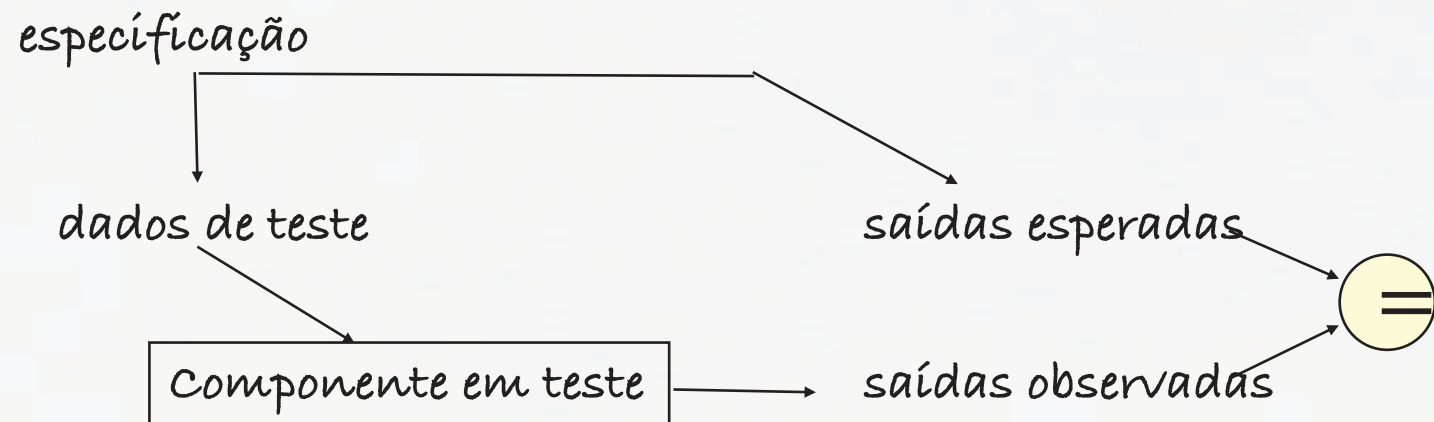


Tópicos

- ☐ Partição de equivalência
- ☐ Testes de valores-límite
- ☐ Grafo causa-efeito

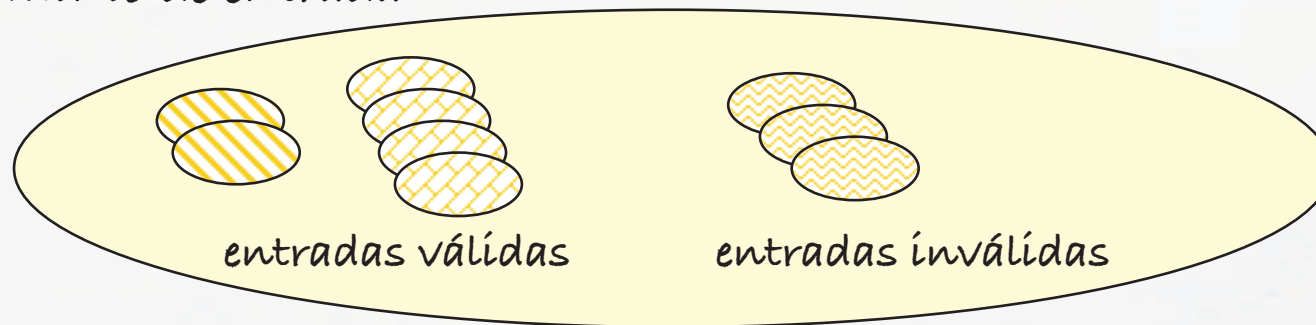
Testes baseados na especificação

- Também chamados de testes caixa preta
- Objetivos dos testes: demonstrar cobertura da especificação
- Modelos de teste obtidos das especificações:
 - requisitos
 - arquitetura



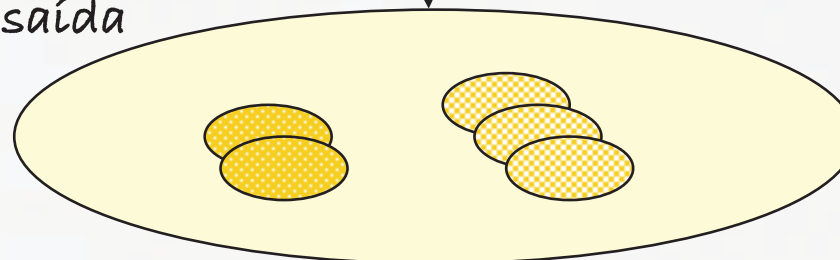
Partições de equivalência: princípio

domínio de entrada



sistema

domínio de saída



Partições de equivalência

- ❑ O domínio de entrada (ou saída) do programa/função é dividido em um número finito de partições (ou classes) de equivalência
 - ❑ supõe-se que dados pertencentes a uma partição revelam as mesmas falhas
 - ❑ partições válidas e inválidas são consideradas
- ❑ Geração de testes: seleccionar um ou mais dados de cada partição
- ❑ Critério de cobertura: cada partição deve ser considerada ao menos 1 vez

Partição de equivalência: passos

- ❑ Decompor o programa em funções
- ❑ Identificar as variáveis que determinam o comportamento de cada função
- ❑ Particionar os valores de cada variável em classes de equivalência (válidas e inválidas)
- ❑ Especificar os casos de teste:
 - ❑ eliminar as classes impossíveis ou os casos desinteressantes
 - ❑ selecionar casos de testes cobrindo as classes válidas das diferentes variáveis
 - ❑ para cada classe inválida escolha um caso de teste que cubra 1 e somente 1 de cada vez

Determinação das classes de equivalência

Definição da variável de entrada	Classes de equivalência
Intervalo	<ul style="list-style-type: none">• Uma classe válida para valores pertencentes ao intervalo• Uma classe inválida para valores menores que o limite inferior• Uma classe inválida para valores maiores que o limite superior
Lista de valores válidos	<ul style="list-style-type: none">• Uma classe válida para os valores incluídos na lista• Uma classe inválida para todos os outros valores

Determinação das classes de equivalência

Definição da variável de entrada	Classes de equivalência
Número de valores válidos	<ul style="list-style-type: none">• Uma classe válida para número de valores igual ao número previsto• Uma classe inválida para número de valores = 0• Uma classe inválida para número de valores maior ou menor que o valor previsto
Restrições (expressão lógica; sintaxe; valor específico; compatibilidade com outras variáveis)	<ul style="list-style-type: none">• Uma classe válida para os valores que satisfazem às restrições• Uma classe inválida para os outros valores

Exemplo 1

- Função:
 - Considere uma função que aceita como entradas de 4 a 6 valores inteiros de 2 dígitos maiores do que 10.
- Identificação das variáveis de entrada e das condições que estas devem satisfazer:
 - $nro_entradas \in [4, 6]$
 - $valor \in [10, 99]$

Exemplo 1

- Determinação das classes de equivalência:

variável	Classes válidas	Classes Inválidas
nro_entradas	C1. $4 \leq \text{nro_entradas} \leq 6$	C3. $\text{nro_entradas} < 4$ C4. $\text{nro_entradas} > 6$
valor	C2. $10 \leq \text{valor} \leq 99$	C5. $\text{valor} < 10$ C6. $\text{valor} > 99$

Exemplo 1

- Casos de teste:
 - seleccionar casos de testes cobrindo as classes válidas das diferentes variáveis

nro_entradas	C1	C2	C3	C4	C5	C6
4 ... 6	X					
< 4						
> 6						
valor						
10 ... 99		X				
< 10						
> 99						

nro_entradas	valores
5	11, 12, 45, 78, 95

Exemplo 1

- Casos de teste:
 - para cada classe inválida escolha um caso de teste que cubra 1 e somente 1 de cada vez

nro_entradas	C1	C2	C3	C4	C5	C6
4 ... 6	X					
< 4			X			
> 6						
valor						
10 ... 99		X				
< 10						
> 99						

	nro_entradas	valores
1.	5	11, 12, 45, 78, 95
2.	3	11, 12, 45

Exemplo 1

- Casos de teste:
 - para cada classe inválida escolha um caso de teste que cubra 1 e somente 1 de cada vez

nro_entradas	C1	C2	C3	C4	C5	C6
4 ... 6	X					
< 4			X			
> 6				X		
valor						
10 ... 99		X				
< 10					X	
> 99						X

	nro_entradas	valores
1.	5	11, 12, 45, 78, 95
2.	2	11, 12
3.	8	11, 12, 45, 78, 95, 67, 77, 54
4.	5	5, 11, 12, 45, 6
5.	5	110, 45, 78, 340, 95

Exemplo 2

- ❑ Considere o teste do procedimento: `valida_Nova_Senha` que recebe como entrada uma senha e valida-a conforme as seguintes regras:
 - ❑ uma senha deve ter de 6 a 10 caracteres
 - ❑ o primeiro caracter dever ser alfabético, numérico ou um "?"
 - ❑ os outros caracteres podem ser quaisquer, desde que não sejam caracteres de controle
 - ❑ a senha não pode existir em um dicionário

Exemplo 2 - Classes de Equivalência

variáveis	válidas	inválidas
tamanho	C1. tamanho $\in [6, 10]$	C7. tamanho < 6
		C8. tamanho > 10
1º caracter (Car1)	C2. Car1 alfabético	C9. Car1 $\in \{\text{caract. controle}\}$
	C3. Car1 numérico	C10. Car1 $\notin \{\text{letras, dígitos, ?}\}$
	C4. Car1 é "?"	
outros caract (Outro)	C5. Outro $\notin \{\text{caract. de controle}\}$	C11. Outro $\in \{\text{caract. de controle}\}$
status	C6. Senha \notin dicionário	C12. Senha \in dicionário

classes de
equivalência

1 2 3 4 5 6 7 8 9

casos de teste

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

Análise de valores-limite

- ❑ Critério de seleção que identifica valores nos limites das classes de equivalência
- ❑ Exemplos:
 - ❑ valor mínimo (máximo) igual ao mínimo (máximo) válido
 - ❑ uma unidade abaixo do mínimo
 - ❑ uma unidade acima do máximo
 - ❑ arquivo vazio
 - ❑ arquivo maior ou igual à capacidade máxima de armazenamento
 - ❑ cálculo que pode levar a "overflow" ("underflow")
 - ❑ erro no primeiro (último) registo

Exemplo 3

□ Casos de teste:

□ para cada classe inválida escolha um caso de teste que cubra 1 e somente 1 de cada vez

nro_entradas	C1	C2	C3	C4	C5	C6
4 ... 6	X					
< 4			X			
> 6				X		
valor						
10 ... 99		X				
< 10					X	
> 99						X

	nro_entradas	valores
1.	4	10, 11, 98, 99
2.	6	10, 11, 50, 55, 98, 99
3.	3	10, 11, 98
4.	7	10, 11, 50, 55, 98, 99, 45
5.	4	9, 10, 11, 99
6.	4	10, 11, 100, 101

Limitação

- ❑ Testes baseados em partições de equivalência ou análise de valores-límite:
 - ❑ consideram cada valor de entrada isoladamente
- ❑ e se existirem combinações de valores que constituam situações interessantes a serem testadas?

Análise causa - efeito

- ❑ Necessária quando se deseja testar combinações de entradas
- ❑ Utiliza tabelas de decisão e árvores de decisão
- ❑ grafo causa-efeito como modelo auxiliar

Definições

Causas:

- condições de entrada (valor lógico)

Efeitos

- ações realizadas em resposta às diferentes condições de entrada

ex.:

causa: $\text{preço} \geq 50 \wedge 0 \leq \text{qtd} \leq 99$

efeito: fornecer 5% de desconto

Árvore de decisão

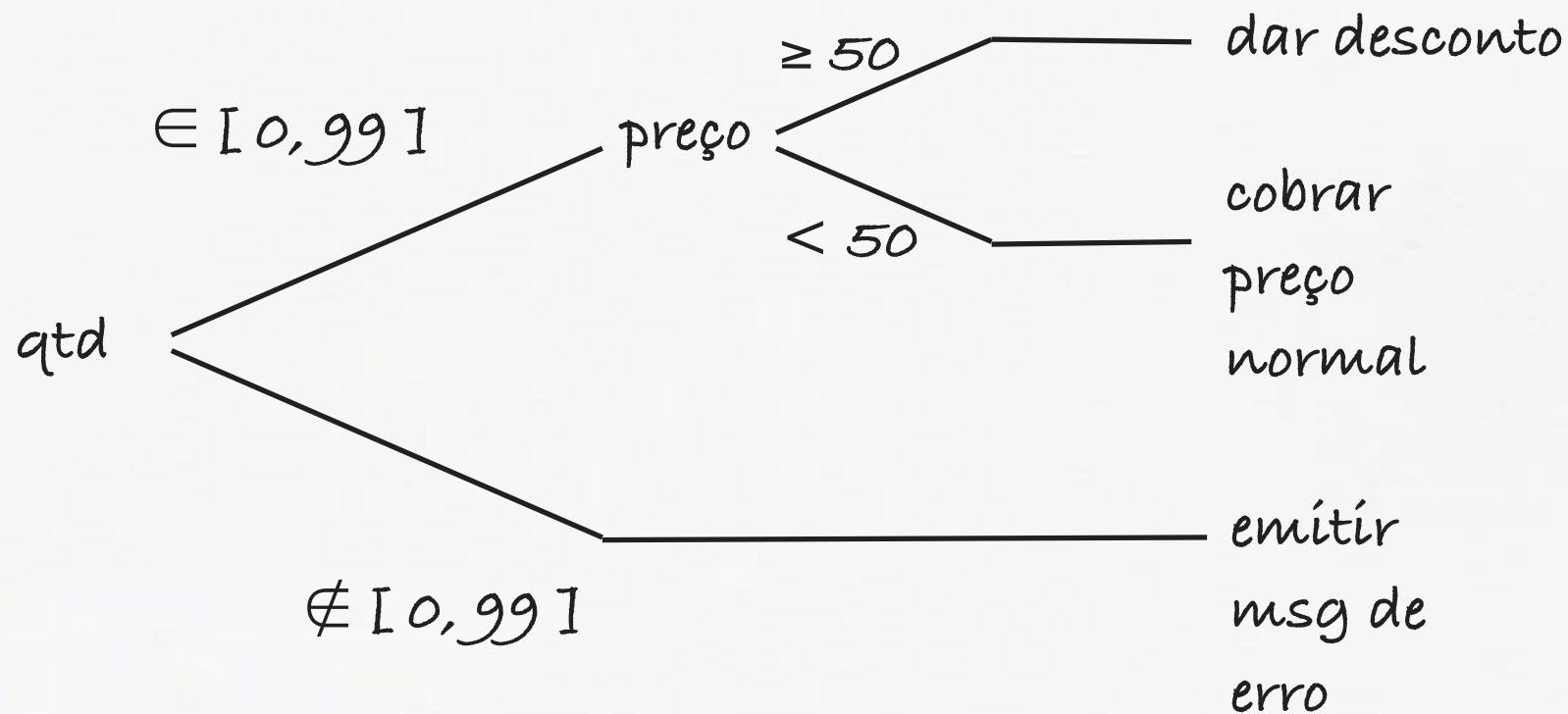


Tabela de decisão

c1	V	...	F													
...																
e1	x															
...																

Condições de entrada
(causas)

Ações
(efeitos)

regra

Construção da tabela de decisão

preço ≥ 50	V	F	X
$0 \leq \text{qtd} \leq 99$	V	V	F
dar desconto	X		
cobrar preço normal		X	
emitir msg de erro			X