

ĐỒ THỊ LUỒNG DỮ LIỆU

DATA FLOW GRAPH

Các mức của Data Flow Testing

- **Static Data Flow Testing:** Nghiên cứu, phân tích mã nguồn để tìm ra lỗi dùng không đúng cách
- **Dynamic Data Flow Testing:** Thiết kế Test case chạy mã nguồn xác định biến được tính đúng giá trị hay không

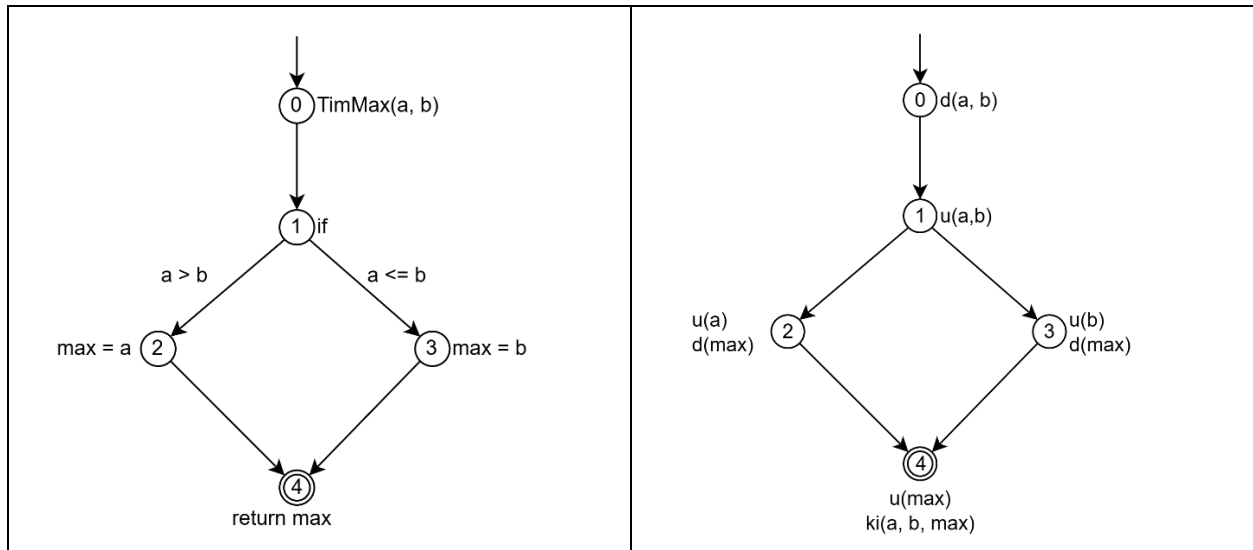
STATIC DATA FLOW TESTING

Cách xây dựng DFG

- Vẽ đồ thị CFG
- Bổ sung các thông tin def, use, kill tại các đỉnh

| Thông tin | Chức năng | Ví dụ |
|-----------|--|--|
| def | Xác định giá trị của biến (khởi tạo, gán, nhập,...) | int max(a, b) → def(a, b) int x = 1; → def(x) x = 2; → def(x) |
| use | Sử dụng biến c-use → Sử dụng tính toán p-use → Sử dụng để kiểm tra | x = a + b → u(a, b); d(x) (c-use) If(a > c) → u(a, c) (p-use) |
| kill | Kết thúc sự tồn tại của biến, giới hạn phạm vi | Biến khởi tạo trong 1 hàm, hay trong vòng lặp,... |

Ví dụ:

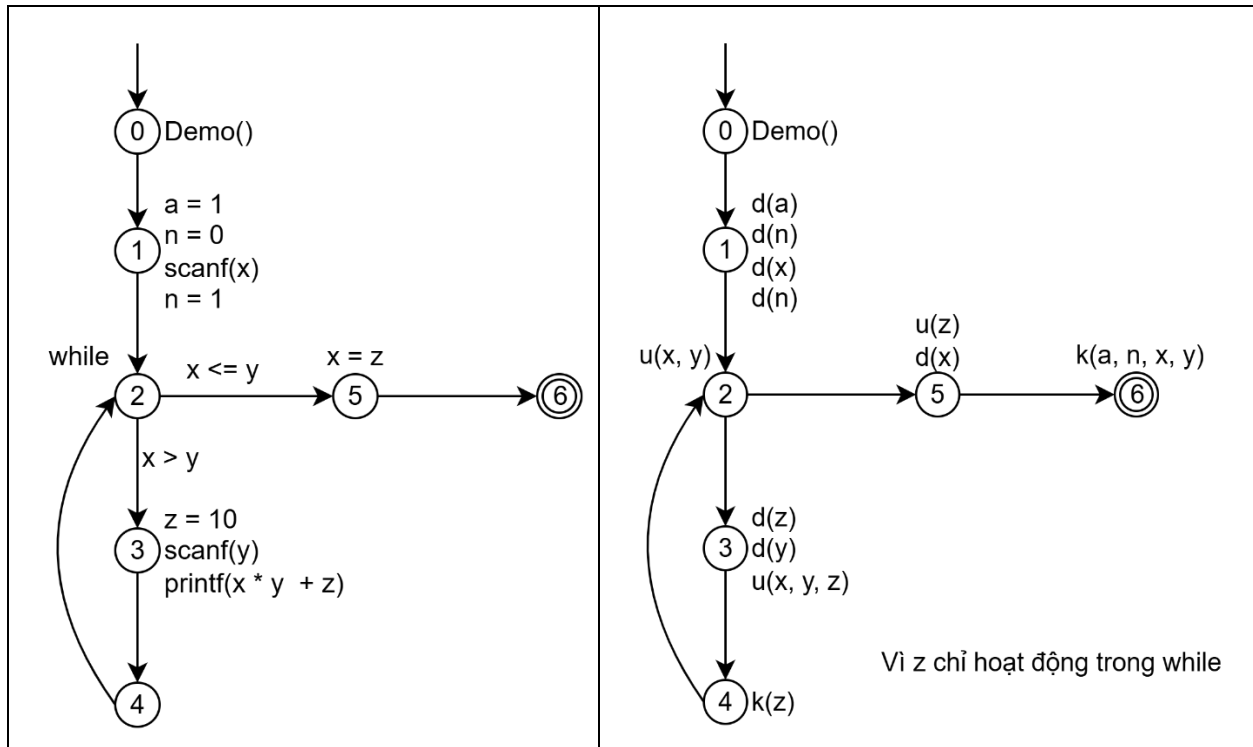


Bài tập 1:

```

void Demo(){
    int n, x, y, a = 1;
    n = 0;
    scanf("%d", &x);
    n = 1;
    while(x > y){
        int z = 10;
        scanf("%d", &y);
        printf("%d", x * y + z);
    }
    x = -z;
}

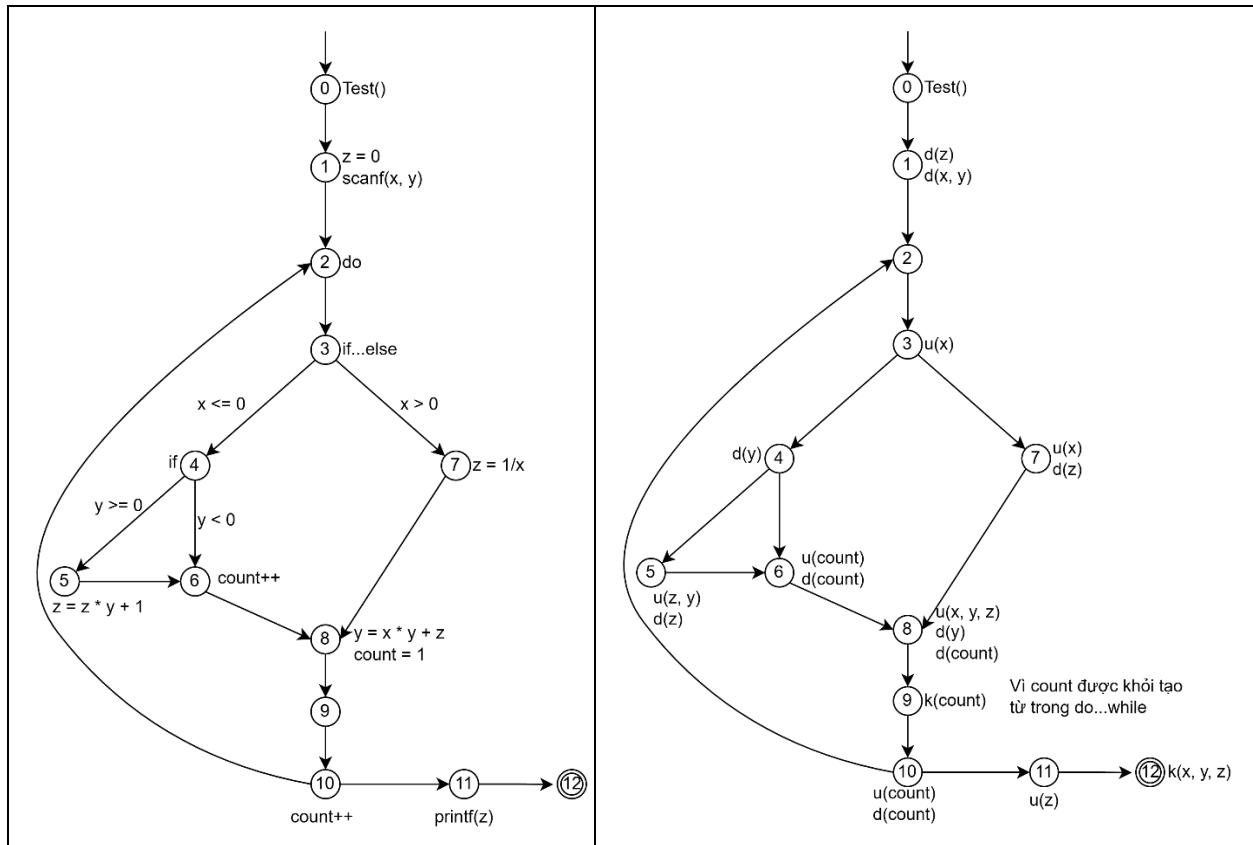
```



Bài tập 2:

```

void Test(){
    float x, y, z = 0;
    int count;
    printf("Nhap x, y: ");
    scanf("%f%f", &x, &y);
    do{
        if(x < 0){
            if(y >= 0)
                z = z * y + 1
                count++;
        }
        else z = 1/x;
        y = x * y + z;
        count = 1;
    }
    while(count > 0);
    printf("KQ: %2f", z);
}
  
```

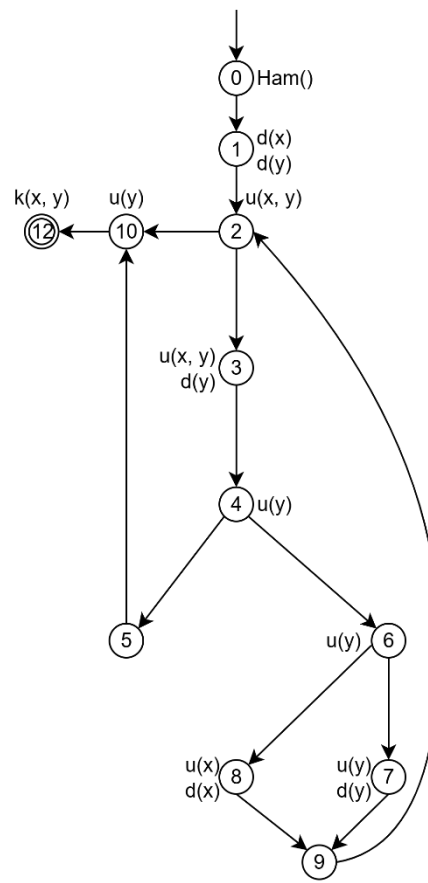
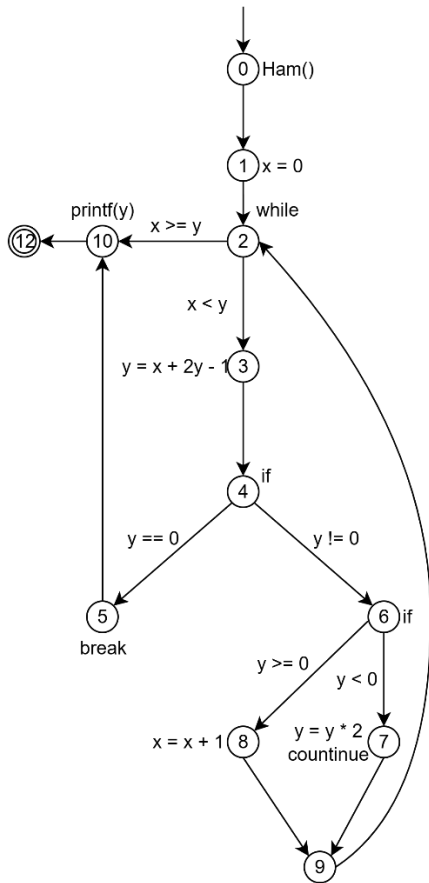


Bài tập 3:

```
void Ham(){
    int y, x = 0;
    scanf("%d", &y);
    while(x < y){
        y = x + 2*y - 1;
        if(y == 0) break;
        else if(y < 0){
            y = y * 2;
            continue;
        }
        x = x + 1;
    }
    printf("%d", y);
}
```

- Vẽ DFG
- Xác định tập lộ trình độc lập
- Xây dựng kịch bản đời sống cho x, y

a. Vẽ DFG



b. Tập lộ trình độc lập

$$C = 3 + 1 = 4$$

| STT | Lộ trình | Data test |
|-----|-----------------------------------|----------------------------------|
| 1 | 012(10)(11) | $x = 0, y = 0 \rightarrow y = 0$ |
| 2 | 012346892(10)(11) | $x = 0, y = 1 \rightarrow y = 1$ |
| 3 | Không có vì điều kiện bị xung đột | |
| 4 | | |

c. Xây dựng kịch bản đời sống cho x, y

| Biến | Lộ trình | Kịch bản |
|------|-------------------|-------------|
| x | 012(10)(11) | ~duk~ |
| | 012346892(10)(11) | ~duuuduk~ |
| y | 012(10)(11) | ~duk~ |
| | 012346892(10)(11) | ~duuuduuuk~ |

Bài tập 4:

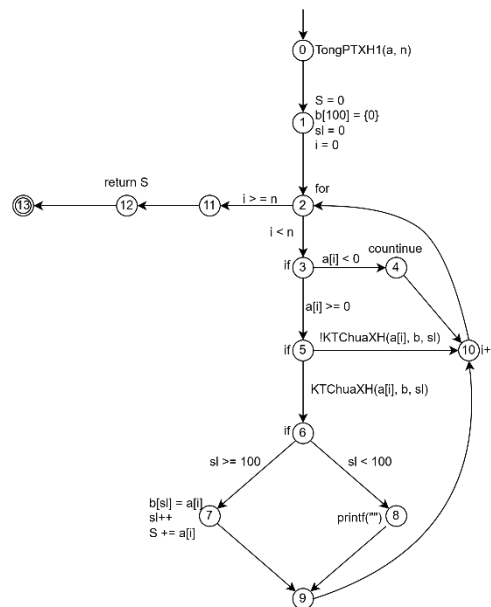
```

int TongPTXH(int a[], int n){
    int S = 0;
    int b[100] = {0};
    int sl = 0;
    for(int i = 0; i < n; i++){
        if(a[i] < 0) continue;
        if(KTChuaXuatHien(a[i], b, sl)){
            if(sl < 100){
                b[sl] = a[i];
                sl++;
                S += a[i];
            }
            else printf("");
        }
    }
    return S;
}

```

- Vẽ CFG
- Tìm tập lộ trình tối thiểu thoả tiêu chí bao phủ quyết định, thiết kế test case.
- Vẽ DFG
- Xác định độ phức tạp chu trình.
- Xác định tập lộ trình độc lập.
- Xây dựng kịch bản cho biến S.

a. Vẽ CFG

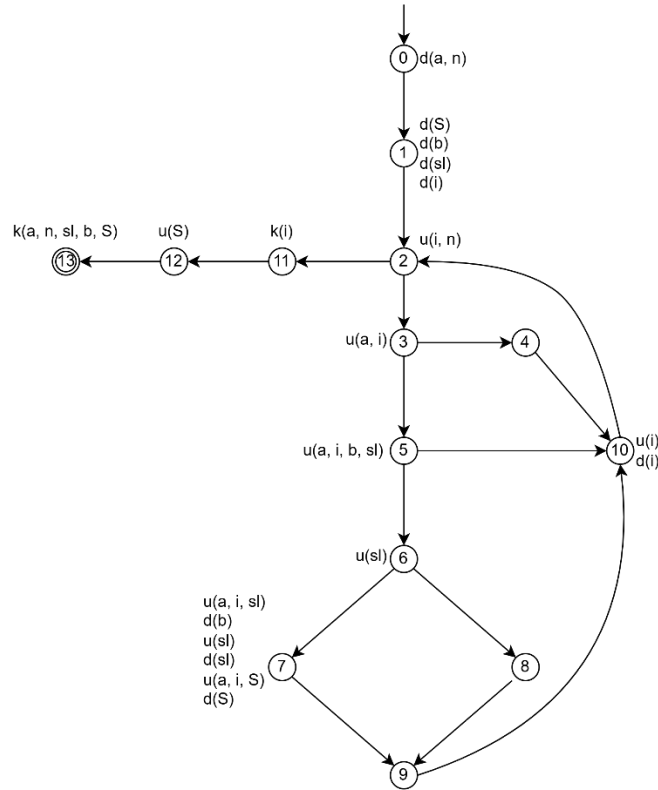


b. Tập lộ trình tối thiểu thoả tiêu chí bao phủ quyết định

Path_{min} = 01(235679(10))*234(10)235689(10)2(11)(12)(13)

Data test: a = {1; 2; 3...99; 2; 5; 100} n = 102

c. Vẽ DFG



d. Độ phức tạp của chu trình:

$$C = 4 + 1 = 5 \text{ (Basic Paths)}$$

e. Tập lộ trình độc lập

| STT | Lộ trình |
|-----|---|
| 1 | 01(235679(10))*234(10)235689(10)2(11)(12)(13) |
| | n = 102 |
| | a = {1; 2; 3...99; 2; 5; 100} |
| 2 | 012(11)(12)(13) |
| | n = 0 |
| | a = {} |
| 3 | 01234(10)2(11)(12)(13) |
| | n = 1 |
| | a = {-5} |
| 4 | 01235679(10)235(10)2(11)(12)(13) |
| | n = 2 |
| | a = {5; -5} |
| 5 | 01235679(10)235679(10)2(11)(12)(13) |
| | n = 2 |
| | a = {5; 10} |

f. Xây dựng kịch bản cho S

| Biến | Lộ trình | Kịch bản |
|-------------|--|------------------|
| S | 01(<u>2356</u> 79(10))*234(10)235689(10)2(11)(12)(13) | ~d(uuuu)*uuuuuk~ |
| | 01 <u>2</u> (11)(12)(13) | ~dk~ |
| | 01234(10) <u>2</u> (11)(12)(13) | ~duk~ |
| | 01235679(10) <u>235</u> (10) <u>2</u> (11)(12)(13) | ~duuuuuuk~ |
| | 01 <u>2356</u> 79(10) <u>2356</u> 79(10) <u>2</u> (11)(12)(13) | ~duuuuuuuuk~ |