

CDIA-ES-MA3-P2 (v0.1.1)

Engenharia de Software

Professor Dr. *Italo S. Vega* (italo@pucsp.br)

FACEI



Pontifícia Universidade Católica de São Paulo

Junho de 2022

Sumário

Apresentação	2
Contexto Geral	3
1 QUESTÃO. Predicado Inicial	4
2 QUESTÃO. Implementação do Predicado Inicial	5
3 QUESTÃO. Tipos de Dados Abstratos	6
4 QUESTÃO. Implementação de Tipos de Dados Abstratos	7
5 QUESTÃO. Parâmetros de um Modelo	8
6 QUESTÃO. Funções-λ	10
7 QUESTÃO. Implementação de Funções-λ	11
8 QUESTÃO. Ações e Propriedades de Estados	12
9 QUESTÃO. Propriedades de um Código	13
10 QUESTÃO. Funções de Interpretação	14
Apêndice A: Código em Python	15
Referências	17

Apresentação

Nesta atividade serão utilizados os elementos básicos de modelagem lógica e de implementação em um particular contexto. Você deverá publicar as suas respostas no *Teams*, na área indicada pelo professor, até o horário final estabelecido.

A seção **Contexto Geral** estabelece um contexto verdadeiro para todas as questões desta avaliação. As subseções *Contexto* definem situações gradualmente verdadeiras ao longo do texto.

Pontuação Respostas assinaladas com “Não sei” receberão 4 pontos. Caso erre a resposta, a pontuação será zero. Caso acerte a resposta, a pontuação será 10. O total de pontos obtidos nesta avaliação será linearmente normalizado para a escala entre 0 e 10. Faz parte da avaliação a correta interpretação das questões.

Errata

1. Questão 3: faltou explicitar o enunciado.
2. Questão 4:
 - a. faltou explicitar o enunciado;
 - b. ao invés de && deve ser and na alternativa 3.
3. Questão 7, na alternativa (I):
 - a. ao invés de && deve ser and;
 - b. ao invés de Grau deve ser m_Grau.

Contexto Geral

No final do seu período de estudos, Fubã decide desenvolver um programa de computador para calcular a sua situação em uma disciplina. Ele chama a aplicação de “SIGA”. O modelo computacional apoia-se em uma lógica refinada, que ele desenvolve gradualmente.

Depois de algum tempo de análise, Fubã cria cinco variáveis cujos valores são alterados ao longo do cálculo da sua situação. A variável g_{MF} representa o seu grau no final do período de avaliação. O texto que indica se foi ou não aprovado corresponde ao valor *info*. A validade do resultado encontra-se no valor *valido*. Os graus, informados pelo Profe, são representados pelas variáveis p_1 e p_2 :

VARIABLES p_1 : Grau
 p_2 : Grau
 valido : Boolean
 g_{MF} : Grau
 info : String



— Traduzirei essas variáveis de estado, respectivamente, da seguinte maneira em Python:

```
# Variáveis de estado
p1 = -1
p2 = -1
valido = False
g_MF = -1
info = '?'
```

Uma das importantes ações do modelo do Fubã refere-se àquela que obtém os graus obtidos por um estudante ao longo do período letivo. Ele chama a ação de *obterGraus* e encontra-se definida na [Questão 5](#). Outra função é aquela que modela o cálculo do grau final de um estudante, mf , definida na [Questão 6](#).

Ao longo do desenvolvimento deste programa, Fubã usou o interpretador Python:

```
# TERMINAL
python3 --version
Python 3.8.10
```

Ele também empregou uma convenção para implementar a sua lógica:

- nomes de funções e de predicados de estado são prefixados com “m_”;
- nomes de ações são prefixados com “op_”.

1 QUESTÃO. Predicado Inicial

Contexto No começo do seu projeto, Fubã admite o seguinte predicado inicial da aplicação “SIGA” como verdadeiro:

$$\begin{aligned} \text{ siga}_{\text{inicio}} &\triangleq p_1 = \perp \\ &\quad p_2 = \perp \\ &\quad \text{valido} = \text{False} \\ &\quad g_{MF} = \perp \\ &\quad \text{info} = "?" \end{aligned}$$



— Tenho dúvidas a respeito das seguintes afirmações:

- I) No modelo, é verdade que o tipo de \perp é Grau.
- II) Se $\text{ siga}_{\text{inicio}}$ é verdadeiro, então o estado $[p_1 = \perp, p_2 = \perp, g_{MF} = \perp, \text{info} = "?"]$ é sempre verdadeiro.
- III) Se o estado $[p_1 = \perp, p_2 = \perp, g_{MF} = \perp, \text{info} = "?", \text{valido} = \text{False}]$ é verdadeiro, então o predicado inicial $\text{ siga}_{\text{inicio}}$ é sempre verdadeiro.

Enunciado Assinale (no *Teams*) a alternativa contendo apenas afirmações verdadeiras:

- 1. I e II.
- 2. I e III.
- 3. II e III.
- 4. I, II e III.
- 5. “Não sei”.

2 QUESTÃO. Implementação do Predicado Inicial

Contexto Fubã elabora o seguinte predicado inicial para o seu modelo da aplicação SIGA:

$$\begin{aligned} \text{sig}_{\text{inicio}} &\triangleq \wedge p_1 = \perp \wedge p_2 = \perp \wedge \text{valido} = \text{False} \\ &\quad \wedge g_{MF} = \perp \\ &\quad \wedge \text{info} = "?" \end{aligned}$$



— Usarei atribuições de Python (Rossum, 2003) para tornar verdadeiro o predicado $\text{sig}_{\text{inicio}}$:

```
# Predicado inicial
def m_siga_inicio () :
    p1 = -1; p2 = -1; valido = False
    g_MF = -1
    info = '?'
```

Quando pensa nos efeitos da execução do método `m_siga_inicio`, ele escreve as seguintes afirmações:

- I) $(p1 == -1) \text{ and } (p2 == -1) \text{ and } (g_MF == -1)$
- II) $(info == '?') \text{ and } '?' == info$
- III) $(p1 == -1) | (p2 == -1) \& \text{valido}$

Enunciado Assinale (no *Teams*) a alternativa contendo uma afirmação verdadeira:

1. (I) é uma proposição inconsistente.
2. A condição (III) produz False, mas é válida.
3. A avaliação da expressão (II) produz True, embora seja inválida.
4. A disjunção lógica das afirmações (II) e (III) produz o valor False.
5. “Não sei”.

3 QUESTÃO. Tipos de Dados Abstratos

Contexto Em seguida, Fubã desenha uma interface com o ambiente (ICA), como aquela mostrada na Fig. 1. O estudante informa os seus graus (x_1, x_2) , como indicado pela seta (A).

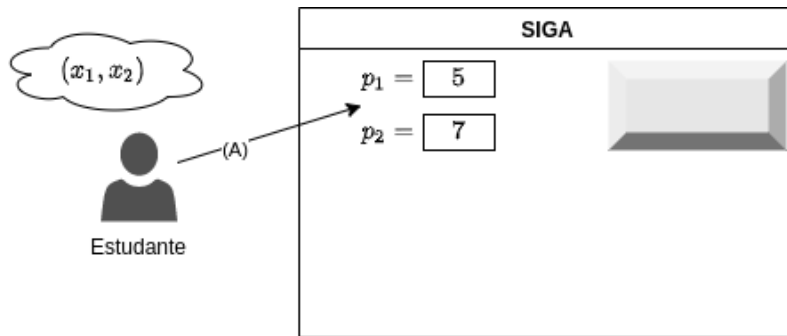


Figura 1: ICA com os parâmetros do SIGA.



— A aplicação SIGA não deverá aceitar valores arbitrários para p_1 e p_2 .

Ele captura essa ideia no tipo de dado abstrato Grau. No seu modelo, a condição (ou proposição) para um valor Float (considerado um tipo primitivo na sua lógica) ser do tipo Grau é que esteja no intervalo entre zero e dez:

$$\text{Grau} \triangleq [g : \text{Float} \mid 0 \leq g \leq 10]$$

Desse modo, interpreta-se “8 : Grau” como: “o valor 8 é do tipo Grau”. Em particular, Fubã imagina que o tipo Float da sua lógica corresponde ao tipo primitivo float da linguagem Python.

Enunciado Assinale (no *Teams*) a alternativa contendo uma afirmação verdadeira:

1. Qualquer que sejam os valores x, y do tipo Grau, x, y também serão do tipo Float portanto, é válido avaliar a expressão $x + y$.
2. Como se deve considerar válida a interpretação $\llbracket 7 : \text{Float} \rrbracket_{\text{Python}} = 7 : \text{float}$, conclui-se que $7 : \text{Grau}$ na Fig. 1.
3. Se x é um valor do tipo Float, então ele pode representar um valor do tipo Grau.
4. Conforme o modelo do Fubã, o valor p_1 é $5 : \text{Grau}$ na Fig. 1.
5. “Não sei”.

4 QUESTÃO. Implementação de Tipos de Dados Abstratos

Contexto Fubã criou o tipo de dado abstrato Grau

$$\text{Grau} \triangleq [g : \text{Float} \mid 0 \leq g \leq 10]$$



— Implementarei esse tipo de dado abstrato usando o método Python:

```
# Predicado de estado
def m_Grau(x) :
    return not (x < 0 or x > 10)
```

Fubã também sabe que valores do tipo `int` são automaticamente convertidos para `float` quando se fizer necessário pelo ambiente de execução Python.

Enunciado Assinale (no *Teams*) a alternativa contendo uma afirmação verdadeira:

1. Se um estudante considera que `x:float` é verdadeiro, conclui-se que `m_Grau(x)` é verdadeiro.
2. Independentemente do valor x do tipo Grau, a avaliação `m_Grau(x)` é `True`, considerando-se que `x:float` e que $\llbracket x \rrbracket_{\text{Python}} = x$.
3. Como `5+3 == 8`, então a condição “`m_Grau(5) and m_Grau(3)`” é verdadeira e 8 representa um valor do tipo Grau na lógica do Fubã.
4. Como `5:int`, então a condição “`m_Grau(5) == False`” é verdadeira pois o método `m_Grau` está definido apenas para valores do tipo `float`.
5. “Não sei”.

5 QUESTÃO. Parâmetros de um Modelo

Contexto Em seguida, Fubã desenha uma interface com o ambiente (Fig. 2) para ajudá-lo na definição dos parâmetros do modelo. Parâmetros são constantes definidas externamente pelo ambiente de uso da aplicação (inspirado em Hedman, 2006, p. 62). Na Figura, primeiro o estudante informa os seus graus (x_1, x_2) , indicado pela seta (A). Em seguida, ele clica no botão, seta (B). Finalmente, ele observa o efeito de informar os seus graus para a aplicação, conforme sugerido pela seta (C).

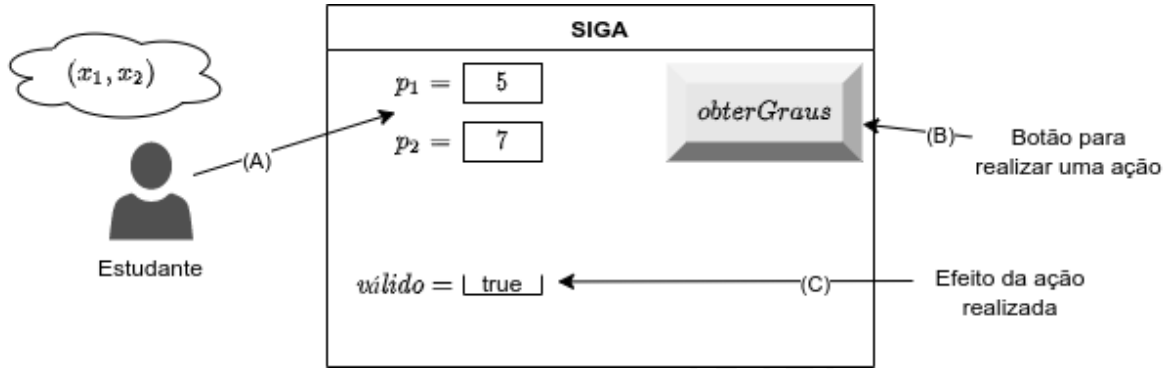


Figura 2: ICA com os parâmetros do SIGA.



— Usarei uma ação para especificar o efeito de clicar no botão:

$$\begin{aligned} obterGraus_1[x_1, x_2] &\triangleq \wedge info' = "?" \\ &\quad \wedge x_1 \in Grau \wedge x_2 \in Grau \\ &\quad \wedge p'_1 = x_1 \wedge p'_2 = x_2 \wedge valido' = True \end{aligned}$$

$$\begin{aligned} obterGraus_2[x_1, x_2] &\triangleq \wedge info' = "?" \\ &\quad \wedge (x_1 \notin Grau \vee x_2 \notin Grau) \\ &\quad \wedge p'_1 = \perp \wedge p'_2 = \perp \wedge valido' = False \end{aligned}$$

$$obterGraus[x_1, x_2] \triangleq obterGraus_1[x_1, x_2] \vee obterGraus_2[x_1, x_2]$$

Fubã implementa esta ação no método `op_obterGraus` apresentado no Anexo.

Enunciado Assinale (no *Teams*) a alternativa contendo uma afirmação verdadeira:

1. *obterGraus* é uma ação, pois representa o botão da interface da aplicação.
2. Partindo-se do estado $[p_1 = \perp, p_2 = \perp]$, alcança-se o estado $[p_1 = 5, p_2 = 7]$ se a ação *obterGraus* for verdadeira de acordo com a Fig. 2.
3. O comportamento que segue corresponde a três cliques no botão *obterGraus* e representa uma situação em que o estudante informa os mesmos graus:

$$\begin{array}{l}
 [p_1 = \perp, p_2 = \perp, \text{valido} = \text{False}] \\
 \xrightarrow{\text{obterGraus}[5,7]} [p_1 = 5, p_2 = 7, \text{valido} = \text{True}] \\
 \xrightarrow{\text{obterGraus}[5,7]} [p_1 = 5, p_2 = 7, \text{valido} = \text{True}] \\
 \xrightarrow{\text{obterGraus}[5,7]} [p_1 = 5, p_2 = 7, \text{valido} = \text{False}]
 \end{array}$$

4. Se x_1 ou x_2 não forem do tipo Grau, então o modelo representará uma situação em que a aplicação estará em um estado inválido.
5. “Não sei”.

6 QUESTÃO. Funções- λ

Contexto A regra mf calcula a média aritmética final (um valor do tipo Grau) considerando os graus p_1 e p_2 obtidos nas provas realizadas pelo estudante. Fubã se lembra que o tipo Grau representa qualquer valor entre 0 e 10, inclusive:

$$mf \triangleq \lambda p_1, p_2 : \text{Grau} \bullet \frac{p_1 + p_2}{2}$$

Ele considera as seguintes afirmações:

- I) A função mf é total e produz um absurdo quando aplicada no argumento $(100, 100)$.
- II) O tipo do domínio da função mf é $(\text{Grau} \times \text{Grau})$ e o codomínio do tipo Grau.
- III) No estado $[p_1 = 5, p_2 = 7]$ é verdade que $mf(p_1, p_2) = 6$.

Enunciado Assinale (no *Teams*) a alternativa contendo apenas afirmações verdadeiras:

- 1. I e II.
- 2. I e III.
- 3. II e III.
- 4. I, II e III.
- 5. “Não sei”.

7 QUESTÃO. Implementação de Funções- λ

Contexto Para calcular o grau da média final de um estudante, Fubã define a seguinte função:

$$mf \triangleq \lambda p_1, p_2 : \text{Grau} \bullet \frac{p_1 + p_2}{2}$$



— Consigo implementar essa função no seguinte método Python:

```
# Função de estado
def m_mf(p1, p2) :
    return (p1+p2) / 2
```

Fubã escreve as seguintes afirmações:

- I) Se ($m_Grau(x1)$ and $m_Grau(x2) == True$), então o método m_mf implementa a função mf para qualquer valor de $x1$ e $x2$ passados como argumentos de chamada.
- II) O método m_mf retorna o valor $100:float$ ao ser chamado com o argumento $(300, 200) : (float, float)$.
- III) A chamada do método m_mf com argumentos 10 e 10 é válida conforme a função mf .

Enunciado Assinale (no *Teams*) a alternativa contendo apenas afirmações verdadeiras:

- 1. I e II.
- 2. I e III.
- 3. II e III.
- 4. I, II e III.
- 5. “Não sei”.

8 QUESTÃO. Ações e Propriedades de Estados

Contexto Fubã cria a ação que especifica a alteração de estado depois que um estudante solicita o cálculo do grau da sua média final:

$$\begin{aligned} mediaFinal &\triangleq \wedge \text{valido} \\ &\wedge g'_{MF} = mf(p_1, p_2) \end{aligned}$$



— Tenho dúvidas sobre as afirmações que seguem de acordo com o meu modelo:

- I) A ação *mediaFinal* é verdadeira sempre que a ação *obterGraus* (definida na [Questão 5](#)) for verdadeira.
- II) $[\text{valido}, g_{MF} = \perp] \rightarrow [\text{valido}, g_{MF} = mf(p_1, p_2)]$ é um passo computacional.
- III) Se *mediaFinal* torna verdadeiro o estado q_i e, nesse estado vale a condição $g_{MF} = k$, então o valor da variável de estado g_{MF} será k em todos os estados seguintes.

Enunciado Assinale (no *Teams*) a alternativa contendo apenas afirmações verdadeiras:

- 1. I e II.
- 2. I e III.
- 3. II e III.
- 4. I, II e III.
- 5. “Não sei”.

9 QUESTÃO. Propriedades de um Código

Contexto Fubã define a ação que estabelece o grau da média final de um estudante, depois que ele informa os seus graus no período letivo (conforme a ação *obterGraus*, definida na [Questão 5](#)):

$$\begin{aligned} grau_mf &\triangleq \wedge \text{valido} \\ &\wedge g'_{MF} = mf(p_1, p_2) \end{aligned}$$



— Essa ação corresponde ao seguinte código em Python:

```
# Ação
def op_grau_mf() :
    global p1, p2, valido, g_MF
    if (valido) :
        g_MF = m_mf (p1, p2)
```

Considerando as propriedades lógicas do código, afirma-se que:

- I) O método `op_grau_mf` recorre a uma função total.
- II) A chamada da operação `op_grau_mf` no estado $[p_1 = 10, p_2 = 10, \text{valido}]$ altera o valor da variável de estado `valido` para `True`.
- III) Se a aplicação estiver no estado $[p_1 = 2, \text{valido}]$, então o valor associado à variável g_{MF} será alterado e do tipo `Grau`, como efeito da execução do método `op_grau_mf`.

Enunciado Assinale (no *Teams*) a alternativa contendo apenas afirmações verdadeiras:

- 1. I e II.
- 2. I e III.
- 3. II e III.
- 4. I, II e III.
- 5. “Não sei”.

10 QUESTÃO. Funções de Interpretação

Contexto Finalmente, Fubã preocupa-se com a função que calcula a situação de um estudante no final do período letivo:

$$\textit{situacao} \triangleq \lambda g : \text{Grau} \bullet \text{if } g < 5 \\ \text{then } \textit{REPROVADO} \\ \text{else } \textit{APROVADO}$$


— Preciso de ajuda para decidir se as afirmações que seguem são ou não verdadeiras:

- I) A função *situacao* é parcial, pois nem todos os valores do tipo *Grau* possuem imagem de acordo a sua definição.
- II) A composição (*situacao* \circ *mf*) produz um valor no conjunto $\{\textit{APROVADO}, \textit{REPROVADO}\}$.
- III) A aplicação de *situacao* no valor g_{MF} , conforme o estado:

$$[p_1 = 5, p_2 = 7, \textit{valido} = \text{True}, g_{MF} = 6]$$

produz o valor *APROVADO*.

Enunciado Assinale (no *Teams*) a alternativa contendo apenas afirmações verdadeiras:

- 1. I e II.
- 2. I e III.
- 3. II e III.
- 4. I, II e III.
- 5. “Não sei”.

Apêndice A: Código em Python

Este é o código-fonte completo do ensaio realizado pelo Fubã, em Python:

```
# siga.python
# Variáveis de estado
p1 = 0
p2 = 0
valido = False
g_MF = 0
info = ""
# Predicado inicial
def siga_inicio() :
    global p1, p2, valido, g_MF, info
    p1 = -1;
    p2 = -1;
    valido = False;
    g_MF = -1;
    info = "?";
# Predicado de estado
def m_Grau (x) :
    return not (x < 0 or 10 < x)
# Ação
def op_obterGraus() :
    global p1, p2, valido
    x1 = 5 # float(input("P1: "))
    x2 = 7 # float(input("P2: "))
    if m_Grau(x1) and m_Grau(x2) :
        p1 = x1
        p2 = x2
        valido = True
    else :
        valido = False
# Função de estado
def m_mf(p1, p2) :
    return (p1+p2) / 2
# Ação
def op_grau_mf() :
    global p1, p2, valido, g_MF
    if (valido) :
        g_MF = m_mf (p1, p2)
# Função de estado
def m_situacao(g_MF) :
    if (g_MF < 5) :
        return "REPROVADO"
    if (g_MF >= 5) :
        return "APROVADO"
# Ação
```

```
def op_situacao() :
    global valido, g_MF, info
    if (valido) :
        info = m_situacao (g_MF)
        print(f'Situacao final= {info} ({g_MF})')
# Controle principal
siga_inicio()
op_obterGraus()
op_grau_mf()
op_situacao()
```


Referências

- Hedman, S. (2006). *A First Course in Logic: An introduction to model theory, proof theory, computability, and complexity* (1º ed). Great Britain: Oxford University Press.
- Rossum, G. V. (2003). *The Python Language Reference Manual*. (F. L. Jr. Drake, Org.). Network Theory.