Universidade de São Paulo
Escola de Artes, Ciências e Humanidades

ACH2013 – Matemática Discreta – $2^{\underline{o}}$ sem. 2024
Professor: José Ricardo G. Mendonça

$1^{\underline{a}}$ Lista de Exercícios – Lógica e Cálculo Proposicional – 1 out. 2024

‘Contrariwise,’ continued Tweedledee, ‘if it was so, it might be;
and if it were so, it would be; but as it isn’t, it ain’t. That’s logic.’
Lewis Carroll, in *Through the Looking-Glass, and What Alice Found There* (1872)

Problemas

I. Lógica elementar

1. Sejam as declarações $p \equiv$ “Eva vai sair para uma caminhada”, $q \equiv$ “o céu está estrelado” e $r \equiv$ “está fazendo frio”. Construa declarações em português para as seguintes proposições lógicas:

$$(a)\ (q \wedge \neg r) \rightarrow p; \qquad (b)\ q \rightarrow (\neg r \rightarrow p); \qquad (c)\ \neg(p \leftrightarrow (q \vee r)).$$

   Agora construa as proposições lógicas correspondendes às seguintes declarações em português:

   (d) Eva vai fazer uma caminhada somente se o céu estiver estrelado;

   (e) Se estiver fazendo frio e o céu não estiver estrelado Eva não vai sair para uma caminhada;

   (f) Está fazendo frio mas Eva vai sair para fazer uma caminhada assim mesmo.

2. Mostre que a proposição $p \rightarrow q$ é equivalente à proposição $\neg p \vee q$.

3. Sejam p uma declaração, q uma tautologia e r uma contradição. Mostre que: (a) $p \vee q$ é uma tautologia; (b) $p \wedge r$ é uma contradição.

4. Demonstre as seguintes proposições:

   (a) $p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r)$ (distributiva de $\wedge$ sobre $\vee$);

   (b) $p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$ (distributiva de $\vee$ sobre $\wedge$).

5. Demonstre a segunda lei de De Morgan, $\neg(p \vee q) \equiv \neg p \wedge \neg q$.

6. Estenda as leis de De Morgan para n proposições, isto é, mostre que

(a) $\neg(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \equiv \neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n$;

(b) $\neg(p_1 \vee p_2 \vee \cdots \vee p_n) \equiv \neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n$.

7. Construa as tabelas verdade para as seguintes proposições:

$$(a)\ (p \to q) \vee \neg p; \qquad (b)\ (p \to (q \to r)) \to ((p \to q) \to (p \to r)).$$

8. Mostre que $p \vee \neg p$ é uma tautologia e que $p \wedge \neg p$ é uma contradição. A tautologia $p \vee \neg p \equiv T$, quando expressa na forma $\vdash (p \vee \neg p)$, é conhecida como ***princípio do terceiro excluído*** (em latim, ***tertium non datur***): ou p é verdade ou $\neg p$ é verdade e não há uma terceira possibilidade. Já a contradição $p \wedge \neg p \equiv F$, quando expressa na forma $\vdash (\neg(p \wedge \neg p))$, é conhecido como ***princípio da não-contradição***: é verdade que a proposição p e $\neg p$ é falsa, ou, alternativamente, a proposição de que $p \wedge \neg p$ é verdade é falsa. Os princípios do terceiro excluído e da não-contradição são obteníveis um pela negação do outro, sendo, portanto, essencialmente equivalentes.[a]

9. Mostre que as proposições $p \to (q \to r)$ e $(p \to q) \to r$ não são equivalentes. A proposição $p \to q \to r$ é, portanto, intrinsecamente ambígua.

10. Seja $p \oplus q$ a operação lógica de disjunção exclusiva definida como "p ou q mas não ambos". Escreva $p \oplus q$ em termos dos conectivos lógicos $\wedge$, $\vee$ e $\neg$ e construa sua tabela verdade. Outras notações para $p \oplus q$ são $p \veebar q$ e p **xor** q.

11. A técnica de ***reductio ad absurdum***, ou redução ao absurdo, consiste em supor inicialmente que vale o contrário do que se deseja verificar e, analisando as expressões envolvidas, sejam elas lógicas ou matemáticas, procurar por inconsistências de qualquer natureza causadas por isso. Usando a técnica de redução ao absurdo determine se cada uma das seguintes proposições é uma tautologia ou não — isto é, suponha inicialmente que cada uma delas é falsa e procure por inconsistências nessa suposição:

$$(a)\ (p \leftrightarrow (\neg q \vee r)) \to (\neg p \to q); \qquad (b)\ (p \to (q \vee r)) \vee (p \to q).$$

A técnica de redução ao absurdo pode ser relacionada ao *modus tollendo tollens*, argumento segundo o qual $p \to q,\ \neg q\ \vdash\ \neg p$ (v. exercício II.4). Em linguagem natural esse tipo de argumento pode facilmente ser confundido com sarcasmo — dê um exemplo.

---

[a] Bertrand Russell declara em *The Problems of Philosophy* (New York: Henry Holt & Co., 1912), Capítulo VII, "On our knowledge of general principles", que os princípios lógicos da identidade ("whatever is, is"), da não-contradição ("nothing can both be and not be") e do terceiro excluído ("everything must either be or not be") são princípios autoevidentes que devem ser garantidos antes que qualquer argumento ou demonstração se torne possível, embora a escolha desses princípios e não de outros igualmente fundamentais e autoevidentes seja arbitrária.

12. É possível definir alguns conectivos binários em termos de outros conectivos binários. Por exemplo, $\vee$ é definível a partir de $\neg$ e $\wedge$, pois a proposição $p \vee q$ é equivalente à proposição $\neg(\neg p \wedge \neg q)$ pela simples aplicação das leis de De Morgan. Mostre que

    (*a*) $\wedge$ é definível a partir de $\neg$ e $\vee$;

    (*b*) $\rightarrow$ é definível a partir de $\neg$ e $\vee$;

    (*c*) $\rightarrow$ é definível a partir de $\neg$ e $\wedge$;

    (*d*) $\wedge$ é definível a partir de $\neg$ e $\rightarrow$;

    (*e*) $\vee$ é definível a partir de $\neg$ e $\rightarrow$.

13. Um conectivo lógico binário muito útil é a ***barra de Sheffer***, $p \mid q$, introduzido pelo logicista norte-americano (nascido ucraniano) Henry M. Sheffer (1882–1964).[b] Outras notações para a barra de Sheffer são $p \uparrow q$, que é a notação que vamos usar, e p **nand** q. A proposição $p \uparrow q$ é equivalente a $\neg(p \wedge q)$, que só é verdade se p e q não forem verdade simultaneamente — lê-se "p e q são incompatíveis" ou ainda "pelo menos uma das duas proposições p ou q é falsa". Sheffer provou que todos os conectivos lógicos que vimos até agora, inclusive o operador de negação, são definíveis a partir de $\uparrow$. Esse fato é conhecido como ***completeza funcional*** da barra de Sheffer e possui enormes implicações para o desenho de circuitos digitais. De fato, pode-se mostrar que qualquer um dos seguintes conjuntos de conectores lógicos é funcionalmente completo: $\{\wedge, \vee, \neg\}$, $\{\wedge, \neg\}$, $\{\vee, \neg\}$, $\{\uparrow\}$ e $\{\downarrow\}$, onde $p \downarrow q \equiv \neg(p \vee q)$, também denotado por p **nor** q, é conhecido como ***seta de Pierce*** (ou como ***adaga de Quine*** quando denotado na forma $p \dagger q$).

    (*a*) Construa as tabelas verdade de $p \uparrow q$ e $\neg p \uparrow \neg q$;

    (*b*) Reescreva $\neg p$, $p \wedge q$ e $p \vee q$ em termos de $p \uparrow q$;

    (*c*) Reescreva $p \rightarrow q$ e $p \leftrightarrow q$ em termos de $p \uparrow q$.

14. Podemos evitar tanto ambiguidades quanto excesso de parênteses nas proposições compostas se concordarmos com a seguinte ***ordem de precedência*** para os conectivos lógicos:

Precedência	Conectivo
1	$\neg$
2	$\wedge$
3	$\vee$
4	$\rightarrow$
5	$\leftrightarrow$

[b] H. M. Sheffer, "A set of five independent postulates for Boolean algebras, with application to logical constants", *Transactions of the American Mathematical Society* **14**, 481–488 (1913). Os estudantes desta disciplina têm plenas condições de ler esse artigo.

Esta ordem de precedência não é única mas é a mais empregada, porque privilegia os operadores unários ($\neg$) sobre os conectivos binários ($\wedge$, $\vee$) que por sua vez possuem precedência sobre os conectivos de implicação ($\rightarrow$) e equivalência ($\leftrightarrow$), que na maior parte das vezes são usados ao final das operações do cálculo proposicional para avaliar as consequências e equivalências das proposições avaliadas.

(a) Mostre que a proposição composta $(p \leftrightarrow (((\neg q) \vee r) \rightarrow p))$ é equivalente à proposição $p \leftrightarrow \neg q \vee r \rightarrow p$ de maneira inequívoca. Um exemplo de proposição que *não* é inequívoca sem os parênteses é dada no exercício I.9.

Podemos verificar a equivalência acima colocando parênteses na proposição novamente de acordo com o seguinte procedimento, cujos passos devem ser repetidos até que a proposição inteira esteja entre parênteses:

(i) Encontre a ocorrência mais à esquerda do conectivo de maior precedência em toda a proposição que ainda não foi "processado";

(ii) Se o conectivo for $\neg$ e estiver precedendo uma proposição p, restauramos os parênteses em torno da proposição negada para obter $(\neg p)$;

(iii) Se um conectivo binário $\star$ ($\wedge$, $\vee$, $\rightarrow$ ou $\leftrightarrow$) for precedido por uma proposição p e seguido por uma proposição q, restauramos os parênteses para obter $(p \star q)$.

(b) Mostre que aplicando o procedimento acima à proposição do item (a) obtemos sucessivamente $p \leftrightarrow (\neg q) \vee r \rightarrow p$, $p \leftrightarrow ((\neg q) \vee r) \rightarrow p$, $p \leftrightarrow (((\neg q) \vee r) \rightarrow p)$ e $(p \leftrightarrow (((\neg q) \vee r) \rightarrow p))$, recuperando a proposição na sua forma inicial.

II. Argumentos

1. Mostre que os argumentos a seguir são todos falaciosos e dê exemplos usando linguagem ordinária para cada um deles:

   (a) $p \vee q, p \vdash \neg q$ (falácia da afirmação da disjunção);

   (b) $p \rightarrow q, q \vdash p$ (falácia da afirmação do consequente);

   (c) $p \rightarrow q, \neg p \vdash \neg q$ (falácia da negação do antecedente).

2. Traduza os seguintes argumentos em proposições lógicas e determine sua validade:

   (a) Se 7 é menor do que 4, então 7 não é um número primo; 7 não é menor do que 4; portanto 7 é um número primo;

   (b) Se dois lados de um triângulo são iguais, então seus ângulos opostos são iguais; dois lados de um triângulo não são iguais; portanto seus ângulos opostos não são iguais.

Repare que nesses exemplos a validade ou não dos argumentos nada tem a ver com a validade ou não das conclusões, e sim com a maneira como os argumentos foram construídos. Embora as conclusões acima consistam de fatos verdadeiros, os argumentos usados para chegar a essas conclusões são falaciosos.

3. A regra do ***modus ponendo ponens*** (que em latim significa mais ou menos "modo de afirmar afirmando"), normalmente abreviada simplesmente por ***modus ponens***, diz que em qualquer sistema lógico formal se uma proposição implica uma segunda proposição e a primeira proposição é verdadeira, então a segunda proposição também é verdadeira. O *modus ponens* é uma ferramenta útil na simplificação de demonstrações em cálculo proposicional. Mostre que essa regra corresponde à proposição $(p \to q) \land p \to q$.

4. Uma versão "negativa" do *modus ponendo ponens* é conhecida como ***modus tollendo tollens*** (que em latim significa mais ou menos "modo de negar negando"), normalmente abreviada simplesmente por ***modus tollens***, cuja expressão lógica é dada pelo argumento $p \to q,\ \neg q\ \vdash\ \neg p$. Determine a validade do *modus tollens* e dê exemplos desse tipo de argumentação usando linguagem ordinária.

5. Além do *modus ponendo ponens* e do *modus tollendo tollens*, existem ainda as versões "mistas" de argumentação conhecidas como ***modus ponendo tollens*** e ***modus tollendo ponens***, cujas expressões lógicas são dadas, respectivamente, por $\neg(p \land q),\ p\ \vdash\ \neg q$ e $p \lor q,\ \neg p\ \vdash\ q$. Determine a validade desses argumentos e dê exemplos para eles em linguagem ordinária — seja criativo!

6. Dado que $p \to q$, construa a tabela verdade para as proposições ***inversa*** $\neg p \to \neg q$, ***conversa*** $q \to p$ e ***contrapositiva*** $\neg q \to \neg p$. Verifique que $(\neg q \to \neg p) \equiv (p \to q)$ e que $(p \to q) \land (q \to p) \leftrightarrow (p \leftrightarrow q)$.

III. *Divertissement*: Notas históricas

A lógica formal se desenvolveu como uma disciplina independente antes mesmo do que a aritmética e a geometria. Aristóteles (384–322 BCE) construiu o primeiro sistema de lógica formal que conhecemos, atualmente denominado ***lógica aristotélica*** ou ***lógica categórica***. Em Aristóteles encontramos, pela primeira vez na história, uma idéia clara de regra lógica universalmente válida e o uso de sentenças que, além das variáveis, contêm apenas constantes lógicas. Seus escritos de lógica foram coligidos no *Organon*, conjunto de seis ensaios que teve enorme influência ao longo de mais de vinte séculos. Nas universidades européias medievais, por exemplo, as sete "artes liberais", disciplinas consideradas essenciais na educação de uma "pessoa livre", eram a gramática, a retórica e a lógica, que formavam o *trivium*, além da geometria, da aritmética, da música e da astronomia, que formavam o *quadrivium*. No prefácio à segunda edição (1787) de sua obra-prima *Crítica da Razão Pura*, o influente filófoso alemão

Immanuel Kant (1724–1804) chegou a declarar que desde Aristóteles "a lógica formal não conseguiu avançar um único passo e, portanto, parece consistir em todos os aspectos de um corpo fechado e completo de doutrina."[c] Essa aparente solidez e imutabilidade da lógica aristotélica desencorajou investigações mais aprofundadas da lógica até meados do século XIX. Contribuiu para esse estado das coisas o fato de que a maior parte dos estudiosos da lógica naquela época era formada por filósofos mais interessados em epistemologia e ontologia.

O sistema de Aristóteles é construído em torno da lógica de termos que identificam categorias de "coisas", concretas (como humanos ou minerais) ou abstratas (como estados de humor). Seu sistema, no entanto, é limitado em alguns aspectos importantes. Por exemplo, todo silogismo deve ter exatamente duas premissas, e as premissas e a conclusão de um silogismo devem ser estruturadas de acordo com regras muito restritivas. A lógica aristotélica não é capaz de acomodar um raciocínio obviamente válido como "a empregada ou o mordomo mataram Watson; se fosse a empregada, Watson teria sido envenenado; Watson não foi envenenado; então o mordomo matou Watson" porque há três premissas, não duas, e a primeira e a segunda premissas têm formas mais complexas do que podem ser acomodadas na lógica aristotélica.

Um dos primeiros defensores da lógica como disciplina matemática foi Gottfried W. Leibniz (1646–1716), filósofo e matemático saxão cujas ideias, no entanto, não tiveram grande influência no desenvolvimento do assunto — talvez porque ele imaginasse ser necessário primeiro criar um compêndio de todo o conhecimento humano! — até serem retomadas por George Boole (1815–1864) com a publicação de *An Investigation of the Laws of Thought, on which Are Founded the Mathematical Theories of Logic and Probabilities* (London: Macmillan, 1854). O verdadeiro renascimento da lógica como disciplina rigorosa se deu na virada do século XIX para o século XX, quando o desenvolvimento da geometria e da análise matemática deu origem à teoria dos conjuntos e com ela a descoberta de inúmeros paradoxos nos fundamentos da matemática. A investigação desses paradoxos levou, em última instância, à criação da teoria da computação pelas mãos de John von Neumann (1903–1957), Alonzo Church (1903–1995), Alan M. Turing (1912–1954) e Claude E. Shannon (1916–2001), depois de ter envolvido pensadores dos calibres de Charles S. Peirce (1839–1914), Georg F. L. P. Cantor (1845–1918), Friedrich L. Gottlob Frege (1848–1925), David Hilbert (1862–1943), Bertrand A. W. Russell (1872–1970) e Kurt F. Gödel (1906–1978), entre outros. O trabalho dessas pessoas revolucionou nossa visão de mundo de maneira profunda, com implicações em todas as áreas do conhecimento, da linguística e da

[c] A passagem original se lê: "Daß die Logik diesen sicheren Gang schon von den ältesten Zeiten her gegangen sei, läßt sich daraus ersehen, daß sie seit dem Aristoteles keinen Schritt rückwärts hat tun dürfen, wenn man ihr nicht etwa die Wegschaffung einiger entbehrlicher Subtilitäten, oder deutlichere Bestimmung des Vorgetragenen als Verbesserungen anrechnen will, welches aber mehr zur Eleganz, als zur Sicherheit der Wissenschaft gehört. Merkwürdig ist noch an ihr, daß sie auch bis jetzt keinen Schritt vorwärts hat tun können, und also allem Ansehen nach geschlossen und vollendet zu sein scheint." (I. Kant, *Kritik der reinen Vernunft*, Zweite hin und wieder verbesserte Auflage (Riga: J. F. Hartknoch, 1787), Vorrede zur zweite Auflage, VIII).

psicologia às teorias para a formação do universo.

No capítulo 1 de seu livro-texto *Proof and Disproof in Formal Logic: An Introduction for Programmers* (New York: Oxford University Press, 2005), intitulado "A rough history of logic", Richard Bornat resume aproximadamente dois mil anos de história com uma divertida analogia futebolística. As seções daquele capítulo são: 1.1 The Greeks invent the game; 1.2 Goals galore, but we're not watching; 1.3 Frege changes the rules; 1.4 Russell kicks Frege in the knee; 1.5 Gödel blows up the stadium; 1.6 Turing and Church play on; 1.7 The beautiful game. Uma revisão histórica abrangente do desenvolvimento da lógica formal clássica através de excertos dos textos originais é dada por Józef M. Bocheński, *History of Formal Logic* (Notre Dame: University of Notre Dame Press, 1961). Uma exposição acessível e empolgante do desenvolvimento da lógica a partir do século XVII até o advento da teoria da computação na primeira metade do século XX é dada por Martin Davis, *The Universal Computer: The Road from Leibniz to Turing*, 3a. ed. (Boca Raton: CRC, 2018), que recomendamos enfaticamente.

IV. *Divertissement*: Tabelas verdade e P versus NP

Para aplicar o método da tabela de verdade a uma fórmula bem formada envolvendo n proposições primitivas precisamos construir uma tabela de 2^n linhas. O problema é que, à medida que n aumenta, 2^n cresce exponencialmente. Por exemplo, supondo que possamos gerar um bilhão de linhas da tabela verdade por segundo com auxílio de um computador, precisaríamos de 2^{80} nanossegundos, isto é, de cerca de 38,3 milhões de anos para construir a tabela verdade completa de uma proposição com 80 proposições primitivas. Isso é 5 vezes mais tempo do que os 7,5 milhões de anos que o enorme supercomputador Deep Thought levou para calcular a resposta à questão fundamental da vida, do Universo e tudo mais no livro de Douglas Adams, *O Guia do Mochileiro das Galáxias*.

Existe um método mais rápido? É possível que exista algum algoritmo que, dado qualquer fórmula bem formada α envolvendo n proposições primitivas determine se α é ou não é uma tautologia em apenas An^k nanossegundos para alguma constante A e expoente k finito ou ainda alguma outra função de n que cresça mais lentamente do que uma exponencial. Por exemplo, se $A = 1000$ e $k = 5$, levaria apenas 55 minutos (à taxa hipotética de um bilhão de linhas por segundo) para construir a tabela verdade completa de uma fórmula bem formada envolvendo 80 proposições primitivas. A resposta a essa questão é desconhecida, mas é amplamente aceito que ela é negativa.

Nossa experiência cotidiana indica que é mais difícil resolver um problema do que verificar se uma possível solução é correta. Você consegue imaginar um mundo em que resolver qualquer problema não seja significativamente mais difícil do que verificar uma solução para ele? A negação da plausibilidade de tal mundo, em que "resolver" não é mais difícil do que "verificar", corresponde à afirmação de que "P $\neq$ NP", onde P representa tarefas que são eficientemente solucionáveis e NP representa tarefas para as quais soluções podem ser verificadas de maneira eficiente. A existência de um algoritmo que pudesse examinar a tabela verdade de uma proposição envolvendo n proposições primitivas em tempo polinomial corresponderia a uma resposta positiva (P = NP) para o problema conhecido como “P versus NP”, o problema mais proeminente na teoria da computação moderna. O leitor interessado em uma introdução didática porém rigorosa ao problema deve consultar o livro de Oded Goldreich, *Computational Complexity: A Conceptual Perspective* (Cambridge: Cambridge University Press, 2008).

⋆ — ⋆ — ⋆