai ca kiểm thử như Bảng 6.1 là đạt 100% độ phủ cho độ đo  $C_1$  với EO (expected output) là giá trị đầu ra mong đợi và RO (real output) là giá trị đầu ra thực tế (giá trị này sẽ được điền khi thực hiện ca kiểm thử).

Bảng 6.1: Các ca kiểm thử cho độ đo  $C_1$  của hàm foo

	<b>VIETTEL AI RACE</b>	TD064
	<b>KIỂM THỬ DÒNG ĐIỀU KHIỂN</b>	Lần ban hành: 1

ID	Inputs	EO	RO	Note
tc1	0, 1, 2, 3	0		
tc2	1, 1, 2, 3	1		

**Độ đo kiểm thử cấp 2 ( $C_2$ ):** các điểm quyết định trong đồ thị dòng điều khiển của đơn vị kiểm thử đều được thực hiện ít nhất một lần cả hai nhánh đúng và sai. Ví dụ, Bảng 6.2 mô tả các trường hợp cần kiểm thử để đạt được 100% độ phủ của độ đo  $C_2$  ứng với hàm foo được mô tả trong Hình 6.3.

**Bảng 6.2:** Các trường hợp cần kiểm thử của độ đo  $C_2$  với hàm foo

Điểm quyết định	Điều kiện tương ứng	Đúng	Sai
2	$a == 0$	tc1	tc2
5	$(a == b) \parallel (c == d)$	tc2	?

Như vậy, với hai ca kiểm thử trong độ đo kiểm thử cấp 1 (tc1 và tc2), ta chỉ kiểm thử được  $3/4 = 75\%$  ứng với độ đo kiểm thử cấp 2. Chúng ta cần một ca kiểm thử nữa ứng với trường hợp sai của điều kiện  $(a == b) \parallel (c == d)$  nhằm đạt được 100% độ phủ của độ đo  $C_2$ . Bảng 6.3 mô tả các ca kiểm thử cho mục đích này.

**Bảng 6.3:** Các ca kiểm thử cho độ đo  $C_2$  của hàm foo

ID	Inputs	EO	RO	Note
tc1	0, 1, 2, 3	0		
tc2	1, 1, 2, 3	1		
tc3	1, 2, 1, 2	Lỗi chia cho 0		

**Độ đo kiểm thử cấp 3 ( $C_3$ ):** Với các điều kiện phức tạp (chứa nhiều điều kiện con cơ bản), việc chỉ quan tâm đến giá trị đúng sai là không đủ để kiểm tra tính đúng đắn của chương trình ứng với điều kiện phức tạp này. Ví dụ, nếu một điều kiện phức tạp gồm hai điều kiện con cơ bản, chúng ta có bốn trường hợp cần kiểm thử chứ không phải hai trường hợp đúng sai như độ đo  $C_2$ . Với các đơn vị chương trình có yêu cầu cao về tính đúng đắn, việc tuân thủ độ đo  $C_3$  là hết sức cần thiết. Điều kiện để đảm bảo độ đo này là các điều kiện con thuộc các điều kiện phức tạp tương ứng với các điểm quyết định trong đồ thị dòng điều khiển của đơn vị cần kiểm thử đều được thực hiện ít nhất một lần cả hai nhánh đúng và sai. Ví dụ, Bảng 6.4 mô tả các trường hợp cần kiểm thử để đạt được 100% độ phủ của độ đo  $C_3$  ứng với hàm foo được mô tả trong Hình 6.3. Như vậy, với ba ca kiểm thử trong độ đo kiểm thử cấp 2 (tc1, tc2 và tc3), ta chỉ kiểm thử được  $7/8 = 87,5\%$  ứng với độ đo kiểm thử cấp 3. Chúng ta cần một ca kiểm thử nữa ứng với trường hợp sai của điều kiện con

	<b>VIETTEL AI RACE</b>	TD064
	<b>KIỂM THỬ DÒNG ĐIỀU KHIỂN</b>	Lần ban hành: 1

cơ bản ( $c == d$ ) nhằm đạt được 100% độ phủ của độ đo  $C_3$ . Bảng 6.5 mô tả các ca kiểm thử cho mục đích này.

**Bảng 6.4: Các trường hợp cần kiểm thử của độ đo  $C_3$  với hàm foo**

Điểm quyết định	Điều kiện tương ứng	Đúng	Sai
2	$a == 0$	tc1	tc2
5	$(a == b)$	tc2	tc3
5	$(c == d)$	?	tc2

**Bảng 6.5: Các ca kiểm thử cho độ đo  $C_3$  của hàm foo**

ID	Inputs	EO	RO	Note
tc1	0, 1, 2, 3	0		
tc2	1, 1, 2, 3	1		
tc3	1, 2, 1, 2	Lỗi chia cho 0		
tc4	1, 2, 1, 1	1		