

KIỂM THỬ VÀ ĐẢM BẢO  
CHẤT LƯỢNG PHẦN MỀM

CÁC KỸ THUẬT KIỂM THỬ  
(tiếp)

*Nguyễn Thị Phương Dung*



KIỂM THỬ DÒNG DỮ LIỆU

# Kiểm thử dòng dữ liệu

- Là kiểm chú trọng vào các biến trong đơn vị chương trình.
- Kiểm thử dòng dữ liệu được chia làm 2 loại:
  - Kiểm thử dựa trên việc gán và sử dụng các biến
  - Kiểm thử dựa trên lát cắt

# Kiểm thử dựa trên việc gán và sử dụng các biến

- Mỗi chương trình/đơn vị chương trình là chuỗi các hoạt động gồm nhận các giá trị đầu vào, thực hiện các tính toán, gán giá trị mới cho các biến (các biến cục bộ và toàn cục) và cuối cùng là trả lại kết quả đầu ra như mong muốn.
- Khi một biến được khai báo và gán giá trị, nó phải được sử dụng ở đâu đó trong chương trình.
- Việc sử dụng biến này có thể trong các câu lệnh tính toán hoặc trong các biểu thức điều kiện.
- Nếu biến này không được sử dụng ở các câu lệnh tiếp theo thì việc khai báo biến này là không cần thiết.

# Các vấn đề phổ biến về dòng dữ liệu

## Gán giá trị rồi gán tiếp giá trị (Loại 1):

Ví dụ, hai câu lệnh tuần tự  $x = f1(y); x = f2(z);$

- Xem xét các tình huống sau:
  - Khi câu lệnh thứ hai được thực hiện, giá trị của biến  $x$  được gán và câu lệnh đầu không có ý nghĩa.
  - Lập trình viên có thể có nhầm lẫn ở câu lệnh đầu. Câu lệnh này có thể là gán giá trị cho một biến khác như là  $w = f1(y)$ .
  - Có thể có nhầm lẫn ở câu lệnh thứ hai. Lập trình viên định gán giá trị cho một biến khác như là  $w = f2(z)$ .
  - Một hoặc một số câu lệnh giữa hai câu lệnh này bị thiếu. Ví dụ như câu lệnh  $w = f3(x)$ .

# Các vấn đề phổ biến về dòng dữ liệu

## **Chưa gán giá trị nhưng được sử dụng (Loại 2):**

Ví dụ, ba câu lệnh tuần tự:  $y = f(x_1)$ ; `int z`;  $x = y + z$ ;

Trong trường hợp này, biến  $z$  chưa được gán giá trị khởi tạo nhưng đã được sử dụng trong câu lệnh để tính giá trị của biến  $x$  ( $x = y + z$ ).

Vấn đề này có thể được lý giải theo các tình huống sau:

- Lập trình viên có thể bỏ quên lệnh gán giá trị cho biến  $z$  trước câu lệnh tính toán giá trị cho biến  $x$ .
- Có thể có sự nhầm lẫn giữa biến  $z$  với một biến đã được khai báo và gán giá trị.

# Các vấn đề phổ biến về dòng dữ liệu

## **Đã được khai báo và gán giá trị nhưng không được sử dụng (Loại 3):**

Nếu một biến đã được khai báo và gán giá trị nhưng không hề được sử dụng. Các tình huống sau có thể được sử dụng để lý giải cho vấn đề này:

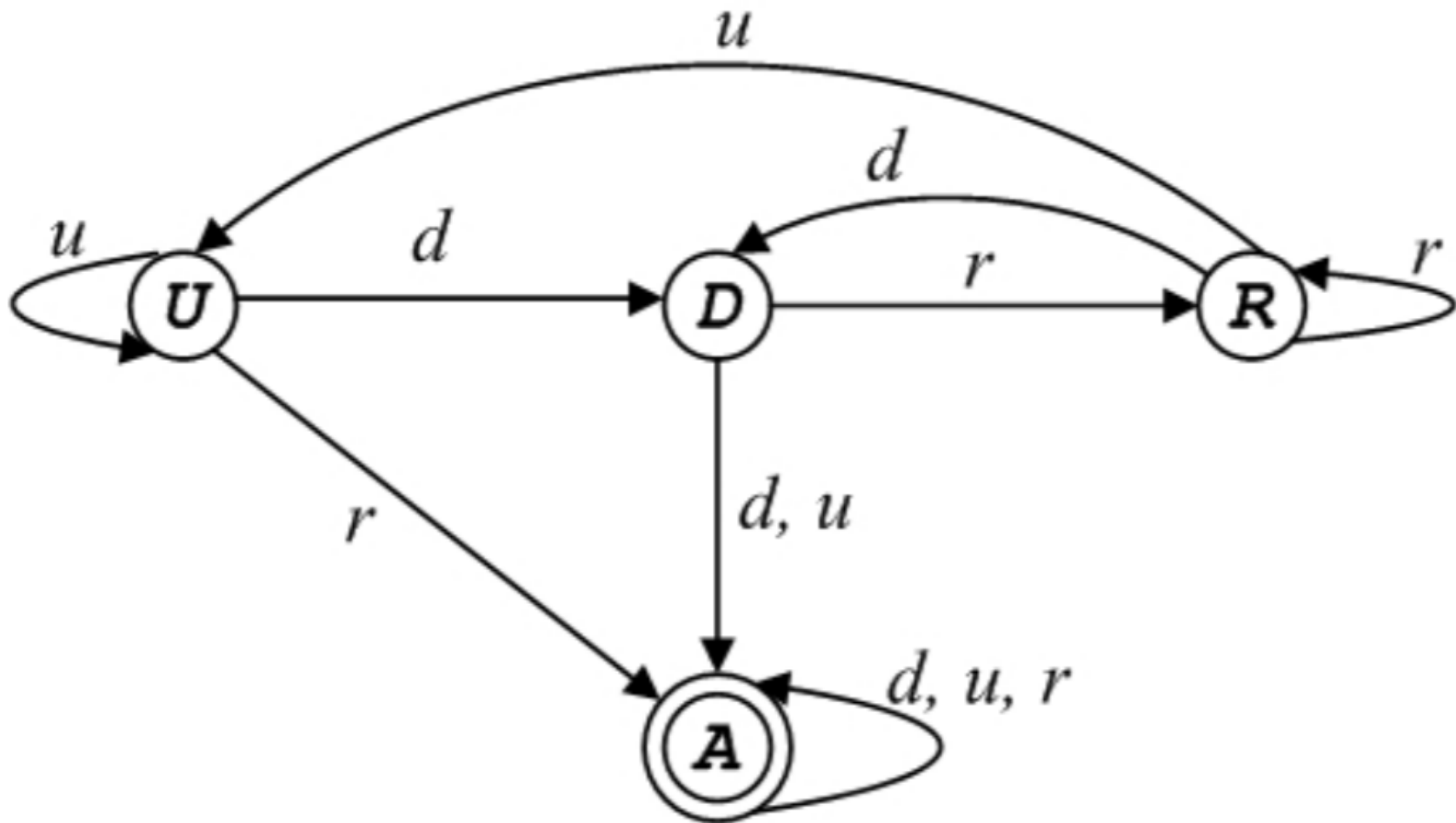
- Có sự nhầm lẫn giữa biến này và một số biến khác được sử dụng trong chương trình. Trong thiết kế, biến này được sử dụng nhưng nó đã bị thay thế (do nhầm lẫn) bởi một biến khác.
- Biến này thực sự không được sử dụng trong chương trình. Lúc đầu lập trình viên định sử dụng nó như là một biến tạm thời hoặc biến trung gian nhưng sau đó lại không cần dùng. Lập trình viên này đã quên xóa các câu lệnh khai báo và gán giá trị cho biến này.

# Sử dụng sơ đồ chuyển trạng thái để xác định những bất thường trong việc sử dụng các biến dữ liệu

- Các trạng thái, gồm:
  - U: biến chưa được gán giá trị
  - D: biến đã được gán giá trị nhưng chưa được sử dụng
  - R: biến đã được sử dụng
  - A: trạng thái lỗi
- Các hành động gồm:
  - d: biến được gán giá trị
  - r: biến được sử dụng
  - u: biến chưa được gán giá trị hoặc được khai báo lại và chưa được gán giá trị



# Ví dụ



# Kiểm thử dòng dữ liệu động

- Cần chắc chắn rằng một biến phải được gán đúng giá trị, tức là phải xác định được một đường đi của biến từ một điểm bắt đầu nơi nó được định nghĩa đến điểm mà biến đó được sử dụng.
- Nếu chưa có các ca kiểm thử để kiểm tra đường đi này, không thể tự tin khẳng định là biến này đã được gán giá trị đúng.
- Ngay cả khi gán đúng giá trị cho biến thì các giá trị được sinh ra chưa chắc đã chính xác do tính toán hoặc các biểu thức điều kiện sai (biến được sử dụng sai).

# Các bước để kiểm thử dòng dữ liệu động

- Xây dựng đồ thị dòng dữ liệu của chương trình/đơn vị chương trình
- Chọn một hoặc một số tiêu chí kiểm thử dòng dữ liệu
- Xác định các đường dẫn chương trình phù hợp với tiêu chí kiểm thử đã chọn
- Lấy ra các biểu thức điều kiện từ tập các đường đi, thực hiện giải các biểu thức điều kiện để có được các giá trị đầu vào cho các ca kiểm thử tương ứng với các đường đi này và tính toán giá trị đầu ra mong đợi của mỗi ca kiểm thử
- Thực hiện các ca kiểm thử để xác định các lỗi (có thể có) của chương trình.
- Sửa các lỗi (nếu có) và thực hiện lại tất cả các ca kiểm thử trong trường hợp bước trên phát hiện ra lỗi.

# Xây dựng đồ thị dòng dữ liệu

- Xét ví dụ:

```
int VarTypes(int x, int y){  
1.   int i;  
2.   int *iptr;  
3.   i = x;  
4.   iptr = malloc(sizeof(int));  
5.   *iptr = i + x;  
6.   if (*iptr > y)  
7.       return (x);  
8.   else {  
9.       iptr = malloc(sizeof(int));  
10.      *iptr = x + y;  
11.      return(*iptr);  
12.  }//end if  
    }//the end
```

# Các khái niệm liên quan đến việc định nghĩa và sử dụng biến

- Định nghĩa 1. (Định nghĩa của một biến.) Một câu lệnh thực hiện việc gán giá trị cho một biến được gọi là câu lệnh định nghĩa của biến đó (ký hiệu là *def*).
- Như trong hàm VarTypes: câu lệnh  $i = x$  (dòng 3) là một ví dụ về việc định nghĩa của biến  $i$ .

# Các khái niệm liên quan đến việc định nghĩa và sử dụng biến

- Định nghĩa 2. (Chưa định nghĩa một biến.) Một câu lệnh khai báo một biến nhưng chưa thực hiện việc gán giá trị được gọi là *undef* với biến đó.
- Trong hàm VarTypes:
  - Câu lệnh `iptr = malloc(sizeof(int));` (dòng 4) là một ví dụ về việc chưa định nghĩa biến `iptr`.
  - Câu lệnh tiếp theo (`*iptr = i + x;` (dòng 5)) là một định nghĩa của biến này.

## Các khái niệm liên quan đến việc định nghĩa và sử dụng biến

- Định nghĩa 3. (Biến được sử dụng.) Một câu lệnh sử dụng một biến (để tính toán hoặc để kiểm tra các điều kiện) được gọi là *use* của biến đó. use được chia làm 2 loại c-use và p-use.

# Các khái niệm liên quan đến việc định nghĩa và sử dụng biến

- Định nghĩa 4. (Biến được sử dụng để tính toán.) Một câu lệnh sử dụng một biến để tính toán giá trị của một biến khác được gọi là *c-use* với biến đó.
- Ví dụ, trong hàm VarTypes, câu lệnh `*iptr = i + x;` (dòng 5) là *c-use* ứng với biến `i` và biến `x`.



# Các khái niệm liên quan đến việc định nghĩa và sử dụng biến

- Định nghĩa 5. (Biến được sử dụng để kiểm tra các điều kiện.) Một câu lệnh sử dụng một biến trong các biểu thức điều kiện (câu lệnh rẽ nhánh, lặp, ...) được gọi là *p-use* với biến đó.
- Ví dụ, trong hàm VarTypes, câu lệnh `if (*iptr > y) ...` (dòng 6) là *p-use* ứng với biến **iptr** và biến **y**.

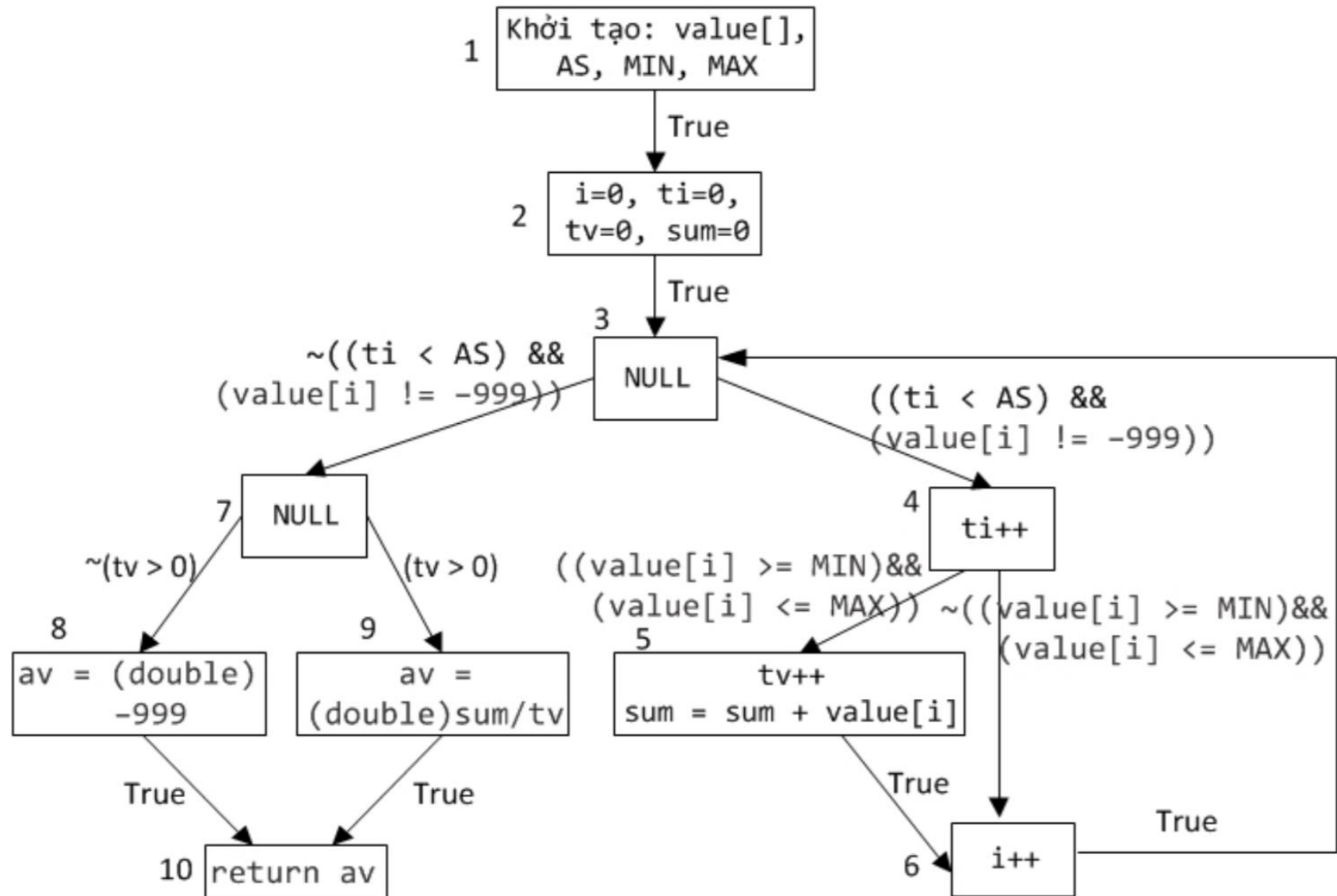
## Các khái niệm liên quan đến việc định nghĩa và sử dụng biến

- Định nghĩa 6. (Đồ thị dòng dữ liệu.) Đồ thị dòng dữ liệu của một chương trình/đơn vị chương trình là một đồ thị có hướng  $G = \langle N, E \rangle$ , với:
  - $N$  là tập các đỉnh tương ứng với các câu lệnh *def* hoặc *c-use* của các biến được sử dụng trong đơn vị chương trình. Đồ thị  $G$  có hai đỉnh đặc biệt là đỉnh bắt đầu (tương ứng với lệnh *def* của các biến tham số) và đỉnh kết thúc đơn vị chương trình.
  - $E$  là tập các cạnh tương ứng với các câu lệnh *p-use* của các biến.

# Xây dựng đồ thị dòng dữ liệu cho hàm sau

```
double ReturnAverage(int value[], int AS, int MIN, int MAX){
    int i, ti, tv, sum;
    double av;
    i = 0; ti = 0; tv = 0; sum = 0;
    while (ti < AS && value[i] != -999) {
        ti++;
        if (value[i] >= MIN && value[i] <= MAX) {
            tv++;
            sum = sum + value[i];
        }
        i++;
    } //end while
    if (tv > 0)
        av = (double)sum/tv;
    else
        av = (double) -999;
    return (av);
} //the end
```

# Đồ thị dòng dữ liệu của hàm ReturnAverage



# Các khái niệm về dòng dữ liệu

- Định nghĩa 7. (Global c-use.) Giả sử biến  $x$  được sử dụng để tính toán (c-use) tại đỉnh  $i$  của đồ thị dòng dữ liệu. Việc sử dụng biến  $x$  tại đỉnh  $i$  được gọi là Global c-use nếu  $x$  đã được định nghĩa ở các đỉnh trước đó.
- Ví dụ: Trong đồ thị dòng dữ liệu của hàm, việc sử dụng biến  $tv$  tại đỉnh 9 là Global c-use vì biến này đã được định nghĩa tại các đỉnh 2 và 5.

# Các khái niệm về dòng dữ liệu

- Định nghĩa 8. (Def-clear path.) Giả sử biến  $x$  được định nghĩa (def) tại đỉnh  $i$  và được sử dụng tại đỉnh  $j$ . Một đường đi từ  $i$  đến  $j$  ký hiệu là  $(i - n_1 - \dots - n_m - j)$  với  $m \geq 0$  được gọi là *Def-clear path* ứng với biến  $x$  nếu biến này không được định nghĩa tại các đỉnh từ  $n_1$  đến  $n_m$ .
- Ví dụ: đường đi  $(2 - 3 - 4 - 5)$  là một *Def-clear path* ứng với biến  $tv$  vì biến này được định nghĩa tại đỉnh 2, được sử dụng tại đỉnh 5, và không được định nghĩa tại các đỉnh 3 và 4.
- Tương tự, đường đi  $(2 - 3 - 4 - 6 - 3 - 4 - 6 - 3 - 4 - 5)$  cũng là một *Def-clear path* ứng với biến  $tv$ .

# Các khái niệm về dòng dữ liệu

- Định nghĩa 9. (Global def.) Một đỉnh  $i$  được gọi là Global def của biến  $x$  nếu đỉnh này định nghĩa biến  $x$  (def) và có một Def-clear path của  $x$  từ đỉnh  $i$  tới đỉnh chứa một Global c-use hoặc cạnh chứa một p-use của biến này.
- Bảng 1 liệt kê tất cả các Global def và Global c-use xuất hiện trong đồ thị dòng dữ liệu. Trong bảng này,  $\text{def}(i)$  được sử dụng để chỉ tập các biến có Global def tại đỉnh  $i$ . Tương tự,  $\text{c-use}(i)$  chỉ tập các biến có Global c-use tại đỉnh  $i$ . Tất cả các điều kiện ứng với các cạnh và các p-use xuất hiện trong đồ thị dòng dữ liệu cũng được liệt kê trong Bảng 2, với  $\text{p-use}(i, j)$  là tập các biến có p-use tại cạnh  $(i, j)$ .

Bảng 1 mô tả  $def()$  và  $c-use()$  của các đỉnh

Đỉnh $i$	$def(i)$	$c-use(i)$
1	{value, AS, MIN, MAX}	{}
2	{i, ti, tv, sum}	{}
3	{}	{}
4	{ti}	{ti}
5	{tv, sum}	{tv, i, sum, value}
6	{i}	{i}
7	{}	{}
8	{av}	{}
9	{av}	{sum, tv}
10	{}	{av}



# Bảng 2: Các điều kiện và p-use của các cạnh

$(i, j)$	Điều kiện tại cạnh $(i, j)$	$p\text{-use}(i, j)$
(1, 2)	True	$\{\}$
(2, 3)	True	$\{\}$
(3, 4)	$(ti < AS) \&\& (value[i] \neq -999)$	$\{i, ti, AS, value\}$
(4, 5)	$(value[i] \geq MIN) \&\& (value[i] \leq MAX)$	$\{i, MIN, MAX, value\}$
(4, 6)	$\sim((value[i] \geq MIN) \&\& (value[i] \leq MAX))$	$\{i, MIN, MAX, value\}$
(5, 6)	True	$\{\}$
(6, 3)	True	$\{\}$
(3, 7)	$\sim((ti < AS) \&\& (value[i] \neq -999))$	$\{i, ti, AS, value\}$
(7, 8)	$\sim(tv > 0)$	$\{tv\}$
(7, 9)	$(tv > 0)$	$\{tv\}$
(8, 10)	True	$\{\}$
(9, 10)	True	$\{\}$

# Các khái niệm về dòng dữ liệu

- Định nghĩa 10. (Simple path.) Một đường đi trong đồ thị dòng dữ liệu được gọi là một Simple path nếu các đỉnh chỉ xuất hiện đúng một lần trừ đỉnh đầu và đỉnh cuối.
- Ví dụ: Trong đồ thị dòng dữ liệu của hàm ReturnAverage, các đường đi (2 - 3 - 4 - 5) và (3 - 4 - 6 - 3) là các Simple paths.

# Các khái niệm về dòng dữ liệu

- Định nghĩa 11. (Loop-free path.) Một đường đi trong đồ thị dòng dữ liệu được gọi là một Loop-free path nếu các đỉnh chỉ xuất hiện đúng một lần.
- Định nghĩa 12. (Complete-path.) Một đường đi được gọi là một Complete-path nếu nó có điểm bắt đầu và điểm kết thúc chính là điểm bắt đầu và điểm kết thúc của đồ thị dòng dữ liệu.

# Các khái niệm về dòng dữ liệu

- Định nghĩa 13. (Du-path.) Một đường đi  $(n1 - n2 - \dots - nj - nk)$  được gọi là một Du-path (definition-use path) ứng với biến  $x$  nếu đỉnh  $n1$  là Global def của biến  $x$  và:
  - Đỉnh  $nk$  có một Global c-use với biến  $x$  và  $(n1 - n2 - \dots - nj - nk)$  là một Def-clear simple path với biến  $x$ , hoặc
  - Cạnh  $(nj, nk)$  có p-use với biến  $x$  và  $(n1 - n2 - \dots - nj)$  là Def-clear loop-free path với biến này.

# Các khái niệm về dòng dữ liệu

- Ví dụ, Trong đồ thị dòng dữ liệu của hàm ReturnAverage:
  - Đường đi (2 - 3 - 4 - 5) là một Du-path ứng với biến tv vì đỉnh 2 là Global def của biến tv, đỉnh 5 có Global c-use với biến này.
  - Đường đi (2 - 3 - 4 - 5) là một Def-clear simple path với biến tv.
  - Đường đi (2 - 3 - 7 - 9) cũng là một Du-path ứng với biến tv vì đỉnh 2 là Global def của biến này, cạnh (7, 9) có p-use với biến tv
  - Đường đi (2 - 3 - 7) là một Def-clear loop-free path với tv.

# Các độ đo cho kiểm thử dòng dữ liệu

- All-defs: tương ứng với tất cả các đường đi
- All-c-uses: đường đi qua tất cả các đỉnh
- All-p-uses: đường đi qua tất cả các cạnh.

# Độ đo All-defs

- All-defs: Mỗi một biến  $x \in V$  và mỗi đỉnh  $i \in N$ , giả sử  $x$  có một Global def tại  $i$ , chọn một Complete-path chứa một Def-clear path từ đỉnh  $i$  tới đỉnh  $j$  sao cho tại  $j$  có chứa một Global c-use của  $x$ , hoặc cạnh  $(j, k)$  chứa một p-use của biến  $x$ .
- Ví dụ: Biến  $tv$  có hai Global def tại các đỉnh 2 và 5, có một Global c-use của biến  $tv$  tại đỉnh 5 và tồn tại một Def-clear path (2 - 3 - 4 - 5) từ đỉnh 2 tới đỉnh 5. Đồng thời tồn tại một Complete-path (1 - 2 - 3 - 4 - 5 - 6 - 3 - 7 - 9 - 10) chứa đường đi này. Do vậy đường đi (2 - 3 - 4 - 5) thỏa mãn độ đo All-defs.

# Độ đo All-c-uses

- All-c-uses: Với mỗi một biến  $x$  và mỗi đỉnh  $i$  sao cho  $i$  là Global def với biến  $x$ , chọn các Complete-path bao gồm các Def-clear path từ đỉnh  $i$  tới tất cả các đỉnh  $j$  sao cho  $j$  là Global c-use của  $x$ .



# Độ đo All-c-uses

- Ví dụ: Tìm tất cả các đường đi thỏa mãn độ đo All-c-uses ứng với biến *ti*
  - Có 2 Global def của biến này tại các đỉnh 2 và 4. Với đỉnh 2, có một Global c-use của biến *ti* tại đỉnh 4 ứng với Def-clear path là (2 - 3 - 4). Tuy nhiên, với Global def của biến này tại đỉnh 4, ta không tìm thấy Global c-use nào của biến *ti*.
  - Có thể sử dụng một trong 4 Complete-path sau để đảm bảo độ đo này với biến *ti*:
    - (1 - 2 - 3 - 4 - 5 - 6 - 3 - 7 - 8 - 10),
    - (1 - 2 - 3 - 4 - 5 - 6 - 3 - 7 - 9 - 10),
    - (1 - 2 - 3 - 4 - 6 - 3 - 7 - 8 - 10),
    - (1 - 2 - 3 - 4 - 6 - 3 - 7 - 9 - 10).
- Để quá trình kiểm thử dòng dữ liệu thỏa mãn độ đo All-c-uses, cần tiến hành một cách tương tự với các biến còn lại (*i*, *tv*, và *sum*)

# Độ đo All-p-uses

- All-p-uses: Với mỗi một biến  $x$  và mỗi đỉnh  $i$  sao cho  $i$  là Global def với biến  $x$ , chọn các Complete-path bao gồm các Def-clear path từ đỉnh  $i$  tới tất cả các cạnh  $(j, k)$  sao cho có một p-use của  $x$  tại cạnh này.

# Độ đo All-p-uses

- Ví dụ: Tìm tất cả các đường đi thỏa mãn độ đo All-p-uses ứng với biến tv.
  - Có 2 Global def tại các đỉnh 2 và 5. Tại đỉnh 2, ta có hai p-use của biến này tại các cạnh (7, 8) và (7, 9).
  - Đường đi (2 - 3 - 7 - 8) là một Def-clear path của tv từ đỉnh 2 đến cạnh (7, 8)
  - Đường đi (2 - 3 - 7 - 9) là một Def-clear path của tv từ đỉnh 2 đến cạnh (7, 9).
  - Đường đi (5 - 6 - 3 - 7 - 8) là một Def-clear path của tv từ đỉnh 5 đến cạnh (7, 8)
  - Đường đi (5 - 6 - 3 - 7 - 9) là một Def-clear path của tv từ đỉnh 5 đến cạnh (7, 9).
  - Có thể sử dụng một trong 4 Complete-path sau để đảm bảo độ đo này với biến tv:
    - (1 - 2 - 3 - 7 - 8 - 10),
    - (1 - 2 - 3 - 7 - 9 - 10),
    - (1 - 2 - 3 - 4 - 5 - 6 - 3 - 7 - 8 - 10),
    - (1 - 2 - 3 - 4 - 5 - 6 - 3 - 7 - 9 - 10).

# Độ đo All-p-uses/Some c-uses

- Some-c-uses: Với mỗi một biến  $x$  và mỗi đỉnh  $i$  sao cho  $i$  là *Global def* với biến  $x$ , chọn các *Complete-path* bao gồm các *Def-clear path* từ đỉnh  $i$  tới một số đỉnh  $j$  sao cho  $j$  là *Global c-use* của  $x$ .

# Độ đo All-p-uses/Some c-uses

- Ví dụ: Tìm tất cả các đường đi thỏa mãn độ đo All-p-uses/Some c-uses ứng với biến i.
  - Biến i có hai Global def tại các đỉnh 2 và đỉnh 6.
  - Trong khi không có p-use của biến này.
  - Vậy, cần tìm đến độ đo Some-c-uses của biến i.
  - Tại đỉnh 2, có một Global c-use của i tại đỉnh 6 và có một Def-clear path (2 - 3 - 4 - 5 - 6).
- => để thỏa mãn độ đo All-p-uses/Some-c-uses với biến i, cần chọn Complete-path (1 - 2 - 3 - 4 - 5 - 6 - 3 - 7 - 9 - 10) chứa đường đi trên.

# Độ đo All-c-uses/Some-p-uses

- Some-p-uses: Với mỗi một biến  $x$  và mỗi đỉnh  $i$  sao cho  $i$  là *Global def* với biến  $x$ , chọn các *Complete-path* bao gồm các *Def-clear path* từ đỉnh  $i$  tới một số cạnh  $(j, k)$  sao cho có một *p-use* của  $x$  tại cạnh này.

# Độ đo All-c-uses/Some-p-uses

- Ví dụ: Tìm tất cả các đường đi thỏa mãn độ đo All-c-uses/Some p-uses ứng với biến AS
  - Biến AS chỉ có một Global def tại đỉnh 1.
  - Không có Global c-use
  - Vậy cần tính theo độ đo Some-p-uses của biến AS.
  - Từ đỉnh 1, có các p-use của AS tại các cạnh (3, 7) và (3, 4) và có các Def-clear path tương ứng với hai cạnh này là (1 - 2 - 3 - 7) và (1 - 2 - 3 - 4). Có rất nhiều Complete-path chứa hai đường đi này. Ví dụ (1 - 2 - 3 - 4 - 5 - 6 - 3 - 7 - 9 - 10)

# Độ đo All-uses

- All-uses: Độ đo này bao gồm các đường đi được sinh ra từ các độ đo All-p-uses và All-c-uses. Điều này có nghĩa là với mỗi việc sử dụng (c-use hoặc p-use) của một biến thì có một đường đi từ định nghĩa (def) của biến đó tới các đỉnh hoặc các cạnh sử dụng nó.



# Độ đo All-du-paths

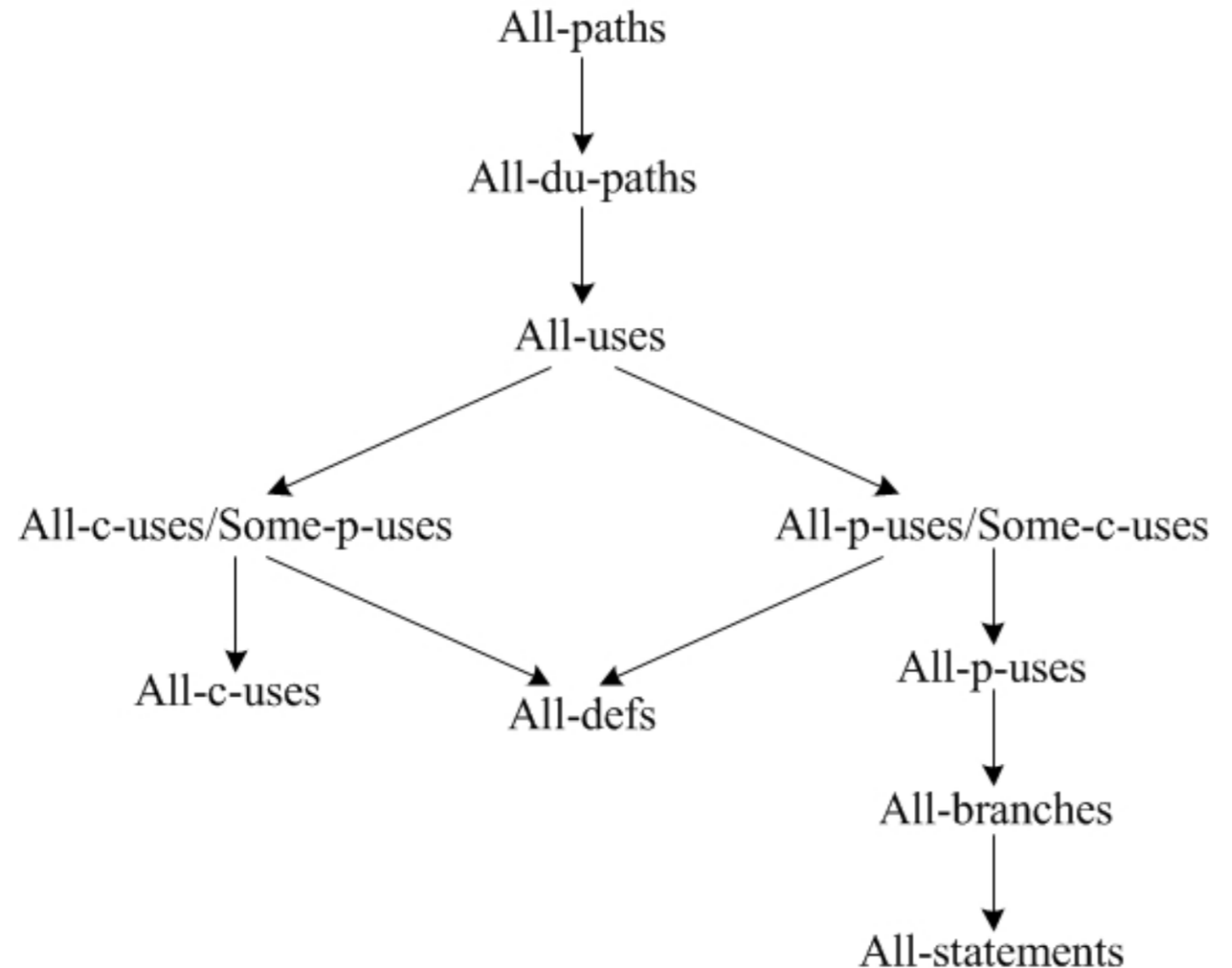
- All-du-paths: Với mỗi một biến  $x$  và mỗi đỉnh  $i$  sao cho  $i$  là Global def với biến  $x$ , chọn các Complete-path chứa các tất cả các Du-path từ đỉnh  $i$  tới:
  - Tất cả các đỉnh  $j$  sao cho có một Global c-use của biến  $x$  tại  $j$
  - Tất cả các cạnh  $(j, k)$  sao cho có một p-use của biến  $x$  tại  $(j, k)$ .

# So sánh các độ đo của kiểm thử dòng dữ liệu

- **Mối quan hệ bao gồm:** Cho hai độ đo  $c1$  và  $c2$ , ta nói  $c1$  bao gồm  $c2$  nếu mọi đường đi đầy đủ (Complete-paths) sinh ra từ đồ thị dòng dữ liệu thỏa mãn  $c1$  thì cũng thỏa mãn  $c2$ .
- **Mối quan hệ bao gồm chặt:** Cho hai độ đo  $c1$  và  $c2$ , ta nói  $c1$  bao gồm chặt  $c2$ , ký hiệu là  $c1 \rightarrow c2$ , nếu  $c1$  bao gồm  $c2$  và tồn tại một số đường đi đầy đủ sinh ra từ đồ thị dòng dữ liệu thỏa mãn  $c2$  nhưng không thỏa mãn  $c1$ .

# So sánh các độ đo của kiểm thử dòng dữ liệu

- Theo các khái niệm về độ đo ở trên thì mối quan hệ giữa các độ đo của kiểm thử dòng dữ liệu được mô tả như hình bên



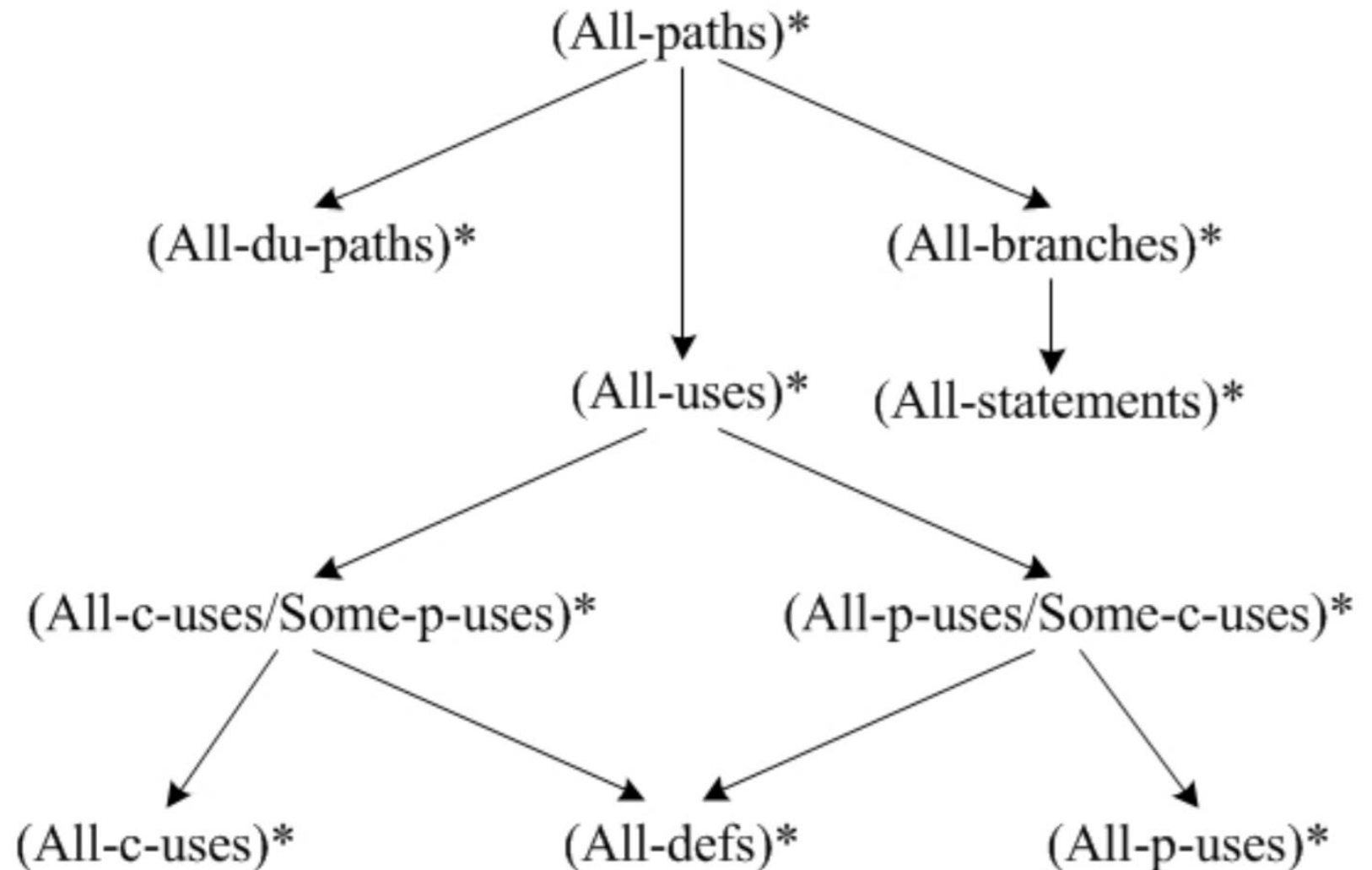
# Sinh các ca kiểm thử

- Sinh đồ thị kiểm thử dòng dữ liệu
  - Với độ đo kiểm thử C, cần xác định đường đi thực thi được từ các đường đi đầy đủ thỏa mãn độ đo này.
  - Trong đó, đường đi thực thi được là đường đi từ một bộ dữ liệu đầu vào thỏa mãn để thực hiện chương trình.
- => cần xác định bộ dữ liệu đầu vào cho các đường đi thực thi được.

# Sinh các ca kiểm thử

- Ví dụ: Xét đường đi đầy đủ (1 - 2 - 3 - 4 - 5 - 6 - 3 - 7 - 9) từ đồ thị dòng dữ liệu. Từ đó xác định các biểu thức thuộc các p-uses nằm trên các cạnh của nó.
  - $((ti < AS) \ \&\& \ (value[i] \neq -999))$  (cạnh 3, 4)
  - $((value[i] \geq MIN) \ \&\& \ (value[i] \leq MAX))$  (cạnh (4, 5))
  - $\sim((ti < AS) \ \&\& \ (value[i] \neq -999))$  (cạnh 3, 7)
  - $(tv > 0)$  (cạnh 7, 9)
- $\Rightarrow$  cần sinh ra các bộ giá trị đầu vào thỏa mãn các điều kiện trên (VD: bộ  $([1, -999], 2, 1, 2)$  )

Mỗi quan hệ bao gồm chặt giữa các độ đo dòng dữ liệu thực thi được



# Bài tập

Cho hàm calFactorial:

- Hãy liệt kê các câu lệnh ứng với các khái niệm def, c-use, và p-use ứng với các biến được sử dụng trong hàm này.
- Hãy vẽ đồ thị dòng dữ liệu của hàm này.

```
int calFactorial (int n){  
    int result = 1;  
    int i=1;  
    while (i <= n){  
        result = result*i;  
        i++;  
    }//end while  
    return result;  
}//the end
```

# Bài tập

Cho đồ thị dòng dữ liệu như hình bên

- Hãy xác định tất cả các Complete-path từ đồ thị này.
- Các đường đi (0 - 1 - 2 - 3), (1 - 2 - 3 - 2 - 4), (3 - 2 - 4), (2 - 3 - 2) và (0 - 1 - 4) có phải là các simple-paths không? Giải thích.
- Các đường đi (2 - 3 - 2), (2 - 3) và (3 - 2 - 4) có phải là các loop-free-paths không? Giải thích.

