Nội dung định dạng font Times New Roman, size 13, spacing /before= 6pt, spacing /line space 1.2

**PHÂN CÔNG NGÀY 24/1/2018**

|  |  |
| --- | --- |
| **Mục nội dung** | **Thành viên** |
| 1.1 Why is testing necessary? |  |
| 1.2 What is testing? |  |
| 1.3 Testing principles |  |
| 1.4 Fundamental test process |  |
| 1.5 The psychology of testing |  |
| 2.1 Software development models |  |
| 2.2 Test levels |  |
| 2.3 Test types: the targets of testing |  |
| 2.4 Maintenance testing |  |
| 3.1 Reviews and the test process |  |
| 3.2 Review process |  |
| 3.3 Static analysisby tools |  |
| 4.1 Identifying test conditions and designing test cases |  |
| 4.2 Categories of test design techniques |  |
| 4.3 Specification-based or black-box techniques | Lê Quốc Huy  Phùng Quốc Phòng |
| 4.4 Structure-based or white-box techniques |  |
| 4.5 Experience-based techniques |  |
| 4.6 Choosing a test technique |  |
| 5.1 Test organization |  |
| 5.2 Test plans, estimates, and strategies |  |
| 5.3 Test progress monitoring and control |  |
| 5.4 Configuration management |  |
| 5.5 Risk and testing |  |
| 5.6 Incident management |  |
| 6.1 Types of test tool |  |
| 6.2 Effective use of tools: Potential benefits and risks |  |
| 6.3 Introducing a tool into an organization |  |
|  |  |

**FOUNDATIONS OF SOFTWARE TESTING**

**ISTQB CERTIFICATION**

# CHAPTER 1. Fundamentals of testing

## Why is testing necessary?

## What is testing?

## Testing principles

## Fundamental test process

## The psychology of testing

# CHAPTER 2. Testing throughout the software life cycle

## Software development models

## Test levels

## Test types: the targets of testing

## Maintenance testing

# CHAPTER 3. Static techniques

## Reviews and the test process

## Review process

## Static analysis by tools

# CHAPTER 4. Test design techniques

## **Identifying test conditions and designing test cases**

## **Categories of test design techniques**

## **Specification-based or black-box techniques**

## **Structure-based or white-box techniques**

## **Experience-based techniques**

## **Choosing a test technique**

# CHAPTER 5. Test management

## Test organization

## Test plans, estimates, and strategies

## Test progress monitoring and control

## Configuration management

## Risk and testing

## Incident management

# CHAPTER 6. Tool support for testing

## Types of test tool

## Effective use of tools: Potential benefits and risks

## Introducing a tool into an organization

## **4.3 Specification-based or black-box techniques**

**1.Viết các trường hợp thử nghiệm từ các mô hình phần mềm cho trước bằng các kỹ thuật thiết kế kiểm thử sau.**

**2. Hiểu được mục đích chính của mỗi trong bốn kỹ thuật, mức độ nào và loại thử nghiệm có thể sử dụng kỹ thuật này, và cách đo lường có thể đo được như thế nào**

**3.Hiểu được khái niệm về thử nghiệm trường hợp sử dụng và lợi ích của nó.**

Trong phần này, chúng tôi sẽ xem xét chi tiết tại bốn kỹ thuật dựa trên đặc điểm kỹ thuật hoặc blackbox. Bốn kỹ thuật này là K3 trong Chương trình - điều này có nghĩa là bạn cần phải có khả năng sử dụng những kỹ thuật này để thiết kế các trường hợp thử nghiệm. Chúng tôi cũng sẽ trình bày ngắn gọn (không phải ở mức K3) kỹ thuật kiểm thử trường hợp sử dụng đặc điểm kỹ thuật. Trong Phần 4.4, chúng ta sẽ xem xét các kỹ thuật dựa trên cơ sở K3.

Trong phần này, hãy tìm các định nghĩa của thuật ngữ thuật ngữ: phân tích giá trị biên, kiểm thử bảng quyết định, phân vùng tương đương, kiểm thử trạng thái chuyển đổi và kiểm thử trường hợp sử dụng.

Bốn đặc điểm kỹ thuật dựa trên chúng tôi sẽ bao gồm chi tiết là:

* phân vùng tương đương
* Phân tích giá trị biên
* Các bảng quyết định
* Kiểm thử chuyển đổi trạng thái

Lưu ý rằng chúng tôi sẽ thảo luận về hai người đầu tiên cùng nhau, bởi vì chúng có quan hệ gần gũi.

**4.3.1 Phân chia tương đương và phân tích giá trị biên**

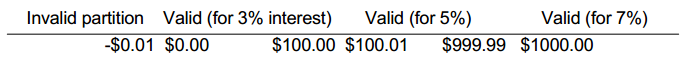
**Phân vùng tương đương**

Phân vùng tương đương (EP) là một kỹ thuật black-box toàn diện dựa trên đặc điểm kỹ thuật. Nó có thể được áp dụng ở bất kỳ mức độ kiểm thử và thường là một kỹ thuật tốt để sử dụng đầu tiên. Đó là một phương pháp thông thường để thử nghiệm .nhiều tester vẫn sử dụng phương pháp này trong công việc nhưng không biết được mình đang dùng phương pháp này.. Tuy nhiên, mặc dù tốt hơn là sử dụng kỹ thuật này một cách không chính thức hơn là không tốt chút nào, tốt hơn hết là sử dụng kỹ thuật một cách chính thức để đạt được những lợi ích đầy đủ mà nó có thể mang lại. Kỹ thuật này sẽ được tìm thấy trong hầu hết các cuốn sách kiểm thử, bao gồm [Myers, 1979] và [Copeland, 2003].

Ý tưởng đằng sau kỹ thuật này là phân chia (tức là phân vùng) một tập hợp các điều kiện kiểm thử thành các nhóm hoặc các bộ có thể được coi là giống hệt nhau (tức là hệ thống phải xử lý chúng tương đương), do đó 'phân vùng tương đương'. Phân vùng tương đương còn được gọi là các lớp tương đương - hai thuật ngữ có nghĩa chính xác cùng một điều.

Kỹ thuật phân vùng tương đương đòi hỏi rằng chúng ta chỉ cần thử nghiệm một điều kiện từ mỗi phân vùng. Điều này là bởi vì chúng tôi giả định rằng tất cả các điều kiện trong một phân vùng sẽ được xử lý theo cùng một cách bởi phần mềm.Nếu một điều kiện trong một phân vùng hoạt động, chúng tôi giả định rằng tất cả các điều kiện trong phân vùng đó sẽ hoạt động, và do đó có rất ít điểm trong việc kiểm thử bất kỳ một trong số những điều này. Ngược lại, nếu một trong các điều kiện trong một phân vùng không hoạt động, thì chúng ta giả sử rằng không có điều kiện trong phân vùng đó sẽ làm việc như vậy một lần nữa có rất ít điểm trong thử nghiệm nữa trong phân vùng đó. Tất nhiên những giả định đơn giản này có thể không phải lúc nào cũng đúng, nhưng nếu chúng tôi viết chúng xuống, ít nhất nó sẽ cho phép người khác có cơ hội thách thức các giả định mà chúng tôi đã thực hiện và hy vọng sẽ giúp xác định các phân vùng tốt hơn. Nếu bạn có thời gian, bạn có thể muốn thử nhiều hơn một giá trị từ một phân vùng, đặc biệt là nếu bạn muốn xác nhận lựa chọn của người sử dụng đầu vào điển hình.

Ví dụ: tài khoản tiết kiệm tại ngân hàng có mức lãi suất khác tùy thuộc vào số dư trong tài khoản. Để kiểm thử phần mềm tính toán lãi suất đến hạn, chúng tôi có thể xác định phạm vi của các giá trị cân bằng có mức lãi suất khác nhau. Ví dụ, nếu số dư từ 0 đến 100 đô la có lãi suất 3%, số dư trên 100 đô la và đến 1000 đô la có lãi suất 5% và số dư 1000 đô la trở lên có lãi suất 7%, chúng tôi sẽ ban đầu xác định ba phân vùng hợp lệ hợp lệ và một phân vùng không hợp lệ như được hiển thị bên dưới.



Lưu ý rằng chúng ta đã xác định được bốn phân vùng ở đây, mặc dù đặc tả chỉ đề cập đến ba. Điều này minh hoạ cho một nhiệm vụ rất quan trọng của người kiểm thử - chúng tôi không chỉ kiểm thử những gì trong đặc tả của chúng tôi mà còn nghĩ về những điều chưa được xác định. Trong trường hợp này, chúng tôi đã nghĩ đến tình huống mà số dư dưới 0. Chúng tôi vẫn chưa xác định được phân vùng không hợp lệ ở bên phải, nhưng đây cũng là một điều tốt để bạn cân nhắc. Để xác định nơi mà phân vùng 7% kết thúc, chúng ta cần phải biết số dư tối đa dành cho tài khoản này là gì (có thể không dễ dàng tìm ra). Trong ví dụ của chúng tôi, chúng tôi đã để lại mở này cho thời gian được. Lưu ý rằng đầu vào không phải số cũng là một phân vùng không hợp lệ (ví dụ như chữ 'a') nhưng chúng ta chỉ thảo luận về phân vùng số.

Chúng tôi đã giả định ở đây về sự khác biệt nhỏ nhất giữa hai giá trị. Chúng ta giả sử hai chữ số thập phân, tức là $ 100.00, nhưng chúng ta có thể giả định số không thập phân (tức là $ 100) hoặc nhiều hơn hai chữ số thập phân (ví dụ: $ 100.0000) Trong mọi trường hợp bạn nên nêu rõ các giả định của mình sau đó người khác có thể nhìn thấy chúng và cho bạn biết nếu họ là chính xác hay không.

Khi thiết kế các trường hợp thử nghiệm cho phần mềm này chúng tôi sẽ đảm bảo rằng ba phân vùng hợp lệ đều được bảo vệ một lần và chúng tôi cũng sẽ kiểm thử phân vùng không hợp lệ ít nhất một lần. Ví dụ: chúng tôi có thể chọn để tính toán mức lãi suất trên các số dư - $ 10.00, $ 50.00, $ 260.00 và $ 1348.00. Nếu chúng tôi không xác định cụ thể những phân vùng này, có thể ít nhất một trong số chúng có thể đã bị bỏ lỡ với chi phí thử nghiệm một số khác nhiều lần. Lưu ý rằng chúng ta cũng có thể áp dụng phân vùng tương đương để kết quả đầu ra là tốt. Trong trường hợp này, chúng tôi có ba mức lãi suất: 3%, 5% và 7%, cộng với thông báo lỗi cho phân vùng không hợp lệ (hoặc phân vùng). Trong ví dụ này, phân vùng đầu ra xếp thẳng hàng với các phân vùng đầu vào.

Làm thế nào sẽ ai đó thử nghiệm này mà không cần suy nghĩ về các phân vùng? Một người thử nghiệm ngây thơ (chúng ta hãy gọi ông Robbie) có thể nghĩ rằng một bộ các bài kiểm thử sẽ là để kiểm thử mỗi $ 50. Điều đó sẽ đưa ra các xét nghiệm sau: $ 50,00, $ 100,00, $ 150,00, $ 200,00, $ 250,00, ... nói lên đến $ 800,00 (sau đó Robbie đã có thể mệt mỏi và nghĩ rằng đã có đủ các bài kiểm thử đã được thực hiện). Nhưng hãy nhìn vào những gì Robbie đã thử nghiệm: chỉ có hai trong số bốn phân vùng! Vì vậy, nếu hệ thống không xử lý đúng cân bằng âm hoặc cân bằng 1000 đô la trở lên thì ông ta sẽ không tìm thấy những khuyết tật này - vì vậy cách tiếp cận ngây thơ ít hiệu quả hơn so với phân vùng tương đương. Đồng thời, Robbie đã có bốn lần kiểm thử (16 bài kiểm thử so với bốn thử nghiệm của chúng tôi sử dụng các phân vùng tương đương), do đó, ông cũng ít hiệu quả hơn! Đây là lý do tại sao chúng ta nói rằng sử dụng các kỹ thuật như vậy làm cho việc thử nghiệm hiệu quả hơn và hiệu quả hơn.

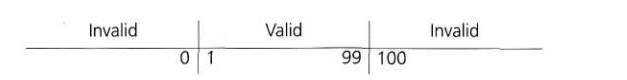
Lưu ý rằng khi nói phân vùng là 'không hợp lệ', nó không có nghĩa là nó đại diện cho một giá trị mà người dùng không thể nhập vào hoặc giá trị mà người dùng không phải nhập. Nó chỉ có nghĩa là nó không phải là một trong những đầu vào dự kiến ​​cho lĩnh vực cụ thể này. Phần mềm phải xử lý chính xác các giá trị từ phân vùng không hợp lệ, bằng cách trả lời thông báo lỗi như 'Số dư ít nhất phải là $ 0,00'.

Cũng lưu ý rằng phân vùng không hợp lệ có thể không hợp lệ chỉ trong ngữ cảnh tín dụng thanh toán lãi suất. Một tài khoản bị đánh cắp sẽ yêu cầu một số hành động khác.

**Phân tích giá trị ranh giới**

Phân tích giá trị ranh giới (BVA) dựa trên thử nghiệm tại ranh giới giữa các phân vùng. Nếu bạn đã từng thực hiện 'kiểm thử phạm vi', có lẽ bạn đã sử dụng kỹ thuật phân tích giá trị biên, ngay cả khi bạn không biết về nó. Lưu ý rằng chúng ta có cả ranh giới hợp lệ (trong các phân vùng hợp lệ) và ranh giới không hợp lệ (trong các phân vùng không hợp lệ).

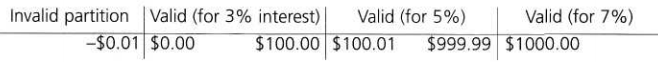
Ví dụ: hãy xem xét máy in có tùy chọn nhập vào số lượng



Để áp dụng phân tích giá trị ranh giới, chúng ta sẽ lấy các giá trị tối thiểu và tối đa (ranh giới) từ phân vùng hợp lệ (1 và 99 trong trường hợp này) cùng với bản sao được thực hiện, từ 1 đến 99.

giá trị đầu tiên hoặc giá trị cuối cùng tương ứng trong mỗi phân vùng không hợp lệ nằm cạnh phân vùng hợp lệ (0 và 100 trong trường hợp này). Trong ví dụ này, chúng ta sẽ có ba phép thử phân vùng tương đương (một trong ba phân vùng) và bốn bài kiểm thử giá trị biên.

Xem xét hệ thống ngân hàng được mô tả trong phần về phân vùng tương đương.



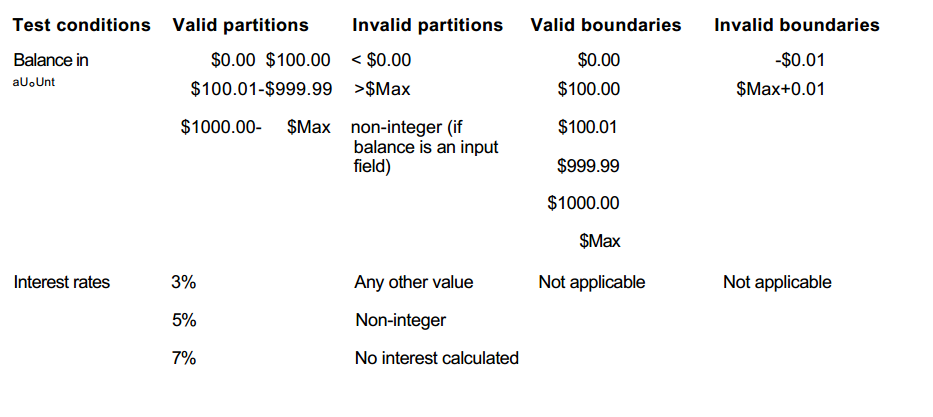
Bởi vì các giá trị ranh giới được định nghĩa là các giá trị trên cạnh của một phân vùng, chúng ta đã xác định các giá trị ranh giới sau: - $ 0,01 (một giá trị ranh giới không hợp lệ bởi vì nó ở cạnh của một phân vùng không hợp lệ), $ 0,00, $ 100,00, $ 100,01, $ 999,99 và $ 1000.00, tất cả giá trị ranh giới hợp lệ.

Vì vậy, bằng cách áp dụng phân tích giá trị ranh giới chúng ta sẽ có sáu bài kiểm thử cho các giá trị ranh giới. So sánh những gì Robbie đã làm: anh ta đã thực sự đánh một trong những giá trị ranh giới ($ 100) mặc dù nó đã được nhiều hơn do tai nạn hơn thiết kế.Vì vậy, ngoài việc thử nghiệm chỉ một nửa phân vùng, Robbie đã chỉ thử nghiệm một trong những ranh giới (vì vậy ông sẽ ít có hiệu quả hơn trong việc tìm kiếm bất kỳ sai sót ranh giới nào). Nếu chúng ta xem xét tất cả các kiểm thử của chúng tôi cho cả phân vùng tương đương và phân tích giá trị ranh giới, các kỹ thuật cho chúng ta tổng cộng chín bài kiểm thử, so với 16 Robbie đã có, vì vậy chúng tôi vẫn hiệu quả hơn đáng kể cũng như hiệu quả hơn ba lần (kiểm thử bốn phân vùng và sáu ranh giới, do đó, 10 điều kiện trong tổng số so với ba).

Lưu ý rằng trong ví dụ về lợi ích ngân hàng, chúng tôi có phân vùng hợp lệ bên cạnh các phân vùng hợp lệ khác. Nếu chúng ta xem xét một ranh giới không hợp lệ với lãi suất 3%, chúng ta có - $ 0,01, nhưng giá trị ở trên $ 100,00 thì sao? Giá trị $ 100.01 không phải là một ranh giới không hợp lệ; nó thực sự là một ranh giới hợp lệ bởi vì nó rơi vào một phân vùng hợp lệ. Ví dụ, phân vùng 5%, không có giá trị ranh giới không hợp lệ liên kết với các phân vùng bên cạnh nó.

Một cách tốt để đại diện cho các phân vùng và ranh giới hợp lệ và không hợp lệ nằm trong một bảng như Bảng 4.1:

Bảng 4.1 Phân vùng và ranh giới tương đương



Bằng cách hiển thị các giá trị trong bảng, chúng ta có thể thấy rằng không có quy định tối đa cho lãi suất 7%. Bây giờ chúng ta muốn biết giá trị tối đa là gì cho số dư tài khoản, để chúng ta có thể kiểm thử ranh giới đó. Đây được gọi là 'ranh giới mở', bởi vì một trong hai mặt của phân vùng vẫn còn mở, nghĩa là không được định nghĩa. Nhưng điều đó không có nghĩa là chúng ta có thể bỏ qua nó - chúng ta vẫn nên thử kiểm thử nó, nhưng làm thế nào?

Mở ranh giới khó kiểm thử hơn nhưng có nhiều cách để tiếp cận chúng. Trên thực tế giải pháp tốt nhất cho vấn đề là tìm ra ranh giới nên được chỉ định như thế nào! Một cách tiếp cận là quay trở lại đặc điểm kỹ thuật để xem nếu một tối đa đã được tuyên bố ở một nơi khác cho một số dư. Nếu vậy, thì chúng ta biết giá trị ranh giới của chúng ta là gì. Cách tiếp cận khác có thể là để điều tra các lĩnh vực liên quan khác của hệ thống. Ví dụ: trường có số dư tài khoản có thể chỉ là sáu con số cộng với hai con số thập phân.Điều này sẽ tạo ra một số dư tài khoản tối đa là $ 999 999,99 để chúng tôi có thể sử dụng nó làm giá trị ranh giới tối đa của chúng tôi. Nếu chúng ta thực sự không thể tìm thấy bất cứ điều gì về ranh giới này, thì có lẽ chúng ta cần phải sử dụng cách tiếp cận dựa trên kinh nghiệm hoặc trực quan để thăm dò các giá trị lớn khác nhau cố gắng làm cho nó thất bại.

Chúng tôi cũng có thể cố gắng tìm hiểu về ranh giới mở thấp hơn - sự cân bằng âm tính thấp nhất là gì? Mặc dù chúng tôi đã bỏ qua điều này trong ví dụ của chúng tôi, nhưng hãy nêu ra trong bảng cho thấy chúng tôi đã bỏ qua nó, vì vậy hãy giúp chúng tôi triệt để hơn nếu chúng tôi muốn.

Việc biểu diễn các phân vùng và ranh giới trong một bảng như thế này cũng giúp bạn dễ dàng kiểm thử xem bạn đã từng thử nghiệm một (nếu đó là mục tiêu của bạn) hay không.

Mở rộng phân vùng tương đương và phân tích giá trị ranh giới Cho đến nay, bằng cách sử dụng EP và BVA chúng tôi đã xác định các điều kiện có thể được kiểm thử, tức là phân vùng và giá trị ranh giới. Các kỹ thuật này được sử dụng để xác định các điều kiện kiểm thử, có thể ở mức khá cao (ví dụ như 'tài khoản lãi suất thấp') hoặc ở mức chi tiết (ví dụ: 'giá trị là 100,00 đô la'). Chúng tôi đã xem xét việc áp dụng các kỹ thuật này vào các phạm vi của con số. Tuy nhiên, chúng ta cũng có thể áp dụng các kỹ thuật này cho những thứ khác.

Ví dụ: nếu bạn đặt chuyến bay, bạn có thể chọn Economy / Coach, Premium Economy, Business hoặc First Class. Mỗi một trong số đó là một phân vùng tương đương với quyền riêng của nó và cần được kiểm thử, nhưng nó không có ý nghĩa để nói về các ranh giới cho loại phân vùng này, mà là một tập hợp các điều hợp lệ. Phân vùng không hợp lệ sẽ là một nỗ lực để gõ vào bất kỳ loại khác của lớp học bay (ví dụ như 'Đặc điểm nhân viên (8) Nếu trường này được thực hiện bằng cách sử dụng danh sách thả xuống, thì không nên gõ bất cứ điều gì khác, nhưng vẫn là một thử nghiệm tốt để thử ít nhất một lần trong một trường thả xuống. Khi bạn phân tích cơ sở kiểm thử (ví dụ như một yêu cầu đặc điểm kỹ thuật), phân vùng tương đương có thể giúp xác định nơi một danh sách thả xuống sẽ thích hợp.

Khi cố gắng xác định lỗi, bạn có thể thử một số giá trị trong một phân vùng. Nếu kết quả này trong hành vi khác nhau mà bạn mong đợi nó sẽ được như nhau, sau đó có thể có hai (hoặc nhiều hơn) phân vùng mà ban đầu bạn nghĩ rằng chỉ có một.

Chúng ta có thể áp dụng phân vùng tương đương và phân tích giá trị ranh giới cho tất cả các cấp độ thử nghiệm. Các ví dụ ở đây ở mức độ khá chi tiết có thể là thích hợp nhất trong kiểm thử thành phần hoặc trong việc kiểm thử chi tiết của một màn hình đơn.

Ở cấp độ hệ thống, chúng ta có thể có ba cấu hình cơ bản mà người dùng của chúng tôi có thể lựa chọn khi thiết lập hệ thống, với một số tùy chọn cho mỗi cấu hình. Các cấu hình cơ bản có thể là quản trị viên hệ thống, quản lý và liên lạc khách hàng. Chúng thể hiện ba phân vùng tương đương có thể được kiểm thử. Chúng ta có thể gặp những vấn đề nghiêm trọng nếu chúng ta quên kiểm thử cấu hình cho quản trị hệ thống.

Chúng ta cũng có thể áp dụng phân vùng tương đương và phân tích giá trị biên hơn một lần cho cùng một mục đặc tả. Ví dụ: nếu một hệ thống điện thoại nội bộ cho một công ty có 200 điện thoại có số mở rộng 3 chữ số từ 100 đến 699, chúng ta có thể xác định các phân vùng và ranh giới sau:

• chữ số (ký tự 0 đến 9) với phân vùng không hợp lệ có chứa các chữ số không

• Số chữ số, 3 (giá trị ranh giới không hợp lệ của 2 số và 4 chữ số)

• dãy số mở rộng, từ 100 đến 699 (giá trị ranh giới không hợp lệ là 099 và 700)

• Các mở rộng đang được sử dụng và những phần không được (hai phân vùng hợp lệ, không có ranh giới)

• Các số mở rộng thấp nhất và cao nhất được sử dụng cũng có thể được sử dụng làm giá trị ranh giới

Một trường hợp thử nghiệm có thể kiểm thử nhiều hơn một phân vùng / ranh giới. Ví dụ, mở rộng 409 được sử dụng sẽ kiểm thử bốn phân vùng hợp lệ: chữ số, số chữ số, phạm vi hợp lệ, và 'sử dụng' phân vùng. Nó cũng kiểm thử các giá trị ranh giới cho các chữ số, 0 và 9.

Có bao nhiêu trường hợp thử nghiệm cần phải kiểm thử tất cả các phân vùng và ranh giới, cả hai hợp lệ và không hợp lệ? Chúng tôi cần một số không chữ số, một số gồm 2 chữ số và 4 chữ số, các giá trị 99, 100, 699 và 700, một phần mở rộng không sử dụng và có thể là các phần mở rộng thấp nhất và cao nhất trong sử dụng. Đây là mười hoặc mười một trường hợp thử nghiệm - số chính xác sẽ phụ thuộc vào những gì chúng ta có thể kết hợp trong một trường hợp thử nghiệm.

So sánh điều này với ví dụ số một con số trong Phần 1.1.6. Ở đây chúng tôi thấy rằng chúng tôi cần 68 bài kiểm thử chỉ để kiểm thử một trường một chữ số, nếu chúng tôi đã để kiểm thử nó kỹ lưỡng. Sử dụng phân vùng tương đương và phân tích giá trị ranh giới giúp chúng tôi xác định các bài kiểm thử có nhiều khả năng tìm thấy lỗi và sử dụng ít trường hợp kiểm thử hơn để tìm chúng. Điều này là bởi vì các nội dung của một phân vùng là đại diện của tất cả các giá trị có thể. Thay vì kiểm thử tất cả mười chữ số riêng biệt, chúng ta kiểm thử một ở giữa (ví dụ 4) và hai cạnh (0 và 9). Thay vì kiểm thử mọi nhân vật không có chữ số, có thể đại diện cho tất cả chúng. Vì vậy, chúng tôi có bốn bài kiểm thử (thay vì 68) cho một trường một chữ số.

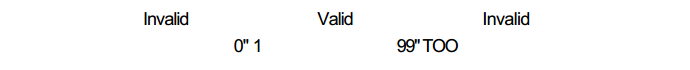
Như đã đề cập ở trên, chúng ta cũng có thể áp dụng các kỹ thuật này để phân vùng đầu ra. Xem xét việc mở rộng sau đây cho ví dụ về tỷ lệ lãi suất ngân hàng của chúng tôi. Giả sử khách hàng có nhiều tài khoản có thể có thêm 1% lợi tức trong tài khoản này nếu họ có ít nhất 1.000 đô la trong tài khoản đó. Bây giờ chúng tôi có hai giá trị đầu ra có thể (7% và 8% lãi) cho cùng số dư tài khoản, vì vậy chúng tôi đã xác định một điều kiện kiểm thử khác (8% lãi suất). (Chúng tôi cũng có thể xác định được điều kiện đầu ra tương tự bằng cách xem xét khách hàng có nhiều tài khoản, là phân vùng các loại khách hàng.)

Phân vùng tương đương cũng có thể áp dụng cho các loại đầu vào khác nhau. Các ví dụ của chúng tôi tập trung vào các đầu vào mà người dùng (con người) nhập vào khi sử dụng hệ thống. Tuy nhiên, hệ thống cũng nhận được dữ liệu đầu vào từ các nguồn khác, chẳng hạn như từ các hệ thống khác thông qua một số giao diện - đây cũng là nơi tốt để tìm các phân vùng (và ranh giới). Ví dụ, giá trị của một tham số giao diện có thể rơi vào các phân vùng tương đương hợp lệ và không hợp lệ. Loại khiếm khuyết này thường gặp khó khăn trong việc thử nghiệm khi các giao diện đã được kết hợp với nhau, do đó đặc biệt hữu ích khi áp dụng trong kiểm thử tích hợp (tích hợp thành phần hoặc tích hợp hệ thống).

Phân tích giá trị ranh giới có thể được áp dụng cho toàn bộ chuỗi ký tự (ví dụ tên hoặc địa chỉ). Số ký tự trong chuỗi là một phân vùng, ví dụ: từ 1 đến 30 ký tự là phân vùng hợp lệ với các ranh giới hợp lệ là 1 và 30. Các ranh giới không hợp lệ sẽ là 0 ký tự (null, chỉ cần nhấn phím Quay lại) và 31 ký tự. Cả hai sẽ tạo ra một thông báo lỗi.

Phân vùng cũng có thể được xác định khi thiết lập dữ liệu kiểm thử. Nếu có nhiều loại bản ghi khác nhau, thử nghiệm của bạn sẽ đại diện hơn nếu bạn bao gồm một bản ghi dữ liệu của từng loại. Kích thước của một bản ghi cũng là một phân vùng có ranh giới, vì vậy chúng ta có thể bao gồm các bản ghi kích thước tối đa và tối thiểu trong cơ sở dữ liệu thử nghiệm.

Nếu bạn có một số kiến ​​thức bên trong về cách dữ liệu được tổ chức thực tế, bạn có thể xác định được một số ranh giới ẩn. Ví dụ: nếu khối lưu trữ tràn được sử dụng khi nhập hơn 255 ký tự vào trường, các bài kiểm thử giá trị biên sẽ bao gồm 255 và 256 ký tự trong trường đó. Điều này có thể kết hợp với việc kiểm thử hộp trắng, vì chúng ta có một số kiến ​​thức về cách cấu trúc dữ liệu, tuy nhiên chúng ta không phân loại mọi thứ miễn là thử nghiệm của chúng tôi có hiệu quả trong việc tìm kiếm các khuyết tật. Không được treo lên trên một sự khác biệt tốt đẹp - chỉ cần làm bất cứ điều gì thử nghiệm có ý nghĩa, dựa trên những gì bạn biết. Một câu tục ngữ cổ của Trung Quốc nói, "Không phải vấn đề là con mèo trắng hay đen; tất cả những vấn đề là con mèo bắt chuột ".

Với phân tích giá trị ranh giới, chúng tôi nghĩ ranh giới là một đường phân chia giữa hai điều. Do đó chúng ta có một giá trị ở mỗi bên của ranh giới (nhưng ranh giới chính nó không phải là một giá trị). 

Nhìn vào các giá trị cho ví dụ máy in của chúng ta, 0 trong một phân vùng không hợp lệ, 1 và 99 nằm trong phân vùng hợp lệ và 100 là trong phân vùng không hợp lệ khác. Vì vậy, ranh giới giữa các giá trị của 0 và 1, và giữa các giá trị 99 và 100. Có một trường phái tư tưởng coi trọng một giá trị thực như là một giá trị ranh giới. Theo truyền thống, đây là các giá trị trong phân vùng hợp lệ (tức là các giá trị được chỉ định). Cách tiếp cận này đòi hỏi ba giá trị cho mỗi ranh giới hai, vì vậy bạn sẽ có 0,1 và 2 cho ranh giới bên trái, và 98, 99 và 100 cho ranh giới bên phải trong ví dụ này. Các giá trị ranh giới được gọi là 'trên và bên cạnh ranh giới' và giá trị 'trên' ranh giới thường được thực hiện trong phân vùng hợp lệ.

Lưu ý rằng Beizer nói về kiểm thử tên miền, sự tổng quát của phân vùng tương đương, với ba giá trị ranh giới. Ông tạo sự khác biệt giữa các ranh giới mở và đóng cửa, nơi mà một ranh giới kín là nơi mà điểm này được bao gồm trong miền. Vì vậy, quy ước là cho các phân vùng hợp lệ để có ranh giới đóng cửa. Bạn có thể vui lòng biết rằng bạn không phải biết điều này cho kỳ thi! Tiêu chuẩn Anh 7925-2 Tiêu chuẩn cho Kiểm thử Hợp phần Phần mềm cũng định nghĩa cách tiếp cận ba giá trị để phân tích giá trị biên.

Vậy cách tiếp cận nào là tốt nhất? Nếu bạn sử dụng cách tiếp cận hai giá trị cùng với phân vùng tương đương, bạn có hiệu quả tương đương và hiệu quả hơn một chút so với cách tiếp cận 3 giá trị. (Chúng tôi sẽ không đi vào chi tiết ở đây nhưng điều này có thể được chứng minh.)Trong cuốn sách này, chúng tôi sẽ sử dụng cách tiếp cận hai giá trị. Trong kỳ thi, bạn có thể có câu hỏi dựa trên phương pháp tiếp cận twovalue hoặc phương pháp ba giá trị, nhưng phải rõ ràng là sự lựa chọn đúng là trong cả hai trường hợp.

**Thiết kế các ca sử dụng**

Sau khi xác định các điều kiện mà bạn muốn kiểm thử, trong trường hợp này bằng phân chia tương đương và phân tích giá trị biên, bước tiếp theo là thiết kế các trường hợp thử nghiệm. Các điều kiện kiểm thử nhiều hơn có thể được bao gồm trong một trường hợp kiểm thử đơn lẻ, ít trường hợp thử nghiệm sẽ được cần thiết để trang trải tất cả các điều kiện. Đây thường là phương pháp tốt nhất để thực hiện các bài kiểm thử dương tính và để kiểm thử rằng bạn tin tưởng chắc chắn sẽ vượt qua được. Tuy nhiên nếu một bài kiểm thử thất bại thì chúng ta cần phải tìm ra lý do tại sao nó thất bại - điều kiện kiểm thử nào được xử lý không chính xác? Chúng ta cần phải có sự cân bằng giữa việc kiểm thử quá nhiều và quá ít các điều kiện kiểm thử trong các thử nghiệm của chúng tôi.

Hãy xem xét một trường hợp thử nghiệm có thể bao gồm một hoặc nhiều điều kiện kiểm thử. Ví dụ về số dư ngân hàng, thử nghiệm đầu tiên của chúng tôi có thể là một khách hàng mới với số dư 500 đô la Mỹ. Điều này sẽ bao gồm một sự cân bằng trong phân vùng từ $ 100,01 đến $ 999,99 và một phân vùng đầu ra của một mức lãi suất 5%. Chúng ta cũng sẽ bao gồm các phân vùng khác mà chúng ta vẫn chưa thảo luận. ví dụ như khách hàng hợp lệ, khách hàng mới, khách hàng chỉ có một tài khoản ... Tất cả các phân vùng được đề cập trong bài kiểm thử này đều là các phân vùng hợp lệ.

Khi chúng tôi đến để kiểm thử các phân vùng không hợp lệ, tùy chọn an toàn nhất có thể chỉ cố gắng để chỉ một điều kiện kiểm thử không hợp lệ cho mỗi trường hợp thử nghiệm. Điều này là bởi vì các chương trình có thể ngừng xử lý đầu vào ngay khi gặp vấn đề đầu tiên. Vì vậy, nếu bạn có tên khách hàng không hợp lệ, địa chỉ không hợp lệ và số dư không hợp lệ, bạn có thể nhận được thông báo lỗi "nhập không hợp lệ" và bạn không biết liệu thử nghiệm đã phát hiện chỉ một đầu vào không hợp lệ hoặc tất cả chúng. (Đây cũng là lý do tại sao các thông báo lỗi cụ thể tốt hơn nhiều so với những cái chung!)

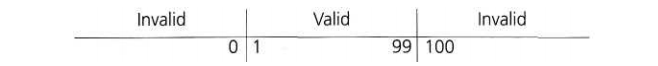
Tuy nhiên, nếu được biết rằng phần mềm đang được kiểm thử là cần thiết để xử lý tất cả các đầu vào không phụ thuộc vào tính hợp lệ, thì có thể tiếp tục như trước và thiết kế các trường hợp thử nghiệm bao gồm nhiều điều kiện không hợp lệ trong một lần. Ví dụ, nếu mỗi trường không hợp lệ trong một mẫu có một số văn bản màu đỏ ở trên hoặc dưới trường nói rằng trường này không hợp lệ và tại sao, sau đó bạn biết rằng mỗi trường đã được kiểm thử, vì vậy bạn đã kiểm thử tất cả các lỗi xử lý trong một bài kiểm thử trường hợp. Trong cả hai trường hợp, cần có các trường hợp thử nghiệm riêng bao gồm các điều kiện hợp lệ và không hợp lệ.

Để bao gồm các trường hợp thử nghiệm ranh giới, có thể kết hợp tất cả các ranh giới tối thiểu hợp lệ cho một nhóm các trường vào một trường hợp kiểm thử và cũng là giá trị giới hạn cực đại. Các ranh giới không hợp lệ có thể được kiểm thử cùng nhau nếu việc xác nhận được thực hiện trên mọi lĩnh vực; nếu không chúng nên được kiểm thử riêng rẽ, như với các phân vùng không hợp lệ.

**Tại sao cả hai phân vùng tương đương và phân tích giá trị ranh giới?**

Về mặt kỹ thuật, bởi vì mỗi ranh giới nằm trong một số phân vùng, nếu bạn chỉ phân tích giá trị ranh giới, bạn cũng đã thử nghiệm tất cả các phân vùng tương đương. Tuy nhiên, cách tiếp cận này có thể gây ra vấn đề nếu giá trị đó thất bại - chỉ có giá trị ranh giới thất bại hay toàn bộ phân vùng thất bại? Cũng bằng cách kiểm thử ranh giới chỉ có thể chúng tôi sẽ không cung cấp cho người dùng nhiều sự tự tin như chúng tôi đang sử dụng các giá trị cực đoan hơn là giá trị bình thường.Ranh giới có thể khó khăn hơn (và do đó tốn kém hơn) để thiết lập.

Ví dụ, trong ví dụ về bản sao của máy in được mô tả trước đó, chúng tôi đã xác định các giá trị ranh giới sau:



Giả sử chúng ta kiểm thử chỉ các giá trị ranh giới hợp lệ 1 và 99 và không có gì ở giữa. Nếu cả hai bài kiểm thử vượt qua, điều này dường như cho thấy rằng tất cả các giá trị ở giữa cũng nên làm việc. Tuy nhiên, giả sử rằng một trang được in đúng, nhưng 99 trang thì không. Bây giờ chúng ta không biết liệu có bất kỳ tập hợp nào của nhiều trang, vì vậy điều đầu tiên chúng ta làm là kiểm thử 10 trang, tức là một giá trị từ phân vùng tương đương.

Chúng tôi khuyên bạn nên kiểm thử các phân vùng riêng biệt với các ranh giới - điều này có nghĩa là chọn các giá trị phân vùng không phải là giá trị ranh giới.

Tuy nhiên, nếu bạn sử dụng cách tiếp cận giá trị ranh giới ba giá trị, bạn sẽ có các giá trị ranh giới hợp lệ là 1, 2, 98 và 99, vì vậy có một giá trị tương đương riêng biệt cộng với hai giá trị ranh giới sẽ không mang lại nhiều lợi ích bổ sung. Nhưng lưu ý rằng một giá trị tương đương, ví dụ như 10, sẽ thay thế cả hai giá trị ranh giới thêm hai (2 và 98). Đây là lý do tại sao phân chia tương đương với phân tích giá trị ranh giới hai giá trị hiệu quả hơn phân tích giá trị ranh giới ba giá trị.

Những phân vùng và ranh giới bạn quyết định tập thể dục (bạn không cần phải kiểm thử tất cả), và những phân vùng mà bạn quyết định thử nghiệm đầu tiên, phụ thuộc vào các mục tiêu thử nghiệm của bạn. Nếu mục tiêu của bạn là cách tiếp cận toàn diện nhất, sau đó làm theo các thủ tục kiểm thử các phân vùng hợp lệ đầu tiên, sau đó các phân vùng không hợp lệ, sau đó ranh giới hợp lệ và cuối cùng ranh giới không hợp lệ. Tuy nhiên, nếu bạn chịu áp lực về thời gian và không thể kiểm thử mọi thứ (và ai không?), Thì các mục tiêu kiểm thử của bạn sẽ giúp bạn quyết định thử nghiệm những gì. Nếu bạn là người dùng tin tưởng vào các giao dịch điển hình với một số lượng tối thiểu các bài kiểm thử, bạn chỉ có thể làm các phân vùng hợp lệ. Nếu bạn muốn tìm càng nhiều khiếm khuyết càng tốt càng tốt, bạn có thể bắt đầu với các giá trị ranh giới, cả hợp lệ và không hợp lệ. Nếu bạn muốn tự tin rằng hệ thống sẽ xử lý đầu vào xấu một cách chính xác, bạn có thể làm phân vùng và ranh giới không hợp lệ chủ yếu. Kinh nghiệm trước của bạn về các loại lỗi tìm thấy có thể giúp bạn tìm ra các lỗi tương tự; ví dụ nếu thường có một số lỗi ranh giới, thì bạn sẽ bắt đầu bằng cách kiểm thử ranh giới.

Phân chia cân bằng và phân tích giá trị ranh giới được mô tả trong hầu hết các cuốn sách thử nghiệm, bao gồm [Myers, 1979] và [Copeland, 2003]. Ví dụ về các loại lớp tương đương để tìm ra được đưa ra trong [Kaner và cộng sự., 1993] phân chia sự cân bằng và phân tích giá trị biên được mô tả trong BS79252, bao gồm các bài kiểm thử thiết kế và đo độ bao phủ.

**4.3.2 Thử nghiệm bảng quyết định**

**Tại sao sử dụng bảng quyết định ?.**

Các kỹ thuật phân chia tương đương và phân tích giá trị ranh giới thường được áp dụng cho các tình huống hoặc đầu vào cụ thể. Tuy nhiên, nếu các kết hợp đầu vào khác nhau dẫn đến các hành động khác nhau, có thể khó hiển thị hơn bằng phân chia tương đương và phân tích giá trị biên, thường tập trung hơn vào giao diện người dùng. Hai kỹ thuật dựa trên đặc tả khác, các bảng quyết định và kiểm thử chuyển đổi trạng thái tập trung nhiều hơn vào logic kinh doanh hoặc các quy tắc kinh doanh.

Một bảng quyết định là một cách tốt để đối phó với sự kết hợp của sự vật (ví dụ như đầu vào). Kỹ thuật này đôi khi còn được gọi là bảng 'nguyên nhân gây ra'. Lý do của việc này là có một kỹ thuật lập sơ đồ logic có liên quan được gọi là 'đồ thị nguyên nhân-kết quả' đôi khi được sử dụng để giúp lấy được bảng quyết định (Myers mô tả điều này như là một mạng logic tổ hợp [Myers, 1979]). Tuy nhiên, hầu hết mọi người thấy hữu ích hơn khi sử dụng bảng mô tả trong [Copeland, 2003].

Nếu bạn bắt đầu sử dụng các bảng quyết định để khám phá những quy tắc kinh doanh cần phải được kiểm thử, bạn có thể thấy rằng các nhà phân tích và nhà phát triển thấy các bảng rất hữu ích và cũng muốn bắt đầu sử dụng chúng. Hãy khuyến khích điều này, vì nó sẽ làm cho công việc của bạn dễ dàng hơn trong tương lai. Bảng quyết định cung cấp một cách có hệ thống các quy tắc kinh doanh phức tạp, rất hữu ích cho các nhà phát triển cũng như cho người kiểm thử. Các bảng quyết định có thể được sử dụng trong thiết kế thử nghiệm dù chúng có được sử dụng trong các chi tiết kỹ thuật hay không, vì chúng giúp người kiểm thử khám phá những ảnh hưởng của việc kết hợp các đầu vào khác nhau và các trạng thái phần mềm khác phải thực hiện chính xác các quy tắc kinh doanh. Giúp các nhà phát triển làm một công việc tốt hơn cũng có thể dẫn đến các mối quan hệ tốt hơn với họ.

Kết hợp thử nghiệm có thể là một thách thức, vì số lượng kết hợp thường rất lớn. Thử nghiệm tất cả các kết hợp có thể không thực tế nếu không phải là không thể. Chúng ta phải hài lòng với thử nghiệm chỉ là một tập con nhỏ của sự kết hợp nhưng làm cho sự lựa chọn của các kết hợp để kiểm thử và để bỏ ra không phải là tầm thường. Nếu bạn không có một cách chọn lọc có hệ thống, một tập hợp con tùy ý sẽ được sử dụng và điều này cũng có thể dẫn đến một nỗ lực kiểm thử hiệu quả.

Các bảng quyết định hỗ trợ việc lựa chọn có hệ thống các trường hợp thử nghiệm có hiệu quả và có thể có tác dụng phụ có lợi trong việc tìm kiếm các vấn đề và sự mơ hồ trong đặc tả. Đó là một kỹ thuật hoạt động tốt kết hợp với phân vùng tương đương. Sự kết hợp các điều kiện thăm dò có thể là sự kết hợp các phân vùng tương đương.

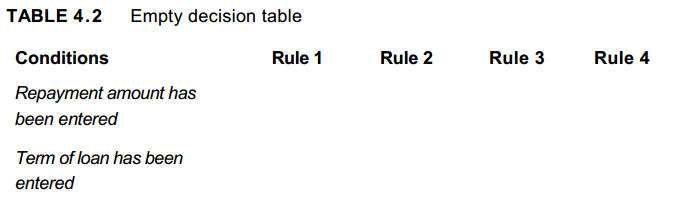
Ngoài các bảng quyết định, còn có các kỹ thuật khác để đối phó với các kết hợp thử nghiệm của sự vật: kiểm thử theo cặp và các mảng trực giao. Chúng được mô tả trong [Copeland, 2003]. Một kỹ thuật khác là [Pol et al., 2001]. Bảng quyết định và đồ thị tác động được mô tả trong [BS7925-2], bao gồm các bài kiểm thử thiết kế và đo độ bao phủ.

Sử dụng các bảng quyết định cho thiết kế kiểm thử

Nhiệm vụ đầu tiên là xác định một chức năng hay hệ thống con phù hợp có hành vi phản ứng dựa trên sự kết hợp của các đầu vào hoặc sự kiện. Hành vi quan tâm không được quá rộng (tức là không nên chứa quá nhiều đầu vào) nếu không thì số lượng kết hợp sẽ trở nên rườm rà và khó quản lý. Nó sẽ tốt hơn để đối phó với số lượng lớn các điều kiện bằng cách chia chúng thành các tập con và xử lý các tập con một lúc.

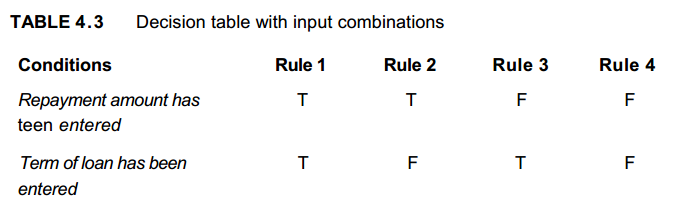
Một khi bạn đã xác định các khía cạnh cần phải được kết hợp, sau đó bạn đưa chúng vào một bảng liệt kê tất cả các kết hợp của Đúng và Sai cho mỗi khía cạnh. Lấy ví dụ về một đơn xin vay, nơi bạn có thể nhập số tiền trả hàng tháng hoặc số năm bạn muốn trả để trả lại (thời hạn của khoản vay). Nếu bạn nhập vào cả hai, hệ thống sẽ thỏa hiệp giữa hai nếu chúng xung đột. Hai điều kiện là số tiền vay và kỳ hạn, vì vậy chúng tôi đưa chúng vào bảng (xem Bảng 4.2).

Bảng 4.2 Bảng quyết định trống



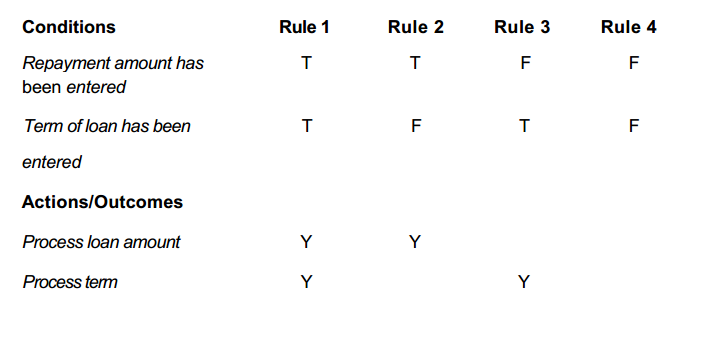
Tiếp theo chúng ta sẽ xác định tất cả các kết hợp của Đúng và Sai (xem Bảng 4.3). Với hai điều kiện, mỗi điều có thể là Đúng hoặc Sai, chúng tôi sẽ có bốn sự kết hợp (hai với sức mạnh của số thứ sẽ được kết hợp). Lưu ý rằng nếu chúng ta có ba điều để kết hợp, chúng ta sẽ có tám kết hợp, với bốn điều, có 16, vv Đó là lý do tại sao rất tốt để giải quyết các bộ kết hợp nhỏ tại một thời điểm. Để theo dõi kết hợp nào chúng ta có, chúng ta sẽ thay đổi True và False ở hàng dưới cùng, đặt hai Trues và hai Falses lên hàng trên hàng dưới cùng, vv, do đó hàng trên cùng sẽ có tất cả Trues và sau đó tất cả các Falses (và nguyên tắc này áp dụng cho tất cả các bảng như vậy).

BẢNG 4.3 Bảng quyết định với sự kết hợp đầu vào



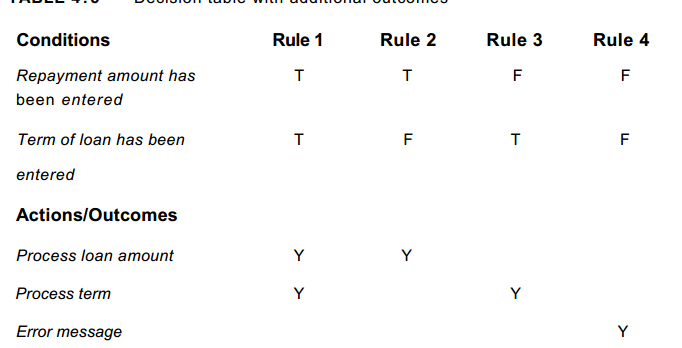
Bước tiếp theo (ít nhất cho ví dụ này) là xác định kết quả chính xác cho mỗi sự kết hợp (xem Bảng 4.4). Trong ví dụ này, chúng ta có thể nhập một hoặc cả hai trường. Mỗi sự kết hợp đôi khi được gọi là một quy tắc.

Bảng 4.4 Bảng quyết định kết hợp và kết quả



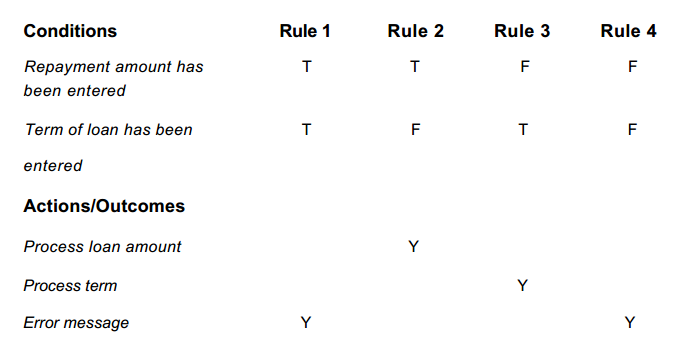
Tại thời điểm này, chúng ta có thể nhận ra rằng chúng tôi đã không nghĩ về những gì sẽ xảy ra nếu khách hàng không nhập bất cứ điều gì trong một trong hai lĩnh vực. Bảng đã làm nổi bật một sự kết hợp mà không được đề cập trong đặc tả cho ví dụ này. Chúng ta có thể giả định kết hợp này sẽ dẫn đến một thông báo lỗi, vì vậy chúng ta cần thêm một hành động nữa (xem Bảng 4.5). Điều này làm nổi bật sức mạnh của kỹ thuật này để phát hiện ra những thiếu sót và mơ hồ trong kỹ thuật. Nó không phải là bất thường đối với một số kết hợp được bỏ qua từ chi tiết kỹ thuật; do đó đây cũng là một kỹ thuật có giá trị để sử dụng khi xem xét cơ sở kiểm thử.

Bảng 4.5 Bảng quyết định với các kết quả bổ sung



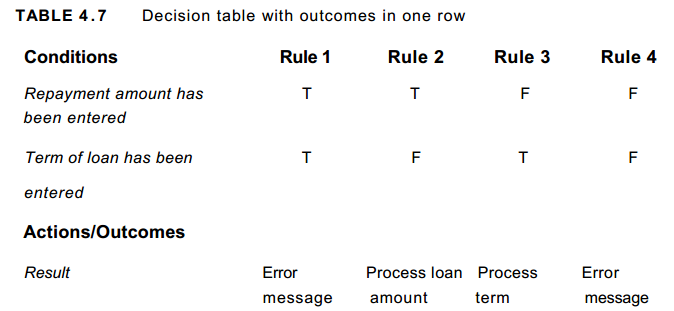
Giả sử chúng ta thay đổi ví dụ của chúng tôi một chút, do đó khách hàng không được phép nhập cả khoản thanh toán và kỳ hạn. Bây giờ bảng của chúng ta sẽ thay đổi, bởi vì cũng nên có một thông báo lỗi nếu cả hai được nhập, vì vậy nó sẽ giống như Bảng 4.6.

BẢNG 4.6 Bảng quyết định với các kết quả đã thay đổi



Bạn có thể nhận thấy rằng chỉ có một 'Có' trong mỗi cột, nghĩa là các hành động của chúng tôi là loại trừ lẫn nhau - chỉ có một hành động xảy ra cho mỗi sự kết hợp các điều kiện. Chúng ta có thể biểu diễn điều này bằng một cách khác bằng cách liệt kê các hành động trong ô của một hàng như thể hiện trong Bảng 4.7. Lưu ý rằng nếu có nhiều kết quả từ bất kỳ kết hợp nào, tốt hơn nên hiển thị chúng dưới dạng các hàng riêng biệt thay vì kết hợp chúng thành một hàng.

BẢNG 4.7 Bảng quyết định với kết quả trong một hàng



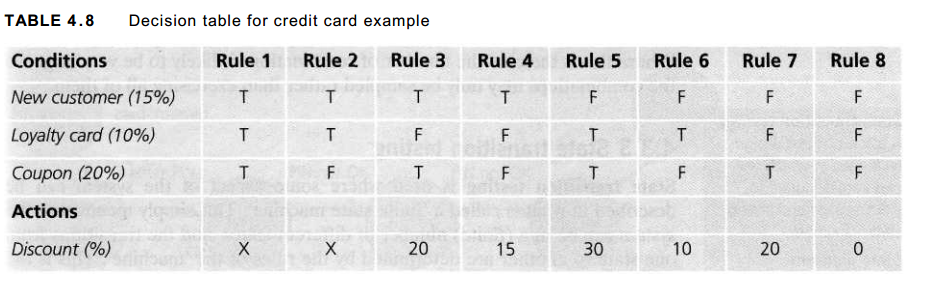
Bước cuối cùng của kỹ thuật này là viết các trường hợp thử nghiệm để thực hiện từng bốn quy tắc trong bảng của chúng tôi.

Trong ví dụ này, chúng tôi bắt đầu bằng cách xác định các điều kiện đầu vào và sau đó xác định các kết quả. Tuy nhiên, trong thực tế nó có thể làm việc theo cách khác xung quanh - chúng ta có thể thấy rằng có một số kết quả khác nhau, và phải làm việc trở lại để hiểu những gì kết hợp của điều kiện đầu vào thực sự lái xe những kết quả. Kỹ thuật này hoạt động cũng như làm theo cách này, và có thể là một cách tiếp cận lặp đi lặp lại khi bạn khám phá thêm về các quy tắc điều khiển hệ thống.

Ví dụ về thẻ tín dụng

Hãy nhìn vào một ví dụ khác. Nếu bạn là khách hàng mới mở một tài khoản thẻ tín dụng, bạn sẽ nhận được 15% giảm giá cho tất cả các giao dịch mua của bạn hôm nay. Nếu bạn là khách hàng hiện tại và bạn giữ một thẻ khách hàng trung thành, bạn sẽ được giảm giá 10%. Nếu bạn có phiếu giảm giá, bạn có thể giảm giá 20% ngay hôm nay (nhưng không thể sử dụng với giảm giá 'khách hàng mới'). Số tiền chiết khấu được thêm, nếu có. Điều này được thể hiện trong Bảng 4.8.

BẢNG 4.8 Bảng quyết định ví dụ về thẻ tín dụng



Trong Bảng 4.8, các điều kiện và hành động được liệt kê trong cột bên trái. Tất cả các cột khác trong bảng quyết định đều đại diện cho một quy tắc riêng biệt, một cho mỗi sự kết hợp các điều kiện. Chúng tôi có thể chọn để kiểm thử mỗi quy tắc / sự kết hợp và nếu chỉ có một vài điều này sẽ là trường hợp. Tuy nhiên, nếu số lượng các quy tắc / kết hợp lớn, chúng tôi có nhiều khả năng lấy mẫu chúng bằng cách chọn một tập hợp con phong phú để thử nghiệm.

Lưu ý rằng chúng tôi đã đặt X cho chiết khấu cho hai trong số các cột (Quy tắc 1 và 2) - điều này có nghĩa là sự kết hợp này không nên xảy ra. Bạn không thể vừa là khách hàng mới và đã có một thẻ khách hàng thân thiết! Cần có một thông báo lỗi nêu rõ điều này, nhưng ngay cả khi chúng ta không biết thông điệp đó nên là gì, nó vẫn sẽ thực hiện một bài kiểm thử tốt.

Chúng tôi đã có một giả định trong Quy tắc 3. Vì phiếu giảm giá có mức chiết khấu cao hơn khách hàng mới giảm giá nên chúng tôi giả định rằng khách hàng sẽ chọn 20% thay vì 15%. Chúng tôi không thể thêm chúng, vì phiếu giảm giá không thể được sử dụng với giảm giá 'khách hàng mới'. Hành động 20% ​​là giả định về phía chúng tôi, và chúng ta nên kiểm thử xem giả định này (và bất kỳ giả định nào khác mà chúng tôi đưa ra) là đúng, bằng cách yêu cầu người đã viết bản mô tả hoặc người sử dụng.

Tuy nhiên, đối với Quy tắc 5, chúng tôi có thể thêm chiết khấu do cả phiếu giảm giá và chiết khấu thẻ khách hàng trung thành phải được áp dụng (ít nhất đó là giả định của chúng tôi).

Các quy tắc 4, 6 và 7 chỉ có một loại chiết khấu và Quy tắc 8 không được giảm giá. vì vậy 0%.

Nếu chúng tôi đang áp dụng kỹ thuật này kỹ lưỡng, chúng tôi sẽ có một bài kiểm thử cho mỗi cột hoặc quy tắc của bảng quyết định của chúng tôi. Lợi thế của việc này là chúng ta có thể thử nghiệm một sự kết hợp của những thứ mà nếu không chúng tôi có thể không có thử nghiệm và có thể tìm thấy một khiếm khuyết.

Tuy nhiên, nếu chúng ta có nhiều kết hợp, có thể không thể hoặc hợp lý để kiểm thử mọi kết hợp. Nếu chúng ta bị hạn chế về thời gian, chúng tôi có thể không có thời gian để kiểm thử tất cả các kết hợp. Không chỉ giả định rằng tất cả các kết hợp cần phải được kiểm thử; tốt hơn là nên ưu tiên và kiểm thử các kết hợp quan trọng nhất. Có bảng đầy đủ cho phép chúng tôi xem kết hợp nào chúng tôi đã quyết định thử nghiệm và không kiểm thử lần này.

Cũng có thể có nhiều hành động khác nhau do kết hợp các điều kiện. Trong ví dụ trên, chúng ta chỉ có một: giảm giá sẽ được áp dụng. Bảng quyết định chỉ ra các hành động nào áp dụng cho mỗi sự kết hợp các điều kiện.

Trong ví dụ trên tất cả các điều kiện là nhị phân, tức là họ chỉ có hai giá trị có thể: Đúng hay Sai (hoặc, nếu bạn thích Có hoặc Không). Thường thì trường hợp điều kiện phức tạp hơn, có nhiều tiềm năng có thể xảy ra. Trường hợp này, số lượng kết hợp có thể là rất lớn, vì vậy kết hợp chỉ có thể được lấy mẫu thay vì tập thể dục tất cả chúng.

**4.3.3 Kiểm thử chuyển đổi trạng thái**

Xét nghiệm chuyển đổi trạng thái được sử dụng khi một số khía cạnh của hệ thống có thể được mô tả trong cái được gọi là 'máy trạng thái hữu hạn'. Điều này đơn giản có nghĩa là hệ thống có thể nằm trong một số (hữu hạn) các trạng thái khác nhau, và sự chuyển tiếp từ một trạng thái này sang trạng thái khác được xác định bởi các quy tắc của 'máy'. Đây là mô hình mà trên đó hệ thống và các bài kiểm thử được dựa. Bất kỳ hệ thống nào mà bạn nhận được một đầu ra khác nhau cho cùng một đầu vào, tùy thuộc vào những gì đã xảy ra trước đó, là một hệ thống trạng thái hữu hạn. Một hệ thống trạng thái hữu hạn thường được biểu diễn dưới dạng sơ đồ trạng thái (xem hình 4.2).

Ví dụ: nếu bạn yêu cầu rút 100 đô la từ máy ATM của ngân hàng, bạn có thể được cấp tiền mặt. Sau đó, bạn có thể thực hiện chính xác cùng một yêu cầu nhưng sẽ bị từ chối tiền (vì số dư của bạn không đủ). Việc từ chối này sau đó là do tiểu bang trong tài khoản ngân hàng của bạn đã thay đổi từ việc có đủ tiền để trang trải việc rút tiền để không có đủ tiền. Giao dịch đã gây ra tài khoản của bạn để thay đổi trạng thái của nó có thể là thu hồi sớm hơn. Một sơ đồ trạng thái có thể mô tả một mô hình theo quan điểm của hệ thống, tài khoản hoặc khách hàng.

Một ví dụ khác là một trình xử lý văn bản. Nếu một tài liệu được mở, bạn có thể đóng nó. Nếu không có tài liệu nào được mở, thì 'Đóng' không có sẵn. Sau khi bạn chọn 'Đóng' một lần, bạn không thể chọn lại cho cùng một tài liệu trừ khi bạn mở tài liệu đó. Một tài liệu do đó có hai trạng thái: mở và đóng.

Một mô hình chuyển đổi trạng thái có bốn phần cơ bản:

• Các tiểu bang mà phần mềm có thể chiếm (quỹ mở / đóng cửa hoặc tài trợ / không đầy đủ);

• Chuyển tiếp từ một bang sang bang khác (không phải tất cả các quá trình chuyển đổi được cho phép);

• các sự kiện gây ra việc chuyển đổi (đóng hồ sơ hoặc rút tiền);

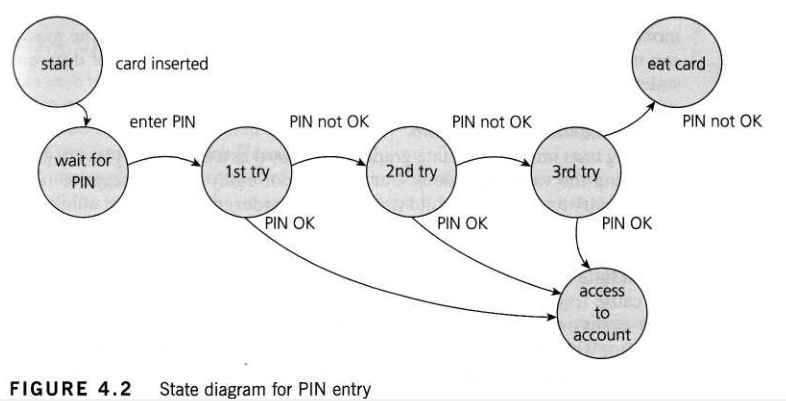
• các hành động phát sinh từ quá trình chuyển đổi (thông báo lỗi hoặc được cấp tiền mặt).

Lưu ý rằng trong bất kỳ trạng thái nào, một sự kiện chỉ có thể gây ra một hành động, nhưng sự kiện tương tự - từ một trạng thái khác - có thể gây ra hành động khác và trạng thái kết thúc khác.

Chúng ta sẽ xem xét đầu tiên các trường hợp thử nghiệm thực hiện chuyển tiếp trạng thái hợp lệ.

Hình 4.2 cho thấy một ví dụ về nhập Mã số Cá nhân (PIN) vào tài khoản ngân hàng. Các trạng thái được hiển thị dưới dạng vòng tròn, sự chuyển tiếp dưới dạng các đường có mũi tên và các sự kiện làm văn bản gần các quá trình chuyển tiếp. (Chúng tôi đã không hiển thị các hành động rõ ràng trên biểu đồ này, nhưng chúng sẽ là một thông báo cho khách hàng nói những điều như 'Hãy nhập mã PIN của bạn')

Biểu đồ trạng thái cho biết bảy trạng thái nhưng chỉ có bốn sự kiện có thể xảy ra (Thẻ đã chèn, Nhập PIN, PIN OK và PIN không OK). Chúng tôi chưa xác định tất cả các bước chuyển tiếp có thể xảy ra ở đây - cũng sẽ có thời gian chờ đợi từ 'chờ PIN' và từ ba lần thử sẽ quay trở lại trạng thái bắt đầu sau thời gian trôi qua và có thể sẽ tháo thẻ. Cũng sẽ có sự chuyển đổi từ trạng thái 'thẻ ăn' trở lại trạng thái bắt đầu. Chúng tôi đã không chỉ định tất cả các sự kiện có thể - hoặc sẽ có một tùy chọn 'hủy bỏ' từ 'chờ đợi cho PIN'



và từ ba lần thử, cũng sẽ quay trở lại trạng thái bắt đầu và tháo thẻ ra. Trạng thái 'tài khoản truy cập' sẽ là sự khởi đầu của một sơ đồ trạng thái khác cho thấy các giao dịch hợp lệ hiện có thể được thực hiện trên tài khoản.

Tuy nhiên, biểu đồ trạng thái này, mặc dù nó không đầy đủ, vẫn cho chúng ta thông tin để thiết kế một số bài kiểm thử hữu ích và giải thích kỹ thuật chuyển tiếp trạng thái.

Trong các trường hợp thử nghiệm bắt nguồn, chúng ta có thể bắt đầu với một kịch bản điển hình. Một trường hợp thử nghiệm đầu tiên hợp lý ở đây sẽ là tình huống bình thường, nơi mã PIN chính xác được nhập vào lần đầu tiên. Để hiểu rõ hơn, chúng tôi có thể muốn đảm bảo rằng chúng tôi bao gồm mọi trạng thái (tức là ít nhất một bài kiểm thử đi qua từng tiểu bang) hoặc chúng tôi có thể muốn đề cập đến mọi quá trình chuyển đổi. Thử nghiệm thứ hai (để truy cập vào mỗi tiểu bang) sẽ nhập một mã PIN không chính xác mỗi lần để hệ thống ăn thẻ. Chúng tôi vẫn chưa kiểm thử mọi quá trình chuyển đổi. Để làm điều đó, chúng tôi sẽ muốn thử nghiệm nơi mã PIN không chính xác lần đầu tiên nhưng OK lần thứ hai và một bài kiểm thử khác nơi mã PIN chính xác vào lần thử thứ ba.Những thử nghiệm này có lẽ ít quan trọng hơn so với hai lần đầu tiên.

Lưu ý rằng quá trình chuyển đổi không cần phải chuyển sang trạng thái khác (mặc dù tất cả các chuyển tiếp được hiển thị ở trên đều chuyển sang trạng thái khác). Vì vậy, có thể có một sự chuyển đổi từ 'tài khoản truy cập' mà chỉ cần trở lại 'truy cập tài khoản' cho một hành động như 'cân bằng yêu cầu'.

Các điều kiện kiểm thử có thể được bắt nguồn từ biểu đồ trạng thái theo nhiều cách khác nhau. Mỗi tiểu bang có thể được lưu ý như là một điều kiện thử nghiệm, cũng như mỗi quá trình chuyển đổi. Trong Chương trình học, chúng ta cần có khả năng xác định phạm vi bảo hiểm của một tập hợp các bài kiểm thử về chuyển tiếp.

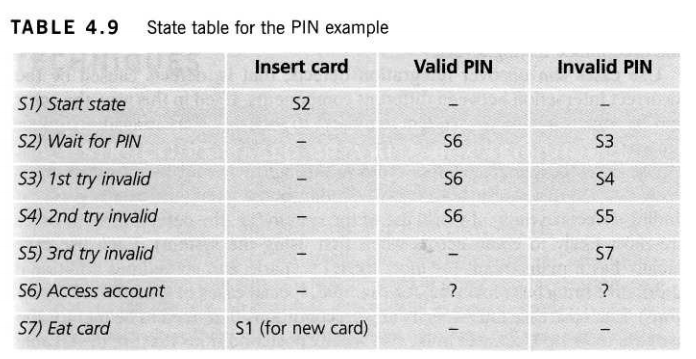
Đi vượt quá mức dự kiến ​​trong Chương trình học, chúng tôi cũng có thể xem xét các cặp chuyển đổi và ba lần và như vậy. Phạm vi bảo hiểm của tất cả các chuyển tiếp riêng lẻ còn được gọi là bảo hiểm chuyển đổi 0, vùng phủ sóng của các cặp chuyển đổi là độ phủ sóng chuyển đổi l, phạm vi bảo hiểm của các chuyển đổi ba lần là độ bao phủ 2 chuyển đổi, vv Các trường hợp thử nghiệm xuất phát từ mô hình chuyển đổi trạng thái là một phương pháp tiếp cận hộp đen . Đo bao nhiêu bạn đã kiểm thử (bao phủ) đang nhận được gần một quan điểm hộp trắng. Tuy nhiên, thử nghiệm chuyển đổi trạng thái được coi là một kỹ thuật hộp đen.

Một trong những ưu điểm của kỹ thuật chuyển đổi trạng thái là mô hình có thể chi tiết hoặc trừu tượng như bạn cần. Trường hợp một phần của hệ thống là quan trọng hơn (có nghĩa là đòi hỏi phải có nhiều thử nghiệm) thì có thể mô phỏng chiều sâu của chi tiết hơn. Trường hợp hệ thống không quan trọng (đòi hỏi ít kiểm thử), mô hình có thể sử dụng một trạng thái để biểu hiện những gì sẽ là một loạt các trạng thái khác nhau.

**Thử nghiệm chuyển tiếp không hợp lệ**

Các bài kiểm thử tạo ra chỉ từ biểu đồ trạng thái (còn được gọi là biểu đồ trạng thái) rất hữu ích khi nhìn thấy các chuyển tiếp hợp lệ, nhưng chúng ta không thể dễ dàng nhìn thấy những bài kiểm thử tiêu cực, ở đó chúng ta cố gắng tạo ra các chuyển tiếp không hợp lệ. Để xem tổng số kết hợp các trạng thái và sự chuyển tiếp, cả hai hợp lệ và không hợp lệ, một bảng trạng thái là hữu ích.

Bảng trạng thái liệt kê tất cả các tiểu bang xuống một bên của bảng và tất cả các sự kiện gây ra sự chuyển tiếp dọc theo phía trên (hoặc ngược lại). Mỗi tế bào sau đó đại diện cho một cặp sự kiện trạng thái. Nội dung của mỗi ô cho biết trạng thái mà hệ thống sẽ di chuyển đến, khi sự kiện tương ứng xảy ra trong khi ở trạng thái liên kết. Điều này sẽ bao gồm các sự kiện sai lầm có thể xảy ra - những sự kiện không được mong đợi xảy ra ở một số tiểu bang. Đây là những điều kiện kiểm thử âm.



Bảng 4.9 liệt kê các trạng thái trong cột thứ nhất và các đầu vào có thể trên hàng đầu. Ví dụ, nếu hệ thống ở trạng thái 1, việc chèn một thẻ sẽ đưa nó đến trạng thái 2. Nếu chúng tôi ở trong tiểu bang 2 và nhập PIN hợp lệ, chúng tôi chuyển đến Tiểu bang 6 để truy cập vào tài khoản. Ở tiểu bang 2 nếu chúng tôi nhập mã PIN không hợp lệ, chúng tôi chuyển đến Tiểu bang 3. Chúng tôi đã đặt một dấu gạch ngang trong các tế bào mà nên là không thể, tức là họ đại diện cho sự chuyển tiếp không hợp lệ từ trạng thái đó.

Chúng tôi đã đặt dấu chấm hỏi cho hai ô, nơi chúng tôi nhập mã PIN hợp lệ hoặc không hợp lệ khi chúng tôi truy cập vào tài khoản. Có lẽ hệ thống sẽ lấy số PIN của chúng tôi như là số tiền thu hồi? Nó có thể là một thử nghiệm tốt! Hầu hết các ô không hợp lệ khác sẽ không thể thực hiện được trong ví dụ này. Các bài kiểm thử không hợp lệ (tiêu cực) sẽ cố gắng tạo các chuyển tiếp không hợp lệ, các chuyển tiếp không nên có thể (nhưng thường xuyên thực hiện những bài kiểm thử tốt khi chúng có thể xảy ra).

Mô tả rộng hơn về máy móc của nhà nước được tìm thấy trong [Marick, 1994]. Kiểm thử chuyển đổi trạng thái cũng được mô tả trong [Craig, 2002], [Copeland, 2003], [Beizer, 1990] và [Broekman, 2003]. Thử nghiệm chuyển tiếp trạng thái được mô tả trong BS7925-2, bao gồm các thử nghiệm thiết kế và các biện pháp bảo hiểm.

**4.3.4 Kiểm thử các ca sử dụng**

Kiểm thử các ca sử dụng là một kỹ thuật giúp chúng tôi xác định các trường hợp thử nghiệm thực hiện toàn bộ hệ thống trên một giao dịch theo cơ sở giao dịch từ đầu đến cuối. Chúng được mô tả bởi Ivar Jacobson trong cuốn sách của ông về Công nghệ phần mềm hướng đối tượng: Cách tiếp cận sử dụng các trường hợp sử dụng [Jacobson, 1992].

Trường hợp sử dụng là mô tả của một người sử dụng cụ thể sử dụng hệ thống (người sử dụng hệ thống). Mỗi trường hợp sử dụng mô tả các tương tác mà tác nhân có với hệ thống để đạt được một nhiệm vụ cụ thể (hoặc, ít nhất, tạo ra giá trị cho người dùng). Các diễn viên nói chung là những người nhưng cũng có thể là những hệ thống khác. Trường hợp sử dụng là một chuỗi các bước mô tả các tương tác giữa diễn viên và hệ thống.

Các trường hợp sử dụng được định nghĩa theo thuật ngữ của diễn viên, chứ không phải hệ thống, mô tả diễn xuất của diễn viên và diễn xuất của diễn viên chứ không phải là những gì đầu vào hệ thống mong muốn và hệ thống sẽ đưa ra. Họ thường sử dụng ngôn ngữ và các điều khoản của doanh nghiệp hơn là các thuật ngữ kỹ thuật, đặc biệt là khi người đó là người sử dụng kinh doanh. Chúng phục vụ như là nền tảng cho việc phát triển các trường hợp thử nghiệm chủ yếu ở hệ thống và các mức thử nghiệm chấp nhận.

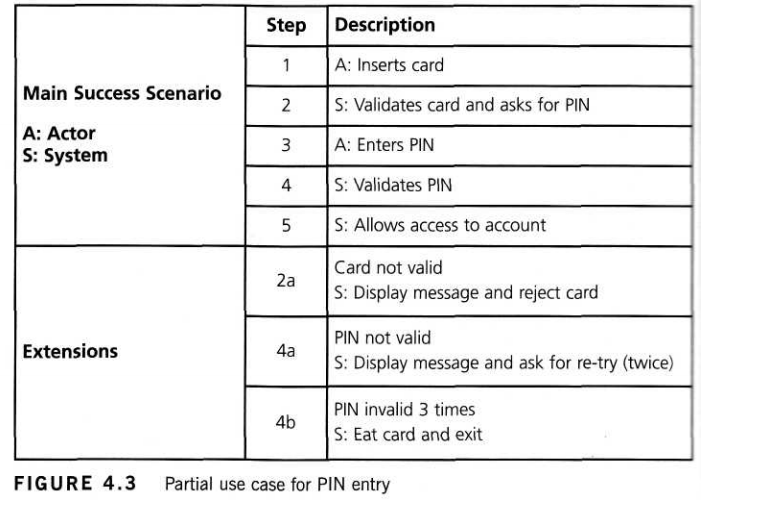
Trường hợp sử dụng có thể phát hiện khuyết tật tích hợp, nghĩa là các khiếm khuyết do sự tương tác không chính xác giữa các thành phần khác nhau. Được sử dụng theo cách này, diễn viên có thể là một cái gì đó mà hệ thống giao diện như một liên kết truyền thông hoặc tiểu hệ thống.

Các trường hợp sử dụng mô tả các luồng quá trình thông qua một hệ thống dựa trên việc sử dụng có nhiều khả năng nhất của nó. Điều này làm cho các trường hợp thử nghiệm bắt nguồn từ các trường hợp sử dụng đặc biệt tốt cho việc tìm ra các khiếm khuyết trong việc sử dụng thực tế của hệ thống (tức là các khiếm khuyết mà người dùng có thể gặp phải khi sử dụng hệ thống lần đầu tiên). Mỗi trường hợp sử dụng thường có kịch bản chính (hoặc có khả năng xảy ra) và đôi khi là các chi nhánh khác thay thế (bao gồm, ví dụ, các trường hợp đặc biệt hoặc các điều kiện đặc biệt). Mỗi trường hợp sử dụng phải xác định điều kiện tiên quyết để đáp ứng yêu cầu sử dụng của trường hợp sử dụng. Trường hợp sử dụng cũng phải xác định điều kiện hậu điều tra là kết quả quan sát được và mô tả trạng thái cuối cùng của hệ thống sau khi trường hợp sử dụng đã được thực hiện thành công.

Ví dụ về mã PIN mà chúng tôi sử dụng để kiểm thử chuyển tiếp trạng thái cũng có thể được định nghĩa trong các trường hợp sử dụng, như thể hiện trong hình 4.3. Chúng tôi cho thấy một kịch bản thành công và các phần mở rộng (thể hiện những cách mà kịch bản có thể không thành công).

Đối với thử nghiệm trường hợp sử dụng, chúng tôi sẽ có một thử nghiệm của kịch bản thành công và một tesi cho mỗi phần mở rộng. Trong ví dụ này, chúng ta có thể cho phần mở rộng 4b ưu tiên cao hơn 4a từ quan điểm bảo mật.

Các yêu cầu về hệ thống cũng có thể được chỉ định như một tập hợp các trường hợp sử dụng. Cách tiếp cận này có thể làm cho nó dễ dàng hơn để liên quan đến người sử dụng trong quá trình thu thập yêu cầu và định nghĩa.



## 