Verificação, Validação & Teste de Software

Jean Phelipe de Oliveira Lima

jpdol.eng16@uea.edu.br

/jpdol

Teste Estrutural

- A técnica estrutural (ou caixa-branca) estabelece os requisitos de teste com base em dada implementação, requerendo a execução de partes ou de componentes elementares do programa.
- Os caminhos lógicos do software são testados, fornecendo-se casos de teste que põem à prova tanto conjuntos específicos de condições e/ou laços, bem como pares de definições e usos de variáveis.
- Os critérios pertencentes à técnica estrutural são classificados com base na **complexidade**, no **fluxo de controle** e no **fluxo de dados**

Teste Estrutural

- Limitações (Teste de Software):
  - não existe um procedimento de teste de propósito geral que possa ser usado para provar a correção de um programa;
  - dados dois programas, é indecidível se eles computam a mesma função;
  - é indecidível, em geral, se dois caminhos de um programa, ou de programas diferentes, computam a mesma função;
  - é indecidível, em geral, se um dado caminho é executável, ou seja, se existe um conjunto de dados de entrada que leve à execução desse caminho.

Teste Estrutural

- Limitações (Teste Estrutural):
  - caminhos ausentes: se o programa não implementa algumas funções, não existirá um caminho que corresponda àquela função; consequentemente, nenhum dado de teste será requerido para exercitá-la; e
  - correção coincidente: o programa pode apresentar, coincidentemente, um resultado correto para dada entrada em particular, satisfazendo um requisito de teste e não revelando a presença de um defeito; entretanto, se escolhida outra entrada, o resultado obtido poderia ser incorreto.

Definições e Conceitos Básicos

- O teste estrutural baseia-se no conhecimento da estrutura interna do programa, sendo os aspectos de implementação fundamentais para a geração/seleção dos casos de teste associados.
- Em geral, a maioria dos critérios da técnica estrutural utiliza uma representação de programa conhecida como “Grafo de Fluxo de Controle” (GFC) ou “Grafo de Programa”.
- Um programa P pode ser decomposto em um conjunto de blocos disjuntos de comandos. A execução do primeiro comando de um bloco acarreta a execução de todos os outros comandos desse bloco, na ordem dada. Todos os comandos de um bloco, possivelmente com exceção do primeiro, têm um único predecessor e exatamente um único sucessor, exceto possivelmente o último comando.

Definições e Conceitos Básicos

- Usualmente, a representação de um programa P como um GFC (G = (N, E, s)) consiste em estabelecer uma correspondência entre vértices (nós) e blocos e em indicar possíveis fluxos de controle entre blocos por meio das arestas (arcos).
- Assume-se que um GFC é um grafo orientado, com um único nó de entrada s $\in$ N e um único nó de saída o $\in$ N , no qual cada vértice representa um bloco indivisível de comandos e cada aresta representa um possível desvio de um bloco para outro. Quando essas condições não forem satisfeitas, esse fato será indicado explicitamente.

Definições e Conceitos Básicos

- Cada bloco de comandos tem as seguintes características:
  - uma vez que o primeiro comando do bloco é executado, todos os demais são executados sequencialmente;
  - não existe desvio de execução para nenhum comando dentro do bloco.

Definições e Conceitos Básicos

- Com base no GFC podem ser escolhidos os elementos que devem ser executados, caracterizando, assim, o teste estrutural.

- Seja um GFC G = (N, E, s) em que N representa o conjunto de nós, E o conjunto de arcos, e s o nó de entrada. Um “caminho” é uma sequência finita de nós (n_1 , n_2 , . . . , n_k), k ≥ 2, tal que existe um arco de n_i para n_{i+1} para i = 1 , 2 , . . . , k − 1. Um caminho é um “caminho simples” se todos os nós que compõem esse caminho, exceto possivelmente o primeiro e o último, são distintos; se todos os nós são distintos, diz-se que esse caminho é um “caminho livre de laço”. Um “caminho completo” é aquele em que o primeiro nó é o nó de entrada e o último nó é um nó de saída do grafo G .

Definições e Conceitos Básicos

- Seja IN(x) e OUT(x) o número de arcos que entram e que saem do nó x , respectivamente. Assumimos IN(s) = 0, tal que s é o nó de entrada, e OUT (o) = 0, tal que o é o nó de saída.
- As ocorrências de uma variável em um programa podem ser uma “definição”, uma “indefinição” ou um “uso”. Usualmente, os tipos de ocorrências de variáveis são definidos por um modelo de fluxo de dados.
- uma definição de variável ocorre quando um valor é armazenado em uma posição de memória. Em geral, em um programa, uma ocorrência de variável é uma definição se esta estiver:
    i. no lado esquerdo de um comando de atribuição;
    ii. em um comando de entrada; ou
    iii. em chamadas de procedimentos como parâmetro de saída.
- A passagem de valores entre procedimentos por meio de parâmetros pode ser por valor, referência ou por nome

Definições e Conceitos Básicos

- Se a variável for passada por referência ou por nome, considera-se que seja um parâmetro de saída.
- As definições decorrentes de possíveis definições em chamadas de procedimentos são diferenciadas das demais e são ditas definidas por referência.
- Por outro lado, diz-se que uma variável está indefinida quando não se tem acesso ao seu valor ou sua localização deixa de estar definida na memória.

Definições e Conceitos Básicos

- A ocorrência de uma variável é um uso quando a referência a essa variável não a definir.
- Dois tipos de usos são caracterizados: “c-uso” e “p-uso”.
  - C-uso afeta diretamente uma computação realizada ou permite que o resultado de uma definição anterior possa ser observado;
  - P-uso afeta diretamente o fluxo de controle do programa.
- Observa-se que, enquanto c-usos estão associados aos nós do GFC, p-usos estão associados a seus arcos.

Definições e Conceitos Básicos

- Considere uma variável x definida em um nó i , com uso em um nó j ou em um arco que chega em j . Um caminho (i, n_1 ,..., n_m , j), m≥0 que não contenha definição de x nos nós n_1 ,..., n_m, é chamado de “caminho livre de definição” (x do nó i ao nó j e do nó i ao arco (n_m,j)).
- Um nó i possui uma “definição global” de uma variável x se ocorre uma definição de x no nó i e existe um caminho livre de definição de i para algum nó ou para algum arco que contém um c-uso ou um p-uso, respectivamente, da variável x . Um c-uso da variável x em um nó j é um “c-uso global” se não houver uma definição de x no nó j precedendo este c-uso; caso contrário, é um “c-uso local”.

Critérios de teste estrutural

- Os critérios estruturais são, em geral, classificados em:

  i. critérios baseados na complexidade;
  ii. critérios baseados em fluxo de controle;
  iii. critérios baseados em fluxo de dados.

- A seguir são apresentados e discutidos os principais critérios de teste associados a cada uma dessas classes, em nível de unidade.

Critérios Baseados na Complexidade

- Os critérios baseados na complexidade utilizam informações sobre a complexidade do programa para derivar os requisitos de teste.
- Critério de McCabe (Teste de Caminho Básico): utiliza a complexidade ciclomática do GFC para derivar os requisitos de teste.
- A complexidade ciclomática é uma métrica de software que proporciona uma medida quantitativa da complexidade lógica de um programa.

Teste do Caminho Básico

- o valor da complexidade ciclomática estabelece o número de caminhos linearmente independentes do conjunto básico de um programa, oferecendo um limite máximo para o número de casos de teste que devem ser derivados a fim de garantir que todas as instruções sejam executadas pelo menos uma vez.
- Um caminho linearmente independente é qualquer caminho do programa que introduza pelo menos um novo conjunto de instruções de processamento ou uma nova condição. Quando estabelecido em termos de um GFC, um caminho linearmente independente deve incluir pelo menos um arco que não tenha sido atravessado antes que o caminho seja definido. Assim, cada novo caminho introduz um arco.

Teste do Caminho Básico

- Caminhos linearmente independentes estabelecem um conjunto básico para o GFC.
- Ou seja, se casos de teste puderem ser projetados a fim de forçar a execução desses caminhos (um conjunto básico), cada instrução do programa terá a garantia de ser executada pelo menos uma vez e cada condição terá sido executada com verdadeiro e falso. É importante observar que o conjunto básico não é único; de fato, diferentes conjuntos básicos podem ser derivados para determinado GFC.

Teste do Caminho Básico

- Para saber quantos caminhos devem ser procurados, é necessário calcular a complexidade ciclomática V (G), que pode ser computada destas três maneiras:
  1. o número de regiões em um GFC. Uma região pode ser informalmente descrita como uma área incluída no plano do grafo. O número de regiões é computado contando-se todas as áreas delimitadas e a área não delimitada fora do grafo;
  2. V (G) = E − N + 2, tal que E é o número de arcos e N é o número de nós do GFC G;
  3. V (G) = P + 1, tal que P é o número de nós predicativos contido no GFC G.

Teste do Caminho Básico

- O valor da complexidade ciclomática V (G) oferece um limite máximo para o número de caminhos linearmente independentes que constitui o conjunto básico e, consequentemente, um limite máximo do número de casos de teste que deve ser projetado e executado para garantir a cobertura de todas as instruções de programa.
- Segundo Pressman, visto que o número de regiões aumenta com o número de caminhos de decisão e laços, a métrica de McCabe oferece uma medida quantitativa da dificuldade de conduzir os testes e uma indicação da confiabilidade final.

Critérios Baseados em fluxo de controle

- Os critérios baseados em fluxo de controle utilizam apenas características de controle da execução do programa, como comandos ou desvios, para determinar quais estruturas são necessárias.
- Os critérios mais conhecidos dessa classe são:
  - Todos-Nós: exige que a execução do programa passe, ao menos uma vez, em cada vértice do GFC; ou seja, que cada comando do programa seja executado pelo menos uma vez.
  - Todas-Arestas (ou Todos-Arcos): requer que cada fluxo de aresta do grado, isto é, cada desvio de fluxo de controle do programa, seja exercitada pelo menos uma vez;
  - Todos-Caminhos: requer que todos os caminhos possíveis do programa sejam executados.

Critérios Baseados em fluxo de dados

- Os critérios baseados em fluxo de dados utilizam a análise de fluxo de dados como fonte de informação para derivar os requisitos de teste.
- Uma característica comum aos critérios dessa categoria é que eles requerem o teste das interações que envolvam definições de variáveis e subsequentes referências a essas definições.
- Portanto, para a derivação de casos de teste, tais critérios baseiam-se nas associações entre a definição de uma variável e seus possíveis usos subsequentes.

Critérios Baseados em fluxo de dados

- Principais Critérios baseados em fluxo de dados:
  - Critérios de Rapps e Weyuker;
  - Critérios Patenciais-Usos;

Verificação, Validação & Teste de Software

Jean Phelipe de Oliveira Lima

jpdol.eng16@uea.edu.br

/jpdol