Tópico 03

Introdução à programação de computadores

Lógica proposicional

1. Introdução

Aluna(o), no mundo computacional, temos de lidar com muitas informações, e cada dia mais é necessário desenvolver códigos rápidos, bancos de dados enxutos e associações que sejam úteis.

Para tal, muitas técnicas são usadas, dentre elas, utilizamos as de análises de implicação e equivalência lógica em nossos algoritmos e na criação do banco de dados.

A álgebra relacionada a este tópico nos permitirá entender algumas operações matemática computacionais.

2. Implicação Lógica

Dadas as proposições compostas P (p, q, r, ...) e Q. (p, q, r, ...), diz-se que ocorre uma implicação lógica entre P e Q, quando a proposição condicional P → Q é uma tautologia.
Indica-se que a proposição P (p, q, r, ...) implica a proposição Q(p, q, r, ...) com a notação a seguir:

$$P\ (p, q, r, \ldots) \Rightarrow Q\ (p, q, r, \ldots)$$

Resumidamente, podemos dizer que P → Q quando, em suas respectivas tabelas verdade, não aparece **V** na última coluna de P e **F** na última coluna de Q, com **V** e **F** em uma mesma linha. Isto é, não ocorre P e Q com valores lógicos simultâneos respectivamente **V** e **F** (ALENCAR, 2000).

A implicação lógica entre proposições utiliza-se das propriedades seguintes:

➔ Reflexiva (**R**): $P(p,q,r,\ldots) \Rightarrow P(p,q,r,\ldots)$

➔ Transitiva (**T**): Se $P(p,q,r,\ldots) \Rightarrow Q(p,q,r,\ldots)$, e

$Q(p,q,r,\ldots) \Rightarrow R(p,q,r,\ldots)$, então,

$P(p,q,r,\ldots) \Rightarrow R(p,q,r,\ldots)$

2.1 Regras de implicação lógica (regras de inferência)

- Adição disjuntiva:

p ⇒ p V q

p	q	p Λ q	p V q	p ↔ q
V	V	V	V	V
V	F	F	V	F
F	V	F	V	F
F	F	F	F	V

A proposição p Λ q é verdadeira (V) somente na linha 1, e, nessa linha, a proposição p V q também é verdadeira (V). Logo, subsiste a implicação lógica p ⇒ p V q.

- Simplificação conjuntiva:

p Λ q ⇒ p

p	q	p Λ q	p V q	p ↔ q
V	V	V	V	V
V	F	F	V	F
F	V	F	V	F
F	F	F	F	V

A proposição p Λ q é verdadeira (V) somente na linha 1, e, nessa linha, a proposição p ↔ q também é verdadeira (V). Logo, subsiste a implicação lógica p Λ q → p.

- Silogismo disjuntivo:

(p V q) Λ ~p ⇒ q

p	q	p V q	~p	(p V q) Λ ~p ⇒ q
V	V	V	F	F
V	F	V	F	F
F	V	V	V	V
F	F	F	V	F

A proposição (p V q) Λ ~p é verdadeira (V) somente na linha 3, e, nessa linha, a proposição também é verdadeira (V). Logo, subsiste a implicação lógica (p V q) Λ ~p ⇒ q.

1. Modus ponens:

(p → q) Λ p ⇒ q

p	q	(p → q)	(p → q) Λ p
V	V	V	V
V	F	F	F
F	V	V	F
F	F	V	F

A proposição (p → q) Λ p é verdadeira (V) somente na linha 1, e, nessa linha, a proposição também é verdadeira (V). Logo, subsiste a implicação lógica (p → q) Λ p ⇒ q.

- Regra modus tollens:

$(p \rightarrow q) \wedge \sim q \Rightarrow \sim p$

p	q	(p → q)	~q	(p → q) Λ ~q	~p
V	V	V	F	F	F
V	F	F	V	F	F
F	V	V	F	F	V
F	F	V	V	V	V

A proposição $(p \rightarrow q) \wedge \sim q$ é verdadeira (V) somente na linha 4, e, nessa linha, a proposição também é verdadeira (V). Logo, subsiste a implicação lógica $(p \rightarrow q) \wedge \sim q \rightarrow \sim p$

Agora, vamos relembrar o conceito de tautologia? Se uma proposição composta é SEMPRE verdadeira (V), então teremos uma tautologia. Assim, podemos afirmar que, em uma tautologia, as proposições compostas serão logicamente verdadeiras (V). Então, vamos estudar as tautologias e implicações lógicas.

> *TEOREMA: a proposição P(p, q, r, ...)* ***implica*** *a proposição Q(p, q, r, ...) isto é*
> *P(p, q, r, ...) ⇒ Q(p, q, r, ...)*
>
> *Se, e somente se, a condicional*
> *P(p, q, r, ...) → Q(p, q, r, ...)*
>
> *é* ***tautológica****.*
> *(ALENCAR, 2000)*

Nos estudos de lógica, é importantíssimo atentar-se para as simbologias e termos utilizados nas mais diversas literaturas. Por exemplo:
Sejam H e G duas fórmulas, então,
{{H implica em G} e {H é tautológica}} → {G é tautológica}

(SOUZA, 2008).

$(p \rightarrow q) \land (q \rightarrow r) \Rightarrow p \rightarrow r$

p	q	r	(p	→	q)	Λ	(q	→	r)	→	(p	→	r)
V	V	V	V	V	V	V	V	V	V	V	V	V	V
V	V	F	V	V	V	F	V	F	F	V	V	F	F
V	F	V	V	F	F	F	F	V	V	V	V	V	V
V	F	F	V	F	F	F	F	V	F	V	V	F	F
F	V	V	F	V	V	V	V	V	V	V	F	V	V
F	V	F	F	V	V	F	V	F	F	V	F	V	F
F	F	V	F	V	F	V	F	V	V	V	F	V	V
F	F	F	F	V	F	V	F	V	F	V	F	V	F

A condicional $(p \rightarrow q) \land (q \rightarrow r) \rightarrow (p \rightarrow r)$ é tautológica, pois, como podemos ver, o resultado da sua tabela verdade é somente verdadeiro (V). Logo, subsiste a implicação lógica $(p \rightarrow q) \land (q \rightarrow r) \Rightarrow p \rightarrow r$.

- Princípio da inconsistência:

$p \land \sim p \Rightarrow q$

p	q	~p	p Λ ~p	p Λ ~p → q
V	V	F	F	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	V

A condicional $p \land \sim p \rightarrow q$ é tautológica, pois, como podemos ver, o resultado da sua tabela verdade é somente verdadeiro (V). Portanto, subsiste a implicação lógica $p \land \sim p \Rightarrow q$, em que de uma contradição $p \land \sim p$ se deduz qualquer proposição q.

O símbolo → representa operação lógica, p.ex.: $p \rightarrow r$, a qual lemos como: se então r.

O símbolo ⇒ representa uma relação, p.ex.: $P(p, q, r, \ldots) \Rightarrow Q(p, q, r, \ldots)$. Neste caso, lemos: P implica em Q.

3. Equivalência lógica

Dizemos que duas proposições $P(p, q, r, \ldots)$ e $Q(p, q, r, \ldots)$ são logicamente equivalentes (ou simplesmente equivalentes) quando os resultados de suas tabelas verdade são idênticos (ALENCAR, 2000).
Indica-se que a proposição $P(p, q, r, \ldots)$ é equivalente à proposição $Q(p, q, r, \ldots)$, com a notação seguinte:

$$P(p, q, r, \ldots) \Leftrightarrow Q(p, q, r, \ldots)$$

Diferentemente do símbolo de equivalência (⇔), que representa uma relação, o símbolo ↔ representa a operação bicondicional, p.ex: , a qual lemos como se e somente se.

A equivalência lógica entre proposições utiliza-se das propriedades:

Regra de equivalência lógica

- Dupla negação:

~~p ⇔ p

p	~p	~~p
V	F	V
F	V	F

As proposições ~~p e p são equivalentes. Portanto, a negação da negação (ou simplesmente dupla negação) equivale à afirmação.

- Absorção:

$p \rightarrow p \wedge q \Leftrightarrow p \rightarrow q$

p	q	p ∧ q	p → p ∧ q	p → q
V	V	V	V	V
V	F	F	F	F
F	V	F	V	V
F	F	F	V	V

As condicionais $p \rightarrow p \wedge q$ e $p \rightarrow q$ possuem tabelas verdades idênticas, então, por consequência, são equivalentes. Logo, subsiste a equivalência lógica $p \rightarrow p \wedge q \Rightarrow p \rightarrow q$.

Mais uma vez, não poderíamos deixar de tratar a tautologia. Reforçando, se uma proposição composta é SEMPRE verdadeira (V), então, teremos uma tautologia. Assim, podemos afirmar que, em uma tautologia, as proposições compostas serão logicamente verdadeiras (V). Então, vamos analisar as tautologias e a equivalência lógica.

> *TEOREMA: A proposição P(p, q, r, ...) é **equivalente** a proposição Q(p, q, r, ...) isto é:*
> *$P(p, q, r, ...) \Rightarrow Q(p, q, r, ...)$*
>
> *Se, e somente se, a bicondicional:*
> *$P(p, q, r, ...) \rightarrow Q(p, q, r, ...)$*
>
> *for **tautológica**.*
> *(ALENCAR, 2000)*

Demonstração por absurdo:

p Λ ~q → c ⇔ p → q

p	q	(p	Λ	~q)	→	c)	↔	(p	→	q)
V	V	V	F	F	V	F	V	V	V	V
V	F	V	V	V	F	F	V	V	F	F
F	V	F	F	F	V	F	V	F	V	V
F	F	F	F	V	V	F	V	F	V	F

A bicondicional (p Λ ~q → c) ↔ (p → q) na qual c é uma proposição de valor lógico falso (F), é tautológica, pois, como podemos ver, o resultado da sua tabela verdade é somente verdadeiro (V). Logo, subsiste a implicação lógica p → p Λ q ⇔ p → q

- Regra de exportação-importação

p Λ q → r ⇔ p → (q → r)

(p	Λ	q	→	r)	↔	(p	→	(q	→	r))
V	V	V	V	V	V	V	V	V	V	V
V	V	V	F	F	V	V	F	V	F	F
V	F	F	V	V	V	V	V	F	V	V
V	F	F	V	F	V	V	V	F	V	F
F	F	V	V	V	V	F	V	V	V	V
F	F	V	V	F	V	F	V	V	F	F
F	F	F	V	V	V	F	V	F	V	V
F	F	F	V	F	V	F	V	F	V	F

A bicondicional (p Λ q → r) ↔ (p → (q → r)) é tautológica, pois, como podemos ver, o resultado da sua tabela verdade é somente verdadeiro (V). Logo, subsiste a implicação lógic $p \land q \to r \Leftrightarrow p \to (q \to r))$

Dada a condicional $(p \to q)$ chamam-se **proposições associadas** a $(p \to q)$ as seguintes condicionais:

1. Proposição recíproca de $p \to q : q \to p$
2. Proposição contrária de $p \to q : \sim p \to \sim q$
3. Proposição contrapositiva de $p \to q : \sim q \to \sim p$

Negação conjunta.

$p \downarrow q \Leftrightarrow \sim p \land \sim q$

p	q	p ↓ q
V	V	F
V	F	F
F	V	F
F	F	V

$p \downarrow q \sim p \land \sim q$ é verdadeira somente no caso em que *p* e *q* são ambas falsas. Lemos a proposição como "não p e não q"

- **Negação disjunta**

$p \uparrow q \Leftrightarrow \sim p \lor \sim q$

p	q	p ↑ q
V	V	F

p	q	p ↑ q
V	F	V
F	V	V
F	F	V

A proposição ~p V ~q falsa somente no caso em que *p* e *q* são ambas verdadeiras. Lemos a proposição como "não *p* **ou** não *q*"

Os símbolos ↑ e ↓ são chamados "conectivos de SCHEFFER".

4. Aplicar a implicação e equivalência lógica em problemas com linguagem natural

Conectando o pensamento.

Agora, vamos buscar aplicações práticas para os conceitos da implicação e da equivalência lógica e buscar, por fim, entender a importância delas para a programação de computadores.

Ana viajando.

Vamos analisar a sentença a seguir:

Ana viajar de carro e de avião implica Ana viajar de carro ou de avião.

Primeiro vamos dividir em proposições simples e conectivos:

Ana viajar de carro/ e/ de avião/ implica / Ana viajar de carro/ ou/ de avião.

Agora vamos usar as notações para descrever a sentença:

- p: Ana viajar de carro
- Λ: e
- q: Ana viajar de avião
- ⇒: implica
- p: Ana viajar de carro
- V: ou
- q: Ana viajar de avião

Assim p Λ q ⇒ p V q. Vamos analisar a partir da tabela da verdade.

Verificação da implicação lógica p Λ q ⇒ p V q

p	q	p Λ q	p V q	p Λ q → p V q
V	V	V	V	V
V	F	F	V	V
F	V	F	V	V
F	F	F	F	V

Repare que a condicional da última coluna é tautológica e portando podemos dizer que p Λ q implica p V q ou podemos escrever p Λ q ⇒ p V q

Agora que vamos tentar repetir o processo usando um exemplo de equivalência lógica:

Ana viajar de carro se e somente se viajar de avião equivale a Ana viajar de carro se viajar de avião, e viajar de avião se viajar de carro.

- Ana viajar de carro/ se e somente se/ viajar de avião/ equivale a/ Ana viajar de carro/ se/ viajar de avião/ e/ viajar de avião/ se/ viajar de carro
- p : Ana viajar de carro
- ↔: se e somente se
- q : Ana viajar de avião
- ⇔: equivale
- p : Ana viajar de carro
- → : se
- q : Ana viajar de avião
- **Λ** : e

- q : Ana viajar de avião
- $\rightarrow$: se
- p : Ana viajar de carro

Então, temos $p \leftrightarrow q \Leftrightarrow p \rightarrow q \land q \rightarrow p$. Vamos analisar na tabela da verdade.

Verificação da equivalência lógica $p \leftrightarrow q \Leftrightarrow p \rightarrow q \land q \rightarrow p$.

p	q	$p \rightarrow q$	$q \rightarrow p$	$p \rightarrow q \land q \rightarrow p$	$p \leftrightarrow q$	$p \leftrightarrow q \Leftrightarrow p \rightarrow q \land q \rightarrow p$
V	V	V	V	V	V	V
V	F	F	V	F	F	V
F	V	V	F	F	F	V
F	F	V	V	V	V	V

Repare que a bicondicional da última coluna é tautológica, portanto, podemos dizer que $p \leftrightarrow q$ equivale a $p \rightarrow q \land q \rightarrow p$, ou podemos escrever $p \leftrightarrow q \Leftrightarrow p \rightarrow q \land q \rightarrow p$. Assim, notamos que faz sentido que chamemos as duas condicionais de duas proposições de “bicondicional”, e que o símbolo seja uma seta de dois sentidos, visto que as duas condicionais que a formam são duas setas simples para cada direção.

Uma vez entendido o passo a passo para transformar a linguagem natural em equivalência ou implicação lógica, podemos também usar esses conceitos para simplificar sentenças nos nossos códigos ou banco de dados, e até mesmo para corrigir a nossa fala no dia a dia. Vamos aos exemplos!

Não vi ninguém equivale a vi alguém.

- p : vi alguém
- $\sim p$: não vi alguém
- $\sim(\sim p)$: não vi ninguém

- ~(~p) ⇔ p

Verificação da equivalência lógica ~(~p) ⇔ p

p	~p	~(~p)
V	F	V
F	V	F

Podemos perceber que ~(~p) ⇔ p. Portanto, não faz sentido lógico dizer que **não viu ninguém** quando nenhuma pessoa foi vista, pois **não ver ninguém** é o mesmo que **ver alguém**, já que é uma dupla negação e, por isso, há uma equivalência lógica.

> *"A quantidade de registros na base de dados do cartão nacional de saúde no brasil, em 23.06.2016, é de 219.903.815 (duzentos e dezenove milhões, novecentos e três mil e oitocentos e quinze) usuários do sus".*
> *(GOVERNO FEDERAL, Número atualizado de usuários do SUS, consultaesic.cgu.gov.br).*

Imagine agora que você desenvolve códigos de otimização para o Sistema Único de Saúde (SUS), que possui cerca de 200 milhões de pessoas inscritas. Estamos, portanto, falando de um caso de big data.
Então, você encontra um padrão de implicação lógica entre um(a) paciente ser diagnosticado(a) com transtorno de ansiedade e a(o) médica(o) receitar um tratamento de acupuntura.
Para um caso de big data, faz toda diferença reservar vagas automaticamente, sempre que alguém recebe o diagnóstico de transtorno de ansiedade, para não superlotar o sistema, além de acelerar o atendimento. A implicação lógica aqui funciona como um algoritmo de otimização.
Constantemente, casos de equivalência lógica são usados para diminuir bancos de dados, uma vez que uma coluna inteira pode

ser “retirada” do banco, já que ela sempre se equivale à outra. Imagine o espaço que ocuparia uma coluna em um banco de dados com 200 milhões de linhas (pessoas inscritas no SUS).

5. Conclusão

Este tópico procurou mostrar alguns conceitos de equivalência e implicação lógicas. Buscamos aprender conceitos e regras sobre essas duas ferramentas. A partir desses conceitos, alcançamos a habilidade de identificar aspectos algébricos importantes, para fazer operações utilizando tais ferramentas.

Assim, procuramos fundamentar as bases da compreensão de um tópico importante, que será utilizado frequentemente daqui para frente em todas linguagens de programação, especialmente naquelas que tratam de big data e machine learning.

Como conclusão de uma sequência de lógica, raciocínio lógico e linguagem natural, apresentamos, aqui, um vídeo para vermos como evoluiu todo nosso pensamento crítico. Boa diversão, queridos alunos!

6. Referências

ALENCAR FILHO, Edgard de. Iniciação à lógica matemática. São Paulo: Nobel, 2000 e edições anteriores.

BRASIL. Governo Federal. consultaesic.cgu.gov.bR, 2016. Número atualizado de usuários do SUS. Disponível em: http://www.consultaesic.cgu.gov.br/busca/dados/Lists/Pedido/Item/displayifs.aspx?List=0c839f31%2D47d7%2D4485%2Dab65%2Dab0cee9cf8fe&ID=486493&Web=88cc5f44%2D8cfe%2D4964%2D8ff4%2D376b5ebb3bef.

SOUZA, João de Nunes. Lógica para ciência da computação: uma introdução concisa. Rio de Janeiro: Elsevier, 2008.

Parabéns, esta aula foi concluída!

O que achou do conteúdo estudado?

Péssimo Ruim Normal Bom Excelente

Deixe aqui seu comentário

Mínimo de caracteres: 0/150

Enviar