A lógica e o *consortio daemoniorum*

Walter A. Carnielli
Marcelo E. Coniglio
IFCH e CLE - UNICAMP Campinas, SP
{carniell, coniglio}@cle.unicamp.br

Versão expandida da resenha publicada pelo primeiro autor em
Espiral- Revista Eletrônica de Divulgação Científica
http://www.eca.usp.br/nucleos/njr/espiral/

Resenha Crítica de "Lógica para Ciência da Computação Fundamentos de Linguagem, Semântica e Sistemas de Dedução"
João Nunes de Sousa
Editora Campus Ltda., 2002

Numa passagem famosa, em resposta aos seguidores dos céticos que sustentavam que o homem não poderia conhecer nada com certeza absoluta, referindo-se aos astrólogos (mas confundindo-os com os matemáticos), Santo Agostinho adverte que "o bom cristão deveria estar atento aos matemáticos e a todos aqueles que fazem profecias vazias. Existe o perigo de que os matemáticos tenham feito um pacto com o diabo para obscurecer o espírito e confinar o homem no inferno"[1].

Ainda que por razões diferentes, o *consortio daemoniorum* denunciado pelo Santo envolve de fato riscos: por um lado, a lógica simbólica ou matemática é uma componente fundamental do arsenal de todo estudante ou profissional da ciência da computação, porque é parte integrante do fundamento de qualquer programa de computador, coletivamente chamado *software*: sem lógica, em especial a lógica clássica de proposições e predicados, e mesmo as lógicas modais e outras ditas não clássicas, pode-se até usar, copiar e piratear mas não se pode, analisar, avaliar ou produzir *software*. Até hoje não se inventou nada melhor do que os conceitos da lógica para formalizar a especificação dos programas, a semântica das linguagens de programação, a análise de algoritmos, a teoria das bases de dados e a corretude dos programas, ou seja, a garantia de que tanto quanto possível os programas façam aquilo que foi encomendado.

Por outro lado, se não for usada com cuidado, a própria linguagem da lógica tende a dar um ar de falsa seriedade a asserções e posições que podem induzir ao erro. Algo disso acontece neste livro.

[1]Santo Agostinho, 354-430 AD De Genesi ad Litteram livro II, xviii, 37, Opera Omnia, 1836.

O livro tem virtudes-uma delas é ser o primeiro texto a contribuir para preencher uma lacuna na literatura brasileira, chamando a atenção dos proponentes de currículos acadêmicos para a importância da lógica para as ciências da computação. O livro tem também dezenas de exercícios originais, sem aquele ar de filme dublado onde o narrador tem que espichar as vogais para que a cadência com o original se mantenha.

Infelizmente, as virtudes de texto estão desbalanceadas em relação aos seus defeitos.

A sensação de se ler este livro, para quem está um pouco familiarizado com o tema, se parece muito com a de caçar semelhanças e discrepâncias naqueles joguinhos de sete erros: parecem corretos à primeira vista, mas quem procura encontra seus defeitos sutis. Contudo, se você é um estudante de ciências da computação com pretensões de fundamentar seus programas, as conseqüências podem ser menos divertidas.

O prefácio confessa que o livro pretende apenas apresentar os principais fundamentos da lógica clássica necessários aos estudantes de ciência(s) da computação e áreas correlatas, e que o texto pode ser utilizado em cursos de graduação e até de pós-graduação, e ainda por filósofos, matemáticos, físicos, engenheiros, advogados, etc.

Pode, mas não deve, não sem uma boa dose de cautela. O livro está tão crivado de erros conceituais, exemplos inconvenientes e definições descuidadas que fica difícil recomendá-lo a quem quer que seja.

Infelizmente, a estratégia de mercado de algumas editoras brasileiras faz vender como "aplicação à computação" muita coisa que só deveria ser publicada depois de uma cuidadosa revisão, a exemplo de certas editoras européias e norte-americanas que pagam cientistas competentes para ler textos de graduação antes de se aventurar a lançá-los no mercado.

A Editora Campus cometeu esse erro, e como professores não podemos deixar de lamentar e de criticar tal procedimento, na esperança que essa prática não pragueje o nascente setor brasileiro de textos acadêmicos. Imaginamos que a Elsevier Science, da qual a Campus faz parte, não cometeria o mesmo erro num texto em língua inglesa.

Escrever um livro acadêmico não é um mero ato de clonagem tipográfica: o autor tem obrigação de esclarecer conceitos e de utilizar no seu próprio texto os princípios fundadores da ciência que se pretende transmitir. O célebre princípio conhecido como "Navalha de Ockham", segundo o qual o lógico e filósofo inglês teria insistido sobre o perigo de se multiplicar conceitos sem necessidade ("*Entia non sunt multiplicanda praeter necessitatem*"), por exemplo, é notoriamente ignorado no texto em questão, em definições por vezes canhestras e repetitivas, e em explicações e chavões que mais confundem que esclarecem. Apontamos abaixo alguns dos principais pontos que servem de alerta ao uso deste livro:

1. Página 4. A definição de "alfabeto" lista os símbolos de verdade "true"

e “falso” (anglicismos desnecessários, já que são conhecidos pelos nomes latinos *verum* e *falsum*) como elementos estranhos aos constituintes da linguagem. Além do anglicismo, é também desnecessário introduzir uma categoria lingüística adicional, considerando que *verum* e *falsum* são apenas símbolos proposicionais zero-ários (constantes lógicas) distinguidos.

2. Página 5: “Esse raciocínio segue indefinidamente...”. O que significa num texto de lógica, afirmar que um raciocínio “segue” indefinidamente? Que o raciocínio não termina? Ou que o raciocínio pode ser repetido arbitrariamente?

3. Página 6: O que significa “Há conectivos proposicionais que não possuem relação de precedência”? O que significa “possuir” uma relação?

4. Página 13. Mesmo que o autor estabeleça no prefácio que sua intenção não inclui a abordagem da lógica de qualquer viés filosófico, isso não o isenta de cuidados mínimos. A distinção entre proposição e sentença não é tocada neste texto (no sentido em que sentenças declarativas distintas como “Chove” ou “Está chovendo” expressam a mesma proposição, que é formalizada pelas fórmulas da lógica proposicional, no caso em questão). O estudante deve ser informado que sentenças declarativas expressam proposições por meio de fórmulas; é duvidoso que “os significados das fórmulas sejam T e F” (de novo anglicismos desnecessários), mas sim que os *referentes* das sentenças declarativas (através das fórmulas que expressam as proposições nelas embutidas) é que são os valores de verdade. (Ver 1)

5. Página 14. A definição de “Interpretação de fórmulas” é redundante com a da página 13 .

6. Página 17. O que significa “Como qualquer argumento deve implicar em si mesmo, ...”? O que é um argumento, neste livro, e como um argumento “implica” em si mesmo?

7. Página 27. A confusão conceitual começa a ficar mais perigosa. Uma tabela na página 27 introduz as noções de “implica” e “equivale” (sic) entre fórmulas como uma categoria linguística adicional fora da linguagem-objeto ($\rightarrow$ e $\leftrightarrow$).

8. Página 31. A definição de “Satisfatibilidade de um conjunto de fórmulas” é redundante com a da página 27. Ademais, só se define tal noção para conjuntos finitos, o que escamoteia completamente a questão da compacidade na lógica.

9. Página 31.O que significa “Conjuntos de fórmulas satisfatíveis são importantes em Ciência da Computação”?

10. Página 31. O texto proclama que: "Os programas lógicos são conjuntos de fórmulas satisfatíveis. Isto significa que deve haver pelo menos uma interpretação que satisfaça o programa. Normalmente, essa interpretação é dada pelo programador, que certamente acredita e interpreta as fórmulas de seu programa como sendo verdadeiras". Uma afirmação destas, colocada num livro didático a respeito de lógica para estudantes de computação, é de extrema periculosidade: o estudante (leitor, em geral, inexperiente) fica com a sensação de que a utilização da lógica como linguagem de programação é algo vago e superficial, dando apenas um suporte para as intuições do programador, sem servir como ferramenta para verificar a correção dos programas. O texto não faz qualquer comentário sobre a semântica de Herbrand (conceito essencial para a teoria de programação lógica tratada no capítulo 15) que assegura que todo programa lógico, incluindo o programa vazio, tem um modelo (de Herbrand).

11. Página 33. A confusão conceitual atinge um ponto crítico: o operador $\Rightarrow$ é definido como o "se-então" da semântica (e não da metalinguagem). E ainda é ilustrado com um exemplo: "H é tautologia $\Rightarrow$ existe interpretação I ; I(H)= T". O símbolo ";" é usado no lugar do usual ":". Embora não se refira a $\Leftrightarrow$, presumivelmente junto com $\Leftrightarrow$ o "implica" da página 27 passa a compor então uma dupla noção semântica, ilustrada no exemplo da página 34:

    "H implica G $\Leftrightarrow$ (H $\rightarrow$ G) é tautologia".

12. Página 37. Há aqui uma outra confusão envolvendo variáveis e metavariáveis que afeta todo o texto. Por exemplo, "Suponha que H= P e G = Q. Observe que H e G não são equivalentes...". Porquê não seriam? Possivelmente porque H e G seriam metavariáveis (variáveis sobre fórmulas) e P e Q seriam fórmulas atômicas distintas, mas o leitor tem que usar aqui suas artes divinatórias.

13. Página 47 e seguintes. O Capítulo 4 estuda métodos para determinação da validade de fórmulas proposicionais; apresenta, ao lado do método das tabelas-verdade, o "método da árvore semântica" e o "método da negação ou absurdo" (entendido como redução ao absurdo). Se o texto usasse o rótulo de "aplicações em Ciência da Computação" mais do que como uma desculpa, teria obrigação de esclarecer nesse ponto a questão da complexidade dos métodos de determinação da satisfatibilidade e da validade, uma das mais (se não a mais) fundamentais em computação, ligada à conhecida questão $P \stackrel{?}{=} NP$. De toda forma, como apresentado aqui o "método da árvore semântica" e o "método da negação ou absurdo" são exatamente o mesmo método repetido de formas diferentes: o primeiro de forma esquemática, e o segundo em linguagem natural.

14. Página 63. Não é verdade que os termos "necessidade" e "suficiência" sejam "utilizados para denotar a implicação e a equivalência".

15. Página 73. O texto introduz um certo "Principio da Indução na lógica" (caso particular da indução aritmética ou finita) e afirma que este se reduz a "indução no comprimento de fórmulas", o que é incorreto, pois a indução em lógica é usada em relação a qualquer parâmetro numérico conveniente, como por exemplo indução no comprimento das demonstrações, indução no número de vezes que uma regra foi usada, etc. Sem dúvida, pelo próprio fato de a indução no comprimento de fórmulas constituir um caso particular da indução aritmética ou finita, o primeiro pode se demonstrado a partir do segundo, mas a demonstração esboçada no texto conclui erroneamente (página 73) que "P-log implica Pa, e o principio da indução na lógica está demonstrado".

16. Página 85. O conectivo *nor*, que é funcionalmente completo, sequer é mencionado na lista de conectivos completos. Ademais, é bastante interessante, mormente para "aplicações em Ciência da Computação", mostrar que *nand* e *nor* são os únicos conectivos (clássicos) funcionais completos.

17. Página 96, sistema Pa. O sistema axiomático para Pa que usa apenas $\vee$ e $\rightarrow$ como conectivos primitivos (note que o acrônimo Pa é usado de maneira totalmente distinta na página 73) é dado sem regras de dedução (presumivelmente, falta introduzir no sistema Pa uma regra tipo corte: $(H, \neg H \vee G/G)$

18. Página 96. Há aqui um problema dos mais graves. O texto salta do sistema Pa ao sistema com disjunção e implicação (que chamamos R) dado pelos seguintes axiomas(mais a regra de Modus Ponens):

    Ax1 $(H \vee H) \rightarrow H$

    Ax2 $H \rightarrow (G \vee H)$

    Ax3 $(H \rightarrow G) \rightarrow ((E \vee H) \rightarrow (G \vee E)$

    Esta axiomática é devida à célebre matemática e lógica polonesa Helena Rasiowa, que a publicou em 1949[2] como uma simplificação da parte proposicional dos sistemas fundadores de Russell e Whitehead em *Principia Mathematica* de 1910, e de Hilbert e Ackermann em seu *Grundzüge der theoretischen Logik* de 1928.

    O sistema Pa e o sistema R (cuja linguagem se baseia em $\vee$ e $\rightarrow$) são usados conjuntamente com finalidade duvidosa de simplificar demonstrações,

[2]Helena Rasiowa. Sur um certain systeme d´axiomes du calcul des propositions. *Norsk Matematisk Tidsskrift* **31**:1-3, 1949.

e a passagem de um a outro é feita esclarecendo-se que $H \to G$ "denota" $\neg H \vee G$ e que $H \to \mathit{falsum}$ "denota" $\neg H$. Contudo, dessa forma descuidada, o tratamento dado à lógica proposicional sofre de um mal incurável: é incompleto. De fato, falta algum axioma que esclareça o papel da constante *falsum*, como por exemplo $(H \to \mathit{falsum}) \to (H \to G)$ ou $(\mathit{falsum} \to H)$. Sem isso, ainda que a implicação $\to$ e a disjunção $\vee$ sejam interpretadas da maneira usual, *falsum* pode ser interpretado como "verdadeiro" numa interpretação semântica alternativa à definida na página 13. Em outras palavras, o sistema R tal como apresentado não consegue demonstrar $(\mathit{falsum} \to H)$. De maneira análoga, o sistema Pa é também essencialmente incompleto: não consegue demonstrar $(\neg \mathit{falsum} \vee H)$, ou $(\mathit{verum} \vee H)$. Para tanto, basta tomar uma semântica usual, mas tal que o valor de *falsum* seja sempre "verdadeiro", já que não há cláusulas axiomáticas que governem *falsum*. Da mesma forma, o sistema S é também incompleto da forma como apresentado: não se consegue demonstrar $(H \to H)$ (tal fato pode ser provado utilizando um bem-conhecido argumento lógico, definindo-se certas matrizes trivalentes que validam os axiomas e a regra Modus Ponens, mas falsificam esta sentença).

Vejamos em detalhes:

Se consideramos como modelos do sistema Pa as matrizes clássicas sobre $\mathbf{2} = \{0, 1\}$ onde 1 é o único valor distinguido, e homomorfismos (valorações) $h : For \to \mathbf{2}$ tais que $h(\bot) = 0$ (onde $\bot$ representa *falsum*) concluímos que o sistema Pa não é completo. De fato, $(\bot \to p_0)$ é uma tautologia, mas não é demonstrável em Pa: basta tomar a mesma semântica acima, mas agora considerando todas as valorações h tais que $h(\bot) = 1$. Esta semântica satisfaz todos os axiomas de Pa e preserva a validade através da regra (MP). Daí, se $\bot \to p_0$ fosse um teorema de Pa, então todas as valorações a nova semântica deveriam validá-lo. Contudo, as valorações h tais que $h(p_0) = 0$ não validam $(\bot \to p_0)$. Isso mostra que $(\bot \to p_0)$ não é demonstrável em Pa, e consequentemente Pa não pode ser um sistema completo com respeito à semântica clássica.

Ainda que acrescentássemos a Pa novos axiomas para a negação (definida) $\neg A := (A \to \bot)$ obtendo uma extensão (Pa*), não chegaríamos à completude:

(Neg1) $A \to (\neg A \to B)$

(Neg2) $\neg\neg A \to A$

De fato, (Pa*) é ainda incompleto com respeito à semântica clássica: para

mostrar este fato, considere as seguintes matrizes sobre $\mathbf{3} = \{0, \frac{1}{2}, 1\}$:

$\vee$	1	$^1/_2$	0
1	1	1	1
$^1/_2$	1	0	0
0	1	0	0

$\rightarrow$	1	$^1/_2$	0
1	1	0	0
$^1/_2$	1	$^1/_2$	1
0	1	1	1

onde 1 é o único valor distinguido. Considere a semântica definida por todas as valorações $h : For \rightarrow \mathbf{3}$ tais que $h(\bot) = 0$. Adicione-se a interpretação para a negação $\neg$ dada por:

	$\neg$
1	0
$^1/_2$	1
0	1

Pode-se mostrar então que a semântica acima valida todos os axiomas de (Pa*), e preserva validade pela regra (MP). Portanto, todo teorema A de (Pa*) é valido com respeito a esta semântica. Contudo, a fórmula $(p_0 \rightarrow p_0)$ (que é uma tautologia clássica) não é validada por esta semântica. Basta tomar a valoração h tal que $h(p_0) = \frac{1}{2}$. Consequentemente nem mesmo a extensão (Pa*) de Pa chega a ser completa com respeito à semântica clássica de valorações binárias.

19. Página 101. Na Proposição 1, o símbolo $\Rightarrow$ relacionando conjuntos não tem nenhum sentido.

20. Página 102. A Proposição 3 (Regra de Substituição) não está demonstrada.

21. Página 103. O exemplo a respeito da regra de substituição está duplamente errado. Não é verdade (obviamente) que "$Q \in \{P, P \rightarrow Q\}$", e não é verdade que "Q não pode ser substituído". A explicação é que uma substituição deve ser global, o que seria claro se este conceito tivesse sido definido.

22. Páginas 104 a 106. A presença do "conjunto de hipóteses β" nas Proposições 6 a 13 não faz sentido. A rigor, estes não são então teoremas (comparados com a definição de teorema na página 101).

23. Página 109. As proposições 14 e 15 (que naturalmente deveriam servir de exemplo ao Teorema da Dedução logo acima demonstrado) não o usam.

24. O Teorema da Completude mostra somente a completude do sistema de Rasiowa, mas não do sistema Pa (com $\neg$ e $\vee$ como primitivos).

25. Página 115. No argumento final do Teorema da Completude, não é verdade que "repetindo este processo $n-2$ vezes conclui-se (a demonstração de) H". O processo deve ser repetido no total 2^n vezes, caso contrário teríamos descoberto um método subexponencial para decidibilidade em lógica proposicional (esta sim, de fato, uma "questão importante para Ciência da Computação").

26. Página 133. No Exemplo, a noção de árvore semântica (tableaux) aplicada a um conjunto (e não a uma única fórmula) não foi definida.

27. Página 136. "Inicializar", segundo o Dicionário Houaiss, é a forma não-preferencial de "iniciar".

28. Página 139. Novamente, a definição de conseqüência lógica árvores semânticas esconde o caso dos conjuntos infinitos.

29. 29. Página 144. A frase: "Portanto, dada uma fórmula H, se o tableau associado a H não é fechado, nada se pode concluir sobre a validade de H. Por outro lado, se o tableau associado a H que é fechado então H é contraditório, isto é, $\neg H$ é uma tautologia", além de mal escrita, expressa uma idéia equivocada sobre tablôs. O método dos tablôs é um método de decisão para a lógica proposicional, e basta que um único tableau "associado" a uma fórmula H seja fechado ou não, para poder decidir se H é ou não uma tautologia. Toda a questão se resume em esclarecer o que significa "associado", o que não é feito neste livro.

30. 30. Página 144. Os créditos sobre o método de resolução fazem referencia a um incerto "Robinson", que deve ser J.A. Robinson (o qual, de resto, não aparece na bibliografia). A referência correta é "A Machine-Oriented Logic Based on the Resolution Principle". *ACM Journal*, **12**(1):23-41, 1965.

31. 31. Página 166. A confusão conceitual agora atinge a semântica da lógica de predicados: no exemplo sobre termos, o autor "esclarece" que os símbolos para as funções de adição e subtração de números naturais "não pertencem ao alfabeto da lógica de predicados". Como se pode então formalizar a aritmética na lógica de predicados? Este é um erro conceitual tão fundamental, que a partir desse ponto não faz sentido gastar mais tempo e espaço criticando os próximos capítulos, que tratam da axiomática e dos tableaux para a lógica de predicados e de alguns tópicos de resolução e programação lógica. Pouco mais se faz a respeito das ciências da computação que repetir exaustivamente o chavão de que os tópicos apresentados são "importantes para a representação e dedução formal de conhecimento".

32. Para dar uma idéia da dificuldade que o leitor enfrentará, convém notar, na página 187, a "explicação" sobre a diferença entre o quantificador universal

$\forall$ na linguagem-objeto e a contraparte denotada por "$\forall$" na metalinguagem: "Nas regras semânticas para interpretação de fórmulas com quantificadores há uma diferença entre $\forall$ e "$\forall$". O primeiro é um conectivo do alfabeto da lógica de predicados, sendo portanto um símbolo sintático. Por outro lado, "$\forall$" é um símbolo semântico que significa "para todo", e está quantificado sobre objetos semânticos."

33. Página 293. Aqui encontra-se mais uma falha conceitual grave, esta envolvendo diretamente o uso da lógica na chamada "programação lógica". O autor afirma que uma cláusula G (num programa lógico Pr) da forma $\leftarrow A_1, \ldots, A_n$ equivale à fórmula $\neg(A_1 \wedge \cdots \wedge A_n)$, quando na verdade a cláusula G equivale ao fecho universal desta última fórmula, isto é, G equivale a uma fórmula fechada. O autor continua o seu raciocínio: ele afirma que $(Pr \rightarrow (A_1 \wedge \cdots \wedge A_n))$ é válida sse $(Pr \wedge \neg(A_1 \wedge \cdots \wedge A_n))$ admite uma refutação. Isto seria verdadeiro, não fosse que o autor esqueceu novamente de considerar os quantificadores: a frase deveria ser: $(Pr \rightarrow (\exists x_1) \cdots (\exists x_m)(A_1 \wedge \cdots \wedge A_n))$ é válida sse $(Pr \wedge \neg(\exists x_1) \cdots (\exists x_m)(A_1 \wedge \cdots \wedge A_n))$ admite uma refutação ou, equivalentemente, see $(Pr \wedge (\forall x_1) \cdots (\forall x_m)\neg(A_1 \wedge \cdots \wedge A_n))$ admite uma refutação (aqui, estamos supondo que tomamos o fecho existencial e o fecho universal das respectivas fórmulas). Ou seja, se uma cáusula objetivo G como acima acrescentada a um programa Pr produz uma refutação significa que a sentença da forma $(\exists x_1) \cdots (\exists x_m)(A_1 \wedge \cdots \wedge A_n)$ (e não $(A_1 \wedge \cdots \wedge A_n)$, como erroneamente aponta o autor no final da pag. 293) é conseqüência lógica do programa Pr. Esquecendo desta maneira os quantificadores (universais e existenciais) que governam um progama lógico, o livro não esclarece que as consultas feitas a um programa lógico são acerca da existência (construtiva) de objetos satisfazendo certas propriedades. Este importante ponto conceitual da programação lógica é omitido pelo autor, que acaba apresentando resultados errados do ponto de vista lógico: sem um entendimento correto da lógica, não há qualquer esperança de que um estudante contrua programas lógicos decentes. Um erro de mesmo calibre é repetido na formulação do teorema da completude na pag. 294.

Outro defeito recorrente ocorre nos capítulos 13 e 14 : em diversos exercícios, o autor pede aos alunos que utilizem o método dos tableaux (e da resolução) para decidir se uma dada fórmula da lógica de predicados é ou não valida, ou para decidir se um determinado conjunto de fórmulas é ou não satisfatível. O autor não esclarece que estes problemas são em geral indecidíveis, como foi provado pelo célebre lógico A. Church já na década de 30, em seu famoso artigo *An unsolvable problem in elementary number theory* publicado no *American Journal of Mathematics* **58**:345-363 (1936), sendo somente possível determinar a validade das fórmulas de primeira ordem por métodos automáticos em certos casos especiais. A ignorância de um resultado dessa magnitude pode levar a preceitos totalmente errados sobre possibilidades de se construir programas.

Ainda outro ponto concerne o tratamento dado à semântica das fórmulas da lógica de Predicados: como consequência da falta de cuidado nas definições envolvidas, o texto evita se referir ao Teorema da Completude sob a forma usual:

"F é consequência lógica de um conjunto S de fórmulas se e somente se F é dedutível de S".

O autor só se refere ao Teorema da Completude sob a forma fraca:

"F é válida se e somente se F é teorema" (no sistema dedutivo em questão)

A ausência do Teorema da Completude (forte) para a lógica de predicados tem, mais uma vez, desagradáveis consequências para estudantes de computação, uma vez que este conceito é essencial em outras disciplinas, como no estudo dos bancos de dados onde o conceito de "dependências funcionais" utiliza a chamada "Axiomática de Armstrong", um sistema axiomático de predicados que necessita uma forma forte do Teorema da Completude para ser aplicado.

Há ainda erros tipográficos e falhas na bibliografia, mas o mais importante é que os estudantes e professores estejam alertas, exigindo das editoras, ou da produção independente de textos livres como é cada vez mais freqüente, qualidade para os textos acadêmicos, exigindo mais do que versões discutíveis do que já existe de forma correta em língua inglesa. Para prevenir que um novo Santo Agostinho, agora interessado apenas nas virtudes das ciências da computação e confundido por suas leituras brasileiras, não venha agora a denunciar um consórcio entre os lógicos (ao invés de os matemáticos) e o diabo.