CDIA-ES-MA3-P1 (v0.1.0)

Engenharia de Software

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

FACEI

Pontifícia Universidade Católica de São Paulo

Maio de 2022

Sumário

Apresentação

Nesta atividade serão aplicados os elementos notacionais 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.

As subseções *Contexto* definem situações verdadeiras no contexto de cada questão, exceto quando explicitamente indicada a sua validade em outras questões.

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.

1 QUESTÃO. Análise do Problema

Contexto Um determinado carro fabricado pela XYZ—1—foi representado por um modelo com destaque para os lugares e pressões de cada um dos seus quatro pneus. No modelo, representaram-se os lugares de cada pneu e a pressão de cada um deles. Os pneus ocupam os lugares, dianteiro esquerdo (DE) e direito (DD), bem como traseiro esquerdo (TE) e direito (TD). Além disso, as pressões indicadas foram medidas em libras. Determinados fabricantes de automóveis recomendam que as pressões dos pneus dianteiros sejam iguais, bem como as dos traseiros, embora não precisem ser as mesmas.

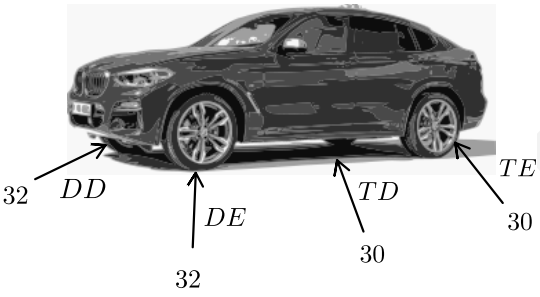

Figura 1: Um carro e elementos do seu modelo.

Apesar de uma parte significativa das marcas proporem calibragens idênticas para os pneus dianteiros e traseiros, não é só a fabricante de veículos XYZ que adota pressões diferentes. A ABC recomenda 32 libras na frente e 30 libras atrás. Disponível em: https://quatrorodas.abril.com.br/auto-servico/e-correto-usar-pressoes-diferentes-nos-pneus-da-frente-e-de-tras. Acesso em: 29/04/2022 (adaptado).

Com base no enunciado que caracteriza o domínio da aplicação, avalie as afirmações a seguir:

I) Em geral, os quatro pneus de um carro devem ter a mesma pressão.
II) Os pneus dianteiros podem ter pressão diferente dos traseiros.
III) Os pneus dos carros da ABC e da XYZ são calibrados com pressões diferentes.

Enunciado É correto apenas o que se afirma em

1. I e II.
2. I e III.
3. II e III.
4. I, II e III.
5. "Não sei".

2 QUESTÃO. Sequência como Elemento de Solução

Contexto Como tarefa de ingresso, o estagiário Fubã foi contratado pela fábrica XYZ e encarregado de desenvolver um código que capacite o computador a responder se um carro está ou não com os pneus calibrados. Para isso, ele elaborou um modelo baseado em sequências.

 — O Profe tinha exemplificado duas sequências, s_1 e s_2:

$$s_1 \triangleq \langle A, B, C \rangle$$
$$s_2 \triangleq \langle 2, 4, 6, 8, 10 \rangle$$

As sequências foram utilizadas para apresentar as operações:

- acesso posicional,
- cardinalidade ($\#$),
- inversa ($\sim$),
- **head**, **tail** e
- concatenação ($\frown$).

Com base nas sequências s_1 e s_2, avalie as afirmações a seguir

I) $s_1(2) = B \wedge s_2(\#s_2) = 10$
II) $s_1^{\sim}(B) = s_2(1)$
III) $\mathbf{tail}(s_1) \frown \langle \mathbf{head}(s_2) \rangle = \langle B, C, 2 \rangle$

Enunciado É correto apenas o que se afirma em

1. I e II.
2. I e III.
3. II e III.
4. I, II e III.
5. "Não sei".

3 QUESTÃO. *Array* como Elemento de Implementação

Contexto Fubã aprendeu que valores do tipo *array* podem ser utilizados para se *implementar* sequências em Python. Então, ele codificou as sequências

$$s_1 \triangleq \langle A, B, C \rangle$$
$$s_2 \triangleq \langle 2, 4, 6, 8, 10 \rangle$$

 — Posso implementar essas sequências usando *arrays* de Python:

```
# Sequências como "arrays" em Python
s1 = ['A', 'B', 'C']
s2 = [2, 4, 6, 8, 10]
```

Do ponto de vista do Fubã, captura-se o esquema de tradução das variáveis lógicas de estado em variáveis Python da seguinte maneira:

$$[\![s_1]\!]_{Python} = \texttt{s1}$$
$$[\![s_2]\!]_{Python} = \texttt{s2}$$

Com base nas variáveis de código, identificadas por `s1` e `s2`, avalie as afirmações a seguir no contexto do interpretador Python:

I) `s1[2] == 'B'`
II) `s2[0] + s2[3] == s2[4]`
III) `len(s1)` representa o valor de $\#s_1$

Enunciado É correto apenas o que se afirma em

1. I e II.
2. I e III.
3. II e III.
4. I, II e III.
5. "Não sei".

4 QUESTÃO. Desenho da Solução com Sequências

Contexto Para modelar os pneus de um carro no contexto do domínio da aplicação, o estagiário Fubã apoiou-se em sequências. Ele definiu uma sequência para representar o lugar dos pneus.

 — A função *lugar* modela os lugares dos pneus:

$$lugar \triangleq \langle DE, DD, TE, TD \rangle$$

Em outra sequência, também com quatro elementos, ele representou os valores das pressões dos pneus do carro.

 — Cada posição i nesta sequência informa a pressão do pneu que se encontra no $lugar(i)$.

Assim, Fubã representou os valores das pressões dos pneus no caso do carro c_1 apresentado no domínio da aplicação pela sequência:

$$c_1 \triangleq \langle 32, 32, 30, 30 \rangle$$

De acordo com as duas sequências propostas pelo Fubã, *lugar* e c_1, avalie as afirmações a seguir:

I) $lugar(2) = DE$
II) $lugar^{-1}(TE) = 2$
III) $c_1(lugar^{-1}(DE)) = 32$
IV) $c_1(lugar^{-1}(TD)) = c_1(lugar^{-1}(TE))$

Enunciado É correto apenas o que se afirma em

1. I e III.
2. II e III.
3. III e IV.
4. I, II e IV.
5. "Não sei".

5 QUESTÃO. Interpretação de Código

Contexto Fubã usou as sequências $lugar$ e c_1 para modelar os lugares e pressões dos pneus de um carro (Questão 4):

$$lugar \triangleq \langle DE, DD, TE, TD \rangle$$
$$c_1 \triangleq \langle 32, 32, 30, 30 \rangle$$

 — Com base nestas sequências, consigo produzir o seguinte código em Python, inclusive implementando os carros `v_c2`, `v_c3` e `v_c4`.

A intenção é que a variável lógica c_1 seja implementada pela variável de código `v_c1`:

```
# Código do modelo dos lugares dos pneus nos carros
lugar = ['DE', 'DD', 'TE', 'TD']
# Código do modelo das pressões dos pneus nos carros
v_c1 = [32, 32, 30, 30]
# Outros carros
v_c2 = [30, 30, 32, 32]
v_c3 = [30, 29, 28, 28]
v_c4 = [(0.1+0.1+0.1)*100, 30.0, 30, 30]
```

Neste código, `v_c1` representa o carro c_1 citado na apresentação do domínio da aplicação; `v_c2`, `v_c3` e `v_c4` são três outros carros informados pelo chefe do Fubã.

Enunciado Considerando este código em Python, é correto apenas o que se afirma em:

1. O valor armazenado na variável `v_c1[3]` representa o valor $c_1(lugar^{-1}(TE)) = 30$, considerando-se as sequências $lugar$ e c_1.
2. As pressões dos pneus dianteiros do carro representado pela variável `v_c2` são iguais as dos pneus traseiros do carro `v_c3` mais `4`.
3. Se $lugar^{-1}(TD) = 4$ então a pressão do pneu traseiro direito de `v_c2` é a interpretação de `32` no código.
4. A avaliação da expressão `(v_c4[0] == v_c3[1])` produz o valor `true`.
5. "Não sei".

6 QUESTÃO. Elementos de Solução: Inversa e Composição

Contexto Encapsular decisões referentes ao modelo de implementação foi uma das recomendações que Fubã recebeu do seu professor. A sequência *lugar* modela os lugares dos pneus:

$$lugar \triangleq \langle DE, DD, TE, TD \rangle$$
$$lugar^{\sim} = \{DE \mapsto 1, DD \mapsto 2, TE \mapsto 3, TD \mapsto 4\}$$

 — Usarei a variável de código `v_lugar` para implementar a sequência *lugar*:

```
v_lugar = ['DE', 'DD', 'TE', 'TD']
```

 — Para melhorar a leitura do código, criarei dois métodos para implementar *lugar* e $lugar^{\sim}$, respectivamente:

```
# <<função de estado>>
def m_lugar(k) :
  return v_lugar [k - 1]
# <<função de estado>>
def m_lugar_inv(p) :
  return v_lugar.index(p) + 1
```

Com base nestes métodos, avalie as afirmações a seguir a respeito das suas chamadas:

I) $[\![lugar(1)]\!]_{Python}$ = `m_lugar(1)` e retorna `'DE'` em Python.
II) $[\![lugar^{\sim}(TE)]\!]_{Python}$ = `m_lugar_inv('TE')` e retorna `3` em Python.
III) $[\![(lugar^{\sim} \circ lugar \circ lugar^{\sim})(DD)]\!]_{Python}$ = `m_lugar_inv(m_lugar(m_lugar_inv('DD')))` e retorna `2` em Python.

Enunciado É correto apenas o que se afirma em

1. I e II.
2. I e III.
3. II e III.
4. I, II e III.
5. "Não sei".

7 QUESTÃO. Desenho da Solução: Composição de Funções

Contexto As duas sequências definidas por Fubã na Questão 4 foram:

$$lugar \triangleq \langle DE, DD, TE, TD \rangle$$
$$c_1 \triangleq \langle 32, 32, 30, 30 \rangle$$

Tais sequências modelam os lugares e as pressões dos pneus de um carro (Questão 1). Fubã admite que os valores da sequência $lugar$ são do tipo `Direção`, ou seja, DE : `Direção`, por exemplo. Os valores de pressão encontram-se no intervalo fechado entre 20 e 100 e são do tipo `Pressão`—correspondem a valores `double` de Python.

 — Como precisarei conhecer a pressão de cada um dos pneus de um carro, definirei a função $pressaoDoPneu$.

Para obter a pressão de um dos pneus x de um particular carro c, ele projetou a seguinte função, resultado da composição de c com $lugar^{-1}$:

$$pressaoDoPneu \triangleq \{\ x : \texttt{Direção}; c : \textbf{seq}\ \texttt{Pressão} \mid \#c = 4 \bullet (x, c) \mapsto (c \circ lugar^{-1})(x))\ \}$$

Com base nesta definição, avalie as afirmações a seguir:

I) É verdade que $pressaoDoPneu(DE, c_1) = 32$.
II) Na aplicação $pressaoDoPneu(DE, c_1)$, o valor de $lugar^{-1}(DE)$ é 1.
III) A imagem da aplicação de $pressaoDoPneu$ em (MM, c_1) é indefinida.

Enunciado É correto apenas o que se afirma em

1. I e II.
2. I e III.
3. II e III.
4. I, II e III.
5. "Não sei".

8 QUESTÃO. Implementação da Solução: Composição de Funções

Contexto Procurando aplicar o princípio da encapsulação de decisões de modelagem, Fubã projetou um método para obter a pressão de um pneu que se encontra em um lugar conhecido do carro :

$$pressaoDoPneu \triangleq \{\ x : \mathsf{Direção}; c : \mathbf{seq}\ \mathsf{Pressão} \mid \#c = 4 \bullet (x, c) \mapsto (c \circ lugar^{-1})(x))\ \}$$

Ele decidiu implementar a função *pressaoDoPneu* (reproduzida da Questão 7) por meio do seguinte método em Python:

```
# Valor da pressão de um pneu de carro
# <<função de estado>>
def m_pressaoDoPneu(x, c) :
  return c[m_lugar_inv(x) - 1]

# Código do modelo dos lugares dos pneus nos carros
lugar = ['DE', 'DD', 'TE', 'TD']
# Código do modelo das pressões dos pneus nos carros
v_c1 = [32, 32, 30, 30]
# Outros carros
v_c2 = [30, 30, 32, 32]
v_c3 = [30, 29, 28, 28]
v_c4 = [(0.1+0.1+0.1)*100, 30.0, 30, 30]
```

Enunciado Considerando esse código, é correto apenas o que se afirma em:

1. A execução da chamada `m_pressaoDoPneu('DE', v_c1)` retorna `32`.
2. Em uma chamada do método `m_pressaoDoPneu`, o valor de `len(c)` é sempre 4.
3. Um programa contendo a chamada `m_pressaoDoPneu('MM', v_c1)` é inválido.
4. No comando `return`, o valor do índice para acessar a variável do *array* `c` deveria ser apenas `m_lugar_inv(x)`.
5. "Não sei".

9 QUESTÃO. Regras de Negócio

Contexto Fubã recorda que, conforme o domínio da aplicação, um carro do fabricante XYZ estará com os pneus calibrados caso as pressões dos pneus dianteiros sejam iguais e a dos traseiros também. Ele decide formalizar esta regra definindo a função de estado *calibrado*:

$$
\begin{aligned}
calibrado \triangleq \{\ & c : \textbf{seq}\ \texttt{Double} \\
& \mid \quad \#c = 4 \quad \bullet \quad c \mapsto \\
& \quad \wedge\, pressaoDoPneu(DE, c) = pressaoDoPneu(DD, c) \\
& \quad \wedge\, pressaoDoPneu(TE, c) = pressaoDoPneu(TD, c)\ \}
\end{aligned}
$$

Enunciado Assinale a opção que completa a seguinte afirmação corretamente:

> "A função *calibrado* é o conjunto formado por pares contendo uma sequência de pressões e um valor …"

1. calculado aritmeticamente.
2. que o computador irá calcular.
3. que depende do fabricante de carros.
4. indicando se o carro está ou não com os pneus calibrados.
5. "Não sei".

10 QUESTÃO. Implementação de Regras de Negócio

Contexto O chefe do Fubã parece satisfeito com o modelo da regra de negócio capturada pela função *calibrado*:

$$
\begin{aligned}
calibrado \triangleq \{\ & c : \textbf{seq}\ \texttt{Double} \\
& \mid \quad \#c = 4 \quad \bullet \quad c \mapsto \\
& \quad \wedge\, pressaoDoPneu(DE, c) = pressaoDoPneu(DD, c) \\
& \quad \wedge\, pressaoDoPneu(TE, c) = pressaoDoPneu(TD, c)\ \}
\end{aligned}
$$

 — Produzi o seguinte método para implementar *calibrado*:

```
# Situação de calibragem dos pneus
# <<função de estado>>
def m_calibrado (c) :
  return \
    m_pressaoDoPneu('DE', c) == m_pressaoDoPneu('DD', c) and \
    m_pressaoDoPneu('TE', c) == m_pressaoDoPneu('TD', c)
```

Desta forma, ele consegue calcular a situação de calibragem dos pneus de vários carros. Ele repete o cenário de alguns carros:

```
# Código do modelo dos lugares dos pneus nos carros
lugar = ['DE', 'DD', 'TE', 'TD']
# Código do modelo das pressões dos pneus nos carros
v_c1 = [32, 32, 30, 30]
# Outros carros
v_c2 = [30, 30, 32, 32]
v_c3 = [30, 29, 28, 28]
v_c4 = [(0.1+0.1+0.1)*100, 30.0, 30, 30]
```

Com base no método `m_calibrado` e nos carros `v_c1`, `v_c2`, `v_c3` e `v_c4`, avalie as afirmações em Python a seguir:

I) `m_calibrado(v_c1) == true`
II) `m_calibrado(v_c2) == true`

III) `m_calibrado(v_c3) == true`
IV) `m_calibrado(v_c4) == true`

Enunciado É correto apenas o que se afirma em:

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