

Kiểm chứng phần mềm

State Transition Testing & Use Case Testing



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



State Transition Testing

- ☐ Giúp kiểm tra sự thay đổi trạng thái của hệ thống
- ☐ Gồm 3 bước
 - ☒ Mô hình hệ thống như một Máy trạng thái (Finite state machine) hoặc Lược đồ chuyển trạng thái (State transition diagram)
 - ☒ Lập Bảng trạng thái (State Table) để xem xét các bước chuyển trạng thái có thể gây lỗi (Invalid transition)
 - ☒ Thiết kế các ca kiểm thử từ Bảng trạng thái và Mô hình

1. State Transition Diagram

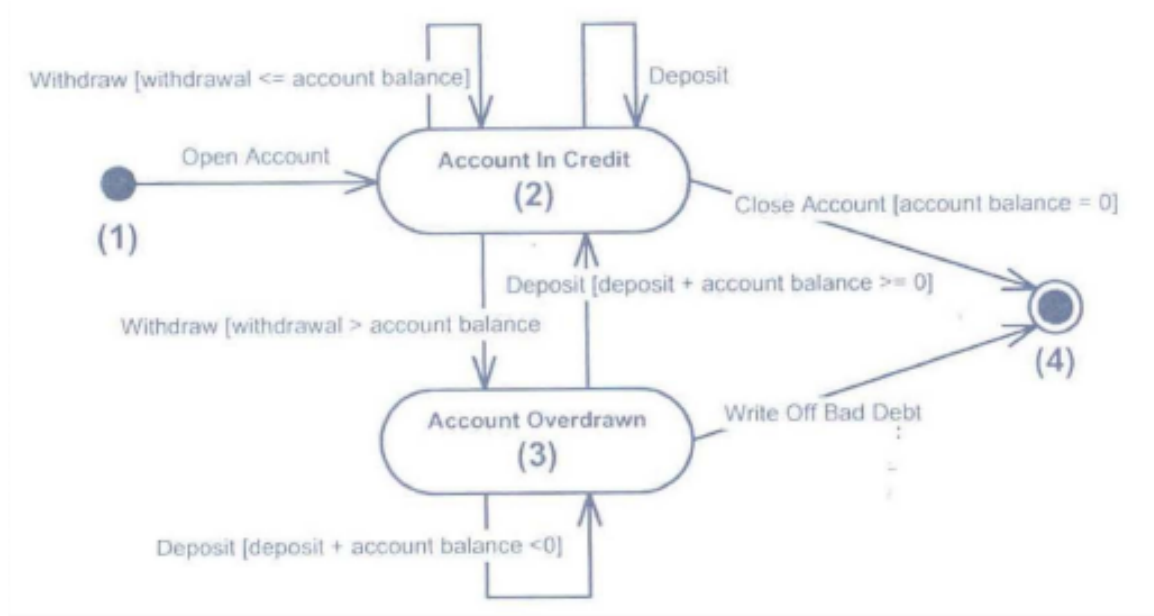
- ☐ **Đỉnh (State)**
 - ☐ Một trạng thái của hệ thống
- ☐ **Cạnh (Transition)**
 - ☐ Bước chuyển từ trạng thái này sang trạng thái khác của hệ thống
- ☐ **Sự kiện (Event)**
 - ☐ Sự kiện khiến hệ thống thay đổi trạng thái
- ☐ **Hành động (Action)**
 - ☐ Kết quả của việc chuyển trạng thái (vd thông báo lỗi...)



- ☐ Nếu chưa có tài khoản khách hàng có thể yêu cầu mở tài khoản. Tài khoản ở trạng thái dư nợ khách hàng có thể yêu cầu nạp tiền và rút tiền. Nếu yêu cầu rút tiền nhiều hơn dư nợ hiện có tài khoản sẽ chuyển sang trạng thái thiếu nợ và chỉ khi khách hàng nạp tiền nhiều hơn nợ hiện có thì mới trở lại trạng thái dư nợ. Khi yêu cầu đóng tài khoản nếu ở trạng thái dư nợ số dư sẽ bằng 0, nếu ở trạng thái thiếu nợ sẽ yêu cầu trả nợ.



Ví dụ: Sơ đồ trạng thái Tài khoản ngân hàng



4

2. State Transition Table

- Liệt kê **tất cả tổ hợp bước chuyển giữa các trạng thái**, không nhất là bước chuyển hợp lệ
 - ▣ Ưu điểm: Giúp phát hiện các đường đi không hợp lệ giữa các trạng thái
 - ▣ Khuyết điểm: Bùng nổ tổ hợp trong trường hợp có nhiều trạng thái và sự kiện

5

Vd: Bảng trạng thái cho Tài khoản ngân hàng

Prior State	New State	Valid Transition	Comment
1	1	N	
1	2	Y	New account
1	3	N	Possible negative test case
1	4	N	
2	1	N	
2	2	Y	Deposit and withdraw [withdrawal <= account balance]
2	3	Y	Withdraw [withdrawal > account balance]
2	4	Y	Closed account [account balance=0]
3	1	N	
3	2	Y	Deposit [deposit + account balance >= 0]
3	3	Y	Deposit [deposit + account balance < 0]
3	4	Y	Write Off Bad Debt [account balance < 0]
4	1	N	
4	2	N	Possible negative test case
4	3	N	Possible negative test case
4	4	N	Possible negative test case

6

3. Thiết kế Test case từ Bảng trạng thái

#TC	Precondition (State)	Condition (Event)	Expected Result (Action)	Note
TC1	Chưa có tài khoản	Mở tài khoản	Tài khoản được tạo với balance >=0	S1 => S2
TC2	Chưa có tài khoản	Rút tiền	Thông báo không tồn tại tài khoản	S1 => S3
TC3	Chưa có tài khoản	Đóng tài khoản	Thông báo không tồn tại tài khoản	S1 => S4
TC4	Tài khoản có balance >=0	Gửi tiền D	Balance = balance + D	S2 => S2
TC5	Tài khoản có balance >=0	Rút số tiền W <= balance	Balance = balance - W >= 0	S2 => S2

7

3. Thiết kế Test case từ Bảng trạng thái

#TC	Precondition (State)	Condition (Event)	Expected Result (Action)	Note
TC6	Tài khoản có balance ≥ 0	Rút số tiền $W > \text{balance}$	Balance = (balance - W) < 0	S2 \Rightarrow S3
TC7	Tài khoản có balance ≥ 0	Đóng tài khoản	Tài khoản bị đóng Balance = 0	S2 \Rightarrow S4
TC8	Tài khoản có balance < 0	Gửi tiền $D + \text{Balance} \geq 0$	Balance = Balance + $D \geq 0$	S3 \Rightarrow S2
TC9	Tài khoản có balance < 0	Gửi tiền $D + \text{Balance} < 0$	Balance = Balance + $D < 0$	S3 \Rightarrow S3
TC10	Tài khoản có balance < 0	Write bad debit	Tài khoản ghi nợ	S3 \Rightarrow S4



8

3. Thiết kế Test case từ Bảng trạng thái (tt)

#TC	Precondition (State)	Condition (Event)	Expected Result (Action)	Note
TC11	Tài khoản ghi nợ	Gửi tiền $D + \text{Balance} \geq 0$	Balance = Balance + $D \geq 0$	S4 \Rightarrow S2
TC12	Tài khoản ghi nợ	Gửi tiền $D + \text{Balance} < 0$	Balance = Balance + $D < 0$	S4 \Rightarrow S3
TC13	Tài khoản bị đóng	Gửi tiền	Thông báo tài khoản đã bị đóng	S4 \Rightarrow S2
TC14	Tài khoản ghi nợ	Rút tiền	Thông báo tài khoản ghi nợ	S4 \Rightarrow S3
TC15	Tài khoản bị đóng	Rút tiền	Thông báo tài khoản đã bị đóng	S4 \Rightarrow S3
TC16	Tài khoản ghi nợ	Đóng tài khoản	Thông báo tài khoản ghi nợ	S4 \Rightarrow S4
TC17	Tài khoản bị đóng	Đóng tài khoản	Thông báo tài khoản đã bị đóng	S4 \Rightarrow S4



9

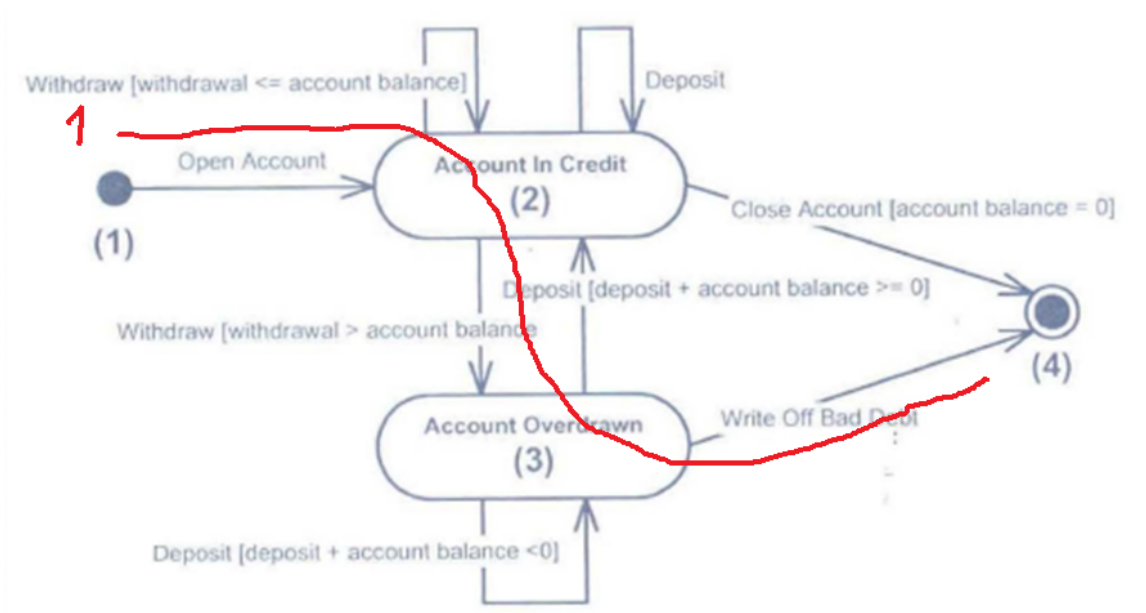
3. Thiết kế Test Case từ Mô hình trạng thái

- Tạo tập Test case phủ
 - ▣ Tất cả các trạng thái (Đỉnh)
 - ▣ Tất cả các bước chuyển trạng thái (Cạnh)



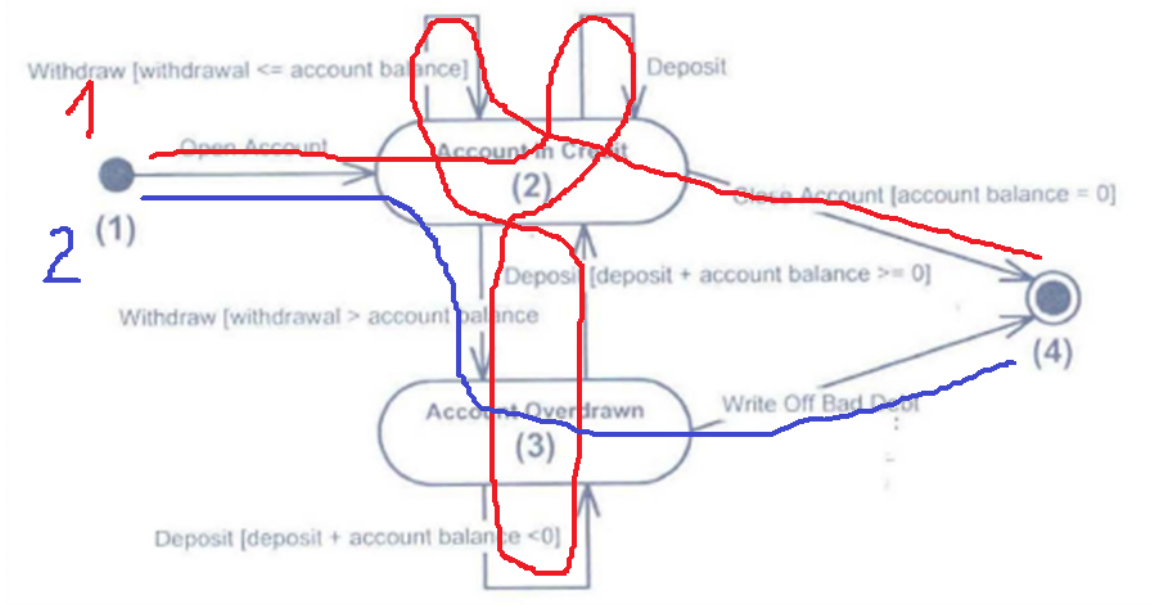
10

Test Case phủ tất cả các đỉnh



11

Test Case phủ tất cả các cạnh



12

Bài tập

- 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:
 - ▣ 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.
 - ▣ Đưa ra một bảng trạng thái. Cho một ví dụ kiểm thử trường hợp chuyển tiếp không hợp lệ.

13

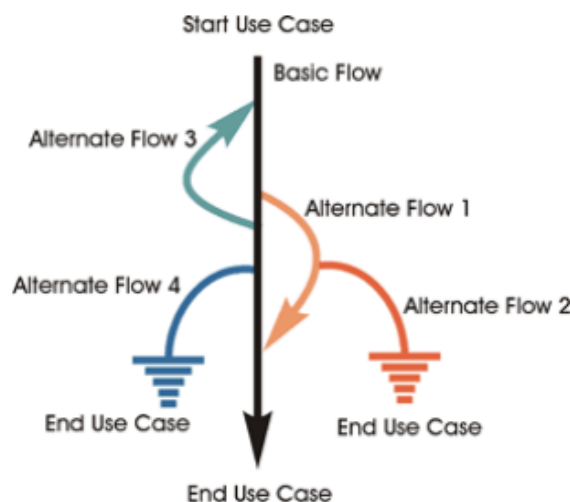
Use Case Testing

- Use Case: mô tả một chuỗi các hành động của người dùng hoặc hệ thống khác (Actor) tương tác với hệ thống
 - Brief description
 - Flow of events
 - Basic flow
 - Alternative flow 1
 - Alternative flow 2
 - Special requirements
 - Preconditions
 - Post-conditions
 - Activity diagram



14

Flow of Events



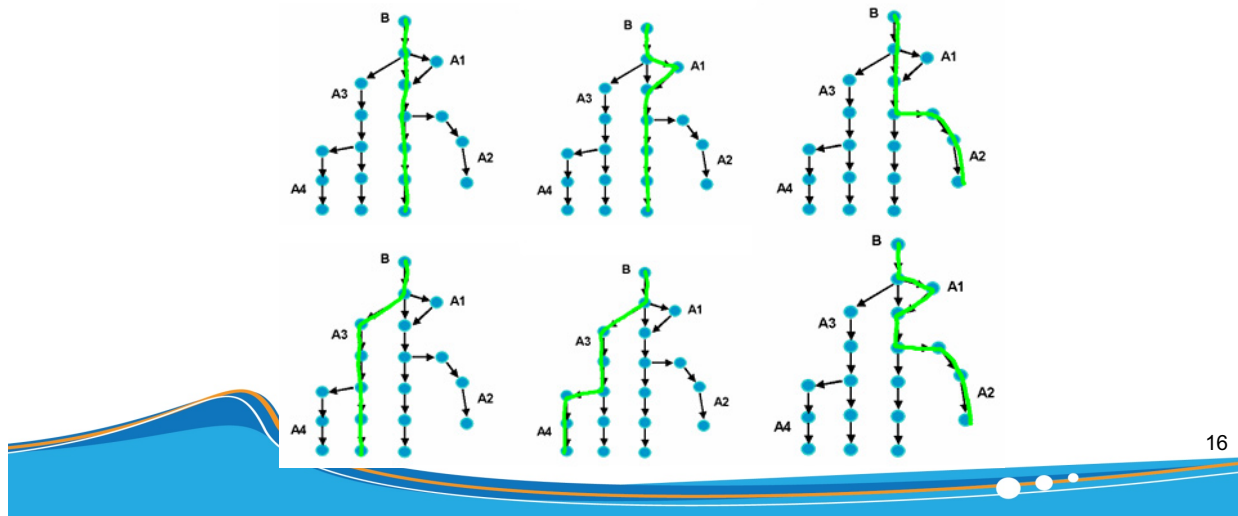
- Scenario: một chuỗi các sự kiện từ lúc bắt đầu đến kết thúc 1 Use Case



15

Xác định Scenario

- ☐ Basic Flow
- ☐ Combination of Basic Flow & Alternate Flows
- ☐ Combination of Alternate Flows



16

Scenario

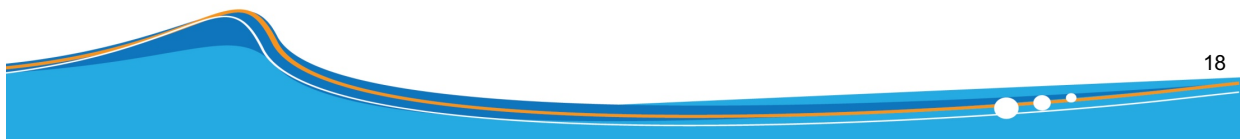
Scenario	Starting Flow	Alternate Flow		
S1	Basic Flow			
S2	Basic Flow	Alternate Flow 1		
S3	Basic Flow	Alternate Flow 1	Alternate Flow 2	
S4	Basic Flow	Alternate Flow 3		
S5	Basic Flow	Alternate Flow 3	Alternate Flow 1	
S6	Basic Flow	Alternate Flow 3	Alternate Flow 1	Alternate Flow 2
S7	Basic Flow	Alternate Flow 4		
S8	Basic Flow	Alternate Flow 3	Alternate Flow 4	



17

Thiết kế Test Case từ Use Case

- ☐ Xác định tất cả Scenario từ Use Case
- ☐ Với mỗi Scenario xác định ít nhất 1 Test Case và tập điều kiện để TC có thể thực thi
- ☐ Với mỗi Test Case xác định dữ liệu kiểm thử (Test Data) cho việc kiểm thử



18

Vd: Đăng ký học phần

- ☐ Basic Flow
 1. Logon
 2. Select “Create a schedule”
 3. Obtain Course Information
 4. Select Courses
 5. Submit Schedule
 6. Display Completed Schedule



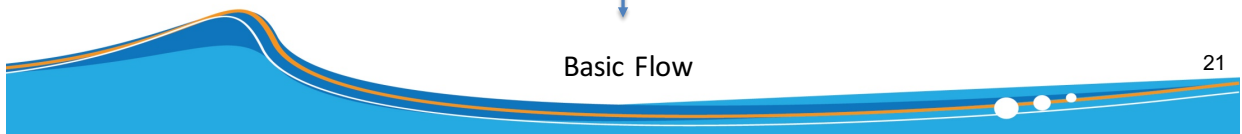
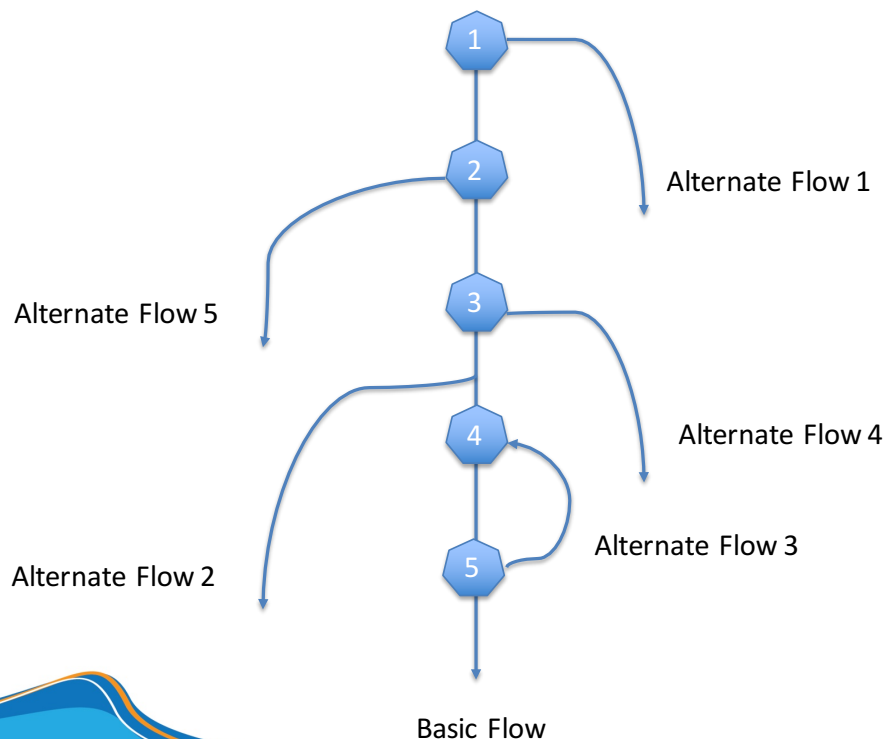
19

□ Alternate Flow

1. Unidentified Student
2. Quit
3. Unfulfilled Prerequisites, Course Full or Schedule Conflicts
4. Course Catalog System Unavailable
5. Course Registration Closed



Activity Diagram



Xác định Scenario

Scenario Name	Starting Flow	Alternate Flow
Scenario 1 – Successful Registration	Basic Flow	
Scenario 2 – Unidentified Student	Basic Flow	Alternate Flow 1
Scenario 3 – User quits	Basic Flow	Alternate Flow 2
Scenario 4 – Course Catalog System Unavailable	Basic Flow	Alternate Flow 4
Scenario 5 – Registration Closed	Basic Flow	Alternate Flow 5
Scenario 6 – Cannot enroll	Basic Flow	Alternate Flow 3



22

Xác định Test Case

- ☐ Xác định input cho từng bước (Step) của Use case
- ☐ Xác định miền giá trị cho từng input
- ☐ Kết hợp miền giá trị của input cho từng Test Case



23

Xác định input cho từng bước

☐ B1

☐ Student ID

☐ Password

☐ B3

☐ Prerequisites Fulfilled

☐ Schedule Open

☐ Course Open

☐ B4

☐ Course Selected



24

Test Case ID	Scenario/Condition	Student ID	Password	Courses selected	Prerequisites fulfilled	Course Open	Schedule Open	Expected Result
RC 1	Scenario 1-successful registration	V	V	V	V	V	V	Schedule and confirmation number displayed
RC 2	Scenario 2-unidentified student	I	N/A	N/A	N/A	N/A	N/A	Error message; back to login screen
RC 3	Scenario 3-valid user quits	V	V	N/A	N/A	N/A	N/A	Login screen appears
RC 4	Scenario 4-course registration system unavailable	V	V	N/A	N/A	N/A	N/A	Error message; back to step 2
RC 5	Scenario 5-registration closed	V	V	N/A	N/A	N/A	N/A	Error message; back to step 2
RC 6	Scenario 6-cannot enroll -- course full	V	V	V	V	I	V	Error message; back to step 3
RC 7	Scenario 6-cannot enroll -- prerequisite not fulfilled	V	V	V	I	V	V	Error message; back to step 4
RC 8	Scenario 6-cannot enroll -- schedule conflict	V	V	V	V	V	I	Error message; back to step 4

Xác định Test Data

- Xác định giá trị cụ thể của từng input của mỗi Test case



26

Test Case ID	Scenario/Condition	Student ID	Password	Courses selected	Prerequisites fulfilled	Course Open	Schedule Open	Expected Result
RC 1	Scenario 1- successful registration	jheumann	abc123	M101> E201 S101	Yes	Yes	Yes	Schedule and confirmation number displayed
RC 2	Scenario 2- unidentified student	jheuman1	N/A	N/A	N/A	N/A	N/A	Error message; back to login screen
RC 3	Scenario 3- valid user quits	jheumann	abc123	N/A	N/A	N/A	N/A	Login screen appears
RC 4	Scenario 4- course registration system unavailable	jheumann	abc123	N/A	N/A	N/A	N/A	Error message; back to step 2
RC 5	Scenario 5- registration closed	jheumann	abc123	N/A	N/A	N/A	N/A	Error message; back to step 2
RC 6	Scenario 6- cannot enroll -- course full	jheumann	abc123	M101 E201 S101	Yes	M101 full	Yes	Error message; back to step 3
RC 7	Scenario 6- cannot enroll -- prerequisite not fulfilled	jheumann	abc123	M101 E201 S101	No for E201	Yes	Yes	Error message; back to step 4
RC 8	Scenario 6- cannot enroll -- schedule conflict	jheumann	abc123	M101 E201 S101	Yes	Yes	E202 and S101 conflict	Error message; back to step 4

Kiểm chứng phần mềm

White-box testing



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

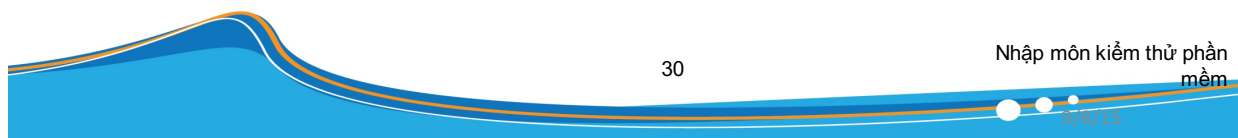


Kiểm thử hộp trắng

- ☐ Structural/Clear box/Glass box testing
- ☐ Thiết kế các trường hợp kiểm thử dựa vào cấu trúc của thủ tục để suy dẫn các trường hợp cần kiểm thử
- ☐ Nguyên tắc
 - ☐ Thực hiện mọi đường dẫn độc lập ít nhất một lần
 - ☐ Thực hiện mọi điều kiện logic trên True/False
 - ☐ Thực hiện mọi vòng lặp tại các biên và trong phạm vi hoạt động
 - ☐ Thực hiện mọi cấu trúc dữ liệu bên trong để đảm bảo tính hợp lệ

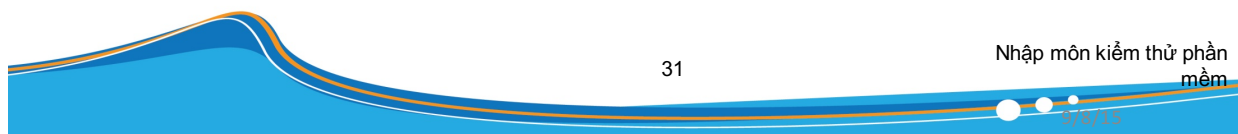
Kiểm thử hộp trắng

- ☐ 2 hướng tiếp cận
 - ☐ Kiểm thử đường dẫn cơ sở (Basic path testing)
 - ☐ Kiểm thử cấu trúc điều kiện (Control structure testing)

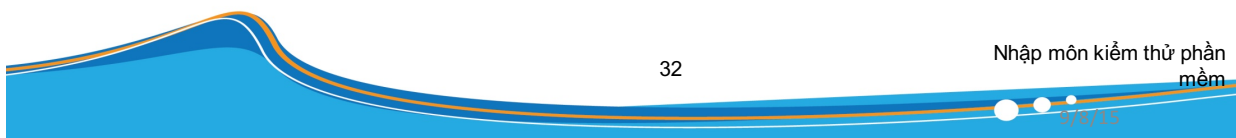


Kiểm thử đường dẫn cơ sở

- ☐ Đảm bảo tất cả đường dẫn độc lập (independent path) đều được kiểm thử
- ☐ Đường dẫn độc lập là đường dẫn đi từ đầu đến cuối chương trình mà không chứa đường dẫn độc lập khác
- ☐ Tập đường dẫn độc lập → tập cơ sở (basic set)



- Các bước thực hiện
 - ▣ Bước 1: Vẽ đồ thị lưu trình (flowgraph)
 - ▣ Bước 2: Xác định độ phức tạp Cyclomat của đồ thị lưu trình
 - ▣ Bước 3: Xác định tập cơ sở các đường dẫn độc lập
 - ▣ Bước 4: Thiết kế test case cho mỗi đường dẫn độc lập

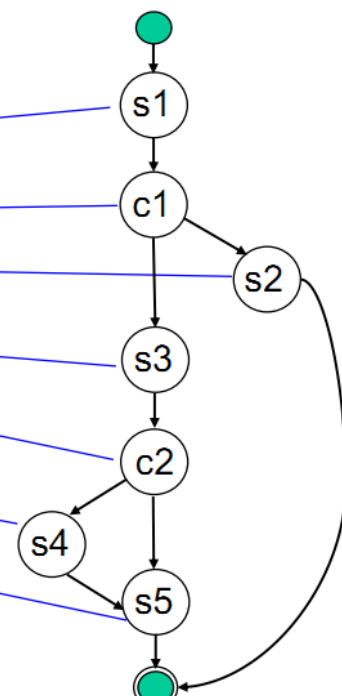


- Bước 1: vẽ đồ thị lưu trình

▪ Thí dụ :

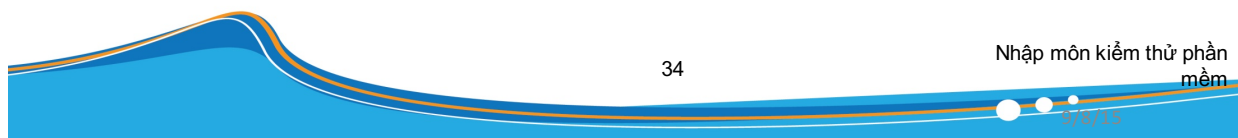
```

1 float foo(int a, int b, int c, int d) {
2   float e;
3   if (a==0)
4     return 0;
5   int x = 0;
6   if ((a==b) || ((c==d) && bug(a)))
7     x = 1;
8   e = 1/x;
9   return e;
10 }
```



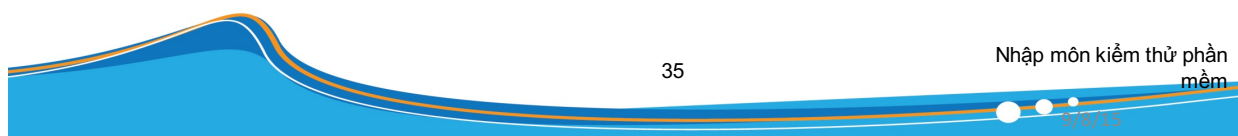
Kiểm thử đường dẫn cơ sở

- Bước 2: Xác định độ phức tạp cyclomat
→ cho biết số lượng đường dẫn độc lập
 - $V(G) = R(\text{số vùng}) = 3$
 - $V(G) = P(\text{số đỉnh điều kiện}) + 1 = 2 + 1 = 3$
 - $V(G) = E(\text{số cạnh}) - N(\text{số đỉnh}) + 2 = 10 - 9 + 2 = 3$



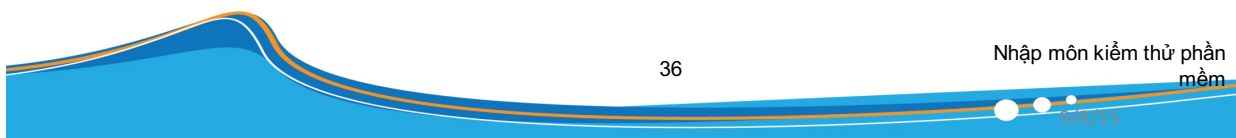
Kiểm thử đường dẫn cơ sở

- Bước 3: tìm tập cơ sở các đường dẫn độc lập
 - Tìm 1 đường dẫn từ đầu đến cuối chương trình
 - Tìm đường dẫn mới có đi qua một cạnh mới mà không trùng với các đường dẫn trước đó
 - Làm cho đến khi đủ số lượng đường dẫn
- Ví dụ:
 - Đường dẫn 1: $S1 \rightarrow C1 \rightarrow S3 \rightarrow C2 \rightarrow S5$
 - Đường dẫn 2: $S1 \rightarrow C1 \rightarrow S2$
 - Đường dẫn 3: $S1 \rightarrow C1 \rightarrow S3 \rightarrow C2 \rightarrow S4 \rightarrow S5$



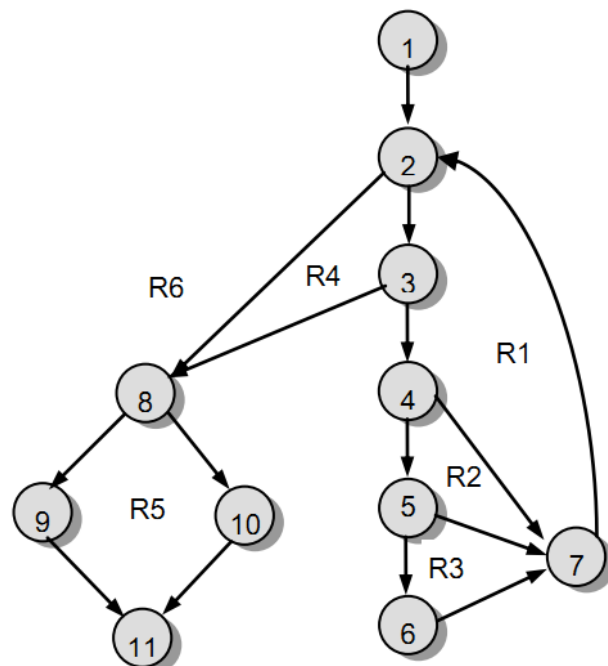
Kiểm thử đường dẫn cơ sở

- Bước 4: thiết kế test case cho từng đường dẫn độc lập
- Ví dụ:
 - ▣ Test case cho đường dẫn 1
 - Đầu vào: ...
 - Đầu ra mong muốn: ...
 - Mục đích: ...



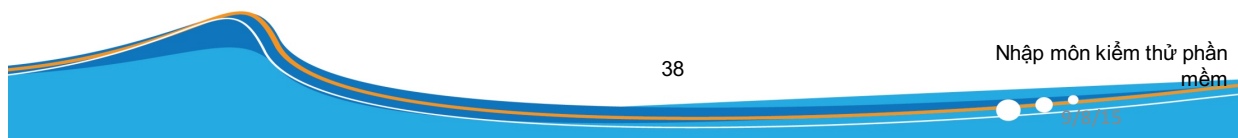
Kiểm thử đường dẫn cơ sở

- Bước 1: đồ thị lưu trình
- ▣ Đỉnh
- ▣ Cung
- ▣ Đỉnh điều kiện
- ▣ Vùng



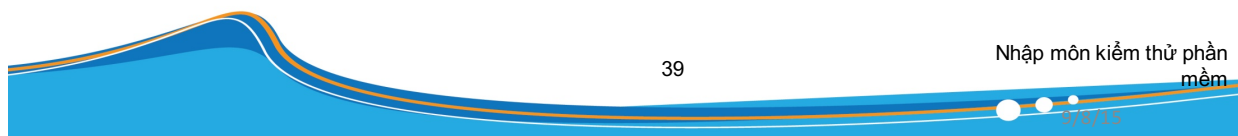
Kiểm thử đường dẫn cơ sở

- Bước 2: Xác định độ phức tạp cyclomat
→ cho biết số lượng đường dẫn độc lập
 - $V(G) = R(\text{số vùng}) = 6$
 - $V(G) = P(\text{số đỉnh điều kiện}) + 1 = 5 + 1 = 6$
 - $V(G) = E(\text{số cạnh}) - N(\text{số đỉnh}) + 2 = 17 - 13 + 2 = 6$



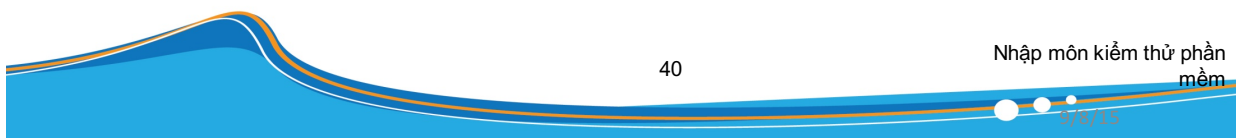
Kiểm thử đường dẫn cơ sở

- Bước 3: tìm tập cơ sở các đường dẫn độc lập
 - Tìm 1 đường dẫn từ đầu đến cuối chương trình
 - Tìm đường dẫn mới có đi qua một cạnh mới mà không trùng với các đường dẫn trước đó
 - Làm cho đến khi đủ số lượng đường dẫn
- Ví dụ:
 - Đường dẫn 1: $1 \rightarrow 2 \rightarrow 8 \rightarrow 9 \rightarrow 11$
 - Đường dẫn 2: $1 \rightarrow 2 \rightarrow 8 \rightarrow 10 \rightarrow 11$
 - Đường dẫn 3: $1 \rightarrow 2 \rightarrow 3 \rightarrow 8 \rightarrow 9 \rightarrow 11$
 - Đường dẫn 4: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 2 \rightarrow \dots$
 - Đường dẫn 5: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 2 \rightarrow \dots$
 - Đường dẫn 6: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 2 \rightarrow \dots$



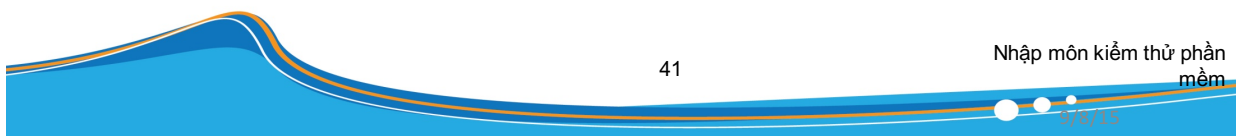
Kiểm thử đường dẫn cơ sở

- ☐ Bước 4: thiết kế test case cho từng đường dẫn độc lập
- ☐ Ví dụ:
 - ☒ Test case cho đường dẫn 1
 - Đầu vào: ...
 - Đầu ra mong muốn: ...
 - Mục đích: ...



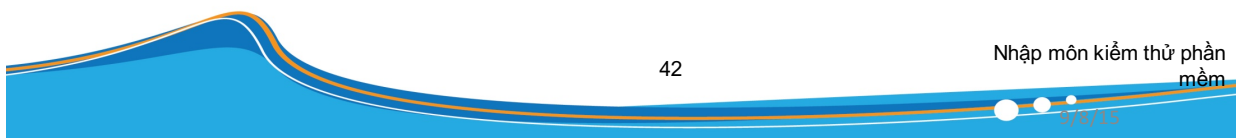
Kiểm thử cấu trúc điều kiện

- ☐ Kiểm thử dòng điều khiển (Control-flow/Coverage testing)
- ☐ Kiểm thử dòng dữ liệu (Data flow testing)
- ☐ Kiểm thử vòng lặp (loop testing)



Kiểm thử dòng điều khiển

- Coverage dùng để đánh giá tính phủ của tập test case
 - Statement coverage
 - Decision/branch coverage
 - Condition coverage
 - Path coverage

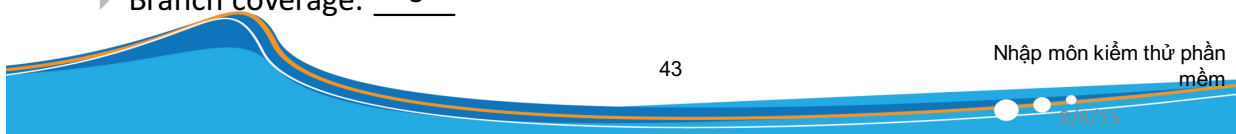
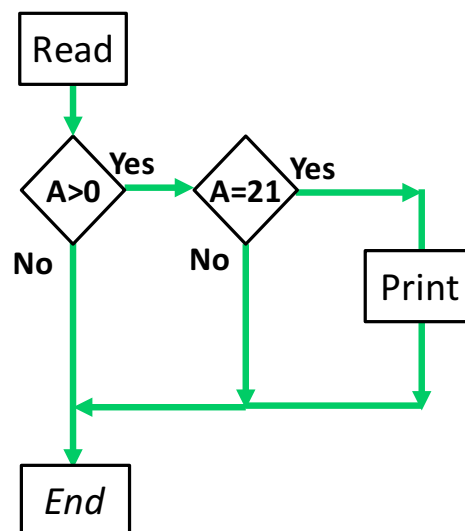


Ví dụ

```

Read A
IF A > 0 THEN
  IF A = 21 THEN
    Print "Key"
  ENDIF
ENDIF
    
```

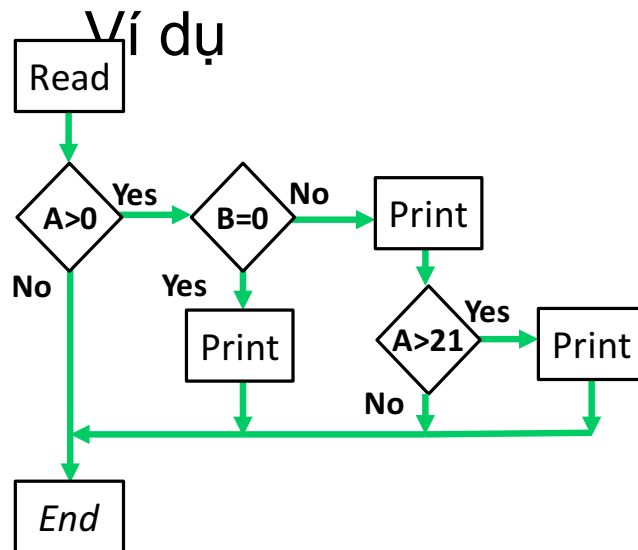
- ▶ Cyclomatic complexity: 3
- ▶ Minimum tests to achieve:
 - ▶ Statement coverage: 1
 - ▶ Branch coverage: 3




```

Read A
Read B
IF A > 0 THEN
  IF B = 0 THEN
    Print "No values"
  ELSE
    Print B
    IF A > 21 THEN
      Print A
    ENDIF
  ENDIF
ENDIF

```



- ▶ Cyclomatic complexity: 4
- ▶ Minimum tests to achieve:
 - ▶ Statement coverage: 2
 - ▶ Branch coverage: 4

44

Nhập môn kiểm thử phần mềm

3/5/15

Kiểm thử dòng dữ liệu

- ☐ Một biến (variable)
 - ☐ Được xác định (define): được gán hay thay đổi giá trị
 - ☐ Được sử dụng (use): tính toán (c-use) hay điều kiện (p-use)
- ☐ Def-use path: đường dẫn từ def đến use của một biến
- ☐ Dữ liệu test được tạo ra để phủ tất cả các def-use

45

Nhập môn kiểm thử phần mềm

3/5/15

Kiểm thử dòng dữ liệu

□ Ví dụ

1	sum = 0	sum, def
2	read (n),	n, def
3	i = 1	i, def
4	while (i <= n)	i, n p-sue
5	read (number)	number, def
6	sum = sum + number	sum, def, sum, number, c-use
7	i = i + 1	i, def, c-use
8	end while	
9	print (sum)	sum, c-use

46

Nhập môn kiểm thử phần mềm

9/9/15

Kiểm thử dòng dữ liệu

□ Ví dụ

Table for sum

pair id	def	use
1	1	6
2	1	9
3	6	6
4	6	9

Table for i

pair id	def	use
1	3	4
2	3	7
3	7	7
4	7	4

phần mềm

9/9/15

Kiểm thử vòng lặp

- ☐ Kiểm tra tính hợp hệ của cấu trúc vòng lặp
- ☐ Bốn dạng vòng lặp:
 - ☐ Lặp đơn (simple loops)
 - ☐ Lặp móc nối (concatenated loops)
 - ☐ Lặp lồng nhau (nested loops)
 - ☐ Lặp không cấu trúc (unstructured loops)

