



Cobertura de Dados nos Testes de Unidade

Testes de Unidade Dinâmicos

A seleção dos dados de testes é baseada nas seguintes técnicas:

- **Teste de Fluxo de Controle**
- ➔ ▪ **Teste de Fluxo de Dados**
- **Teste de Domínio**
- **Teste Funcional**

Teste de Fluxo de Dados

Os testes são feitos considerando os valores dos dados. O código fonte é analisado para encontrar potenciais defeitos conhecidos como anomalias no fluxo de dados:

- Variáveis definidas e, então, definidas novamente
 - Variáveis não definidas mas referenciadas
 - Variáveis definidas mas não referenciadas
- É responsabilidade do programador examinar as anomalias encontradas.

Anomalias

- Anomalia: Variáveis definidas e, então, definidas novamente.

Exemplo:

```
:  
x = f1(y)  
x = f2(z)  
:
```

Possíveis interpretações sobre esta anomalia:

- O primeiro comando é redundante se o segundo executa o que é necessário.
- O primeiro comando tem uma falta. Por exemplo, deveria ser $w = f1(y)$.
- O segundo comando tem uma falta. Por exemplo, deveria ser $y = f2(z)$.
- Poderia ter uma falta entre os dois comandos. Por exemplo, deveria ter o comando $v = f3(x)$ entre o primeiro e o segundo.

Anomalias

- Anomalia: Variáveis não definidas mas referenciadas.

Exemplo: $x = x - y - w$

onde a variável w não foi inicializada.

Possíveis interpretações sobre esta anomalia:

- A inicialização de w foi esquecida; ou
- A intenção poderia ter sido usar outra variável inicializada, como a variável y, no lugar de w.

Anomalias

- Anomalia: Variáveis definidas mas não referenciadas

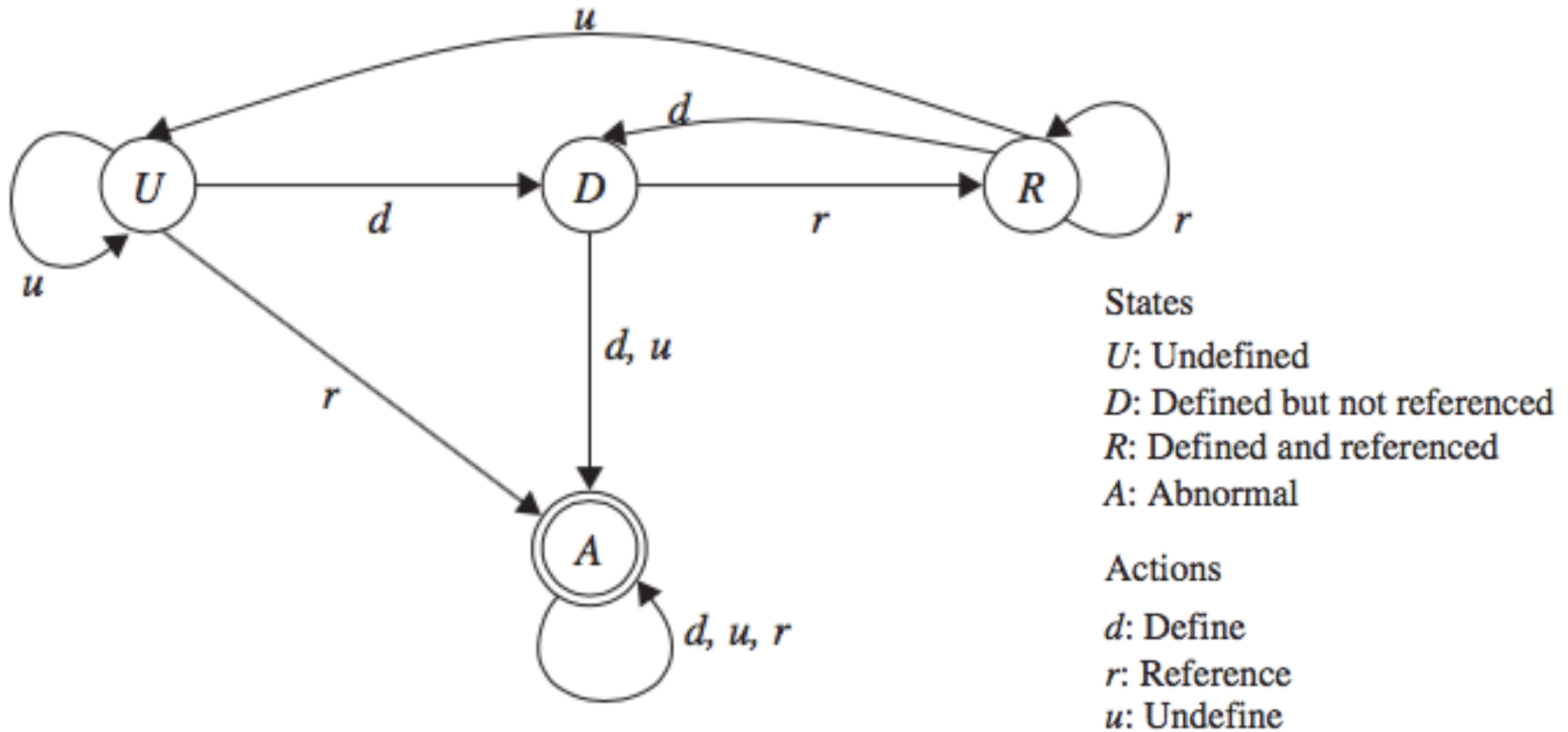
Exemplo: $x = f(x, y)$

após esta atribuição x não é mais usada.

- Por que o valor x não foi mais usado?

Anomalias

Para identificar anomalias no fluxo de dados, pode-se utilizar a ideia de “estados” das variáveis de programa:



Anomalias

- A presença de uma anomalia significa que o programa pode falhar e que o programador deve investigar a causa da anomalia.
- Também é importante verificar se o valor correto foi atribuído a uma variável. Isso deve ser feito através de um teste que execute um caminho desde a atribuição até um ponto onde o seu valor é usado.

Teste de Fluxo de Dados

- O teste de fluxo de dados envolve a seleção de caminhos com base em critérios de definição e uso de dados.
- O objetivo é encontrar caminhos que incluem a definição e uso das variáveis.

Teste de Fluxo de Dados

Passos:

1. Desenhe um grafo de fluxo de dados da unidade.
2. Selecione um critério de seleção de fluxo de dados.
3. Identifique os caminhos que satisfazem o critério.
4. Derive expressões de predicado de caminho a partir dos caminhos selecionados.
5. Gere valores de entradas dos testes da unidade que exercitam os caminhos selecionados.

Teste de Fluxo de Dados

Passos:

1. **Desenhe um grafo de fluxo de dados da unidade.**
2. Selecione um critério de seleção de fluxo de dados.
3. Identifique os caminhos que satisfazem o critério.
4. Derive expressões de predicado de caminho a partir dos caminhos selecionados.
5. Gere valores de entradas dos testes da unidade que exercitam os caminhos selecionados.

Grafo de Fluxo de Dados



São identificados caminhos no código fonte baseados no fluxo de dados.

Um grafo de fluxo de dados é desenhado para identificar as variáveis e seus usos.

Grafo de Fluxo de Dados

Exemplo de identificação de variáveis e seus usos.

```
int VarTypes(int x, int y){  
    int i;  
    int *iptr;  
    i = x;  
    iptr = malloc(sizeof(int));  
    *iptr = i + x;  
    if (*iptr > y)  
        return (x);  
    else {  
        iptr = malloc(sizeof(int));  
        *iptr = x + y;  
        return(*iptr);  
    }  
}
```

variável i definida

c-use de i e x

p-use de iptr e y

o endereço apontado anteriormente por iptr fica indefinido

Grafo de Fluxo de Dados

Cada ocorrência de uma variável é classificada como:

- **Definida:** um valor é movido para o endereço da variável.
- **Indefinida ou Morta (Kill):** o valor ou endereço ficam sem referência (*unbound*).
- **Usada:** o valor é buscado do endereço da variável. Pode ser: usada em uma computação (c-use) ou usada em um predicado (p-use).

Como construir um GFD?

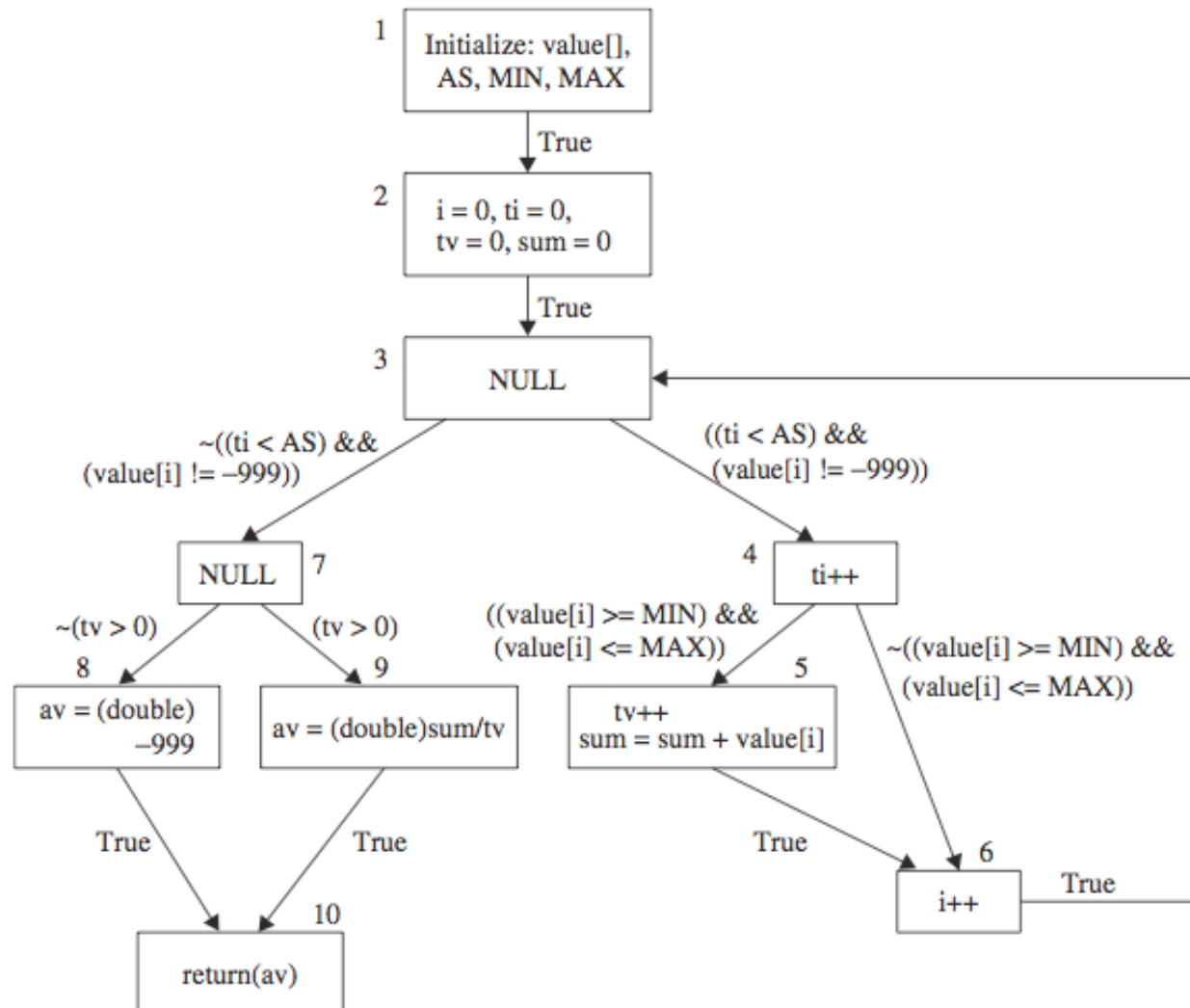
Um grafo de fluxo de dados é um grafo dirigido construído da seguinte maneira:

- Uma sequência de definições e c-uses é associada com cada nodo do grafo.
- Um conjunto de p-uses é associado com cada aresta do grafo.
- O nodo de entrada tem uma definição de cada parâmetro e variável não local que aparece no subprograma.

GFD - Exemplo

```
public static double ReturnAverage(int value[],
                                   int AS, int MIN, int MAX){
    /*
    Function: ReturnAverage Computes the average
    of all those numbers in the input array in
    the positive range [MIN, MAX]. The maximum
    size of the array is AS. But, the array size
    could be smaller than AS in which case the end
    of input is represented by -999.
    */
    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++;
    }
    if (tv > 0)
        av = (double)sum/tv;
    else
        av = (double) -999;
    return (av);
}
```


GFD - Exemplo



Teste de Fluxo de Dados

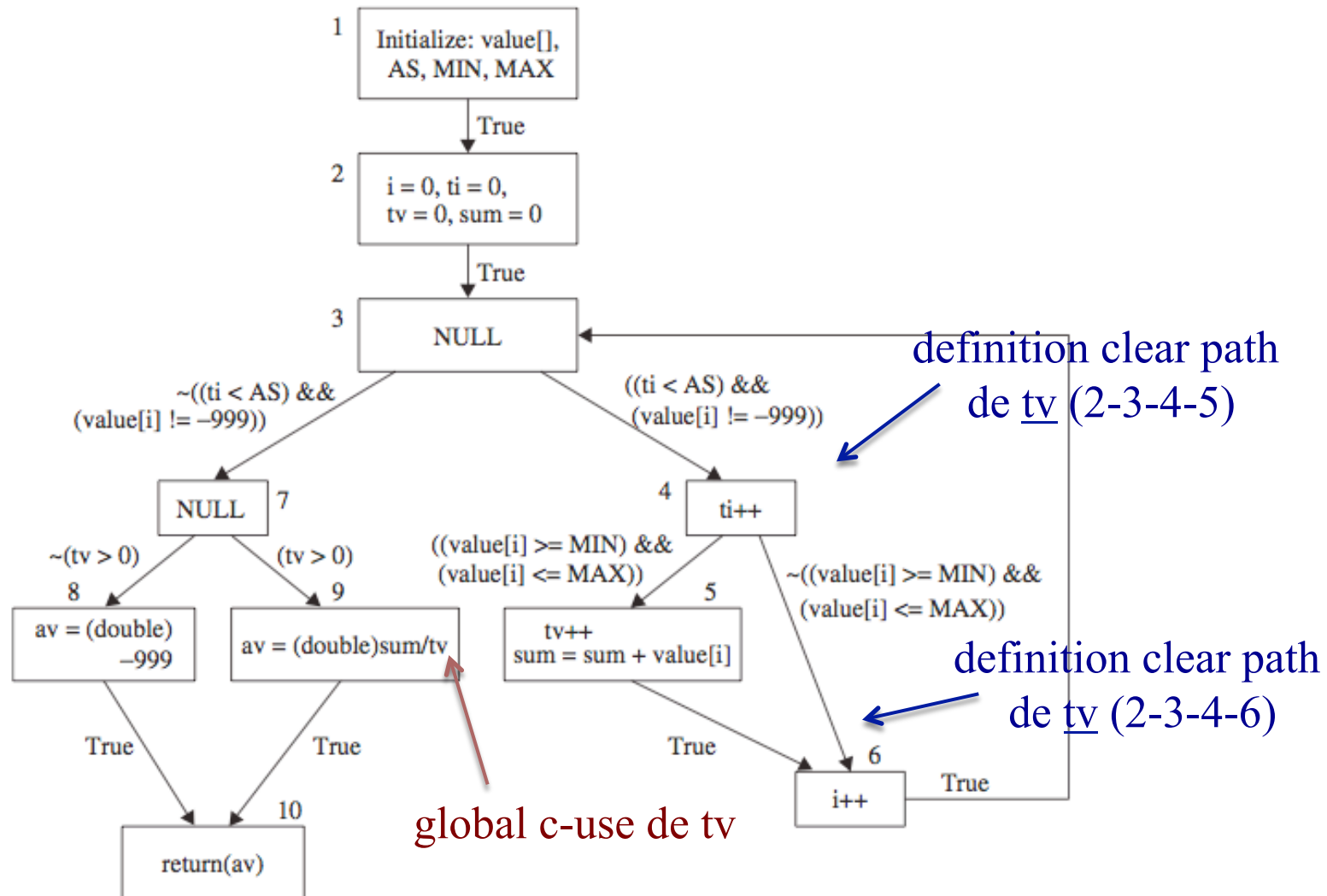
Passos:

1. Desenhe um grafo de fluxo de dados da unidade.
2. **Selecione um critério de seleção de fluxo de dados.**
3. **Identifique os caminhos que satisfazem o critério.**
4. Derive expressões de predicado de caminho a partir dos caminhos selecionados.
5. Gere valores de entradas dos testes da unidade que exercitam os caminhos selecionados.

Definições (1)

- **Global c-use:** Um c-use de uma variável x em um nodo i é dito ser global se x foi definida em algum nodo antes do nodo i .
- **Definition Clear Path:** Um caminho $(i - n_1 - \dots - n_m - j)$, $m \geq 0$, é um def-clear path em relação à variável x do nodo i ao nodo j e do nodo i à aresta $(n_m - j)$ se x não tem sido definida nem indefinida nos nodos n_1, \dots, n_m .

Definições - Exemplo



Definições (2)

- **Global definition:** um nodo i tem uma definição global da variável x se o nodo i tem uma definição de x e existe um def-clear path em relação à x a partir do nodo i (i) a algum nodo que contenha um global c-use de x ou (ii) a alguma aresta que contenha um p-use de x .

Definições - Exemplo

- Global definitions e global c-uses do GFD anterior.

Nodes 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}

Definições - Exemplo

- Predicados e p-uses do GFD anterior.

Edges (i, j)	predicate(i, j)	p-use(i, j)
(1, 2)	True	{}
(2, 3)	True	{}
(3, 4)	$(ti < AS) \ \&\& \ (value[i] \neq -999)$	{i, ti, AS, value}
(4, 5)	$(value[i] \leq MIN) \ \&\& \ (value[i] \geq MAX)$	{i, MIN, MAX, value}
(4, 6)	$\sim((value[i] \leq MIN) \ \&\& \ (value[i] \geq MAX))$	{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	{}

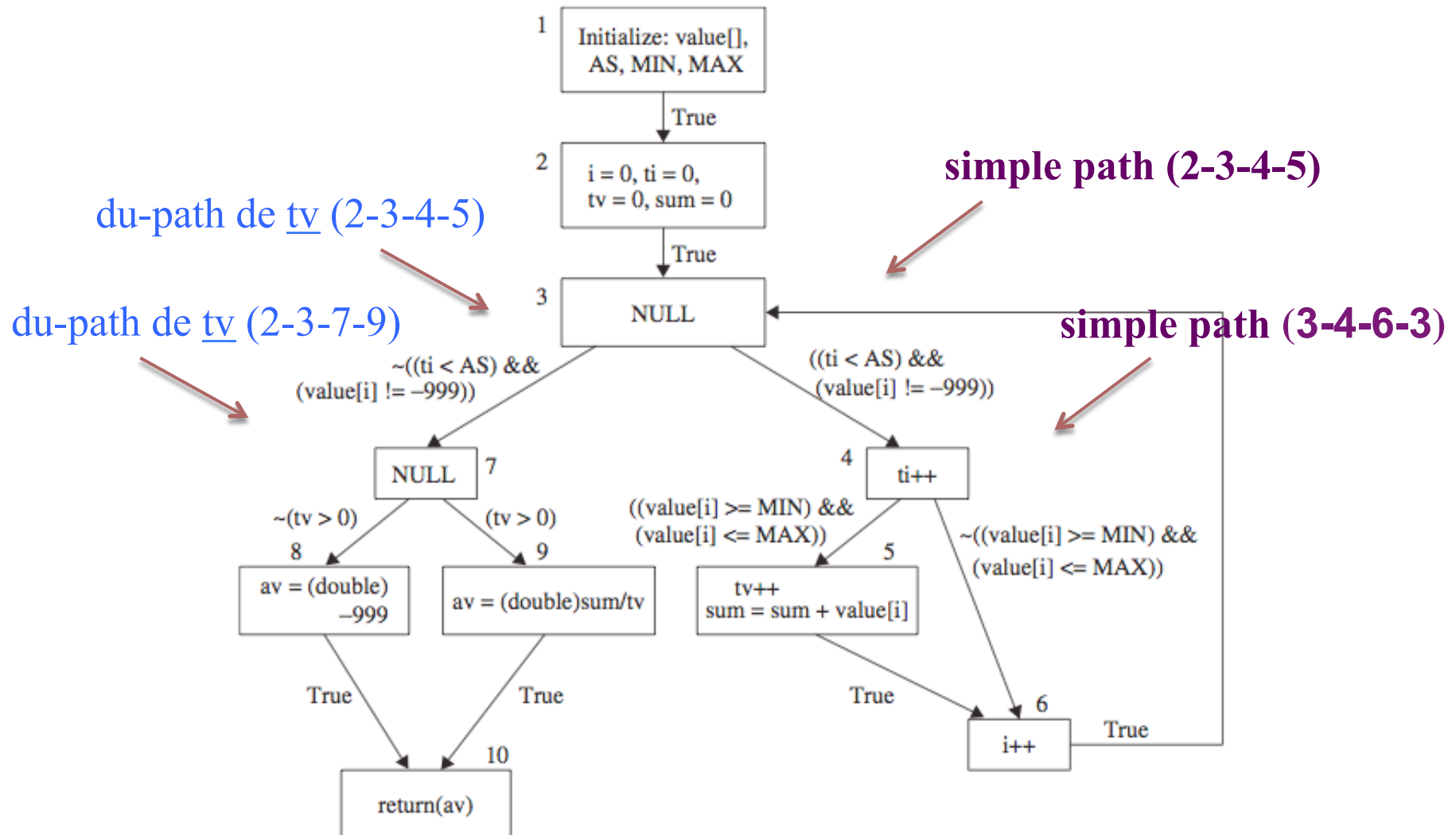
Definições (3)

- **Simple Path:** é um caminho no qual todos os nodos, exceto o primeiro e o último são distintos.
- **Loop-Free Path:** é um caminho no qual todos os nodos são distintos.
- **Complete Path:** é um caminho a partir do nodo de entrada até o nodo de saída.

Definições (4)

- **Du-path:** um caminho $(n_1 - n_2 - \dots - n_i - n_k)$ é um definition-use path (du-path) em relação à variável x se o nodo n_1 tem uma definição global de x e
 - (i) o nodo n_k tem um global c-use de x e $(n_1 - n_2 - \dots - n_i - n_k)$ é um def-clear simple path em relação à x ou
 - (ii) a aresta (n_i, n_k) tem um p-use de x e $(n_1 - n_2 - \dots - n_i)$ é um def-clear loop-free path em relação à x.

Definições - Exemplo



Critérios de Seleção de Caminhos

Tipos de critérios de teste de fluxo de dados:

- **1. All-defs**
- **2. All-c-uses**
- **3. All-p-uses**
- **4. All-p-uses/Some-c-uses**
- **5. All-c-uses/Some-p-uses**
- **6. All-uses**
- **7. All-du-paths**

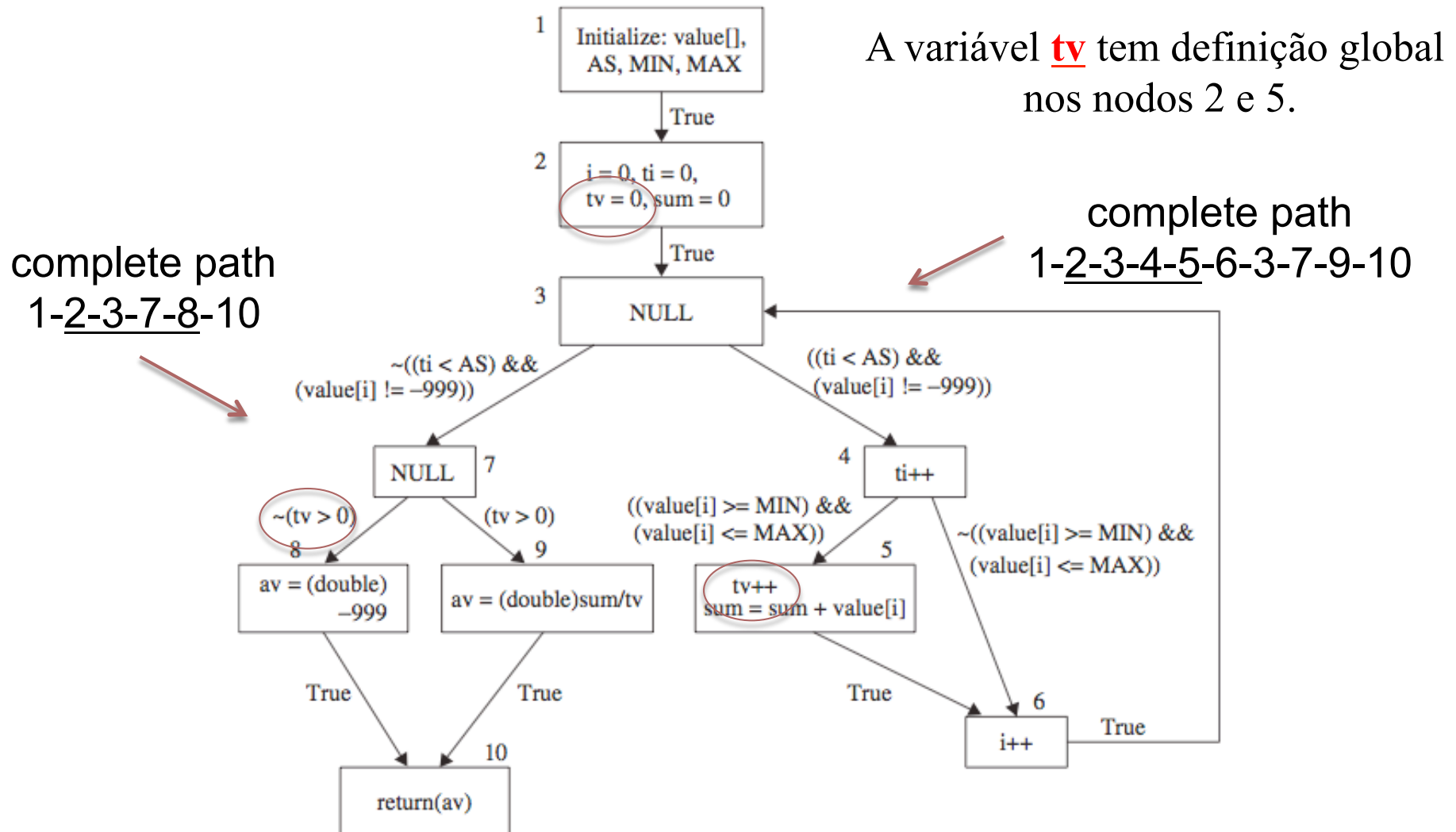
Critérios de Seleção de Caminhos

➤ 1 - All-defs

Para cada variável x e para cada nodo i tal que x tem uma definição global no nodo i , selecione um caminho completo que inclua um def-clear path a partir do nodo i até

- o nodo j que tem uma global c-use de x ou
- a aresta (i,k) que tem um p-use de x .

Exemplo de All-defs



Critérios de Seleção de Caminhos

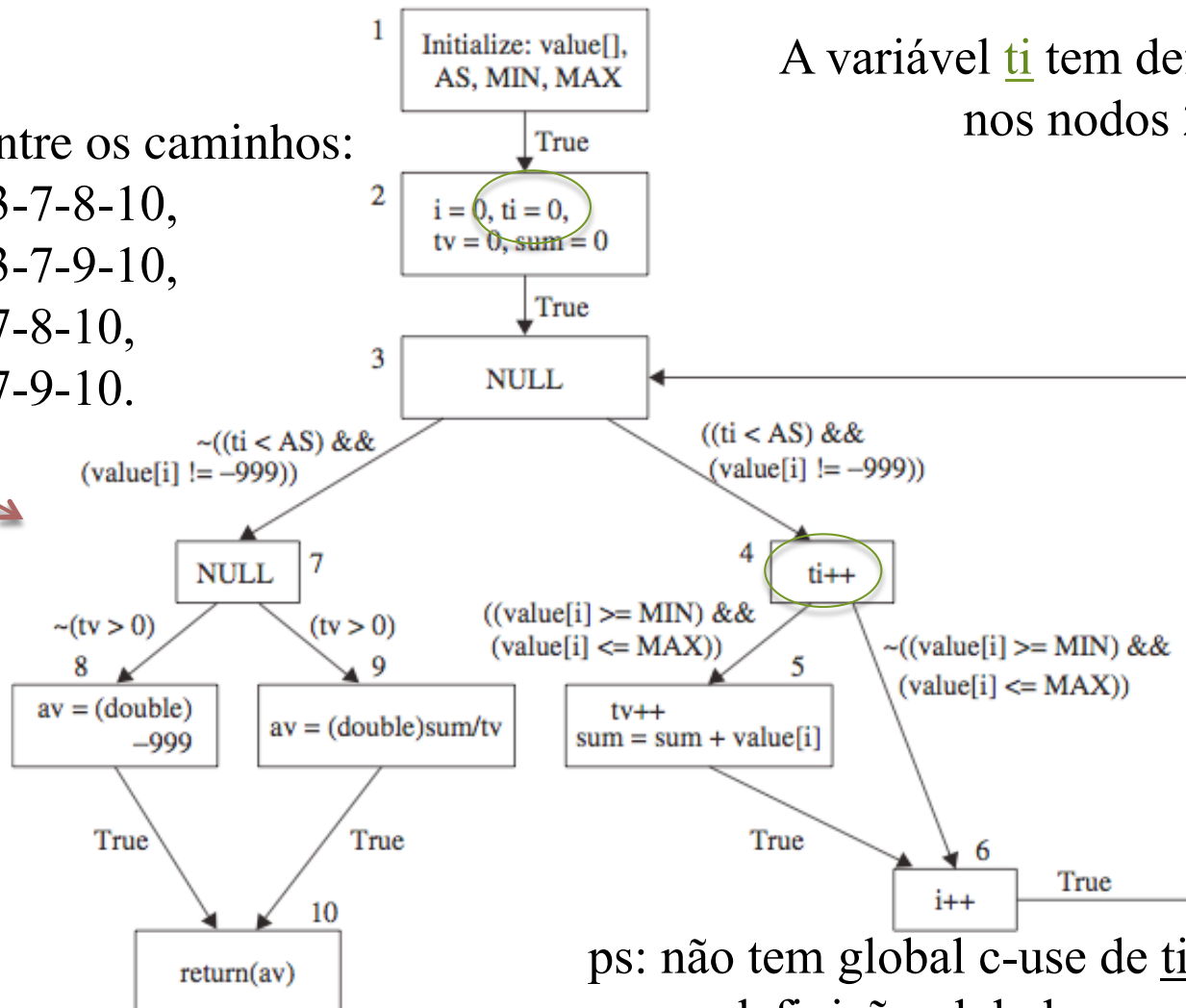
2. All-c-uses

Para cada variável x e para cada nodo i tal que x tem uma definição global no nodo i, selecione caminhos completos que incluam def-clear paths a partir do nodo i até todos os nodos j tal que exista um global c-use de x no nodo j.

Exemplo de All-c-uses

Escolher um entre os caminhos:

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.



A variável ti tem definição global nos nodos 2 e 4.

ps: não tem global c-use de ti a partir da definição global no nodo 4.

Critérios de Seleção de Caminhos

3. All-p-uses:

Para cada variável x e para cada nodo i tal que x tem uma definição global no nodo i , selecione caminhos completos que incluam def-clear paths a partir do nodo i até todas as arestas (i,k) tal que exista um p-use de x na aresta (i,k) .

Exemplo de All-p-uses

Caminhos encontrados:

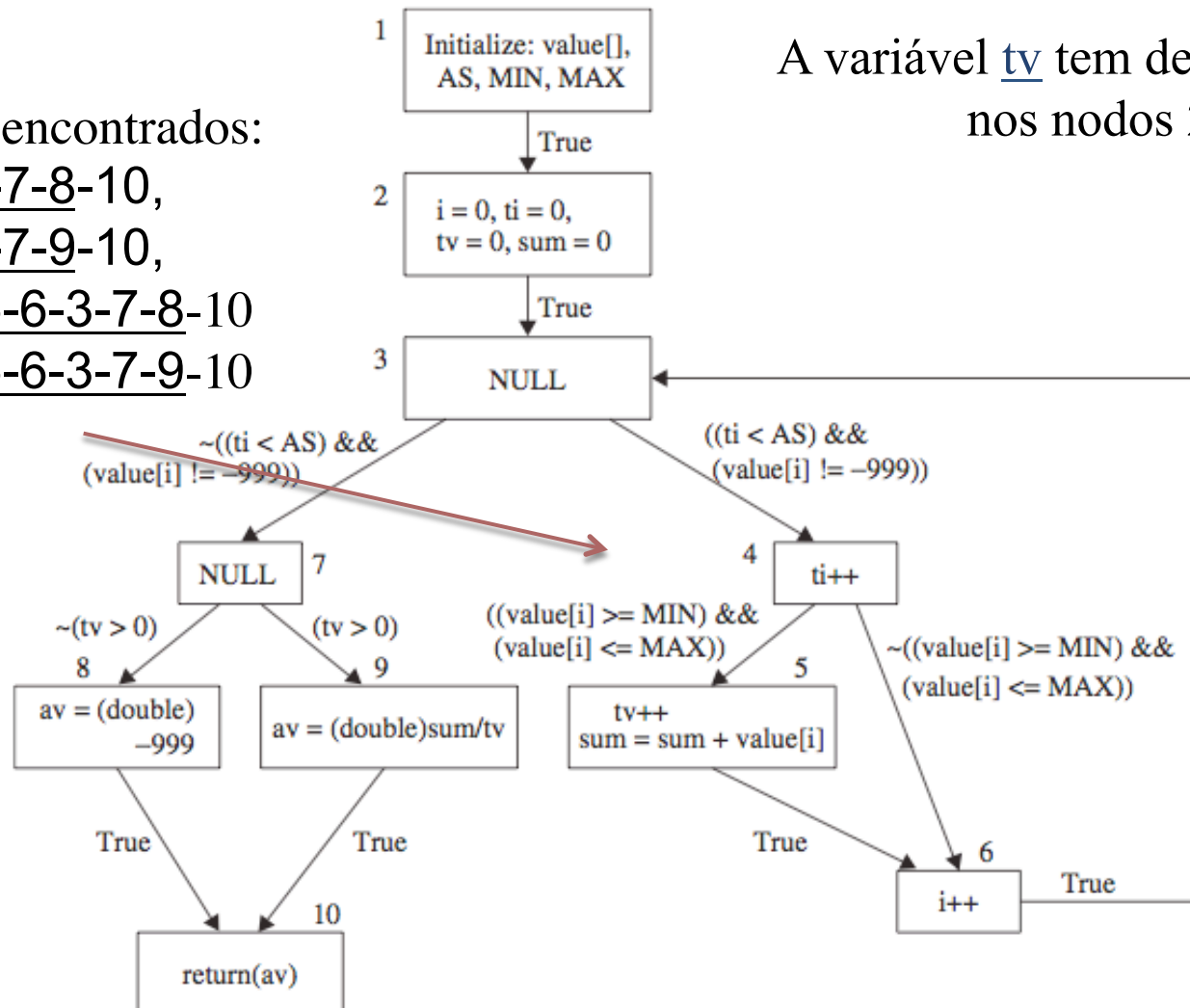
1-2-3-7-8-10,

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

A variável tv tem definição global nos nodos 2 e 5.



Critérios de Seleção de Caminhos

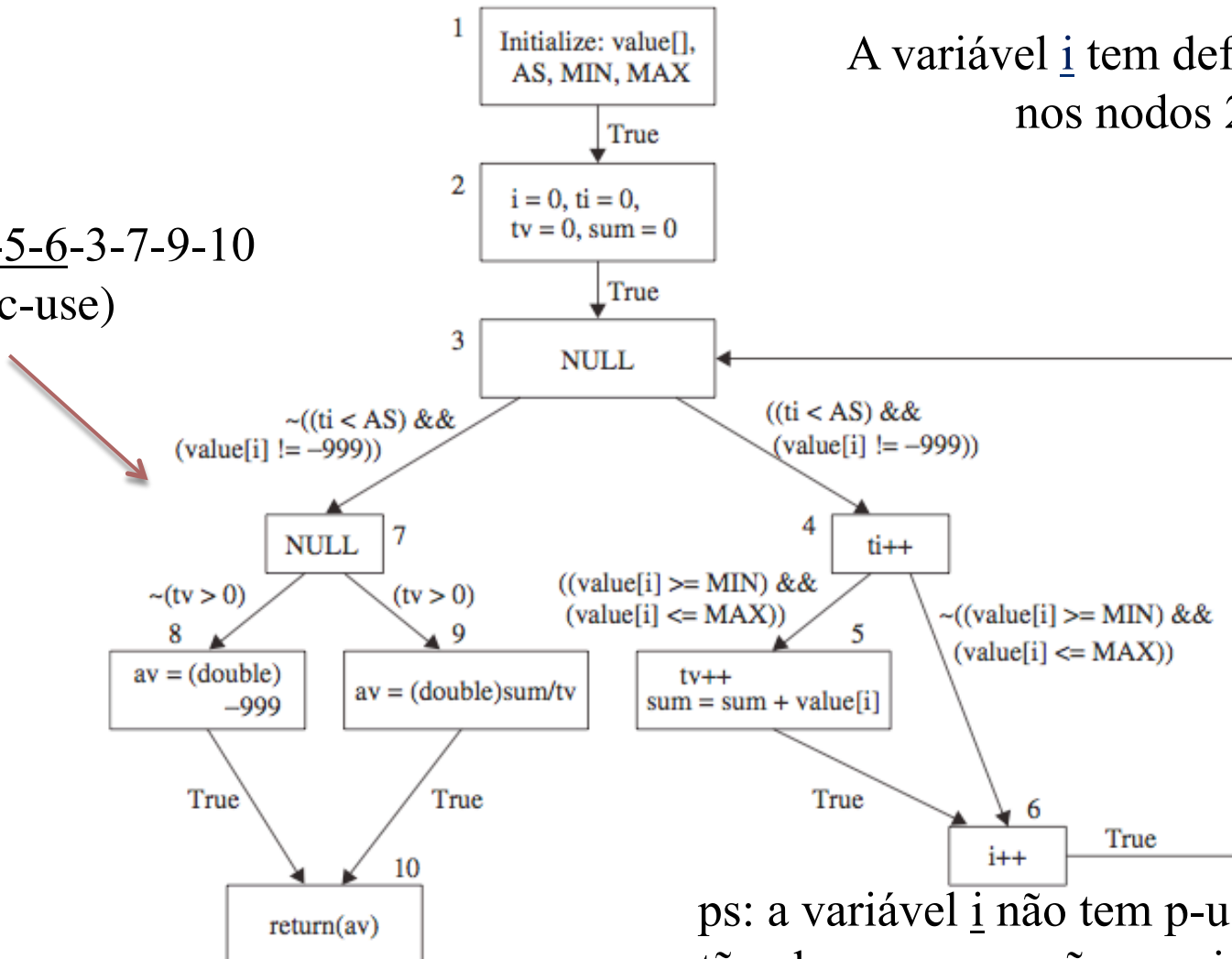
4. All-p-uses/Some-c-uses

Critério idêntico ao all-p-uses **exceto** quando a variável x não tem p-use. **Se** x não tem p-use, o critério é reduzido a some-c-uses.

- Some-c-uses: Para cada variável x e para cada nodo i tal que x tem uma definição global no nodo i, selecione caminhos completos que incluam def-clear paths a partir do nodo i até alguns nodos j tal que exista um global c-use de x no nodo j.

Exemplo de All-p-uses/Some-c-uses

1-2-3-4-5-6-3-7-9-10
(c-use)



A variável i tem definição global nos nodos 2 e 6.

ps: a variável i não tem p-use, então alguns c-uses são considerados.

Critérios de Seleção de Caminhos

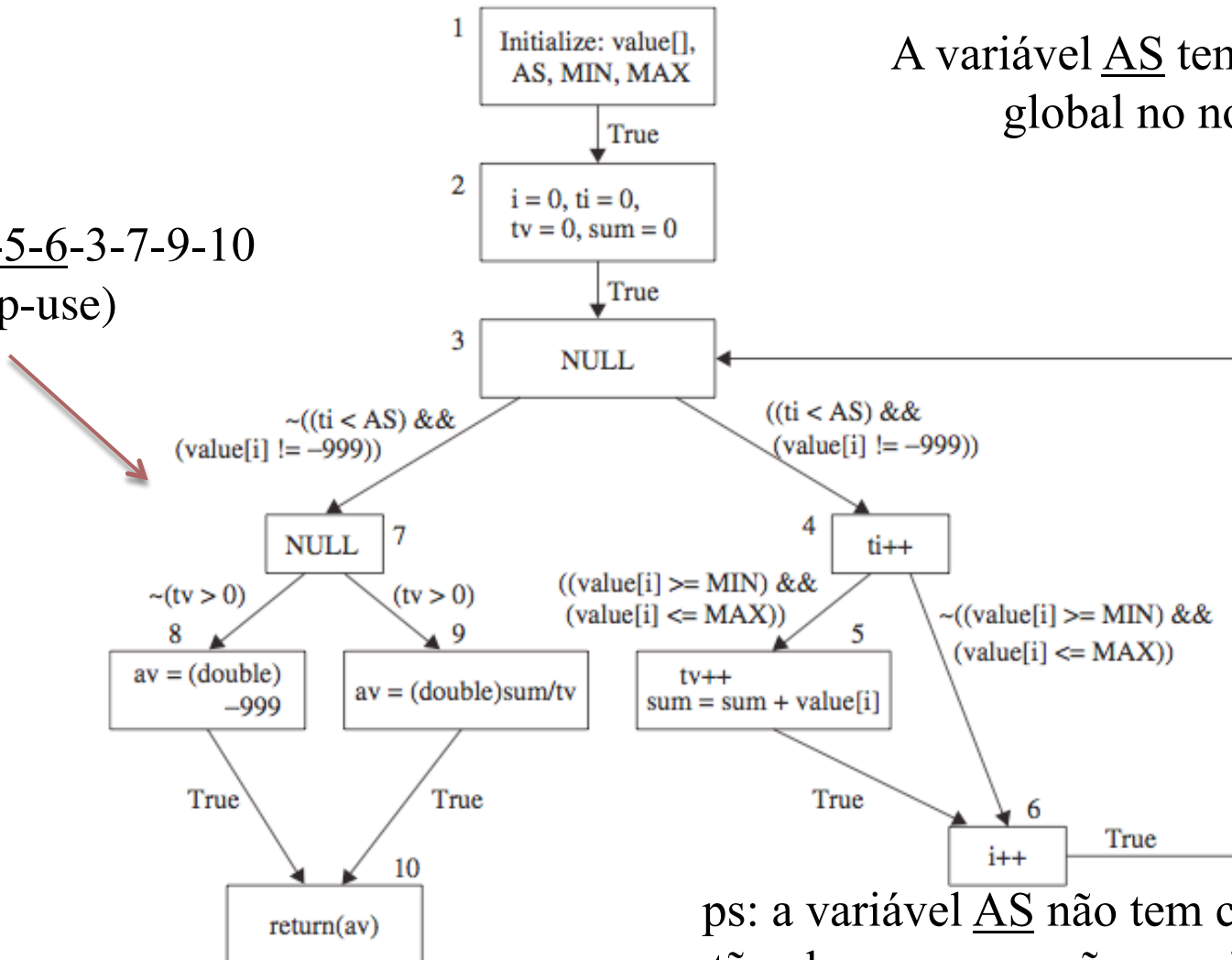
5. All-c-uses/Some-p-uses

Critério idêntico ao all-c-uses **exceto** quando a variável x não tem global c-use. **Se** x não tem global c-use, o critério é reduzido a some-p-uses.

- *Some-p-uses*: Para cada variável x e para cada nodo i tal que x tem uma definição global no nodo i , selecione caminhos completos que incluam def-clear paths a partir do nodo i até algumas arestas (j,k) tal que exista um p-use de x na aresta (j,k) .

Exemplo de All-c-uses/Some-p-uses

1-2-3-4-5-6-3-7-9-10
(p-use)



A variável AS tem definição global no nodo 1.

ps: a variável AS não tem c-use, então alguns p-uses são considerados.

Critérios de Seleção de Caminhos

6. All-uses:

É uma combinação dos critérios all-p-uses e all-c-uses.

7. All-du-paths:

Para cada variável x e para cada nodo i tal que x tem uma definição global no nodo i , selecione caminhos completos que incluam todos os caminhos a partir do nodo i até

- todos os nodos j que tem uma global c-use de x e
- todas as arestas (i,k) que tem um p-use de x .

Teste de Fluxo de Dados

Passos:

1. Desenhe um grafo de fluxo de dados da unidade.
2. Selecione um critério de seleção de fluxo de dados.
3. Identifique os caminhos que satisfazem o critério.
4. Derive expressões de predicado de caminho a partir dos caminhos selecionados.
5. Gere valores de entradas dos testes da unidade que exercitam os caminhos selecionados.

Geração de valores de entrada



Valores de entrada precisam ser selecionados para que os caminhos selecionados sejam executados.