



TRƯỜNG CAO ĐẲNG KỸ THUẬT CAO THẮNG

KHOA ĐIỆN TỬ - TIN HỌC
BỘ MÔN TIN HỌC



KIỂM THỬ PHẦN MỀM



Kiểm chứng phần mềm State transition testing & Use case testing

GV: Nguyễn Thị Ngọc



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 chuyển trạng thái (state transition 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.

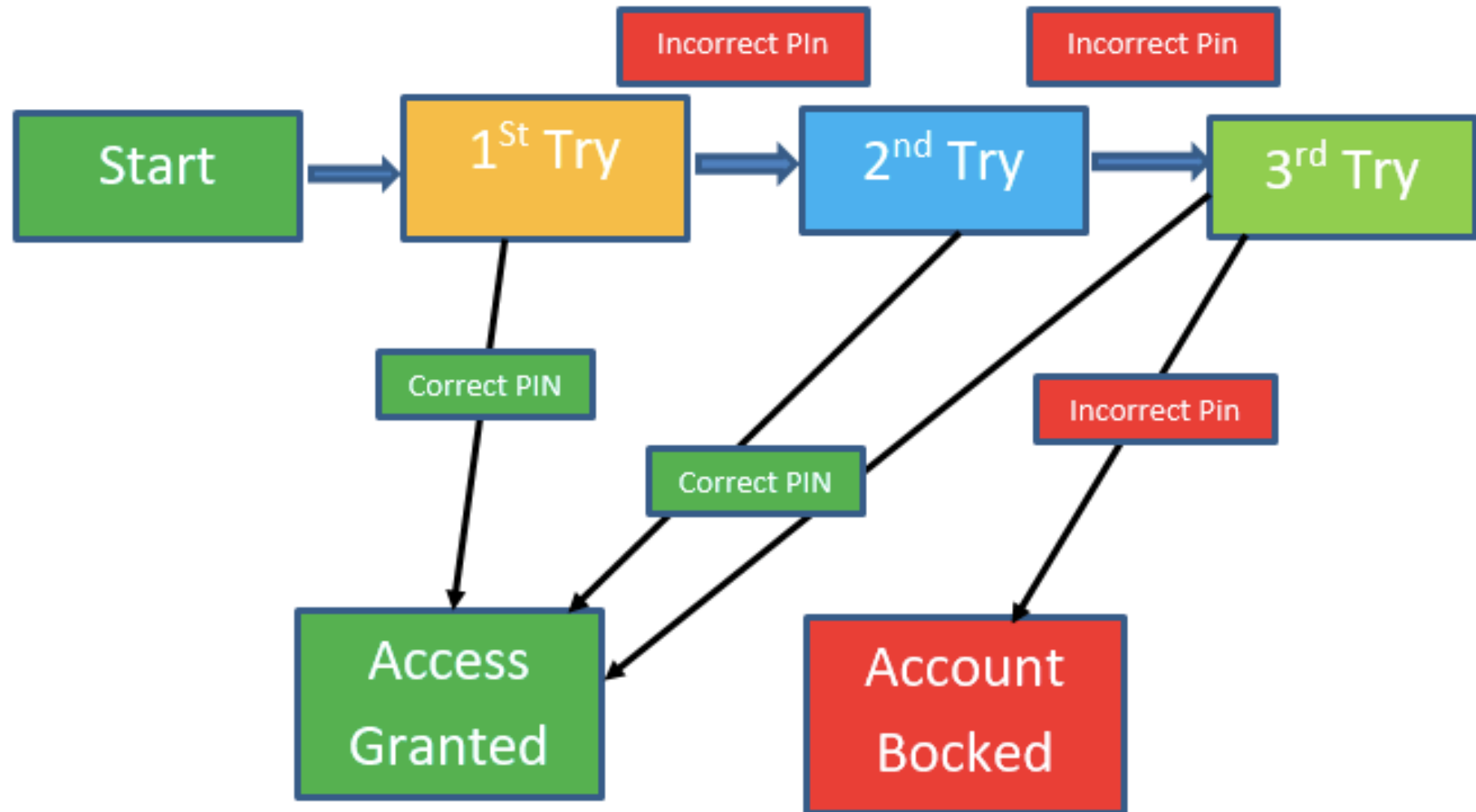
State transition diagram

- ❑ **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...)

Ví dụ

- ❑ Hãy xem chức năng của hệ thống ATM: nếu người dùng nhập mật khẩu không hợp lệ ba lần thì tài khoản sẽ bị khóa.
- ❑ Trong hệ thống này, nếu người dùng nhập mật khẩu hợp lệ, người dùng sẽ đăng nhập thành công. Nếu người dùng nhập mật khẩu không hợp lệ trong lần thử đầu tiên hoặc lần thứ hai, người dùng sẽ được yêu cầu nhập lại mật khẩu, nếu người dùng nhập mật khẩu không đúng lần thứ 3, tài khoản sẽ bị chặn.

Sơ đồ chuyển trạng thái



State transition table

- ❑ Liệt kê tất cả các tổ hợp bước chuyển giữa các trạng thái, không nhất thiết là bước chuyển hợp lệ
 - Ưu điểm: Phát hiện các đường đi không hợp lệ giữa các trạng thái
 - Nhược điểm: Bùng nổ tổ hợp trong trường hợp có nhiều trạng thái và sự kiện.

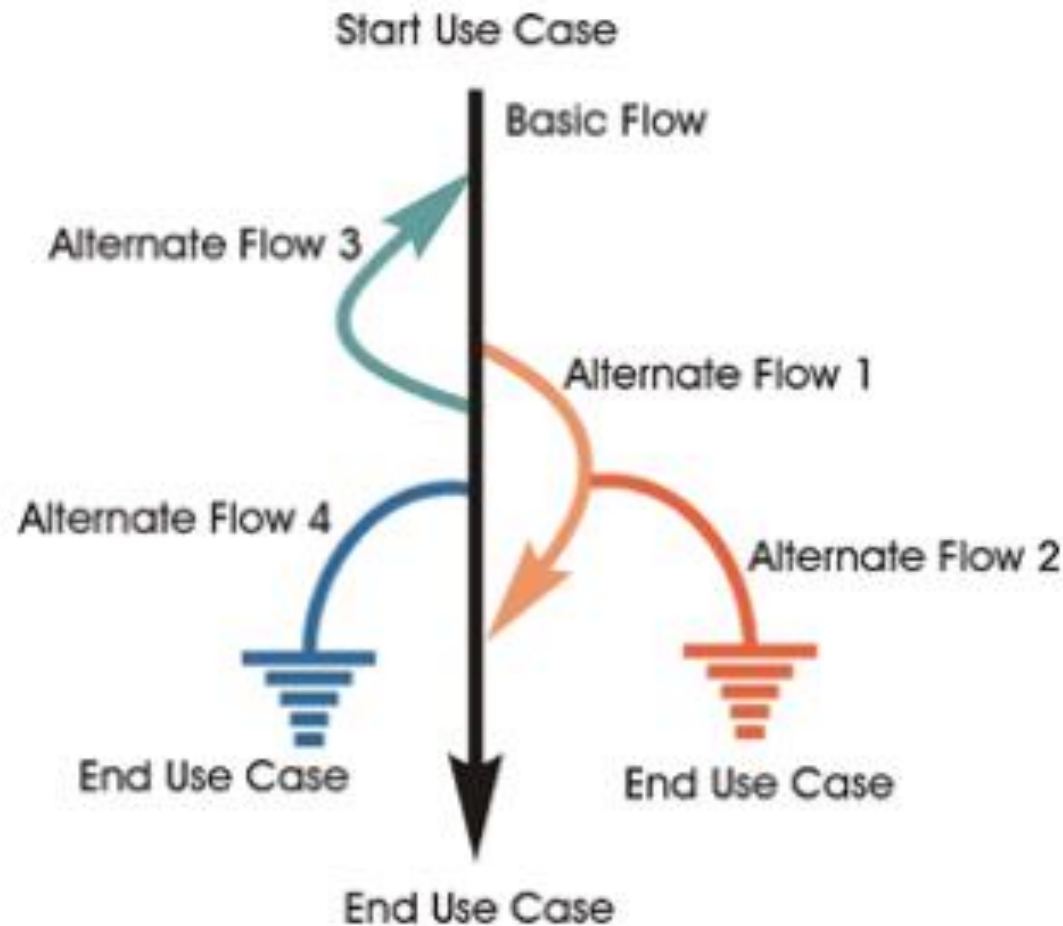
Bài tập

- ❑ 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.

Use case testing

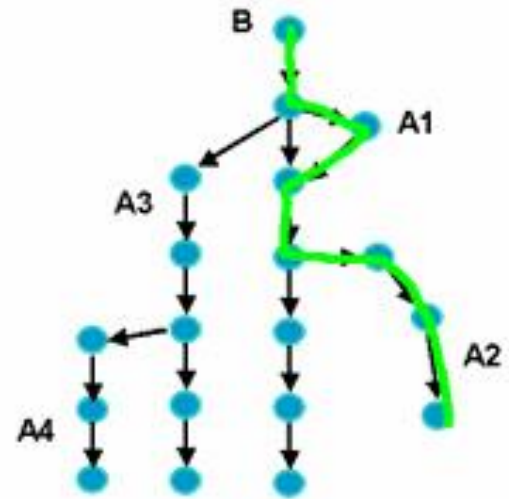
- ❑ Use case: mô tả một chuỗi các hành động của người dùng hoặc hệ thống khác (Actor) tương tác với hệ thống
 - Brief description
 - Flow of events
 - ▶ basic follow
 - ▶ alternative flow 1
 - ▶ alternative flow 2
 - Special requirements
 - Preconditions
 - Post-conditions
 - Activity diagram

Flow of events



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

Xác định scenario



Thiết kế TC từ use case

- ❑ Xác định tất cả scenario từ use case
- ❑ Với mỗi scenario xác định ít nhất 1 TC và tập điều kiện để TC có thể thực thi
- ❑ Với mỗi TC xác định dữ liệu kiểm thử (Test data) cho việc kiểm thử.

Vd: Đăng ký học phần

❑ Basic flows

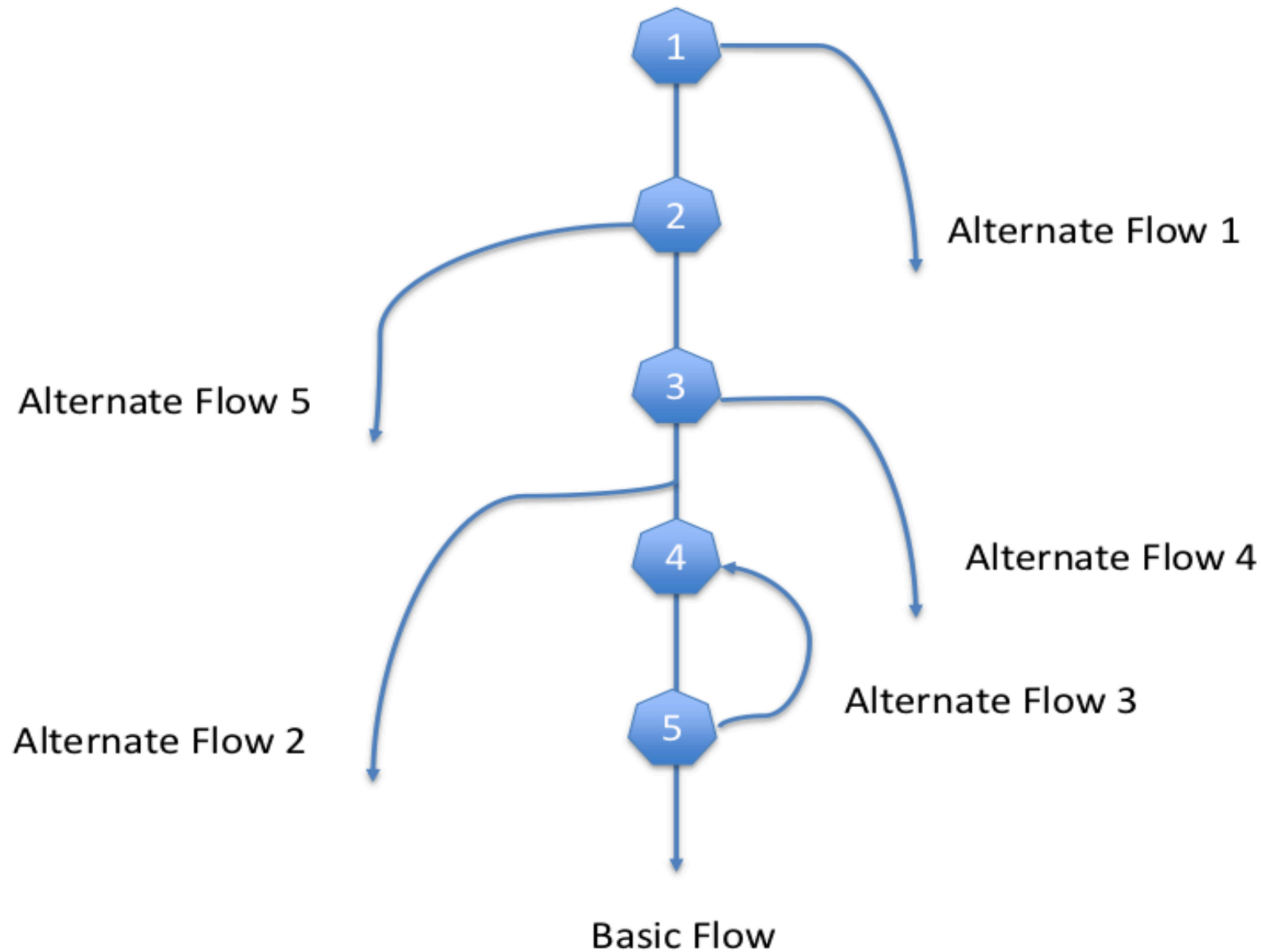
1. Đăng nhập
2. Chọn "Tạo thời khoá biểu"
3. Lấy thông tin khoá học
4. Chọn lớp học
5. Chọn lịch học
6. Hiển thị thời khoá biểu cuối cùng.

Vd: Đăng ký học phần

❑ Alternate flow

1. Sinh viên không tồn tại
2. Thoát khỏi hệ thống
3. Vi phạm điều kiện môn học tiên quyết, lớp học full, đụng lịch học các môn khác
4. Lớp học không có sẵn trên hệ thống
5. Lớp học đã đóng không cho đăng ký

Activity diagram



Xác định scenario

Thảo luận tại lớp.

Xác định test case

- ❑ Xác định input cho từng bước của use case
- ❑ Xác định miền giá trị cho từng input
- ❑ Kết hợp miền giá trị của input cho từng TC

Thảo luận

Kiểm chứng phần mềm White-box testing

GV: Nguyễn Thị Ngọc

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 các đường dẫn độc lập 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 \rightarrow tập cơ sở.

Kiểm thử đường dẫn cơ sở

❑ Các bước thực hiện

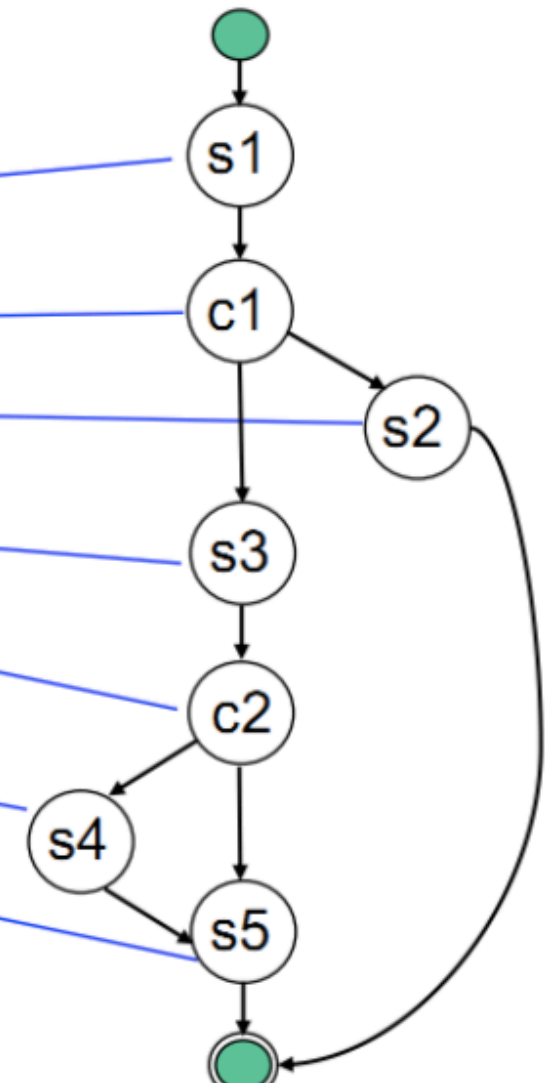
- Bước 1: Vẽ đồ thị lưu trình (flowgraph)
- Bước 2: Xác định độ phức tạp Cyclomat của đồ thị lưu trình
- Bước 3: Xác định tập cơ sở các đường dẫn độc lập
- Bước 4: Thiết kế test case cho mỗi đường dẫn độc lập.

Kiểm thử đường dẫn cơ sở

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

Ký hiệu: $V(G)$ và được tính bằng 1 trong 3 công thức sau:

$$V(G) = R(\text{số vùng/miền}) = 3$$

$$V(G) = P(\text{số đỉnh điều kiện}) + 1 = 2 + 1 = 3$$

$$V(G) = E(\text{số cạnh}) - N(\text{số đỉnh}) + 2 = 10 - 9 + 2 = 3$$

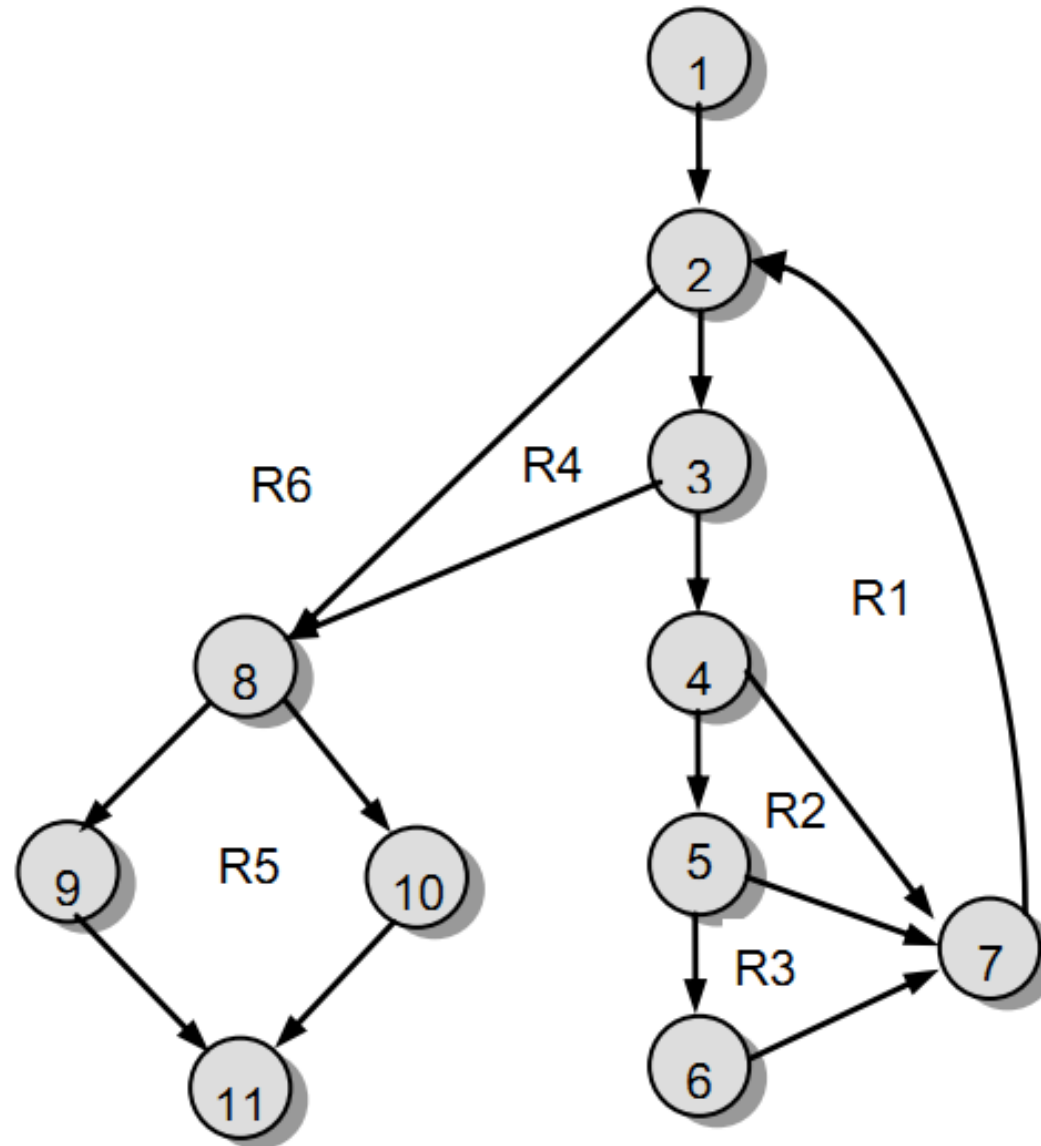
Kiểm thử đường dẫn cơ sở

- ❑ 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 TC cho từng đường dẫn độc lập
- ❑ Ví dụ:
 - TC cho đường dẫn 1
 - ▶ Đầu vào....
 - ▶ Đầu ra mong muốn....
 - ▶ Mục đích....

Bài tập



❑ Bước 1: Đồ thị lưu trình

- Đỉnh
- Cung
- Đỉnh điều kiện
- Vùng

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

❑ Bước 3: Tìm tập cơ sở các đường dẫn độc lập

- Tìm 1 đường từ đầu tới cuối chương trình
- Tìm đường dẫn mới có đi qua một cạnh mới mà không trùng với các đường dẫn trước đó
- Làm cho tới khi đủ số lượng đường dẫn

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 cấu trúc điều kiện

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

Ví dụ

Ví dụ

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

Ví dụ

Kiểm thử vòng lặp

- ❑ Kiểm tra tính hợp lệ của cấu trúc vòng lặp
- ❑ Bốn dạng vòng lặp
 - Lặp đơn
 - Lặp móc nối
 - Lặp lồng nhau
 - Lặp không cấu trúc