



# Critério Análise do Valor Limite

## Técnica Caixa-Preta

Auri Marcelo Rizzo Vincenzi<sup>1</sup>, Márcio Eduardo Delamaro<sup>2</sup> e  
José Carlos Maldonado<sup>2</sup>

<sup>1</sup>Instituto de Informática  
Universidade Federal de Goiás

<sup>2</sup>Instituto de Ciências Matemáticas e de Computação  
Universidade de São Paulo

# Organização

Análise do Valor Limite

Introdução

Técnica de Aplicação

Exemplo de Aplicação

Aplicabilidade e Limitações

Resumo

Exercício

Leitura Recomendada



## Análise do Valor Limite

### Introdução

### Técnica de Aplicação

### Exemplo de Aplicação

### Aplicabilidade e Limitações

### Resumo

### Exercício

### Leitura Recomendada

# Introdução (1)

- ▶ Um dos critérios de teste mais básico que existe.
- ▶ Auxilia na seleção de um pequeno subconjunto de casos de teste que mantém uma boa cobertura do código.
- ▶ Considerando o exemplo utilizado anteriormente (extraído de Copeland (2004)):

0 – 16	Não empregar.
16 – 18	Pode ser empregado tempo parcial.
18 – 55	Pode ser empregado tempo integral.
55 – 99	Não empregar.

- ▶ Observe que os limites, tal como o 16, aparece em duas classes de equivalência. O mesmo ocorre com o 18 e o 55.

# Introdução (2)

- As condições acima, na verdade, deveriam ser escritas como:

$0 \leq idade < 16$	Não empregar.
$16 \leq idade < 18$	Pode ser empregado tempo parcial.
$18 \leq idade < 55$	Pode ser empregado tempo integral.
$55 \leq idade < 99$	Não empregar.

ou

$0 \leq idade \leq 15$	Não empregar.
$16 \leq idade \leq 17$	Pode ser empregado tempo parcial.
$18 \leq idade \leq 54$	Pode ser empregado tempo integral.
$55 \leq idade \leq 99$	Não empregar.

- Na primeira regra, 16 não deve ser incluído.
- Na segunda 16 pode ser empregado em tempo parcial.

# Introdução (3)

A implementação abaixo implementa as regras acima:

```
1 if (idade >= 0 && idade <= 15)
2     empregar = "NAO";
3 if (idade >= 16 && idade <= 17)
4     empregar = "PAR";
5 if (idade >= 18 && idade <= 54)
6     empregar = "INT";
7 if (idade >= 55 && idade <= 99)
8     empregar = "NAO";
```

(extraído de Copeland (2004))

► Valores limites a serem considerados:

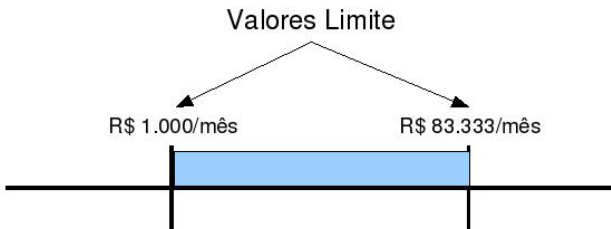
- {−1, 0, 1}, {14, 15, 16}
- {15, 16, 17}, {16, 17, 18}
- {17, 18, 19}, {53, 54, 55}
- {54, 55, 56}, {98, 99, 100}

# Passos de Aplicação

1. Identificar as classes de equivalência (requisitos de teste do critério).
2. Identificar os limites de cada classe.
3. Criar casos de teste para os limites escolhendo:
  - ▶ Um ponto abaixo do limite.
  - ▶ O limite.
  - ▶ Um ponto acima do limite.
4. Observe que “acima” e “abaixo” são termos relativos e dependente do valor dos dados.
  - ▶ Números inteiros: limite = 16; abaixo = 15; acima = 17.
  - ▶ Números reais: limite = \$5,00; abaixo = \$4,99; acima = \$5,01.
5. Casos de teste adicionais podem ser criados dependendo dos recursos disponíveis.

# Definição das Classes (1)

- Mais adequado para entradas que apresentam valores contínuos.



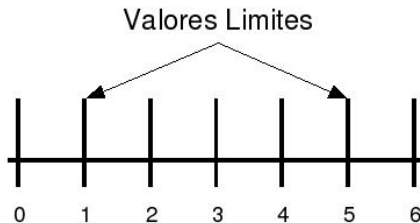
(extraído de Copeland (2004))

- Dados de teste para o limite inferior: {\$999, \$1.000, \$1.001}
- Dados de teste para o limite superior: {\$83.332, \$83.333, \$83.334}



## Definição das Classes (2)

- Intervalo de dados discretos (hipotecas de 1 a 5 casas):



(extraído de Copeland (2004))

## Definição das Classes (3)

Renda	# Moradores	Resultado	Descrição
\$1,000	1	Válido	Mín. renda, mín. moradores
\$83,333	1	Válido	Max. renda, mín. moradores
\$1,000	5	Válido	Mín. renda, max.. moradores
\$83,333	5	Válido	Max. renda, max. moradores
\$1,000	0	Inválido	Mín. renda, abaixo mín. moradores
\$1,000	6	Inválido	Mín. renda, acima max. moradores
\$83,333	0	Inválido	Max. renda, abaixo mín. moradores
\$83,333	6	Inválido	Max. renda, acima max. moradores
\$999	1	Inválido	Abaixo mín. renda, mín. moradores
\$83,334	1	Inválido	Acima max. renda, mín. moradores
\$999	5	Inválido	Abaixo mín. renda, max. moradores
\$83,334	5	Inválido	Acima max. renda, max. moradores



Análise do Valor Limite  
Introdução  
Técnica de Aplicação

Exemplo de Aplicação  
Aplicabilidade e Limitações

Resumo

Exercício

Leitura Recomendada

## Programa Identifier (1)

Especificação (extraído de Maldonado et al. (2004)):

*O programa deve determinar se um identificador é válido ou não em Silly Pascal (uma variante do Pascal). 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.*

Exemplos de Identificadores:

abc12 (válido);

cont\*1 (inválido);

1soma (inválido);

a123456 (inválido)

# Programa Identifier (cont.)

Classes de Equivalência:

Condições de Entrada	Classes Válidas	Classes Inválidas	
Tamanho $t$ do identificador	$1 \leq t \leq 6$ (1)	$t < 1$ (2)	$t > 6$ (3)
Primeiro caractere $c$ é uma letra	Sim (4)	Não (5)	
Só contém caracteres válidos	Sim (6)	Não (7)	

## Programa Identifier (2)

Conjunto de Teste (<http://pt.wikipedia.org/wiki/ASCII>):

Identificador	Resultado	Descrição
a	Válido	Primeiro minúsculo válido, tam. mínimo
b3	Válido	Segundo minúsculo válido, tam. mínimo
Xkl	Válido	Penúltimo maiúsculo válido, tam. mínimo
Z9	Válido	Último maiúsculo válido, tam. acima mínimo
xkl	Válido	Penúltimo minúsculo válido, tam. mínimo
zAaZ1	Válido	Último minúsculo válido, tam. abaixo máximo
AaZz91	Válido	Primeiro maiúsculo válido, tam. máximo
BaZz91	Válido	Segundo maiúsculo válido, tam. máximo
abcdefg	Inválido	Caracteres válidos, tam. acima máximo
@	Inválido	Primeiro minúsculo abaixo, tam. mínimo
[	Inválido	Primeiro minúsculo acima, tam. mínimo
]	Inválido	Primeiro maiúsculo abaixo, tam. mínimo
{	Inválido	Primeiro maiúsculo acima, tam. mínimo
A-&\$#	Inválido	Caracteres inválidos, tam. máximo

## Outros Exemplos

- ▶ Outros exemplos do critério Particionamento em Classe de Equivalência pode ser encontrado no Capítulo 4 do livro de (Copeland, 2004).

# Aplicabilidade e Limitações

- ▶ Reduz significativamente o número de casos de teste em relação ao teste exaustivo.
- ▶ Mais adequado para o teste de produtos com domínios de entrada divididos em intervalos ou conjuntos.
- ▶ Aplicável em todas as fases de teste: unidade, integração, sistema e aceitação.





Análise do Valor Limite  
Introdução  
Técnica de Aplicação

Exemplo de Aplicação  
Aplicabilidade e Limitações

Resumo

Exercício

Leitura Recomendada

# Resumo

- ▶ Reduz o tamanho do conjunto de teste a ser utilizado.
- ▶ Testa os limites das classes de equivalência nos quais diferentes tipos de erros estão escondidos.
- ▶ Casos de teste mantém uma boa cobertura do código em teste.
- ▶ Simples e intuitiva para a maioria dos programadores.



Análise do Valor Limite  
Introdução  
Técnica de Aplicação

Exemplo de Aplicação  
Aplicabilidade e Limitações

Resumo

Exercício

Leitura Recomendada

# Programa cal do Unix – Especificação

`cal [[month] year]`

*“Um único parâmetro especifica o ano (*year*) a ser exibido e pode variar entre 1 e 9999; observe que o ano deve ser completamente especificado: `cal89` não exibe o calendário do ano 1989 mas sim do ano 89.*

*Dois parâmetros são utilizados para denotar o mês (*month*) e o ano, sendo que o mês pode variar entre 1 e 12). Caso nenhum parâmetro seja fornecido, o mês do ano atual é exibido.*

*O ano se inicia em 1 de Jan.*

*A reforma no calendário Gregoriano (The Gregorian Reformation) ocorreu no dia 3 de setembro de 1752. Até o momento, a maioria dos países reconheceu a reforma realizada (embora poucos ainda não o tenham feito até os anos 90). Com a reforma, dez dias foram eliminados do calendário a partir da data acima exibindo um calendário diferente para o mês e ano em questão.”*

Com base na especificação acima, considerando o critério **Análise do Valor Limite**, defina quais as classes de equivalência válidas e inválidas, identifique os limites e derive casos de testes que satisfaçam o critério.



Análise do Valor Limite  
Introdução  
Técnica de Aplicação

Exemplo de Aplicação  
Aplicabilidade e Limitações

Resumo

Exercício

**Leitura Recomendada**

# Leitura Recomendada

Mais informações sobre esse tema podem ser encontrados em:

- Seção 1, Capítulo 4 do livro de Copeland (2004).

# Referências I

Copeland, L. *A practitioner's guide to software test design*. Artech House Publishers, 2004.

Maldonado, J. C.; Barbosa, E. F.; Vincenzi, A. M. R.; Delamaro, M. E.; Souza, S. R. S.; Jino, M. *Introdução ao teste de software*. Relatório Técnico 65 – Versão 2004-01, Instituto de Ciências Matemáticas e de Computação – ICMC-USP, disponível on-line:  
[http://www.icmc.usp.br/CMS/Arquivos/arquivos\\_enviados/BIBLIOTECA\\_113\\_ND\\_65.pdf](http://www.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_ND_65.pdf)., 2004.