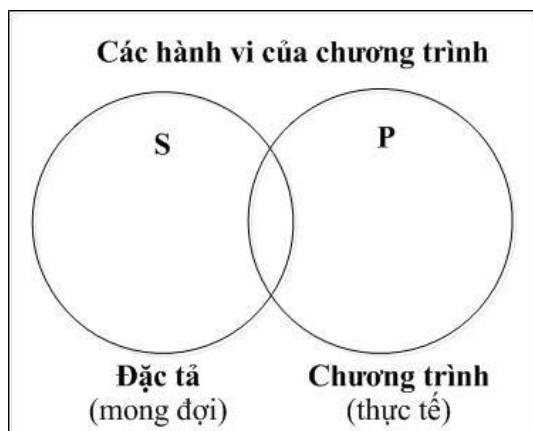


	<b>VIETTEL AI RACE</b>	TD058
	<b>TỔNG QUAN KIỂM THỬ</b>	Lần ban hành: 1

## 1. Mô tả bài toán kiểm thử qua biểu đồ Venn

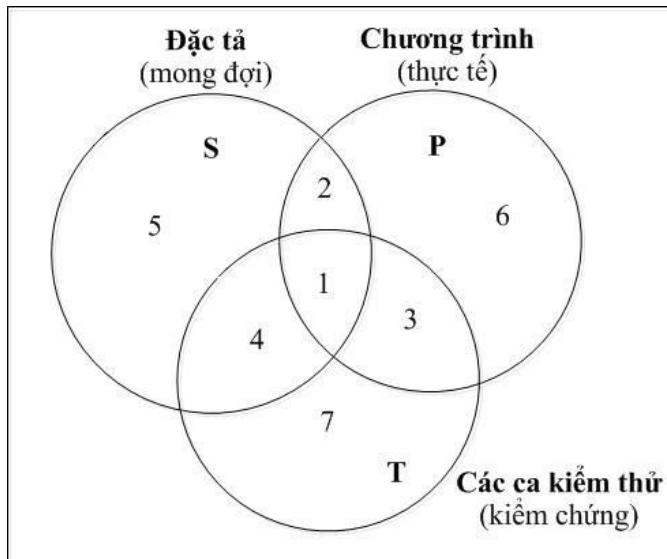
Kiểm thử chủ yếu liên quan tới hành vi của chương trình nơi mà hành vi phản ánh quan điểm về cấu trúc phổ biến đối với các nhà phát triển hệ thống hoặc phần mềm. Sự khác biệt là quan điểm cấu trúc tập trung vào “là cái gì”, còn quan điểm hành vi lại tập trung vào “làm gì”. Một trong những nguyên nhân gây khó cho người kiểm thử là các tài liệu cơ sở thường được viết bởi và viết cho người phát triển. Kết quả là các tài liệu này thường thiên về thông tin cấu trúc và coi nhẹ thông tin về hành vi của chương trình cần kiểm thử. Trong mục này, chúng ta sẽ phát triển một biểu đồ Venn đơn giản nhằm làm sáng tỏ một số vấn đề về kiểm thử. Chi tiết về biểu đồ Venn sẽ được trình bày trong chương 3.



**Hình 1.3: Các hành vi được cài đặt và được đặc tả.**

Xét một vũ trụ của hành vi chương trình cần kiểm thử, lưu ý rằng chúng ta đang quan tâm đến bản chất của việc kiểm thử. Cho trước một chương trình cùng đặc tả của nó. Gọi  $S$  là tập các hành vi được đặc tả và  $P$  là tập các hành vi của chương trình. Hình 1.3 mô tả mối quan hệ giữa vũ trụ các hành vi được lập trình và hành vi được đặc tả. Trong tất cả các hành vi có thể của chương trình, những hành vi được đặc tả nằm trong vòng tròn với nhãn  $S$ , còn những hành vi được lập trình là ở trong vòng tròn với nhãn  $P$ . Từ biểu đồ này, ta thấy rõ các bài toán mà người kiểm thử cần đối mặt là gì. Nếu có hành vi được đặc tả nhưng không được lập trình thì theo thuật ngữ trước đây, đây là những sai lầm về bỏ quên. Tương tự, nếu có những hành vi được lập trình nhưng không được đặc tả, thì điều đó tương ứng với những sai lầm về nhiệm vụ, và chúng tương ứng với những lỗi xuất hiện sau khi đặc tả đã hoàn thành. Tương giao giữa  $S$  và  $P$  là phần đúng đắn, gồm có các hành vi vừa được đặc tả vừa được cài đặt. Chú ý rằng tính đúng đắn chỉ có nghĩa đối với đặc tả và cài đặt và là khái niệm mang tính tương đối.

	<b>VIETTEL AI RACE</b>	TD058
	<b>TỔNG QUAN KIỂM THỬ</b>	Lần ban hành: 1



**Hình 1.4: Các hành vi được cài đặt, được đặc tả và được kiểm thử.**

Vòng tròn mới (vòng tròn  $T$ ) trong hình 1.4 là cho các ca kiểm thử. Lưu ý rằng tập các hành vi của chương trình nằm trọn trong vũ trụ chuyên đề của ta. Ở đây một ca kiểm thử cũng được coi là xác định một hành vi. Xét mối quan hệ giữa  $S$ ,  $P$  và  $T$ . Có thể có các hành vi được đặc tả mà không được kiểm thử (các miền 2 và 5), các hành vi được đặc tả và được kiểm thử (các miền 1 và 4), và các ca kiểm thử tương ứng với các hành vi không được đặc tả (các miền 3 và 7).

Tương tự, có thể có các hành vi được lập trình mà không được kiểm thử (các miền 2 và 6), các hành vi được lập trình và được kiểm thử (các miền 1 và 3), và các ca kiểm thử tương ứng với các hành vi không được lập trình (các miền 4 và 7). Việc xem xét tất cả các miền này là hết sức quan trọng. Nếu có các hành vi được đặc tả mà không có các ca kiểm thử tương ứng, việc kiểm thử là chưa đầy đủ. Nếu có các ca kiểm thử tương ứng với các hành vi không được đặc tả, có thể có hai khả năng: hoặc đặc tả còn thiếu hoặc ca kiểm thử không đảm bảo. Theo kinh nghiệm, một người kiểm thử giỏi sẽ thường phát triển các ca kiểm thử thuộc loại đầu, và đây chính là lý do người kiểm thử cần tham gia vào giai đoạn khảo duyệt đặc tả và thiết kế (xem chương 4).

Ta có thể thấy việc kiểm thử như là công việc của một nghệ nhân: người kiểm thử có thể làm gì để làm cho miền tương giao của các tập (miền 1) là lớn nhất có thể? Làm thế nào để xác định các ca kiểm thử trong tập  $T$ ? Câu trả lời là các ca kiểm thử cần được xác định bởi một phương pháp kiểm thử. Chính khuôn khổ này cho phép ta so sánh tính hiệu quả của các phương pháp kiểm thử khác nhau như sẽ được giới thiệu trong các chương 5, 6 và 7.

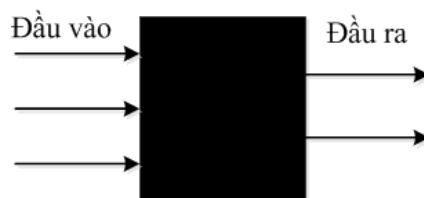
## 2. Việc xác định các ca kiểm thử

Có hai cách tiếp cận cơ bản để xác định các ca kiểm thử là kiểm thử chức năng hay kiểm thử hộp đen (black-box testing) và kiểm thử cấu trúc hay kiểm thử hộp trắng (white-box testing). Mỗi cách tiếp cận có các phương pháp xác định các ca kiểm thử khác nhau và được gọi chung là các phương pháp kiểm thử.

	<b>VIETTEL AI RACE</b>	TD058
	<b>TỔNG QUAN KIỂM THỬ</b>	Lần ban hành: 1

## 2.1 Kiểm thử chức năng

Kiểm thử chức năng (kiểm thử hộp đen) dựa trên quan niệm rằng bất kỳ chương trình nào cũng được coi là một hàm ánh xạ các giá trị từ miền dữ liệu đầu vào tới miền dữ liệu đầu ra của nó. Khái niệm này được dùng chung trong kỹ thuật khi các hệ thống đều được coi là các hộp đen. Chính điều này dẫn đến thuật ngữ kiểm thử hộp đen, trong đó nội dung của hộp đen (việc cài đặt) không được biết hoặc không cần quan tâm, và chức năng của hộp đen được hiểu theo các dữ liệu đầu vào và dữ liệu đầu ra của nó. Trong thực tế, chúng ta thường thao tác hiệu quả với những kiến thức về hộp đen. Chính điều này là trung tâm của khái niệm định hướng đối tượng. Chẳng hạn, hầu hết mọi người lái xe thành thạo với kiến thức hộp đen.



**Hình 1.5: Một hộp đen kỹ thuật.**

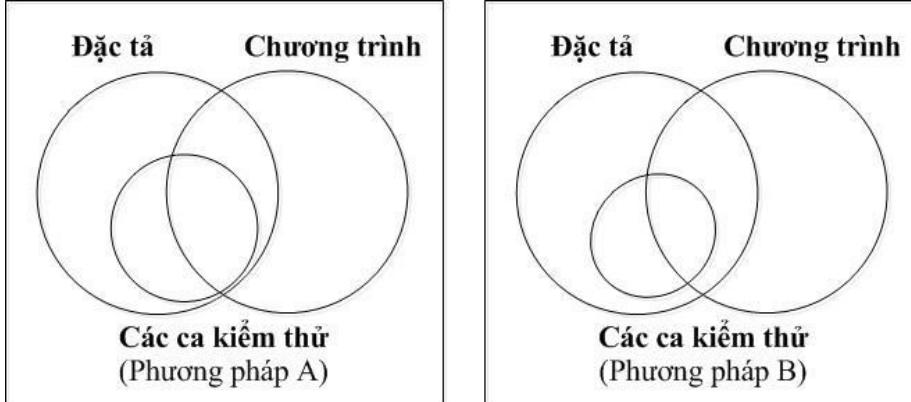
Với cách tiếp cận của kiểm thử chức năng, để xác định các ca kiểm thử, thông tin duy nhất được dùng là đặc tả của phần mềm cần kiểm thử. Có hai lợi điểm chính của các ca kiểm thử được sinh ra bởi cách tiếp cận kiểm thử chức năng: chúng độc lập với việc phần mềm được cài đặt thế nào, và vì thế khi cài đặt thay đổi thì các ca kiểm thử vẫn dùng được, đồng thời các ca kiểm thử được phát triển song song và độc lập với việc cài đặt hệ thống. Do đó, cách tiếp cận này rút gọn được thời gian phát triển của dự án. Điểm yếu của cách tiếp cận này là các ca kiểm thử thường có thể có tính dư thừa đáng kể trong các ca kiểm thử và vẫn đề hô phân cách.

Hình 1.6 mô tả các ca kiểm thử được xác định bởi các phương pháp kiểm thử chức năng khác nhau. Phương pháp A xác định một tập các ca kiểm thử lớn hơn so với phương pháp B. Lưu ý rằng đối với cả hai phương pháp này, tập các ca kiểm thử đều chứa trọn trong tập các hành vi được đặc tả.

Do các phương pháp kiểm thử chức năng đều dựa trên các hành vi đặc tả, các phương pháp này khó có thể xác định được các hành vi không được đặc tả. Trong chương 5 ta sẽ so sánh các ca kiểm thử sinh bởi các phương pháp kiểm thử chức năng khác nhau cho các ví dụ được mô tả trong chương 2.

	<b>VIETTEL AI RACE</b>	TD058
	<b>TỔNG QUAN KIỂM THỬ</b>	Lần ban hành: 1

**Hình 1.6: So sánh các phương pháp xác định các ca kiểm thử chức năng.**



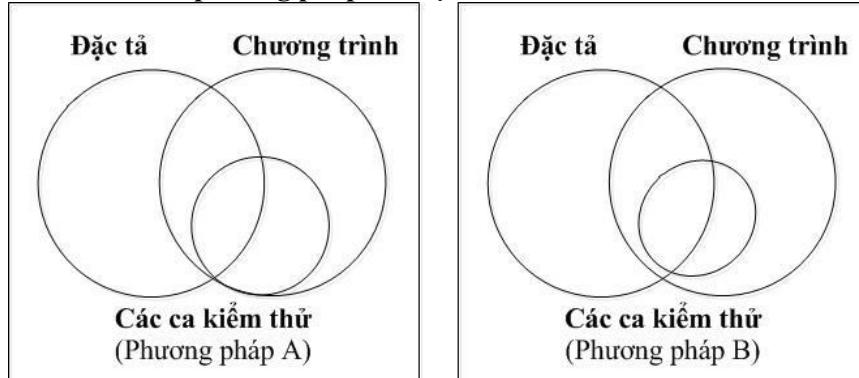
Trong chương 5, chúng ta sẽ khảo sát các cách tiếp cận chủ yếu cho các phương pháp kiểm thử chức năng bao gồm phân tích giá trị biên, kiểm thử tính bền vững, phân tích trường hợp xấu nhất, kiểm thử giá trị đặc biệt, kiểm thử phân lớp tương đương của miền dữ liệu đầu vào, lớp tương đương của miền dữ liệu đầu ra, kiểm thử dựa trên bảng quyết định. Điều xuyên suốt trong các kỹ thuật này là tất cả đều dựa trên thông tin xác định về các thành phần đang được kiểm thử. Cơ sở toán học trình bày trong chương 3 chủ yếu được áp dụng cho cách tiếp cận kiểm thử chức năng.

## 2.2 Kiểm thử cấu trúc

Kiểm thử cấu trúc (kiểm thử hộp trắng) là cách tiếp cận khác để xác định các ca kiểm thử. Biểu tượng hộp trong suốt (hộp trắng) là thích hợp cho cách tiếp cận này vì sự khác nhau cốt lõi của cách tiếp cận này so với kiểm thử hộp đen là việc cài đặt của hộp đen (mã nguồn) được cung cấp và được dùng làm cơ sở để xác định các ca kiểm thử. Việc hiểu biết được bên trong của hộp đen cho phép người kiểm thử dựa trên việc cài đặt để xác định các ca kiểm thử. Kiểm thử cấu trúc đã trở thành chủ đề của một lý thuyết tương đối mạnh. Để hiểu rõ kiểm thử cấu trúc, các khái niệm về lý thuyết đồ thị tuyến tính được trình bày trong chương 3 là cần thiết. Với những khái niệm này, người kiểm thử có thể mô tả chính xác các yêu cầu kiểm thử và hệ thống cần kiểm thử. Do có cơ sở lý thuyết mạnh, kiểm thử cấu trúc cho phép định nghĩa chính xác và sử dụng các độ đo về độ bao phủ. Các độ đo về độ phủ cho phép khẳng định tường minh phần mềm đã được kiểm thử tới mức nào và do đó giúp cho việc quản lý quá trình kiểm thử tốt hơn.

	<b>VIETTEL AI RACE</b>	TD058
	<b>TỔNG QUAN KIỂM THỬ</b>	Lần ban hành: 1

**Hình 1.7: So sánh các phương pháp xác định ca kiểm thử đối với kiểm thử cấu trúc.**



Hình 1.7 phản ánh các ca kiểm thử được xác định bởi hai phương pháp kiểm thử cấu trúc khác nhau. Giống như trước đây, phương pháp A xác định tập các ca kiểm thử lớn hơn so với phương pháp B. Có chắc là tập các ca kiểm thử lớn hơn là tốt hơn không? Đây là một câu hỏi thú vị và kiểm thử cấu trúc cung cấp các giải pháp để tìm câu trả lời cho vấn đề này.

Lưu ý rằng cả hai phương pháp A và B đều cho các tập các ca kiểm thử nằm trọn trong tập các hành vi được lập trình. Do các ca kiểm thử của các phương pháp này được sinh ra dựa trên chương trình nên rất khó để xác định các lỗi liên quan đến các hành vi đã được đặc tả nhưng không được lập trình. Tuy nhiên, dễ thấy rằng tập các ca kiểm thử cấu trúc là tương đối nhỏ so với tập tất cả các hành vi được lập trình.

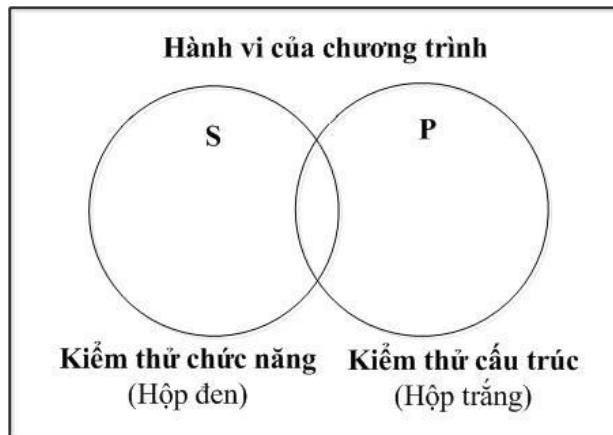
Chúng ta sẽ tìm hiểu các so sánh đánh giá về các ca kiểm thử được sinh bởi các phương pháp kiểm thử cấu trúc khác nhau ở mục 1.4.3. Một số phương pháp kiểm thử cấu trúc (kiểm thử dòng điều khiển, kiểm thử dòng dữ liệu và kiểm thử dựa trên lát cắt) sẽ được giới thiệu chi tiết trong các chương 6 và 7.

### 2.3 Tranh luận về kiểm thử chức năng so với kiểm thử cấu trúc

Cho trước hai cách tiếp cận khác nhau để xác định các ca kiểm thử, câu hỏi tự nhiên được đặt ra là phương pháp nào tốt hơn? Cho đến nay chúng ta vẫn chưa có câu trả lời thỏa đáng cho câu hỏi này.

Nói về kiểm thử cấu trúc, Robert Poston viết: công cụ này lãng phí thời gian của người kiểm thử vì từ những năm bảy mươi (của thế kỷ trước) nó chẳng trợ giúp tốt việc thực hành kiểm thử phần mềm và đừng có đưa nó vào bộ công cụ của người kiểm thử [Pos91]. Nhằm bảo vệ cho việc kiểm thử cấu trúc, Edward Miller [Mil91] viết: Độ bao phủ nhánh (một độ đo độ bao phủ của kiểm thử), nếu đạt được 85% hoặc cao hơn, có thể xác định số lỗi gấp đôi so với số lỗi phát hiện bởi kiểm thử trực quan (kiểm thử chức năng).

	<b>VIETTEL AI RACE</b>	TD058
	<b>TỔNG QUAN KIỂM THỬ</b>	Lần ban hành: 1



**Hình 1.8: Nguồn các ca kiểm thử.**

Biểu đồ Venn được mô tả trong hình 1.8 có thể giúp ta trả lời câu hỏi mà cuộc tranh luận này đã đề cập. Chúng ta cần khẳng định lại rằng mục đích của cả hai cách tiếp cận trên là để xác định các ca kiểm thử. Kiểm thử chức năng chỉ dùng đặc tả để xác định ca kiểm thử, trong khi kiểm thử cấu trúc dùng mã nguồn của chương trình (cài đặt) để làm cơ sở xác định các ca kiểm thử. Những bàn luận trước đây cho thấy chẳng có cách tiếp cận nào là đủ tốt.

Xét các hành vi chương trình: nếu tất cả các hành vi được đặc tả vẫn chưa được cài đặt, kiểm thử cấu trúc sẽ không thể nhận biết được điều đó. Ngược lại, nếu các hành vi được cài đặt chưa được đặc tả, điều đó chẳng khi nào có thể được phơi bày nhờ kiểm thử chức năng. Một con vi rút là một ví dụ tốt về các hành vi không được đặc tả. Câu trả lời sơ bộ cho câu hỏi trên là cả hai cách tiếp cận đều là rất cần thiết; còn câu trả lời cẩn thận hơn là cách kết hợp khôn khéo giữa hai cách tiếp cận này sẽ cung cấp niềm tin cho kiểm thử chức năng và độ đo của kiểm thử cấu trúc. Ta đã khẳng định ở trên rằng kiểm thử chức năng có khiếm khuyết về tính duy thừa và hổ phách. Nếu kiểm thử chức năng được tiến hành kết hợp với các số đo về độ phủ của kiểm thử cấu trúc thì khiếm khuyết trên có thể được phát hiện và giải quyết.

Quan điểm biểu đồ Venn cho việc kiểm thử đặt ra câu hỏi về quan hệ giữa tập các ca kiểm thử ( $T$ ) với các tập  $S$  và  $P$  của các hành vi cài đặt và đặc tả như thế nào? Rõ ràng, các ca kiểm thử trong  $T$  được xác định bởi phương pháp xác định ca kiểm thử được dùng. Một câu hỏi rất hay cần đặt ra là thế thì phương pháp này thích hợp và hiệu quả ra sao. Ta có thể đóng lại vòng luẩn quẩn này bằng những lời bàn trước đây. Với đường đi từ lỗi đến sai, thất bại và sự cố, nếu biết loại lỗi nào ta hay phạm, và loại sai nào hay có trong phần mềm được kiểm thử, ta có thể dùng thông tin này để lựa chọn phương pháp thích hợp để xác định các ca kiểm thử. Chính điểm này làm cho việc kiểm thử thành một nghệ thuật.

### 3. Phân loại các lỗi và sai

Các định nghĩa về lỗi và sai được trình bày trong mục 1.1 xoay quanh sự phân biệt giữa quy trình và sản phẩm. Trong khi quy trình cho chúng ta biết cần làm điều gì đó như thế nào thì sản phẩm là kết quả cuối cùng của quy trình. Kiểm thử phần mềm và đảm bảo chất lượng phần mềm (Software Quality Assurance - SQA) gặp nhau ở điểm là SQA có gắng cải tiến chất lượng sản phẩm bằng việc cải tiến quy trình. Theo nghĩa này thì kiểm thử là các hoạt động định hướng sản phẩm. SQA quan tâm nhiều hơn đến việc giảm thiểu lỗi trong quá trình phát triển, còn kiểm thử

	<b>VIETTEL AI RACE</b>	TD058
	<b>TỔNG QUAN KIỂM THỦ</b>	Lần ban hành: 1

quan tâm chủ yếu đến phát hiện sai trong sản phẩm. Cả hai nguyên lý này đều sử dụng định nghĩa về các loại sai. Các sai được phân loại theo nhiều cách khác nhau: giai đoạn phát triển khi cái sai tương ứng xuất hiện, các hậu quả của các thất bại tương ứng, độ khó cho việc giải quyết, độ rủi ro của việc không giải quyết được, v.v. Một cách phân loại được ưa thích là dựa trên việc xuất hiện bất thường: chỉ một lần, thỉnh thoảng, xuất hiện lại hoặc lặp đi lặp lại nhiều lần. Hình 1.9 minh họa một cách phân loại sai [Bor84] dựa trên độ nghiêm trọng của hậu quả mà các lỗi gây ra.