SSC721 – Teste e Inspeção de Software

Técnica de Teste Estrutural: Critérios Baseados em Fluxo de Dados

Profa. Ellen Francine Barbosa
francine@icmc.usp.br

Instituto de Ciências Matemáticas e de Computação — ICMC/USP

Roteiro

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

Técnica Estrutural

- 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.

Critérios da Técnica Estrutural

- 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
    - Todos-Potenciais-Usos, Todos-Potenciais-Usos/DU e outros.

Critérios Baseados em Fluxo de Dados

- 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.

Critérios Baseados em Fluxo de Dados

- O teste baseado em fluxo de dados constitui uma ferramenta poderosa para identificar o uso incorreto de valores resultantes 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."

Definição e Uso de Variáveis

- 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`.
- 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)`.

Critérios Baseados em Fluxo de Dados

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

Critérios de Rapps e Weyuker

- 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.

Grafo Def-Uso

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

Grafo Def-Uso

GFC + Definição e Uso de Variáveis

Grafo Def-Uso

- 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

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.

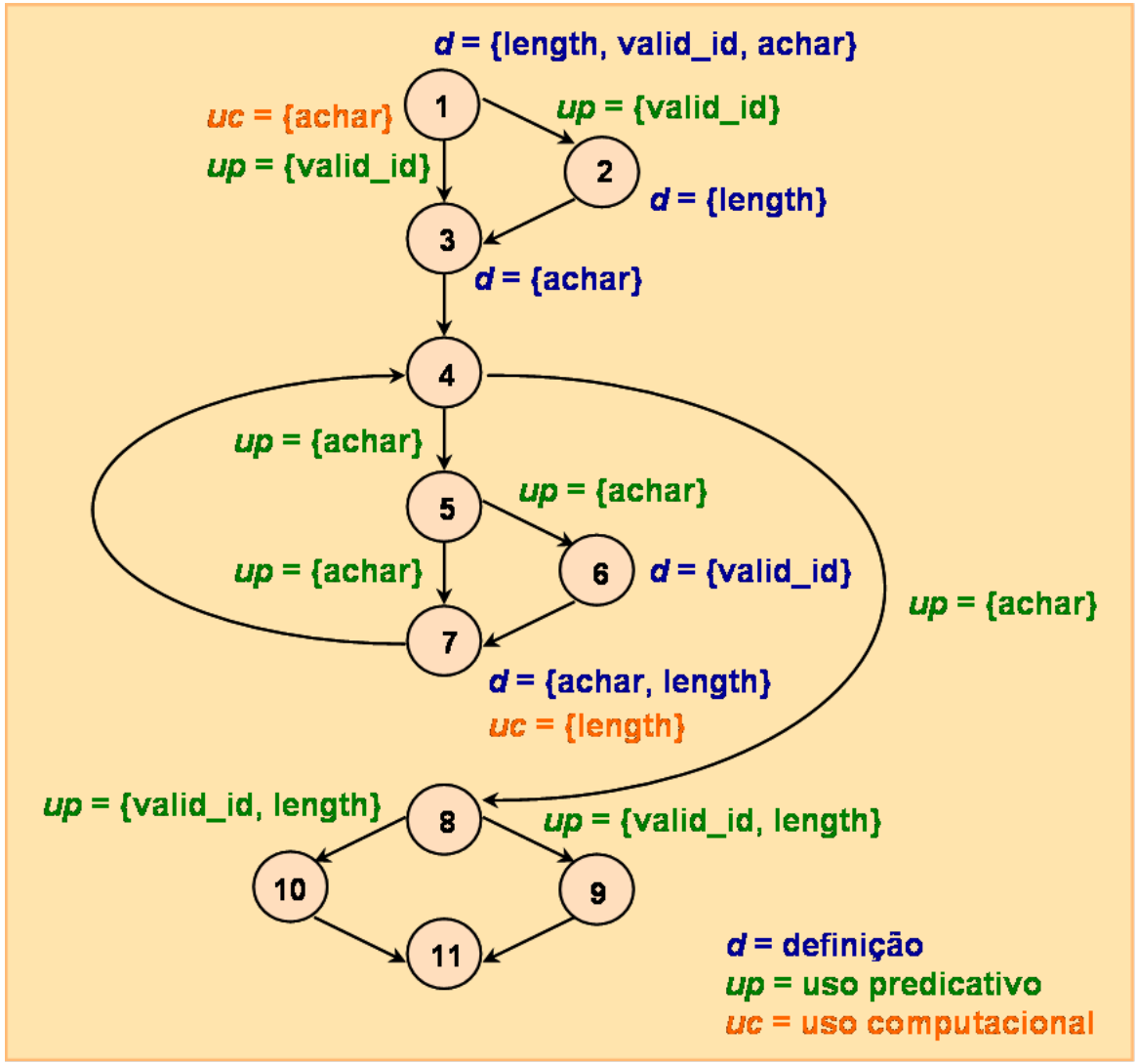
Programa Identifier

```
/* 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



Associações Definição-Uso

Tripla $<i,j,var>$ ou $<i,(j,k),var>$ 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).

Caminho livre de definição

- Caminho onde a variável não é redefinida.

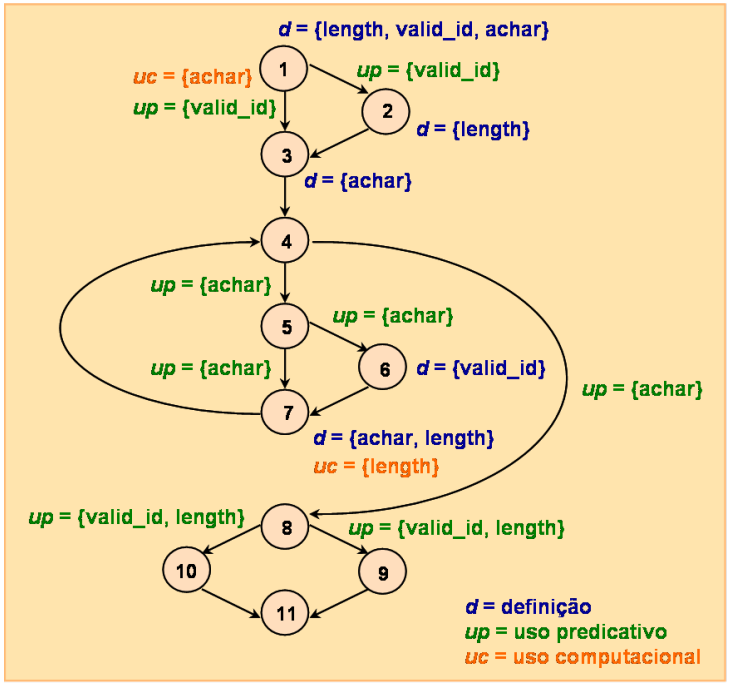
Critério Todas-Definições

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.

Critério Todas-Definições

- Elementos Requeridos (definição da variável `length` no nó 1):
  - Associações: <1,7,length>; <1,(8,9),length> ou <1,(8,10),length>.
  - 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).



Exemplo de Grafo Def-Uso: Programa Identifier

- Caminho Não-Executável (Executabilidade): não existe um dado de entrada que leve à execução desse caminho.

```
/* 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 */   }
```

Programa *Identifier* (função main).

Exemplo de Grafo Def-Uso: Programa Identifier

- Caminho Não-Executável (Executabilidade): não existe um dado de entrada que leve à execução desse caminho.

```
/* 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 */   }
```

Programa *Identifier* (função main).

Critério Todas-Definições

- 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)

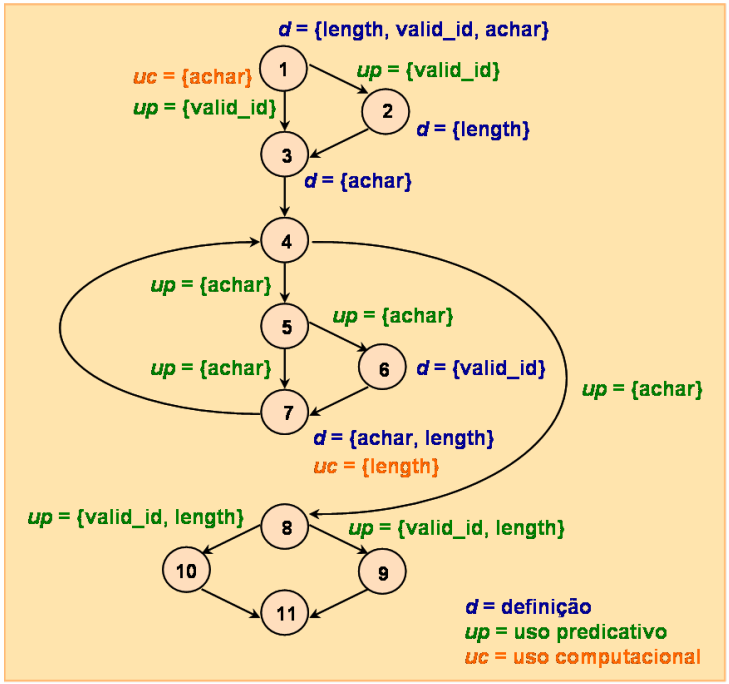
Critério Todos-Usos

Requer que todas as associações entre uma definição de variável e seus subsequentes 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.

Critério Todos-Usos

- Elementos Requeridos (definição da variável `length` no nó 1):
  - Associações: <1,7,length>; <1,(8,9),length> e <1,(8,10),length>.



Critério Todos-Usos

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

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

Critério Todos-Usos

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

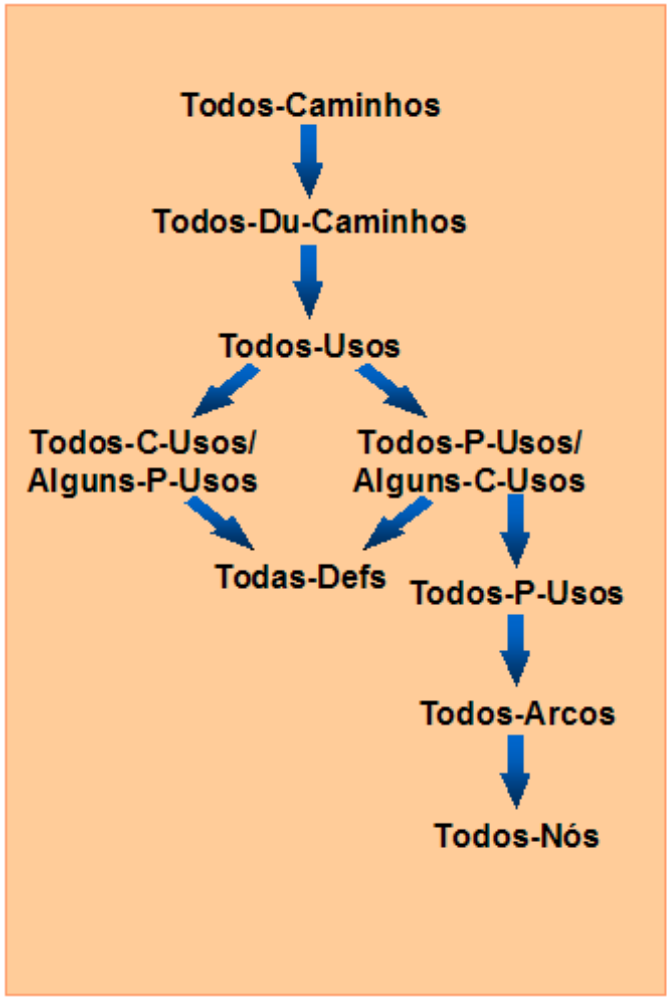
Critério	**Descrição**
Todos-P-Usos	Para todas as definições de variáveis deve ser exercitado um caminho para todos os seus p-usos.
Todos-P-Usos/ Alguns-C-Usos	Para todas as definições de variáveis deve ser exercitado um caminho para todos os seus p-usos e alguns c-usos.
Todos-C-Usos/ Alguns-P-Usos	Para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

Limitações

- 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.

Critérios de Rapps e Weyuker

- Hierarquia entre os Critérios



Limitações

- 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.
  - A maioria dos programas reais contém caminhos não executáveis.
- Diz-que que tais critérios não estabelecem uma ponte (*bridge the gap*) entre os critérios Todos-Arcos e Todos-Caminhos.

Critérios Potenciais-Usos

- 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.

Critérios Potenciais-Usos

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.

Critérios Potenciais-Usos

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

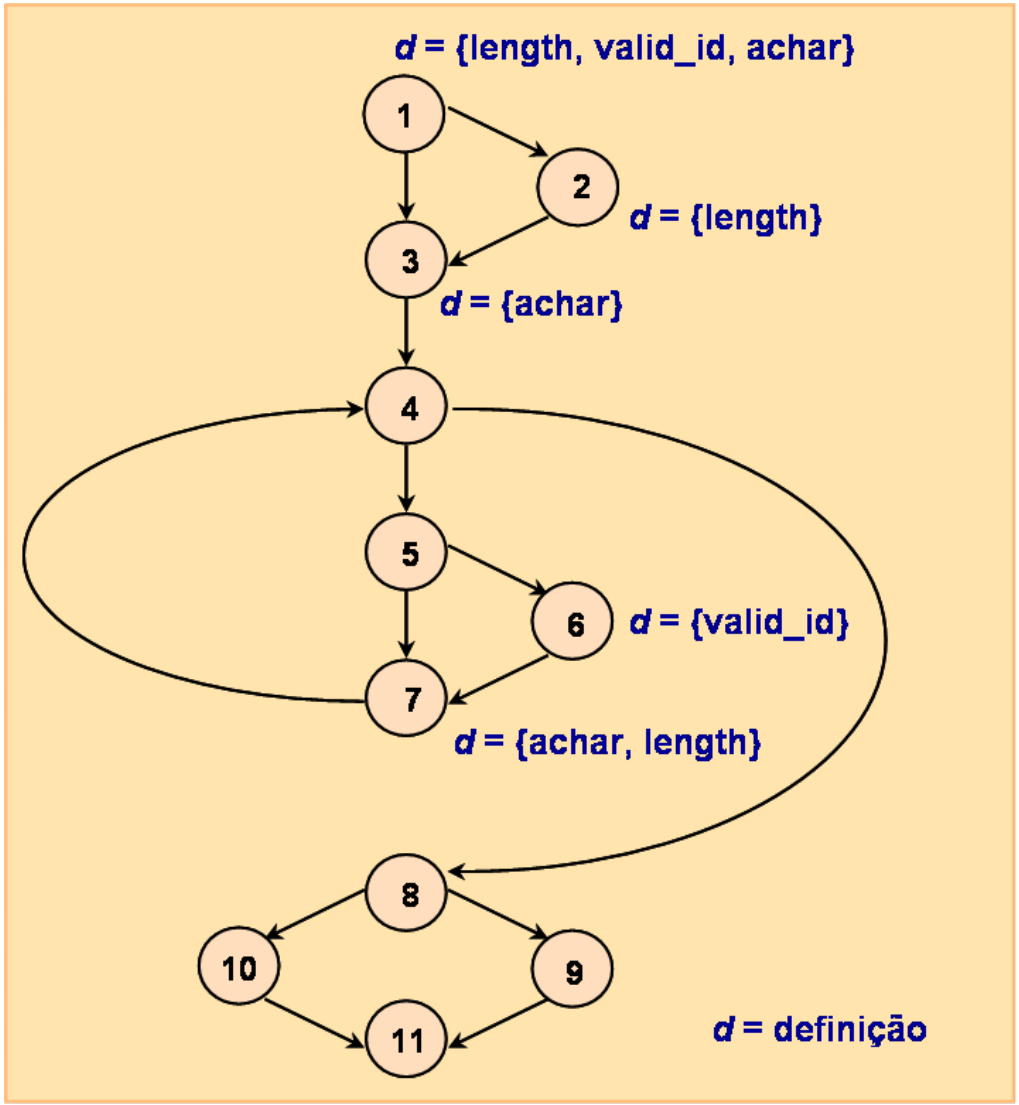
Grafo Def

- 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

Exemplo de Grafo Def: Programa Identifier

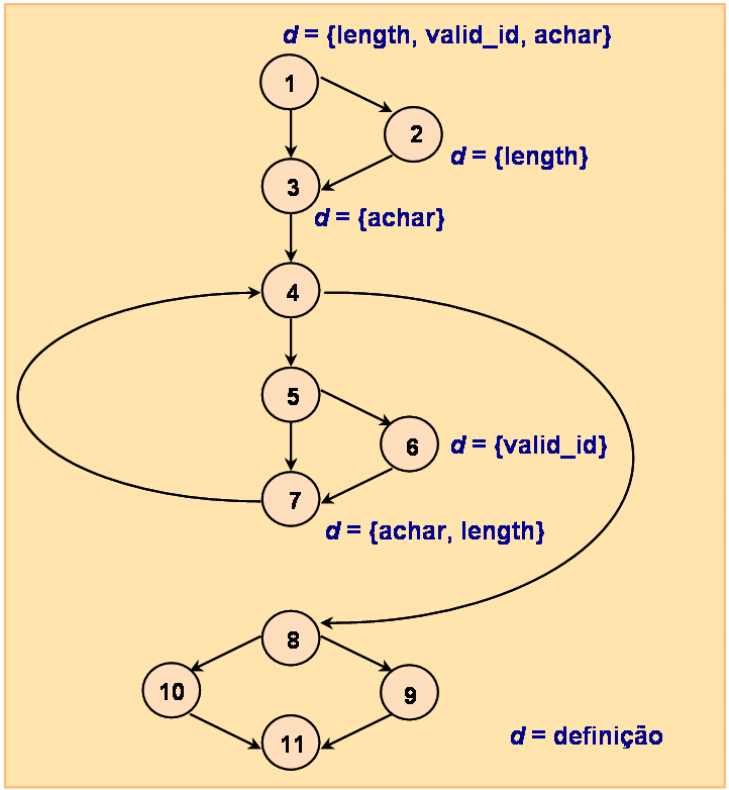


Critério Todos-Potenciais-Usos

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.

Critério Todos-Potenciais-Usos

- Elementos Requeridos:
  - Associações: <1,(1,2),length>; <1,(1,3),length>; <1,(5,6),length>; <1,(5,7),length>; <1,(8,9),length>; <1,(8,10),length>.



Critério Todos-Potenciais-Usos

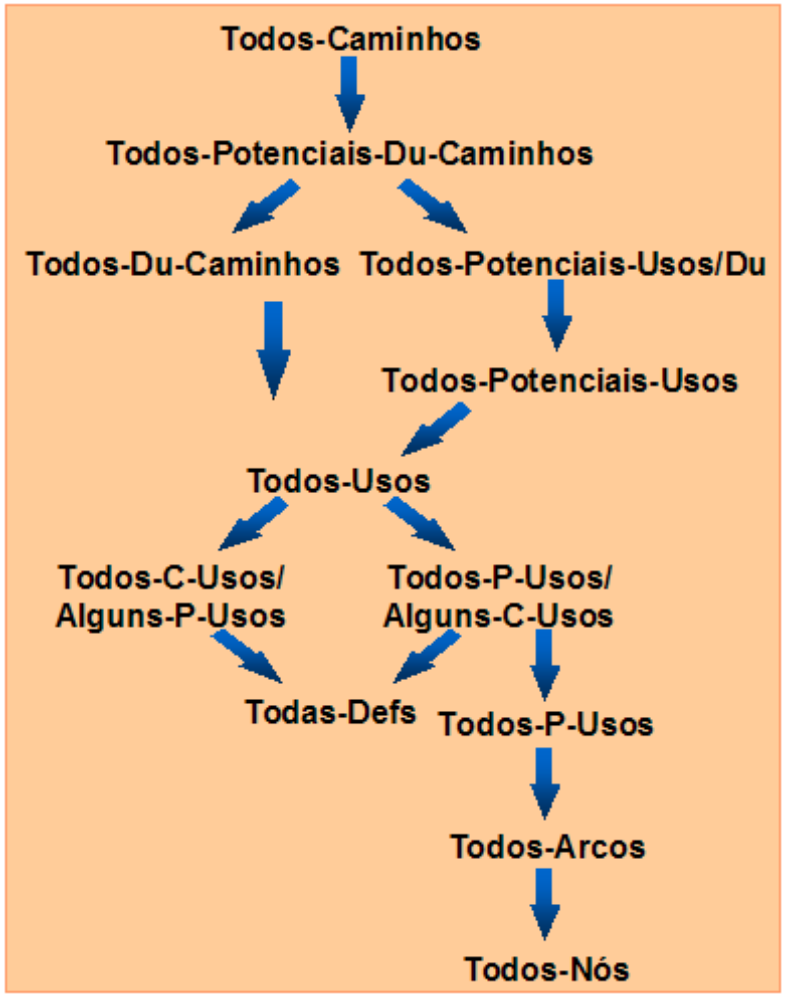
- Utilizando-se o conceito de potencial-uso, tem-se que a variável `length` definida no nó 1 poderia ter um uso predicativo no arco $(5, 7)$, o que faz com que a potencial-associação <1,(5,7),length> seja requerida pelo critério Todos-Potenciais-Usos.
- A potencial-associação <1,(5,7),length> não seria 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.

Critério Todos-Potenciais-Usos

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

Critério Todos-Potenciais-Usos

- Hierarquia entre os Critérios



Critério Todos-Potenciais-Usos

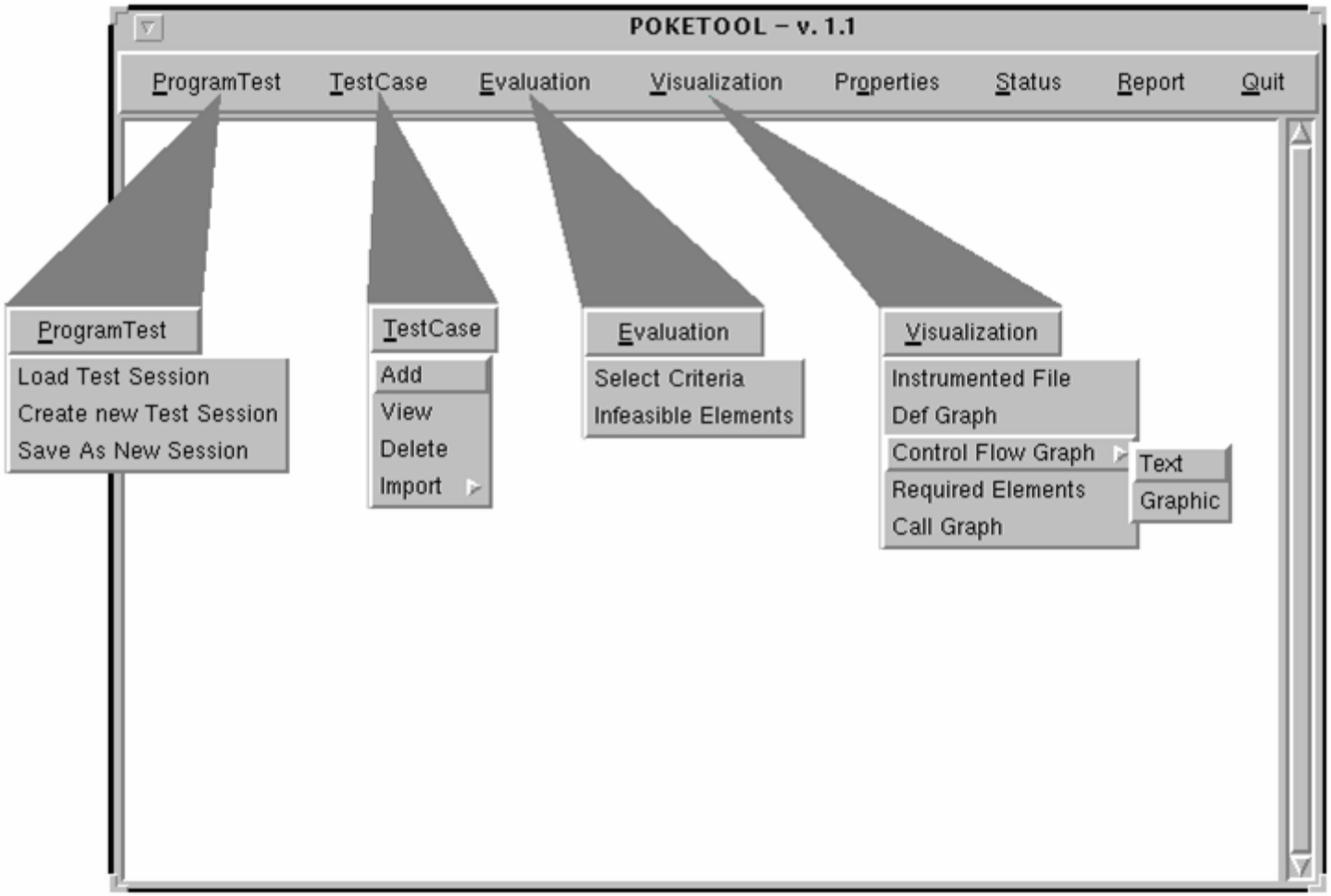
- Os critérios Potenciais-Usos são os únicos critérios baseados em fluxo de dados que satisfazem as três propriedades mínimas esperadas de um critério de teste.
  - Estabelecem uma hierarquia entre os critérios Todos-Arcos e Todos-Caminhos, mesmo na presença de caminhos não executáveis.
- Nenhum outro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

Ferramenta PokeTool

- *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.

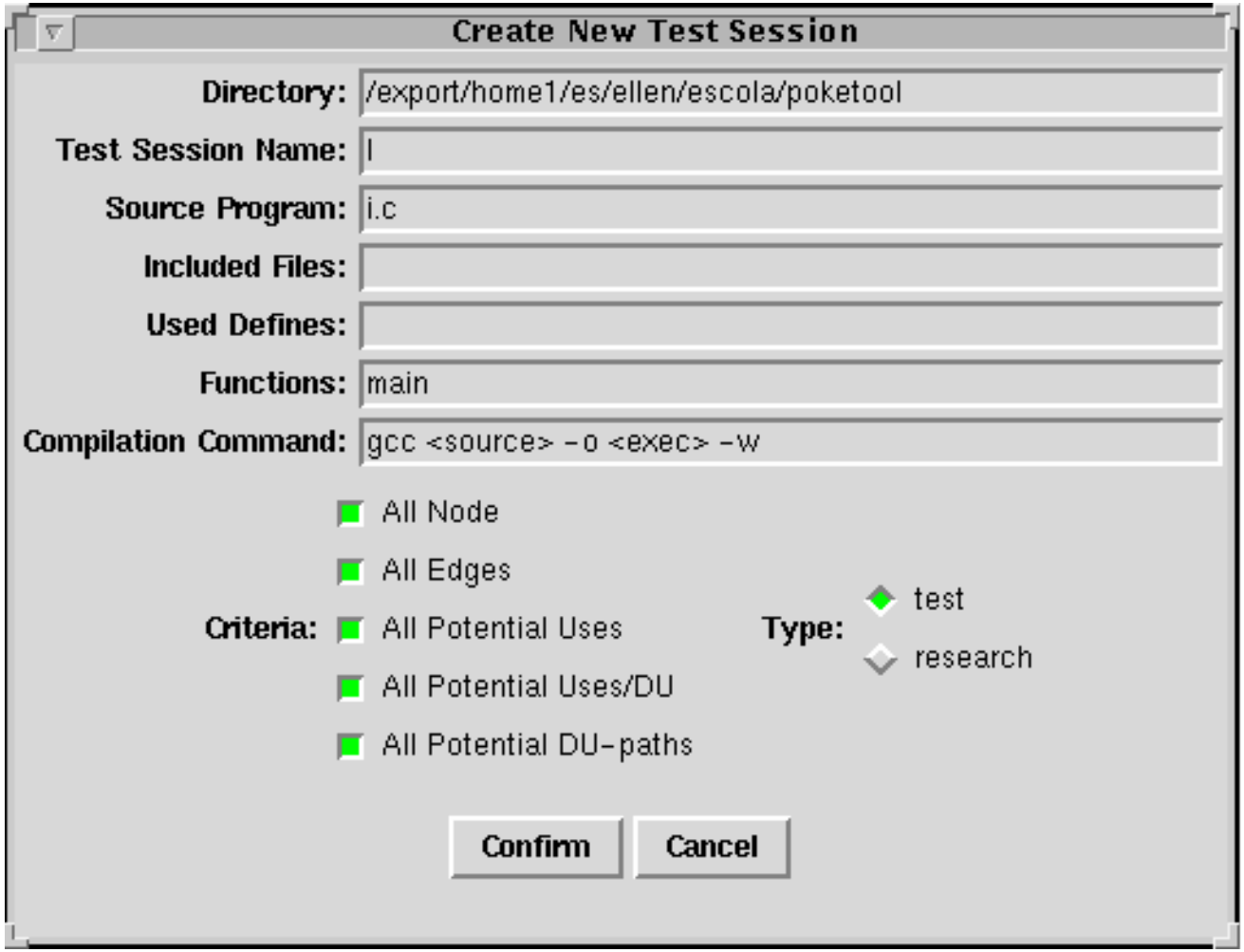
Ferramenta PokeTool

- Interface Gráfica

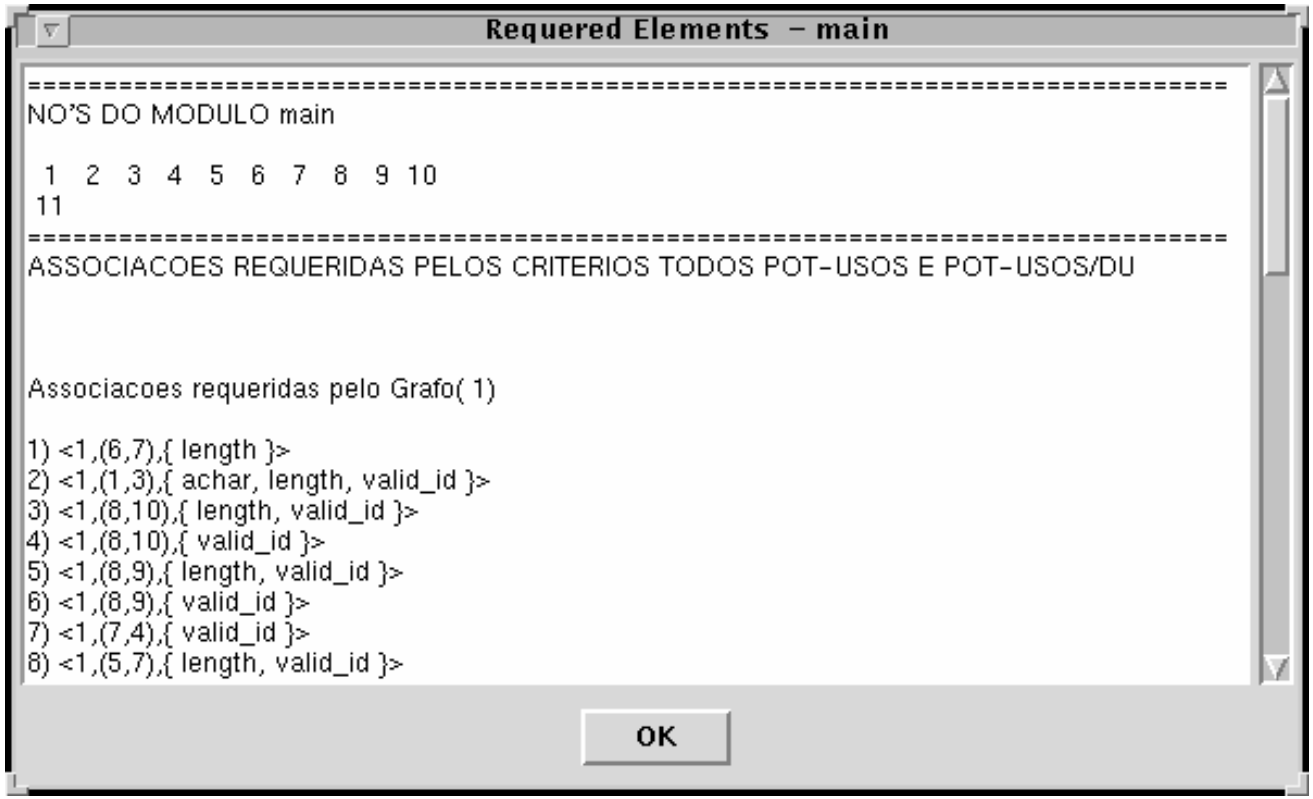


Ferramenta PokeTool

- Sessão de Teste

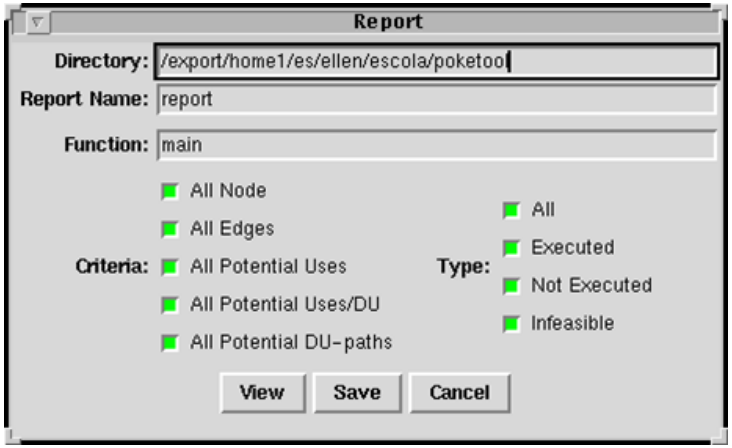


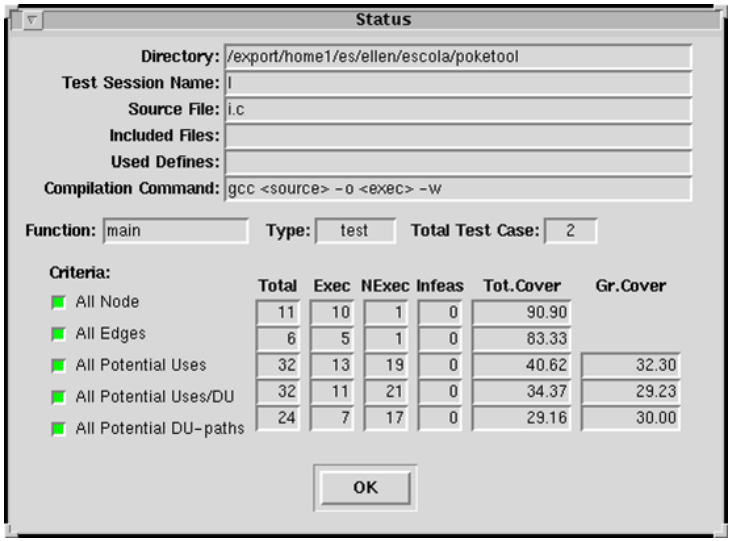
- Elementos Requeridos



Ferramenta PokeTool

- Relatórios de Teste





Ferramenta PokeTool

- Relatórios de Teste

Status

Directory:	/home/auri/identifier/poke
Test Session Name:	Identifier
Source File:	identifier.c
Included Files:	
Used Defines:	
Compilation Command:	gcc <source> -o <exec> -w

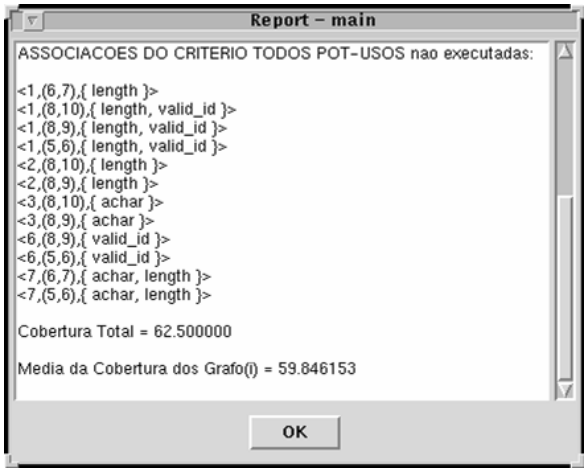
Function: main Type: test Total Test Case: 8

Criteria:	Total	Exec	NExec	Infeas	Tot.Cover	Gr.Cover
All Node	11	11	0	0	100.	
All Edges	6	6	0	0	100.	
All Potential Uses	32	29	3	0	90.62	89.46
All Potential Uses/DU	32	29	3	0	90.62	89.46
All Potential DU-paths	24	20	4	0	83.33	85.00

OK

Ferramenta PokeTool

- Relatórios de Teste



Resumindo...

- Critérios Baseados em Fluxo de Dados identificam pares definição-uso para variáveis de um programa.
- Critérios de Rapps e Weyuker
  - Grafo Def-Uso 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.

Exercício de Fixação

Exercício de Fixação

- 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.

Programa Sort (Bolha)

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