SSC5877
Validação e Teste de Software
SSC0721
Inspeção e Teste de Software

Técnica de Teste Estrutural
Critérios de Fluxo de Dados

Simone Senger de Souza
srocio@icmc.usp.br

ICMC/USP

2020

Aula Anterior

- Técnica de Teste Estrutural -> conhecida como teste caixa-branca
- Baseia-se no conhecimento da estrutura interna (implementação) do programa
  - Teste dos detalhes procedimentais
- A maioria dos critérios dessa técnica utiliza uma representação de programa conhecida como Grafo de Fluxo de Controle (GFC).

Critérios baseados em Fluxo de Dados

- Complementares aos critérios baseados em fluxo de controle

- Objetivo: testar o uso das variáveis em um programa, ou seja, como os dados são usados nas computações

- Utilizam informações do fluxo de dados do programa para determinar os requisitos de teste.
  - Exploram as interações que envolvem definições de variáveis e referências a tais definições.

Anomalias de Fluxo de Dados

- Uso de variável não inicializada
- Atribuição de valor a uma variável mais de uma vez sem que tenha havido uma referência a essa variável entre essas atribuições
- Liberação ou reinicialização de uma variável antes que ela tenha sido criada ou inicializada
- Liberação ou reinicialização de uma variável antes que ela tenha sido usada
- Atribuir novo valor a um ponteiro sem que variável tenha sido liberada

Exemplos de anomalias

```
main(){
    int x;
    if (x==42){...}
}
```

```
main() {
    int x;
    while (x > 0)
      printf ("%d",x);
}
```

Definição e Uso de Variáveis

- Definição (Def): um valor é atribuído a uma variável (diferente de declaração da variável)
  - int x = 10; a = 5;
- Uso Computacional (c-uso): uso da variável em computações
  - a = b * 1; (c-uso de b)
- Uso Predicativo (p-uso): uso da variável em expressões predicativas em estruturas de decisão e repetição
  - if (a >=b) (p-uso de a e b)

Definição e Uso de Variáveis

- E o comando i++? O que ocorre?

Deve-se levar em conta que significa:
i = i +1

def de i e
c-uso de i

Grafo Def-Uso

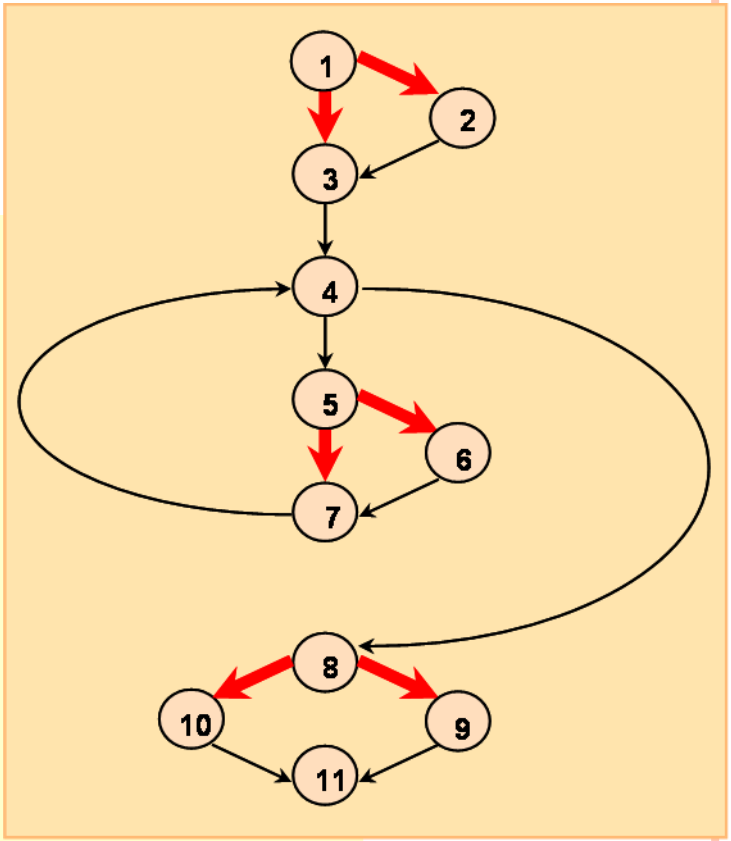
- Critérios de Fluxo de Dados utilizam o Grafo Def-Uso (Def-Use Graph) para derivar os requisitos de teste.
  - Informações a respeito do fluxo de dados do programa.

- Grafo Def-Uso é uma extensão do GFC:

  - GFC + Definição de Variáveis + Uso de Variáveis

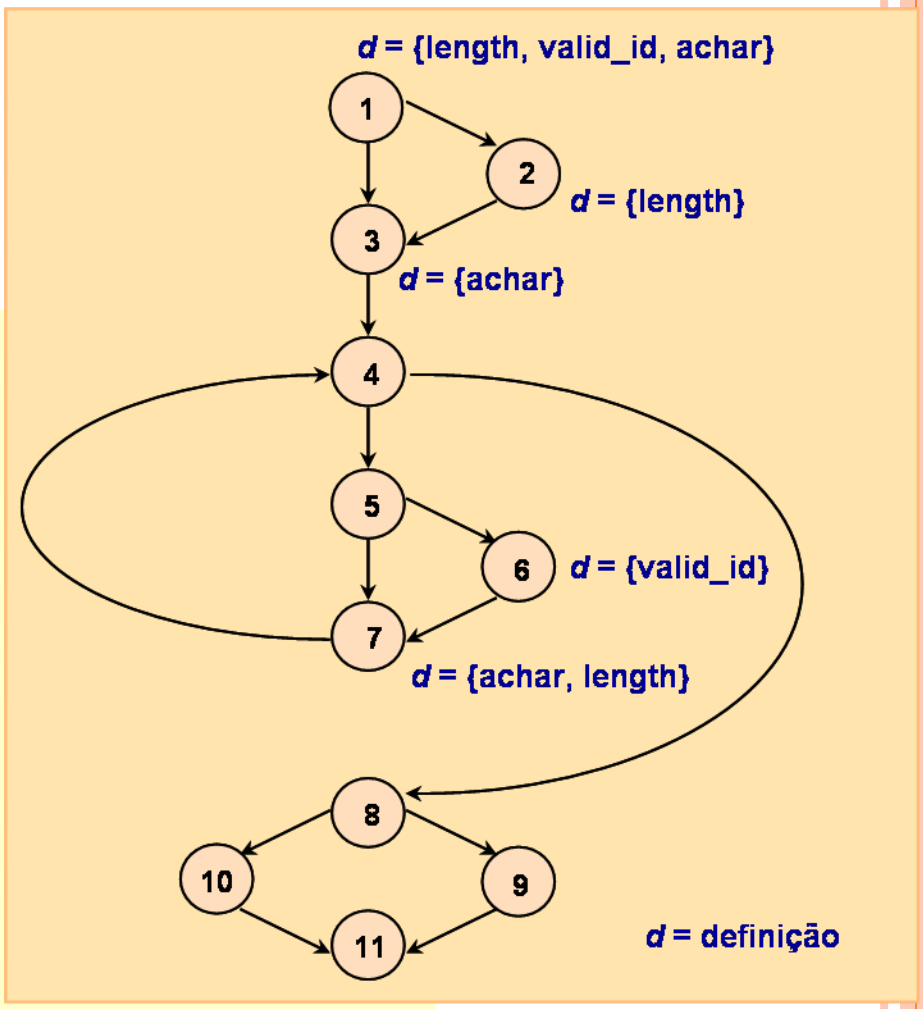
Grafo Def-Uso: Exemplo

```
/* 01 */      {
/* 01 */            char  achar;
/* 01 */            int length, valid_id;
/* 01 */            length = 0;
/* 01 */            printf ("Identificador: ");
/* 01 */            achar = fgetc (stdin);
/* 01 */            valid_id = valid_s(achar);
/* 01 */            if (valid_id)
/* 02 */               length = 1;
/* 03 */            achar = fgetc (stdin);
/* 04 */            while (achar != '\n')
/* 05 */            {
/* 05 */               if (!(valid_f(achar)))
/* 06 */                  valid_id = 0;
/* 07 */               length++;
/* 07 */               achar = fgetc (stdin);
/* 07 */            }
/* 08 */            if (valid_id && (length >= 1) && (length < 6))
/* 09 */               printf ("Valido\n");
/* 10 */            else
/* 10 */               printf ("Invalido\n");
/* 11 */      }
```



Grafo Def-Uso: Exemplo

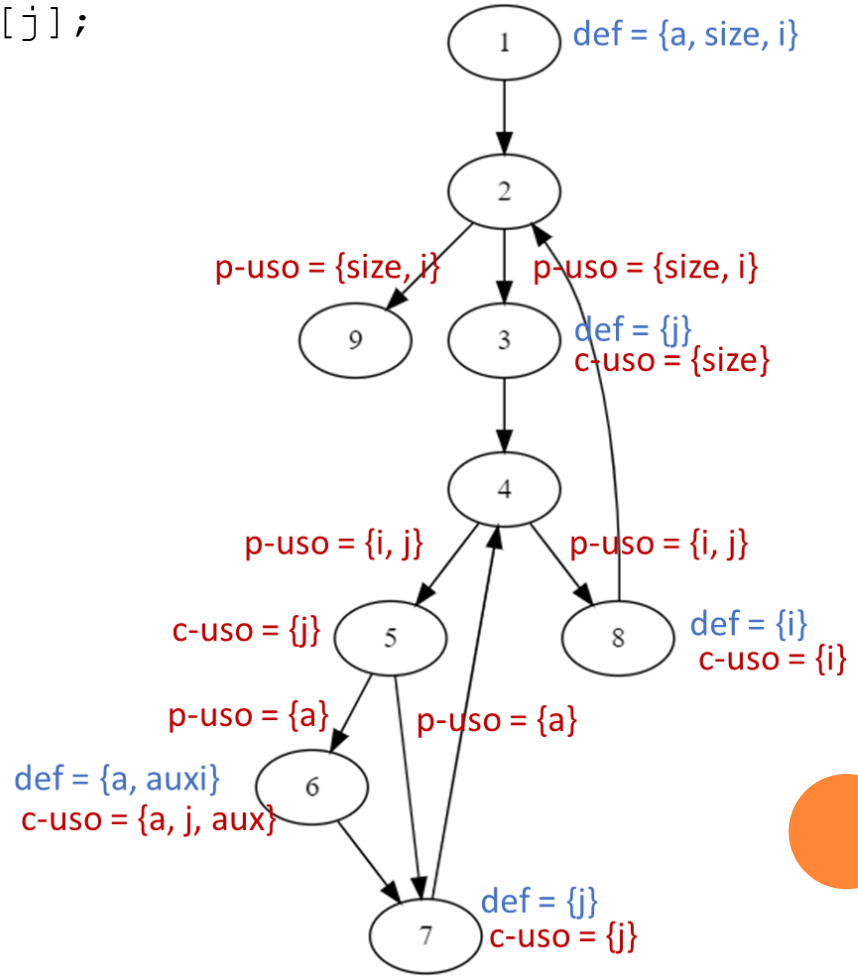
```
/* 01 */     {
/* 01 */           char  achar;
/* 01 */           int length, valid_id;
/* 01 */           length = 0;
/* 01 */           printf ("Identificador: ");
/* 01 */           achar = fgetc (stdin);
/* 01 */           valid_id = valid_s(achar);
/* 01 */           if (valid_id)
/* 02 */              length = 1;
/* 03 */           achar = fgetc (stdin);
/* 04 */           while (achar != '\n')
/* 05 */           {
/* 05 */              if (!(valid_f(achar)))
/* 06 */                 valid_id = 0;
/* 07 */              length++;
/* 07 */              achar = fgetc (stdin);
/* 07 */           }
/* 08 */           if (valid_id && (length >= 1) && (length < 6))
/* 09 */              printf ("Valido\n");
/* 10 */           else
/* 10 */              printf ("Invalido\n");
/* 11 */     }
```



Exercício Fixação – Grafo Def-Uso

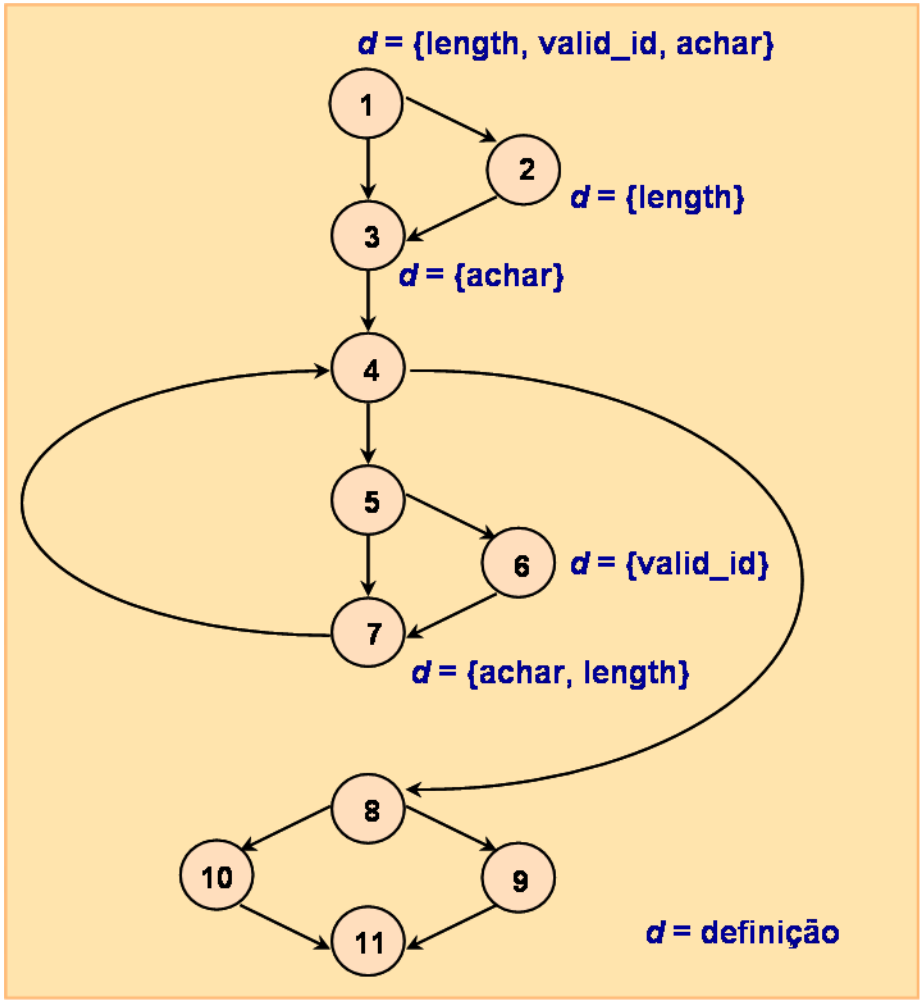
```
void bolha(int a[], int size) {
     int i, j, aux;
     for (i = 0; i < size; i++) {
        for (j = size - 1; j > i; j--) {
           if (a[j - 1] > a[j]) {
              aux = a[j - 1];
              a[j - 1] = a[j];
              a[j] = aux;
           }
        }
     }
  }
```

```
/*1*/void bolha(int a[], int size) {
/*1*/           int i, j, aux;
/*1,2,8*/       for (i = 0; i < size; i++) {
/*3,4,7*/          for (j = size - 1; j > i; j--) {
/*5*/                if (a[j - 1] > a[j]) {
/*6*/                   aux = a[j - 1];
/*6*/                   a[j - 1] = a[j];
/*6*/                   a[j] = aux;
/*7*/                }
/*7*/             }
/*8*/           }
/*9*/     }
```



[Graphviz Online]

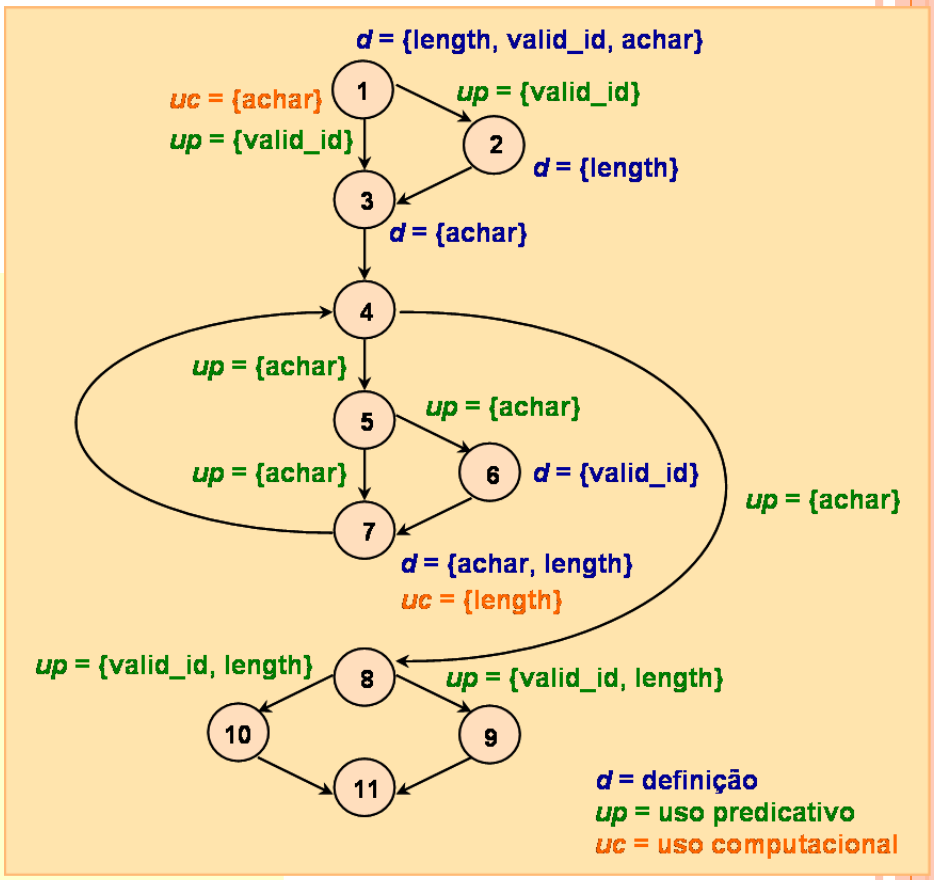
Grafo Def-Uso: Exemplo



Definição de variáveis
def(length) = 1, 2, 7
def(valid_id) = 1, 6
def(achar) = 1, 3, 7

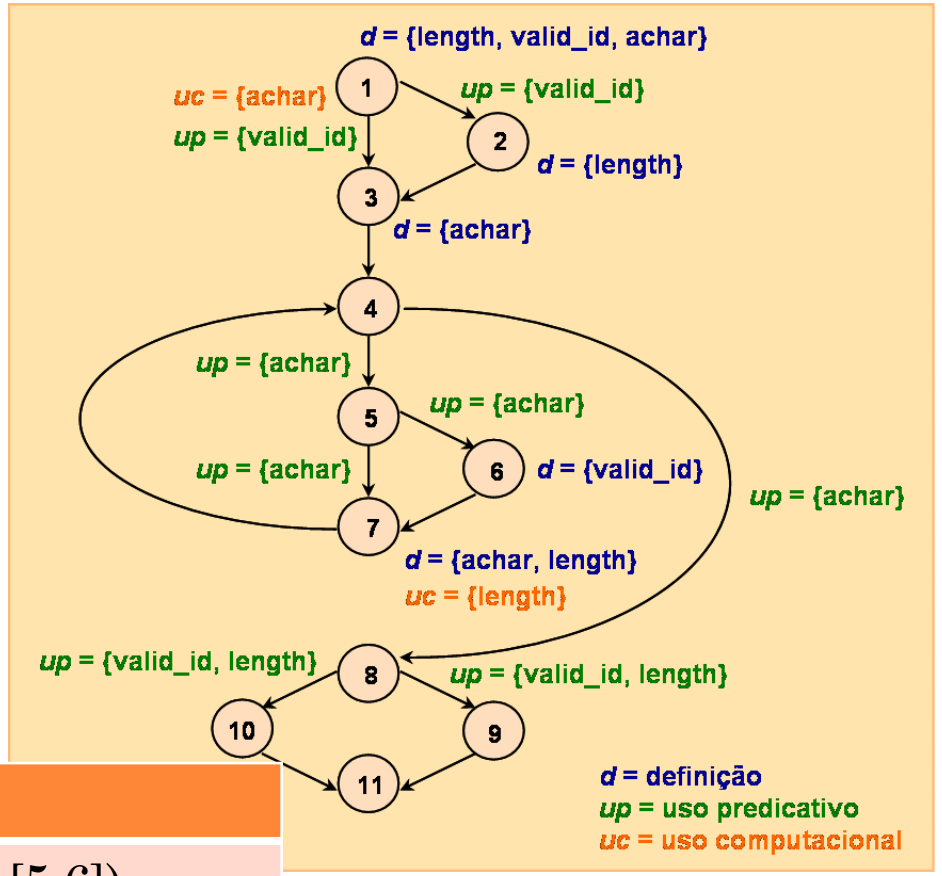
Grafo Def-Uso: Exemplo

```
/* 01 */     {
/* 01 */           char  achar;
/* 01 */           int length, valid_id;
/* 01 */           length = 0;
/* 01 */           printf ("Identificador: ");
/* 01 */           achar = fgetc (stdin);
/* 01 */           valid_id = valid_s(achar);
/* 01 */           if (valid_id)
/* 02 */              length = 1;
/* 03 */           achar = fgetc (stdin);
/* 04 */           while (achar != '\n')
/* 05 */           {
/* 05 */              if (!(valid_f(achar)))
/* 06 */                 valid_id = 0;
/* 07 */              length++;
/* 07 */              achar = fgetc (stdin);
/* 07 */           }
/* 08 */           if (valid_id && (length >= 1) && (length < 6))
/* 09 */              printf ("Valido\n");
/* 10 */           else
/* 10 */              printf ("Invalido\n");
/* 11 */     }
```



Grafo Def-Uso: Exemplo

Importante: Caminho livre de definição



Var.	Par Def-Uso
achar	(1, 1), (1, [4,5]), (1, [4,8]), (1, [5,6]), (1, [5,7]) (3, [4,5]), (3, [4,8]), (3, [5,6]), (3, [5,7]), (7, [4,5]), (7, [4,8]), (7, [5,6]), (7, [5,7]),

Definição e uso de variáveis

- Conceito de caminho livre de definição:
  - A associação entre uma definição de uma variável e seu uso é válida se for possível estabelecer um caminho entre a def e o uso onde a variável não é redefinida.
  - Esse caminho é chamado caminho livre de definição

Critérios Baseados em Fluxo de dados

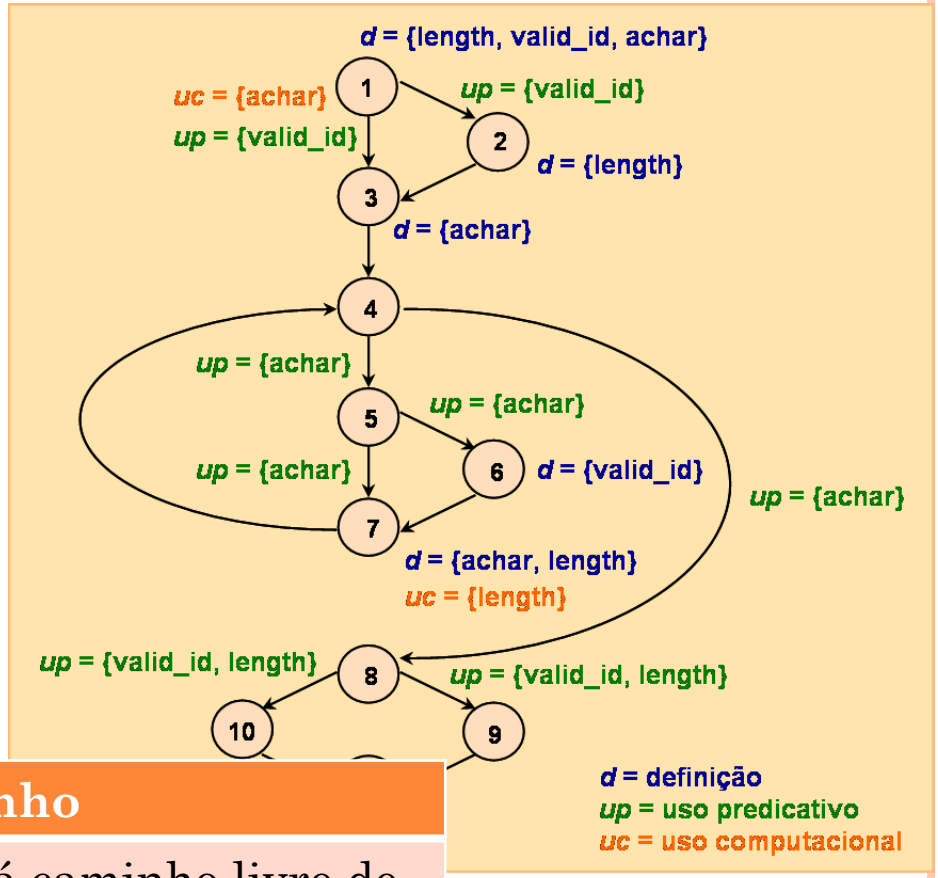
- Principais critérios:
  - Todas-Defs
  - Todos-Usos
  - Todos-DU-Caminhos

- Variações:
  - Todos-P-Usos
  - Todos-P-Usos/Alguns-C-Usos
  - Todos-C-Usos/Alguns-P-Usos

Critérios Baseados em Fluxo de dados

- Critério Todas-Defs:
  - Requer que cada definição de variável seja exercitada ao menos uma vez, por um c-uso ou por um p-uso
  - Para todas as definições de variáveis, o seu uso deve ser alcançado por um caminho livre de definição.

Exemplo – Critério Todas-Defs

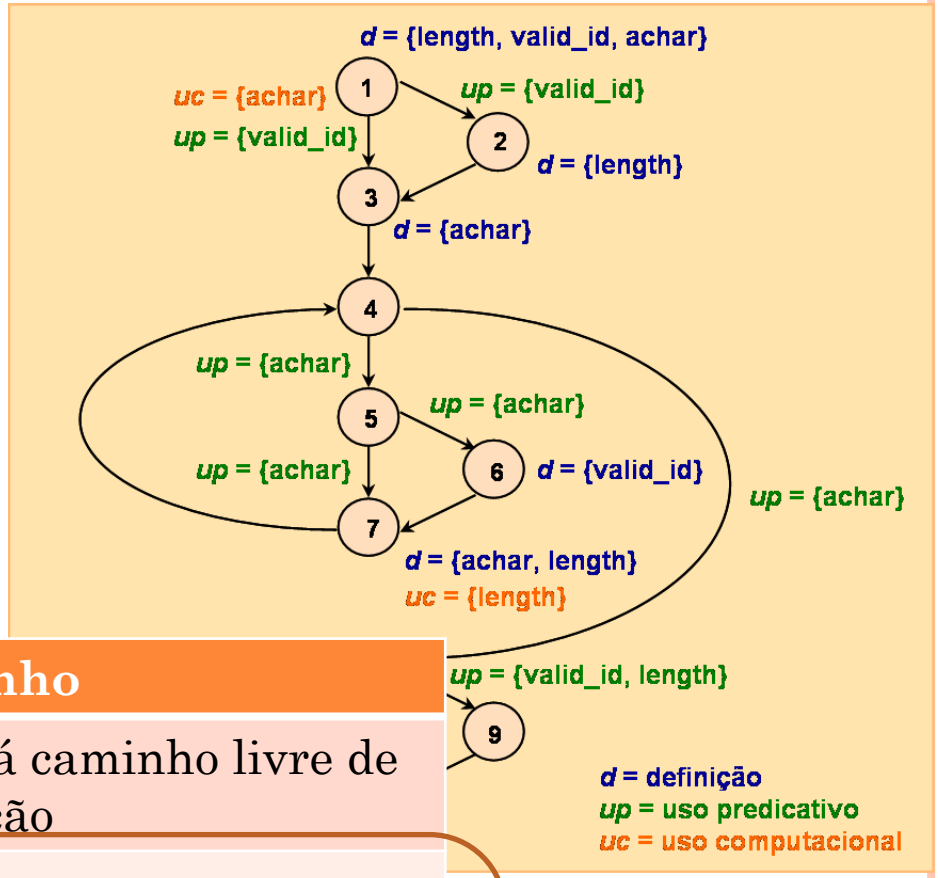


Var	Def	Uso	Caminho
achar	1	(4,5)	não há caminho livre de definição
achar	1	1	1, 2, 3...
achar	3	(4,5)	1, 2, 3, 4, 5, 7 ...
achar	3	(5,7)	1, 2, 3, 4, 5, 7 ...
....			

Critérios Baseados em Fluxo de dados

- Critério Todos-Usos:
  - Requer que todos os usos (c-usos e p-usos) de uma variável definida sejam executados

Exemplo – Critério Todos-Usos

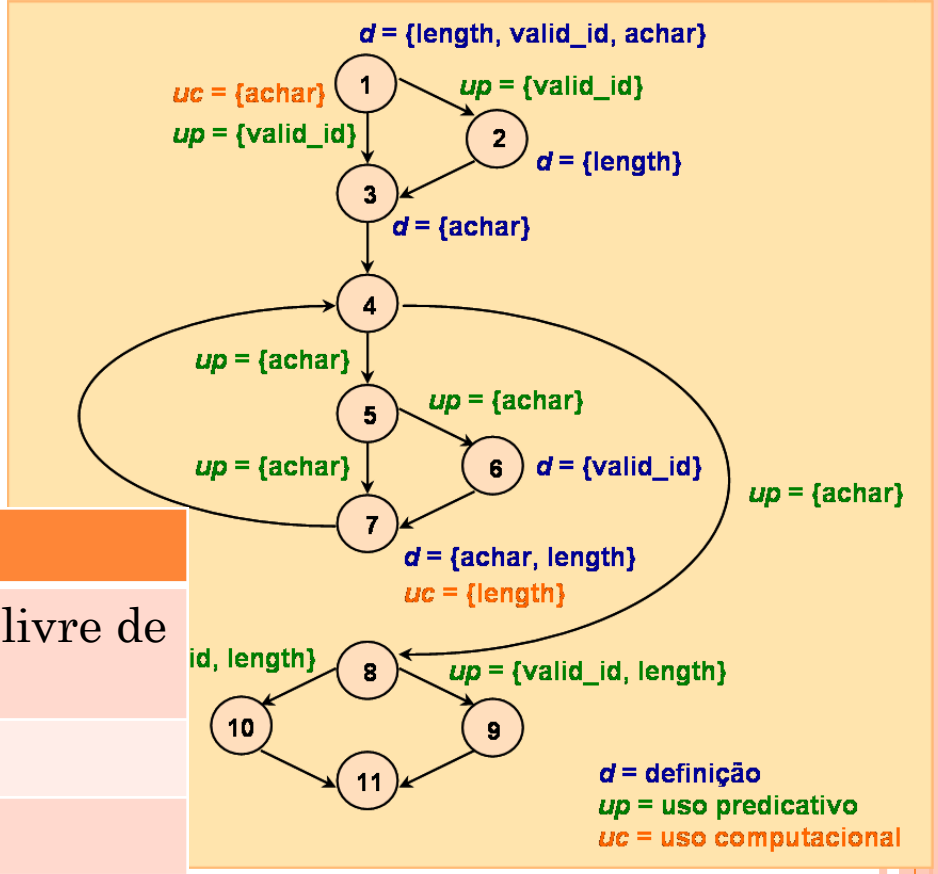


Var	Def	Uso	Caminho
achar	1	(4,5)	não há caminho livre de definição
achar	1	1	1, 2, 3...
achar	3	(4,5)	1, 2, 3, 4, 5, 7 ...
achar	3	(4,8)	1, 2, 3, 4, 8 ...
achar	3	(5,7)	1, 2, 3, 4, 5, 7 ...
achar	3	(5,6)	1, 2, 3, 4, 5, 6, 7 ...

Critérios Baseados em Fluxo de dados

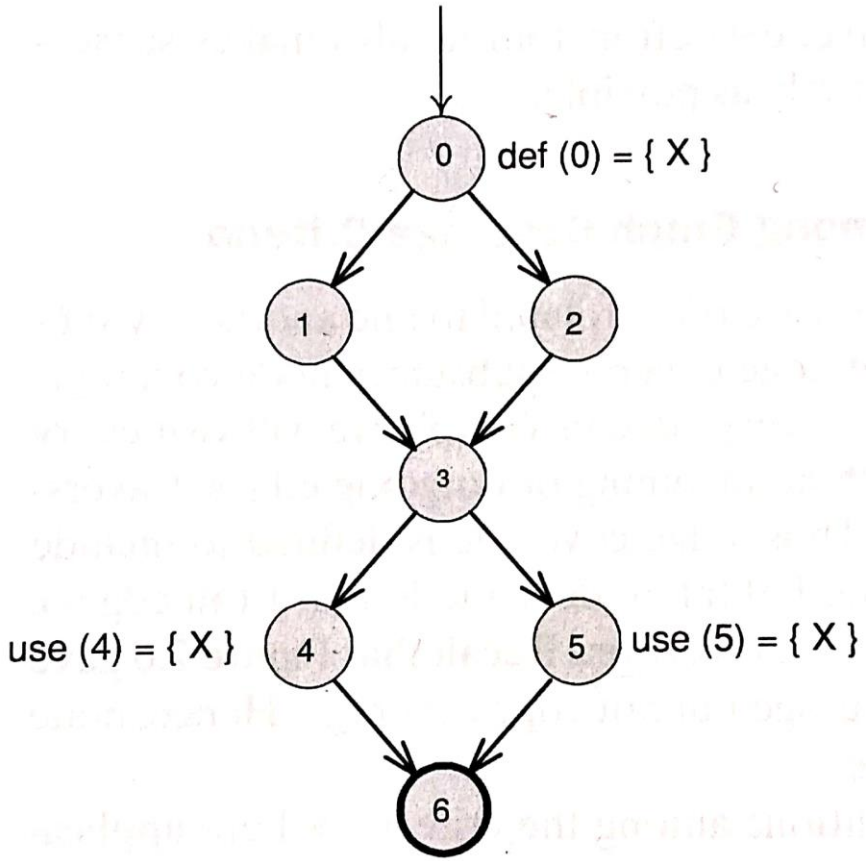
- Critério Todos-DU-Caminhos:
  - Requer que todos os usos (c-usos e p-usos) de uma variável definida sejam executados por todos os caminhos livres de definição e livres de laço que executem os pares def-uso

Exemplo – Critério Todos-DU-Caminhos



Var	Def	Uso	Caminho
achar	1	(4,5)	não há caminho livre de definição
achar	1	1	1, 2, 3...
achar	3	(4,5)	1, 2, 3, 4, 5, 7 ...
achar	3	(4,5)	1, 3, 4, 5, 7 ...
achar	3	(4,8)	1, 2, 3, 4, 8 ...
achar	3	(4,8)	1, 3, 4, 8 ...
achar	3	(5,7)	1, 2, 3, 4, 5, 7 ...
achar	3	(5,7)	1, 3, 4, 5, 7 ...
achar	3	(5,6)	1, 2, 3, 4, 5, 6, 7 ...
achar	3	(5,6)	1, 3, 4, 5, 6, 7 ...

Exemplo Critérios Estruturais



All-defs
0-1-3-4

All-uses
0-1-3-4
0-1-3-5

All-du-paths
0-1-3-4
0-1-3-5
0-2-3-4
0-2-3-5

Figure 2.14. Example of the differences among the three data flow coverage criteria.

Critérios Baseados em Fluxo de Dados

- Outros critérios:

Critério	Descrição
Todos-P-Usos	Requer a execução de todos os p-usos associados a uma definição de variável
Todos-P-Usos/Alguns-C-Usos	Requer a execução de todos os p-usos e alguns c-usos associados a uma definição de variável
Todos-C-Usos/Alguns-P-Usos	Requer a execução de todos os c-usos e alguns p-usos associados a uma definição de variável

Teste Estrutural - Hierarquia

Propriedades Mínimas de um Critério de Teste:

- Garantir, do ponto de vista de fluxo de controle, a cobertura de todos os desvios condicionais.
  - Incluir o critério Todos-Arcos.
- Requerer, do ponto de vista de fluxo de dados, ao menos um uso de todo resultado computacional.
  - Incluir o critério Todas-Defs.
- Requerer um conjunto de casos de teste finito.

Concluindo

- Critérios Baseados em Fluxo de Dados identificam pares def-uso para as variáveis de um programa.
  - *"Assim como um programa em que os testes não executaram todas as instruções dele é visto como um programa não confiável, o mesmo sentimento de não confiabilidade deve ocorrer caso os testes não observem todo o efeito de usar o valor produzido por toda e qualquer computação do programa" (Rapps e Weyuker, 1982)*
- Uso do Grafo Def-Uso para representação do programa.
- Limitação: caminhos não executáveis