

Fundamentos Matemáticos para Computação - COM150 - Turma 003

Página inicial

Avisos

Cronograma

Atividades

Fóruns

Collaborate

Calendário Lives

Notas

Menu das Semanas

Semana 1

Semana 2

Semana 3

Semana 4

Semana 5

Semana 6

Semana 7

Semana 8

Orientações para realização da prova

Exame

Documentos e Informações gerais

Gabaritos

Referências da disciplina

Facilitadores da Disciplina

Repositório de REA's

Revisar envio do teste: Semana 2 - Atividade Avaliativa

Usuário

LIZIS BIANCA DA SILVA SANTOS

Curso

Fundamentos Matemáticos para Computação - COM150 - Turma 003

Teste

Semana 2 - Atividade Avaliativa

Iniciado

04/05/23 10:29

Enviado

04/05/23 10:46

Data de vencimento

05/05/23 05:00

Status

Completada

Resultado da tentativa

10 em 10 pontos

Tempo decorrido

16 minutos

Instruções

Olá, estudante!

1. Para responder a esta atividade, selecione a(s) alternativa(s) que você considerar correta(s);

2. Após selecionar a resposta correta em todas as questões, vá até o fim da página e pressione “Enviar teste”.

3. A cada tentativa, você receberá um conjunto diferente de questões.

Pronto! Sua atividade já está registrada no AVA.

Resultados exibidos

Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente

Pergunta 1

1,67 em 1,67 pontos

Para que possa ser descrita, de maneira mais formal, uma demonstração de correção, é possível denotar por X uma coleção arbitrária de valores de entrada para determinado programa ou mesmo segmento de programa, rotulado como P. Dessa forma, as ações de P realizam a transformação de X em um conjunto correspondente de valores de saída Y. Nesse sentido, diz-se que a notação $Y = P(X)$ indica que os valores de Y são dependentes de X, mediante ações executadas pelo programa P.

Analise as asserções a seguir e a relação proposta entre elas.

I. Um predicado Q (X) descreve as condições que os valores de entrada são incapazes de satisfazer.

PORQUE

II. Por exemplo, se, supostamente, um programa é capaz de calcular a raiz quadrada de um valor positivo, então, X consiste em um valor de entrada x, com Q (x) podendo ser " $x > 0$ ".

Avaliando essas asserções, é correto afirmar que:

Resposta Selecionada: b, a primeira asserção é falsa e a segunda é verdadeira.

Respostas:

a, as duas asserções são verdadeiras, mas a segunda não justifica a primeira.

b, a primeira asserção é falsa e a segunda é verdadeira.

c, as duas asserções são verdadeiras e a segunda justifica a primeira.

d, a primeira asserção é verdadeira e a segunda é falsa.

e, as duas asserções são falsas.

Comentário da resposta:

JUSTIFICATIVA

A asserção I está incorreta, porque um predicado Q (X) descreve as condições que os valores de entrada são, supostamente, capazes de satisfazer (e não incapazes). A asserção II está correta, porque a indicação " $x > 0$ " conduz necessariamente a números positivos, o que atende à necessidade de o programa calcular a raiz quadrada de valores positivos.

Pergunta 2

1,67 em 1,67 pontos

O axioma de atribuição é uma daquelas regras de inferência que se mostra especialmente apropriada para declarações de atribuição. Afinal, o que esse axioma preconiza é que, quando a precondition e a pós-condição se encontram devidamente relacionadas, a consequência é que a Tripla de Hoare consegue ser posicionada em qualquer lugar junto a uma sequência de demonstração, dispensando que seja preciso inferir a seu respeito, quanto a alguma proposição anterior nessa sequência. Essa é a razão pela qual a Tripla de Hoare, diante de uma declaração de atribuição, seja similar a uma hipótese.

Analise as asserções a seguir e a relação proposta entre elas.

I. Na precondition, busca-se a localização de todos os lugares onde aparece a variável sobre a qual se conduz uma atribuição, na declaração de atribuição, logo acima da precondition.

PORQUE

II. Em cada um desses lugares, é preciso substituir a expressão que está sendo atribuída, o que consequentemente resulta na pós-condição.

Avaliando essas asserções, é correto afirmar que:

Resposta Selecionada: d, as duas asserções são falsas.

Respostas:

a, as duas asserções são verdadeiras e a segunda justifica a primeira.

b, a primeira asserção é falsa e a segunda é verdadeira.

c, a primeira asserção é verdadeira e a segunda é falsa.

d, as duas asserções são falsas.

e, as duas asserções são verdadeiras, mas a segunda não justifica a primeira.

Comentário da resposta:

JUSTIFICATIVA

A asserção I está incorreta, porque é na pós-condição (e não na precondition) que se deve localizar todos os lugares onde aparece a variável sobre a qual se realiza a atribuição na declaração de atribuição, logo acima da pós-condição (e não da precondition). A asserção II está incorreta, porque o resultante é a precondition (e não a pós-condição), quando, em cada um desses lugares, efetua-se a substituição da expressão que está sendo atribuída.

Pergunta 3

1,66 em 1,66 pontos

Em matéria de fundamentos matemáticos para computação, especificamente em termos de programação lógica, determinado conceito refere-se a uma fbf constituída por predicados ou negação de predicados, conectada por disjunções, de tal forma que no máximo um predicado não se encontre negado.

Assinale a alternativa que apresenta a descrição correta do conceito em questão.

Resposta Selecionada: e, Cláusulas de Horn.

Respostas:

a, Teorema de Pitágoras.

b, Leis de De Morgan.

c, Paradoxos de Zenão.

d, Princípios de Russell-Burch.

e, Cláusulas de Horn.

Comentário da resposta:

JUSTIFICATIVA

No campo da lógica, uma Cláusula de Horn é constituída por disjunção de literais com, no máximo, um literal positivo. A denominação Cláusula de Horn é uma homenagem ao pesquisador Alfred Horn, a quem se credita a vanguarda no exame dessas cláusulas, em 1951. As alternativas Teorema de Pitágoras, Leis de De Morgan, Paradoxos de Zenão e Princípios de Russell-Burch compreendem temas completamente alheios à programação lógica e que não se relacionam com o objeto da questão.

Pergunta 4

1,66 em 1,66 pontos

Dentre os principais fundamentos matemáticos para computação, há a lógica de predicados, que abrange as regras de dedução. Aliás, uma regra de dedução fica evidente no raciocínio “Todas as pessoas morrerão um dia. Sérgio é uma pessoa. Portanto, Sérgio morrerá um dia”.

Assinale a alternativa que apresenta a descrição correta da regra de dedução em questão.

Resposta Selecionada: e, Particularização universal.

Respostas:

a, Generalização universal.

b, Generalização existencial.

c, Existencialismo universal.

d, Particularização existencial.

e, Particularização universal.

Comentário da resposta:

JUSTIFICATIVA

A regra de particularização universal sustenta que é possível deduzir $P(x)$, $P(y)$, $P(z)$, $P(a)$, ... de $\forall x (P(x))$, retirando, dessa forma, um quantificador universal. Ocorre que, se P é verdadeira para todos os elementos do conjunto universo, então, é possível nomear um elemento por uma designação arbitrária de variável (como x, y ou z) ou especificar uma constante particular no domínio, mantendo P como verdadeira para todos esses elementos. A alternativa "existencialismo universal" não é uma regra de dedução existente, e "generalização existencial", "particularização existencial" e "generalização universal" são regras de dedução que existem, mas com propósitos de aplicação diferentes do raciocínio apresentado no enunciado.

Pergunta 5

1,67 em 1,67 pontos

A linguagem de programação Prolog tem seu nome formado pelas iniciais de Programming in Logic (Programando em Lógica). Para todos os efeitos, trata-se de uma linguagem de programação declarativa. O conjunto de declarações que perfazem um programa em Prolog também é denominado banco de dados Prolog. A peculiaridade é que, necessariamente, em um banco de dados Prolog, os itens têm uma de duas formas.

Assinale a alternativa que apresenta a descrição correta das duas formas em questão.

Resposta Selecionada: b. Fatos e regras.

Respostas:

a. Indícios e hipóteses.

b. Fatos e regras.

c. Ações e opções.

Opt-in e opt-out.

d.

e. Linhas e colunas.

Comentário da resposta:

JUSTIFICATIVA

As duas formas em Prolog são os fatos e as regras. É válido esclarecer que as regras de Prolog são apenas outra espécie de fatos, portanto, não se pode confundir com regras de inferência. Os fatos de Prolog possibilitam a definição de predicados, mediante a declaração de quais itens pertencentes a dado conjunto universo de fato satisfazem os predicados. Por sua vez, uma regra é uma descrição de um predicado, por meio de um condicional (simbolizado pela seta na regra que vai da direita para a esquerda). As demais alternativas levam à formulação de racionais tecnicamente inconsistentes, razão pela qual estão incorretas.

Pergunta 6

1,67 em 1,67 pontos

O propósito dos testes de programa é comprovar que valores particulares de dados de entrada produzem respostas aceitáveis. Assim, os testes de programa são uma parte indispensável de qualquer esforço de desenvolvimento de projetos. Ademais, prevalece a máxima de que os testes podem provar a existência de determinadas ocorrências, mas nunca sua ausência.

Assinale a alternativa que apresenta a descrição correta das ocorrências em questão.

Resposta Selecionada: a. Erros.

Respostas:

a. Erros.

b. Hipóteses.

c. Postulados.

d. Contradições.

e. Inferências.

Comentário da resposta:

JUSTIFICATIVA

Testes de programa existem para detectar erros. Se um teste de programa não detectar um erro, isso não significa necessariamente que o erro inexistia. Nesses termos, caso um teste, com execução dirigida por determinado conjunto de condições e dados de entrada, apresente um *bug* no código do programa, há a possibilidade de corrigir esse tipo de erro. Todavia, com a exceção de programas bem elementares, testes múltiplos que não acusem erros não garantem, por si só, que tais erros não existam. Por exemplo, há os ocultos em determinada localização do código, que, rodado em tal ou qual circunstância, possa, enfim, revelar o problema. As demais alternativas levam à formulação de sentenças tecnicamente inconsistentes, razão pela qual estão incorretas.

Quinta-feira, 15 de Agosto de 2024 20h03min46s BRT

← OK