

4.1 Tổng quát về kiểm thử dòng dữ liệu

Mục tiêu của chương trình là xử lý dữ liệu. Dữ liệu của chương trình là tập nhiều biến độc lập. Phương pháp kiểm thử dòng dữ liệu sẽ kiểm thử đời sống của từng biến dữ liệu có "tốt lành" trong từng luồng thi hành của chương trình.

Phương pháp kiểm thử dòng dữ liệu là 1 công cụ mạnh để phát hiện việc dùng không hợp lý các biến do lỗi coding phần mềm gây ra :

- Phát biểu gán hay nhập dữ liệu vào biến không đúng.
- Thiếu định nghĩa biến trước khi dùng
- Tiên đề sai (do thi hành sai luồng thi hành).
- ...

Mỗi biến nên có chu kỳ sống tốt lành thông qua trình tự 3 bước : được tạo ra, được dùng và được xóa đi.

Chỉ có những lệnh nằm trong tầm vực truy xuất biến mới có thể truy xuất/xử lý được biến. Tầm vực truy xuất biến là tập các lệnh được phép truy xuất biến đó.

Thường các ngôn ngữ lập trình cho phép định nghĩa tầm vực cho mỗi biến thuộc 1 trong 3 mức chính yếu : toàn cục, cục bộ trong từng module, cục bộ trong từng hàm chức năng.

```
int x, y;  
void func1() { //thân hàm  
    int x;    // định nghĩa biến x mới cục bộ trong hàm  
    ...;      // mỗi lần truy xuất x là x cục bộ trong hàm  
    {        // khối lệnh bên trong bắt đầu  
        int y; // định nghĩa biến y mới cục bộ trong lệnh phức hợp
```

```

    ...;    //mỗi lần truy xuất y là y cục bộ trong lệnh phức hợp
}    // y bên trong tự động bị xóa
    ...;    //truy xuất y ngoài cùng, x cục bộ trong hàm
} // x cục bộ trong hàm bị xóa tự động

```

4.2 Phân tích đời sống của 1 biến

Các lệnh truy xuất 1 biến thông qua 1 trong 3 hành động sau :

- d : định nghĩa biến, gán giá trị xác định cho biến (nhập dữ liệu vào biến cũng là hoạt động gán trị cho biến).
- u : tham khảo trị của biến (thường thông qua biểu thức).
- k : hủy (xóa bỏ) biến đi.

Như vậy nếu ký hiệu ~ là miêu tả trạng thái mà ở đó biến chưa tồn tại, ta có 3 khả năng xử lý đầu tiên trên 1 biến :

- ~d : biến chưa tồn tại rồi được định nghĩa với giá trị xác định.
- ~u : biến chưa tồn tại rồi được dùng ngay (trị nào ?)
- ~k : biến chưa tồn tại rồi bị hủy (lạ lùng).

3 hoạt động xử lý biến khác nhau kết hợp lại tạo ra 9 cặp đôi hoạt động xử lý biến theo thứ tự :

- dd : biến được định nghĩa rồi định nghĩa nữa : hơi lạ, có thể đúng và chấp nhận được, nhưng cũng có thể có lỗi lập trình.
- du : biến được định nghĩa rồi được dùng : trình tự đúng và bình thường.
- dk : biến được định nghĩa rồi bị xóa bỏ : hơi lạ, có thể đúng và chấp nhận được, nhưng cũng có thể có lỗi lập trình.
- ud : biến được dùng rồi định nghĩa giá trị mới : hợp lý.
- uu : biến được dùng rồi dùng tiếp : hợp lý.
- uk : biến được dùng rồi bị hủy : hợp lý.

- kd : biến bị xóa bỏ rồi được định nghĩa lại : chấp nhận được.
- ku : biến bị xóa bỏ rồi được dùng : đây luôn là lỗi.
- kk : biến bị xóa bỏ rồi bị xóa nữa : có lẽ là lỗi lập trình.

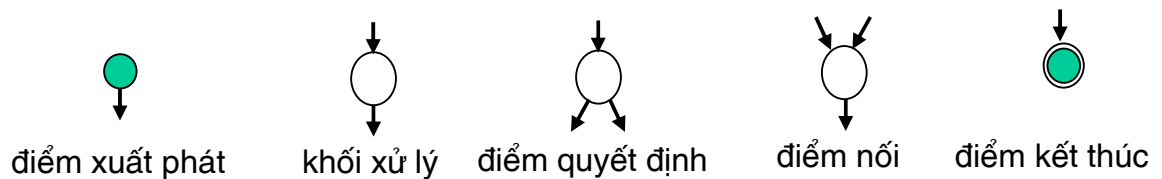
4.3 Đồ thị dòng dữ liệu

Là một trong nhiều phương pháp miêu tả các kịch bản đời sống khác nhau của các biến.

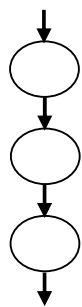
Qui trình xây dựng đồ thị dòng dữ liệu dựa trên qui trình xây dựng đồ thị dòng điều khiển của TPPM cần kiểm thử.

Gồm 2 loại thành phần : các nút và các cung nối kết giữa chúng.

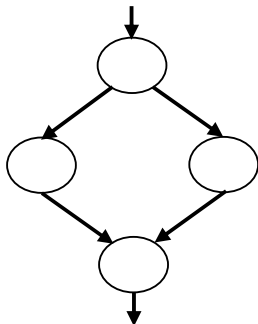
Các loại nút trong đồ thị dòng điều khiển :



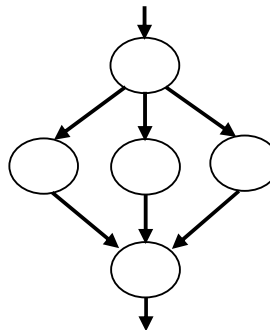
Miêu tả các cấu trúc điều khiển phổ dụng :



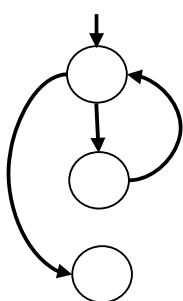
tuần tự



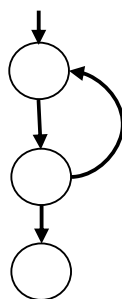
If



switch



while c do...



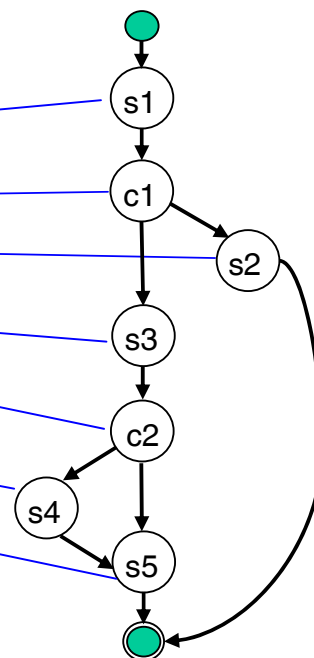
do ... while c

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.}

```



Độ phức tạp Cyclomatic C

Ta cũng dùng độ phức tạp Cyclomatic $C = V(G)$ của đồ thị dòng điều khiển của TPPM cần kiểm thử để xác định số đường thi hành tuyến tính độc lập của TPPM cần kiểm thử.

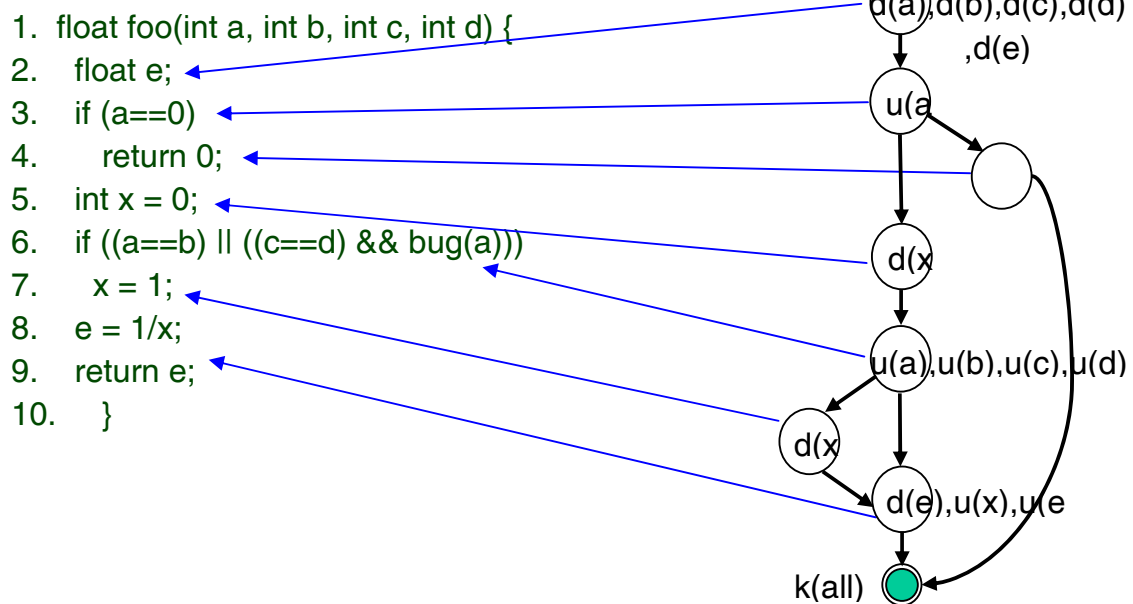
Mục tiêu của kiểm thử dòng dữ liệu là chọn lựa được đúng C đường thi hành tuyến tính độc lập của TPPM cần kiểm thử rồi kiểm thử đời sống của từng biến trên từng đường thi hành này xem có lỗi gì không.

4.4 Qui trình kiểm thử dòng dữ liệu

Qui trình kiểm thử dòng dữ liệu của 1 TPPM gồm các bước công việc sau :

- Từ TPPM cần kiểm thử, xây dựng đồ thị dòng điều khiển tương ứng, rồi chuyển thành đồ thị dòng điều khiển nhị phân, rồi chuyển thành đồ thị dòng dữ liệu.
- Tính độ phức tạp Cyclomatic của đồ thị ($C = P + 1$).
- Xác định C đường thi hành tuyến tính độc lập cơ bản cần kiểm thử (theo thuật giải chi tiết ở chương 3).
- Lặp kiểm thử đời sống từng biến dữ liệu :
 - mỗi biến có thể có tối đa C kịch bản đời sống khác nhau.
 - trong từng kịch bản đời sống của 1 biến, kiểm thử xem có tồn tại cặp đời hoạt động không bình thường nào không ? Nếu có hãy ghi nhận để lập báo cáo kết quả và phản hồi cho những người có liên quan.

4.5 Thí dụ

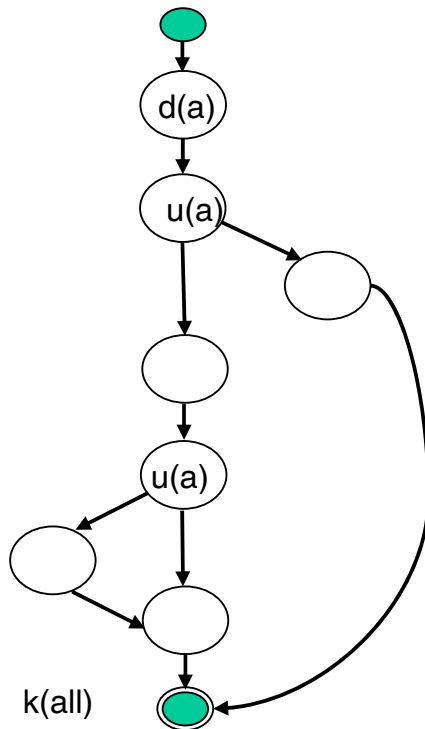


Đồ thị ở slide trước có 2 nút quyết định nhị phân nên có độ phức tạp $C = 2 + 1 = 3$.

Nó có 4 biến đầu vào (tham số) và 2 biến cục bộ.

Hãy lập kiểm thử đời sống từng biến a, b, c, d, e, x.

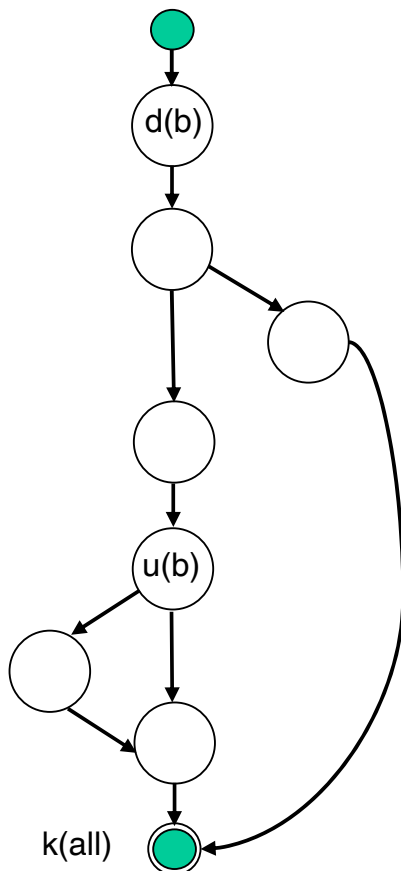
Kiểm thử đời sống biến a



- Kịch bản 1 : $\sim \text{duuk}$
- Kịch bản 2 : $\sim \text{duuk}$ (giống kịch bản 1).
- Kịch bản 3 : $\sim \text{duk}$

Cả 3 kịch bản trên đều không chứa cặp đôi hoạt động nào bất thường cả.

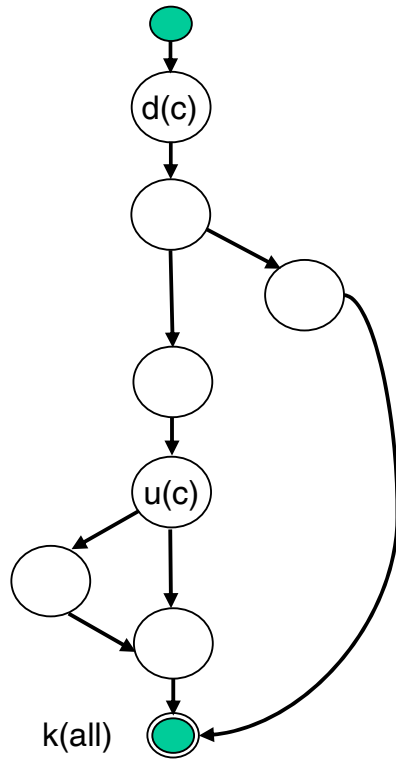
Kiểm thử đời sống biến b



- Kịch bản 1 : $\sim \text{duk}$
- Kịch bản 2 : $\sim \text{duk}$ (giống kịch bản 1).
- Kịch bản 3 : $\sim \text{dk}$

Cả 3 kịch bản trên đều không chứa cặp đôi hoạt động nào bất thường cả.

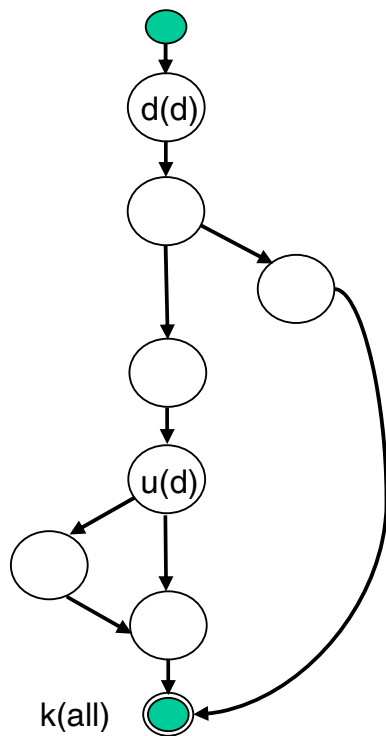
Kiểm thử đời sống biến c



- Kịch bản 1 : $\sim duk$
- Kịch bản 2 : $\sim duk$ (giống kịch bản 1).
- Kịch bản 3 : $\sim dk$

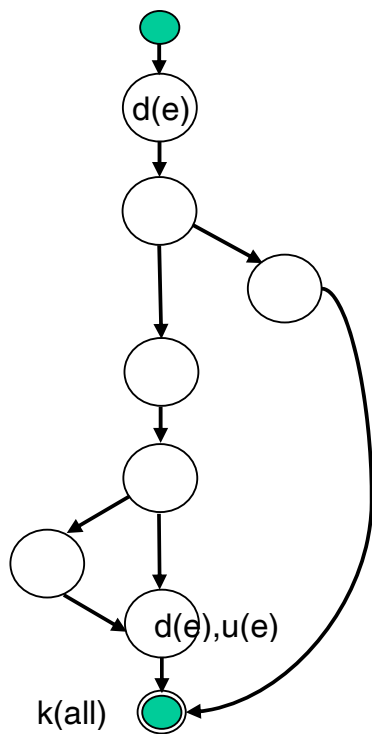
Cả 3 kịch bản trên đều không chứa cặp đôi hoạt động nào bất thường cả.

Kiểm thử đời sống biến d



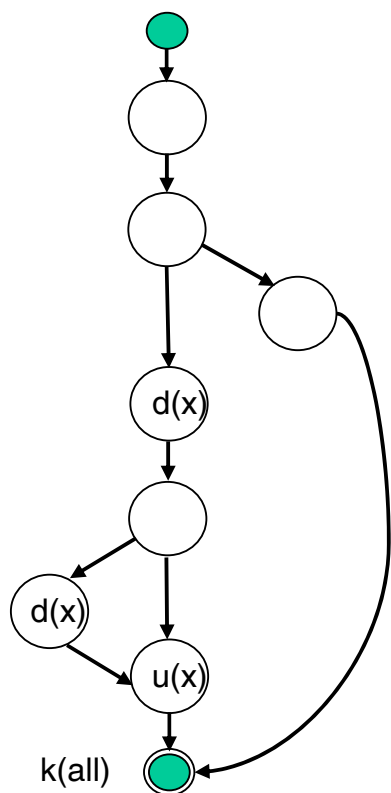
- Kịch bản 1 : $\sim duk$
 - Kịch bản 2 : $\sim duk$ (giống kịch bản 1).
 - Kịch bản 3 : $\sim dk$
- Cả 3 kịch bản trên đều không chứa cặp đôi hoạt động nào bất thường cả.

Kiểm thử đời sống biến e



- Kịch bản 1 : $\sim dduk$
 - Kịch bản 2 : $\sim dduk$ (giống kịch bản 1).
 - Kịch bản 3 : $\sim dk$
- Trong 3 kịch bản trên, kịch bản 1 & 2 có chứa cặp đôi *dd* bất thường nên cần tập trung chú ý kiểm tra xem có phải là lỗi không.

Kiểm thử đời sống biến x



- Kịch bản 1 : $\sim dduk$
- Kịch bản 2 : $\sim duk$
- Kịch bản 3 : \sim

Trong 3 kịch bản trên, chỉ có kịch bản 1 có chứa cặp đôi dd bất thường nên cần tập trung chú ý kiểm tra xem có phải là lỗi không.

4.6 Kết chương

Chương này đã giới thiệu tiếp 1 kỹ thuật khác để kiểm thử hộp trắng TPPM, đó là kỹ thuật kiểm thử dòng dữ liệu.

Chúng ta đã phân tích đời sống của biến dữ liệu, các cặp đôi hoạt động trên biến được gọi là hợp lệ hay nghi ngờ có lỗi hay tệ hơn là chắn chắn gây lỗi.

Chương này cũng đã giới thiệu 1 thí dụ cụ thể về quy trình kiểm thử dòng dữ liệu trên 1 TPPM.