SELEÇÃO DE CASOS DE TESTES BASEADA EM FLUXO DE DADOS ATRAVÉS DOS CRITÉRIOS POTENCIAIS USOS*

José Carlos Maldonado[1], Marcos Lordello Chaim[2], Mário Jino[3]

SUMÁRIO:

São introduzidos dois novos critérios, para seleção de casos de testes, orientados pela análise de fluxo de dados -- todos-potenciais-usos e todos-potenciais-du-caminhos --, e uma análise é realizada, de acordo com uma relação de inclusão, para determinar a ordem parcial destes critérios em relação à família de critérios de Rapps e Weyuker. Adicionalmente, suas complexidades são estabelecidas e os aspectos relevantes do projeto (em andamento) de uma ferramenta que apoia a aplicação destes critérios são apresentados.

ABSTRACT:

Two new data flow analysis oriented criteria for selection of test cases -- all-potential-uses and all-potential-du-paths -- are introduced, and an analysis is performed in order to determine, according to an inclusion relation, the partial order of these criteria in relation to Rapps and Weyuker criteria family. Furthermore, their complexities are estabilished and relevant design aspects of a software tool supporting these new criteria are presented.

1 Eng. Eletr. (EESC/USP, 1978), Mestre em Telecomun./S.D.A. (INPE, 1983), Dout. Comp. e Aut. (UNICAMP; em andamento); Eng. de Software, Espec., Valid., Verif. e Testes de Software; Docente ICMSC/USP; ICMSC/USP - C.P. 668 - CEP 13560 - São Carlos - S.P.

2 Eng. Eletr. (UNICAMP, 1987), Mestr. em Eng. Elétr. (UNICAMP, em andamento); Eng. de Software, Valid., Verif. e Testes de Software; DCA/FEE/UNICAMP - C.P. 6101 - CEP 13081 - Campinas - S.P.

3 Eng. de Eletr. (ITA, 1967); Mestre Eng. Elétr. (UNICAMP, 1974); Ph.D. em C. Comput. (Univ. of Illinois-Urbana, 1978); Eng. de Software, Amb. de Desenv., Bancos de Dados; DCA/FEE/UNICAMP; C.P. 6101 - CEP 13081 - Campinas - S.P.

* Este trabalho teve o apoio financeiro do CNPq, CAPES e SID INFORMÁTICA S.A.

I - Introdução

Recentemente, têm sido introduzidos critérios para a seleção de casos de testes baseados em análise de fluxo de dados [RAP82], [WEY84], [RAP85], [FRA85], [NTA84] e [LAS83]. A análise de fluxo de dados foi inicialmente utilizada para otimização de código por compiladores e, em geral, estabelece que a ocorrência de uma variável pode ser de dois tipos: definição e uso [HEC77]. No contexto de testes de software, estratégias baseadas em análise de fluxo de dados requerem queas interações que envolvem definições de variáveis de programa e subsequentes referências que são afetadas por essas definições sejam testadas. Em [NTA88] esses critérios são objetivamente sintetizados e é apresentada uma comparação, entre algumas estratégias de testes estruturais, fundamentada essencialmente em termos de inclusão e em termos do número de casos de testes requerido para o pior caso em cada uma das estratégias. Uma estratégia A *inclui estritamente* a estratégia B se qualquer conjunto de casos de teste que satisfaz A também satisfaz B e existe algum conjunto de casos de testes que satisfaz B mas não satisfaz A, denotado por A => B. Diz-se que duas estratégias A e B são *incomparáveis* se nem A => B e nem B => A. A ordem parcial apresentada em [NTA88] é basicamente uma extensão, com inclusão de novos critérios, da ordem apresentada para uma família de critérios para a seleção de casos de testes baseada na análise de fluxo de dados, introduzida em [RAP85].

Visando a implementação de uma ferramenta de teste, introduz-se na seção III dois critérios para a seleção de casos de testes, que consistem numa extensão da família de critérios apresentada por Rapps e Weyuker. Adicionalmente, é estabelecida uma comparação em termos de inclusão com a família de critérios de Rapps e Weyuker e em termos da complexidade do conjunto de testes requeridos para o pior caso. Na seção II, são apresentados os conceitos e a família de critérios introduzidos em [WEY84] e [RAP85]. Finalmente, são discutidos, na seção IV, aspectos de implementação de uma ferramenta que apoie a utilização destes novos critérios. Nas conclusões é discutida a continuação deste trabalho.

II - A família de Critérios de Seleção de Casos de Testes baseada em Análise de Fluxo de Dados de Rapps e Weyuker

Nesta seção, são apresentados os conceitos essenciais para a compreensão dos aspectos discutidos neste trabalho e a família de critérios de seleção de casos de testes baseada em fluxo de dados introduzidos em [RAP82] e [RAP85].

II-1 - Conceitos Básicos

Um programa na sintaxe da linguagem definida em [RAP85] pode ser unicamente decomposto em um conjunto de blocos, onde cada bloco é um conjunto de comandos executados como uma unidade, ou seja, uma vez executado o primeiro comando todos os outros também o são. Um programa é representado por um grafo de controle [MCC76] onde os nós são os blocos e os arcos indicam possíveis fluxos de controle entre os blocos.

[RAP85] distingue dois tipos de uso de variáveis. O primeiro, ou afeta diretamente a computação que está sendo realizada, ou permite a saída do resultado de uma definição anterior. É denominado *uso computacional* e denotado por *c-uso*. O segundo afeta o fluxo de controle do programa, é chamado de *uso predicativo* e denotado por *p-uso*.

Diz-se que um nó contém um c-uso ou uma definição de variável se no seu bloco correspondente existir um c-uso ou uma definição de variável, respectivamente. São de interesse para a análise de fluxo de dados os *c-usos globais*, isto é, os casos em que a definição ocorre num bloco i e o seu c-uso ocorre num bloco j, diferente de i.

Seja um nó i cujo bloco correspondente tem como último comando o comando "if <expressão> then goto <rótulo>" onde as variáveis $x_1,\ldots,x_n$ ocorrem em <expressão>. Sejam dois outros nós k e j sucessores do nó i. Diz-se, então, que os arcos (i,j) e (i,k) contém p-usos de $x_1,\ldots,x_n$.

Seja x uma variável que ocorre em um programa e um caminho $(i,n_1,\ldots,n_m,j)$, $m\geq 0$, que não contém definição de x nos nós $n_1,\ldots,n_m$. Diz-se que esse é um *caminho livre de definição com respeito à (c.r.a) variável* x do nó i ao nó j e do nó i ao arco (n_m,j). Uma definição de uma variável x é uma *definição global*, denotado por def, se ela é a última definição de x que ocorre em i e existe um caminho livre de definição c.r.a. x de i para algum nó que contém c-uso de x ou algum arco que contém p-uso de x. Uma definição de uma variável x em um nó i, que não é global, é uma definição *local* se existe um c-uso de x no nó i que segue essa definição e nenhuma outra definição de x aparece entre a definição e o c-uso local.

A partir dos tipos de ocorrência de variáveis discutidos acima, [RAP85] cria um novo grafo, denominado grafo definição/uso - grafo def/uso - a partir do qual estabelece a família de critérios para seleção de casos de testes. O grafo def/uso é criado associando-se a cada nó i

os conjuntos *def(i)* e *c-uso(i)* e a cada arco (i,j) o conjunto *p-uso(i,j)*. Além do grafo def/uso , dois conjuntos são também definidos: *dcu(x,i)* e *dpu(x,i)* onde x é uma variável definida no nó i. Seja V o conjunto das variáveis, N o conjunto dos nós e E o conjunto dos arcos; define-se:

def(i) = {x ∈ V | x tem uma def no bloco i}

c-uso(i) = {x ∈ V | x tem um c-uso global no nó i}

p-uso(i,j) = {x ∈ V | x tem um p-uso no arco (i,j)}

dcu(x,i) = {j ∈ N | x ∈ def(i) ∧ x ∈ c-uso(j) ∧ existe um caminho livre de definição de i para j}

dpu(x,i) = {(j,k) ∈ E | x ∈ def(i) ∧ x ∈ p-uso(j,k) ∧ existe um caminho livre de definição de i para (j,k)}

II-2 - Família de Critérios de Rapps e Weyuker

A partir das definições e conceitos discutidos na seção anterior [RAP85], considerando P um conjunto de caminhos de um grafo G de fluxo de controle, introduz, entre outros, os critérios apresentados a seguir:

■ P satisfaz o critério *todos-usos* se para cada nó i e todo x que possui uma def em i, P inclui caminhos livre de definição c.r.a. x para todo elemento de dcu(x,i) e todo elemento de dpu(x,i). Isto é, todo uso (c-uso ou p-uso) em relação a uma def é exercitado pelo menos uma vez.

■ Um *caminho simples* é aquele em que todos os nós, exceto possivelmente o primeiro e o último, são distintos. Um *caminho livre de laços* é aquele em que todos os nós são distintos. Um caminho $(n_1,\ldots,n_j,n_k)$ é um *du-caminho* c.r.a x se n_1 tem uma definição de x e

i) n_k tem um c-uso de x e $(n_1,n_2\ldots,n_j,n_k)$ é um caminho simples livre de definição c.r.a. x, ou

ii) (n_j,n_k) tem um p-uso de x e $(n_1,n_2\ldots,n_j,n_k)$ é um caminho simples livre de definição c.r.a. x[1].

P satisfaz o critério *todos-du-caminhos* se, para cada nó i e toda variável x que possui uma def em i, P inclui todos os du-caminhos c.r.a. x.

[1] [RAP85] exige que $(n_1,\ldots,n_j)$ seja um caminho livre de laços e livre de definição c.r.a. x. A modificação introduzida visa a eliminar possíveis interpretações erroneas do critério; de forma nenhuma pretende-se alterar a semantica da definição.

[RAP85] estabelece uma ordem parcial para os critérios acima, e para os critérios todas-defs, todos-p-usos, todos-c-usos/alguns-p-usos, e todos-p-usos/alguns-c-usos, de acordo com o conceito de inclusão estrita discutido na introdução deste texto, que pode ser observada na Fig.2. Essa ordem parcial, apesar de constituir uma indicação do poder relativo desses critérios não significa necessariamente que um critério é melhor do que o outro,pois não são considerados aspectos de custos. Uma estimativa do custo relativo dos critérios pode ser obtida determinando-se o número de casos de teste necessário para satisfazer os requisitos de cada critério [NTA88].

[NTA88] apresenta uma análise do número de casos de testes requeridos para o pior caso, em função de um programa com n comandos, de um conjunto significativo de critérios de testes estruturais, entre eles os apresentados em [NTA84], [LAS83] e [RAP85]. Em relação aos critérios de Rapps e Weyuker, o critério *todos-du-caminhos* requer 2^n casos de teste, *todas-defs* requer n e todos os demais critérios requerem $O(n^2)$ casos de teste. Essa análise é apresentada detalhadamente em [WEY84].

Deve-se observar que um dado conjunto $\mathbb{P}$ de caminhos requeridos por um particular critério pode conter um caminho não executável. Um caminho não executável é aquele para o qual não existe um caso de teste que force sua execução. A determinação de que um particular caminho é executável ou não é indecidível [FRA85].

III - Extensão dos Critérios de Rapps e Weyuker

Introduzem-se dois novos critérios: *todos-potenciais-usos* e *todos-potenciais-du-caminhos*, que são fundamentalmente variações dos critérios todos-usos e todos-du-caminhos apresentados em [RAP85]. A motivação para a introdução destes critérios é que ao exercitarmos, a partir da definição de uma determinada variável x em um nó i, caminhos livre de definição c.r.a. x, independentemente da ocorrência de uso de x, ganhamos maior confiança de que a computação correta é realizada, pois pode-se verificar, por exemplo, que o valor de x foi preservado nesses caminhos, o que vem de encontro à filosofia discutida em [MYE79]: "Um erro está claramente presente se um programa não faz o que supõe-se que ele faça, mas erros estão também presentes se um programa faz o que supõe-se que não faça". O critério todos-potenciais-usos inclui o critério-todos-usos e o critério todos-potenciais-du-caminhos inclui o critério todos-du-caminhos; além disto apresentam a mesma complexidade dos critérios de Rapps e Weyuker e exigem para sua implementação menor

pré-processamento do programa a ser testado, pois não utilizam os conjuntos c-uso(i) e p-uso(i,j). Adicionalmente, podem mostrar-se mais adequados para certos tipos de erros computacionais [HOW76].

III - 1 - Critérios Potenciais Usos

Os critérios potenciais usos -- todos-potenciais-usos e todos-potenciais-du-caminhos -- requerem basicamente que caminhos livre 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.

■ *Todos-potenciais-usos* - $\mathbb{P}$ satisfaz o critério todos-potenciais-usos se, para todo nó i e para toda variável x para a qual existe uma definição em i, $\mathbb{P}$ inclui pelo menos um *caminho livre de definição* c.r.a. x do nó i para todo nó e para todo arco possível de ser alcançado a partir de i.

■ Um caminho $(n_i,\ldots,n_j,n_k)$ simples livre de definição c.r.a. x é um *potencial du-caminho* c.r.a. x se n_i tem uma definição de x.

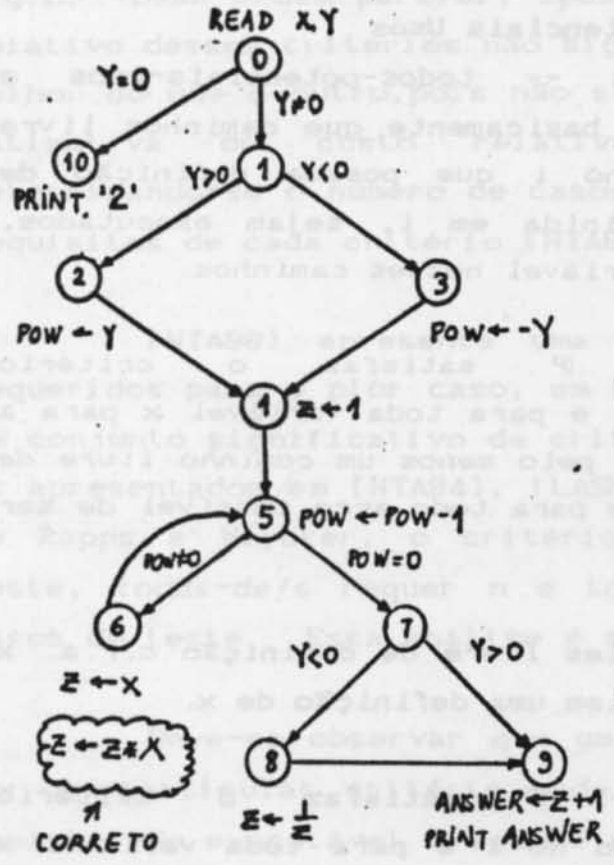
Todos-potenciais-du-caminhos - $\mathbb{P}$ satisfaz o critério todos-potenciais-du-caminhos se para todo nó i e para toda variável x para a qual existe uma definição em i, $\mathbb{P}$ inclui todos *potenciais du-caminhos* c.r.a. x em relação ao nó i.

Considere o exemplo modificado de [RAP85], cujo grafo def/uso é reproduzido na Fig.1; temos possíveis conjuntos de caminhos que satisfazem os critérios propostos por Rapps e Weyuker e os critérios potenciais usos. Suponha o defeito computacional introduzido no nó 6; para detecção deste particular defeito é necessária a execução do caminho (6,3,6). Observe que os critérios potenciais usos incluem este caminho, o que não ocorre com os critérios de Rapps e Weyuker. Este fato motiva estudos posteriores de comparação, do ponto de vista de adequação para classes de erros, dos critérios potenciais usos com os demais critérios de testes estruturais.

III - 2 - Ordem Parcial dos Critérios Potenciais Usos

A ordenação das estratégias de testes apresentada nesta seção é baseada na relação de inclusão introduzida em [RAP85] e [NTA88]. É fácil demonstrar que, conforme ilustrado na Fig.2, o critério *todos-potenciais-usos inclui estritamente* o critério *todos-usos* e que o

critério *todos-potenciais-du-caminhos inclui estritamente* o critério *todos-du-caminhos*. Ainda, pode-se extrair do exemplo da Fig.1 que os critérios todos-potenciais-usos e todos-du-caminhos são *incomparáveis*.



GRAFO DEF/USO

NÓ	C-USO	DEF	ARCO	P-USO
0	∅	⟨X,Y⟩	(0,10)	⟨Y⟩
1	∅	∅	(0,1)	⟨Y⟩
2	⟨Y⟩	⟨POW⟩	(1,2)	⟨Y⟩
3	⟨Y⟩	⟨POW⟩	(1,3)	⟨Y⟩
4	∅	⟨Z⟩	(5,6)	⟨POW⟩
5	⟨POW⟩	⟨POW⟩	(5,7)	⟨POW⟩
6	⟨X⟩	⟨Z⟩	(7,8)	⟨Y⟩
7	∅	∅	(7,9)	⟨Y⟩
8	⟨Z⟩	⟨Z⟩		
9	⟨Z⟩	∅		

$\mathbb{P}_{todos\text{-}usos}$ = ⟨(0,1,2,4,5,6,5,7,8,9), (0,1,3,4,5,6,5,7,9), (0,10), (0,1,3,4,5,7,9,), (0,1,2,4,5,7,8,9)⟩

$\mathbb{P}_{todos\text{-}du\text{-}caminhos} = \mathbb{P}_{todos\text{-}usos} \cup$ ⟨(0,1,3,4,5,7,8,9), (0,1,2,4,5,7,9)⟩

$\mathbb{P}_{todos\text{-}potenciais\text{-}usos} = \mathbb{P}_{todos\text{-}usos} \cup$ ⟨(0,1,2,4,5,6,5,6,5,7,8,9)⟩

$\mathbb{P}_{todos\text{-}potenciais\text{-}du\text{-}caminhos} = \mathbb{P}_{todos\text{-}du\text{-}caminhos} \cup$ ⟨(0,1,2,4,5,6,5,6,5,7,8,9)⟩

Figura 1 - Ilustração dos Critérios Potenciais Usos e de Rapps e Weyuker

Lema 1: Todos-potenciais-usos *inclui estritamente* todos-usos

Considere o exemplo da Fig.1. O conjunto $\mathbb{P}_{todos\text{-}usos}$ não inclui o caminho (0,1,2,4,5,6,5,6,5,7,8,9) e nenhum outro que contém um caminho livre c.r.a. z do nó 6 ao nó 6; portanto, não satisfaz o critério todos-potenciais-usos. O critério todos-potenciais-usos requer, para todo nó i, que tenha definição de variável, que pelo menos um caminho livre de definição c.r.a. x, qualquer que seja x definida em i, para todo nó e para todo arco possível de ser alcançado a partir de i seja executado, em particular para aqueles nós e aqueles arcos pertencentes a dcu(x,i) e dpu(x,i), respectivamente. □

Lema 2: Todos-potenciais-du-caminhos *inclui estritamente* todos-du-caminhos.

Considere o exemplo da Fig.1. O conjunto $\mathbb{P}_{\text{todos-du-caminhos}}$ não inclui o caminho (0,1,2,4,5,6,5,6,5,7,8,9) que contém o potencial du-caminho (6,5,6) c.r.a. z e portanto não satisfaz o critério todos-potenciais-du-caminhos que requer a execução de todos os potenciais du-caminhos. Para mostrar que o critério todos-potenciais-du-caminhos incluí o critério todos-du-caminhos, basta verificar que todo du-caminho é um potencial du-caminho.

Suponha que $(n_1,\ldots,n_j,n_k)$ seja um du-caminho c.r.a. variável x definida em n_i. Ou 1) n_k tem um c-uso de x e $(n_1 n_2,\ldots,n_k)$ é um caminho simples livre de definição c.r.a. x ou 2) (n_j,n_k) tem um p-uso de x e $(n_1,n_2,\ldots,n_j,n_k)$ é um caminho simples livre de definição c.r.a. x. Em qualquer das hipóteses, $(n_1,n_2,\ldots,n_j,n_k)$ é um caminho simples livre de definição c.r.a. x e n_i tem uma definição global de x e portanto $(n_1,n_2,\ldots,n_j,n_k)$ é um potencial du-caminho, por definição.□

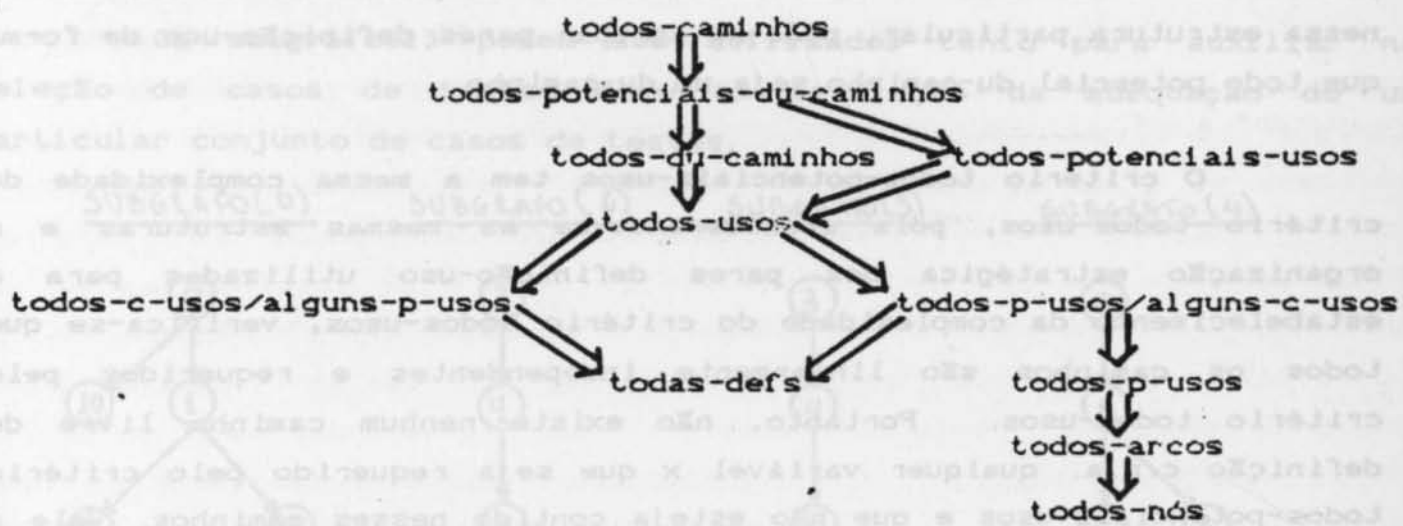


Figura 2 - Ordem parcial para os critérios potenciais usos e os de Rapps e Weyuker.

III - 3 - Complexidade dos Critérios Potenciais Usos

[WEY84] determina limites superiores do número de casos de testes necessários para satisfazer os critérios introduzidos em [RAP85]. Esses limites, denominados *complexidade* dos critérios, são obtidos determinando-se estruturas de fluxo de controle que maximizam o número de pares definição-uso e consequentemente o número de caminhos a serem exercitados pelos casos de testes. Para o critério todos-du-caminhos é mostrado que o pior caso é uma estrutura de fluxo de controle que maximiza o número de caminhos livre de laços; essa estrutura consiste em

uma seqüência de n comandos IF-THEN-ELSE que resultam, no máximo em, 2^n caminhos livre de laços. Para o critério todos-usos, é estabelecido que a estrutura de fluxo de controle que maximiza os pares definição-uso não contém junções e/ou laços, o que poderia, eventualmente, diminuir o número de casos de testes requeridos, dado por $\frac{1}{4}(n^2+4n+3)$. No entanto, dada uma estrutura consistindo de uma seqüência de n comandos IF-THEN-ELSE e m variáveis, $m>2^n$, é possível organizar estrategicamente as definições e usos dessas variáveis de forma a requerer a execução de 2^n casos de testes para satisfazer o critério todos-usos [MAL88]; pode-se dizer então que o pior caso para o critério todos-usos é $\max(m,\frac{1}{4}(n^2+4n+3))$. Porém, esta condição dificilmente ocorre em programas reais, pois é uma condição extremamente particular. Portanto, em geral, pode-se assumir que o critério todos-usos tem complexidade $O(n^2)$.

O critério todos-potenciais-du-caminhos tem a mesma complexidade que o critério todos-du-caminhos, ou seja, 2^n, pois a estrutura que maximiza o número de du-caminhos é a mesma que maximiza o número de potenciais-du-caminhos -- todo du-caminho é um potencial du-caminho e, nessa estrutura particular, pode-se agrupar pares definição-uso de forma que todo potencial du-caminho seja um du-caminho.

O critério todos-potenciais-usos tem a mesma complexidade do critério todos-usos, pois considerando-se as mesmas estruturas e a organização estratégica dos pares definição-uso utilizadas para o estabelecimento da complexidade do critério todos-usos, verifica-se que todos os caminhos são linearmente independentes e requeridos pelo critério todos-usos. Portanto, não existe nenhum caminho livre de definição c.r.a. qualquer variável x que seja requerido pelo critério todos-potenciais-usos e que não esteja contido nesses caminhos. Vale a mesma ressalva apontada para o critério todos-usos.

IV - Aspectos de Implementação de uma Ferramenta de Apoio aos Critérios Potenciais Usos

A importância de incorporar facilidades de análise de fluxo de controle em ferramentas de desenvolvimento de software para a obtenção de produtos mais confiáveis e de melhor qualidade tem sido relatada na literatura, entre elas [PRI87]. Nesta seção são discutidos aspectos relevantes que estão sendo utilizados para a definição de uma ferramenta que suporta a utilização dos critérios potenciais usos, tanto para auxiliar na seleção de casos de testes como para avaliar a adequação de um determinado conjunto de casos de testes.

Para a implementação dessa ferramenta é fundamental estender o grafo de fluxo de controle do programa associando-se a cada nó i o conjunto def(i); este grafo será referenciado por grafo *def*. Observe que não é necessária a determinação dos conjuntos c-uso(i) e p-uso(i,j).

A partir do grafo def pode-se determinar, para cada nó i que tenha definição de variável, um subgrafo, denotado por subgrafo(i). Este subgrafo fornece todos os caminhos livres de definição c.r.a. qualquer variável definida em i para todo nó e todo arco alcançável a partir de i. Estes subgrafos são obtidos percorrendo-se o grafo def em profundidade. A Fig.3 ilustra a construção dos subgrafos para o exemplo da Fig.1, de acordo com o algoritmo de [MAL88]. Na geração de cada um dos subgrafo(i), dois quaisquer nós k e j do subgrafo(i) serão considerados iguais se e somente se possuírem o mesmo número no grafo def e deff(k)=deff(j), onde deff(n) consiste do conjunto de variáveis definidas em i as quais são livres de definição no particular caminho selecionado do nó i até o nó n. Observe que os subgrafo(i) possuem somente caminhos livre de laços.

Os subgrafo(i) podem ser utilizados tanto para auxiliar na seleção de casos de testes como na avaliação da adequação de um particular conjunto de casos de testes.

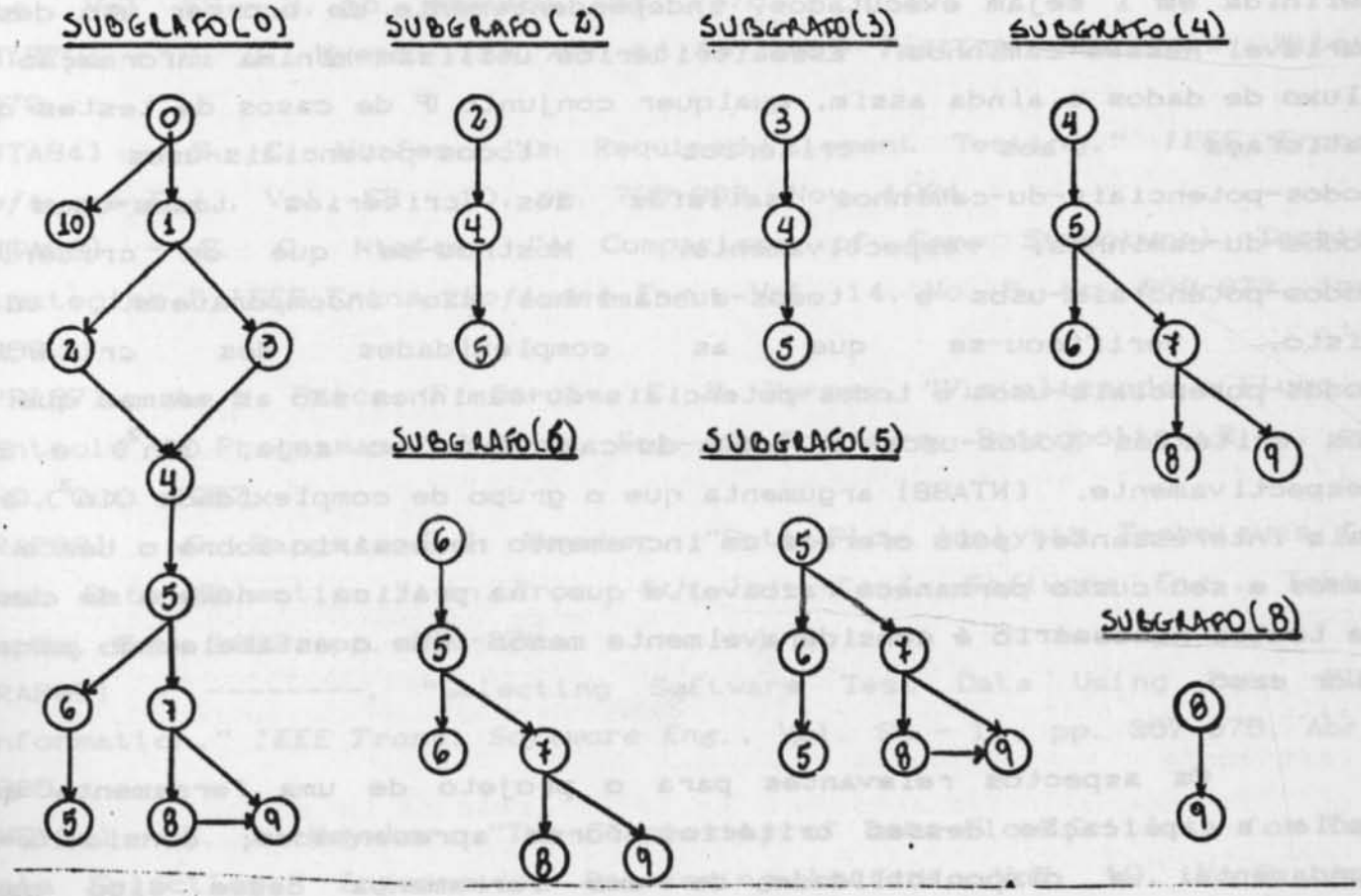


Figura 3 - Subgrafos do Exemplo da Fig.1

Para satisfazer o critério todos-potenciais-usos deve-se selecionar para cada subgrafo(i) um conjunto mínimo de caminhos que leve à execução de todos os ramos do subgrafo(i) e, então, compor esses conjuntos em um único conjunto de caminhos de testes, eliminando-se eventuais redundâncias, pois um caminho de um subgrafo(j) pode estar contido em um caminho de um subgrafo(i), conforme pode ser observado na Fig.3. Para satisfazer o critério todos-potenciais-du-caminhos deve-se obter para cada subgrafo(i) o conjunto de todos caminhos possíveis e, então, semelhantemente, compor esses conjuntos em um único conjunto de caminhos de testes. Essencialmente, um conjunto P de casos de testes satisfaz o critério todos-potenciais-uso se todos os ramos de todos os subgrafo(i) forem executados; satisfaz o critério todos-potenciais-du-caminhos se todos os possíveis caminhos de cada um dos subgrafo(i) forem executados. Para determinar-se esta adequação pode-se construir aceitadores para cada um dos caminhos, semelhantes aos propostos em [FRA85].

V - Conclusões

Apresentaram-se dois critérios de seleção de casos de testes que requerem basicamente que caminhos livre 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. Esses critérios utilizam mínima informação de fluxo de dados e ainda assim, qualquer conjunto P de casos de testes que satisfaça aos critérios todos-potenciais-usos e todos-potenciais-du-caminhos satisfaz aos critérios todos-usos e todos-du-caminhos, respectivamente. Mostrou-se que os critérios todos-potenciais-usos e todos-du-caminhos são *incomparáveis*. Além disto, verificou-se que as complexidades dos critérios todos-potenciais-usos e todos-potenciais-du-caminhos são as mesmas que as dos critérios todos-usos e todos-du-caminhos, ou seja, $O(n^2)$ e 2^n, respectivamente. [NTA88] argumenta que o grupo de complexidade $O(n^2)$ é o mais interessante, pois oferece um incremento necessário sobre o teste de ramos e seu custo permanece razoável, e que, na prática, o número de casos de testes necessário é consideravelmente menor que o estabelecido para o pior caso.

Os aspectos relevantes para o projeto de uma ferramenta que apoie a aplicação desses critérios foram apresentados; considera-se fundamental a disponibilidade de uma ferramenta desse tipo para viabilizar a realização de estudos comparativos entre os diversos

critérios de testes, bem como para auxiliar no estudo e domínio das diversas questões em aberto nessa linha de pesquisa, por exemplo: adequação dos critérios baseados em fluxo de dados para a detecção de certos tipos de erros; formas para tratar chamadas de procedimentos e recursão; etc.

Referências:

[FRA85] - F. G. Frankl e E.J. Weyuker, "Data Flow Testing Tool," in *Proc. Softfair II*, San Francisco, CA, Dez. 1985, pp.46-53.

[HEC77] - M. S. Hecht, *Flow Analysis of Computer Programs*, North Holland, 1977.

[HOW76] - W. E. Howden, "Reliability of the Path Analysis Testing Strategy," *IEEE Trans. Software Eng.*, Vol. SE - 2. No.3, 1976, pp. 208-215.

[LAS83] - J. W. Laski e B. Korel, "A Data Flow Oriented Program Testing Strategy," *IEEE Trans. Software. Eng.*, Vol. SE - 9, No.3, Maio 1983, pp. 347-354.

[MAL88] - J. C. Maldonado, M. L. Chaim, M. Jino, "Resultados do Estudo de uma Família de Critérios de Seleção de Casos de Testes Baseada em Fluxo de Dados," Relatório Técnico Interno (em preparo), DCA/FEE/UNICAMP, Campinas, 1988.

[MCC76] - T. J. McCabe, "A Complexity Measure," *IEEE Trans. Software Eng.*, Vol. SE - 2, pp. 308-320, Dez. 1976.

[MYE79] - G. J. Myers, *The Art of Software Testing*. New York: Wiley, 1979.

[NTA84] - S. C. Ntafos, "On Required Element Testing," *IEEE Trans. Software Eng.*, Vol. SE - 10, pp. 795-803, Nov. 1984.

[NTA88] - S. C. Ntafos, "A Comparison of Some Structural Testing Strategies," *IEEE Trans. Software Eng.*, Vol. 14, No. 6, pp. 868-873, Jun. 1988.

[PRI87] - A. M. Price, F. Garcia, C. B. Purper, "Visualizando o Fluxo de Controle de Programas," *I Simp. Eng. de Software*, Petropólis, R.J., pp. 1-9, Out. 1987.

[RAP82] - S. Rapps e E.J. Weyuker, "Data Flow Analysis Techniques for Test Data Selection," in *Proc. 6th Int. Conf. Software Eng.*, Tokio, Japão, Set. 1982, pp. 272-278.

[RAP85] - --------, "Selecting Software Test Data Using Data Flow Information," *IEEE Trans. Software Eng.*, Vol. SE - 11, pp. 367-375, Abril 1985.

[WEY84] - E. J. Weyuker, "The Complexity of Data Flow Criteria for Test Data Selection," *Information Processing Letters*, Vol. 19, No.2, Ago. 1984.