

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.
 - ✓ Dịch chuyển trạng thái.
- 4.4. Một số loại kiểm thử hộp đen.

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

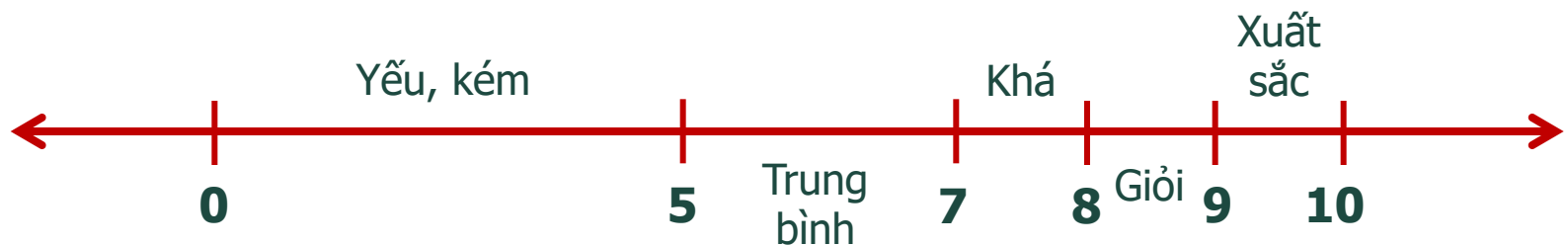
Ví dụ

- ❖ 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, 3.0, 5.5, 7.5, 8.5, 9.5, 12.0

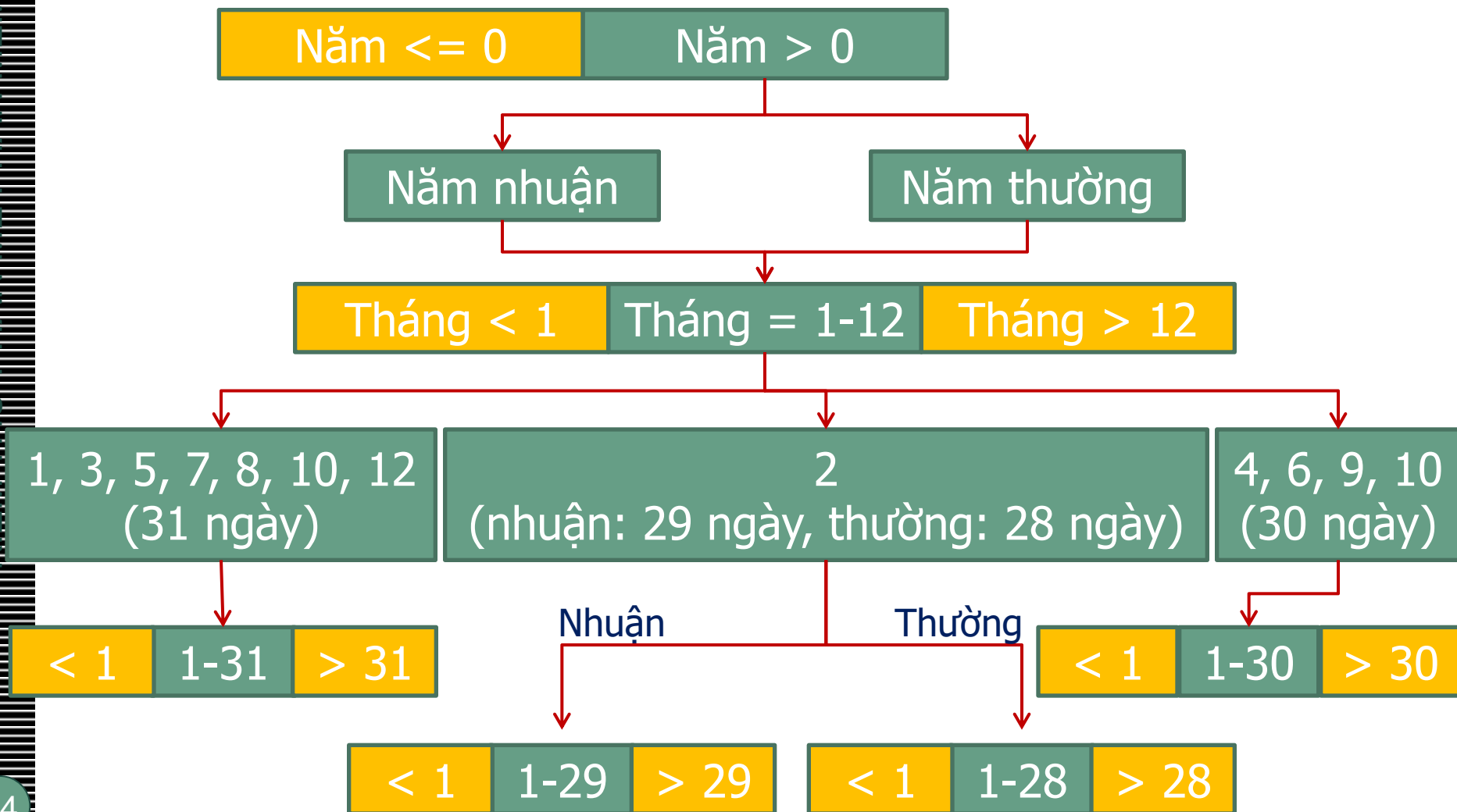
4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

Bài tập 1

- ✓ Chương trình kiểm tra một ngày tháng nhập vào có hợp lệ hay không? 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



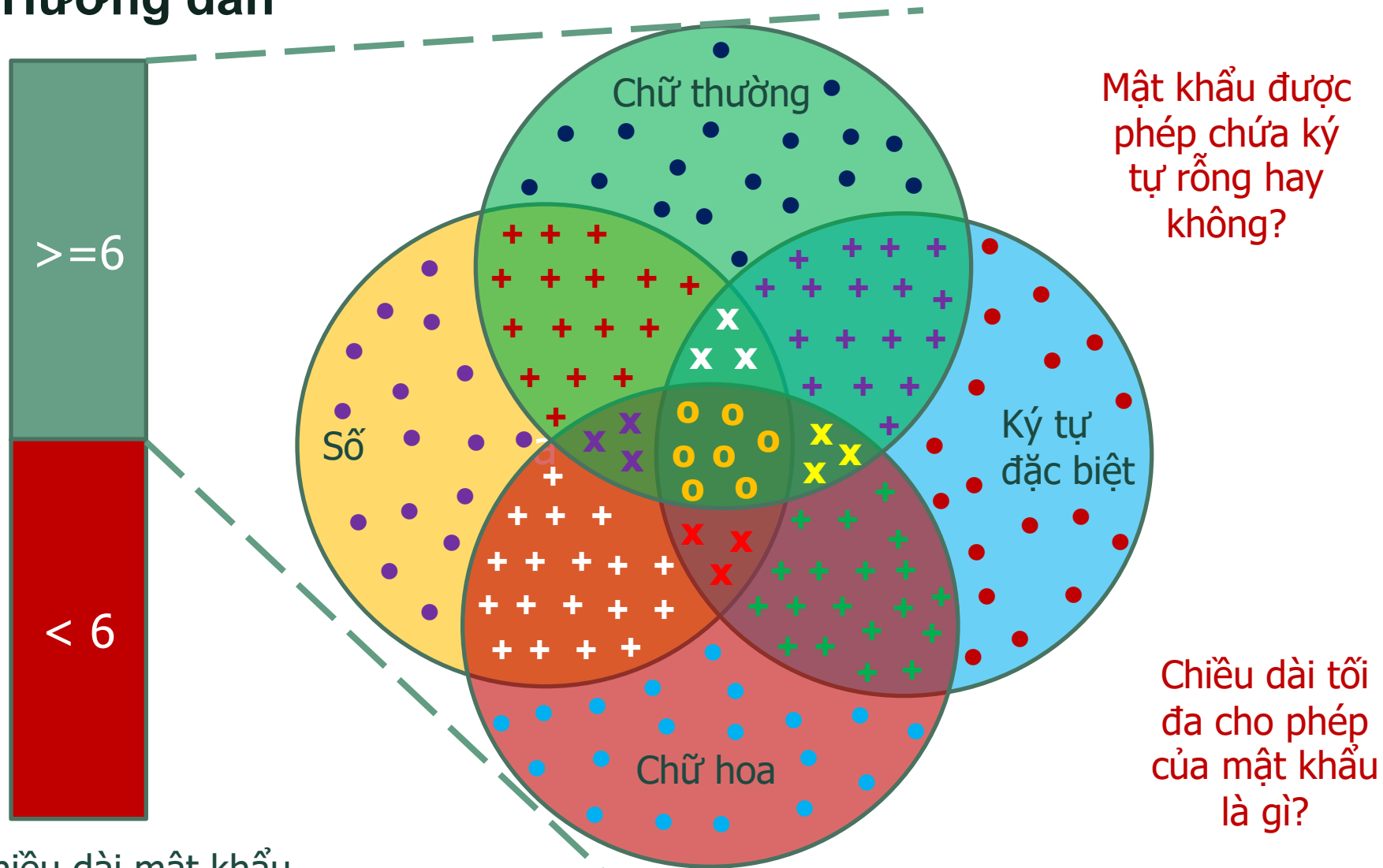
4.3.1. PHÂN VÙNG TƯƠNG ĐƯƠNG

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



Chiều dài mật khẩu

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).

4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

❖ Một số quy tắc xác định giá trị biên khi thiết kế test case

- Giá trị nhỏ nhất (biên dưới)
- Giá trị nhỏ thứ hai (cận biên dưới)
- Giá trị lớn nhất (biên trên)
- Giá trị lớn thứ hai (cận biên trên)
- Giá trị bình thường

❖ Ví dụ 1: trở lại ví dụ xếp loại học sinh



4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

Ví dụ 1

- ❖ 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.

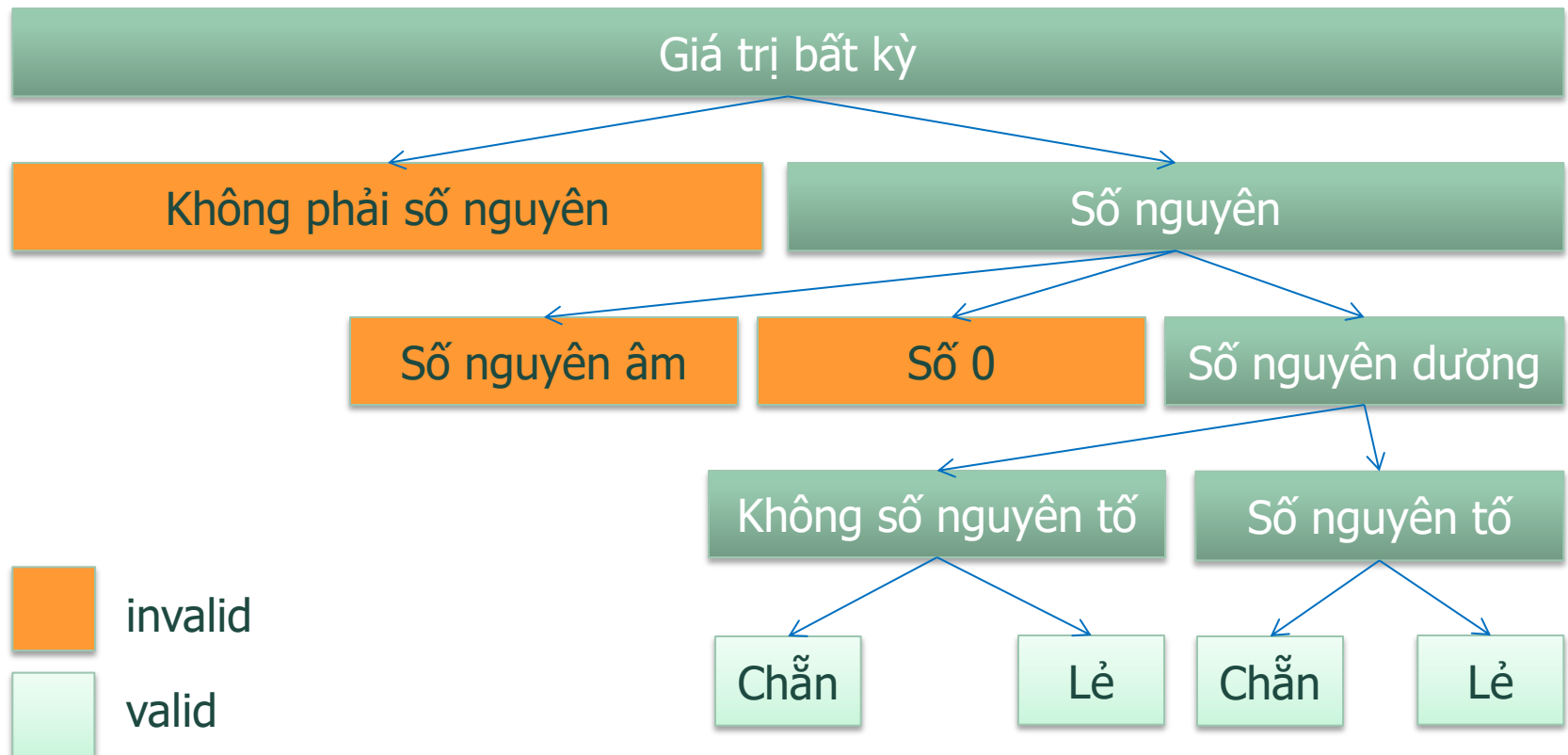
C:\WINDOWS\system32\cmd.exe	C:\WINDOWS\system32\cmd.exe	C:\WINDOWS\system32\cmd.exe
n = 2	n = 12	n = 0
NGUYEN TO	KHONG NGUYEN TO	KHONG NGUYEN TO
n = 7	n = 143	n = 1
NGUYEN TO	KHONG NGUYEN TO	KHONG NGUYEN TO
n = 13	n = 112	n = -7
NGUYEN TO	KHONG NGUYEN TO	KHONG NGUYEN TO
n = 113	n = 58	n = -6
NGUYEN TO	KHONG NGUYEN TO	KHONG NGUYEN TO
n = 71	n = 98	n = -53
NGUYEN TO	KHONG NGUYEN TO	KHONG NGUYEN TO



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

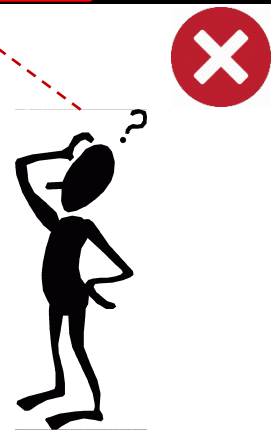
Ví dụ 2

❖ 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
```

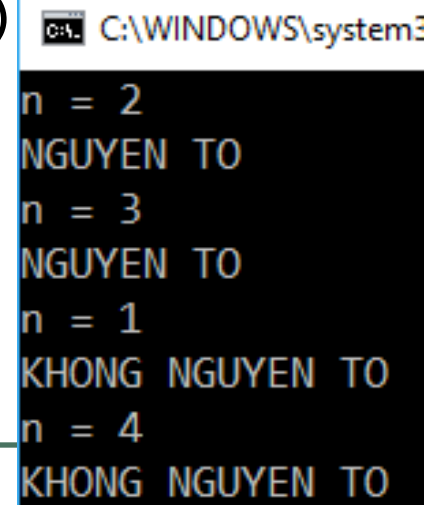


4.3.2. PHÂN TÍCH GIÁ TRỊ BIÊN

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;  
}
```



```
C:\WINDOWS\system32\cmd.exe  
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 điều 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.

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 trên 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 gom 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.

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			Mong muốn	
	K	X	M	Mượn sách?	Số sách tối đa được mượn.
C ₁	×	1	499	×	0
C ₂	✓	6	494	×	0
	✓	6	495	×	0
C ₃	✓	1	500	×	0
	✓	5	496	×	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 ≥ 20 tr. 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?*

4.3.4. DỊCH CHUYỂN TRẠNG THÁI

- ❖ Các khía cạnh của hệ thống được mô tả thông qua **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 CHUYỂN 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

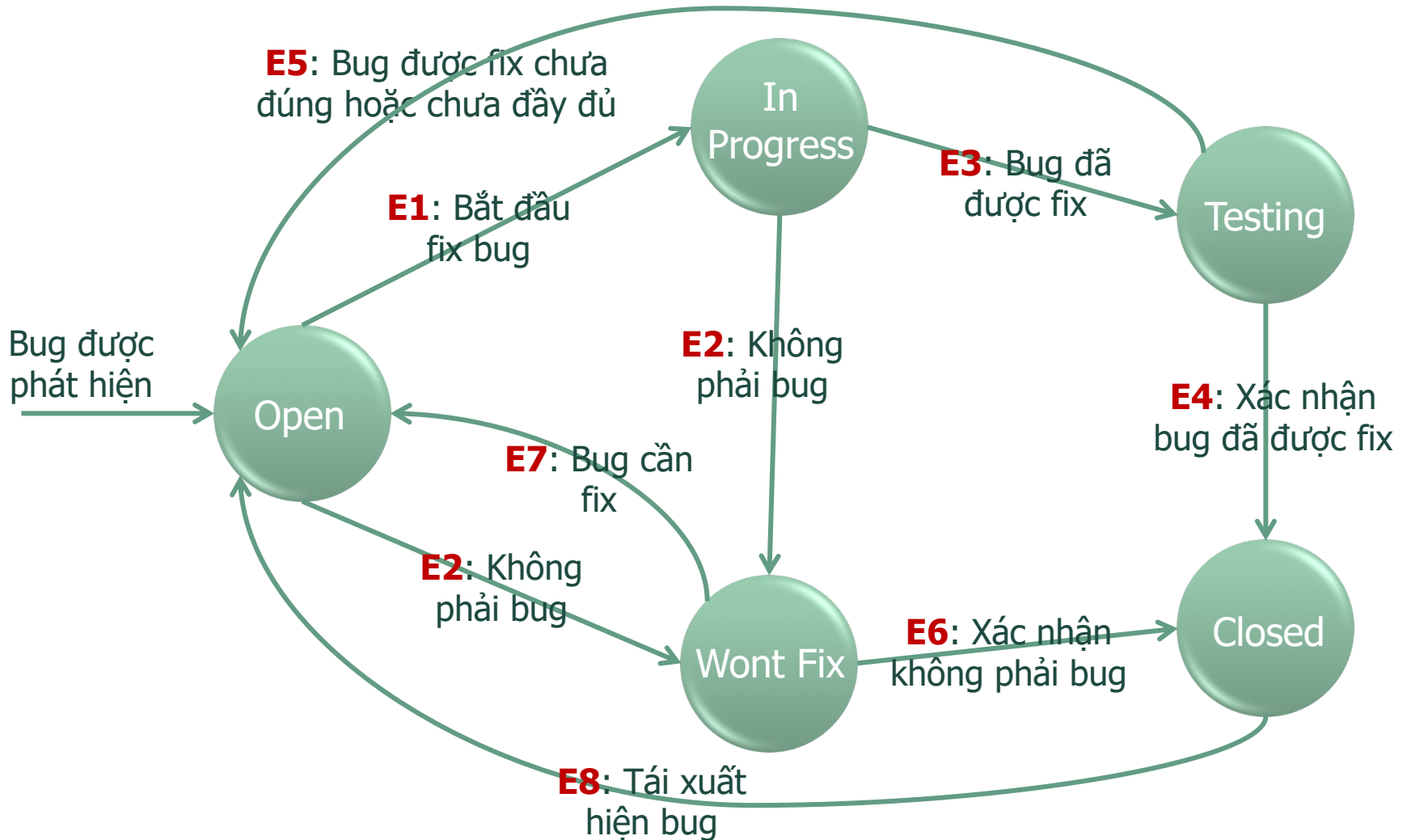
4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Ví dụ hệ thống bug tracking

- ❖ Các hành động kết quả: khi có bất kỳ 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

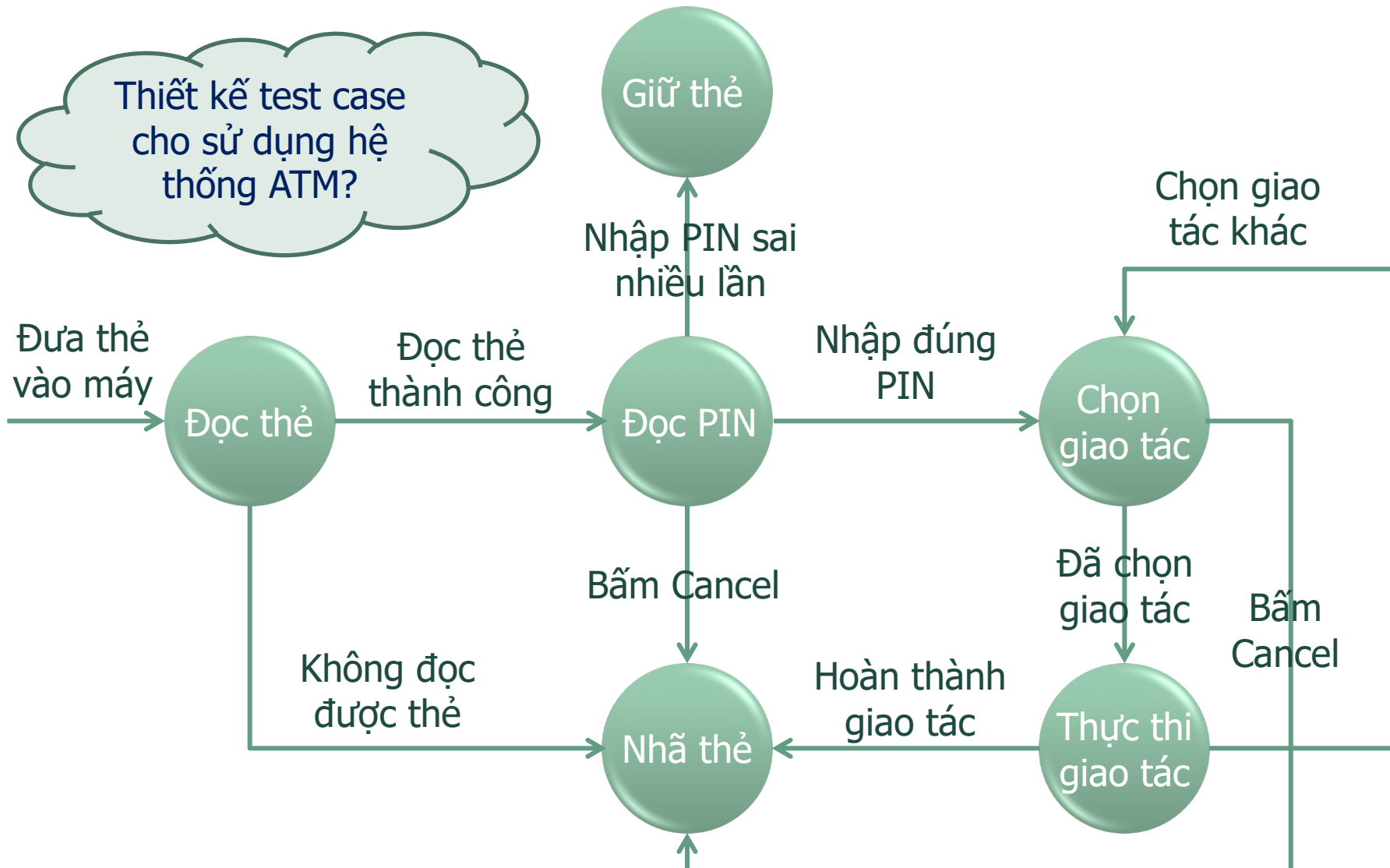
❖ Bảng mô tả trạng thái

	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: $O \rightarrow IP$, $O \rightarrow WF$, $IP \rightarrow WF$, $IP \rightarrow T$, $T \rightarrow C$, $T \rightarrow O$; $C \rightarrow O$, $WF \rightarrow C$, $WF \rightarrow O$ và một vài trường hợp không hợp lệ.

4.3.4. DỊCH CHUYỂN TRẠNG THÁI

Ví dụ sử dụng hệ thống ATM



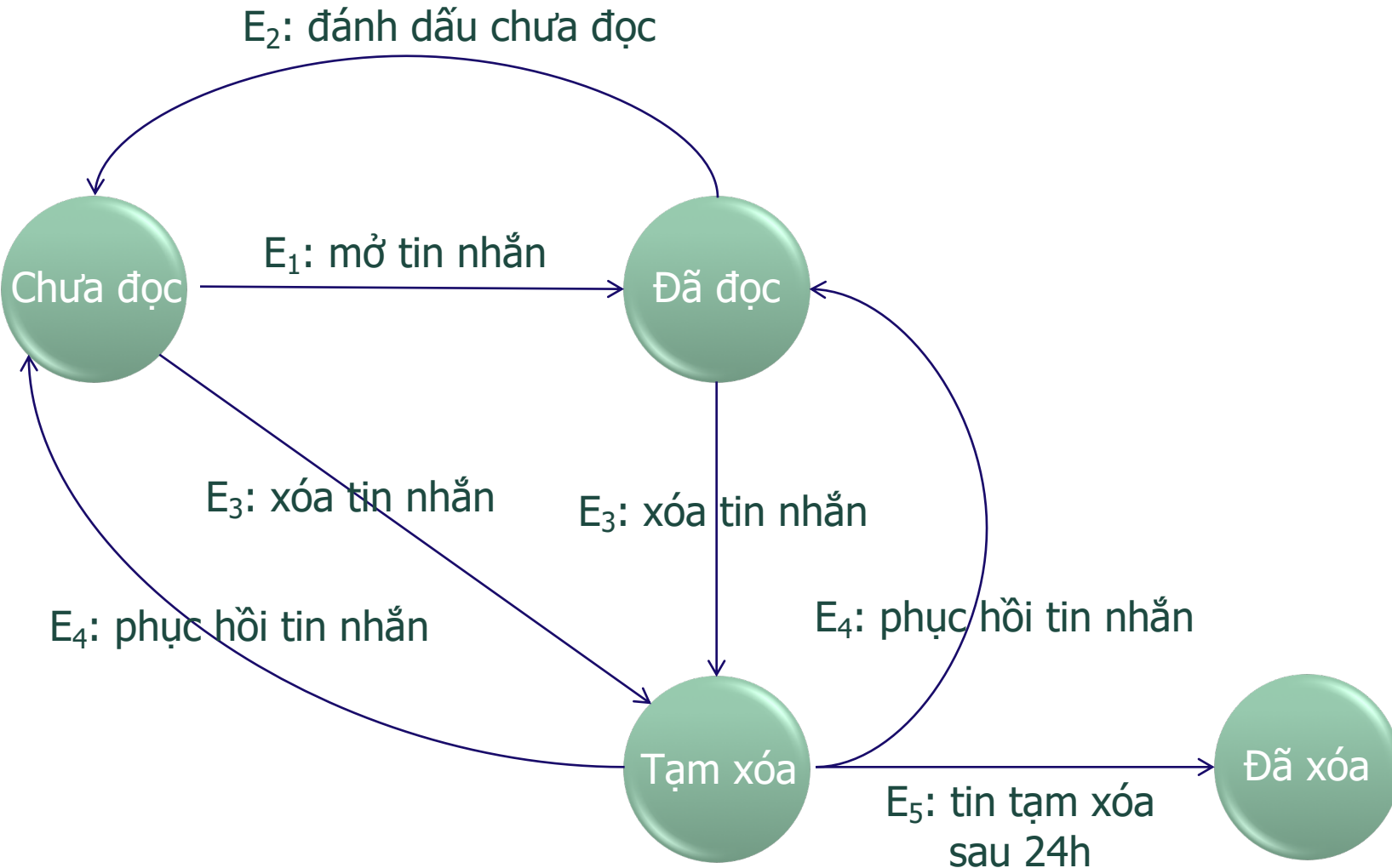
4.3.4. DỊCH CHUYỂN TRẠNG THÁI

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 CHUYỂN 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 HỘP ĐEN

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 (đầy đủ), boundary (*biên), best representative (giá trị tốt nhất).

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.



HUTECH

Đại học Công nghệ Tp.HCM

KIỂM THỬ PHẦN MỀM

Bài 4:

THIẾT KẾ TEST CASE BACKBOX

Q&A