04. Modelagem com Funções (Errata)

Princípios de Engenharia de Software (Texto em Elaboração)

Italo S. Vega
italo@pucsp.br

Faculdade de Estudos Interdisciplinares (FACEI)

Pontifícia Universidade Católica de São Paulo

26/03/2022

Sumário

4 Lógica e Código

4.1 Condições Compostas

No texto original, defini logicamente a função f_1 como segue:

$$f_1 \triangleq \lambda x : \mathbb{R} \mid -10 \leq x \leq 10 \bullet x^2 - 3x + 2$$

e produzi um código explicitando as partes constituintes da condição $-10 \leq x \leq 10$:

```
# CENÁRIO de implementação da função lambda "f1"
def f1 (x) : # declaração de parâmetro
  if (-10.0 <= x and x <= 10.0) : # predicado
    return x*x - 3*x + 2 # expressão que calcula a imagem de "x"
```

Como muito bem observado pelo Gustavo Tessitore no encontro presencial, o seguinte código apresenta uma forma ainda mais próxima daquela da lógica (alterada no texto original):

```
# CENÁRIO de implementação da função lambda "f1"
def f1 (x) : # declaração de parâmetro
  if (-10.0 <= x <= 10.0) : # predicado
    return x*x - 3*x + 2 # expressão que calcula a imagem de "x"
```

4.2 Diferença entre Lógica e Código

Modifiquei o caderno original, removendo o erro de lógica no exemplo de especificação e implementação do método de Newton-Raphson. No texto original, criei uma situação para confrontar uma diferença entre o modelo lógico e a sua implementação em Python. A ação passo contém um erro de lógica na condição envolvendo r, n e ϵ:

$$\begin{aligned} \text{início} \triangleq & \wedge n = 2 \\ & \wedge \epsilon = 0.001 \\ & \wedge r = n \end{aligned}$$

$$\begin{aligned} \text{passo} \triangleq & \wedge |r - \frac{n}{r}| \leq r \cdot \epsilon \\ & \wedge r' = \frac{r + \frac{n}{r}}{2} \end{aligned}$$

O código em Python não preservou a lógica original, "corrigindo-a":

```
# CENÁRIO de implementação de Newton-Raphson
# Raiz quadrada de 'n' com precisão 'EPS'
n   = 2.0
EPS = 0.001
r   = n
# o predicado inicial "início" é verdadeiro
while (abs(r - n / r) > r * EPS) :
  r = (n / r + r) / 2.0
  print (abs(r - n / r))
  # a ação "passo" é verdadeira
# a ação "passo" é verdadeira
print (n , EPS , f"{r:.3f}" )
```

Na continuação do texto, Espec alerta sobre a necessidade de se provar o argumento, levando à eventual identificação do erro de lógica. Entretanto, por alguma razão, o ambiente presencial não caminhou para o questionamento do nível lógico, mesmo com a evidência do resultado correto produzido pela execução do código em Python.

Intenção do erro de lógica Como a intenção desse cenário não levou ao efeito desejado de levantar suspeitas a respeito da inconsistência lógica, cabe esta errata. O valor de r deve ser alterado enquanto o erro mínimo, representado por ϵ não tiver sido alcançado. Ou seja, a condição lógica que reflete tal intenção não é aquela apresentada no texto original. Especifica-se a lógica desejada da seguinte forma:

$$\begin{aligned} \text{passo} \triangleq & \wedge |r - \frac{n}{r}| > r \cdot \epsilon \\ & \wedge r' = \frac{r + \frac{n}{r}}{2} \end{aligned}$$

A ação "passo" afirma a atualização de r enquanto a condição $|r - \frac{n}{r}| > r \cdot \epsilon$ for verdadeira, uma vez que tal condição reflete um distanciamento do erro mínimo aceitável. Ainda assim, vale o esforço de se provar o argumento do texto original com o intuito de se identificar esse erro de lógica.