



ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM

KIỂM THỬ PHẦN MỀM

Bài 4:

THIẾT KẾ TEST CASE BACKBOX

Thời gian: 6 tiết



NỘI DUNG

4.1. Tổng quan kiểm thử hộp đen.

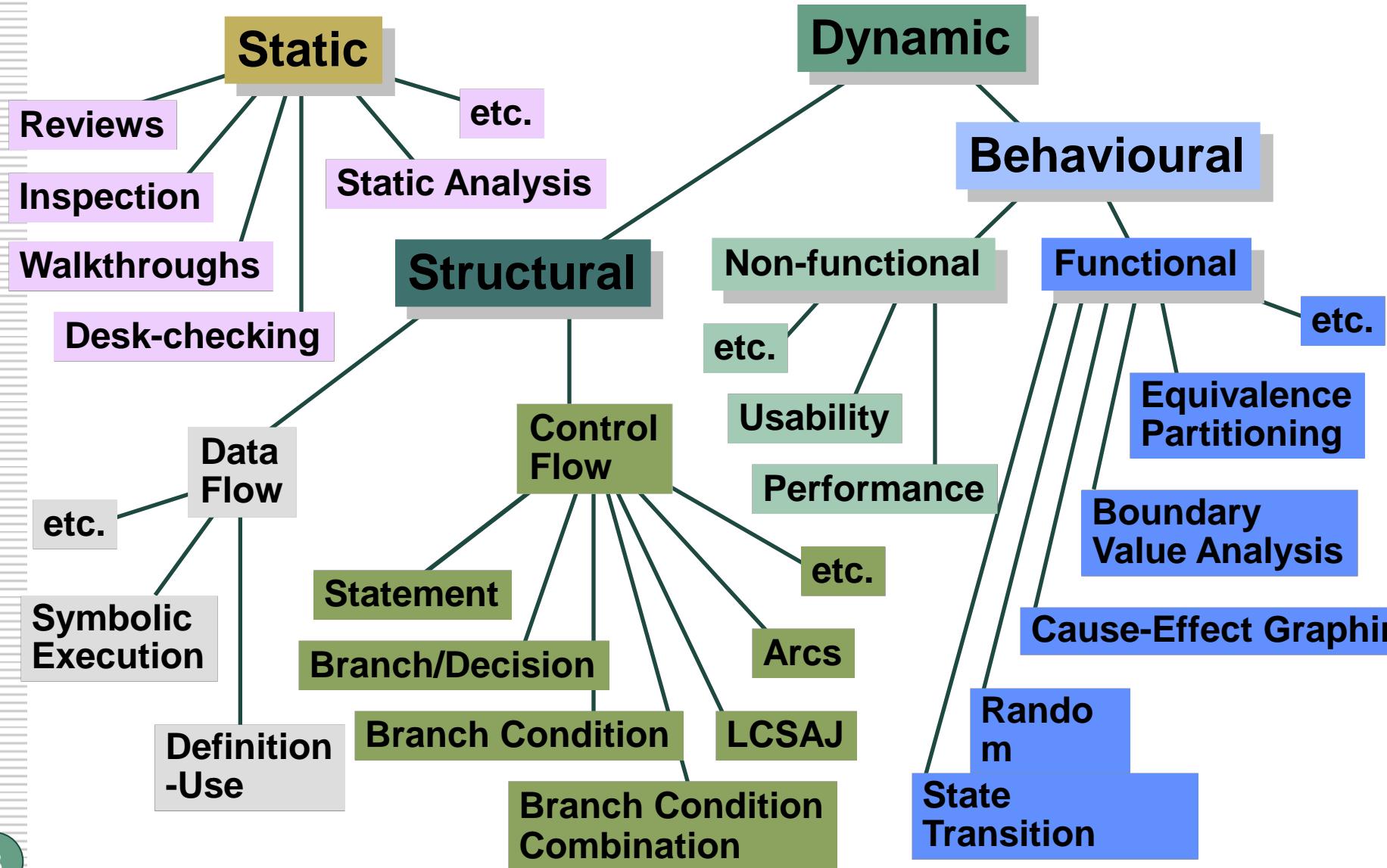
4.2. Quy trình kiểm thử hộp đen.

4.3. Các kỹ thuật kiểm thử hộp đen.

- ✓ Phân vùng tương đương.
- ✓ Phân tích giá trị biên.
- ✓ Bảng quyết định => Đồ thị nguyên nhân – kết quả
- ✓ Dịch chuyển trạng thái.

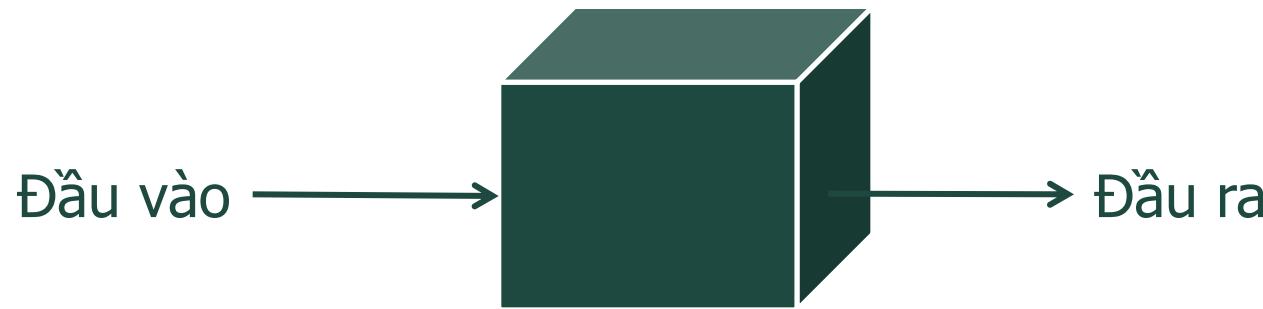
4.4. Một số loại kiểm thử hộp đen.

Some test techniques



4.1. TỔNG QUAN KIỂM THỬ HỘP ĐEN

- ❖ Kiểm thử hộp đen (black-box testing) còn gọi kiểm thử **dựa trên đặc tả** (specification-based testing) vì thông tin duy nhất làm cơ sở để kiểm thử hộp đen là bảng đặc tả yêu cầu chức năng của từng thành phần phần mềm.
- ❖ Tester tập trung vào những gì phần mềm làm được (**WHAT**), không quan tâm nó làm như thế nào (**HOW**).



4.1. TỔNG QUAN KIỂM THỬ HỘP ĐEN

❖ Ưu điểm

- Không cần truy cập mã nguồn.
- Tách biệt khung nhìn của user và developer.
- Nhiều người có thể tham gia test.

❖ Khuyết điểm

- Kiểm thử không có hiệu quả cao.
- Khó thiết kế test case.
- Khó kiểm thử phủ được hết các trường hợp.
- Không có định hướng kiểm thử rõ ràng.

4.2. QUY TRÌNH KIỂM THỬ HỘP ĐEN

- ✓ Phân tích các đặc tả chức năng của các thành phần phần mềm.
- ✓ Thiết kế test case để kiểm thử.
- ✓ Thực thi các test case để kiểm thử.
- ✓ So sánh kết quả đạt được với kết quả mong muốn trong từng test case.
- ✓ Lập báo cáo kết quả kiểm thử.

4.3. CÁC KỸ THUẬT KIỂM THỬ HỘP ĐEN

- 4.3.1. Phân vùng tương đương (Equivalence Partitioning hoặc Equivalence Class)
- 4.3.2. Phân tích giá trị biên (Boundary Value Analysis)
- 4.3.3. Bảng quyết định (Decision Tables hay còn gọi là Cause Effect)
- 4.3.4. Dịch chuyển trạng thái (State Transition Testing)

4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

- ✓ Ý tưởng của kỹ thuật này nếu một **giá trị đại diện** trong nhóm đúng thì các giá trị còn lại trong nhóm cũng đúng và ngược lại.
- ✓ Phương pháp phù hợp cho các bài toán có giá trị đầu vào là một miền xác định.

4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

- ✓ Phân vùng tương đương (EP) là phân chia một tập các điều kiện kiểm thử thành các tập con có các giá trị tương đương nhau và kiểm thử các tập con này.
- ✓ Mục đích của EP là **giảm thiểu số lượng test case** không cần thiết khi kiểm thử.
- ✓ EP có thể áp dụng tất cả cấp độ kiểm thử.

4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

- ✓ Hai giá trị tương đương theo một trong các cách sau:
 - Chúng tương tự nhau (intuitive similarity).
 - Tài liệu đặc tả mô tả chương trình sẽ xử lý chúng theo cùng một cách thức (specified as equivalent)
 - Chúng lái chương trình theo cùng đường (chẳng hạn cùng nhánh if) (equivalent path).
 - Chúng cho cùng kết quả với những giả thiết đưa ra (risk-based).

4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

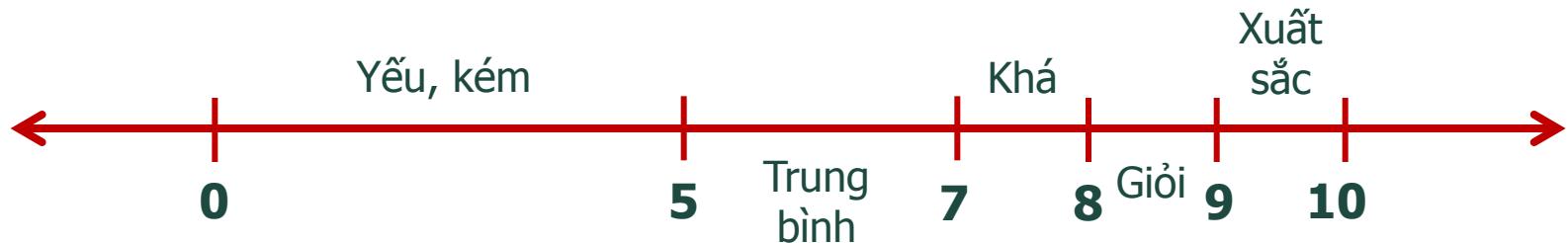
Ví dụ

- ❖ Xếp loại cuối năm học sinh ở trường THPT dựa trên điểm trung bình (ĐTB) như sau:
 - $0 \leq \text{ĐTB} < 5$: yếu, kém
 - $5 \leq \text{ĐTB} < 7$: trung bình
 - $7 \leq \text{ĐTB} < 8$: khá
 - $8 \leq \text{ĐTB} < 9$: giỏi
 - $9 \leq \text{ĐTB} \leq 10$: xuất sắc
- ❖ Biết điểm trung bình làm tròn 1 chữ số thập phân.

4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

Ví dụ

- ✓ Để kiểm thử ứng dụng này chia thành 7 phân vùng tương đương, trong đó:
 - 5 phân vùng hợp lệ (valid): [0, 5), [5, 7), [7, 8), [8, 9), [9, 10].
 - 2 phân vùng không hợp lệ (invalid): < 0 và > 10



4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

Ví dụ

- ✓ Tester không nên chỉ test những gì trong đặc tả yêu cầu, mà còn phải **nghĩ ra những thứ không được đề cập.**
Trong ví dụ trên 2 phần vùng invalid không được đề cập trong đặc tả, nhưng cần được kiểm thử.
- ✓ Khi thiết kế test case phải **đảm bảo tất cả các phân vùng (valid & invalid) được test** qua ít nhất một lần.
Trong ví dụ trên ít nhất cần test 7 điểm trung bình sau đại diện với 7 phân vùng -5.0, 5.5, 7.5, 8.5, 9.5, 12.0

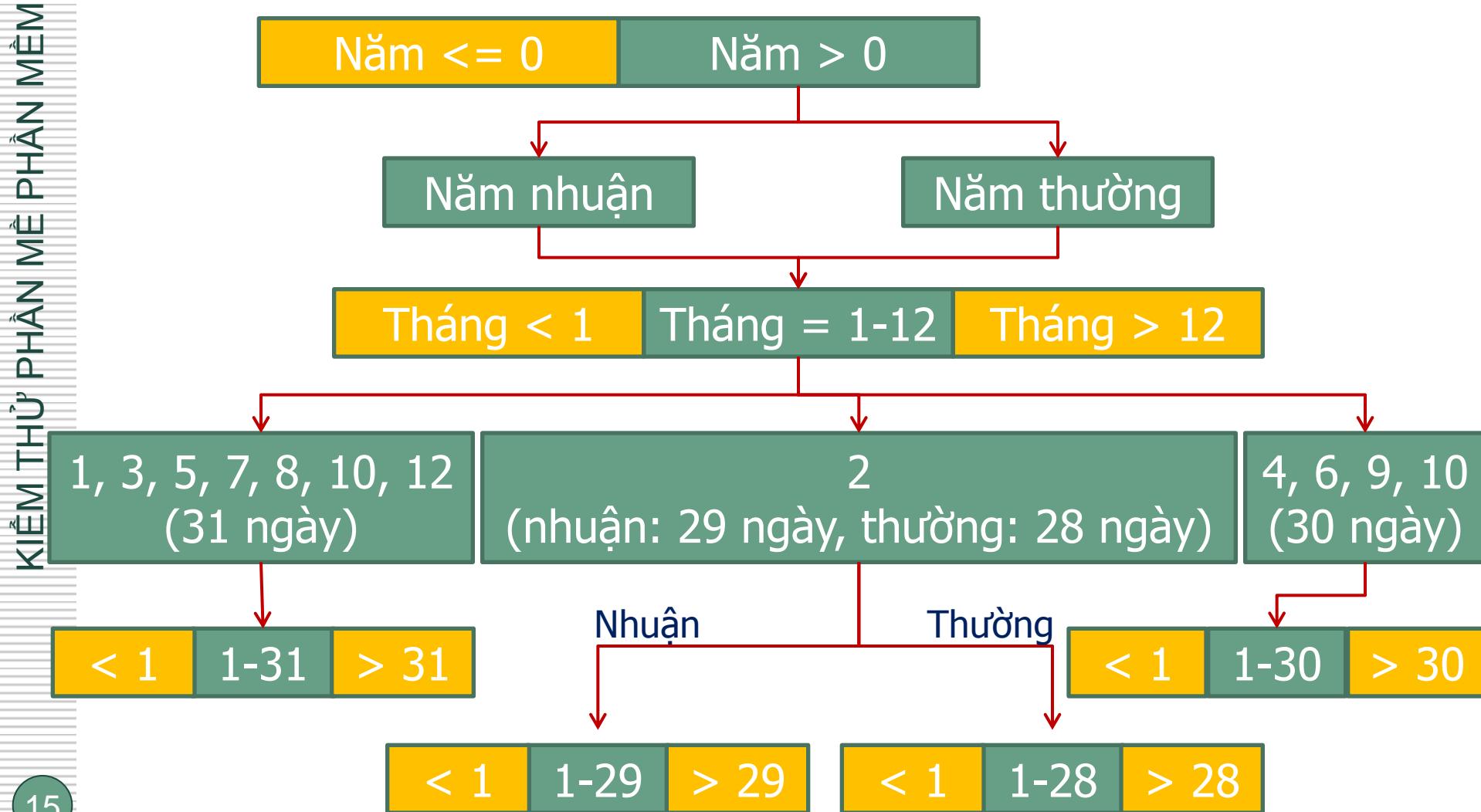
4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

Bài tập 1

- ✓ Chương trình kiểm tra một ngày tháng nhập vào có hợp lệ hay không? Người cần nhập vào năm, tháng, ngày của ngày muốn kiểm tra.
- ✓ Ví dụ 12/12/2017 là ngày hợp lệ, 32/12/2017 là ngày không hợp lệ.
- ✓ *Sử dụng phương pháp phân vùng tương đương thiết kế các test case để kiểm thử chương trình trên.*

4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

Hướng dẫn



4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

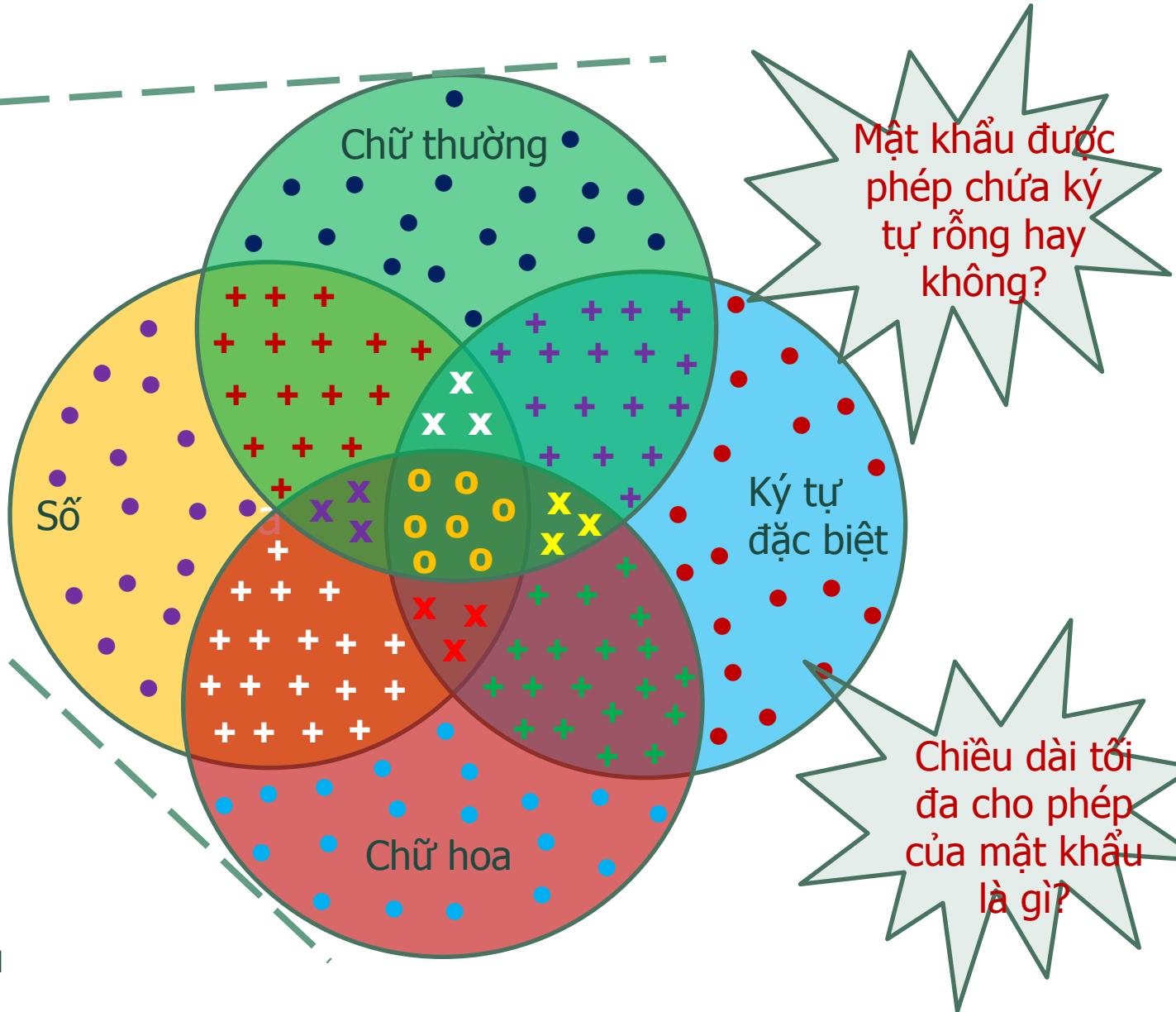
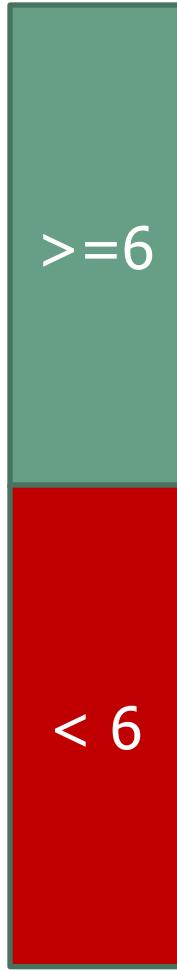
Bài tập 2

- ✓ Chức năng đăng ký tài khoản của một ứng dụng học tập yêu cầu mật khẩu phải có chứa chữ hoa, chữ thường, số, ký tự đặc biệt và chiều dài tối thiểu là 6.
- ✓ *Sử dụng kỹ thuật phân vùng tương đương viết các test case kiểm tra ô nhập liệu mật khẩu.*

4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

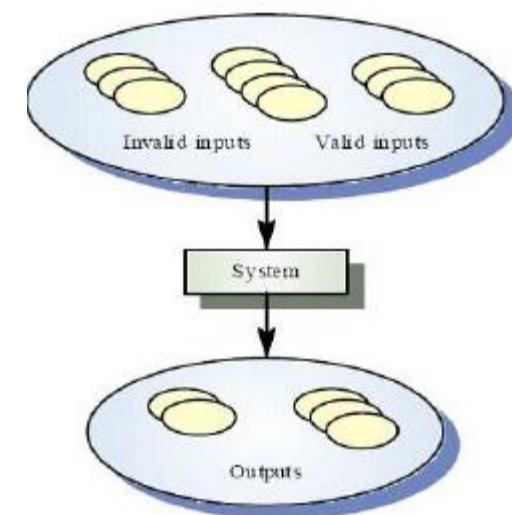
Hướng dẫn

KIỂM THỬ PHẦN MỀM PHẦN MỀM



Kỹ thuật phân lớp tương đương

- ❖ Ý tưởng: Chia miền vào chương trình thành các lớp dữ liệu. Xác định đầu vào hợp lệ và không hợp lệ để lập các ca kiểm thử theo các lớp đó
- ❖ Mỗi lớp dùng để kiểm thử một chức năng, gọi là lớp tương đương.
- ❖ Thay vì kiểm tra tất cả các giá trị đầu vào, có thể lựa chọn từ đầu vào cho riêng từng lớp



Kỹ thuật phân lớp tương đương

- ❖ Nguyên tắc xác định lớp tương đương:
 - Nếu điều kiện đầu vào định rõ giới hạn của một mảng, hoặc một giá trị xác định thì chia vùng tương đương thành:
 - Một lớp tương đương hợp lệ
 - Hai lớp không hợp lệ
 - Một lớp đặc biệt (nếu có)
 - Nếu điều kiện đầu vào chỉ định là một tập giá trị, hoặc xác định là một kiểu đúng sai thì chia vùng tương đương thành :
 - Một lớp tương đương hợp lệ.
 - Một lớp tương đương không hợp lệ.
 - Một lớp đặc biệt (nếu có)

Kỹ thuật phân lớp tương đương

❖ Ví dụ

STT	Trường hợp của điều kiện đầu vào	Lớp tương đương	
		Hợp lệ	Không hợp lệ
1	Là 1 khoảng các giá trị Ví dụ: [3,10]	1 $3 < x < 10$	2 $x < 3$ hoặc $x > 10$
	Là 1 giá trị cụ thể Ví dụ: 6	1 $x = 6$	2 $x < 6$ hoặc $x > 6$
3	Là 1 tập hợp các giá trị Ví dụ: $A = \{1, 5, 4, 8, 6\}$	1 x	1
	Là 1 giá trị Boolean	1	1

Kỹ thuật phân lớp tương đương

- ❖ Ví dụ: Một textbox chỉ cho phép nhập số nguyên từ 1 đến 100
 - Ta không thể nhập tất cả các giá trị từ 1 đến 100
- ❖ **Ý tưởng của kỹ thuật này:** Chia (partition) đầu vào thành những nhóm nhóm tương đương nhau (equivalence).
- ❖ **Giảm đáng kể số lượng test case** cần phải thiết kế vì với mỗi lớp tương đương ta chỉ cần test trên các phần tử đại diện

Kỹ thuật phân lớp tương đương

- ❖ Có hai yếu tố ảnh hưởng đến việc thiết kế test case
 - Dựa trên giả định (Assumption)
 - Single fault assumption → Weak ECT (Equivalence Class Testing)
 - Multiple fault assumption → Strong ECT
 - Dựa trên loại dữ liệu inputs
 - Kiểm thử trên dữ liệu hợp lệ → Normal ECT
 - Kiểm thử trên dữ liệu không hợp lệ → Robust ECT

Assumption Data	Single Fault	Multiple Faults
Valid		
Invalid		

Kỹ thuật phân lớp tương đương

- ❖ Weak Normal Equivalence Class Testing
- ❖ Strong Normal Equivalence Class Testing
- ❖ Weak Robust Equivalence Class Testing
- ❖ Strong Robust Equivalence Class Testing

Weak Normal Equivalence Class Testing

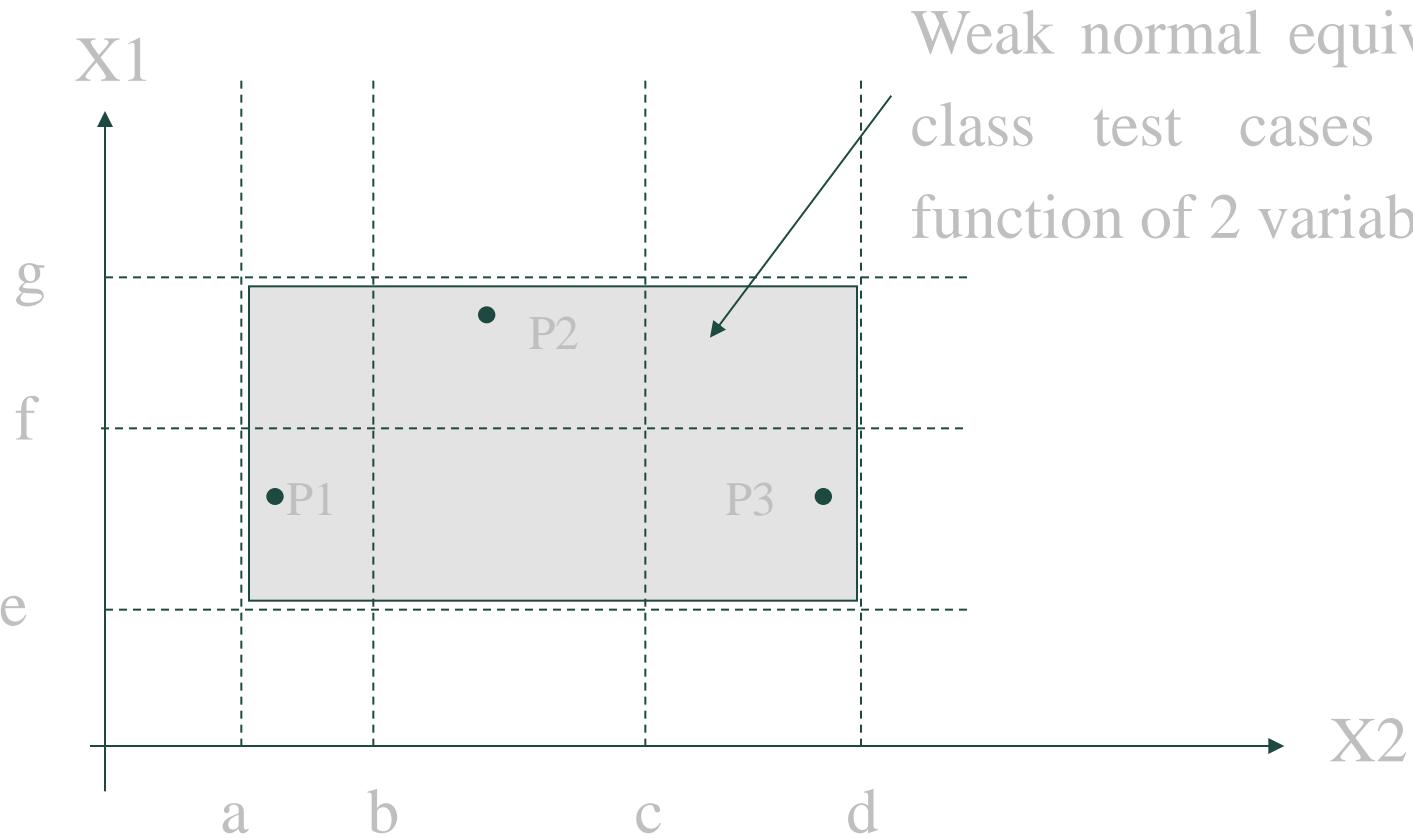
❖ Dựa trên Single Fault Assumption

- Một failure ít khi nào là kết quả của 2 hay nhiều faults xảy ra cùng 1 lúc

❖ Ví dụ:

- $e \leq x_1 \leq g$, x_1 có 2 lớp tương đương $[e, f)$ $[f, g]$
- $a \leq x_2 \leq d$, x_2 có 3 lớp tương đương $[a, b)$ $[b, c)$, $[c, d]$

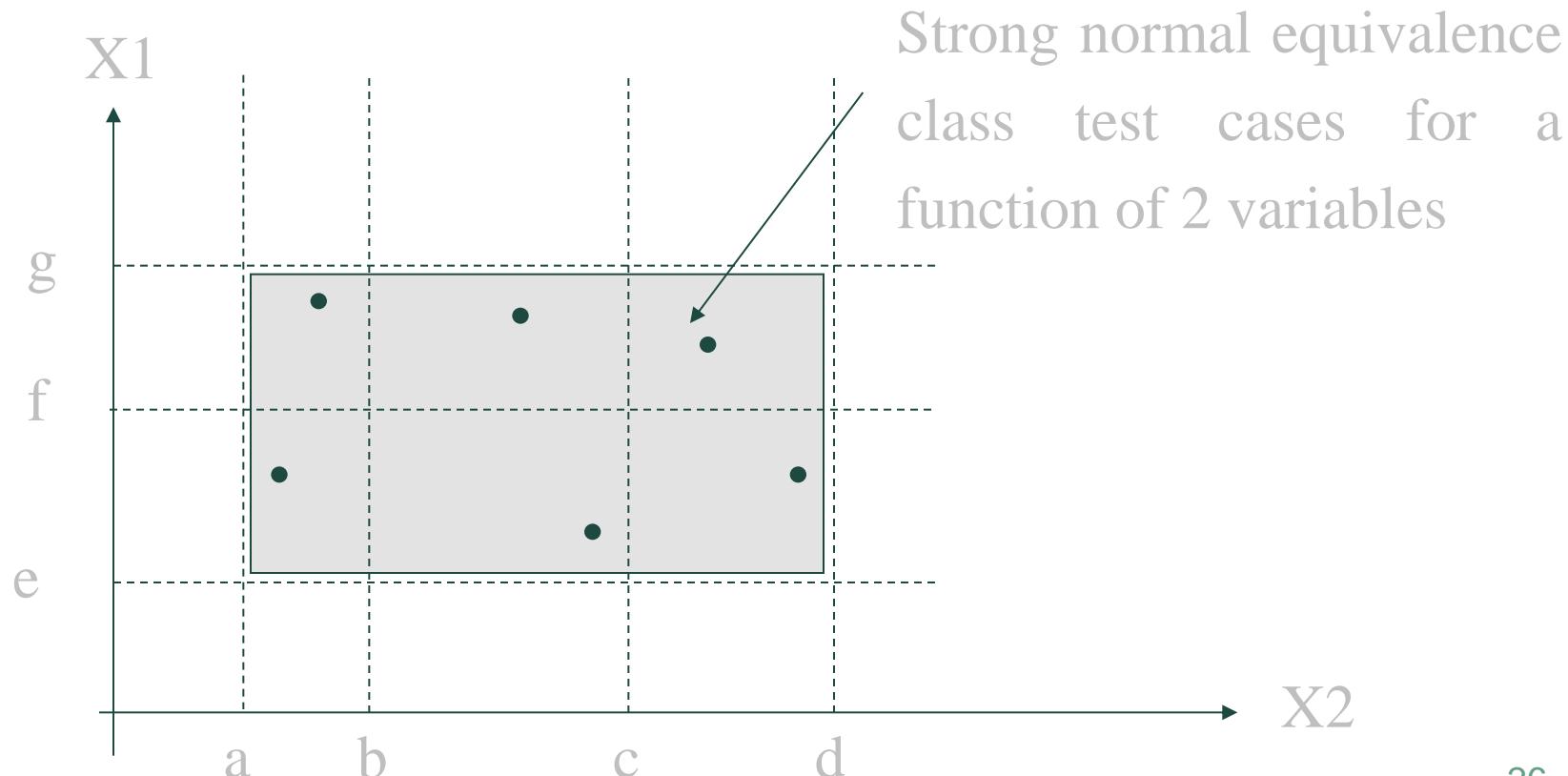
Weak Normal Equivalence Class Testing



Strong Normal Equivalence Class Testing

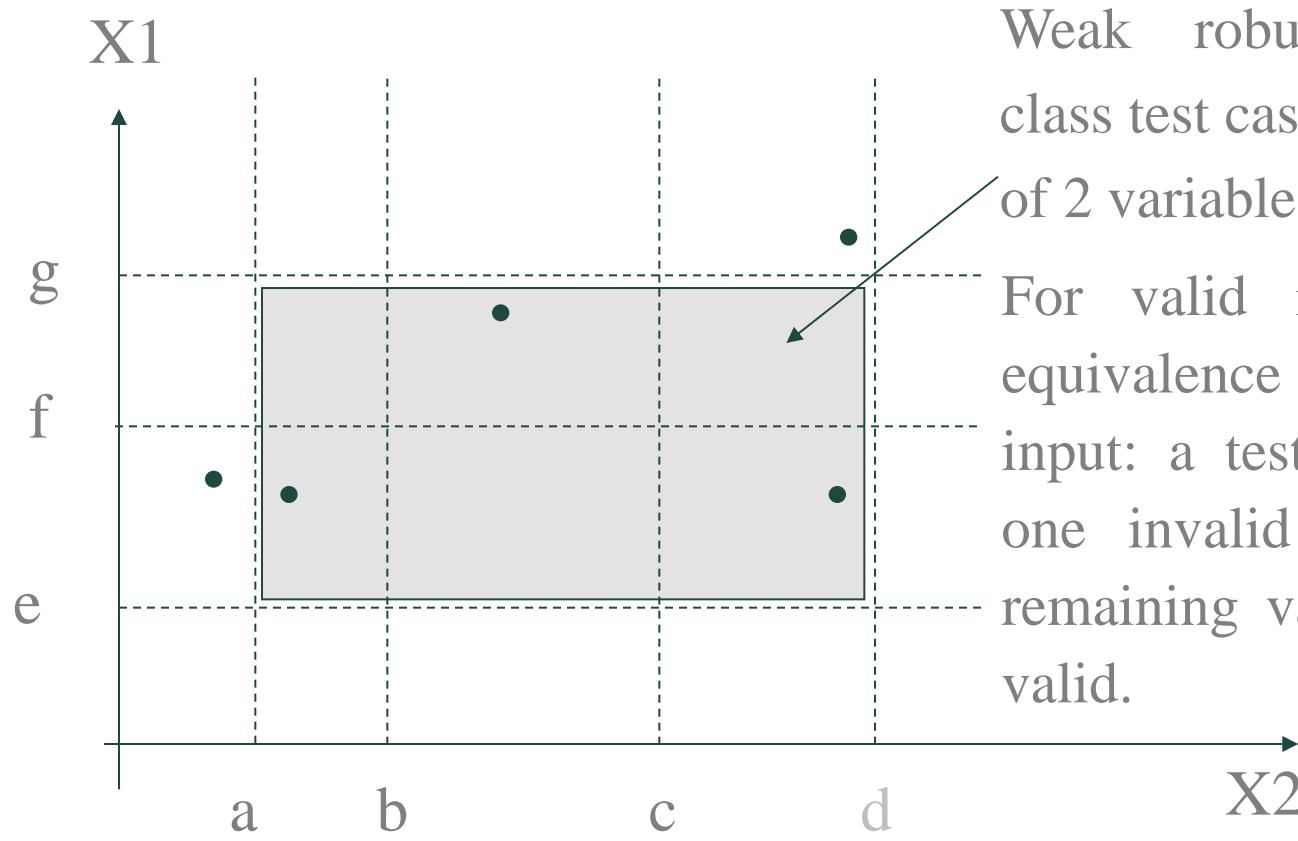
❖ Dựa trên Multiple Fault Assumption

- Một failure có thể là kết quả của 2 hay nhiều faults xảy ra cùng 1 lúc

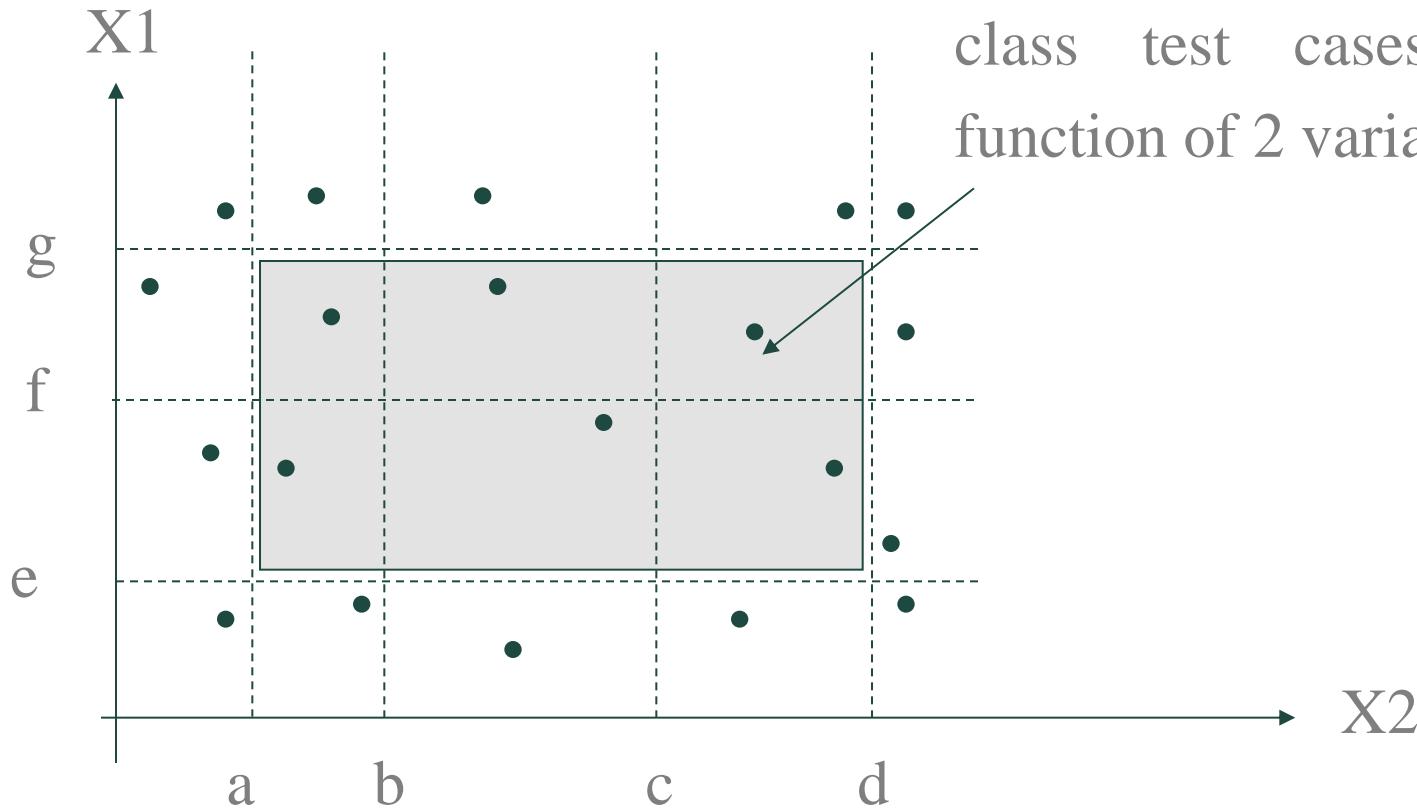


Weak Robust Equivalence Class Testing

- ❖ Tương tự Weak Equivalence Class Testing, tuy nhiên test thêm trường hợp 1 biến với giá trị không hợp lệ



Strong Robust Equivalence Class Testing



Strong robust equivalence
class test cases for a
function of 2 variables

Ví dụ 1

- ❖ TPPM “ Quản lý hồ sơ nhân lực” với đặc tả chức năng như sau: sau mỗi lần nhận 1 hồ sơ xin việc, TPPM sẽ ra quyết định ban đầu dựa và tuổi của ứng viên theo bảng sau:
 - **Tuổi ứng viên Kết quả sơ bộ**
 - 0-15 Không thuê
 - 16-17 Thuê dạng bán thời gian
 - 18-54 Thuê toàn thời gian
 - 55-99 Không thuê
 - Bằng phương pháp phân hoạch tương đương và phân tích giá trị biên, hãy thiết kế các trường hợp kiểm thử cho TPPM trên.

Ví dụ 1

❖ Bảng phân lớp tương đương

Tuổi	<0	0-15	16-17	18-54	55-99	>99
Kết quả	Nhập dữ liệu sai	Ko Thuê	Thuê bán thời gian	Thuê toàn thời gian	Ko Thuê	Nhập dữ liệu sai
Lớp tương đương	Ko Hợp lệ	Hợp lệ	Hợp lệ	Hợp lệ	Hợp lệ	Ko Hợp lệ
Đánh dấu	I1	V1	V2	V3	V4	I2

❖ Từ mỗi lớp tương đương, xét các biên cần kiểm thử

- Lớp I1: {-2,-1,0} Lớp I2: {99,100,101}
- Lớp V1: {-1,0,1} {14,15,16}
- Lớp V2: { 15,16,17} {16,17,18}
- Lớp V3: {17,18,19}, {53,54,55}
- Lớp V4: {54,55,56}, {98,99,100}

Ví dụ 1:

- ❖ Xét trong các trường hợp trên thấy có nhiều giá trị test case trùng nhau, nếu loại bỏ các test case trùng nhau đó thì ta còn: - 1, 0, 1, 14, 15, 16, 17, 18, 19, 53, 54, 55, 56, 98, 99, 100.
- ❖ Do trường hợp -1 và 100 đã nằm tại biên của hai lớp tương đương ko hợp lệ nên ta ko xét trường hợp -2, và 101 ở lớp này.
- ❖ 16 ca kiểm thử được thiết kế như sau:

Ví dụ 1: các ca kiểm thử

TC	Đầu vào	Đầu ra mong đợi
1	-1	Nhập sai dữ liệu
2	0	Không thuê
3	1	Không thuê
4	14	Không thuê
5	15	Không thuê
6	16	Thuê bán thời gian
7	17	Thuê bán thời gian
8	18	Thuê toàn thời gian

TC	Đầu vào	Đầu ra mong đợi
9	19	Thuê toàn thời gian
10	53	Thuê toàn thời gian
11	54	Thuê toàn thời gian
12	55	Không thuê
13	56	Không thuê
14	98	Không thuê
15	99	Không thuê
16	100	Nhập sai dữ liệu

Ví dụ 2: Ứng dụng vay nợ

Customer Name	<input type="text"/>	2-64 chars.
Account number	<input type="text"/>	6 digits, 1st non-zero
Loan amount requested	<input type="text"/>	£500 to £9000
Term of loan	<input type="text"/>	1 to 30 years
Monthly repayment	<input type="text"/>	Minimum £10
Term:		
Repayment:		
Interest rate:		
Total paid back:		

4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

- ✓ Phân tích giá trị biên (BVA) là kỹ thuật kiểm thử dựa **các giá trị tại biên** giữa các phân vùng tương đương, bao gồm trường hợp:
 - Hợp lệ (valid).
 - Không hợp lệ (invalid).

Kỹ thuật phân tích giá trị biên

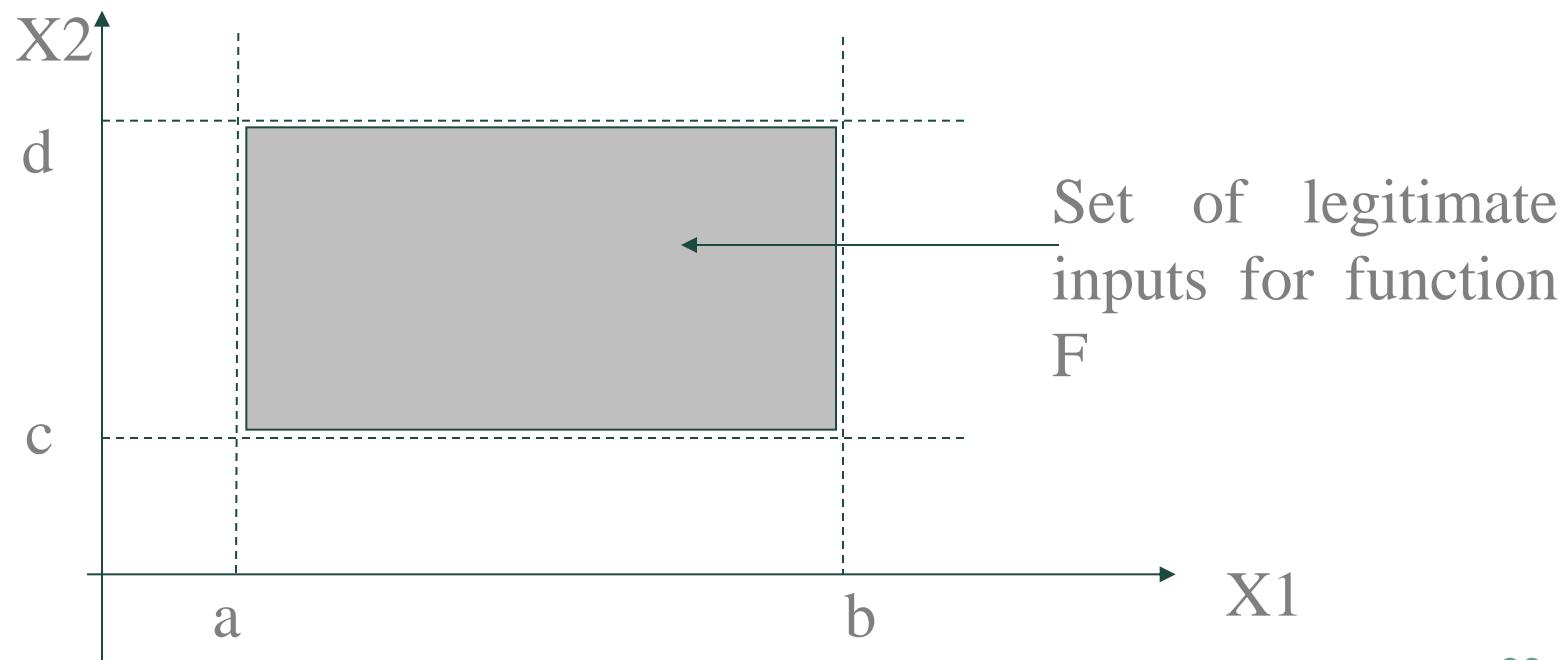
- ❖ Phân tích giá trị biên - Boundary Value Analysis
- ❖ Thường được áp dụng đối với các đối số của một phương thức
- ❖ Tập trung vào việc kiểm thử các giá trị biên của miền giá trị inputs để thiết kế test case do “lỗi thường tiềm ẩn lại các ngõ ngách và tập hợp tại biên” (Beizer)
- ❖ BVA hiệu quả nhất trong trường hợp “các đối số đầu vào (input variables) độc lập với nhau và mỗi đối số đều có một miền giá trị hữu hạn”

Kỹ thuật phân tích giá trị biên

❖ Giả sử hàm F có hai biến X_1, X_2 như sau:

- $a \leq X_1 \leq b$
- $c \leq X_2 \leq d$

❖ Input domain of a function of two variables:



Một số kỹ thuật kiểm thử giá trị biên

- ❖ Standard BVA (Boundary Value Analysis)
- ❖ Robustness testing
- ❖ Worst-case testing
- ❖ Robust worst-case testing

Standard BVA

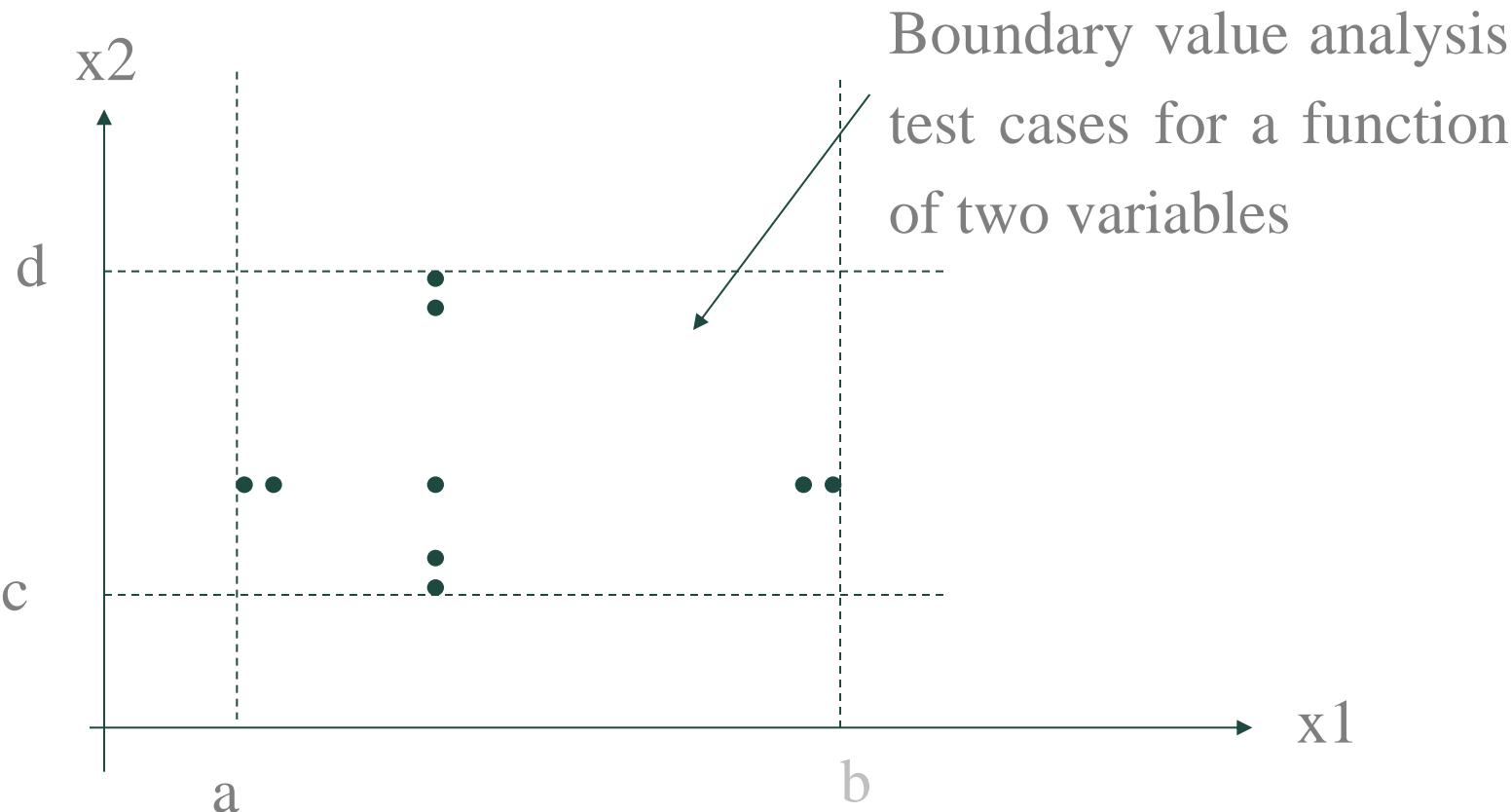
❖ Giả sử biến x có miền giá trị [min,max]

→ Các giá trị được chọn để kiểm tra

- Min - Minimal
- Min+ - Just above Minimal
- Nom - Average
- Max- - Just below Maximum
- Max - Maximum

Kỹ thuật phân tích trên giá trị biên

- ❖ Số test case là $4n+1$, với n là số lượng biến

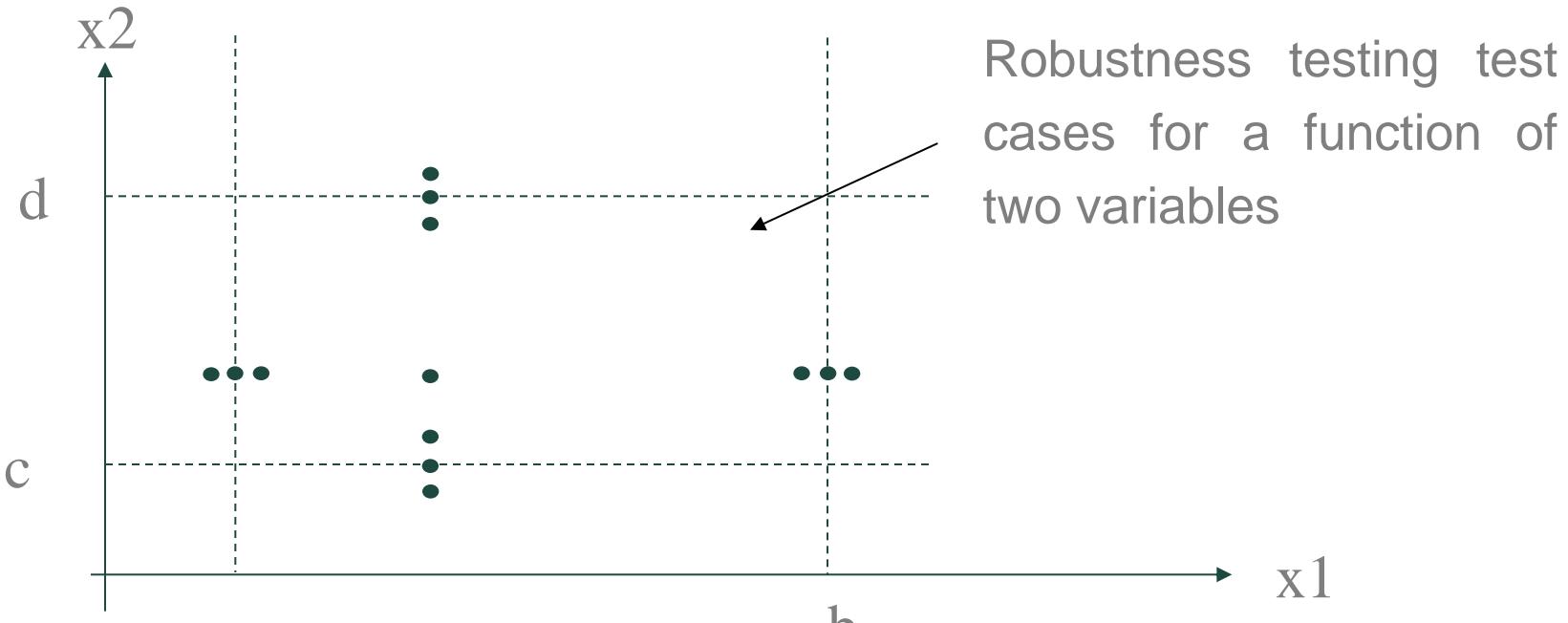


Robustness Testing

- ❖ Mở rộng của Standard BVA
- ❖ Kiểm thử cả hai trường hợp:
 - Input variable hợp lệ (clean test cases)
→ Kiểm thử tương tự như Standard BVA trên các giá trị (min, min+, average, max-, max)
 - Input variable không hợp lệ (dirty test cases)
→ Kiểm thử trên 2 giá trị: min-, max+ (nằm ngoài miền giá trị hợp lệ)

Robustness Testing

- ❖ Số lượng test case là $6n + 1$, với n là số lượng biến



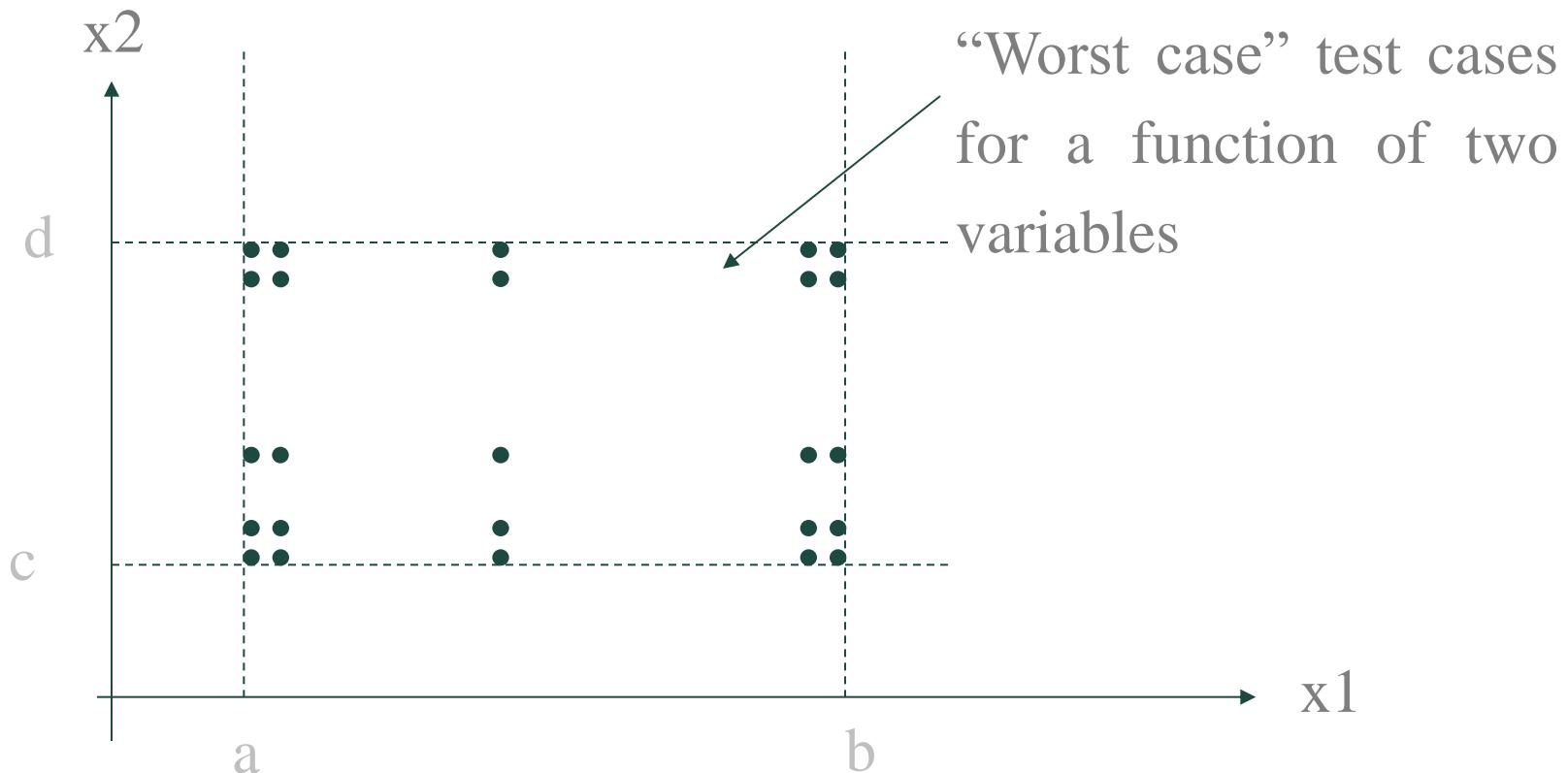
- ❖ Tập trung ^a vào việc kiểm thử trên các giá trị không hợp lệ và đòi hỏi ứng dụng phải xử lý ngoại lệ một cách đầy đủ

Worst-case testing

- ❖ Dựa trên **Multiple Fault Assumption** để thiết kế test case
- ❖ Các biến sẽ được kiểm tra đồng thời tại biên để dò lỗi
- ❖ Chúng ta không kiểm thử tại các giá trị không hợp lệ

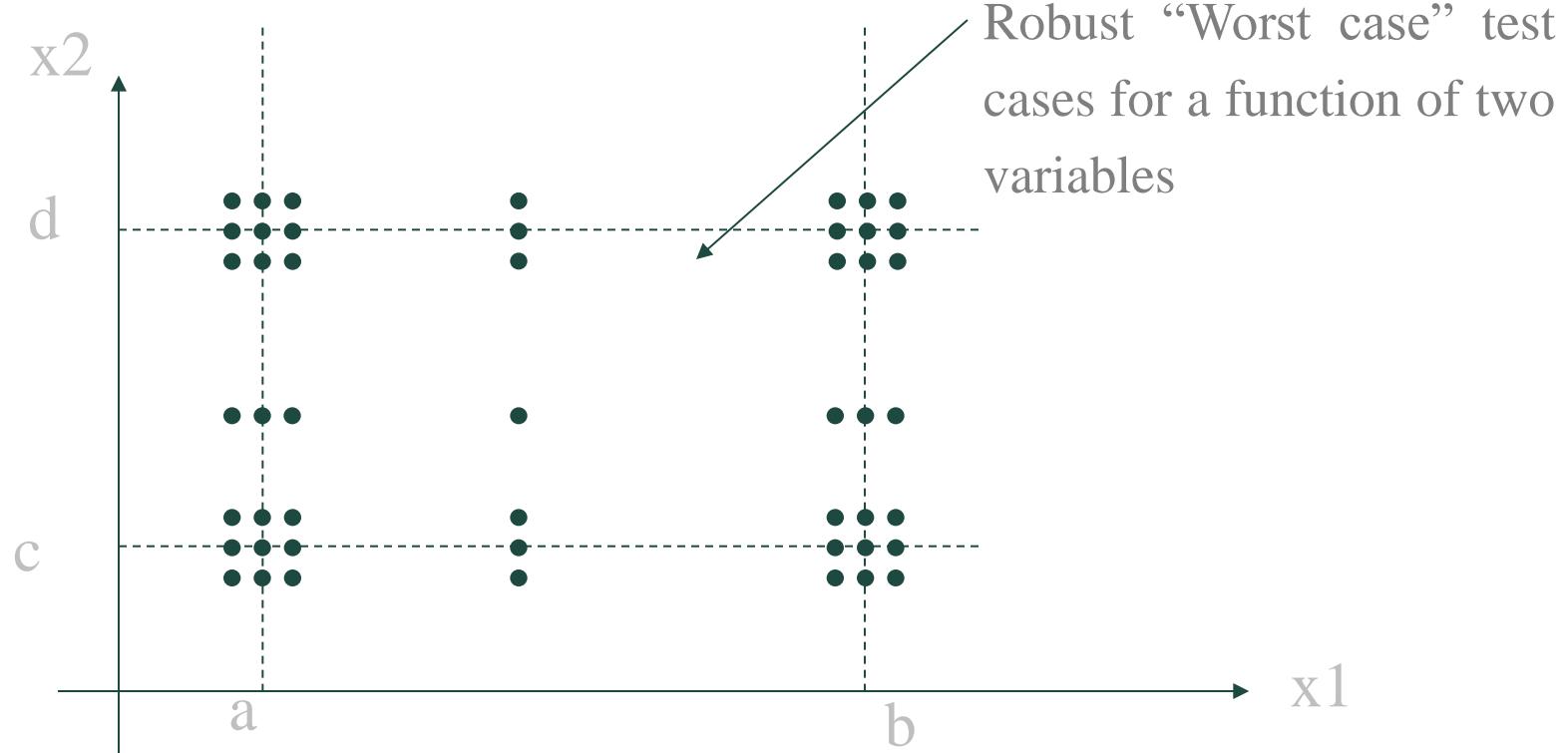
Worst-case testing

- ❖ Số lượng test case là 5^n , với n là số biến



Robust worst-case testing

- ❖ Tương tự Worst-case Testing nhưng kiểm tra thêm tại các giá trị không hợp lệ của input variables (min-, max+)
- ❖ Số lượng test case là 7^n , với n là số biến



Ví dụ hàm kiểm tra tam giác

- ❖ Ràng buộc: $1 \leq a, b, c \leq 200$.
- ❖ Áp dụng Standard BVA (số test case $4*3 + 1 = 13$)
 - min = 1
 - min+ = 2
 - nom = 100
 - max- = 199
 - max = 200

Boundary Value Analysis Test Cases				
Case	a	b	c	Expected Output
1	100	100	1	Isosceles
2	100	100	2	Isosceles
3	100	100	100	Equilateral
4	100	100	199	Isosceles
5	100	100	200	Not a Triangle
6	100	1	100	Isosceles
7	100	2	100	Isosceles
8	100	199	100	Isosceles
9	100	200	100	Not a Triangle
10	1	100	100	Isosceles
11	2	100	100	Isosceles
12	199	100	100	Isosceles
13	200	100	100	Not a Triangle

Ví dụ hàm kiểm tra tam giác

❖ Áp dụng
Worst-
case
testing

Worst Case Test Cases (60 of 125)				
Case	a	b	c	Expected Output
1	1	1	1	Equilateral
2	1	1	2	Not a Triangle
3	1	1	100	Not a Triangle
4	1	1	199	Not a Triangle
5	1	1	200	Not a Triangle
6	1	2	1	Not a Triangle
7	1	2	2	Isosceles
8	1	2	100	Not a Triangle
9	1	2	199	Not a Triangle
10	1	2	200	Not a Triangle
11	1	100	1	Not a Triangle
12	1	100	2	Not a Triangle
13	1	100	100	Isosceles
14	1	100	199	Not a Triangle
15	1	100	200	Not a Triangle
16	1	199	1	Not a Triangle
17	1	199	2	Not a Triangle
18	1	199	100	Not a Triangle
19	1	199	199	Isosceles
20	1	199	200	Not a Triangle
21	1	200	1	Not a Triangle
22	1	200	2	Not a Triangle
23	1	200	100	Not a Triangle
24	1	200	199	Not a Triangle
25	1	200	200	Isosceles
26	2	1	1	Not a Triangle
27	2	1	2	Isosceles
28	2	1	100	Not a Triangle
29	2	1	199	Not a Triangle
30	2	1	200	Not a Triangle
31	2	2	1	Isosceles
32	2	2	2	Equilateral
33	2	2	100	Not a Triangle
34	2	2	199	Not a Triangle
35	2	2	200	Not a Triangle
36	2	100	1	Not a Triangle
37	2	100	2	Not a Triangle
38	2	100	100	Isosceles
39	2	100	199	Not a Triangle
40	2	100	200	Not a Triangle
41	2	199	1	Not a Triangle
42	2	199	2	Not a Triangle
43	2	199	100	Not a Triangle
44	2	199	199	Isosceles
45	2	199	200	Scalene
46	2	200	1	Not a Triangle
47	2	200	2	Not a Triangle
48	2	200	100	Not a Triangle
49	2	200	199	Scalene
50	2	200	200	Isosceles
51	100	1	1	Not a Triangle
52	100	1	2	Not a Triangle
53	100	1	100	Isosceles
54	100	1	199	Not a Triangle
55	100	1	200	Not a Triangle
56	100	2	1	Not a Triangle
57	100	2	2	Not a Triangle
58	100	2	100	Isosceles
59	100	2	199	Not a Triangle
60	100	2	200	Not a Triangle

Ví dụ hàm tìm ngày kế tiếp

- ❖ Bài toán tìm ngày kế tiếp với các ràng buộc:
 - $1 \leq \text{Day} \leq 31$.
 - $1 \leq \text{month} \leq 12$.
 - $1812 \leq \text{Year} \leq 2012$
- ❖ Áp dụng Standard BVA (số test case $4^*3 + 1 = 13$)

Ví dụ hàm tìm ngày kế tiếp

<u>month</u>	<u>day</u>
min = 1	min = 1
min+ = 2	min+ = 2
nom = 6	nom = 15
max- = 11	max- = 30
max = 12	max = 31

<u>year</u>
min = 1812
min+ = 1813
nom = 1912
max- = 2011
max = 2012

Boundary Value Analysis Test Cases

Case	month	day	year	Expected Output
1	6	15	1812	June 16, 1812
2	6	15	1813	June 16, 1813
3	6	15	1912	June 16, 1912
4	6	15	2011	June 16, 2011
5	6	15	2012	June 16, 2012
6	6	1	1912	June 2, 1912
7	6	2	1912	June 3, 1912
8	6	30	1912	July 1, 1912
9	6	31	1912	error
10	1	15	1912	January 16, 1912
11	2	15	1912	February 16, 1912
12	11	15	1912	November 16, 1912
13	12	15	1912	December 16, 1912

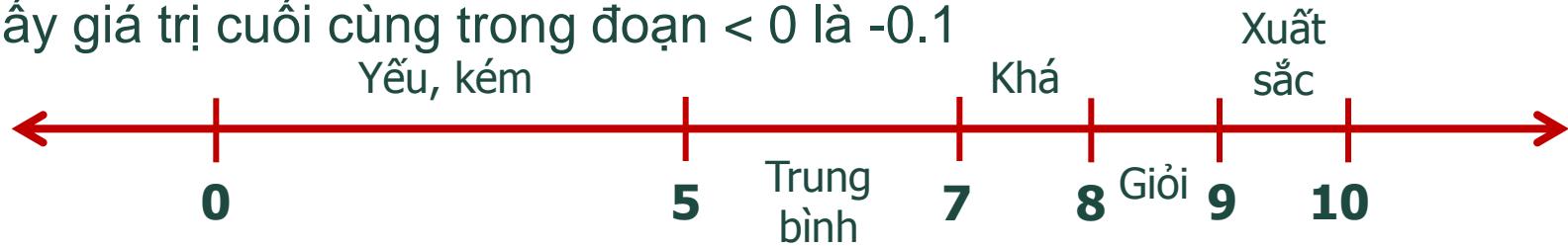
Ví dụ hàm tìm ngày kế tiếp

❖ Áp dụng Worst-case testing, Số lượng test case: 5^3

Worst Case Test Cases (60 of 125)				
Case	month	day	year	Expected Output
1	1	1	1812	January 2, 1812
2	1	1	1813	January 2, 1813
3	1	1	1912	January 2, 1912
4	1	1	2011	January 2, 2011
5	1	1	2012	January 2, 2012
6	1	2	1812	January 3, 1812
7	1	2	1813	January 3, 1813
8	1	2	1912	January 3, 1912
9	1	2	2011	January 3, 2011
10	1	2	2012	January 3, 2012
11	1	15	1812	January 16, 1812
12	1	15	1813	January 16, 1813
13	1	15	1912	January 16, 1912
14	1	15	2011	January 16, 2011
15	1	15	2012	January 16, 2012
16	1	30	1812	January 31, 1812
17	1	30	1813	January 31, 1813
18	1	30	1912	January 31, 1912
19	1	30	2011	January 31, 2011
20	1	30	2012	January 31, 2012
21	1	31	1812	February 1, 1812
22	1	31	1813	February 1, 1813
23	1	31	1912	February 1, 1912
24	1	31	2011	February 1, 2011
25	1	31	2012	February 1, 2012
26	2	1	1812	February 2, 1812
27	2	1	1813	February 2, 1813
28	2	1	1912	February 2, 1912
29	2	1	2011	February 2, 2011
30	2	1	2012	February 2, 2012
31	2	2	1812	February 3, 1812
32	2	2	1813	February 3, 1813
33	2	2	1912	February 3, 1912
34	2	2	2011	February 3, 2011
35	2	2	2012	February 3, 2012
36	2	15	1812	February 16, 1812
37	2	15	1813	February 16, 1813
38	2	15	1912	February 16, 1912
39	2	15	2011	February 16, 2011
40	2	15	2012	February 16, 2012
41	2	30	1812	error
42	2	30	1813	error
43	2	30	1912	error
44	2	30	2011	error
45	2	30	2012	error
46	2	31	1812	error
47	2	31	1813	error
48	2	31	1912	error
49	2	31	2011	error
50	2	31	2012	error
51	6	1	1812	June 2, 1812
52	6	1	1813	June 2, 1813
53	6	1	1912	June 2, 1912
54	6	1	2011	June 2, 2011
55	6	1	2012	June 2, 2012
56	6	2	1812	June 3, 1812
57	6	2	1813	June 3, 1813
58	6	2	1912	June 3, 1912
59	6	2	2011	June 3, 2011
60	6	2	2012	June 3, 2012

4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

- ❖ Ví dụ 1: xếp loại học sinh
- ❖ Để áp dụng phân tích giá trị biên vào kiểm thử chương trình lấy giá trị lớn nhất và nhỏ nhất trong các đoạn
 - Đoạn “yếu, kém”: 0 và 4.9
 - Đoạn “trung bình”: 5 và 6.9
 - Đoạn “khá”: 7 và 7.9
 - Đoạn “giỏi”: 8 và 8.9
 - Đoạn “xuất sắc”: 9 và 10
- ❖ Lấy giá trị đầu tiên trong đoạn > 10 là 10.1
- ❖ Lấy giá trị cuối cùng trong đoạn < 0 là -0.1



4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

Ví dụ 2

- ❖ Một developer đã phát triển xong chức năng **kiểm tra số nguyên dương n có phải là số nguyên tố không** và giao cho một tester tiến hành kiểm thử chức năng này với màn hình console cho phép nhập số nguyên dương n và xuất kết quả thông báo số đó có phải nguyên tố hay không?
- ❖ *Ghi chú: số nguyên tố là số tự nhiên chỉ có hai ước số dương phân biệt là 1 và chính nó.*

4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

Ví dụ 2

- Một **tester ít kinh nghiệm** kiểm thử bằng cách chạy một số giá trị nguyên n để kiểm tra chương trình trên.

```
C:\WINDOWS\system32\cmd.exe
n = 2
NGUYEN TO
n = 7
NGUYEN TO
n = 13
NGUYEN TO
n = 113
NGUYEN TO
n = 71
NGUYEN TO
```

```
C:\WINDOWS\system32\cmd.exe
n = 12
KHONG NGUYEN TO
n = 143
KHONG NGUYEN TO
n = 112
KHONG NGUYEN TO
n = 58
KHONG NGUYEN TO
n = 98
KHONG NGUYEN TO
```

```
C:\WINDOWS\system32\cmd.exe
n = 0
KHONG NGUYEN TO
n = 1
KHONG NGUYEN TO
n = -7
KHONG NGUYEN TO
n = -6
KHONG NGUYEN TO
n = -53
KHONG NGUYEN TO
```



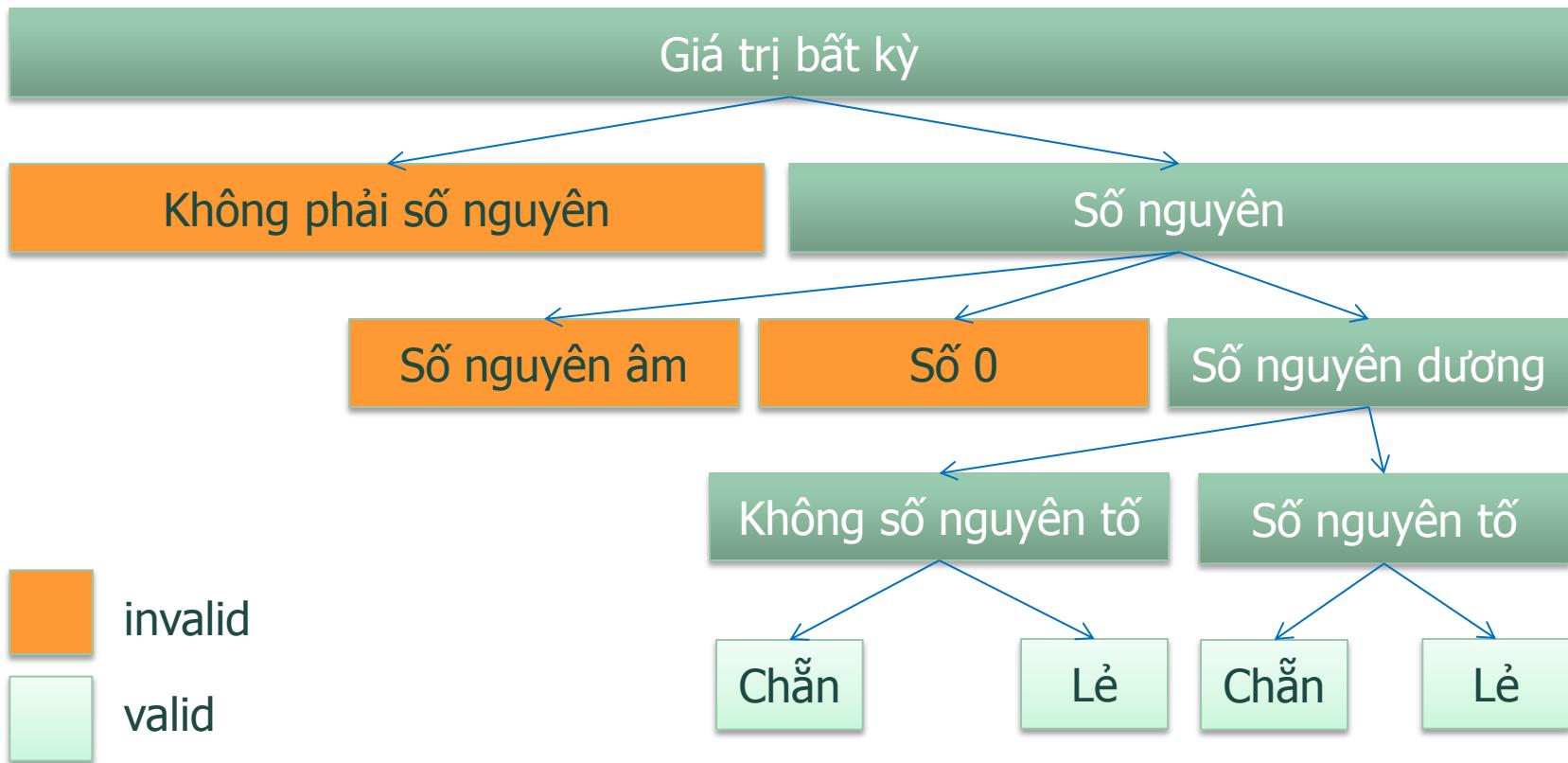
Chương
trình chắc
ổn?



4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

Ví dụ 2

Một tester có kinh nghiệm tiến hành phân vùng tương đương để thiết kế test case kiểm thử như sau:



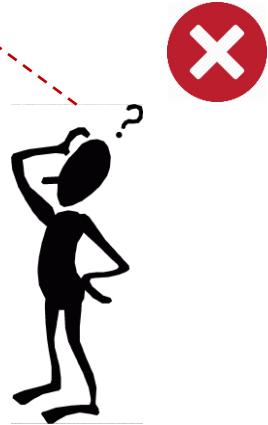
4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

Ví dụ 2

- ❖ Áp dụng phân tích giá trị biên cho phân vùng **số nguyên dương**, chọn các giá trị biên sau đại diện cho các phân vùng để kiểm thử:
 - Phân vùng số nguyên tố chẵn: **2**
 - Phân vùng số nguyên tố lẻ: **3**
 - Phân vùng số lẻ không phải số nguyên tố: **1**
 - Phân vùng số chẵn không là số nguyên tố: **4**

C:\WINDOWS\system32

```
n = 2  
NGUYEN TO  
n = 3  
NGUYEN TO  
n = 1  
KHONG NGUYEN TO  
n = 4  
NGUYEN TO
```



4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

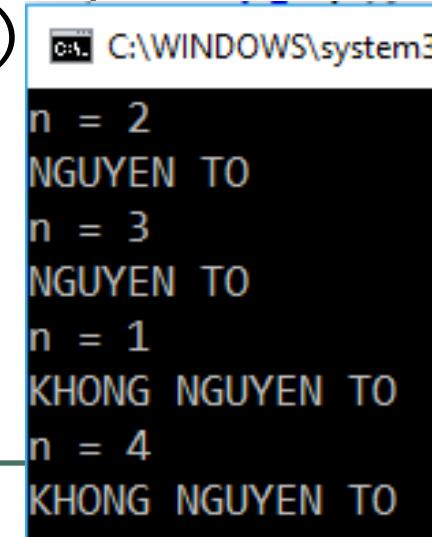
Ví dụ 2

- ❖ Developer mở chương trình kiểm tra

```
bool ktNguyenTo(int n) {
    if (n < 2)
        return false;

    for (int i = 2; i <=sqrt(n); i++)
        if (n % i == 0)
            return false;

    return true;
}
```



The terminal window shows the following output:

```
C:\WINDOWS\system32>n = 2
NGUYEN TO
n = 3
NGUYEN TO
n = 1
KHONG NGUYEN TO
n = 4
KHONG NGUYEN TO
```

4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

Ưu điểm và khuyết điểm

- ❖ **Ưu điểm**
 - Đơn giản.
 - Hiệu quả cho các hàm có biến độc lập.
 - Có thể tự động sinh test case khi xác định được giá trị biên của các biến.
- ❖ **Khuyết điểm**
 - Không quan tâm đặc trưng của hàm, ngữ nghĩa các biến, cũng như quan hệ giữa các biến.
 - Khó áp dụng cho trường hợp các biến có quan hệ ràng buộc nhau.

4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

Bài tập

- ✓ Một hệ thống ngân hàng trực tuyến của ngân hàng ABC quy định không được chuyển khoản quá 10 triệu (tr) trong ngày, tối thiểu mỗi lần chuyển khoản là 1tr.
- ✓ *Sử dụng phương pháp phân tích giá trị biên thiết kế test case để kiểm tra số tiền chuyển khoản của khách hàng A có được phép chuyển trong ngày hiện tại không?*

4.3.3. BẢNG QUYẾT ĐỊNH

- ✓ Bảng quyết định (DT) là kỹ thuật kiểm thử hộp đen dùng xác định **những kịch bản** (scenario) kiểm thử cho những trường hợp có logic nghiệp vụ phức tạp.
- ✓ DT giúp tester xác định hiệu quả sự **kết hợp các đầu vào khác nhau** với các tình trạng phần mềm thực thi đúng quy tắc nghiệp vụ.
- ✓ DT nên được sử dụng trong những chương trình có **nhiều lệnh rẽ nhánh và các biến đầu vào có mối quan hệ với nhau**.

4.3.3. BẢNG QUYẾT ĐỊNH

Cấu trúc bảng quyết định

Các điện kiện: một biến, một quan hệ hoặc một mệnh đề.

Các giá trị điều kiện: một giá trị nào đó của điều kiện (các cột quy tắc - rule).

Các điều kiện	Các giá trị điều kiện
Hành động	Xảy ra hay không

Hành động: một thủ tục hoặc thao tác cần thực hiện.

Các giá trị của hành động: một hành động có thể xảy ra phụ thuộc vào tổ hợp các giá trị điều kiện.

4.3.3. BẢNG QUYẾT ĐỊNH

Xây dựng bảng quyết định

- ❖ **Liệt kê tất cả các điều kiện/đầu vào.**
- ❖ **Tính số lượng kết hợp** các giá trị của các điều kiện/đầu vào và đặt các kết hợp đó vào trong phần giá trị các điều kiện.
- ❖ **Xác định các test case** tương ứng cho các điều kiện được thỏa mãn.
- ❖ **Các hành động** chính là kết quả mong đợi của test case.

4.3.3. BẢNG QUYẾT ĐỊNH

Xây dựng bảng quyết định

- ✓ Trong giá trị các điều kiện có một giá trị đặc biệt là “–” thể hiện cho các điều kiện kết hợp chỉ định không thể xảy ra.
- ✓ Chú ý: thứ tự các điều kiện và thứ tự thực hiện hành động là không quan trọng.

Bảng quyết định

- ❖ Liệt kê các **nguyên nhân (cause)** – **kết quả (effect)** trong 1 ma trận. Mỗi cột trong ma trận đại diện cho 1 phép kết hợp giữa các cause trong việc tạo ra 1 effect

Causes	Values	Combinations							
		1	2	3	4	5	6	7	8
Cause 1	Y, N	Y	Y	Y	Y	N	N	N	N
Cause 2	Y, N	Y	Y	N	N	Y	Y	N	N
Cause 3	Y, N	Y	N	Y	N	Y	N	Y	N
Effects									
Effect 1		X		X				X	
Effect 2			X			X		X	

Cause = Condition

Effect = Actions = Expected Results

Các bước để tạo ra Bảng quyết định

- ❖ Liệt kê tất cả các nguyên nhân (causes) trong bảng quyết định
- ❖ Tính tổng số lượng kết hợp giữa các cause
- ❖ Điền vào các cột với tất cả các kết hợp có thể có
- ❖ Rút bớt số lượng các phép kết hợp dư thừa
- ❖ Kiểm tra các phép kết hợp có bao phủ hết mọi trường hợp hay không
- ❖ Bổ sung kết quả (effects) vào bảng quyết định

B1: Liệt kê tất cả các nguyên nhân

Causes	Values	Combinations							
		1	2	3	4	5	6	7	8
Cause 1	Y, N	Y	Y	Y	Y	N	N	N	N
Cause 2	Y, N	Y	Y	N	N	Y	Y	N	N
Cause 3	Y, N	Y	N	Y	N	Y	N	Y	N
Effects									
Effect 1		X		X				X	
Effect 2			X			X		X	

- c1: $a < b + c ?$
- c2: $b < a + c ?$
- c3: $c < a + b ?$
- c4: $a = b ?$
- c5: $a = c ?$
- c6: $b = c ?$

- ❖ Điền vào các giá trị trong từng causes
 - ❖ Gom nhóm các causes có liên quan với nhau
 - ❖ Sắp xếp các cause theo thứ tự giảm dần theo độ ưu tiên
- Ví dụ: xét bài toán kiểm tra loại của 1 tam giác dựa vào chiều dài 3 cạnh a, b, c.

B2: Tính tổng số kết hợp giữa các causes

❖ Tổng số phép kết hợp

= (số lượng values của cause 1) * ... * (số lượng values của cause n)

c1: a < b+c?
c2: b < a+c?
c3: c < a+b?
c4: a = b?
c5: a = c?
c6: b = c?

Mỗi cause có 2 giá trị true, false
→ Tổng số phép kết hợp = $2^6 = 64$

B3: Điền giá trị các cột trong bảng

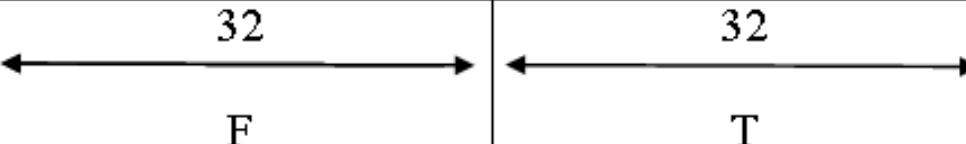
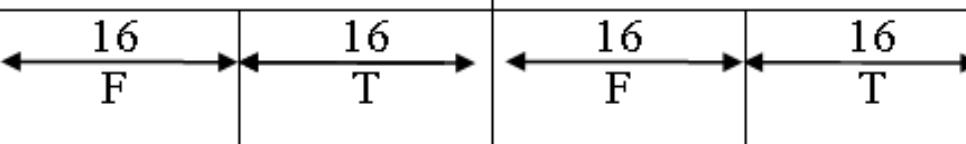
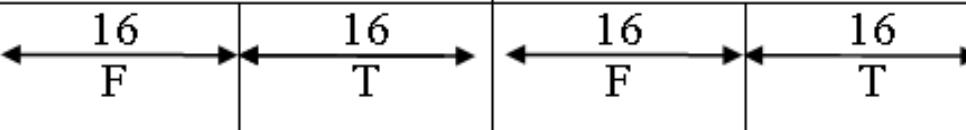
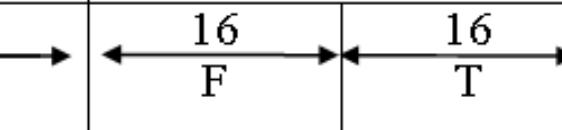
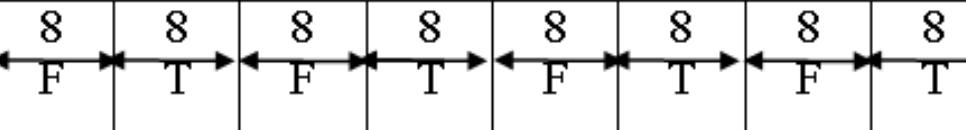
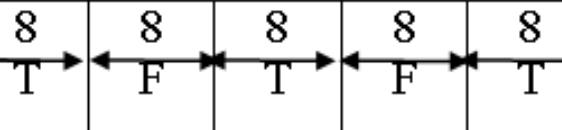
Causes	Values	Combinations							
		1	2	3	4	5	6	7	8
Cause 1	Y, N	Y	Y	Y	Y	N	N	N	N
Cause 2	Y, N	Y	Y	N	N	Y	Y	N	N
Cause 3	Y, N	Y	N	Y	N	Y	N	Y	N
Effects									
Effect 1		X		X				X	
Effect 2			X			X		X	X

❖ Thuật toán:

- Xác định số lần lặp lại (RF) trong từng giá trị của cause bằng cách lấy tổng số phép kết hợp **còn lại** chia cho số values mà cause có thể nhận
- Điền dữ liệu cho dòng thứ i: điền RF lần **giá trị đầu tiên** của cause i, tiếp theo RF lần **giá trị tiếp theo** của cause i... cho đến khi dòng đầy
- Chuyển sang dòng kế tiếp, quay lại bước 1 và tiếp tục thực hiện

B3: Điền giá trị các cột trong bảng

❖ Ví dụ:

c1: $a < b+c$?		
c2: $b < a+c$?		
c3: $c < a+b$?		
c4: $a = b$?	

$$RF = 64 / 2 = 32$$

$$RF = 32 / 2 = 16$$

$$RF = 16 / 2 = 8$$

B4: Giảm số phép kết hợp

- ❖ Duyệt qua tất cả các ô trong từng cột, ô nào mà kết quả của nó không ảnh hưởng đến effect thì đặt giá trị trên ô này là “-” (don’t care entry)
- ❖ Ghép các cột với nội dung giống nhau thành 1 cột

F	T	T	T	T	T	T	T	T	T	T	T	T
-	F	T	T	T	T	T	T	T	T	T	T	T
-	-	F	T	T	T	T	T	T	T	T	T	T
-	-	-	T	T	T	T	F	F	F	F	F	F
-	-	-	T	T	F	F	T	T	F	F	F	F
-	-	-	T	F	T	F	T	F	T	F	T	F

B5: Kiểm tra độ bao phủ các phép kết hợp

- ❖ Tính rule-count trên từng cột (số lượng phép kết hợp mà cột này có thể thực hiện)
- ❖ Với các dòng có giá trị là ‘-’ thì luỹ thừa 2
- ❖ Nếu tổng của các rule-count bằng với tổng số kết hợp giữa các cause trong bước 2 thì bảng quyết định là đầy đủ

c1: a < b + c?	F	T	T	T	T	T	T	T	T	T	T
c2: b < a + c?	-	F	T	T	T	T	T	T	T	T	T
c3: c < a + b?	-	-	F	T	T	T	T	T	T	T	T
c4: a = b?	-	-	-	T	T	T	T	F	F	F	F
c5: a = c?	-	-	-	T	T	F	F	T	T	F	F
c6: b = c?	-	-	-	T	F	T	F	T	F	T	F
Rule Count	32	16	8	1	1	1	1	1	1	1	1

B6: Bổ sung kết quả (effect) vào trong bảng

KIỂM THỬ PHẦN MỀM

Duyệt qua từng cột và check vào kết quả (effect)

Nhiều cột khác nhau có thể cho ra cùng 1 kết quả giống nhau

c1: $a < b+c?$	F	T	T	T	T	T	T	T	T	T	T	T
c2: $b < a+c?$	-	F	T	T	T	T	T	T	T	T	T	T
c3: $c < a+b?$	-	-	F	T	T	T	T	T	T	T	T	T
c4: $a = b?$	-	-	-	T	T	T	T	F	F	F	F	F
c5: $a = c?$	-	-	-	T	T	F	F	T	T	F	F	F
c6: $b = c?$	-	-	-	T	F	T	F	T	F	T	F	F
Rule Count	32	16	8	1	1	1	1	1	1	1	1	1
a1: Not a Triangle	X	X	X									
a2: Scalene												X
a3: Isosceles							X		X	X		
a4: Equilateral				X								
a5: Impossible					X	X		X				

Ví dụ

❖ Bảng quyết định hoàn chỉnh

c1: $a < b+c?$	F	T	T	T	T	T	T	T	T	T	T	T
c2: $b < a+c?$	-	F	T	T	T	T	T	T	T	T	T	T
c3: $c < a+b?$	-	-	F	T	T	T	T	T	T	T	T	T
c4: $a = b?$	-	-	-	T	T	T	T	F	F	F	F	F
c5: $a = c?$	-	-	-	T	T	F	F	T	T	F	F	F
c6: $b = c?$	-	-	-	T	F	T	F	T	F	T	F	F
Rule Count	32	16	8	1	1	1	1	1	1	1	1	1
a1: Not a Triangle	X	X	X									
a2: Scalene												X
a3: Isosceles								X		X	X	
a4: Equilateral				X								
a5: Impossible					X	X		X				

4.3.3. BẢNG QUYẾT ĐỊNH

Xây dựng bảng quyết định

- ❖ Ví dụ: xây dựng bảng quyết định cho chức năng login với hai thông tin đầu vào là username và password trên một ứng dụng web.
 - Xác định các điều kiện/đầu vào: username và password.
 - Mỗi đầu vào nhận một trong 3 giá trị: rỗng (blank - B), hợp lệ (valid - V) và không hợp lệ (invalid - I).
 - Số kết hợp giá trị các điều kiện có thể xảy ra là 9

4.3.3. BẢNG QUYẾT ĐỊNH

Xây dựng bảng quyết định

Các điều kiện	Username	B	B	B	I	I	I	V	V	V
	Password	B	I	V	B	I	V	B	I	V
Các hành động	Thông điệp lỗi	M1	M1	M1	M3	M3	M3	M2	M4	
	Chuyển đến trang	L	L	L	L	L	L	L	L	H

M1: Vui lòng nhập username

M2: Vui lòng nhập password

M3: Username không hợp lệ

M4: Password không hợp lệ

L: Trang login

H: Trang home

4.3.3. BẢNG QUYẾT ĐỊNH

Rút gọn bảng quyết định

- ❖ Rút gọn các kết hợp các giá trị đầu vào.
 - Những trường hợp (cột quy tắc) có **cùng giá trị hành động**, nhưng chỉ **khác giá trị của một điều kiện** duy nhất.
 - Chuyển giá trị của điều kiện khác nhau đó thành “-” và gôm các cột lại thành một.
 - Lặp lại hai bước trên cho đến khi không còn các test case nào như thế.

4.3.3. BẢNG QUYẾT ĐỊNH

Rút gọn bảng quyết định

Các điều kiện	Username	B	B	B	I	I	I	V	V	V
	Password	B	I	V	B	I	V	B	I	V
Các hành động	Thông điệp lỗi	M1	M1	M1	M3	M3	M3	M2	M4	
	Chuyển đến trang	L	L	L	L	L	L	L	L	H



Các điều kiện	Username	B	I	V	V	V
	Password	-	-	B	I	V
Các hành động	Thông điệp lỗi	M1	M3	M2	M4	
	Chuyển đến trang	L	L	L	L	H

4.3.3. BẢNG QUYẾT ĐỊNH

Chuyển thành các test case

- ❖ Các quy tắc chuyển từ bảng quyết định thành các test case:
 - Nếu giá trị các điều kiện nhập là các giá trị **luận lý** (true/false) thì **mỗi cột** quy tắc được chuyển thành một test case.
 - Nếu giá trị các điều kiện nhập có **nhiều giá trị** thì mỗi cột quy tắc được chuyển thành nhiều test case sử dụng kỹ thuật phân vùng tương đương hoặc phân tích giá trị biên.

Ví dụ 1

- ❖ Công ty Honda trao học bổng cho những bạn sinh viên thỏa mãn ít nhất 1 trong 2 điều kiện sau: Là sinh viên giỏi , là cán bộ lớp.
- ❖ Nếu thỏa mãn cả 2 điều kiện sẽ được học bổng 600\$, nếu thỏa mãn là sinh viên giỏi được học bổng 400\$, nếu là cán bộ lớp được học bổng là 300\$.
- ❖ Lập bảng hỗ trợ quyết định để thiết kế các ca kiểm thử

VD1: Bảng hỗ trợ quyết định

		Luật 1	Luật 2	Luật 3	Luật 4
Điều kiện	Là cán bộ lớp	Y	Y	N	N
	Là học sinh giỏi	Y	N	Y	N
Hành động	Học bổng	600\$	300\$	400\$	0\$

Các ca kiểm thử

Ca	Đầu vào	Đầu ra mong đợi
1	Là cán bộ lớp, là sv giỏi	600\$
2	Là cán bộ lớp, ko fai là sv giỏi	300\$
3	Ko phải cán bộ lớp, là sv giỏi	400\$
4	Ko phải cán bộ lớp, ko phải sv giỏi	0\$

VD 2

- ❖ Chương trình quản lý tiền vé và số lượng mū phát cho khách hàng vào thăm quan bảo tàng được mô tả như sau:
- ❖ Vé bán có các mức sau: đối với trẻ em dưới 5 tuổi được miễn phí, đối tượng từ 5 tuổi tới 65 tuổi phải trả 20\$/vé, đối tượng lớn hơn 65 tuổi phải trả 10\$/vé.
- ❖ Chương trình tặng mū cho khách: với những đối tượng là nữ sẽ tặng mū hồng, đối tượng là nam sẽ tặng mū xanh
- ❖ Dùng bảng hỗ trợ quyết định xây dựng các ca kiểm thử cho chương trình trên

VD2

KIỂM THỬ PHẦN MỀM PHẦN MỀM		Luật 1	Luật 2	Luật 3	Luật 4	Luật 5	Luật 6
Điều kiện	Tuổi <5	Y	Y				
	5<=tuổi<=65			Y	Y		
	Tuổi >65					Y	Y
	Giới tính	Y	N	Y	N	Y	N
Hành động	Miễn phí	Y	Y				
	Giá 10 \$					Y	Y
	Giá 20\$			Y	Y		
	Tặng mũ xanh	Y		Y		Y	
	Tặng mũ hồng		Y		Y		Y

VD2

❖ Các ca kiểm thử dựa trên bảng hỗ trợ quyết

Ca	Đầu vào	Đầu ra mong đợi
1	Tuổi <5, giới tính: Nữ	Miễn phí vé vào, tặng mũ đỏ
2	Tuổi <5, giới tính: Nam	Miễn phí vé vào, tặng mũ xanh
3	5<= tuổi<= 65, giới tính: Nữ	Giá vé 20\$, tặng mũ đỏ
4	5<= tuổi<= 65, giới tính: Nam	Giá vé 20\$, tặng mũ xanh
5	Tuổi >65, giới tính: Nữ	Giá vé 10\$, tặng mũ đỏ
6	Tuổi >65, giới tính: Nam	Giá vé 10\$, tặng mũ xanh

4.3.3. BẢNG QUYẾT ĐỊNH

Bài tập 1

- ❖ Cho chương trình xác định một tam giác có phải là tam giác cân không, biết người dùng nhập vào chiều dài 3 cạnh a, b, c của tam giác.
- ❖ Sử dụng bảng quyết định thiết kế test case để kiểm thử chương trình trên.

Điều kiện	$a + b > c$	T	T	T	T	F	-	-
	$b + c > a$	T	T	T	T	-	F	-
	$a + c > b$	T	T	T	T	-	-	F
	$a = b$	T	-	-	F	-	-	-
	$b = c$	-	T	-	F	-	-	-
	$c = a$	-	-	T	F	-	-	-
Hành động	Tam giác cân?	T	T	T	F	F	F	F

4.3.3. BẢNG QUYẾT ĐỊNH

Bài tập 2

- ❖ Một thư viện ABC có chức năng cho phép độc giả mượn sách. Theo đó, độc giả mượn sách không được quá 500 quyển sách trong năm (không phân biệt tên đầu sách), nhưng không được phép mượn quá 5 quyển trong một lần mượn, và phải trả các cuốn sách đã mượn mới được phép mượn tiếp nữa.
- ❖ Sử dụng bảng quyết định và phân tích giá trị biên thiết kế test case kiểm thử độc giả có được phép mượn sách không và được mượn tối đa bao nhiêu quyển trong lần mượn mới?

4.3.3. BẢNG QUYẾT ĐỊNH

Hướng dẫn

- ❖ K = không còn nợ sách đã mượn.
- ❖ M là số sách đã mượn trong năm đến thời điểm hiện tại ($0 \leq M \leq 500$). X là số sách định mượn ($1 \leq X \leq 5$). Độc giả sẽ được mượn số sách này nếu $0 \leq M + X \leq 500$ và K đúng.
- ❖ Bảng quyết định

		C ₁	C ₂	C ₃	C ₄
Điều kiện	K	✗	✓	✓	✓
	$1 \leq X \leq 5$	-	✗	✓	✓
	$0 \leq M + X \leq 500$	-	-	✗	✓
Hành động	Mượn sách?	✗	✗	✗	✓
	Số sách tối đa được mượn.	0	0	0	$500 - M < 5 ? 500 - M : 5$

4.3.3. BẢNG QUYẾT ĐỊNH

Hướng dẫn

Cột	Đầu vào			Mượn sách? Số sách tối đa được mượn.	
	K	X	M		
C_1	x	1	499	x	0
C_2	✓	6	494	x	0
	✓	6	495	x	0
C_3	✓	1	500	x	0
	✓	5	496	x	0
C_4	✓	1	499	✓	1
	✓	1	498	✓	2
	✓	5	495	✓	5
	✓	5	494	✓	5
	✓	2	498	✓	2
	✓	2	497	✓	3
	✓	4	496	✓	4
	✓	4	495	✓	5

4.3.3. BẢNG QUYẾT ĐỊNH

Bài tập 3

Một chương trình khuyến mãi tri ân khách hàng của hãng A cho những khách hàng mua dòng điện thoại cao cấp của hãng diễn ra từ ngày 20/11/2017 đến hết ngày 31/12/2017 – dòng điện thoại được gọi là cao cấp nếu giá bán lớn hơn hoặc bằng 20tr. Theo đó, nếu khách hàng mua điện thoại cao cấp của hãng A trong khoảng thời gian đó sẽ được tặng 1 loa bluetooth và miếng dán màn hình. Ngoài ra đối với những khách hàng đã từng dùng dòng điện thoại cao cấp của hãng A, tính từ thời điểm đã mua cho đến thời điểm mua mới, nếu khoảng thời gian này

- Không quá 1 năm thì khách hàng sẽ được giảm thêm 2 triệu trên giá sản phẩm.
- Từ trên 1 năm đến nhỏ hơn hoặc bằng 2 năm thì khách hàng được giảm thêm 1 triệu trên giá sản phẩm.

4.3.3. BẢNG QUYẾT ĐỊNH

Bài tập 3

- ✓ *Sử dụng các phương pháp bảng quyết định và phân tích giá trị biên thiết kế các test case kiểm thử các khuyến mãi mà người dùng nhận được khi mua điện thoại cao cấp hãng A?*
- ✓ *Có cần đặt câu hỏi gì thêm để làm rõ hơn yêu cầu?*

Đồ thị nguyên nhân – kết quả

Kỹ thuật dựa trên đồ thị nguyên nhân – kết quả (causes-effects)

- ❖ Là kỹ thuật thiết kế test case dựa trên đồ thị
- ❖ Tập trung vào việc xác định các mối kết hợp giữa các conditions và kết quả mà các mối kết hợp này mang lại

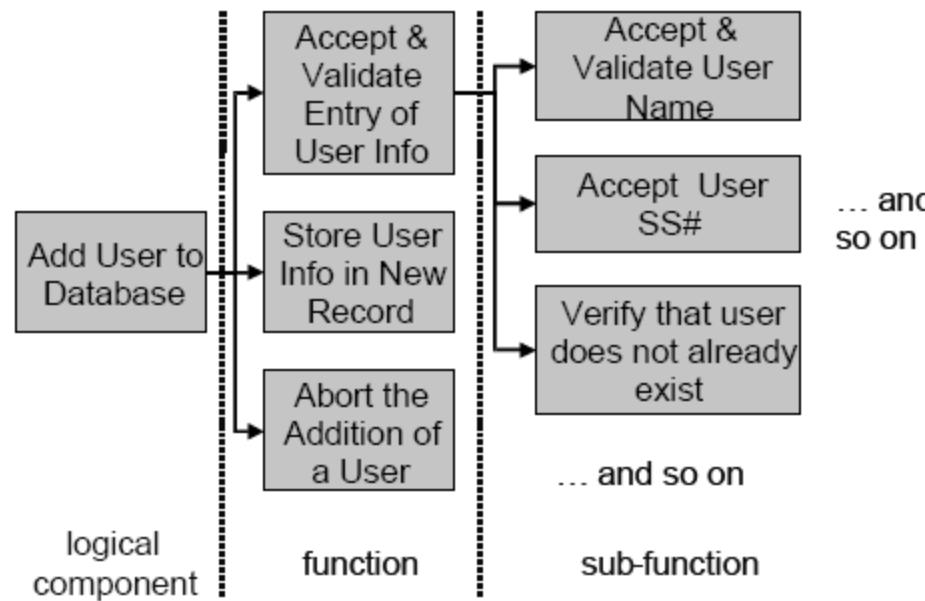
Các bước xây dựng đồ thị

- ❖ **Bước 1:** Phân chia hệ thống thành các vùng hoạt động
- ❖ **Bước 2:** Xác định các nguyên nhân (causes), kết quả (effects)
- ❖ **Bước 3:** Chuyển nội dung ngũ nghĩa trong đặc tả thành đồ thị liên kết các cause và effects
- ❖ **Bước 4:** Chuyển đổi đồ thị thành bảng quyết định
- ❖ **Bước 5:** Thiết lập danh sách test case từ bảng quyết định. Mỗi test case tương ứng với một cột trong bảng quyết định

Bước 1

❖ Phân chia hệ thống thành các vùng hoạt động

- Phân rã các yêu cầu chức năng thành danh sách các functions hay sub-functions



Bước 2

- ❖ **B 2.1:** Dựa vào đặc tả, xác định các causes và chỉ định mỗi causes này 1 định danh ID
 - Một cause có thể được xem như là 1 input conditions hoặc là đại diện của 1 lớp tương đương input conditions
- ❖ **B 2.2:** Dựa vào đặc tả, xác định effects hoặc sự thay đổi trạng thái của hệ thống và chỉ định mỗi effect 1 định danh ID
 - Effect có thể là output action, output condition hay là đại diện của 1 lớp tương đương output conditions

Xác định các causes, effects

- ❖ Ví dụ: Xét đặc tả hệ thống tính phí bảo hiểm xe hơi
 - Đối với nữ < 65 tuổi, phí bảo hiểm là: 500\$
 - Đối với nam < 25 tuổi, phí bảo hiểm là: 3000\$
 - Đối với nam từ 25 đến 64, phí bảo hiểm là: 1000\$
 - Nếu tuổi từ 65 trở lên, phí bảo hiểm là: 1500\$

→ Có 2 yếu tố xác định phí bảo hiểm: **giới tính và tuổi**

Causes (input conditions)	Effects (output conditions)
1. Sex is Male	100. Premium is \$1000
2. Sex is Female	101. Premium is \$3000
3. Age is <25	102. Premium is \$1500
4. Age is >=25 and < 65	103. Premium is \$500
5. Age is >= 65	

Bước 3

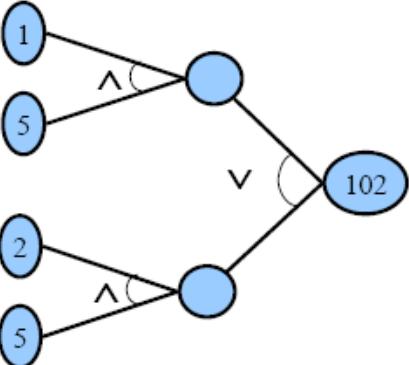
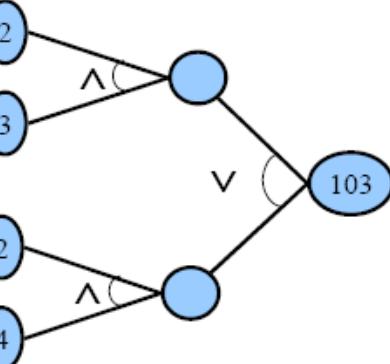
❖ Chuyển nội dung ngũ nghĩa trong đặc tả thành đồ thị liên kết các cause và effects

- CEG #1: Đối với nam từ 25 đến 64, phí bảo hiểm là 1000\$
- CEG #2: Đối với nam < 25 tuổi phí bảo hiểm là

CEG	Interpretation
CEG #1: 	Causes: 1. Sex is Male and (^) 4. Age is ≥ 25 and < 65 Effect: 100: Premium is \$1000
CEG #2: 	Causes: 1. Sex is Male and (^) 3. Age is < 25 Effect: 101: Premium is \$3000

Bước 3

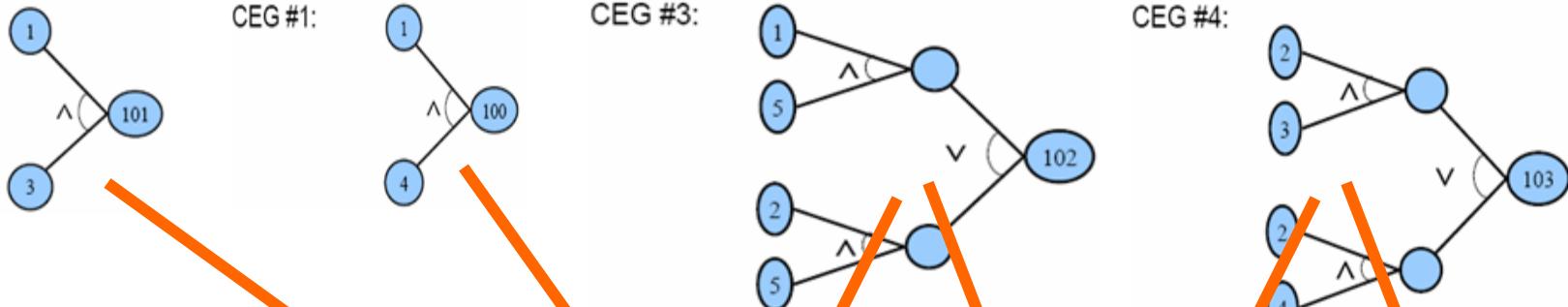
- ❖ CEG #3: Nếu tuổi từ 65 trở lên, phí bảo hiểm là: 1500\$
- ❖ CEG #4: Đối với nữ < 65 tuổi, phí bảo hiểm là: 500\$

CEG #3:  <pre>graph LR; 1((1)) --> C1(()); 2((2)) --> C1; 5_1((5)) --> C1; 5_2((5)) --> C1; C1 --> 102((102)); 102 --> E1(());</pre>	Causes: 1. Sex is Male and (^) 5. Age is ≥ 65 or (∨) 2. Sex is Female and (^) 5. Age is ≥ 65 Effect: 102: Premium is \$1500
CEG #4:  <pre>graph LR; 2_1((2)) --> C2(()); 3((3)) --> C2; 2_2((2)) --> C2; 4((4)) --> C2; C2 --> 103((103)); 103 --> E2(());</pre>	Causes: 2. Sex is Female and (^) 3. Age is < 25 or (∨) 2. Sex is Female and (^) 4. Age is ≥ 25 and < 65 Effect: 103: Premium is \$500

Bước 4: Chuyển đổi đồ thị thành Bảng quyết định

KIỂM THỬ PHẦN MỀM

Test Case	1	2	3	4	5	6
Causes:						
1 (male)	1	1	1	0	0	0
2 (female)	0	0	0	1	1	1
3 (<25)	1	0	0	0	1	0
4 (≥ 25 and < 65)	0	1	0	0	0	1
5 (≥ 65)	0	0	1	1	0	0
Effects:						
100 (Premium is \$1000)	0	1	0	0	0	0
101 (Premium is \$3000)	1	0	0	0	0	0
102 (Premium is \$1500)	0	0	1	1	0	0
103 (Premium is \$500)	0	0	0	0	1	1



Bước 5: Lập danh sách test case từ Bảng quyết định

Test Case	1	2	3	4	5	6
Causes:						
1 (male)	1	1	1	0	0	0
2 (female)	0	0	0	1	1	1
3 (<25)	1	0	0	0	1	0
4 (>=25 and < 65)	0	1	0	0	0	1
5 (>= 65)	0	0	1	1	0	0
Effects:						
100 (Premium is \$1000)	0	1	0	0	0	0
101 (Premium is \$3000)	1	0	0	0	0	0
102 (Premium is \$1500)	0	0	1	1	0	0
103 (Premium is \$500)	0	0	0	0	1	1

Test Case #	Inputs (Causes)		Expected Output (Effects) Premium
	Sex	Age	
1	Male	<25	\$3000
2	Male	>=25 and < 65	\$1000
3	Male	>= 65	\$1500
4	Female	>= 65	\$1500
5	Female	<25	\$500
6	Female	>=25 and < 65	\$500

4.3.4. DỊCH CHUYỂN TRẠNG THÁI

- ❖ Các khía cạnh của hệ thống được mô tả thông qua **máy trạng thái hữu hạn**.
 - Hệ thống xử lý cùng đầu vào, nhưng cho kết quả đầu ra khác nhau gọi máy trạng thái hữu hạn.
 - Máy trạng thái hữu hạn được biểu diễn giống như **lược đồ trạng thái**.
- ❖ Hệ thống sẽ có nhiều trạng thái khác nhau, sự dịch chuyển từ một trạng thái này sang trạng thái khác được quyết định bởi một sự kiện nào đó.

4.3.4. DỊCH CHUYỂN TRẠNG THÁI

- ❖ Một mô hình dịch chuyển trạng thái có 4 phần cơ bản:
 - Các trạng thái (**states**) phần mềm có thể xảy ra.
 - Sự dịch chuyển (**transitions**) từ trạng thái này sang trạng thái khác.
 - Các sự kiện (**events**) dẫn đến sự dịch chuyển trạng thái.
 - Các hành động (**actions**) là kết quả của việc dịch chuyển trạng thái.

4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Ví dụ hệ thống bug tracking

- ✓ Một quy trình sửa lỗi (fix bug) của một hệ thống bug tracking ở một công ty phần mềm như sau:
- ✓ Tester phát hiện bug và tạo báo cáo bug bắt đầu với trạng thái “Open”.
- ✓ Developer xem xét nếu thấy nó không phải bug thì chuyển bug sang thái “Wont Fix” và giải thích cho tester.
- ✓ Nếu tester cũng đồng ý đó không phải bug thì chuyển bug sang trạng thái “Closed”, ngược lại chuyển về trạng thái “Open”.

4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Ví dụ hệ thống bug tracking

- ❖ Nếu developer xem qua thấy bug được tạo đúng là lỗi phần mềm và tiến hành fix bug thì chuyển bug sang trạng thái “In Progress”.
- ❖ Sau khi fix bug xong, developer chuyển nó sang trạng thái “Testing” để tester tiến hành xác nhận (verify) thật sự bug đã được fix.
- ❖ Trong quá trình đang fix bug, developer nhận ra nó không phải lỗi phần mềm thì developer chuyển về trạng thái “Wont Fix”.

4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Ví dụ hệ thống bug tracking

- ❖ Nếu tester kiểm tra qua bug đã được fix và thấy ổn thì chuyển bug sang trạng thái “Closed”, ngược lại chuyển nó về trạng thái “Open” và yêu cầu developer fix lại.
- ❖ Sau khi bug đã đóng, nhưng quá trình test sau đó lại thấy nó tái xuất hiện thì tester có thể chuyển nó về trạng thái “Open” và yêu cầu developer tiếp tục fix.

4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Ví dụ hệ thống bug tracking

- ❖ Các trạng thái: O = Open, IP = In Progress, WF = Wont Fix, T = Testing, C = Closed.
- ❖ Các dịch chuyển:
 - Open → In Progress, Wont Fix
 - In Progress → Wont Fix, Testing
 - Wont Fix → Open, Close
 - Testing → Open, Closed
 - Closed → Open

4.3.4. DỊCH CHUYÊN TRẠNG THÁI

Ví dụ hệ thống bug tracking

❖ Các sự kiện

- E1: bắt đầu fix bug
- E2: không phải bug
- E3: bug đã được fix
- E4: xác nhận bug đã được fix
- E5: bug được fix chưa đúng hoặc chưa đầy đủ
- E6: xác nhận không phải bug
- E7: bug cần fix
- E8: tái xuất hiện bug

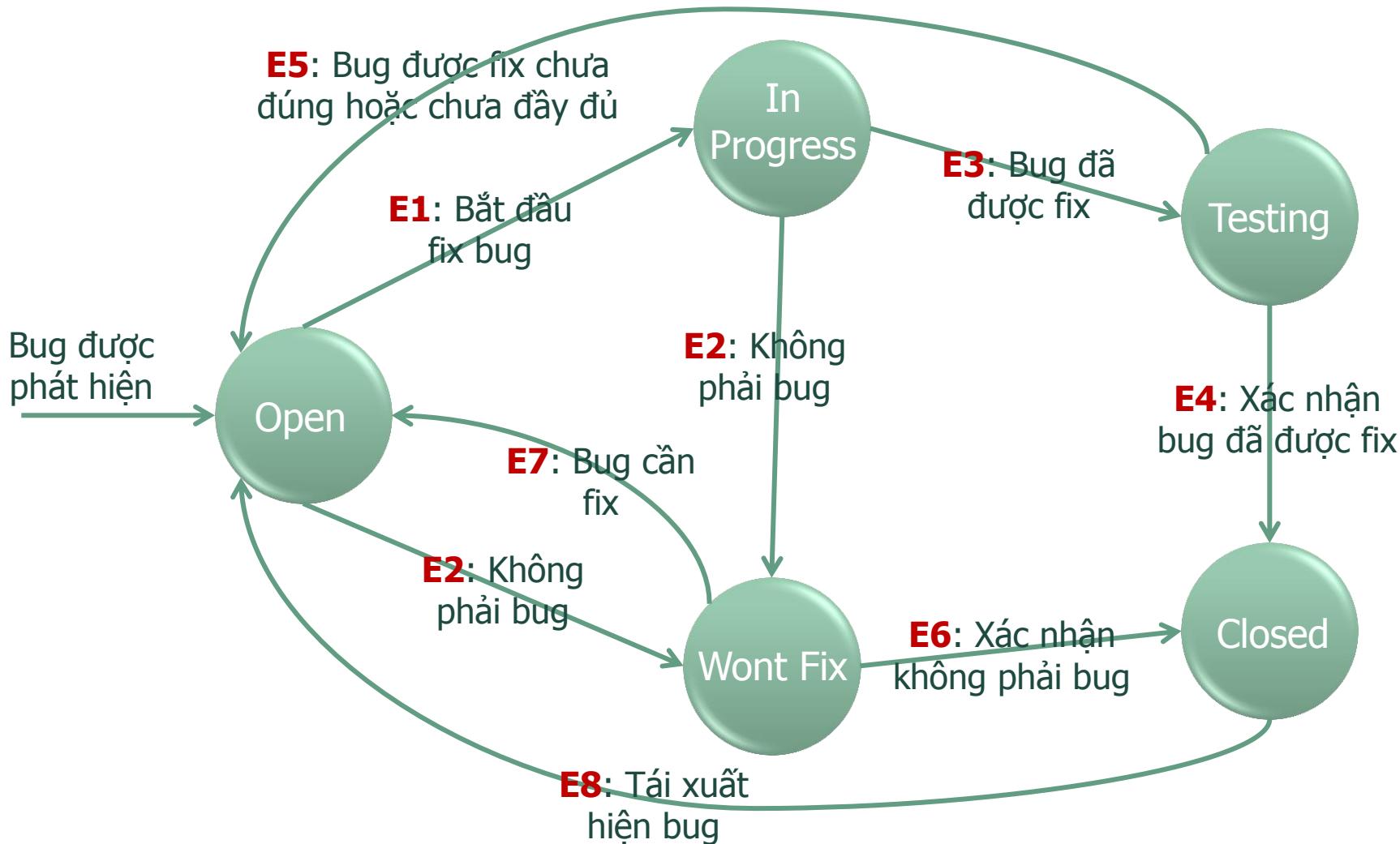
4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Ví dụ hệ thống bug tracking

- ❖ Các hành động kết quả: khi có bất cứ sự thay đổi trạng thái nào của bug, hệ thống sẽ gửi email thông báo đến tất cả các thành viên có liên quan đến bug đó về trạng thái hiện tại của bug.

4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Ví dụ hệ thống bug tracking



4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Ví dụ hệ thống bug tracking

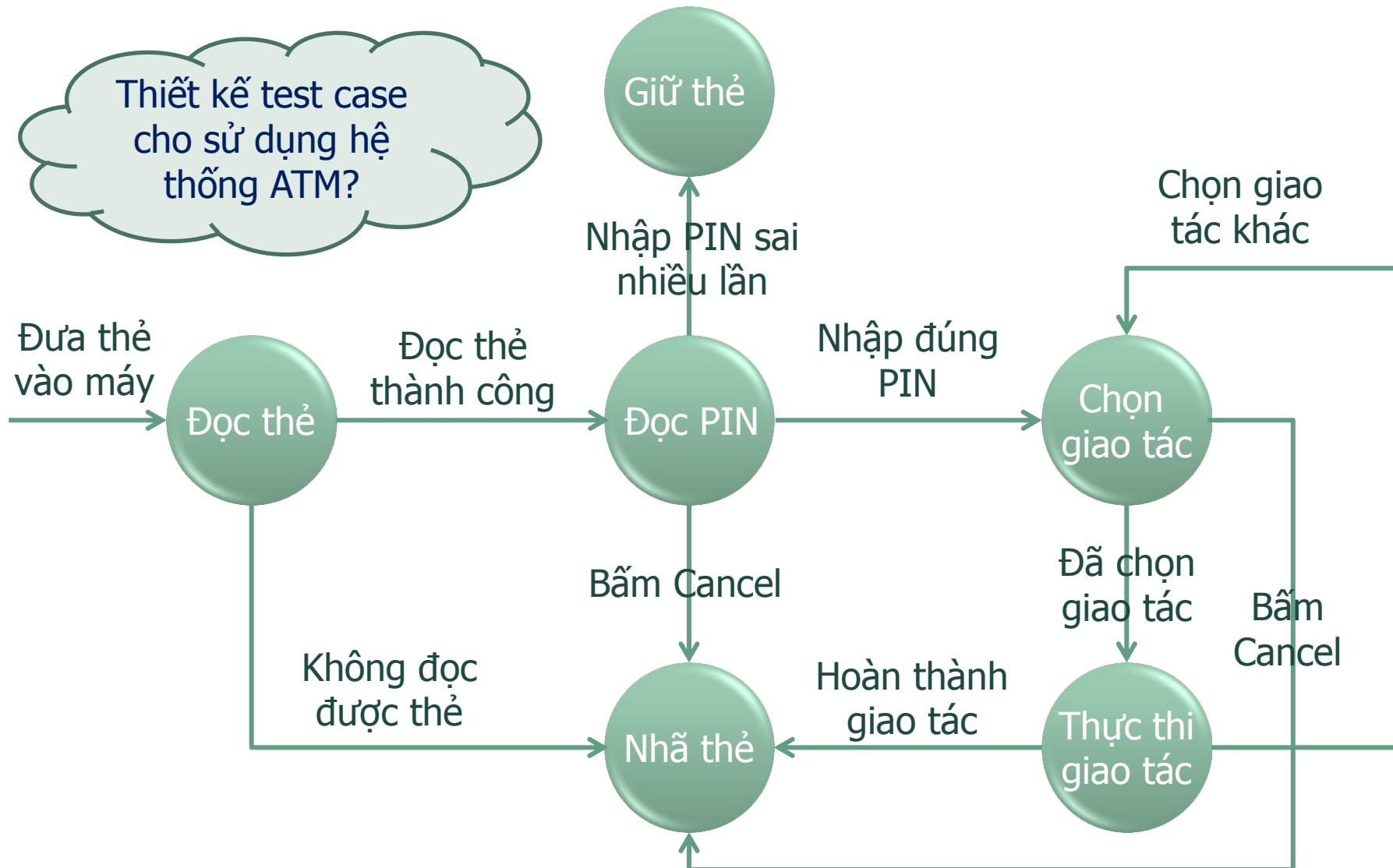
- ❖ Bảng mô tả trạng thái

	E1	E2	E3	E4	E5	E6	E7	E8
O	IP	WF						
IP		WF	T					
T				C	O			
C								O
WF					C	O		

- ❖ Dựa vào bảng trạng thái có thể thiết kế 9 test case hợp lệ: O → IP, O → WF, IP → WF, IP → T, T → C, T → O; C → O, WF → C, WF → O và một vài trường hợp không hợp lệ.

4.3.4. DỊCH CHUYÊN TRẠNG THÁI

Ví dụ sử dụng hệ thống ATM



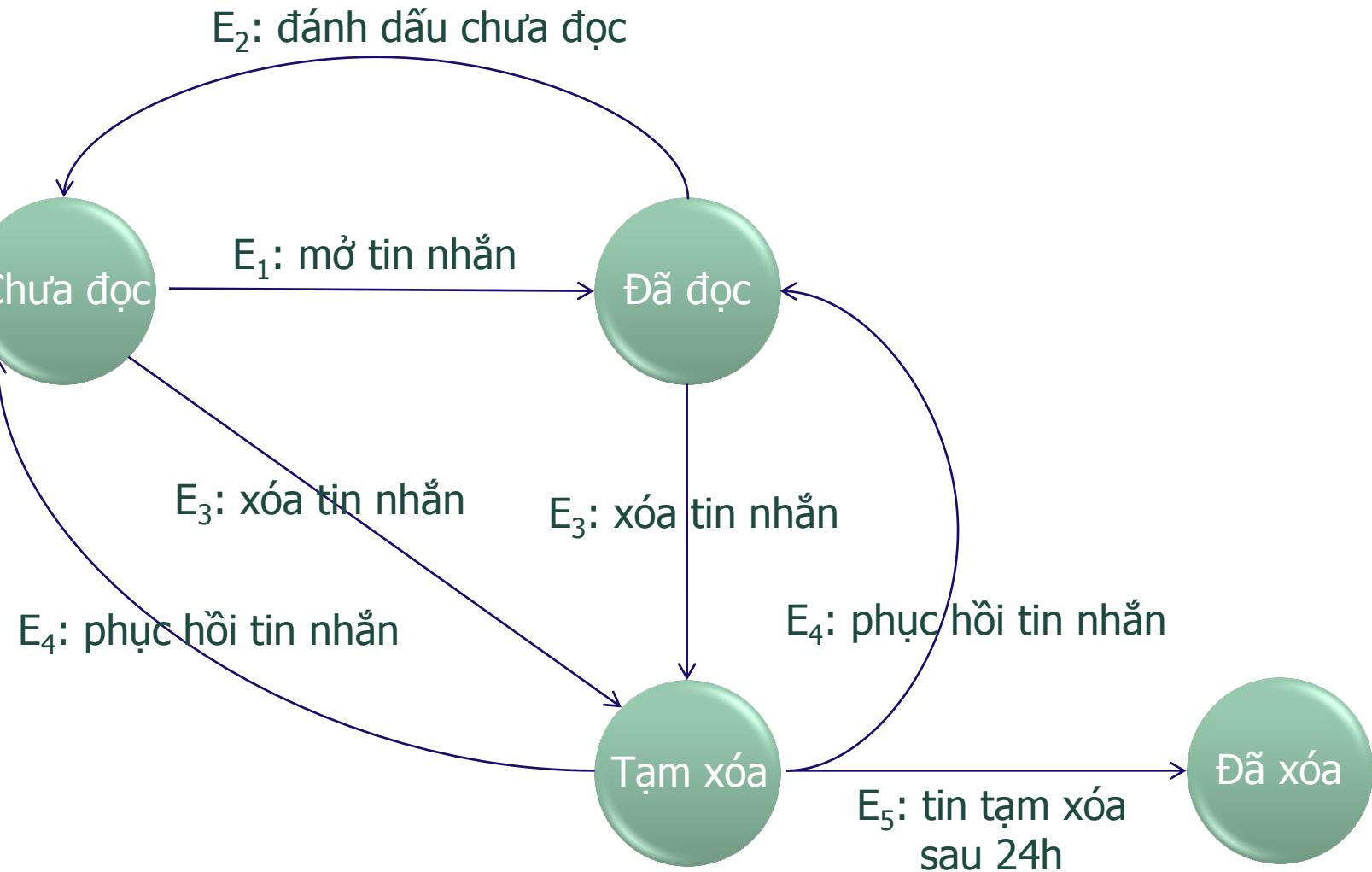
4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Bài tập 1

- ❖ Một hệ thống quản lý cho phép gửi và nhận tin nhắn trong hệ thống, khi người dùng nhận tin mới có trạng thái là tin chưa đọc, nếu người nhận mở ra đọc thì tin đó thành trạng thái đã đọc. Sau khi đọc tin, người dùng cũng có thể chuyển nó thành tin chưa đọc để ghi nhớ. Ngoài ra, người dùng cũng có thể xóa tin tức, ban đầu tin xóa tạm nằm trong thùng rác, trong 24h kể từ lúc xóa người dùng có thể phục hồi lại trạng thái trước khi xóa, sau khoảng thời gian này tin sẽ bị xóa vĩnh viễn.
- ❖ Vẽ lược đồ dịch chuyển trạng thái tin nhắn và viết các test case cho chúng.

4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Hướng dẫn



4.3.4. DỊCH CHUYÊN TRẠNG THÁI

Bài tập 2

Một hệ thống quản lý học tập có chức năng cho phép đăng bài viết, một bài viết khi đăng mới chỉ được phép cập nhật hoặc xóa trong vòng 15 phút kể từ lúc submit đăng bài, sau khoảng thời gian này bài viết không được phép chỉnh sửa hay xóa nữa và bài viết sẽ tự động được xuất bản trên hệ thống để người khác có thể đọc. Ngoài ra, khi vừa soạn xong bài viết hoặc trong vòng 15 phút từ lúc submit bài viết, tác giả bài viết có quyền bấm nút “Publish” để xuất bản bài viết, và tất nhiên không được xóa hoặc cập nhật bài viết sau khi đã xuất bản. Sau khi một bài viết được xuất bản, tác giả bài viết muốn xóa hoặc cập nhật bài viết cần phải liên hệ với admin thực hiện. Chú ý sau khi admin chỉnh sửa bài viết đã xuất bản, bài viết đó vẫn ở trạng thái xuất bản để người khác đọc.

4.4. MỘT SỐ LOẠI KIỂM THỬ HỢP ĐỀN

4.4.1. Function testing

4.4.2. Domain testing

4.4.3. Stress testing

4.4.4. Scenario testing

4.4.5. User testing

4.4.6. Regression testing

4.4.1. FUNCTION TESTING

- ❖ Ý tưởng: kiểm thử qua các chức năng.
- ❖ Một chức năng (function) là một công việc mà phần mềm có thể làm được.
- ❖ Các bước thực hiện
 - Xác định các chức năng
 - Xác định tiêu chuẩn biết chức năng đó làm việc đúng.
 - Mỗi chức năng nên được kiểm thử ít nhất một lần.

4.4.2. DOMAIN TESTING

- ✓ Ý tưởng: dựa trên chiến lược “chia để trị”
- ✓ Một miền (domain) là tập dữ liệu (test data) để kiểm thử.
- ✓ Các bước thực hiện
 - Phân tích giới hạn và các thuộc tính của mỗi miền.
 - Xác định sự kết hợp giữa các miền để kiểm thử
 - Chọn chiến lược test: exhaustive, boundary, best representative.

4.4.3. STRESS TESTING

- ❖ Ý tưởng: kiểm thử với điều kiện bất thường để xem ứng xử của phần mềm.
- ❖ Các bước thực hiện
 - Chọn mục muốn thực hiện stress test.
 - Xác định dữ liệu và các thành phần liên quan.
 - Tạo dữ liệu (challenge data) và thực hiện cấu hình kiểm thử với: cấu trúc dữ liệu lớn và phức tạp, hoặc tải lớn (high load), ...



4.4.4. SCENARIO TESTING

- ❖ Ý tưởng: Kiểm tra kịch bản (scenario) thực hiện bằng cách đặt các kịch bản sử dụng hệ thống và sử dụng nó để viết test case.
- ❖ Kịch bản (scenario) là một câu chuyện (story) mô tả cách thức mà hệ thống được sử dụng.

4.4.5. USER TESTING

- ✓ Ý tưởng: người dùng kiểm thử sản phẩm.
- ✓ Các bước thực hiện:
 - Xác định danh mục người dùng.
 - Tìm người dùng nhờ kiểm thử hoặc hỗ trợ thiết kế test case.

4.4.6. REGRESSION TESTING

- ✓ Ý tưởng: kiểm tra sự thay đổi
- ✓ Kiểm thử hồi quy (Regression testing) được thực hiện mỗi khi hệ thống có thay đổi, nhằm xác định sự thay đổi không ảnh hưởng các chức năng đã được kiểm thử trước đó.
- ✓ Kiểm thử hồi quy có thể thực hiện ở bất cứ mức độ kiểm thử nào.

Bài tập 1

- ❖ Nếu bạn đi xe điện chuyến 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. Tàu hoạt động từ 4:00 sáng tới 23:00 đêm
- ❖ Thiết kế các ca kiểm thử để kiểm tra yêu cầu trên dựa vào phương pháp phân vùng tương đương và phân tích giá trị biên.

Bài tập 2

- ❖ TPPM “xét đơn cầm cố nhà” với đặc tả như sau: mỗi lần nhận 1 đơn xin cầm cố, phần mềm sẽ ra quyết định chấp thuận nếu 4 điều kiện sau thỏa mãn:
 - Thu nhập hàng tháng của người nộp đơn nằm trong khoảng từ 1000\$ với 83333\$
 - Số nhà xin cầm cố từ 1-5
- ❖ Dùng phương pháp phân hoạch tương đương và phân tích giá trị biên để thiết kế các trường hợp kiểm thử cho TPPM trên.

Bài tập 3

- ❖ Viết chương trình dịch, trong đó có câu lệnh FOR, đặc tả câu lệnh FOR như sau: “Lệnh FOR chỉ chấp nhận một tham số duy nhất là biến đếm. Tên biến không được sử dụng quá hai ký tự khác rỗng. Sau ký hiệu = là cận dưới và cận trên của biến đếm. Các cận trên và cận dưới là các số nguyên dương và được đặt giữa từ khóa TO”.
- ❖ Dùng phương pháp phân hoạch tương đương, thiết kế các ca kiểm thử cho câu lệnh FOR

Bài tập 4

- KIỂM THỬ PHẦN MỀM MẶM
- ❖ Bài toán tìm nghiệm thực cho phương trình bậc 2:
 - ❖ Biết a,b,c là các số thực $\in [-10, 100]$
 - ❖ Đầu ra có thể gấp sau khi nhập bộ 3 số a,b,c và bấm nút Calculate là:
 - 1. Không phải là phương trình bậc 2.
 - 2. Phương trình vô nghiệm
 - 3. Phương trình có một nghiệm (đưa ra giá trị của nghiệm)
 - 4. Phương trình có 2 nghiệm (đưa ra giá trị của 2 nghiệm)
 - 5. Nhập sai dữ liệu
 - ❖ Thiết kế các ca kiểm thử dùng phương pháp phân hoạch tương đương

Calculator for the quadratic equation:

$$0 = ax^2 + bx + c$$

a : : b : : c : :

Bài tập 5

- ❖ Chương trình tính chi phí cho bệnh nhân dựa trên độ tuổi và giới tính có màn hình và các yêu cầu như sau:

Calculate the Payment for the Patient

Male Female Child (0 - 17 years)

Age (Years)

Payment is euro €

Male	
Age	Payment
18-35	100 euro
36-50	120 euro
51-145	140 euro

Female	
Age	Payment
18-35	80 euro
36-50	110 euro
51-145	140 euro

Child	
Age	Payment
0-17	50 euro

Bài tập 5

❖ Mô tả chức năng:

- 1. Khởi tạo màn hình:
 - Item 2 được check mặc định ở "Male"
 - Item 3 và 5 null
 - Item 4 ở trạng thái enable (có thể click được)
- 2. Mô tả xử lý chính:
 - Khi click vào item 4 thì xử lý như sau:
 - + Nếu là Male
 - ++ Độ tuổi từ 18 đến 35 thì nhận được 100€
 - ++ Độ tuổi từ 36 đến 50 thì nhận được 120€
 - ++ Độ tuổi từ 51 đến 145 thì nhận được 140€
 - ++ Độ tuổi khác thì hiển thị thông báo lỗi: "Xin vui lòng nhập độ tuổi chính xác"
 - + Nếu là Female
 - ++ Độ tuổi từ 18 đến 35 thì nhận được 80€
 - ++ Độ tuổi từ 36 đến 50 thì nhận được 110€
 - ++ Độ tuổi từ 51 đến 145 thì nhận được 140€
 - ++ Độ tuổi khác thì hiển thị thông báo lỗi: "Xin vui lòng nhập độ tuổi chính xác"

Bài tập 5

- ❖ Dùng phương pháp phân hoạch tương đương và phân tích giá trị biên để xây dựng các ca kiểm thử cho chương trình trên.

Bài tập 1 (Bảng quyết định)

- ❖ Nếu bạn có thẻ đường sắt "over 60s" thì được giảm giá 34% trên tất cả các vé bạn mua.
- ❖ Khi bạn đi cùng với trẻ em (dưới 16 tuổi), thì bạn sẽ được giảm 50% nếu bạn có thẻ "family rail card", trong trường hợp ko có thẻ bạn chỉ được giảm 10%
- ❖ Bạn chỉ được sử dụng 1 hình thức khuyến mại trong 1 giao dịch
- ❖ Hãy viết bảng quyết định liệt kê toàn bộ các kết hợp loại thẻ và kết quả giảm giá. Và viết test case từ bảng quyết định này

Bài tập 2 (Bảng quyết định)

- ❖ Một chương trình phân loại kết quả học của sinh viên dựa trên tổng điểm. Biết tổng điểm của sinh viên (tối đa là 100) trong một kỳ bẳng điểm thành phần cộng điểm thi.
 - ❖ Trong đó điểm thi tối đa là 75 điểm, điểm thành phần tối đa là 25 điểm
 - ❖ Kết quả được phân loại như sau
- | Tổng điểm TD | Kết quả |
|--------------|---------|
| ▪ TD>70 | A |
| ▪ 50<TD<=70 | B |
| ▪ 30< TD<=50 | C |
| ▪ TD<=30 | D |
- ❖ Xây dựng các ca kiểm thử dựa trên:
 - 1. PP phân hoạch tương đương
 - 2. PP phân tích giá trị biên
 - 3. PP bảng hỗ trợ quyết định

BT1 (Giải)

Lập bảng quyết định

KIỂM THỦY PHẦN MỀM PHẦN MỀM

		L1	L2	L3	L4	L5	L6	L7	L8
Điều kiện	Thẻ over 60s	Y	Y	Y	Y	N	N	N	N
	Thẻ family	Y	Y	N	N	Y	Y	N	N
	Đi cùng trẻ em <16t	Y	N	Y	N	Y	N	Y	N
Hành động	Giám giá 34%		Y	Y	Y				
	Giám giá 50%	Y				Y			
	Giám giá 10%							Y	
	Không giảm						Y		Y

Rút gọn L2,4 và L6,8

BT1

❖ Xây dựng testcase (có tất cả 6 testcase)

TC	Inputs	Expected outputs
1	Gia đình Smiths có thẻ over 60s, có thẻ family rail card, đi cùng Oliver (10 tuổi)	Được giảm 50%
2	Gia đình Smiths có thẻ over 60s, không đi cùng trẻ em	Được giảm 34%
3	Gia đình Smiths có thẻ over 60s, đi cùng Oliver (10 tuổi)	Được giảm 34%
4	Gia đình Smiths ko có thẻ over 60s, có thẻ family raid card, đi cùng Oliver (10t)	Được giảm 50%
5	Ông Smith ko có thẻ over 60s, ko có thẻ family raid card, đi cùng Oliver (10t)	Được giảm 10%
6	Ông Smith đi một mình, ko có thẻ nào	Không được giảm

BT2 (Giải)

❖ A. Bảng phân hoạch tương đương

Đầu vào	Điểm thi	Điểm thành phần	Điểm tổng	Kết quả
Vùng hợp lệ	[0,75] (H1)	[0,25] (H2)	TĐ>70	A
			50< TĐ <= 70	B
			30 < TĐ <= 50	C
			TĐ <= 30	D
Vùng không hợp lệ	<0 (K1)	<0 (K3)		
	>75 (K2)	>25 (K4)		

BT2

❖ Các ca kiểm thử

TC	Inputs Điểm thi	Điểm TP	E.Output Điểm tổng	Cover
1	60	15	75 A	H1 H2
2	30	20	60B	H1 H2
3	20	20	35C	H1 H2
4	10	20	20 D	H1 H2
5	60	-10	Không hợp lệ	H1 K3
6	60	30	Không hợp lệ	H1 K4
7	60	20	Không hợp lệ	K1 H2
8	-60	20	Không hợp lệ	K2 H2

BT2

❖ B Phân tích các giá trị biên

TC	Inputs		E. Output	TC	Inputs		E. Output
	Điểm thi	Điểm TP			Điểm thi	Điểm TP	
1	-1	0	Ko hợp lệ	13	0	1	1 D
2	-1	25	Ko hợp lệ	14	13	16	29 D
3	75	26	Ko hợp lệ	15	14	16	30 D
4	76	25	Ko hợp lệ	16	15	16	31 C
5	75	-1	Ko hợp lệ	17	20	29	49 C
6	76	0	Ko hợp lệ	18	21	29	50 C
7	0	26	Ko hợp lệ	19	22	29	51 B
8	0	-1	Ko hợp lệ	20	49	20	69 B
9	0	0	0 D	21	50	20	70 B
10	0	25	25 D	22	51	20	71 A
11	75	0	75 A	23	75	24	99 A
12	75	25	100 A				