



SISTEMAS DA INFORMAÇÃO

MATERIAL INSTRUCIONAL ESPECÍFICO

CQA/UNIP – Comissão de Qualificação e Avaliação da UNIP

SISTEMAS DA INFORMAÇÃO

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 4

Christiane Mazur Doi

Doutora em Engenharia Metalúrgica e de Materiais, Mestra em Ciências - Tecnologia Nuclear, Especialista em Cálculo e Matemática Aplicada, Especialista em Língua Portuguesa e Literatura, Especialista em Recursos Gramaticais para Revisão Textual, Engenheira Química e Licenciada em Matemática, com Aperfeiçoamento em Estatística e MBA em Engenharia Econômica. Professora titular da Universidade Paulista.

José Carlos Morilla

Doutor e Mestre em Engenharia de Materiais, Mestre em Engenharia de Produção, Especialista em Engenharia Metalúrgica, Especialista em Física e Engenheiro Mecânico, com MBA em Gestão de Empresas. Professor adjunto da Universidade Paulista.

Tiago Guglielmeti Correale

Doutor em Engenharia Elétrica, Mestre em Engenharia Elétrica e Engenheiro Elétrico (ênfase em Telecomunicações). Professor titular da Universidade Paulista.

Material instrucional específico, cujo conteúdo integral ou parcial não pode ser reproduzido ou utilizado sem autorização expressa, por escrito, da CQA/UNIP – Comissão de Qualificação e Avaliação da UNIP – UNIVERSIDADE PAULISTA.

Questão 1

Questão 1.¹

Seja A um conjunto e seja $\sim$ uma relação entre pares de elementos de A . Diz-se que $\sim$ é uma relação de equivalência entre pares de elementos de A se as seguintes propriedades são verificadas, para quaisquer elementos a, a' e a'' de A :

- (i) $a \sim a$;
- (ii) se $a \sim a'$, então $a' \sim a$;
- (iii) se $a \sim a'$ e $a' \sim a''$, então $a \sim a''$.

Uma classe de equivalência do elemento a de A com respeito à relação $\sim$ é o conjunto:

$$\bar{a} = \{x \in A: x \sim a\}.$$

O conjunto quociente de A pela relação de equivalência $\sim$ é o conjunto de todas as classes de equivalência relativamente à relação $\sim$, definido e denotado como a seguir:

$$A/\sim = \{\bar{a}: a \in A\}$$

A função $\pi: A \rightarrow A/\sim$ é chamada projeção canônica e é definida como:

$$\pi(a) = \bar{a}, \forall a \in A.$$

Considerando as definições acima, analise as afirmativas.

- I. A relação de equivalência $\sim$ no conjunto A particiona o conjunto A em subconjuntos disjuntos: as classes de equivalência.
- II. A união das classes de equivalência da relação de equivalência $\sim$ no conjunto A resulta no conjunto das partes de A .
- III. As três relações seguintes

=

$\equiv (\text{mod } n)$

$\geq$

são relações de equivalência no conjunto dos números inteiros $\mathbb{Z}$.

- IV. Qualquer relação de equivalência no conjunto A é proveniente de sua projeção canônica. É correto apenas o que se afirma em

- A. II.
- B. III.
- C. I e III.
- D. I e IV.
- E. II e IV.

¹Questão 9 - Enade 2011.

1. Introdução teórica

1.1. Relação binária e relação de equivalência

Dados dois conjuntos não vazios A e B , uma relação binária do conjunto A no conjunto B é um subconjunto qualquer do produto cartesiano $A \times B$.

Se R é uma relação binária para determinado par $(a, b) \in R$, com $a \in A, b \in B$, então a está relacionado com b por R , ou seja, aRb .

Tomemos uma relação binária $\sim$ sobre um conjunto não vazio A , ou seja, uma relação R binária de A em A . Essa relação R é uma relação de equivalência sobre A se apresentar as seguintes propriedades:

- (i) R é reflexiva, ou seja, se $x \sim x, \forall x \in A$;
- (ii) R é transitiva, ou seja, se $x \sim y$, então $y \sim x, \forall x, y \in A$;
- (iii) R é simétrica, ou seja, se $x \sim y$ e $y \sim z$, então $x \sim z, \forall x, y, z \in A$.

Outra maneira de formalizar uma relação de equivalência é a descrita a seguir.

Seja X um conjunto. Uma relação R é uma relação de equivalência no conjunto X se $R \subseteq X \times X$, sendo $X \times X$ o produto cartesiano de X por X , se e somente se,

- (i) R é reflexiva em X , ou seja, se $\forall x \in X, (x, x) \in R$;
- (ii) R é transitiva em X , ou seja, se $\forall a, b, c \in X, (a, b) \in R \wedge (b, c) \in R \longrightarrow (a, c) \in R$;
- (iii) R é simétrica em X , ou seja, se $\forall a, b \in X, (a, b) \in R \longrightarrow (b, a) \in R$.

1.2. Classe de equivalência

Se R é uma relação de equivalência em A e se $a \in A$, então definimos $[a] = \{b \in A : a \sim b\}$ como classe de equivalência de a .

1.3. Partição

Seja X um conjunto não vazio. Uma partição P de X é um conjunto de subconjuntos não vazios de X tais que, se $A, B \in P$ e $A \neq B$, então

$$(i) \alpha \in P \longrightarrow \alpha \neq \emptyset ;$$

$$(ii) \text{ a reunião de } \alpha \text{ tal que } \alpha \in P \text{ resulta em } X, \text{ logo } \alpha \in P \longrightarrow \alpha \subseteq X ;$$

$$(iii) \alpha, \beta \in P, \alpha \neq \beta \longrightarrow \alpha \cap \beta = \emptyset .$$

Intuitivamente, uma partição do conjunto X , sendo X um conjunto não vazio, é uma subdivisão de X em “pedaços” (partes) que formam conjuntos não vazios e mutuamente exclusivos (disjuntos).

2. Análise das afirmativas

I – Afirmativa correta.

JUSTIFICATIVA. Uma relação de equivalência $\sim$ entre pares de elementos de um conjunto A divide (particiona) o conjunto A em subconjuntos mutuamente exclusivos (disjuntos) em suas classes de equivalência.

II – Afirmativa incorreta.

JUSTIFICATIVA. A união das classes de equivalência da relação de equivalência $\sim$ no conjunto A não resulta no conjunto das partes de A . O conjunto das partes de A é a coleção de todos os subconjuntos de A .

III – Afirmativa incorreta

JUSTIFICATIVA. A relação de equivalência deve satisfazer às propriedades de reflexividade, simetria e transitividade. A relação $\geq$ não é uma relação de equivalência em $\mathbb{Z}$, pois não é simétrica: por exemplo, $2 \geq 1$, mas 1 não é ≥ 2 .

IV – Afirmativa correta.

JUSTIFICATIVA. Como $A/\sim = \{\bar{a} : a \in A\}$, para cada elemento $a \in A$, podemos associar um elemento de $A/\sim$, pois existe uma função de $A \rightarrow A/\sim$ e suas classes de equivalência são consideradas projeções canônicas.

Alternativa correta: D.

3. Indicações bibliográficas

- ALENCAR FILHO, E. de. *Iniciação à lógica matemática*. São Paulo: Nobel, 2008.
- FEITOSA, H. de A.; PAULOVICH, L. *Um prelúdio à lógica*. São Paulo: UNESP, 2005.
- WATANABE, O. K. *Iniciação à lógica matemática*. São Paulo: Alexa Cultural, 2010.
- ZHAN, M. *Introdução à álgebra*. Rio de Janeiro: Ciência Moderna, 2013.

Questão 2**Questão 2.²**

Em determinado período letivo, cada estudante de um curso universitário tem aulas com um de três professores, esses identificados pelas letras X, Y e Z. As quantidades de estudantes (homens e mulheres) que têm aulas com cada professor é apresentada na tabela de contingência abaixo.

	Professor X	Professor Y	Professor Z
Estudantes homens	45	5	32
Estudantes mulheres	67	2	4

A partir do grupo de estudantes desse curso universitário, escolhe-se um estudante ao acaso. Qual é a probabilidade de que esse estudante seja mulher, dado que ele tem aulas apenas com o professor X?

- A. 61/73 B. 61/155 C. 67/155 D. 22/112 E. 67/112

1. Introdução teórica**Probabilidade**

A probabilidade de um evento ocorrer, considerando o espaço amostral formado pelo conjunto de todos os resultados possíveis de um experimento aleatório, é igual à razão entre o número de elementos do evento e o número de elementos do espaço amostral, desde que ele seja um conjunto equiprovável, ou seja, todos os seus elementos tenham a mesma possibilidade de ocorrer.

Ou seja, em um fenômeno aleatório, no qual as possibilidades são igualmente prováveis, a probabilidade $P(A)$ de ocorrer um evento A é igual a

$$P(A) = \frac{\text{número de casos favoráveis}}{\text{número de casos possíveis}}$$

²Questão 10 - Enade 2011.

2. Análise da questão

O número de casos possíveis é a quantidade de estudantes do professor X (soma da quantidade de estudantes homens com a quantidade de estudantes mulheres) e o número de casos favoráveis é a quantidade de mulheres para o professor X.

A probabilidade de estudantes mulheres para aulas em relação a apenas um professor X é a relação entre o total de estudantes mulheres, que é igual a 67 alunas, e o número total de alunos, que é igual a 112 alunos, ou seja, vale $67/112$.

Alternativa correta: E.

3. Indicações bibliográficas

- BARBETTA, P. A.; REIS M. M.; BORNIA A. C. *Estatística para cursos de engenharia e informática*. São Paulo: Atlas, 2008.
- LARSON, R.; FARBER, B. *Estatística aplicada*. São Paulo: Pearson, 2004.
- SCHAUM, S.; MURRAY, R.; SCHILLER, J.; SRINIVASAN, R. A. *Probabilidade e estatística*. Porto Alegre: Bookman, 2004.

Questão 3

Questão 3.³

O problema da parada para máquinas de *Turing*, ou simplesmente problema da parada, pode ser assim descrito: determinar, para qualquer máquina de *Turing* M e palavra w , se M irá eventualmente parar com entrada w .

Mais informalmente, o mesmo problema também pode ser assim descrito: dados um algoritmo e uma entrada finita, decidir se o algoritmo termina ou se executará indefinidamente.

Para o problema da parada,

- A. existe algoritmo exato de tempo de execução polinomial para solucioná-lo.
- B. existe algoritmo exato de tempo de execução exponencial para solucioná-lo.
- C. não existe algoritmo que o solucione, não importa quanto tempo seja disponibilizado.
- D. não existe algoritmo exato, mas existe algoritmo de aproximação de tempo de execução polinomial que o soluciona, fornecendo respostas aproximadas.
- E. não existe algoritmo exato, mas existe algoritmo de aproximação de tempo de execução exponencial que o soluciona, fornecendo respostas aproximadas.

1. Introdução teórica

Máquina de Turing

A máquina de Turing, concebida pelo matemático britânico Alan Turing (1912-1954), é um modelo abstrato de computador de aspectos lógicos, composto de memória, estados e transições. Por meio dessa máquina, é possível modelar um computador digital. Além disso, a máquina de Turing pode ser utilizada como um modelo de computador universal.

Segundo Alan Turing, o problema de parada é indecidível para uma máquina de Turing. Está relacionado à tomada de decisão e à computabilidade. Uma vez iniciado um programa para certa entrada finita, esse programa pode ser executado de forma finita ou infinita.

O problema computacional é uma tarefa executada por um computador. Ele pode ser considerado difícil quando a sua solução requer muitos recursos para solucioná-lo, independentemente do tipo do algoritmo. Por isso são utilizados modelos matemáticos para estudar e quantificar a dimensão dos recursos em relação ao tempo e ao armazenamento

³Questão 11 - Enade 2011.

necessários para a solução. A complexidade computacional determina, de maneira prática, os limites dos computadores.

2. Análise da questão

Não existe algoritmo para solucionar problemas de parada, pois, para dado algoritmo e para dada entrada finita, não é possível decidir se o programa terminará ou se será executado indefinidamente.

Alternativa correta: C.

3. Indicações bibliográficas

- GERSTING, J. L. *Fundamentos matemáticos para a ciência de computação*. Rio de Janeiro: LTC, 2004.
- LOPES, L. *Manual da indução matemática*. Rio de Janeiro: Interciência, 1999.
- SCHEINERMAN, E. R. *Matemática discreta - uma introdução*. São Paulo: Pioneira Thomson, 2003.

Questão 4**Questão 4.**⁴

Considere a gramática a seguir, em que S, A e B são símbolos não terminais, 0 e 1 são terminais e ε é a cadeia vazia.

$$S \rightarrow 1S|0A|\varepsilon$$

$$A \rightarrow 1S|0B|\varepsilon$$

$$B \rightarrow 1S|\varepsilon$$

A respeito dessa gramática, analise as afirmativas a seguir.

- I. Nas cadeias geradas por essa gramática, o último símbolo é 1.
- II. O número de zeros consecutivos nas cadeias geradas pela gramática é, no máximo, dois.
- III. O número de uns em cada cadeia gerada pela gramática é maior que o número de zeros.
- IV. Nas cadeias geradas por essa gramática, todos os uns estão à esquerda de todos os zeros.

É correto apenas o que se afirma em

- A. I. B. II. C. I e III. D. II e IV. E. III e IV.

1. Introdução teórica**1.1. Gramática**

A gramática formal pode ser definida como o conjunto de regras de produção de cadeias dentro de dada linguagem.

As regras são responsáveis por descrever como as cadeias devem ser formadas a partir de um alfabeto e como devem ser válidas segundo a sintaxe dessa linguagem.

A gramática descreve apenas as formas das cadeias e não o significado dessa cadeia. A teoria da linguagem formal estuda a gramática e as linguagens formais da matemática aplicada. Uma gramática formal pode ser utilizada como gerador de linguagem ou como base para um reconhecedor, conhecido como teoria dos autômatos.

⁴Questão 12 - Enade 2011.

1.2. Cadeias geradas por meio da gramática

Uma cadeia de caracteres sobre um alfabeto é formada por uma sequência finita de símbolos. Portanto, uma cadeia sem símbolos também é uma palavra válida.

1.3. Símbolos terminais e não terminais

Símbolos terminais e não terminais são elementos léxicos usados na especificação das regras de produção em uma gramática formal. Ambos são considerados conjuntos disjuntos.

2. Análise da questão e das afirmativas

Para facilitar a análise da gramática proposta, podemos utilizar sua representação gráfica por meio de uma subárvore de derivação (figura 1).

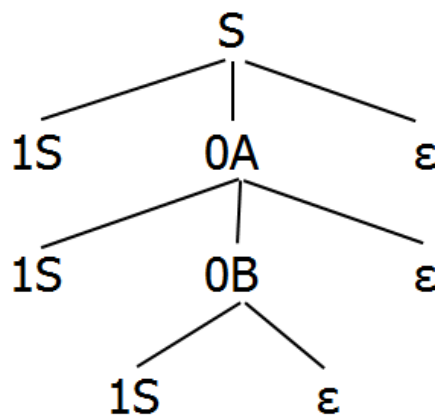


Figura 1. Subárvore da gramática.

I – Afirmativa incorreta.

JUSTIFICATIVA. Para a gramática descrita, a sequência 00 é aceita.

Derivação: $S \rightarrow 0A \rightarrow 00B \rightarrow 00$

II – Afirmativa correta.

JUSTIFICATIVA. Para a gramática descrita, o número máximo de zeros consecutivos é dois, pois, substituindo-se os símbolos não terminais "S" em "A" e "A" em "B", o resultado é, no máximo, dois zeros.

Derivação: $S \rightarrow 0A \rightarrow 00B \rightarrow 00$

III – Afirmativa incorreta

JUSTIFICATIVA. A cadeia 0 é gerada por meio da gramática e apresenta apenas um 0 e nenhum 1.

Derivação: $S \rightarrow 0A \rightarrow 0$

IV – Afirmativa incorreta.

JUSTIFICATIVA. A sequência 1010 é aceita pelas regras da gramática.

Derivação: $S \rightarrow 1S \rightarrow 10A \rightarrow 101S \rightarrow 1010A \rightarrow 1010$

Alternativa correta: B.

3. Indicação bibliográfica

- MENEZES, P. B. *Linguagens formais e autômatos*. Porto Alegre: Sagra Luzzatto, 2001.

Questão 5

Questão 5.⁵

O problema do escalonamento de intervalos tem como entrada um conjunto de intervalos numéricos (usualmente interpretados como início e fim de atividades), e o objetivo é escolher, desse conjunto, o maior número possível de intervalos disjuntos dois a dois. Há vários problemas práticos que podem ser modelados dessa forma, como, por exemplo, a seleção de tarefas com horário marcado.

O problema do escalonamento de intervalos pode ser resolvido com o algoritmo descrito a seguir. O conjunto de intervalos dados inicialmente é R e o conjunto de intervalos escolhidos, A , começa vazio.

enquanto R não estiver vazio,

seja x o intervalo de R com menor tempo de término, e que não tenha interseção com algum intervalo em A

retire x de R e adicione ao conjunto A

retorne A

A respeito desse algoritmo, analise as seguintes asserções.

Para checar se o algoritmo está correto, basta verificar que o primeiro intervalo adicionado ao conjunto A necessariamente faz parte de uma solução ótima.

PORQUE

Pode-se mostrar, por indução no número máximo de intervalos calculados (ou seja, no número de vezes que o laço “enquanto” é executado), que, embora possa haver soluções tão boas quanto A , nenhuma delas é estritamente melhor que A . O conjunto com um único intervalo é a base de indução.

Acerca dessas asserções, assinale a opção correta.

- A. As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
- B. As duas asserções são proposições verdadeiras, e a segunda não é uma justificativa correta da primeira.
- C. A primeira asserção é uma proposição verdadeira, e a segunda é uma proposição falsa.
- D. A segunda asserção é uma proposição falsa, e a segunda é uma proposição verdadeira.
- E. Tanto a primeira quanto a segunda asserções são proposições falsas.

⁵Questão 13 - Enade 2011.

1. Introdução teórica

1.1. Escalonamento

O escalonamento de processos (ou scheduling) está relacionado à atividade de organizar a CPU (Unidade Central de Processamento) por meio do escalonador de processos, o que permite executar ou viabilizar os processos concorrentes, priorizando os que devem ser mantidos na memória principal e evitando, dessa forma, a ociosidade da CPU.

1.2. Conjuntos e intervalos disjuntos

Conjunto é uma coleção de elementos e a sua ordem ou quantidade de elementos listados não é relevante.

Conjuntos disjuntos são aqueles que não apresentam nenhum elemento em comum, ou seja, aqueles cuja intersecção gera o conjunto vazio. São os chamados de mutuamente exclusivos.

1.3. Indução e proposições

A indução é utilizada na matemática como método e prova matemática para demonstrar a verdade de proposições. Pode ser utilizada para provar que certo enunciado pode ser válido para todos os números naturais.

2. Análise da questão

A questão foi anulada devido à existência de incoerências relacionadas às asserções apresentadas.

Na primeira asserção, justifica-se o primeiro intervalo adicionado ao conjunto A como parte necessária de uma solução ótima. O algoritmo proposto é um algoritmo guloso, isto é, ele constrói uma solução escolhendo, a cada interação, a melhor solução, de acordo com critérios estabelecidos anteriormente. Dessa forma, podemos não encontrar uma solução ótima em determinado conjunto de intervalos R. Há incoerências na asserção.

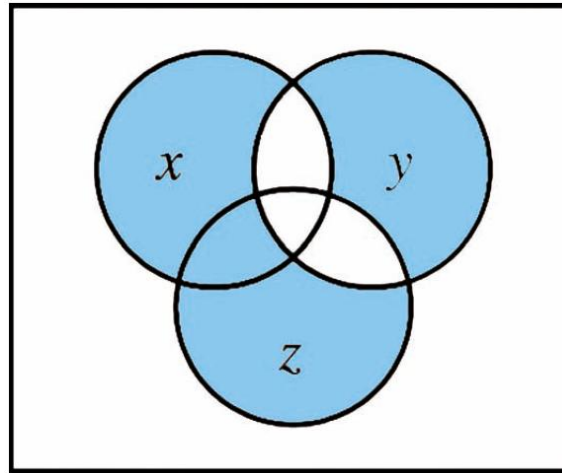
Para a segunda asserção, um conjunto com um único intervalo é usado como base de indução. Porém a hipótese de A ser estritamente melhor não é válida, conforme explicado na análise da primeira asserção.

3. Indicações bibliográficas

- MOTA FILHO, J. E. *Descobrendo o Linux – Entenda o sistema operacional GNU/Linux*. 2. ed. São Paulo: Novatec, 2007.
- SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. *Sistemas operacionais com Java*. Rio de Janeiro: Elsevier, 2008.
- TANENBAUM, A. S. *Sistemas operacionais modernos*. 2. ed. São Paulo: Prentice Hall, 2003.

Questão 6**Questão 6.**⁶

Observe o diagrama de Venn a seguir.



A função representada em azul no diagrama também poderia ser expressa pela função lógica $f(x, y, z) =$

- A. $(x + z)y + x\bar{y}z$.
- B. $(x + z)y + \bar{x}y\bar{z}$.
- C. $(x + z)y + \bar{x}\bar{y}\bar{z}$.
- D. $(x + z)\bar{y} + x\bar{y}z$.
- E. $(x + z)\bar{y} + \bar{x}y\bar{z}$.

1. Introdução teórica**Diagrama de Venn**

O diagrama de Venn representa, de forma gráfica, relações lógicas de uma quantidade finita de conjuntos.

Vejamos, por exemplo, as relações entre os conjuntos x e y expressas no diagrama de Venn mostrado na figura 1. Nesse diagrama, a indicação $x + y$ refere-se à soma dos elementos de x com os elementos de y , sem que haja repetição de contagem dos elementos comuns aos dois conjuntos. A indicação $\overline{x+y}$ refere-se ao complemento de $x + y$.

⁶Questão 14 - Enade 2011.

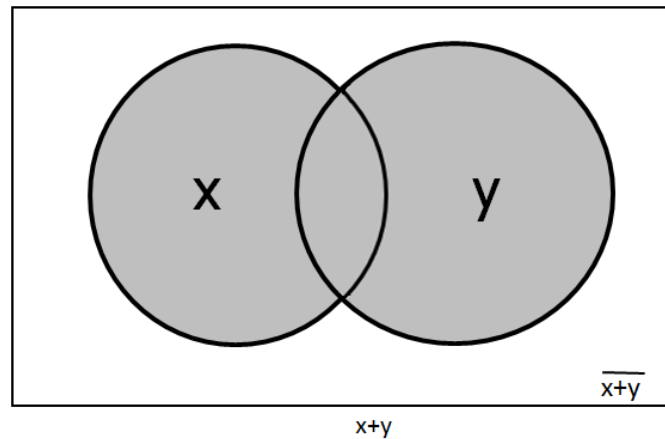


Figura 1. Primeiro exemplo de diagrama de Venn.

Vejamos, na figura 2, o segundo exemplo de diagrama de Venn. Nele, temos a indicação dos elementos comuns aos conjuntos x e y ($x \cdot y$), dos elementos de x excluídos dos elementos que ele tem em comum com y ($x \cdot \bar{y}$) e dos elementos de y excluídos dos elementos que ele tem em comum com x ($\bar{x} \cdot y$).

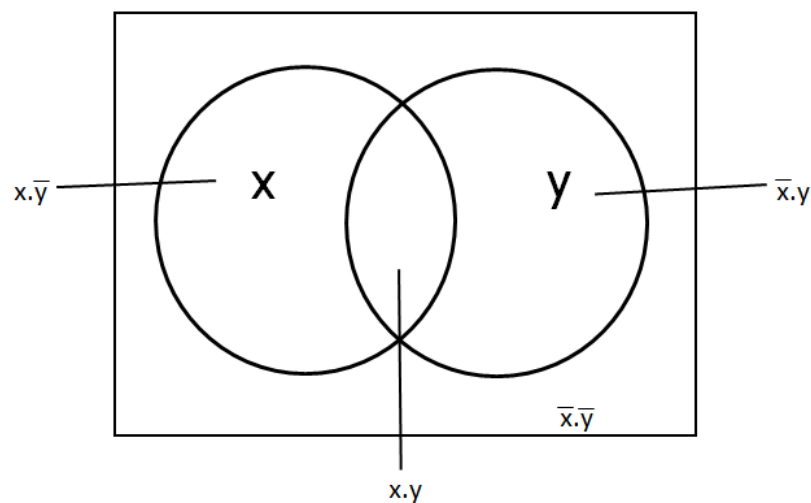


Figura 2. Segundo exemplo de diagrama de Venn.

Vejamos, na figura 3, o terceiro exemplo de diagrama de Venn. Nele, além dos conjuntos x e y , temos o conjunto z e, analogamente ao que foi explicado para a figura 2, as indicações de diversas relações entre esses conjuntos.

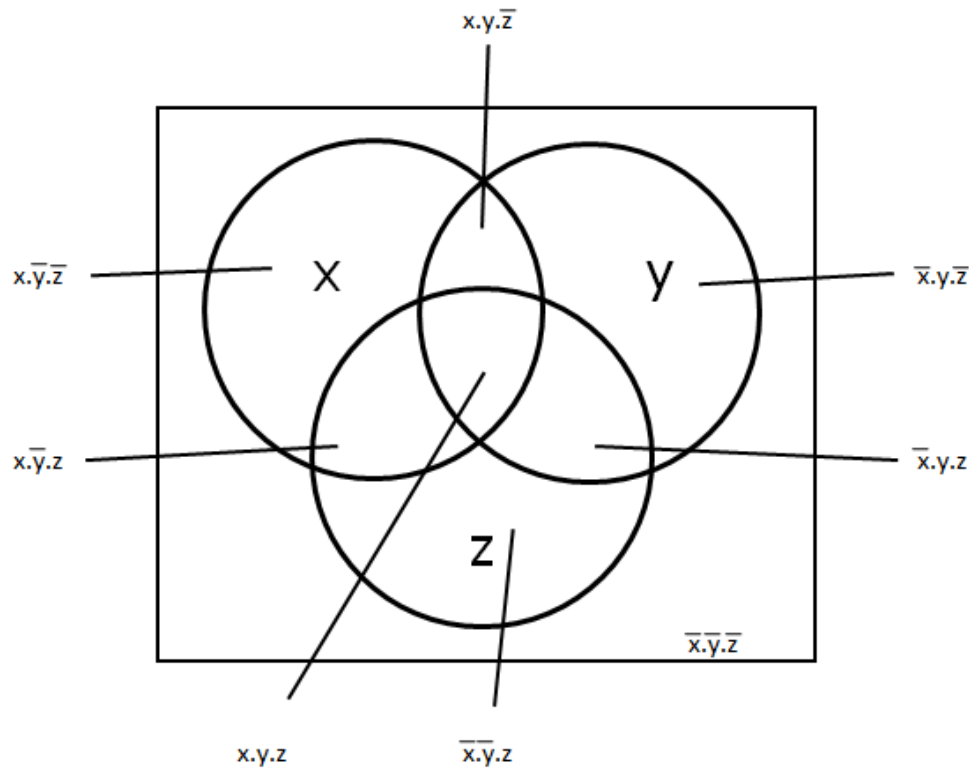


Figura 3. Terceiro exemplo de diagrama de Venn.

2. Análise da questão

A expressão da função lógica $f(x, y, z)$ pode ser expressa por meio da função $(x + y)\bar{y} + \bar{x}yz$, pois representa a função lógica da união dos conjuntos x e z, intersecção com o complementar de y e união com a intersecção do complementar de xy e complementar de z.

Alternativa correta: E.

3. Indicações bibliográficas

- ALENCAR FILHO, E. de. *Iniciação à lógica matemática*. São Paulo: Nobel, 2008.
- FEITOSA, H. de A.; PAULOVICH, L. *Um prelúdio à lógica*. São Paulo: UNESP, 2005.
- WATANABE, O. K. *Iniciação à lógica matemática*. São Paulo: Alexa Cultural, 2010.

Questão 7

Questão 7.⁷

Suponha que seja necessário desenvolver uma ferramenta que apresente o endereço IP dos múltiplos roteadores, salto a salto, que compõem o caminho do hospedeiro em que a ferramenta é executada até um determinado destino (segundo seu endereço IP), assim como o *round-trip time* até cada roteador. Tal ferramenta precisa funcionar na *Internet* atual, sem demandar mudanças em roteadores nem a introdução de novos protocolos.

Considerando o problema acima, qual dos seguintes protocolos representaria a melhor (mais simples e eficiente) solução?

- A. IP: Internet Protocol.
- B. UDP: User Datagram Protocol.
- C. TCP: Transmission Control Protocol.
- D. ICMP: Internet Control Message Protocol.
- E. DHCP: Dynamic Host Configuration Protocol.

1. Introdução teórica

Protocolos TCP, IP, UDP, ICMP e DHCP

O TCP (Transmission Control Protocol) e o IP (Internet Protocol) formam um conjunto de pilhas de protocolos, que, por sua vez, são modelos de camadas. Nesse conjunto, cada camada é responsável por tarefas e serviços bem definidos, que tratam de dados abstratos para a comunicação entre servidor e clientes. Os principais benefícios oferecidos por esse modelo são padronização, interconectividade, roteamento, protocolo robusto e internet. As principais camadas do TCP/IP são rede, internet, transporte e aplicação.

O UDP (User Datagram Protocol) é um protocolo da camada de transporte que permite escrever um datagrama e encapsulá-lo em um pacote do tipo Ipv4 ou Ipv6 para ser enviado ao seu destino de tipo não confiável e sem conexão. Além disso, os seus serviços de broadcast e multicast enviam pacotes para toda a rede.

O ICMP (Internet Control Message Protocol) é um protocolo que fornece relatórios de erros (da fonte original) e suas mensagens são enviadas de forma automática.

O DHCP (Dynamic Host Configuration Protocol) é um protocolo de serviço do TCP/IP responsável pela configuração dinâmica entre os terminais.

⁷Questão 15 - Enade 2011.

2. Análise da questão

Um ICMP (Internet Control Message Protocol) envia suas mensagens de forma automática e não utiliza mudanças nos roteadores ou novos protocolos.

Alternativa correta: D.

3. Indicações bibliográficas

- COULORIS, G.; *Sistemas distribuídos*. 4. ed. Porto Alegre: Bookman, 2007.
- GOMES, D. A. *Web Services SOAP em Java*. São Paulo: Novatec, 2009.
- TANENBAUM, A. S.; STEEN, M. V. *Sistemas distribuídos, princípios e paradigmas*. 2. ed. São Paulo: Person Prentice Hall, 2007.

Questão 8

Questão 8.⁸

Um navegador *web* executa em um hospedeiro **A**, em uma rede de uma organização, e acessa uma página localizada de um servidor *Web* em um hospedeiro **B**, situado em outra rede na *internet*. A rede em que **A** se situa conta com um servidor DNS local. Um profissional deseja fazer uma lista com a sequência de protocolos empregados e comparar com o resultado apresentado por uma ferramenta de monitoramento executada no hospedeiro **A**. A lista assume que

- i) todas as tabelas com informações temporárias e caches estão vazias;
- ii) o hospedeiro cliente está configurado com o endereço IP do servidor DNS local.

Qual das sequências a seguir representa a ordem em que mensagens, segmentos e pacotes serão observados em um meio físico ao serem enviados pelo hospedeiro **A**?

- A. ARP, DNS/UDP/IP, TCP/IP e HTTP/TCP/IP.
- B. ARP, DNS/UDP/IP, HTTP/TCP/IP e TCP/IP.
- C. DNS/UDP/IP, ARP, HTTP/TCP/IP e TCP/IP.
- D. DNS/UDP/IP, ARP, TCP/IP e HTTP/TCP/IP.
- E. HTTP/TCP/IP, TCP/IP, DNS/UDP/IP e ARP.

1. Introdução teórica

Protocolos ARP, TCP, IP, UDP, HTTP e DNS

O ARP (Address Resolution Protocol) é um protocolo que permite encontrar um endereço na camada de enlace utilizando a camada de rede e um pacote ARP com endereço IP de outro host. O ARP permite que endereços de IP sejam independentes do endereço da Ethernet, logo, sua funcionalidade depende do funcionamento de todos os hosts. O ARP é utilizado em vários tipos de redes, além de ser empregado para solucionar endereços de diferentes protocolos na rede.

O TCP (Transmission Control Protocol) e o IP (Internet Protocol) formam conjuntos de pilhas de protocolos, também chamados de modelos de camadas. Nesses conjuntos, cada camada é responsável por tratar de dados abstratos para comunicação entre servidor e clientes. Os principais benefícios oferecidos por essa comunicação são padronização,

⁸Questão 16 - Enade 2011.

interconectividade, roteamento, protocolo robusto e internet. As principais camadas do TCP/IP são rede, internet, transporte e aplicação.

O UDP (User Datagram Protocol) é um protocolo da camada de transporte que permite escrever um datagrama e encapsulá-lo em um pacote do tipo Ipv4 ou Ipv6 para ser enviado ao seu destino, mesmo que o protocolo seja do tipo não confiável e sem conexão.

Os serviços de broadcast e multicast enviam pacotes para toda a rede.

O HTTP (Hypertext Transfer Protocol) é um protocolo de comunicação (camada de aplicação) que permite que a comunicação de dados seja estabelecida em sistemas de informação de hipermídia colaborativa da World Wide Web (www). Um hipertexto é uma estrutura que utiliza lógicas hiperlinks e nós de uma rede de internet, enquanto o HTTP corresponde ao protocolo responsável pela troca de dados ou pela transferência de hipertexto.

Um servidor web apresenta programas de aceitação de pedidos de protocolos HTTP recebidos de navegadores (*browsers*) de clientes e é responsável pelo encaminhamento das respostas por meio do HTTP em documentos do tipo HTML, juntamente com os dados solicitados pelos clientes.

O DNS (Domain Name System) é um sistema responsável pelo gerenciamento de hierarquia de nomes distribuídos: trata dos nomes de domínios de endereços de redes IP. Esse sistema traduz os nomes solicitados pelos endereços IP para localizar hosts e domínios específicos.

Um cache é um dispositivo que permite o acesso rápido a um processo na memória principal de um computador.

2. Análise da questão

Os protocolos ARP, DNS/UDP/IP, TCP/IP e HTTP/TCP/IP mantêm suas informações temporárias, caches, sempre vazias e suas configurações de endereços IP sempre no servidor local.

Alternativa correta: A.

3. Indicações bibliográficas

- COULORIS, G. *Sistemas distribuídos*. 4. ed. Porto Alegre: Bookman, 2007.

- GOMES, D. A. *Web Services SOAP em Java*. São Paulo: Novatec, 2009.
- TANENBAUM, A. S.; STEEN, M. V. *Sistemas distribuídos, princípios e paradigmas*. 2. ed. São Paulo: Person Prentice Hall, 2007.

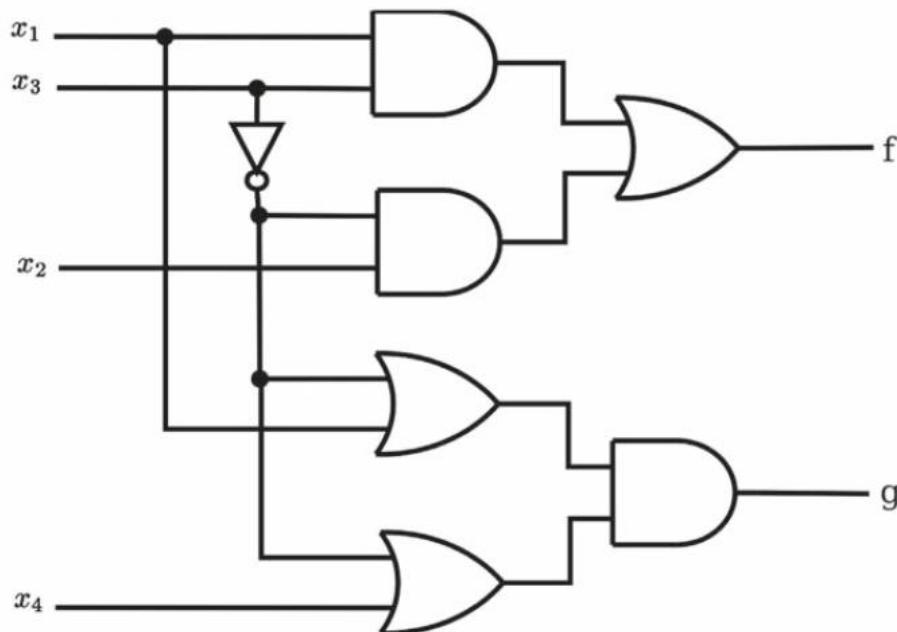
Questão 9

Questão 9.⁹

A tabela a seguir apresenta a relação de mintermos e maxtermos para três variáveis.

Linha	x_1	x_2	x_3	Mintermo	Maxtermo
0	0	0	0	$m_0 = \bar{x}_1\bar{x}_2\bar{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \bar{x}_1\bar{x}_2x_3$	$M_1 = x_1 + x_2 + \bar{x}_3$
2	0	1	0	$m_2 = \bar{x}_1x_2\bar{x}_3$	$M_2 = x_1 + \bar{x}_2 + x_3$
3	0	1	1	$m_3 = \bar{x}_1x_2x_3$	$M_3 = x_1 + \bar{x}_2 + \bar{x}_3$
4	1	0	0	$m_4 = x_1\bar{x}_2\bar{x}_3$	$M_4 = \bar{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\bar{x}_2x_3$	$M_5 = \bar{x}_1 + x_2 + \bar{x}_3$
6	1	1	0	$m_6 = x_1x_2\bar{x}_3$	$M_6 = \bar{x}_1 + \bar{x}_2 + x_3$
7	1	1	1	$m_7 = x_1x_2x_3$	$M_7 = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$

Analise o circuito de quatro variáveis a seguir.



Considerando esse circuito, as funções **f** e **g** são, respectivamente,

- A. $\Sigma m(0,1,2,3,6,7,8,9)$ e $\Sigma m(2,3,6,7,10,14)$.
- B. $\Sigma m(4,5,10,11,12,13,14,15)$ e $\Sigma m(0,1,4,5,8,9,11,12,13,15)$.
- C. $\Pi M(0,1,2,3,6,7,8,9)$ e $\Pi M(0,1,4,5,8,9,11,12,13,15)$.
- D. $\Pi M(4,5,10,11,12,13,14,15)$ e $\Sigma m(2,3,6,7,10,14)$.
- E. $\Pi M(4,5,10,11,12,13,14,15)$ e $\Pi M(2,3,6,7,10,14)$.

⁹Questão 17 - Enade 2011.

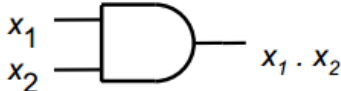
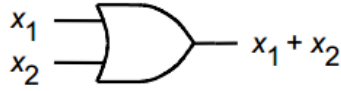
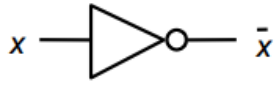
1. Introdução teórica

1.1. Portas lógicas

Portas lógicas, ou circuitos lógicos, operam sinais lógicos na entrada e na saída e são dependentes de funções implementadas dentro desses circuitos. As portas são aplicadas nos circuitos eletrônicos. O sinal de ligado é representado por “1” e o de desligado, “0”.

O quadro 1 a seguir mostra as portas lógicas AND, OR e NOT e os seus símbolos.

Quadro 1. Portas lógicas AND, OR e NOT.

Porta lógica	Símbolo	Saída
AND		A saída será 1 se ambas as entradas forem iguais a 1; nos demais casos, a saída será 0.
OR		A saída será 0 se ambas as entradas forem iguais a 0; nos demais casos, a saída será 1.
NOT		Se a entrada for igual a 1, a saída será 0; se não, a saída será 1.

1.2. Funções lógicas

Admitindo que as portas lógicas representem também circuitos eletrônicos que, de alguma maneira, realizam funções, é possível, então, escrever as funções do circuito, conforme a figura 1 a seguir.

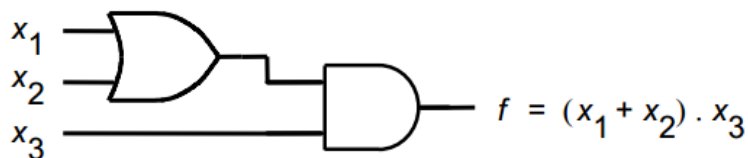


Figura 1. Funções do circuito.

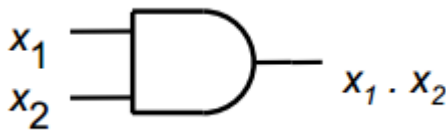
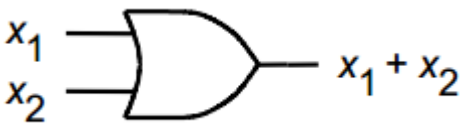
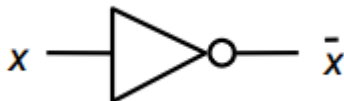
Temos x_1 e x_2 entrando em uma porta lógica OR ($x_1 + x_2$) e a saída dessa porta entrando em uma porta AND, juntamente com x_3 .

1.3. Tabelas verdade e análise de circuitos lógicos

Tabela verdade é uma tabela matemática na qual são aplicadas as lógicas que determinam fórmulas válidas e corretas para a análise de circuitos lógicos. As portas lógicas

do circuito são analisadas considerando todas as possíveis entradas, como pode ser visto no quadro 2 a seguir.

Quadro 2. Análise de circuito lógico.

Porta lógica	Símbolo	Tabela															
AND		<table><tr><th>x1</th><th>x2</th><th>x1.x2</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x1	x2	x1.x2	0	0	0	0	1	0	1	0	0	1	1	1
x1	x2	x1.x2															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
OR		<table><tr><th>x1</th><th>x2</th><th>x1+x2</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x1	x2	x1+x2	0	0	0	0	1	1	1	0	1	1	1	1
x1	x2	x1+x2															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
NOT		<table><tr><th>x</th><th>$\overline{x}$</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	$\overline{x}$	0	1	1	0									
x	$\overline{x}$																
0	1																
1	0																

1.4. Mintermos e maxtermos

Mintermos e maxtermos permitem reescrever funções lógicas padronizadas para que seja possível a simplificação dessas funções. Dessa forma, podemos reduzir o número de portas lógicas para serem implementadas por meio dos circuitos lógicos.

A função expressa como soma mintermos deve incluir os mintermos cujos números correspondem às linhas na tabela verdade, na qual a função tem valor lógico 1.

A função expressa como produto de maxtermos deve incluir os maxtermos cujos números correspondem às linhas na tabela verdade, na qual a função tem valor lógico 0.

2. Análise da questão

Para a função f , temos o que segue.

- x_1 e x_3 entrando em uma porta AND ($x_1 \cdot x_3$), saída S_1 .
- x_2 e o inverso de x_3 entrando em uma porta AND ($x_2 \cdot \overline{x_3}$), saída S_2 .
- As duas saídas (S_1 e S_2) entrando em uma porta OR ($S_1 + S_2$).

Portanto, $f = x_1 \cdot x_3 + x_2 \cdot \overline{x_3}$

Para a função g, temos o que segue.

- x_1 e o inverso de x_3 entrando em uma porta OR ($x_1 + \overline{x_3}$), saída S_3 .
- O inverso de x_3 e x_4 entrando em uma porta OR ($\overline{x_3} + x_4$), saída S_4 .
- As duas saídas (S_3 e S_4) entrando em uma porta AND ($S_3 \cdot S_4$).

Portanto, $g = (x_1 + \overline{x_3})(\overline{x_3} + x_4)$

O próximo passo é a criação da tabela verdade com todas as situações possíveis de entrada para o circuito e os cálculos para as funções f e g. Como temos quatro entradas (x_1 , x_2 , x_3 e x_4), há 2^4 variações, ou seja, 16 variações (quadro 3).

Quadro 3. Tabela verdade com 16 linhas.

Linha	x_1	x_2	x_3	$\overline{x_3}$	x_4	$x_1 \cdot x_3$	$x_2 \cdot \overline{x_3}$	f	$x_1 + \overline{x_3}$	$\overline{x_3} + x_4$	g
0	0	0	0	1	0	0	0	0	1	1	1
1	0	0	0	1	1	0	0	0	1	1	1
2	0	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	1	0	0	0	0	1	0
4	0	1	0	1	0	0	1	1	1	1	1
5	0	1	0	1	1	0	1	1	1	1	1
6	0	1	1	0	0	0	0	0	0	0	0
7	0	1	1	0	1	0	0	0	0	1	0
8	1	0	0	1	0	0	0	0	1	1	1
9	1	0	0	1	1	0	0	0	1	1	1
10	1	0	1	0	0	1	0	1	1	0	0
11	1	0	1	0	1	1	0	1	1	1	1
12	1	1	0	1	0	0	1	1	1	1	1
13	1	1	0	1	1	0	1	1	1	1	1
14	1	1	1	0	0	1	0	1	1	0	0
15	1	1	1	0	1	1	0	1	1	1	1

Para encontrarmos as equações para uma função mintermos a partir de sua tabela verdade, consideramos os 1s da função e para uma função maxtermos consideramos os 0s.

Mintermos:

$$f = \sum m(4,5,10,11,12,13,14,15)$$

$$g = \sum m(0,1,4,5,8,9,11,12,13,15)$$

Maxtermos:

$$f = \prod M(0,1,2,3,6,7,8,9)$$

$$g = \prod M(2,3,6,7,10,14)$$

Alternativa correta: B.

3. Indicações bibliográficas

- IDOETA, I. V.; CAPUANO, F. G. *Elementos de eletrônica digital*. São Paulo: Érica, 1998.
- LOURENÇO, A. C. de; CRUZ, E. C. A.; FERREIRA, S. *Circuitos digitais*. São Paulo: Érica, 1996.
- WAGNER, F. R.; REIS, A. I.; RIBAS, R. P. *Fundamentos de circuitos digitais*. Porto Alegre: Bookman, 2008.

Questão 10**Questão 10.**¹⁰

Um vendedor de artigos de pesca obteve com um amigo o código executável (já compilado) de um programa que gerencia vendas e faz o controle de estoque, com o intuito de usá-lo em sua loja. Segundo o seu amigo, o referido programa foi compilado em seu sistema computacional pessoal (sistema A) e funciona corretamente. O vendedor constatou que o programa executável também funciona corretamente no sistema computacional de sua loja (sistema B). Considerando a situação relatada, analise as afirmações a seguir.

- I. Os computadores poderiam ter quantidades diferentes de núcleos (*cores*).
- II. As chamadas ao sistema (*system call*) do sistema operacional no sistema A devem ser compatíveis com as do sistema B.
- III. O conjunto de instruções do sistema A poderia ser diferente do conjunto de instruções do sistema B.
- IV. Se os registradores do sistema A forem de 64 *bits*, os registradores do sistema B poderiam ser de 32 *bits*.

É correto o que se afirma em

- A. III, apenas.
- B. I e II, apenas.
- C. III e IV, apenas.
- D. I, II e IV, apenas.
- E. I, II, III e IV.

1. Introdução teórica**1.1. Sistema operacional**

Um sistema operacional contém coleções de softwares responsáveis pelo gerenciamento de um computador e pelos recursos oferecidos para que seja possível a execução de um programa. É o componente essencial de um sistema de software dentro de um sistema de computador, no qual são executadas as aplicações de software. Esse sistema permite gerenciar alocações de memória, chamadas de sistemas, solicitações de interrupções, compartilhamento de recursos, escalonamento de processos e controle dos dispositivos instalados, além de facilitar a interação com usuários.

¹⁰Questão 18 - Enade 2011.

1.2. Compiladores e chamadas de sistemas

Um compilador é um programa (ou vários programas) responsável pela tradução de um código fonte (ou linguagem de alto nível) em outra linguagem (de baixo nível), a fim de que ele possa ser executado por uma máquina.

Uma chamada de sistema (*system call*) é um mecanismo que permite ao programa requisitar serviços de um sistema operacional acessando o seu núcleo.

1.3. CPU e registradores

A CPU (Unidade Central de Processamento) controla todo o funcionamento de um computador. Apresenta internamente os registrados, que são unidades de memória capazes de armazenar *bits*.

Os registradores são circuitos digitais capazes de armazenar temporariamente as informações binárias. Esses circuitos permitem a execução de instruções de programas na memória principal de um computador.

2. Análise da questão

I – Afirmativa correta.

JUSTIFICATIVA. Programas executáveis já compilados podem funcionar corretamente em computadores que apresentem quantidades diferentes de núcleos, desde que as chamadas de sistemas (*system call*) sejam compatíveis.

II – Afirmativa correta.

JUSTIFICATIVA. Como o programa executável funcionou tanto no sistema A quanto no sistema B, as chamadas de sistemas (*system call*) devem ser compatíveis.

III – Afirmativa incorreta

JUSTIFICATIVA. Os conjuntos de instruções do sistema A e do sistema B são iguais, pois executam o mesmo código já compilado.

IV – Afirmativa incorreta.

JUSTIFICATIVA. A escolha do uso dos registradores, no caso do programa que gerencia vendas, fica a cargo do compilador. Se o programa for compilado em um sistema de 64 bits, o sistema B não poderia ser de 32 bits.

Alternativa correta: B.

3. Indicações bibliográficas

- AHO, A. V.; LAM, M. S.; SETHI, R.; ULLMANN, J. D. *Compiladores: princípios, técnicas e ferramentas*. 2. ed. São Paulo: Pearson Addison-Wesley, 2008.
- LOUDEN, K. C. *Compiladores: princípios e práticas*. São Paulo: Pioneira Thomson Learning, 2004.
- TOSCANI, S. S.; PRICE, A. M. de A. *Implementação de linguagens de programação – compiladores*. 3. ed. Porto Alegre: Artmed, 2009. v. 9.

Questões 11 e 12**Questão 11.**¹¹

Leia o texto a seguir.

UML é uma linguagem padrão para desenvolver e documentar projetos de software e permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados. Ela surgiu como uma proposta de ser uma linguagem para modelagem de dados que usava diversos artefatos para representar o modelo de negócio e um desses artefatos é o diagrama de classes.

PRESSMAN, R.S. *Engenharia de software*. 6. ed. Porto Alegre: Bookman, 2006 (com adaptações).

Se um projeto não tem a documentação apropriada ou se está com a documentação desatualizada, uma opção é a engenharia reversa que possibilita mapear códigos para diagramas UML. A seguir, é apresentado um código na linguagem de programação JAVA.

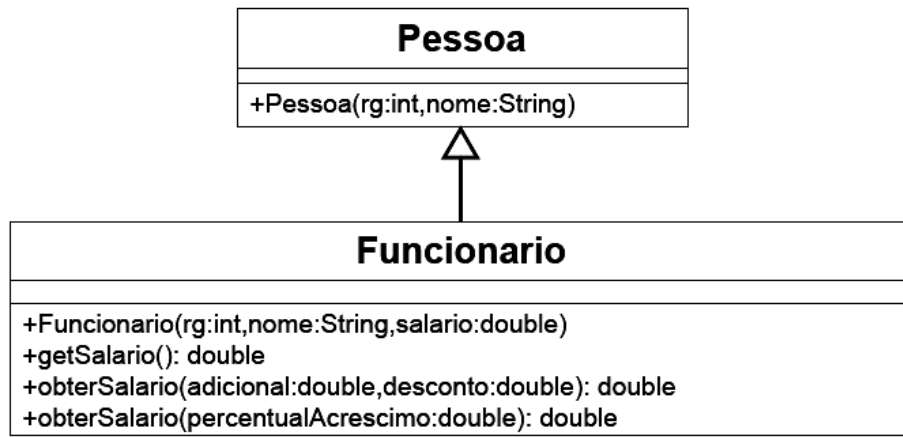
```

1 package default;
2
3 public class Funcionario extends Pessoa {
4     private double salario;
5
6     public Funcionario(int rg, String nome,
7         double salario) {
8         super(rg, nome);
9         this.salario = salario;
10    }
11    public double getSalario() {
12        return salario;
13    }
14    }
15    public double obterSalario(double
16        percentualAcrescimo) {
17        double salarioReajustado = salario +
18            salario * percentualAcrescimo / 100;
19        return salarioReajustado;
20    }
21    }
22    public double obterSalario(double
23        adicional, double desconto){
24        return this.getSalario() + adicional - desconto;
25    }
26    }
27 }
```

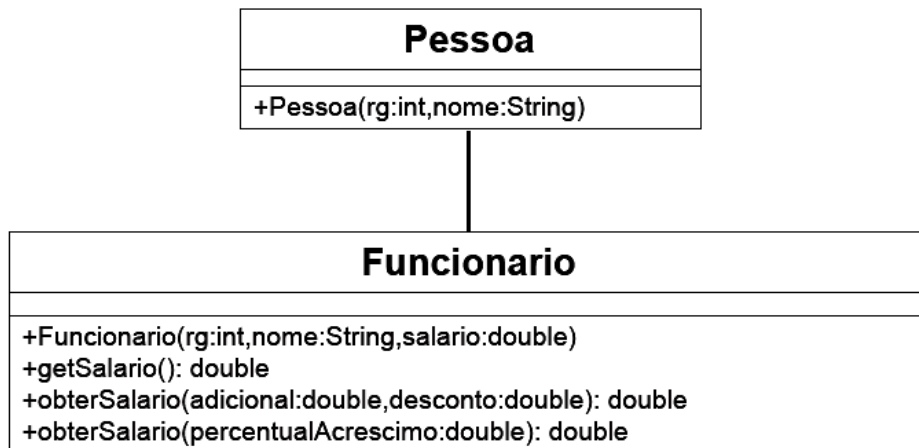
Utilizando a engenharia reversa nesse trecho de código, o diagrama UML de classes correspondente é

¹¹Questão 29 – Enade 2014.

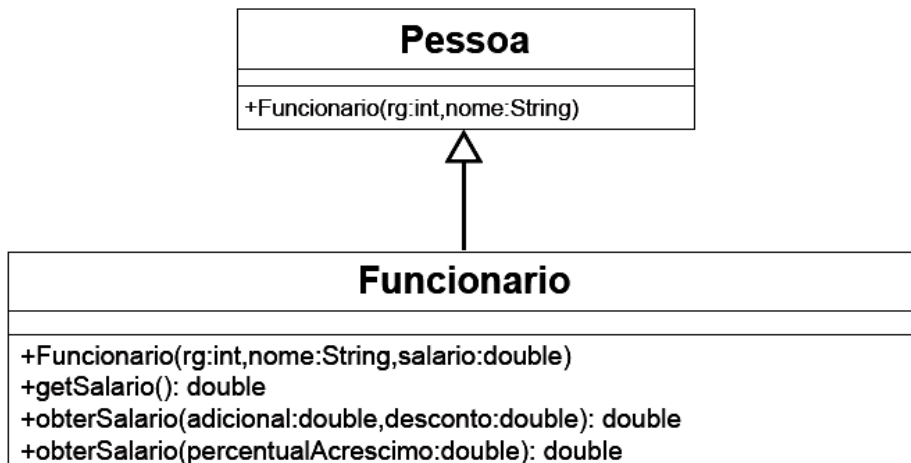
A.



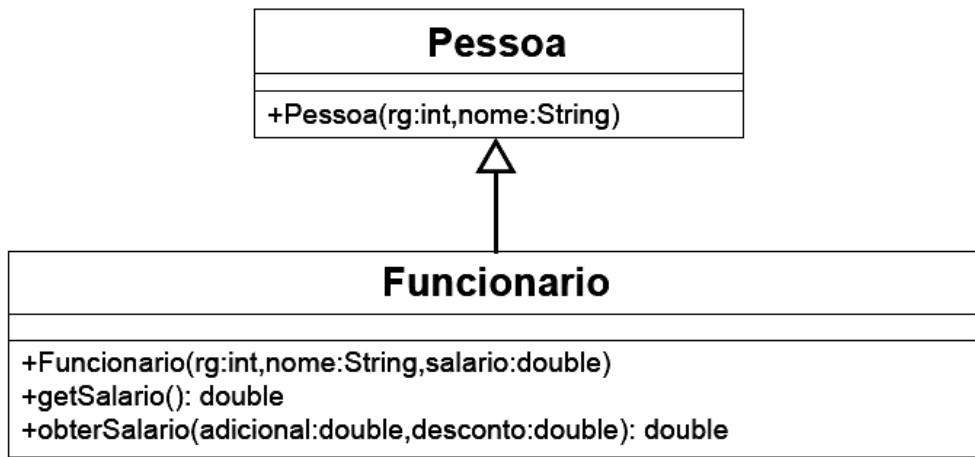
B.



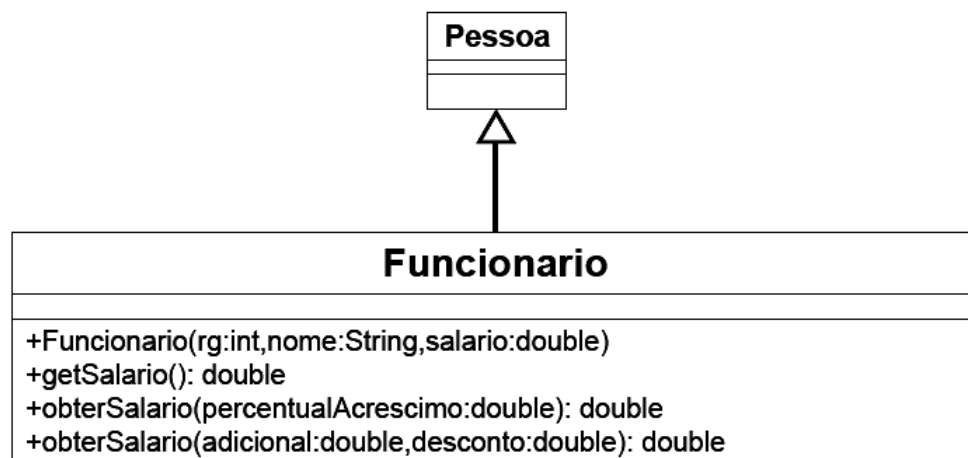
C.



D.



E.

**Questão 12.**¹²

Leia o texto a seguir.

Casos de uso podem ser organizados agrupando-os em pacotes do mesmo modo como são organizadas as classes. Também podem ser organizados pela especificação de relacionamentos de generalização, inclusão e extensão, existentes entre eles.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. *UML - Guia do Usuário*. Campus, 2006 (com adaptações).

Considerando os relacionamentos existentes entre os casos de uso, avalie as afirmativas a seguir.

- I. Para casos de uso, a generalização significa que o caso de uso filho herda o comportamento e o significado do caso de uso pai e no caso de uso filho deverá acrescentar ou sobrescrever o comportamento de seu pai.
- II. Um relacionamento de inclusão entre casos de uso significa que o caso de uso base incorpora explicitamente o comportamento de outro caso de uso em uma localização especificada. O caso de uso base poderá permanecer isolado, mas, sob certas

¹²Questão 17 – Enade 2014.

condições, seu comportamento poderá ser incluído pelo comportamento de outro caso de uso.

- III. Um relacionamento estendido entre casos de uso significa que o caso de uso base incorpora implicitamente o comportamento de outro caso de uso em um local especificado indiretamente pelo caso de uso estendido. O caso de uso estendido nunca permanece isolado, mas é apenas instanciado como parte de alguma base maior que o estende.
- IV. Um relacionamento estendido é utilizado para a modelagem de parte de um caso de uso que o usuário poderá considerar como um comportamento opcional do sistema e para a modelagem de um subfluxo separado, que é executado somente sob determinadas condições.

É correto apenas o que se afirma em

- A. I e II.
- B. I e IV.
- C. II e III.
- D. I, III e IV.
- E. II, III e IV.

1. Introdução teórica

1.1. UML (Unified Modeling Language)

A UML (Unified Modeling Language) é uma linguagem-padrão para a elaboração da estrutura de projetos de software.

Os autores destacam os quatro objetivos básicos da UML, quanto aos artefatos de um sistema complexo de software: visualizar, especificar, construir e documentar.

O principal objetivo da UML é estabelecer um modelo ou representar conceitual e fisicamente um sistema.

1.1.1. UML para visualização

Um projeto de software envolve mais do que o código- Além disso, um mesmo problema pode ser resolvido de diferentes formas, com códigos muito distintos. Ainda que

programas com códigos diferentes possam atender aos mesmos requisitos, nem todas as soluções são igualmente úteis ou interessantes.

Se dispusermos apenas do código-fonte de um programa, sem nenhum tipo de documentação, o entendimento da solução fica muito mais difícil. Programas comerciais podem ser muito longos e conter milhões de linhas de código- Dessa forma, é importante ter uma maneira de visualizar o todo, a estrutura geral de um software, sem que, necessariamente, haja a obrigação de se ler o código-fonte de modo completo.

Desenhos e diagramas informais podem ajudar nessa visualização, mas a UML fornece uma linguagem-padrão que permite que diferentes pessoas e empresas façam e interpretem diagramas e modelos de uma mesma forma.

1.1.2. UML para especificação

Antes de escrevermos o código para um programa, é preciso conhecer quais são as necessidades do cliente. Além disso, grandes projetos envolvem problemas complexos de software.

Muitas empresas contratam profissionais especializados para encontrar a melhor solução, como os arquitetos de software. Esses profissionais precisam de uma linguagem para especificar, construir e documentar a solução e a UML oferece uma linguagem especificamente desenvolvida para esse propósito.

1.1.3. UML para construção

Ainda que a UML não seja uma linguagem de programação propriamente dita, é possível estabelecer um mapeamento entre os modelos construídos pela UML e o código-fonte em diversas linguagens de programação, como Java ou C#.

Essa facilidade do mapeamento vem do alto grau de expressividade aliado à baixa ambiguidade, o que não é o caso quando consideramos outras formas de especificação, como um texto puro. Essa precisão faz com que existam ferramentas capazes de converter alguns diagramas UML em código-fonte com pouca ou nenhuma intervenção do usuário.

1.1.4. UML para documentação

Um projeto de software envolve mais do que apenas o código- Empresas de software costumam produzir diversos outros elementos, chamados de artefatos, como:

- requisitos do software;
- documentações da arquitetura;
- projetos e planos de projeto;
- código-fonte;
- testes e planos de teste;
- protótipos.

A UML é utilizada para documentar diversos desses artefatos, de modo formal e com baixa ambiguidade.

1.2. Diagrama de classes

Formalmente, define-se uma classe como uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.

É importante diferenciar as classes dos objetos: uma classe define um conjunto de objetos, mas não é um objeto. Por exemplo, podemos definir a classe das “xícaras de café” como sendo objetos cerâmicos, ocos, que servem para armazenar uma quantidade pequena de líquido quente. Essa classe abrange todas as possíveis xícaras, de diferentes formas e tamanhos. Se tomarmos um exemplo específico, como uma xícara de restaurante, dizemos que essa xícara é um objeto, ou uma instância específica de uma classe. Utilizamos uma classe para definir uma categoria de objetos similares.

A UML apresenta um diagrama específico para a representação das classes, chamado de diagrama de classes. Nesse diagrama, as classes são representadas por retângulos contendo nomes, atributos e operações, como ilustrado na figura 1. Os atributos representam propriedades associadas a cada um dos objetos que serão instanciados pela classe. Por exemplo, a classe “Cliente” pode ter um atributo chamado de “nome”. Outros atributos que essas classes podem ter são: “endereço”, “telefone” e “email”.

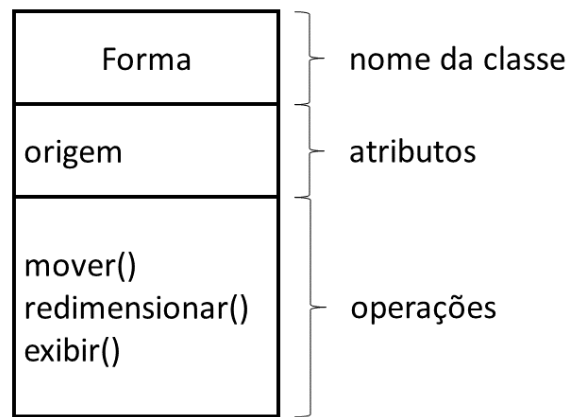


Figura 1. Representação de uma classe na UML.
BOOCH, RUMBAUGH e JACOBSON, 2006 (com adaptações).

Além de atributos, uma classe também apresenta comportamentos associados. Por exemplo, suponha um editor de imagens que tenha uma funcionalidade de desenho de formas geométricas, como losangos. Essa funcionalidade pode ser implementada por meio de uma classe, chamada de “Losango”, que apresenta uma série de comportamentos associados, como a movimentação do losango e a alteração do seu tamanho, entre outras. Essas operações são representadas no retângulo da classe, logo abaixo dos atributos.

As operações têm um nome e costumam terminar com parênteses “()”: elas são similares a “funções” e podem receber parâmetros. Os parâmetros são representados nos parênteses e separados por vírgulas (no caso da existência de mais de um parâmetro).

As diversas classes que compõem um programa podem estar relacionadas entre si e existem, essencialmente, três tipos de relacionamentos entre classes:

- dependência;
- generalização;
- associação.

Utilizamos uma relação de dependência quando queremos dizer que uma classe necessita de outra classe para o seu funcionamento. Por exemplo, suponhamos uma classe chamada “Cachorro” e uma classe chamada “Animal”. Podemos dizer que “Animal” representa uma generalização de “Cachorro”, uma vez que um cachorro é um animal, mas nem todo animal é um cachorro. Finalmente, uma associação é um relacionamento semântico entre dois elementos do modelo. Uma associação estabelece o motivo pelo qual duas classes relacionam-se e estabelece as regras que controlam esse relacionamento.

1.3. Casos de uso

Segundo Booch, Rumbaugh e Jacobson (2006), um caso de uso

especifica o comportamento de um sistema ou de parte de um sistema e é uma descrição de um conjunto de sequências de ações, incluindo variantes realizadas pelo sistema para produzir um resultado observável do valor de um ator.

Desenvolvedores de software precisam compreender as necessidades dos usuários e comunicá-las, sem precisar entrar em detalhes da implementação. A elaboração de casos de uso facilita esse processo de comunicação entre os diversos envolvidos em um projeto de software.

2. Análise das alternativas e das afirmativas

Questão 11

A - Alternativa correta.

JUSTIFICATIVA. O trecho de diagrama de classes apresentado está correto. Nele, a classe Pessoa é uma generalização da classe Funcionário.

B - Alternativa incorreta.

JUSTIFICATIVA. O diagrama está incorreto, pois as classes Pessoa e Funcionário estão representadas com relacionamento de associação. Na realidade, essas duas classes deveriam apresentar relacionamento de generalização.

C - Alternativa incorreta.

JUSTIFICATIVA. O construtor da classe Pessoa está representado de forma errada. Deveria ser: +Pessoa(rg:int, nome:String), e não +Funcionário(re:int, nome:String).

D - Alternativa incorreta.

JUSTIFICATIVA. Falta um método na classe Funcionário.

E - Alternativa incorreta.

JUSTIFICATIVA. A classe Pessoa do diagrama não apresenta o seu construtor.

Alternativa correta: A.

Questão 12

I - Afirmativa correta.

JUSTIFICATIVA. A generalização nos casos de uso tem papel similar à generalização nos diagramas de classes.

II - Afirmativa incorreta.

JUSTIFICATIVA. Como é o caso de uso-base que incorpora o incluído, o relacionamento incluído não pode ser isolado. Além disso, esse relacionamento tem seu comportamento incluído (por isso o nome) e, também, não é o caso de uso-base.

III - Afirmativa incorreta.

JUSTIFICATIVA. A descrição não corresponde a um relacionamento de extensão, pois o caso de uso-base pode permanecer isolado.

IV - Afirmativa correta.

JUSTIFICATIVA. O relacionamento de extensão é utilizado justamente para a modelagem de comportamentos opcionais, que podem ou não ocorrer em determinadas condições.

Alternativa correta: B.

3. Indicações bibliográficas

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.; *UML: Guia do usuário*. Rio de Janeiro: Campus, 2006.
- PENDER, T. *UML: a Bíblia*. Rio de Janeiro: Campus, 2004.

ÍNDICE REMISSIVO

Questão 1	Relação binária e relação de equivalência. Classe de equivalência. Partição.
Questão 2	Probabilidade.
Questão 3	Máquina de Turing.
Questão 4	Gramática: símbolos e cadeias.
Questão 5	Escalonamento. Conjuntos e intervalos disjuntos. Indução e proposições.
Questão 6	Diagrama de Venn.
Questão 7	Protocolos TCP, IP, UDP, ICMP e DHCP
Questão 8	Protocolos ARP, TCP, IP, UDP, HTTP e DNS.
Questão 9	Portas lógicas. Funções lógicas. Tabelas verdade e análise de circuitos lógicos.
Questão 10	Sistema operacional. Compiladores e chamadas de sistemas. CPU e registradores.
Questão 11	UML (Unified Modeling Language). Diagrama de classes. Casos de uso.
Questão 12	UML (Unified Modeling Language). Diagrama de classes. Casos de uso.