

# SCE221 – Verificação, Validação e Teste SCE702 – Teste e Inspeção de Software

## Técnica de Teste Estrutural: Critérios Baseados em Fluxo de Dados

Profa. Ellen Francine Barbosa  
Profa. Simone do Rocio Senger de Souza  
{francine, srocio}@icmc.usp.br

Instituto de Ciências Matemáticas e de Computação — ICMC/USP

SCE221 – Verificação,  
Validação e Teste  
SCE702 – Teste e  
Inspeção de Software

Aula Anterior

Critérios Baseados em  
Fluxo de Dados

Critérios de Rapps e  
Weyuker

Critérios Potenciais-Usos

Ferramenta PokeTool

Resumo

Exercício

- Aula Anterior
- Critérios Baseados em Fluxo de Dados
- Critérios de Rapps e Weyuker
- Critérios Potenciais-Usos
- Ferramenta PokeTool
- Resumo
- Exercício

- Conhecida como teste caixa-branca (em oposição ao teste caixa-preta).
- 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 programa**.

- **Baseados em Fluxo de Controle**
  - Todos-Nós
  - Todas-Arestas
  - Todos-Caminhos: Simples, Completo, Livre de Laço, ...
- **Baseados em Complexidade**
  - Critério de McCabe (teste do caminho base).
- **Baseados em Fluxo de Dados**
  - Critérios de Rapps & Weyuker
    - Todas-Defs, Todos-Usos, Todos-P-Usos e outros.
  - Critérios Potenciais-Usos (Maldonado)
    - Todos-Potenciais-Usos, Todos-Potenciais-Usos/DU e outros.

- Critérios pertencentes à técnica estrutural.
- 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.

- Exemplo de **erro** de fluxo de dados:

```

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

- 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  }
```

- O teste baseado em fluxo de dados constitui uma ferramenta poderosa para identificar o uso incorreto de valores resultante de erros de codificação.
- 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.”

- Critérios principais:
  - Família de critérios de Rapps e Weyuker
  - Família de critérios Potenciais-Usos (Maldonado)

- 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.

- 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:
    - $\text{if } (a \geq b)$ .
- 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  $b = 5$ .

- **Objetivo:** exercitar caminhos ligando definições globais a usos globais de variáveis do programa.
- **Critérios:**
  - Todas-Definições.
  - Todos-P-Usos.
  - Todos-P-Usos/Alguns-C-Usos.
  - Todos-C-Usos/Alguns-P-Usos.
  - Todos-Usos.
  - Todos-DU-Caminhos.

- Utilizam o **Grafo Def-Uso** (*Def-Use Graph*) para derivar os requisitos de teste.
  - Informações a respeito do fluxo de dados do programa.
  - Extensão do GFC.

## Grafo Def-Uso

### GFC + Definição e Uso de Variáveis

- **Definição:**
  - Atribuição de um valor a uma variável.
  - $a = 1$
- **Uso Predicativo (p-uso):**
  - A variável é utilizada em uma condição.
  - if ( $a > 0$ )
- **Uso Computacional (c-uso):**
  - A variável é utilizada em uma computação.
  - $b = a + 1$



# Programa Identifier

SCE221 – Verificação,  
Validação e Teste  
SCE702 – Teste e  
Inspeção de Software

Aula Anterior

Critérios Baseados em  
Fluxo de Dados

Critérios de Rapps e  
Weyuker

Grafo Def-Usos

Critério Todas-Definições

Critério Todos-Usos

Critério Todos-DU-Caminhos

Limitações

Critérios Potenciais-Usos

Ferramenta PokeTool

Resumo

Exercício

O programa *Identifier* determina se um identificador é válido ou não. Um identificador válido deve começar com uma letra e conter apenas letras ou dígitos. Além disso, deve ter no mínimo um caractere e no máximo seis caracteres de comprimento.

```

/* 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 */      }

```

Implementação do Programa *Identifier* (função main).

- Função `valid_s()`: determina se o primeiro caractere é válido.
- Função `valid_f()`: determina se o próximo caractere é válido.

# Exemplo de Grafo Def-Uso: Programa Identifier

SCE221 – Verificação,  
Validação e Teste  
SCE702 – Teste e  
Inspeção de Software

Aula Anterior

Critérios Baseados em  
Fluxo de Dados

Critérios de Rapps e  
Weyuker

Grafo Def-Uso

Critério Todas-Definições

Critério Todos-Usos

Critério Todos-DU-Caminhos

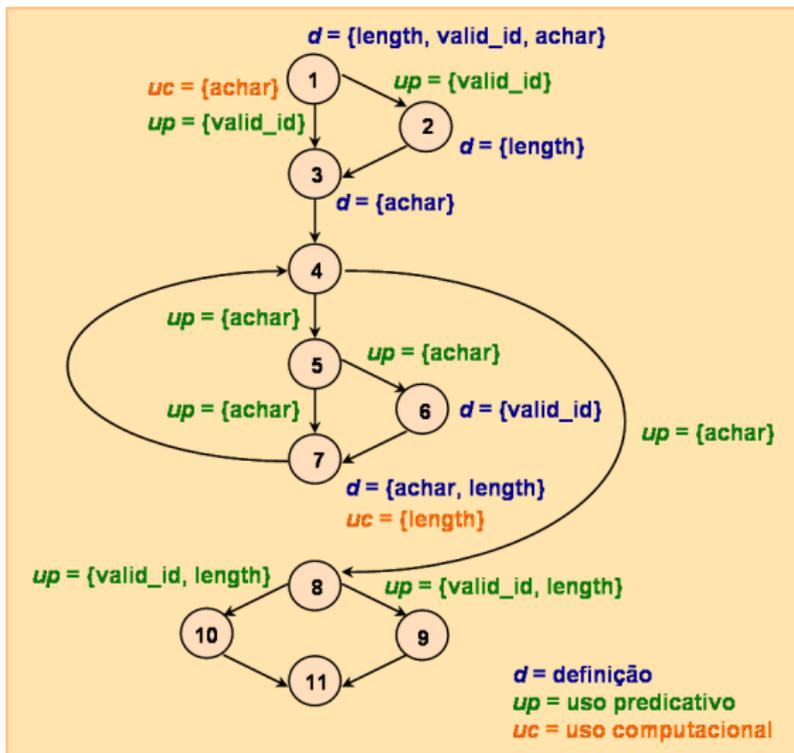
Limitações

Critérios Potenciais-Usos

Ferramenta PokeTool

Resumo

Exercício



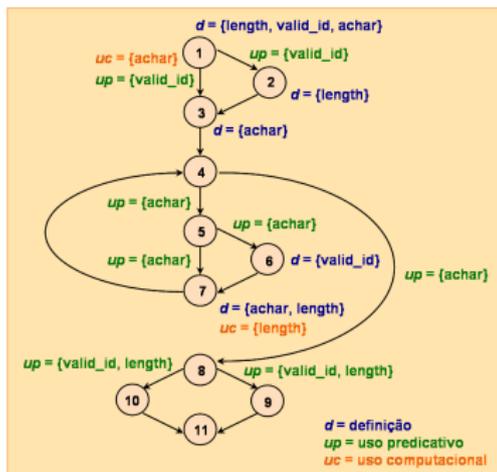
Requer que cada definição de variável seja exercitada pelo menos uma vez, não importa se por um c-uso ou por um p-uso.

- Para **todas as definições** de variáveis deve ser exercitado um **caminho livre de definição** para **pelo menos um de seus usos**.
  - Caminho onde a variável não é redefinida.

- Associações Definição-Uso

Tripla  $\langle i, j, var \rangle$  ou  $\langle i, (j, k), var \rangle$  indicando que a variável  $var$  é definida no nó  $i$  e existe um uso computacional de  $var$  no nó  $j$  ou um uso predicativo de  $var$  no arco  $(j, k)$ , respectivamente, bem como pelo menos um caminho livre de definição do nó  $i$  ao nó  $j$  ou ao arco  $(j, k)$ .

- Elementos Requeridos  
(definição da variável *length* no nó 1):
  - Associações:  $\langle 1,7,length \rangle$ ;  $\langle 1,(8,9),length \rangle$  ou  $\langle 1,(8,10),length \rangle$ .
  - Basta ser executado um dos seguintes sub-caminhos:  
(1,3,4,5,7); (1,3,4,5,6,7); (1,3,4,8,9); (1,3,4,8,10).



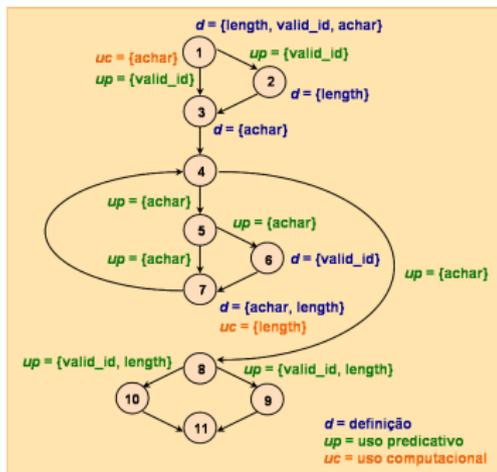
- Para satisfazer o critério Todas-Definições, a análise anterior deve ser realizada para **toda definição** que ocorre no programa.

Nó	Variável	Caminhos Requeridos
1	length	(1,3,4,5,7)
		(1,3,4,5,6,7)
		(1,3,4,8,9) ×
		(1,3,4,8,10)
	valid_id	(1,2,3,4,8,9)
		(1,3,4,8,10)
	achar	(1,3,4,5,7)
(1,3,4,5,6)		
(1,3,4,8)		
2	length	(2,3,4,5,7)
		(2,3,4,5,6,7)
		(2,3,4,8,9)
		(2,3,4,8,10) ×
3	achar	(3,4,5,7)
		(3,4,5,6)
		(3,4,8)
6	valid_id	(6,7,4,8,9) ×
		(6,7,4,8,10)
7	length	(7,4,5,7)
		(7,4,5,6,7)
		(7,4,8,9)
		(7,4,8,10)
	achar	(7,4,5,7)
		(7,4,5,6)
		(7,4,8)

Requer que todas as associações entre uma definição de variável e seus subseqüentes usos (c-usos e p-usos) sejam exercitadas pelos casos de teste, através de pelo menos um caminho livre de definição.

- Para **todas as definições** de variáveis deve ser exercitado um caminho para **todos os seus c-usos** e para **todos os seus p-usos**.

- Elementos Requeridos (definição da variável length no nó 1):
  - Associações:  $\langle 1,7,length \rangle$ ;  $\langle 1,(8,9),length \rangle$  e  $\langle 1,(8,10),length \rangle$ .



- Para satisfazer o critério Todos-Usos, a análise anterior deve ser realizada para **todas** as demais variáveis e associações pertinentes.

Associações Requeridas
$\langle 1, 7, \{length\} \rangle$
$\langle 1, (8, 9), \{length, valid\_id\} \rangle \times$
$\langle 1, (8, 10), \{length, valid\_id\} \rangle$
$\langle 1, (1, 2), \{valid\_id\} \rangle$
$\langle 1, (1, 3), \{valid\_id\} \rangle$
$\langle 1, 1, \{achar\} \rangle$
$\langle 1, (4, 5), \{achar\} \rangle$
$\langle 1, (4, 8), \{achar\} \rangle$
$\langle 1, (5, 6), \{achar\} \rangle$
$\langle 1, (5, 7), \{achar\} \rangle$
$\langle 2, 7, \{length\} \rangle$
$\langle 2, (8, 9), \{length\} \rangle$
$\langle 2, (8, 10), \{length\} \rangle \times$
$\langle 3, (4, 5), \{achar\} \rangle$
$\langle 3, (4, 8), \{achar\} \rangle$
$\langle 3, (5, 6), \{achar\} \rangle$
$\langle 3, (5, 7), \{achar\} \rangle$
$\langle 6, (8, 9), \{valid\_id\} \rangle \times$
$\langle 6, (8, 10), \{valid\_id\} \rangle$
$\langle 7, 7, \{length\} \rangle$
$\langle 7, (8, 9), \{length\} \rangle$
$\langle 7, (8, 10), \{length\} \rangle$
$\langle 7, (4, 5), \{achar\} \rangle$
$\langle 7, (4, 8), \{achar\} \rangle$
$\langle 7, (5, 6), \{achar\} \rangle$
$\langle 7, (5, 7), \{achar\} \rangle$

Aula Anterior

Critérios Baseados em  
Fluxo de Dados

Critérios de Rapps e  
Weyuker

Grafo Def-Use

Critério Todas-Definições

Critério Todos-Usos

Critério Todos-DU-Caminhos

Limitações

Critérios Potenciais-Usos

Ferramenta PokeTool

Resumo

Exercício

- Os critérios Todos-P-Usos, Todos-P-Usos/Alguns-C-Usos e Todos-C-Usos/Alguns-P-Usos representam variações do critério Todos-Usos.

<b>Critério</b>	<b>Descrição</b>
Todos-P-Usos	Para todas as definições de variáveis deve ser exercitado um caminho para todos os seus p-usos.
Todos-P-Usos/ Alguns-C-Usos	Para todas as definições de variáveis deve ser exercitado um caminho 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 para todos os seus c-usos e para alguns p-usos.

SCE221 – Verificação,  
Validação e Teste  
SCE702 – Teste e  
Inspeção de Software

Aula Anterior

Critérios Baseados em  
Fluxo de Dados

Critérios de Rapps e  
Weyuker

Grafo Def-Usos

Critério Todas-Definições

Critério Todos-Usos

Critério Todos-DU-Caminhos

Limitações

Critérios Potenciais-Usos

Ferramenta PokeTool

Resumo

Exercício

Requer que toda associação entre uma definição de variável e subseqüentes p-usos ou c-usos dessa variável seja exercitada por todos os caminhos livres de definição e **livres de laço** que cubram essa associação.

- Propriedades Mínimas de um Critério de Teste:
  - 1 Garantir, do ponto de vista de fluxo de controle, a cobertura de todos os **desvios condicionais**.  
Ou seja, incluir o critério **Todos-Arcos**.
  - 2 Requerer, do ponto de vista de fluxo de dados, ao menos um **uso** de todo resultado computacional.  
Ou seja, incluir o critério **Todas-Definições**.
  - 3 Requerer um conjunto de casos de teste **finito**.

## ● Hierarquia entre os Critérios



- A principal desvantagem dos critérios baseados em análise de fluxo de dados é que na presença de caminhos não executáveis estes **não garantem a inclusão do critério Todos-Arcos**.
- Diz-se que tais critérios não estabelecem uma **ponte** (*bridge the gap*) entre os critérios Todos-Arcos e Todos-Caminhos.
- Ainda, a maioria dos programas reais contém caminhos não executáveis.

- Os elementos requeridos são caracterizados independentemente da ocorrência **explícita** de uma referência (um uso) a uma definição de variável.
  - **Potencial-Associação (Potencial-Uso)**

Se um uso dessa definição pode existir, a potencial-associação entre a definição e o potencial-uso é caracterizada, e eventualmente requerida.

Requerem basicamente que caminhos livres de definição em relação a qualquer nó  $i$  que possua definição de variável e a qualquer variável  $x$  definida em  $i$  sejam executados, **independentemente** de ocorrer uso dessa variável nesses caminhos.

- É possível verificar, por exemplo, que o valor de  $x$  não foi alterado nesses caminhos (possivelmente devido a efeitos colaterais), ganhando-se maior confiança de que a computação correta é realizada.

SCE221 – Verificação,  
Validação e Teste  
SCE702 – Teste e  
Inspeção de Software

Aula Anterior

Critérios Baseados em  
Fluxo de Dados

Critérios de Rapps e  
Weyuker

Critérios Potenciais-Usos

Grafo Def

Critério Todos-Potenciais-Usos

Ferramenta PokeTool

Resumo

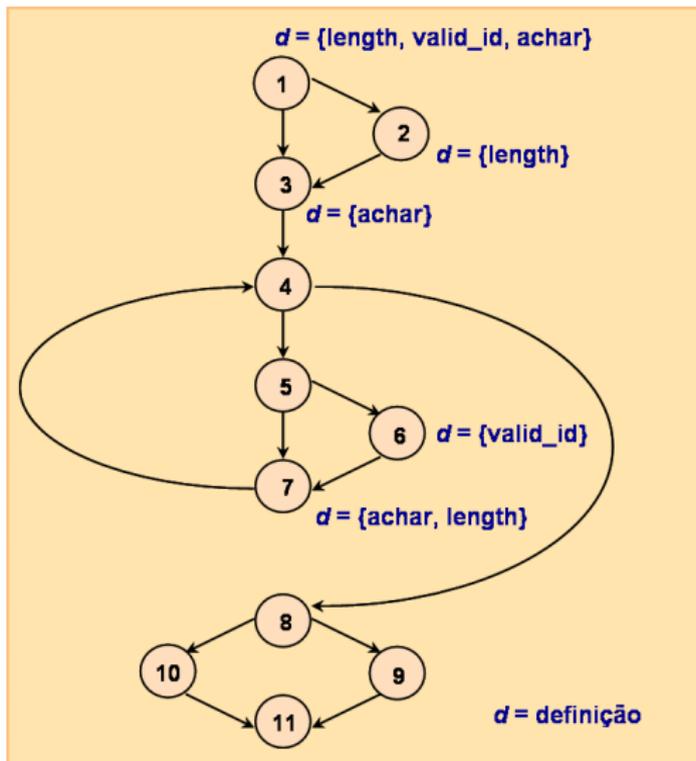
Exercício

- Critérios Básicos:
  - Todos-Potenciais-Usos
  - Todos-Potenciais-Usos/DU
  - Todos-Potenciais-DU-Caminhos

- Utilizam o **Grafo Def** para derivar os requisitos de teste.
- Associa-se, a cada nó do grafo, informações a respeito das **definições** que ocorrem nesses nós.
  - Extensão do grafo de programa.

## Grafo Def

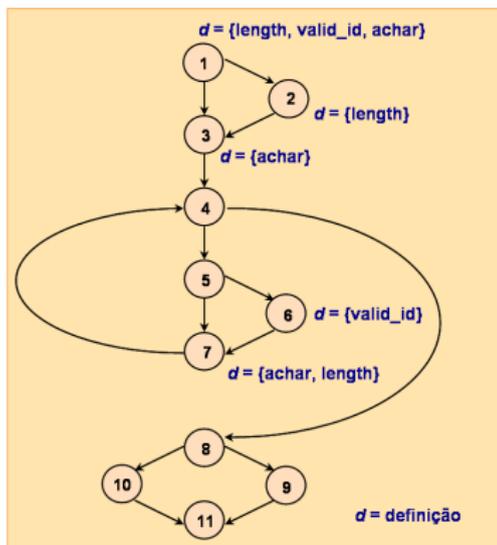
Grafo de Programa + Definição de Variáveis



Requer, para todo nó  $i$  e para toda variável  $x$ , para a qual existe uma definição em  $i$ , que pelo menos um caminho livre de definição com relação à variável  $x$  do nó  $i$  para todo nó e para todo arco possível de ser alcançado a partir de  $i$  por um caminho livre de definição com relação a  $x$  seja exercitado.

- Elementos Requeridos:

- Associações:  $\langle 1, (1,2), \text{length} \rangle$ ;  $\langle 1, (1,3), \text{length} \rangle$ ;  
 $\langle 1, (5,6), \text{length} \rangle$ ;  $\langle 1, (5,7), \text{length} \rangle$ ;  $\langle 1, (8,9), \text{length} \rangle$ ;  
 $\langle 1, (8,10), \text{length} \rangle$ ;  $\langle 1, (6,7), \text{length} \rangle$ .

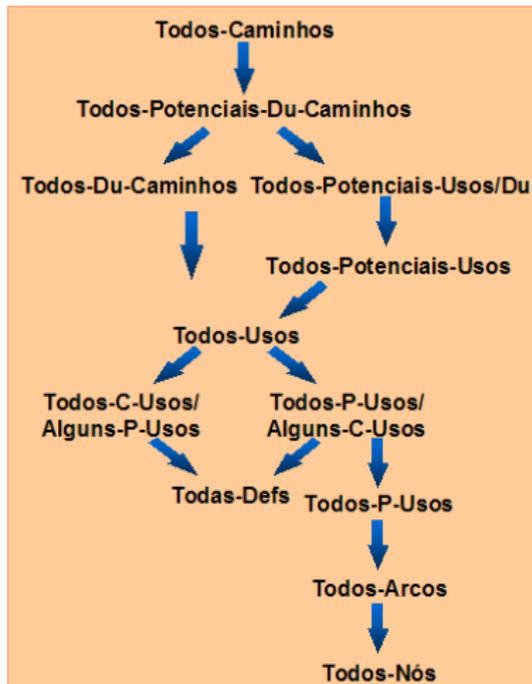


- Utilizando-se o conceito de potencial-uso, tem-se que a variável `length` definida no nó 1 poderia ter um uso predicativo no arco  $(6, 7)$ , o que faz com que a potencial-associação  $\langle 1, (6, 7), \text{length} \rangle$  seja requerida pelo critério Todos-Potenciais-Usos.
- A potencial-associação  $\langle 1, (6, 7), \text{length} \rangle$  não seria requerida pelos demais critérios de fluxo de dados que não fazem uso do conceito de potencial-uso.
- Toda associação é uma potencial-associação e, desse modo, as associações requeridas pelo critério Todos-Usos são um **subconjunto** das potenciais-associações requeridas pelo critério Todos-Potenciais-Usos.

## Associações Requeridas

1) $\langle 1, (6, 7), \{\text{length}\} \rangle$	17) $\langle 2, (6, 7), \{\text{length}\} \rangle$
2) $\langle 1, (1, 3), \{\text{achar, length, valid\_id}\} \rangle$	18) $\langle 2, (5, 6), \{\text{length}\} \rangle$
3) $\langle 1, (8, 10), \{\text{length, valid\_id}\} \rangle$	19) $\langle 3, (8, 10), \{\text{achar}\} \rangle$
4) $\langle 1, (8, 10), \{\text{valid\_id}\} \rangle$	20) $\langle 3, (8, 9), \{\text{achar}\} \rangle$
5) $\langle 1, (8, 9), \{\text{length, valid\_id}\} \times$	21) $\langle 3, (5, 7), \{\text{achar}\} \rangle$
6) $\langle 1, (8, 9), \{\text{valid\_id}\} \rangle$	22) $\langle 3, (6, 7), \{\text{achar}\} \rangle$
7) $\langle 1, (7, 4), \{\text{valid\_id}\} \rangle$	23) $\langle 3, (5, 6), \{\text{achar}\} \rangle$
8) $\langle 1, (5, 7), \{\text{length, valid\_id}\} \rangle$	24) $\langle 6, (8, 10), \{\text{valid\_id}\} \rangle$
9) $\langle 1, (5, 7), \{\text{valid\_id}\} \rangle$	25) $\langle 6, (8, 9), \{\text{valid\_id}\} \times$
10) $\langle 1, (5, 6), \{\text{length, valid\_id}\} \rangle$	26) $\langle 6, (5, 7), \{\text{valid\_id}\} \rangle$
11) $\langle 1, (5, 6), \{\text{valid\_id}\} \rangle$	27) $\langle 6, (5, 6), \{\text{valid\_id}\} \rangle$
12) $\langle 1, (2, 3), \{\text{achar, valid\_id}\} \rangle$	28) $\langle 7, (8, 10), \{\text{achar, length}\} \rangle$
13) $\langle 1, (1, 2), \{\text{achar, length, valid\_id}\} \rangle$	29) $\langle 7, (8, 9), \{\text{achar, length}\} \rangle$
14) $\langle 2, (8, 10), \{\text{length}\} \times$	30) $\langle 7, (5, 7), \{\text{achar, length}\} \rangle$
15) $\langle 2, (8, 9), \{\text{length}\} \rangle$	31) $\langle 7, (6, 7), \{\text{achar, length}\} \rangle$
16) $\langle 2, (5, 7), \{\text{length}\} \rangle$	32) $\langle 7, (5, 6), \{\text{achar, length}\} \rangle$

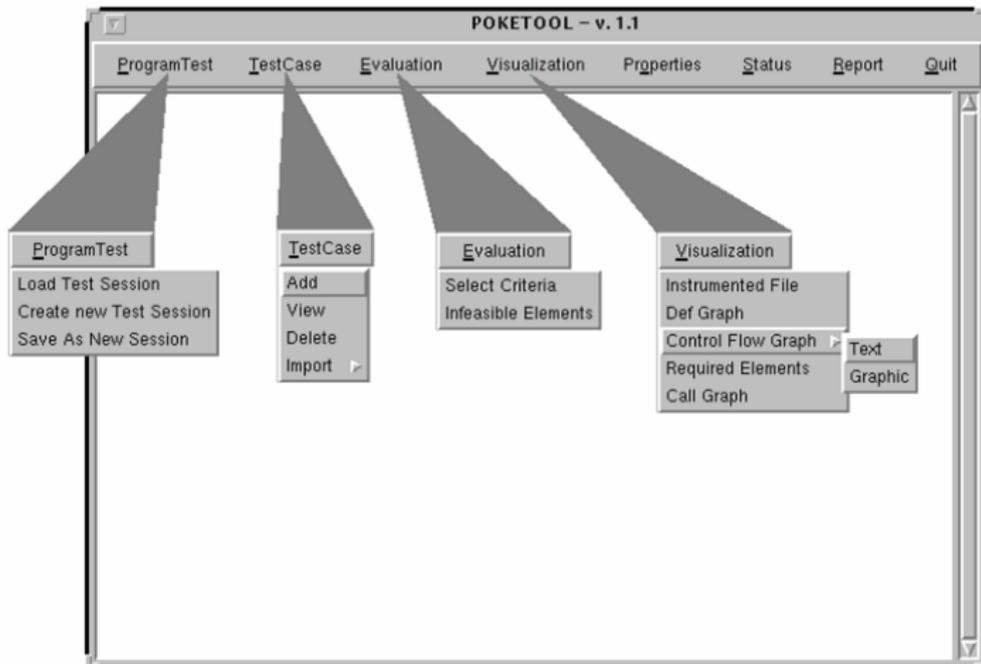
- Hierarquia entre os Critérios



- Os critérios Potenciais-Usos são os únicos critérios baseados em fluxo de dados que satisfazem as três propriedades mínimas esperadas de um critério de teste.
  - Estabelecem uma **hierarquia** entre os critérios Todos-Arcos e Todos-Caminhos, mesmo na presença de caminhos não executáveis.
- Nenhum outro critério de teste baseado em fluxo de dados inclui os critérios Potenciais-Usos.

- *Potential Uses Criteria Tool for Program Testing*
- Critérios Baseados em Fluxo de Controle.
- Critérios de Rapps e Weyuker.
- Critérios Potenciais-Usos.
- Linguagem C
- Características
  - Sessão de teste.
  - Importação de casos de teste.
  - Inserção e remoção de casos de teste dinamicamente.
  - Casos de teste podem ser habilitados ou desabilitados.
  - Geração de relatórios.

## ● Interface Gráfica



## ● Sessão de Teste

**Create New Test Session**

**Directory:** /export/home1/es/ellen/escola/poketool

**Test Session Name:** l

**Source Program:** i.c

**Included Files:**

**Used Defines:**

**Functions:** main

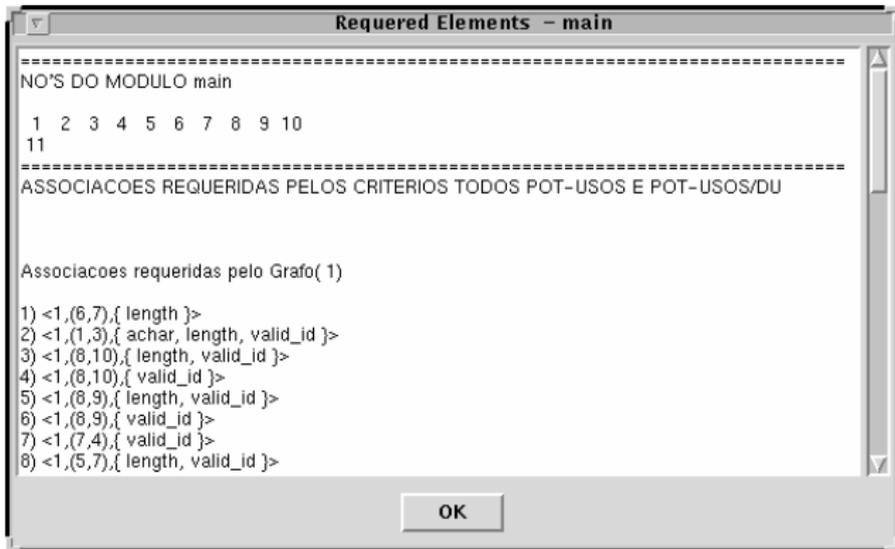
**Compilation Command:** gcc <source> -o <exec> -w

All Node  
 All Edges

**Criteria:**
 All Potential Uses  
 All Potential Uses/DU  
 All Potential DU-paths

**Type:**
 test  
 research

## ● Elementos Requeridos



## ● Relatórios de Teste

**Report**

Directory: /export/home1/es/ellen/escola/poketool

Report Name: report

Function: main

All Node  
 All Edges  
 All Potential Uses  
 All Potential Uses/DU  
 All Potential DU-paths

All  
 Executed  
 Not Executed  
 Infeasible

View Save Cancel

**Status**

Directory: /export/home1/es/ellen/escola/poketool

Test Session Name: |

Source File: | C:

Included Files: |

Used Defines: |

Compilation Command: gcc <source> -o <exec> -w

Function: main Type: test Total Test Case: 2

Criteria:

	Total	Exec	NExec	Infeas	ToL.Cover	Gr.Cover
<input checked="" type="checkbox"/> All Node	11	10	1	0	90.90	
<input checked="" type="checkbox"/> All Edges	6	5	1	0	83.33	
<input checked="" type="checkbox"/> All Potential Uses	32	13	19	0	40.62	32.30
<input checked="" type="checkbox"/> All Potential Uses/DU	32	11	21	0	34.37	29.23
<input checked="" type="checkbox"/> All Potential DU-paths	24	7	17	0	29.16	30.00

OK

- Relatórios de Teste

**Status**

Directory:

Test Session Name:

Source File:

Included Files:

Used Defines:

Compilation Command:

Function:  Type:  Total Test Case:

Critéria:

	Total	Exec	NExec	Infeas	Tot.Cover	Gr.Cover
<input checked="" type="checkbox"/> All Node	11	11	0	0	100.	
<input checked="" type="checkbox"/> All Edges	6	6	0	0	100.	
<input checked="" type="checkbox"/> All Potential Uses	32	29	3	0	90.62	89.46
<input checked="" type="checkbox"/> All Potential Uses/DU	32	29	3	0	90.62	89.46
<input checked="" type="checkbox"/> All Potential DU-paths	24	20	4	0	83.33	85.00

## ● Relatórios de Teste

Aula Anterior

Critérios Baseados em  
Fluxo de Dados

Critérios de Rapps e  
Weyuker

Critérios Potenciais-Usos

Ferramenta PokeTool

Resumo

Exercício

```

Report - main
ASSOCIACOES DO CRITERIO TODOS POT-USOS executadas:
<1,(1,3),( achar, length, valid_id )>
<1,(8,10),( valid_id )>
<1,(8,9),( valid_id )>
<1,(7,4),( valid_id )>
<1,(5,7),( length, valid_id )>
<1,(5,7),( valid_id )>
<1,(5,6),( valid_id )>
<1,(2,3),( achar, valid_id )>
<1,(1,2),( achar, length, valid_id )>
<2,(5,7),( length )>
<2,(6,7),( length )>
<2,(5,6),( length )>
<3,(5,7),( achar )>
<3,(6,7),( achar )>
<3,(5,6),( achar )>
<6,(8,10),( valid_id )>
<6,(5,7),( valid_id )>
<7,(8,10),( achar, length )>
<7,(8,9),( achar, length )>
<7,(5,7),( achar, length )>

Cobertura Total = 62.500000
Media da Cobertura dos Grafo() = 59.846153
    
```

```

Report - main
ASSOCIACOES DO CRITERIO TODOS POT-USOS nao executadas:
<1,(8,7),( length )>
<1,(8,10),( length, valid_id )>
<1,(8,5),( length, valid_id )>
<1,(5,5),( length, valid_id )>
<2,(8,10),( length )>
<2,(8,9),( length )>
<3,(8,10),( achar )>
<3,(8,5),( achar )>
<6,(8,5),( valid_id )>
<6,(5,5),( valid_id )>
<7,(6,7),( achar, length )>
<7,(5,5),( achar, length )>

Cobertura Total = 62.500000
Media da Cobertura dos Grafo() = 59.846153
    
```

- Critérios Baseados em Fluxo de Dados identificam pares **definição-uso** para variáveis de um programa.
- **Critérios de Rapps e Weyuker**
  - Grafo Def-Usos é a base a partir do qual os requisitos de testes são derivados.
  - Requisitos definidos como associações definição-uso.
- **Critérios Potenciais-Usos**
  - Grafo Def é a base a partir do qual os requisitos de testes são derivados.
  - Requisitos definidos como potenciais-associações.
  - Preenchem a lacuna entre os critérios de fluxo de controle Todos-Arcos e Todos-Caminhos.

SCE221 – Verificação,  
Validação e Teste  
SCE702 – Teste e  
Inspeção de Software

Aula Anterior

Critérios Baseados em  
Fluxo de Dados

Critérios de Rapps e  
Weyuker

Critérios Potenciais-Usos

Ferramenta PokeTool

Resumo

Exercício



- Considerando o programa Sort (Bolha) a seguir...
  - Gerar o **Grafo Def-Uso**.
  - Derivar as **associações definição-uso** válidas para o critério Todos-Usos.
  - Derivar as **potenciais-associações** que não seriam geradas pelo critério Todos-Usos para o critério Todos-Potenciais-Usos.
  - Derivar os **casos de teste** para os critérios Todos-Usos e Todos-Potenciais-Usos.

```
2 public void bolha(int [] a, int size) {  
3     int i, j, aux;  
4     for (i = 0; i < size; i++) {  
5         for (j = size - 1; j > i; j--) {  
6             if (a[j - 1] > a[j]) {  
7                 aux = a[j - 1];  
8                 a[j - 1] = a[j];  
9                 a[j] = aux;  
10            }  
11        }  
12    }  
13 }
```

