



Critérios de Fluxo de Dados

Teste Caixa-Branca

Auri Marcelo Rizzo Vincenzi¹, Márcio Eduardo Delamaro² e José Carlos Maldonado²

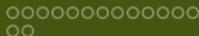
¹Departamento de Computação
Universidade Federal de São Carlos

²Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

Este material pode ser utilizado livremente respeitando-se a licença Creative Commons: Atribuição – Uso Não Comercial – Compartilhamento pela mesma Licença (by-nc-sa).



[Ver o Resumo da Licença](#) | [Ver o Texto Legal](#)



Organização

Fluxo de Dados

Técnica de Aplicação

Definições

Exemplo de Aplicação

Critérios de Fluxo de Dados

Critérios de Rapps e Weyuker

Critérios Potenciais-Usos

Relação de Inclusão

Aplicabilidade e Limitações

Resumo

Exercício

Leitura Recomendada



Fluxo de Dados Técnica de Aplicação Definições

Exemplo de Aplicação

Critérios de Fluxo de Dados
Critérios de Rapps e Weyuker
Critérios Potenciais-Usos
Relação de Inclusão
Aplicabilidade e Limitações

Resumo

Exercício

Leitura Recomendada

Introdução (1)

- ▶ Critérios pertencentes à Técnica de Teste Caixa Branca.
- ▶ Complementares aos critérios baseados em fluxo de controle.
- ▶ Busca testar o uso das variáveis em um programa, ou seja, como os dados são usados nas computações.



Introdução (2)

Exemplo de defeito em fluxo de dados:

```
1 main() {  
2     int x;  
3     if (x==42) {...}  
4 }
```

- ▶ Engano: referenciar uma variável sem esta ter sido inicializada.
 - ▶ Assumir que o compilador inicializa a variável com algum valor padrão quando ele não o faz. Qual a saída do programa abaixo?

```
1 #include <stdio.h>  
2 main() {  
3     int x;  
4     printf ("%d", x);  
5 }
```



Introdução (3)

- ▶ Teste de fluxo de dados é uma ferramenta poderosa para o uso incorreto de valores resultante de erros de codificação;
- ▶ Um dos primeiros trabalhos sobre o tema foi publicado por (Herman, 1976);
- ▶ Tornou-se popular com a publicação do trabalho de Rapps e Weyuker (1982):

“It is our belief that, just as one would not feel confident about a program without executing every statement in it as part of some test, one should not feel confident about a program without having seen the effect of using the value produced by each and every computation.”

Introdução (4)

Princípio da Definição dos Critérios (Martins, 2003)

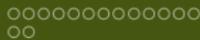
Programa = sequência de ações realizadas sobre variáveis



Fluxo de controle



Informações sobre onde variáveis são definidas e onde essas definições são usadas.



Introdução (5)

- ▶ Qual o modelo de falhas?
 - ▶ Anomalias de fluxo de dados ([Fosdick e Osterweil, 1976](#); [Hecht, 1977](#)):
 - ▶ 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.



Ciclo de Vida de Variáveis (1)

- ▶ Variáveis são criadas, usadas e destruídas.
- ▶ Em algumas linguagens de programação (BASIC e FORTRAN, por exemplo) a criação e destruição são automáticas.
- ▶ Em outras (C, C++ e Java, por exemplo), a criação deve ser explícita.
- ▶ Exemplos de declaração:

```
1 int x;           // x é criada como um inteiro  
2 String y; // y é criada como uma String
```

Ciclo de Vida de Variáveis (2)

- ▶ Declarações, em geral, ocorrem dentro de um bloco.
 - ▶ Variáveis são criadas quando a definição das mesmas são executadas.
 - ▶ Variáveis são destruídas no final do bloco (conceito de escopo da variável).

```
1 { // início do bloco 1
2   int x; // x é definida no bloco 1
3   ...; // x pode ser usada aqui
4   { // início do bloco 2
5     int y; // y é definida no bloco 2
6     ...; // x e y podem ser usadas aqui
7   } // y é automaticamente destruída no final do bloco 2
8   ...; // x ainda pode ser usada daqui em diante
9 } // x é automaticamente destruída no final do bloco 1
```



Tipos de Uso

- ▶ Existem dois tipos de **uso** de variáveis:
 - ▶ Uso em computações, denominados **uso computacional**. Por exemplo: $a = b * 1$.
 - ▶ Uso em condições, denominado **uso predicativo**. Por exemplo: `if (a >= b)`.
- ▶ Independentemente do tipo de uso, é imprescindível que antes de ser usada a variável tenha sido **definida**.
 - ▶ A **definição** de uma variável ocorre quando ela recebe um valor. Por exemplo, via comando de atribuição: $a = 10$ e $b = 5$.



Tipos de Ocorrência de Variáveis (1)

- ▶ Considere a seguinte notação:
 - \sim não existência da variável
 - d definição de variável
 - u uso de variável
 - k destruição de variável
- ▶ Existem três possibilidades de ocorrência em um dado caminho do programa:
 - $\sim d$ variável não existe e é definida (correto)
 - $\sim u$ variável não existe e é usada (incorreto)
 - $\sim k$ variável não existe e é destruída (provavelmente incorreto)

Tipos de Ocorrência de Variáveis (2)

- ▶ Ocorrências no decorrer do programa:

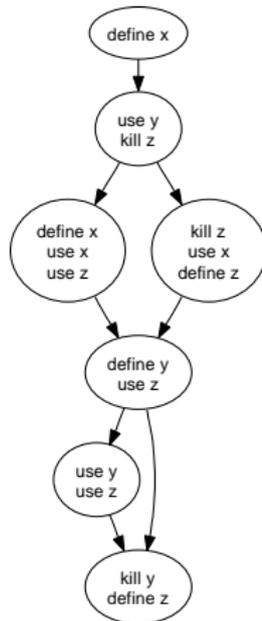
<i>dd</i>	variável definida e redefinida	(provavelmente incorreto)
<i>du</i>	variável definida e usada	(correto)
<i>dk</i>	variável definida e destruída	(provavelmente incorreto)
<i>ud</i>	variável usada e definida	(aceitável)
<i>uu</i>	variável usada e reusada	(aceitável)
<i>uk</i>	variável usada e destruída	(aceitável)
<i>kd</i>	variável destruída e redefinida	(aceitável)
<i>ku</i>	variável destruída e reusada	(incorreto)
<i>kk</i>	variável destruída e redestruída	(provavelmente incorreto)

Grafo Definição-Uso (1)

- ▶ Para avaliar os diferentes estados das variáveis no programa é utilizado um grafo denominado **Grafo Definição-Uso** ou Grafo Def-Uso ([Rapps e Weyuker, 1982](#)).
- ▶ Semelhante ao Grafo de Fluxo de Controle.
- ▶ Inclui ainda variáveis definidas, usadas e destruídas em cada nó.
- ▶ Análise estática e dinâmica do Grafo Def-Uso.
 - ▶ Estático: examinar o grafo procurando por problemas, simulando mentalmente o comportamento.
 - ▶ Dinâmico: executar o programa com casos de testes e avaliar o resultado.

Grafo Definição-Uso (2)

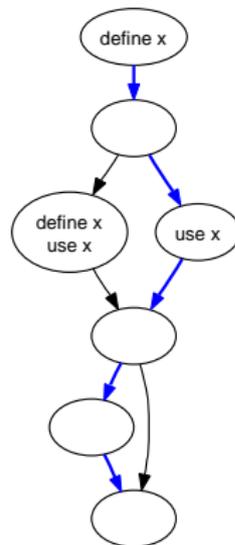
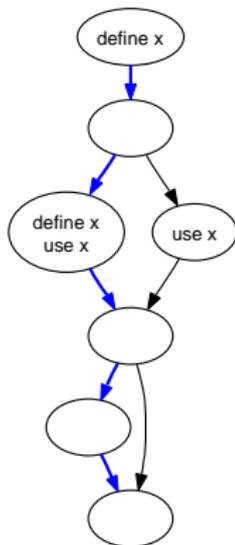
- Exemplo de GFC com anotações de fluxo de dados.



Grafo Definição-Uso: Análise Estática (1)

Examinar o padrão definição-uso-destruição de cada variável ao longo do GDU. Padrões observados em relação à x :

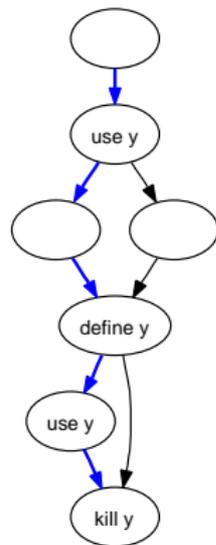
$\sim d$ correto
 dd suspeito?
 du normal



Grafo Definição-Uso: Análise Estática (2)

Examinar o padrão definição-uso-destruição de cada variável ao longo do GDU. Padrões observados em relação à y :

$\sim u$ erro grave
 ud aceitável
 du correto
 uk aceitável
 dk provável erro

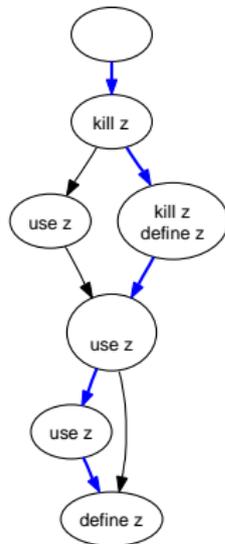
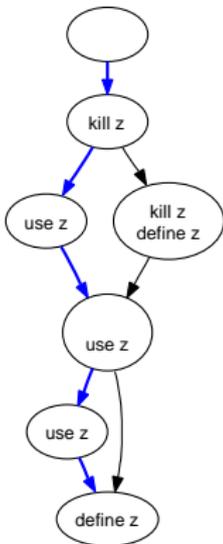




Grafo Definição-Uso: Análise Estática (3)

Examinar o padrão definição-uso-destruição de cada variável ao longo do GDU. Padrões observados em relação à *z*:

$\sim k$ erro de programação
 ku erro grave
 uu correto
 ud aceitável
 kk provável erro
 kd aceitável
 du correto





Grafo Definição-Uso: Análise Estática (4)

- ▶ Resultados obtidos com a Análise Estática:

<i>x</i>	define-define	<i>dd</i>
<i>y</i>	~-use	<i>~ u</i>
<i>y</i>	define-kill	<i>dk</i>
<i>z</i>	~-kill	<i>~ k</i>
<i>z</i>	kill-use	<i>ku</i>
<i>z</i>	kill-kill	<i>kk</i>



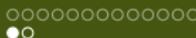
Grafo Definição-Uso: Análise Estática (5)

- ▶ Nem todos tipos de erros podem ser detectados por meio de análise estática.
 - ▶ Coleções do tipo `int vetor[100];`
Destruído como uma unidade, referenciado individualmente.
Em geral referenciado como `vetor[j]`. Difícil capturar erros com análise estática.
 - ▶ Embora possa existir associações inválidas que existam estaticamente, na prática, tais caminhos podem ser não-executáveis.
 - ▶ Interrupção de processos pode dificultar a análise estática do GDU.
- ▶ Para os casos acima é necessário realizar a análise dinâmica das associações de fluxo de dados.



Grafo Definição-Uso: Análise Dinâmica

- ▶ Assumir que o fluxo de controle do módulo está correto.
- ▶ Criar casos de testes de modo que:
 - ▶ Cada definição de variável é rastreada até cada um de seus usos.
 - ▶ Cada uso é rastreado a partir de sua definição correspondente.
- ▶ Para fazer isso:
 - ▶ Enumerar os caminhos no GDU usando a mesma abordagem do teste de fluxo de controle.
 - ▶ Criar casos de testes que cubram cada par “definição-uso” (associações de fluxo de dados) entre as variáveis.



Definições (1)

- ▶ A ocorrência de variáveis em um programa pode ser classificada em:
 - ▶ **Definição** (*def* ou *d*): ocorre quando uma variável recebe um valor.
$$a = 1 \text{ ou } \text{read}(a)$$
 - ▶ **Uso**: ocorre quando a variável é referenciada e tem o seu valor consultado. Um uso pode ser:
 - ▶ **Computacional** (*c-uso* ou *uc*): a variável é utilizada em uma computação.
$$b = a * 2$$
 - ▶ **Predicativo** (*p-uso* ou *up*): a variável é utilizada em uma condição.

$$\text{if } (a > 0)$$



Definições (2) (Martins, 2003)

- ▶ **c-uso global** quando não existe *def* da variável no bloco em que ocorre o *c-uso*.
- ▶ **caminho livre de definição**: um caminho $(n_1, n_2, \dots, n_k, n_j)$ é livre de definição com respeito a uma variável x (c.l.d.(x)) definida no nó n_1 em relação ao nó n_k ou arco (n_k, n_j) se a variável x não é redefinida no caminho (n_2, \dots, n_k) ou seja, não existe redefinição de x nos nós pertencentes ao caminho (n_2, \dots, n_k) .
- ▶ **def global**: quando a *def* de uma variável x em um bloco B_1 é usada em um bloco B_2 (ou em um predicado) sem que haja redefinição de x entre os blocos B_1 e B_2 ou existe um c.l.d.(x) entre B_1 e B_2 .



Fluxo de Dados
Técnica de Aplicação
Definições

Exemplo de Aplicação

Critérios de Fluxo de Dados
Critérios de Rapps e Weyuker
Critérios Potenciais-Usos
Relação de Inclusão
Aplicabilidade e Limitações

Resumo

Exercício

Leitura Recomendada



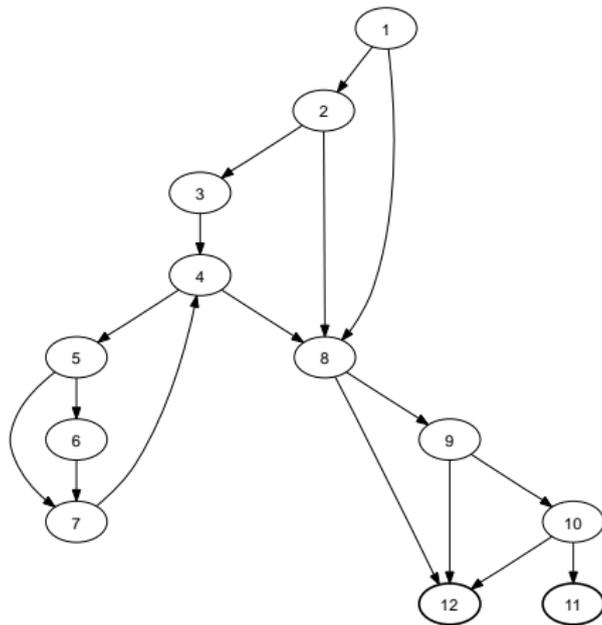
Programa Identifier: Implementação em Java

```

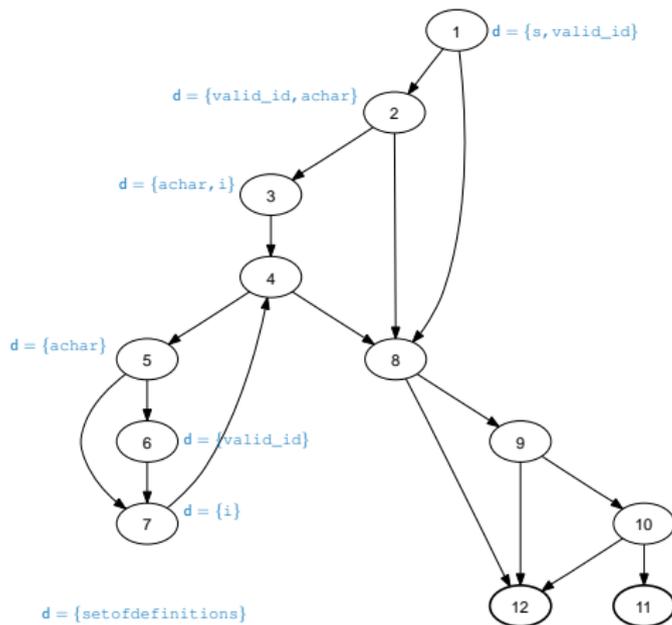
4  public boolean validateIdentifier(String s) {
5      char achar;
6      /*01*/ boolean valid_id = false;
7      /*01*/ if (s.length() > 0) {
8          /*02*/     achar = s.charAt(0);
9          /*02*/     valid_id = valid_s(achar);
10         /*02*/     if (s.length() > 1) {
11             /*03*/     achar = s.charAt(1);
12             /*03*/     int i = 1;
13             /*04*/     while (i < s.length() - 1) {
14                 /*05*/     achar = s.charAt(i);
15                 /*05*/     if (!valid_f(achar))
16                     /*06*/     valid_id = false;
17                 /*07*/     i++;
18             }
19         }
20     }
21     /*08*/     /*09*/     /*10*/
22     if (valid_id && (s.length() >= 1) && (s.length() < 6))
23     /*11*/     return true;
24     else
25     /*12*/     return false;
26 }

```

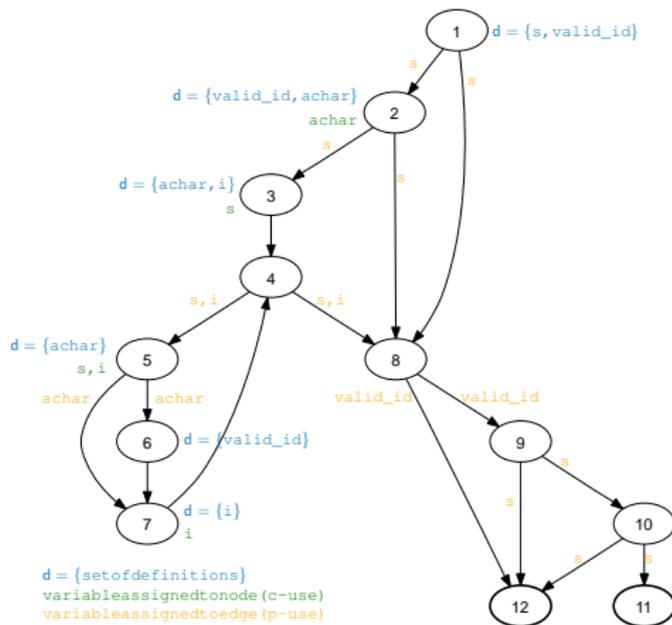
Programa Identifier: GFC

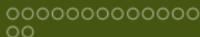


Programa Identifier: GFC + Definições



Programa Identifier: GFC + Definições + Usos





Fluxo de Dados
Técnica de Aplicação
Definições

Exemplo de Aplicação

Critérios de Fluxo de Dados
Critérios de Rapps e Weyuker
Critérios Potenciais-Usos
Relação de Inclusão
Aplicabilidade e Limitações

Resumo

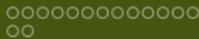
Exercício

Leitura Recomendada



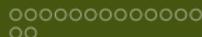
Critérios de Rapps e Weyuker (1982) (1)

- ▶ Baseiam-se no Grafo Def-Usos para derivar os requisitos de testes.
- ▶ Objetivos: exercitar caminhos ligando definições globais a usos globais de variáveis do programa.
- ▶ Tipos:
 - ▶ todas as definições.
 - ▶ todos os p-usos.
 - ▶ todos os p-usos e alguns c-usos.
 - ▶ todos os c-usos e alguns p-usos.
 - ▶ todos os usos.



Critérios de Rapps e Weyuker (1982) (2)

Critério	Descrição
todas-defs	para todas as definições de variáveis deve ser exercitado um caminho livre de definição para um de seus usos.
todos-p-usos	para todas as definições de variáveis deve ser exercitado um caminho livre de definição para todos os seus p-usos.
todos-p-usos- alguns-c-usos	para todas as definições de variáveis deve ser exercitado um caminho livre de definição para todos os seus p-usos e alguns c-usos.
todos-c-usos- alguns-p-usos	para todas as definições de variáveis deve ser exercitado um caminho livre de definição para todos os seus c-usos e para alguns p-usos.
todos-usos	para todas as definições de variáveis deve ser exercitado um caminho livre de definição para todos os seus c-usos e para todos os seus p-usos.

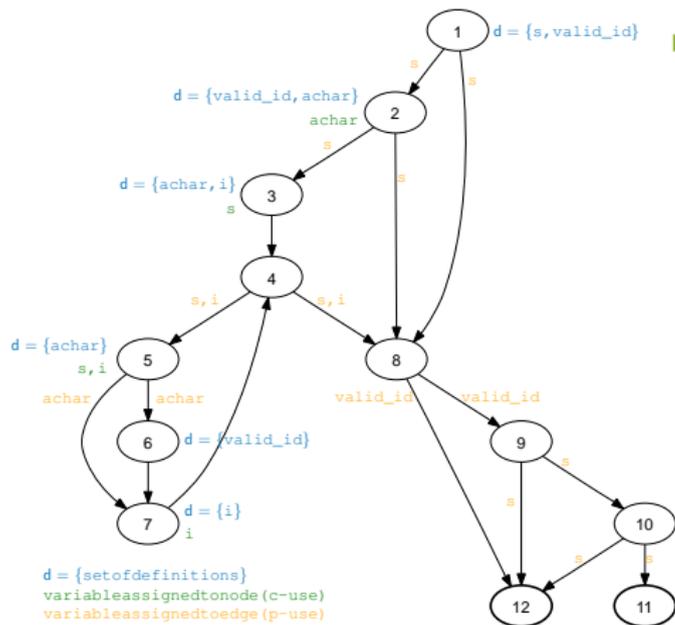


Requisitos de Teste de Fluxo de Dados (1)

- ▶ Requisitos gerados pelos critérios de fluxo de dados:
- ▶ **Associações Definição-Uso**
 - ▶ Em geral uma tripla: $\langle var, def, uso \rangle$, sendo:
 - ▶ *var* - variável para a qual a associação definição-uso foi estabelecida.
 - ▶ *def* - nó contendo a definição da variável *var*.
 - ▶ *uso* - nó/arco com uso computacional/predicativo de *var*.



Requisitos de Teste de Fluxo de Dados (2)



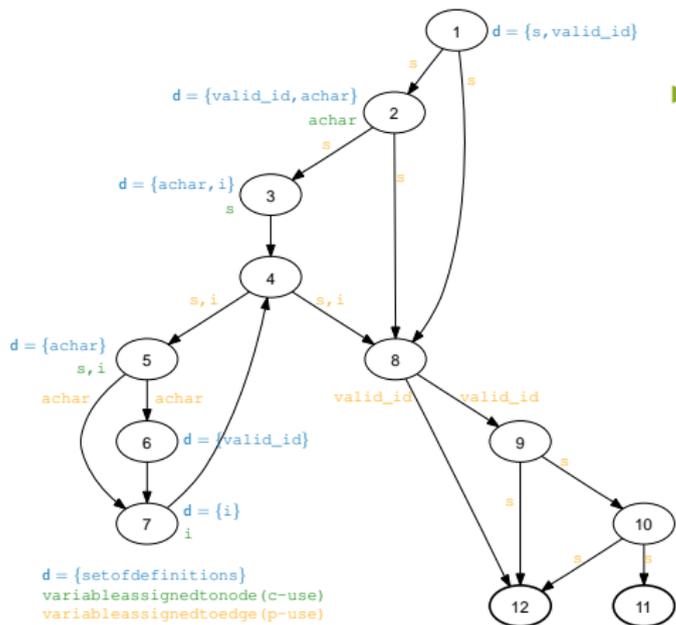
► Critérios de Fluxo de Dados:

► Todas-Defs

- Por exemplo, para exercitar a definição da variável `valid_id` definida no nó 1, de acordo com o critério Todas-Defs, poder-se-ia executar um dos seguintes sub-caminhos: (1,8,9) e (1,8,12);
- Observa-se que o sub-caminho (1,8,9) é não executável, e qualquer caminho completo que o inclua também é não executável.



Requisitos de Teste de Fluxo de Dados (3)



► Critérios de Fluxo de Dados:

- Em relação ao critério Todos-Usos, com respeito à mesma definição, seriam requeridas as seguinte associações:
 - $\langle \text{valid_id}, 1, (8, 9) \rangle$
 - $\langle \text{valid_id}, 1, (8, 12) \rangle$
- Observe que destas, apenas a associação $\langle \text{valid_id}, 1, (8, 12) \rangle$ é executável, a outra é não executável pois o único c.l.d. possível de exercitá-la seria um caminho que incluísse o sub-caminho (1,8,9);



Critérios Potenciais-Usos (Maldonado, 1991)

- ▶ Baseados no conceito de **potencial-associação**

Associações são estabelecidas sem a necessidade de um uso explícito

- ▶ Necessitam apenas do Grafo Def para derivar os requisitos de testes: GFC + Definições.

Requisitos de Teste de Fluxo de Dados (1)

- ▶ Requisitos gerados pelos critérios potenciais-usos:
- ▶ **Potenciais Associações Definição-Uso**
 - ▶ Em geral uma tripla: $\langle var, def, uso \rangle$, sendo:
 - ▶ *var* - variável para a qual a associação definição-uso foi estabelecida.
 - ▶ *def* - nó contendo a definição da variável *var*.
 - ▶ *uso* - nó/arco com potencial uso computacional/predicativo de *var*.



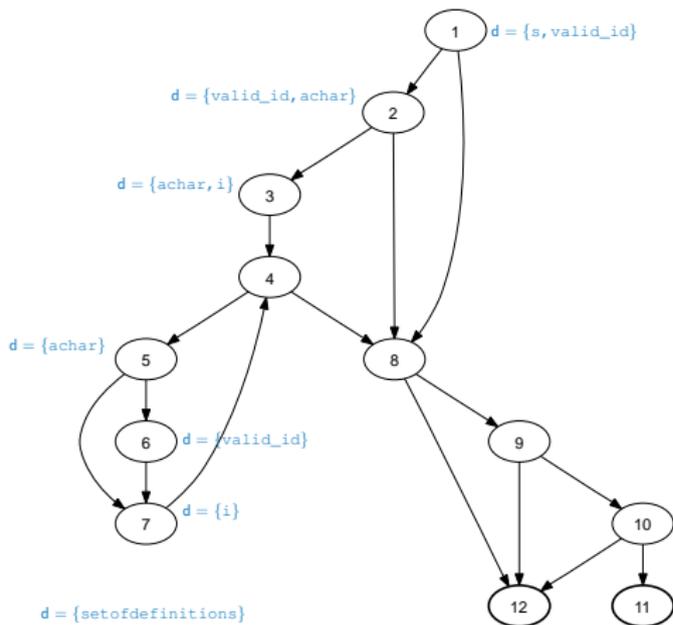
Requisitos de Teste de Fluxo de Dados (2)

Critérios Potenciais-Usos:

► Todas-Potenciais-Usos

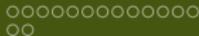
- $\langle \text{valid_id}, 1, (8, 9) \rangle$
- $\langle \text{valid_id}, 1, (8, 12) \rangle$
- $\langle \text{valid_id}, 1, (9, 10) \rangle$
- $\langle \text{valid_id}, 1, (9, 12) \rangle$
- $\langle \text{valid_id}, 1, (10, 11) \rangle$
- $\langle \text{valid_id}, 1, (10, 12) \rangle$

⋮



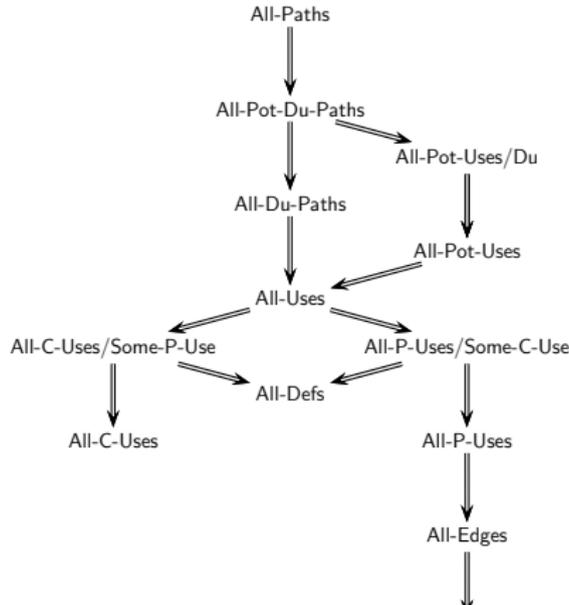
Outros Exemplos

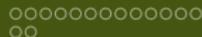
- ▶ Outros exemplos dos critérios de Fluxo de Dados podem ser encontrados no Capítulo 11 do livro de ([Copeland, 2004](#)).



Relação de Inclusão (1)

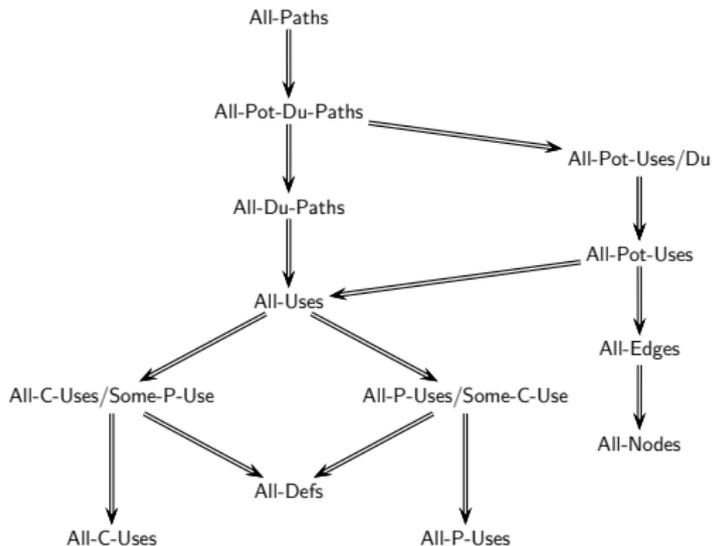
Sem considerar a presença de caminhos não-executáveis





Relação de Inclusão (2)

Considerando a presença de caminhos não-executáveis



Aplicabilidade e Limitações

- ▶ Complementares aos critérios de fluxo de controle.
- ▶ Podem ser aplicados em todas as fases de testes, sendo mais comum no teste de unidade e de integração.
- ▶ Também requerem conhecimento do programa para serem aplicados.
- ▶ Exigem a análise de associações definição-uso quanto à sua executabilidade.



Fluxo de Dados
Técnica de Aplicação
Definições

Exemplo de Aplicação

Critérios de Fluxo de Dados
Critérios de Rapps e Weyuker
Critérios Potenciais-Usos
Relação de Inclusão
Aplicabilidade e Limitações

Resumo

Exercício

Leitura Recomendada

Resumo

- ▶ Critérios de Fluxo de Dados identificam pares “definição-uso” entre variável em um programa.
- ▶ Grafo Definição-Usos é a base a partir do qual os requisitos de testes são derivados.
- ▶ Requisitos definidos como associações definição-uso.
- ▶ Tais critérios preenchem a lacuna entre os critérios de fluxo de controle todos-arcos e todos-caminhos.



Fluxo de Dados
Técnica de Aplicação
Definições

Exemplo de Aplicação

Critérios de Fluxo de Dados
Critérios de Rapps e Weyuker
Critérios Potenciais-Usos
Relação de Inclusão
Aplicabilidade e Limitações

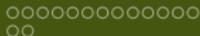
Resumo

Exercício

Leitura Recomendada

Exercício

- ▶ Incluir as anotações de fluxo de dados (definições e usos de variáveis), considerando o programa e o GFC a seguir.
- ▶ Considerando o critério Todos-Usos, derive 10 associações definição-uso válidas.
- ▶ Considerando o critério Todos-Potenciais-Usos, derive 10 potenciais-associações que não seriam geradas pelo critério Todos-Usos.



Fluxo de Dados
Técnica de Aplicação
Definições

Exemplo de Aplicação

Critérios de Fluxo de Dados
Critérios de Rapps e Weyuker
Critérios Potenciais-Usos
Relação de Inclusão
Aplicabilidade e Limitações

Resumo

Exercício

Leitura Recomendada

Leitura Recomendada

Mais informações sobre esse tema podem ser encontrados em:

- ▶ Nota didática de Introdução ao Teste de Software de [Maldonado et al. \(2004\)](#). Seção 2 do livro de [Copeland \(2004\)](#) e o livro de [Roper \(1994\)](#).

