

Test Design Techniques

By Thuy Nguyen

Agenda

- **Phân vùng tương đương (Equivalence partition)**
- **Phân tích giá trị biên (Boundary value analysis)**
- **Bảng quyết định (Decision table)**
- **Chuyển đổi trạng thái (State transition testing)**

Phân vùng tương đương (Equivalence partition)

- phân vùng tập các điều kiện test vào các nhóm mà các phần tử trong nhóm đó có thể xem là như nhau khi dùng để kiểm thử

- Ví dụ:

Tỉ lệ lãi suất phụ thuộc vào số dư tài khoản (balance) trong ngân hàng. Phần mềm có chức năng tính toán và đưa ra tiền lãi thích hợp:

\$0 - \$100 : 3%

\$100.01 - \$ 999.99: 5%

Lớn hơn hoặc bằng \$ 1000 : 7%

Phân vùng tương đương (Equivalence partition)

Invalid partition	Valid (for 3% interest)	Valid (for 5%)	Valid (for 7%)		
-\$0.01	\$0.00	\$100.00	\$100.01	\$999.99	\$1000.00

Phân tích giá trị biên (Boundary value analysis)

Phân tích giá trị biên dựa trên giá trị biên giữa các phân vùng. Cần phải chú ý rằng ta có cả giá trị bên cho cả 2 vùng valid và invalid

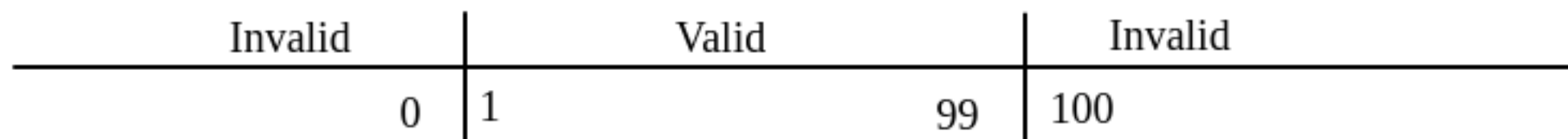
Với ví dụ interest payment trên, ta có các giá trị biên như sau:

Invalid boundary value:

-\$0.01

Valid boundary value:

\$0.00, \$100.00, \$100.01, \$999.99 and \$1000.00



3. Decision table

Tại sao lại dùng decision table?

Phân vùng tương đương và Phân tích giá trị biên

thường được áp dụng cho một input. Trong trường hợp kết hợp nhiều input trong một chức năng, rất khó để sử dụng Phân vùng tương đương hay Phân tích giá trị biên. Có 2 phương pháp khác nữa có thể áp dụng trong trường hợp này đó là **Bảng quyết định** (Decision table) và **Chuyển đổi trạng thái** (State transition)

3. Decision table

Sử dụng Bảng quyết định trong quá trình kiểm thử:

Bước 1: Nhiệm vụ đầu tiên là xác định các chức năng cần sử dụng bảng quyết định để kiểm thử

Bước 2: Xác định các condition sau đó chia chúng thành các tập con và xử lý với từng tập con một

TABLE 4.2 Empty decision table

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
<i>Repayment amount has been entered</i>				
<i>Term of loan has been entered</i>				

Bước 3: Xác định tất cả các sự kết hợp của True và False.

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
<i>Repayment amount has been entered</i>	T	T	F	F
<i>Term of loan has been entered</i>	T	F	T	F

TABLE 4.4 Decision table with combinations and outcomes

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
<i>Repayment amount has been entered</i>	T	T	F	F
<i>Term of loan has been entered</i>	T	F	T	F
Actions/Outcomes				
<i>Process loan amount</i>	Y	Y		
<i>Process term</i>	Y		Y	

Bước 4: Cuối cùng là viết test case để thực thi các rule đã có ở trên.

Nếu bạn có một khách hàng mới mở một tài khoản credit card, bạn sẽ có **15%** lãi suất chiết khấu (discount) trong tất cả các hoạt động mua sắm trong ngày hôm nay. Nếu bạn đã là khách hàng trung thành, bạn có **10%** lãi suất chiết khấu. Nếu bạn có phiếu mua hàng (coupon)(nhưng k phải là khách hàng mới), bạn được giảm giá **20%** trong ngày hôm nay.

Từ đặc tả yêu cầu trên ta có các điều kiện sau:

- **New customer (15%)**
- **Loyalty card (10%)**
- **Coupon (20%)**

Ta có 3 điều kiện => ta có 8 rule cần test. Lập bảng quyết định như sau:

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
New customer (15%)	T	T	T	T	F	F	F	F
Loyalty card (10%)	T	T	F	F	T	T	F	F
Coupon (20%)	T	F	T	F	T	F	T	F
Actions								
Discount (%)	X	X	20	15	30	10	20	0

4. State transtition testing

State transition testing được sử dụng khi một vài mặt của hệ thống được mô tả bằng máy trạng thái. Hiểu một cách đơn giản là hệ thống có thể ở một số lượng (có hạn) các trạng thái và phiên làm việc từ một trạng thái này đến một trạng thái kia được qui định bởi một rule của máy trạng thái.

Một mô hình chuyển đổi trạng thái bao gồm 4 phần sau:

- Các trạng thái (state) mà chương trình có thể có (ví dụ như không đủ số dư, tài khoản hết hạn...)
- Các phiên chuyển đổi (transition) từ một trạng thái này đến trạng thái kia
- Các sự kiện (event) gây ra phiên chuyển đổi (ví dụ như nhập mã PIN, đóng file...)
- Các action là kết quả của phiên chuyển đổi (ví dụ như message báo lỗi, trả lại thẻ ATM...)

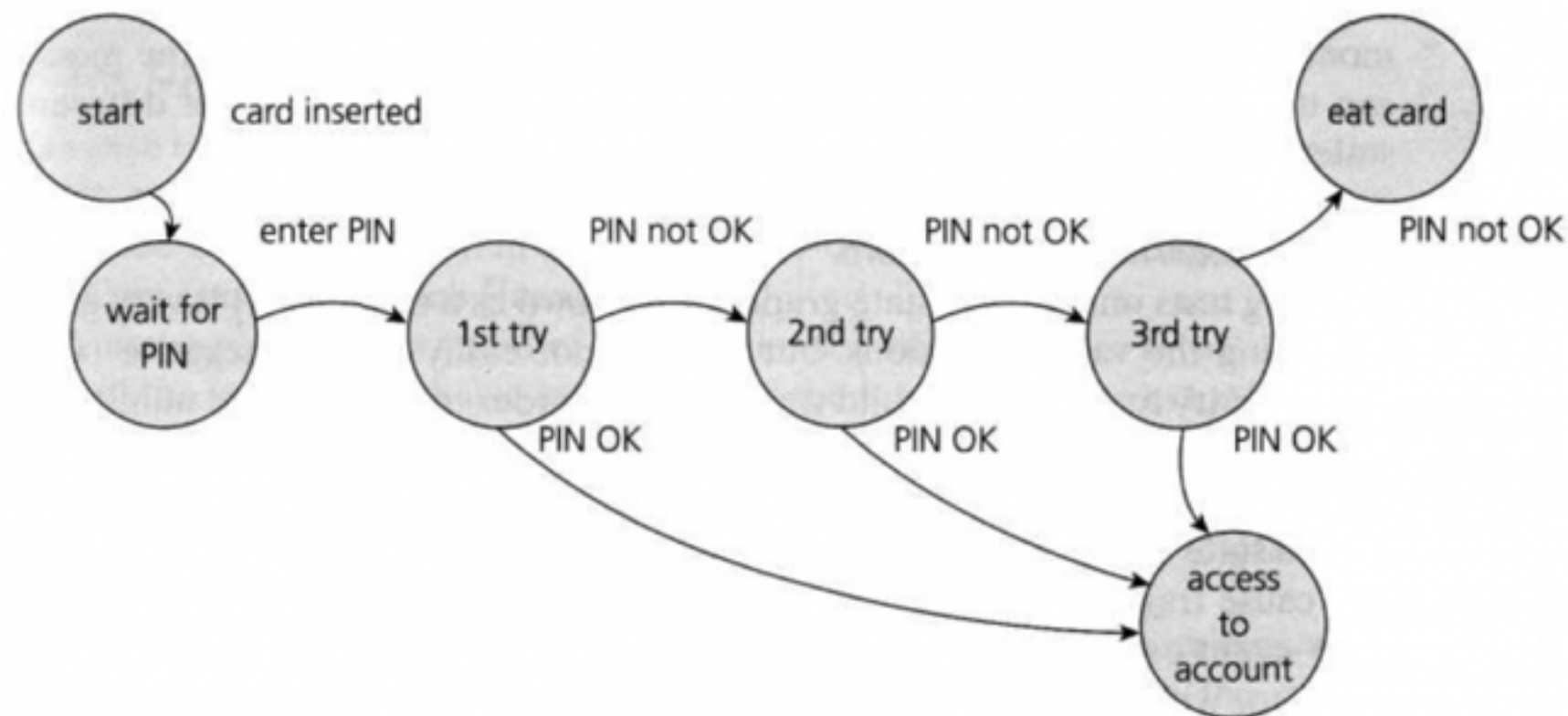


FIGURE 4.2 State diagram for PIN entry

Example 1

Yêu cầu:

Nếu bạn đi xe điện chuyển trước 9:30 sáng hoặc từ sau 4:00 chiều đến 7:30 tối (giờ cao điểm), thì bạn phải mua vé thường. Vé tiết kiệm (giá thấp hơn vé thường) có hiệu lực cho các chuyến xe từ 9:30 sáng đến 4:00 chiều và sau 7:30 tối.

Các bạn có câu hỏi nào cho yêu cầu trên không? Có chỗ nào chưa rõ không?

Dựa vào yêu cầu trên, các bạn hãy:

- + Liệt kê ra các vùng và các giá trị biên để kiểm thử thời gian của tàu đối với các loại vé.
- + Liệt kê các vùng hợp lệ và không hợp lệ. Cho biết đâu là giá trị biên (Dùng bảng để dễ liệt kê các vùng tương đương và các giá trị biên)
- + Viết test case để test yêu cầu trên dựa vào các vùng tương đương và giá trị biên.

Example 2

Giỏ hàng trên một trang mua bán trực tuyến được bắt đầu với trạng thái là rỗng (không có món hàng nào). Khi bạn chọn một sản phẩm thì nó sẽ được đưa vào giỏ hàng. Bạn cũng có thể bỏ chọn các món hàng trong giỏ hàng. Khi bạn quyết định mua hàng, thì sẽ xuất hiện màn hình tổng hợp các món hàng đang có trong giỏ cùng với thông tin về giá tiền, số lượng và tổng tiền của giỏ hàng, để cho bạn xác nhận xem đúng hay chưa. Nếu bạn thấy số lượng hàng và giá tiền OK thì bạn sẽ được chuyển sang trang thanh toán. Ngược lại bạn sẽ quay lại trang mua hàng (lúc này bạn có thể bỏ chọn các món hàng bạn muốn bỏ bớt).

Yêu cầu:

- a. Đưa ra sơ đồ trạng thái - state diagram – cho thấy các trạng thái/states và sự chuyển tiếp/transition khác. Xác định test case – một loạt các trạng thái – bao phủ toàn bộ các chuyển tiếp.
- b. Đưa ra một bảng trạng thái. Cho một ví dụ kiểm thử trường hợp chuyển tiếp không hợp lệ.

Example 3

Nếu bạn có thẻ đường sắt "over 60s" thì được giảm giá 34% trên tất cả các vé bạn mua. Nếu bạn đi cùng với trẻ em (dưới 16 tuổi), thì bạn sẽ được giảm 50% nếu bạn có thẻ "family rail card", ngược lại bạn sẽ được giảm 10%. Bạn chỉ được sử dụng 1 loại thẻ đường sắt.

Thank you !