

Teste Funcional (parte 2)

Prof. Otávio Lemos (UNIFESP)

Prof. Fabiano Ferrari (UFSCar)

- Myers – experiência → casos de teste que exploram condições limites têm maior probabilidade de encontrar defeitos
- Valores imediatamente acima ou abaixo dos limitantes das classes de equivalência
- Análise de Valor Limite: usado em conjunto com Particionamento de Equivalência – não escolher dados de teste aleatoriamente – selecionar de forma a explorar limites
- Não existem diretrizes bem definidas, somente recomendações

- 1 se a CE especifica um intervalo de valores – dados de teste para os limites do intervalo e dados de teste imediatamente subsequentes → explorando classes inválidas vizinhas
 - Exemplo: classe válida estiver no intervalo $-1,0$ e $+1,0$
– dados de teste: $-1,0$; $+1,0$; $-1,001$ e $+1,001$
- 2 se a CE especifica uma quantidade de valores, por exemplo, de 1 a 255 valores – dados de teste: \emptyset ; somente um valor; 255 valores; e 256 valores
- 3 usar a diretriz (i) para as condições de saída
- 4 usar a diretriz (ii) para as condições de saída
- 5 se a entrada ou saída for um conjunto ordenado, maior atenção ao primeiro e último elemento
- 6 usar a intuição para definir outras condições limite

- Para o programa de busca de caracter (Programa “Cadeia de Caracteres”), adicionar os seguintes CTs:

Variáveis de Entrada				Saída Esperada
T	CC	C	O	
21				<aguarda nova entrada>
0				<aguarda nova entrada>
1	a	a	s	Caracter aparece na posição 1 da cadeia
1	a	x	n	Caracter não pertence à cadeia
20	abcdefghijklmnpqrst	a	s	Caracter aparece na posição 1 da cadeia
20	abcdefghijklmnpqrst	t	n	Caracter aparece na posição 20 da cadeia

- Similar ao Particionamento de Equivalência, porém considera recomendações para dados de teste – provavelmente mais efetivo

- Combina Particionamento e Análise de Valor Limite
- Requer ao menos dois CTs de cada partição, para minimizar problema de defeitos coincidentes
 - Exemplo: programa que calcula o quadrado de um número; dado de teste – 2; resultado 4 é correto, mas pode ter vindo de uma soma $2 + 2$.
- Várias diretrizes...

- Para domínio de entrada, selecionar valores de entrada da seguinte forma:
 - 1 Valores discretos: testar todos os valores
 - 2 Intervalo de valores: testar os extremos e um valor no interior do intervalo
- Para domínio de saída, selecionar valores de entrada que resultem nos valores de saída (tipo da entrada pode ser \neq tipo de saída; Ex.: entrada numérica, saída booleana); escolher como entrada valores que explorem os valores de saída da seguinte forma:
 - 1 Valores discretos: gerar cada um deles
 - 2 Intervalo de valores: gerar cada um dos extremos e ao menos um valor no interior do intervalo

- Devem ser explorados na entrada e na saída; Ex.: espaço em branco que pode ser interpretado como zero em um campo numérico.
- Casos especiais (por ex., valor zero): sempre selecionados individualmente, mesmo que esteja dentro de um intervalo
- O mesmo para valores nos limites da representação binária dos dados; Ex.: para campos inteiros de 16 bits, selecionar valores -32768 e +32767

- Valores que correspondem a entradas inválidas – incluir nos CTs – assegurar que o software rejeita
- Tentar gerar valores inválidos, os quais não devem ser bem sucedidos.
- Importante – limites dos intervalos numéricos (tanto inferior como superior); valores imediatamente fora dos limites desses intervalos e também os valores imediatamente subsequentes aos limites do intervalo e pertencentes ao mesmo

- Checar o limite para números reais pode não ser exato – ainda assim, incluir verificação como caso de teste
- Definir margem de erro de tal forma que, se ultrapassada, o valor pode ser considerado distinto
- Além disso, selecionar números reais pequenos e também zero

- Quando se usa arranjo, tanto como entrada como saída – considerar o fato do tamanho do arranjo ser variável, bem como dos dados serem variáveis.
- Elementos do arranjo devem ser testados como se fossem variáveis comuns
- Além disso – tamanho do arranjo deve ser testado com valores intermediários, mínimo e máximo
- Para simplificar o teste – considerar linhas e colunas de um arranjo como se fossem subestruturas a serem testadas separadamente. Assim – testar arranjo em isolamento; como uma coleção de subestruturas, e testar cada subestrutura independentemente

- Explorar comprimentos variáveis e também validar caracteres que compõem; Ex.: apenas alfabéticos; alfanuméricos; com caracteres especiais.
- **Observem que várias dessas diretrizes são também aplicáveis a Particionamento de Equivalência e Análise de Valor Limite**

- Para o programa de busca de caracter, adicionar os seguintes CTs:

Entrada				Saída esperada
T	CC	C	O	
a				<aguarda nova entrada>
1.0				<aguarda nova entrada>
1	!	' '	n	Caracter não pertence à cadeia
1	}	~	n	Caracter não pertence à cadeia
20	!"#\$%&()*+' /01234567	!	s	Caracter aparece na posição 1 da cadeia
		"	s	Caracter aparece na posição 2 da cadeia
		+	s	Caracter aparece na posição 10 da cadeia
		6	s	Caracter aparece na posição 19 da cadeia
		7	n	Caracter aparece na posição 20 da cadeia
2	ab	b	nao	Caracter aparece na posição 2 da cadeia
3	a2b	2	0	Caracter aparece na posição 2 da cadeia

- Evolua o conjunto de casos de teste do programa do pequeno mercado (Aula 4) de acordo com o critério Teste Funcional Sistemático.