
Teste de Software

Técnicas Funcionais

Particionamento de Equivalência

- Dois passos:

1. Identificar classes de equivalência;
2. Definir os casos de teste a partir das classes de equivalência.

Particionamento de Equivalência

- Divide o domínio de entrada do programa em classes de dados (classes de equivalência).
- Os casos de teste são derivados a partir das classes de equivalência.
- Idealmente, um teste de um valor representativo de uma classe é equivalente a um teste de qualquer outro valor da classe; visa identificar classes de erros.

Particionamento de Equivalência

- **Classes de Equivalência:** representa um conjunto de estados válidos ou inválidos das condições de entrada.
- **Condições de Entrada:** usualmente expressar em uma sentença ou frase da especificação.

Particionamento de Equivalência

■ Identificação das Classes de Equivalência

- Cada condição de entrada caracteriza duas ou mais classes.
- Condições de Entrada **Válidas**: entradas válidas e esperadas do programa.
- Condições de Entrada **Inválidas**: outros possíveis valores associados à condição; entradas não esperadas.

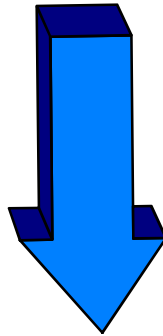
Particionamento de Equivalência

- **Identificação da Classes de Equivalência**
- Se a condição de entrada:
 - **Intervalo:** uma classe válida e duas inválidas;
 - **Valor específico:** uma classe válida e duas inválidas;
 - **Membro de conjunto:** uma classe válida e uma inválida;
 - **Booleana:** uma classe válida e uma inválida;

Particionamento de Equivalência

Identificação de Classes de Equivalência

Se os elementos de uma classe não são tratados de maneira idêntica.



Divide-se a Classe de Equivalência em classes menores.

Particionamento de Equivalência

■ Identificação dos Casos de teste

- Enumerar cada classe de equivalência. Atribua um único número para cada classe de equivalência.
- Elaborar casos de teste relacionados com as classes de equivalências **válidas** até todas serem cobertas.
- Escreva um caso de teste cobrindo o maior **número de classes** de equivalência possível.

Particionamento de Equivalência

- **Identificação dos Casos de Teste (cont.)**
 - Elaborar casos de teste relacionados com as classes de equivalência **inválidas** até **todas** serem cobertas.
 - Escreva **um caso de teste para cada** classe de equivalência **inválida** não coberta.

Particionamento de Equivalência

■ Exemplo:

Usuário pode "dispar" para o banco usando seu computador pessoal, fornecer uma senha de seis dígitos e uma série de comandos-chaves que acionam várias funções bancárias.

DDD: em branco ou número de 4 dígitos.

Prefixo: número de 3 dígitos que não se iniciam por 0 ou 1.

Sufixo: número de 4 dígitos.

Senha: valor numérico de 6 dígitos.

Comandos: "aplicação", "depósito", "pagamento de conta", "extrato", "saldo", etc.

Particionamento de Equivalência

■ DDD:

- Condição de entrada **booleana**: o DDD pode ou não estar presente. (1,2)
- Condição de entrada **intervalo**: os valores definidos entre 0011 e 0999. (3,4,5)

■ Prefixo:

- Condição de entrada **intervalo**: valor especificado menor que 999 e maior que 200. (6,7,8)

■ Sufixo:

- Condição de entrada **valor**: valor de quatro dígitos. (9,10,11)

Particionamento de Equivalência

- **Senha:**

- Condição de entrada **booleana**: uma senha pode ou não estar presente. (12,13)
- Condição de entrada **valor**: cadeia de 6 caracteres numéricos. (14,15,16)

- **Comando:**

- Condição de entrada **conjunto**: contendo os comandos anotados anteriormente. (17,18)

Particionamento de Equivalência

■ Exemplo: Identificação dos Casos de Teste

Caso de Teste #1

DDD: 0192

Prefixo: 311

Sufixo: 5149

Senha: 675494

Comando: "depósito"

São exercitadas as classes válidas: **1,3,7,10,12, 15 e 17.**

Particionamento de Equivalência

■ Exemplo: **Identificação de Casos de Teste**

Caso de Teste #2

DDD: ausente => classe válida exercitada **2**

Caso de Teste #3

DDD: 456 => classe inválida exercitada: **4**

Caso de Teste #4

DDD: 9999 => classe inválida exercitada: **5**

...

Caso de Teste #16

Comando: "Aplicação em Commodities" => classe inválida exercitada: **18**

Particionamento de Equivalência

■ Vantagens

- Reduz o tamanho do domínio de entrada, criando casos de teste baseados somente na especificação.
- Útil para testar sistemas de informação, onde as variáveis de entrada são bem identificadas e recebem valores distintos.

Particionamento de Equivalência

■ Desvantagens

- Pouco útil para sistemas cujo domínio de entrada é simples porém de processamento complexo.
- A execução de um elemento de uma classe de equivalência sugere que todos os elementos serão processados de maneira semelhante. **Nem sempre isto ocorre.**
- Não prevê a combinação de entradas para a elaboração de casos de teste.

Análise de Valores Limites

- Um número maior de erros tende a ocorrer nos **limites** (fronteiras) das **classes de equivalência** do domínio do programa.
- A análise de valores limite leva à seleção de casos de teste que exercitam as **condições limites** das classes de equivalência dos **domínios de entrada** e de **saída**.

Análise de Valores Limites

- **Diferenças em relação ao Particionamento de Equivalência:**

- requer que **um** ou **mais** elementos sejam selecionados tal que **cada condição limite** de uma classe seja um objetivo de teste (**ao invés** de selecionar **qualquer** elemento).
- **casos de teste** derivados a partir tanto das condições de **entrada** quanto das condições de **saída**.

Análise de Valores Limites

- **Identificação das classes de equivalência**
 - **Se condição de entrada ou de saída:**
 - **intervalo (a,b):** casos de teste para valores *a* e *b*; imediatamente **abaixo** e **acima** de *a* e de *b*.
 - **série de valores:** casos de teste para **mínimo** e **máximo**, imediatamente **abaixo** e **acima** do mínimo e do máximo.

Análise de Valores Limites

- Se **condição de entrada** ou de **saída**:
 - **Conjunto ordenado**: casos de teste para o primeiro e último elemento
 - **Estruturas de dados** internas com limites pré-estabelecidos: casos de teste para exercitar os limites.
 - Utilize sua criatividade para determinar outras condições limites.

Análise de Valores Limites

- **Exemplo:** Considerando o programa anterior, emitir um relatório de "Extrato" com as operações realizadas nos últimos 7 dias. Este relatório deve ter um mínimo de 1 e um máximo de 20 operações; deve-se ajustar o número de dias para apresentar somente 20 operações.

Análise de Valores Limites

- Exemplo:
 - **Condição de Entrada:**
 - **DDD: condição intervalo** (0011,0999)

Casos de Teste:

1. DDD: 0011
2. DDD: 0999
3. DDD: 0010
4. DDD: 1000

Análise de Valores Limites

- **Exemplo:**
 - **Condição de Saída:**
Número de Operações: condição série de valores: (0 a 20)

Casos de Teste:

Número de Operações: -1
Número de Operações: 0
Número de Operações: 1
Número de Operações: 19
Número de Operações: 20
Número de Operações: 21
Número de Operações: 45

Exercício:

- ◆ Usando o particionamento de equivalência defina os casos de teste para o teste funcional da seguinte aplicação:
 - Um programa que dado o valor de um produto e o valor pago pelo cliente calcula o valor do troco e indica o número de notas e moedas de forma a usar a menor quantidade de notas e moedas para formar o troco.
 - Execute o programa troco.exe com os dados de teste gerados e anote os resultados obtidos. Não esqueça de anotar os casos de teste que identificaram a presença de defeitos no software.