

Capítulo 4

Lógica Simbólica

Com o advento da inteligência artificial, houve um incremento no interesse por processos automatizados, ou semi-automatizados, para a escrita de programas, atendimento à estímulos e, em últimas palavras, prova sistemática de teoremas. As bases da prova automática de teoremas foram desenvolvidas em 1930 por Herbrand, cujo método foi impossível de ser aplicado até a invenção do computador digital.

A lógica simbólica pode ser estudada sob diversas óticas, como por exemplo, a filosófica e a matemática. Neste capítulo o interesse está na aplicação da lógica simbólica, isto é, pretende-se usar a lógica simbólica para representar problemas e obter sua solução.

Uma vez que a lógica simbólica ainda não foi apresentada ao leitor, pede-se que o mesmo confie em sua intuição para acompanhar o início deste texto, ao menos por alguns instantes. Assuma as sentenças do exemplo a seguir:

S1 : Se está quente e está úmido, então vai chover.

S2 : Se está úmido, então está quente.

S3 : Está úmido agora.

Pergunta : Vai chover?

Estas sentenças estão representadas em português, porém pode-se empregar símbolos na sua representação. Assuma que Q , U , C representam, respectivamente, as sentenças “Está quente”, “Está úmido” e “Vai chover”. Também serão necessários alguns conectivos para fazer a junção das sentenças. O “e” será representado pelo

\wedge , e o sentido de implicação das sentenças do tipo “Se ... Então” será descrito por \rightarrow . Assim o conjunto de sentenças em português pode ser reescrito como:

$$S1: Q \wedge U \rightarrow C$$

$$S2: U \rightarrow Q$$

$$S3: U$$

$$Pergunta: C$$

Ao reescrever as sentenças através de símbolos e conectivos aqui propostos, foram criadas o que se chamam *fórmulas lógicas*. No decorrer deste capítulo, o leitor irá observar que se $S1$, $S2$ e $S3$ forem verdadeiros então a fórmula associada à pergunta também será verdadeira, isto é, vai chover. Neste sentido diz-se que a pergunta é *logicamente dedutível* de $S1$, $S2$ e $S3$.

No exemplo, foi preciso mostrar que uma fórmula pôde ser obtida a partir de outras, e chama-se esta fórmula obtida de *teorema*. Para que a prova de um teorema seja verdadeira, é preciso que a fórmula seja logicamente dedutível a partir de outras. Sob esta ótica, o problema relacionado com provadores automáticos de teoremas está associado com a busca por métodos sistemáticos para demonstração destes teoremas.

4.1 Lógica Proposicional

Na lógica proposicional se está interessado em sentenças que podem assumir valores verdadeiro ou falso, conhecidos como *valores verdade*, e tradicionalmente representados por 1 e 0 respectivamente. A estas sentenças dá-se o nome de *proposições*. Como exemplo de proposições verdadeiras podemos citar: “ $100 > 99$ ” e “Todo pássaro voa”. Proposições falsas podem ser ilustradas por “ $99 > 100$ ” e “Pássaros não voam”.

Entretanto, existem sentenças que não podem ser aceitas como proposições pois não é possível lhes atribuir valor verdade, como por exemplo, “Esta sentença é falsa”, “Nunca diga nunca” e “Eu não sou demonstrável”. Em alguns casos, proposições isoladas podem ser livres de contradição, mas quando onservadas em conjunto emerge uma contradição: “A afirmação a seguir é verdadeira” e “A afirmação

anterior é falsa”.

Além disto, também existem aquelas proposições as quais possuem valor verdade, mas não há concessão sobre qual é esse valor: “Monteiro Lobato foi um escritor mais importante que Jorge Amado” e “Flamengo é freguês do Fluminense. Vasco, coitado, é saco de pancada”.

As proposições que não usam conectivos são chamadas de *fórmulas atômicas*, ou *átomos*, caso contrário são chamadas “proposições compostas”. São exemplos de proposições compostas, com conectivos “e” e “se ... então”: “Chapeuzinho Vermelho foi à floresta e o Lobo Mau foi para a casa da Vovó” e “Se tenho olhos grandes, então é para te ver melhor”. Na lógica proposicional pode-se empregar cinco conectivos lógicos: \neg (não), \wedge (e), \vee (ou), \rightarrow (se ... então), \leftrightarrow (se somente se). A ordem de precedência decrescente entre estes conectivos é dada por: \leftrightarrow , \rightarrow , \wedge , \vee , \neg .

Definição 4.1 [Fórmulas bem fundadas] Fórmulas bem fundadas, ou simplesmente fórmulas, na lógica proposicional são definidas recursivamente conforme a seguir:

1. Um átomo é uma fórmula.
2. Se G é uma fórmula, então $\neg G$ também é uma fórmula.
3. Se G e H são fórmulas, então $(G \wedge H)$, $(G \vee H)$, $(G \rightarrow H)$ e $(G \leftrightarrow H)$ são fórmulas.
4. Todas as fórmulas são geradas a partir destas regras.

Sejam G e H duas fórmulas. De acordo com seus valores verdade tem-se que a sua combinação imediata com os conectivos é dada pela tabela verdade 4.1.

Tabela 4.1: Tabela verdade do \neg , \wedge , \vee , \rightarrow e \leftrightarrow .

G	H	$\neg G$	$G \wedge H$	$G \vee H$	$G \rightarrow H$	$G \leftrightarrow H$
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

Definição 4.2 [Interpretação] Dada a fórmula proposicional G , e sejam A_1, A_2, \dots, A_n átomos que ocorrem nesta fórmula. Uma interpretação de G é uma possibilidade de valores verdade, 0 ou 1, assumidos por A_1, A_2, \dots, A_n .

Considere a fórmula $G = ((P \rightarrow Q) \wedge P) \rightarrow Q$. Os valores verdade para cada uma das quatro interpretações desta fórmula são apresentados na Tabela 4.2. Observe ainda nesta tabela que os valores de G são sempre 1 quaisquer que sejam as interpretações. Quando isto ocorre, diz-se que a fórmula é uma *fórmula válida*, ou simplesmente uma *tautologia*.

Tabela 4.2: Tabela verdade de uma tautologia.

P	Q	$(P \rightarrow Q)$	$(P \rightarrow Q) \wedge P$	$((P \rightarrow Q) \wedge P) \rightarrow Q$
1	1	1	1	1
1	0	0	0	1
0	1	1	0	1
0	0	1	0	1

Analogamente, seja a fórmula $G = (P \rightarrow Q) \wedge (P \wedge \neg Q)$, cujos valores-verdade são apresentados na Tabela 4.3. Neste caso as quatro interpretações associadas à fórmula assumem o valor 0. Sob estas circunstâncias, diz-se que G é uma *fórmula inconsistente*, ou também, uma *contradição*.

Tabela 4.3: Tabela verdade de uma contradição.

P	Q	$\neg Q$	$(P \rightarrow Q)$	$(P \wedge \neg Q)$	$(P \rightarrow Q) \wedge (P \wedge \neg Q)$
1	1	0	1	0	0
1	0	1	0	1	0
0	1	0	1	0	0
0	0	1	1	0	0

4.2 Forma Normal na Lógica Proposicional

Durante a manipulação de fórmulas é comum ser necessário trocar determinadas fórmulas por outras, seja por motivos de simplicidade de representação, seja

por necessidade de torná-las mais claras. Para que esta troca possa ser realizada é necessário que a nova fórmula F seja equivalente à antiga G , isto é, que os valores-verdade de F e G sejam os mesmos em qualquer que seja a interpretação. A Tabela 4.4 ilustra que as fórmulas $P \rightarrow Q$ e $\neg P \vee Q$ são logicamente equivalentes, ou seja, $P \rightarrow Q \equiv \neg P \vee Q$.

Tabela 4.4: Tabela verdade de uma equivalência lógica.

P	Q	$\neg P$	$(P \rightarrow Q)$	$\neg P \vee Q$
1	1	0	1	1
1	0	1	0	0
0	1	0	1	1
0	0	1	1	1

O processo para a verificação da equivalência lógica exige a construção de tabelas verdade tal como a Tabela 4.4. Conforme o número de átomos cresce, a quantidade de interpretações cresce significativamente ($2^{\text{átomos}}$), tornando o processo demorado, tedioso e muitas vezes, inviável.

De modo a agilizar a verificação de equivalências, torna-se desejável o emprego de algumas regras, listadas na Tabela 4.5, que contribuem para a redução do tempo gasto nesse processo. Estas relações de equivalência lógica podem ser facilmente comprovadas através de tabelas-verdade.

Tabela 4.5: Regras de equivalência para normalização.

$r.1)$	$P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$		
$r.2)$	$P \rightarrow Q \equiv \neg P \vee Q$		
$r.3a)$	$P \vee Q \equiv Q \vee P$	$r.3b)$	$P \wedge Q \equiv Q \wedge P$
$r.4a)$	$(P \vee Q) \vee R \equiv P \vee (Q \vee R)$	$r.4b)$	$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$
$r.5a)$	$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$	$r.5b)$	$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
$r.6a)$	$P \vee \mathbf{0} \equiv P$	$r.6b)$	$P \wedge \mathbf{0} \equiv \mathbf{0}$
$r.7a)$	$P \vee \mathbf{1} \equiv \mathbf{1}$	$r.7b)$	$P \wedge \mathbf{1} \equiv P$
$r.8a)$	$P \vee \neg P \equiv \mathbf{1}$	$r.8b)$	$P \wedge \neg P \equiv \mathbf{0}$
$r.9)$	$\neg(\neg P) \equiv P$		
$r.10a)$	$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$	$r.10b)$	$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$

Observe que se $(P \vee Q) \vee R \equiv P \vee (Q \vee R)$ então pode-se afirmar que são desnecessários os parênteses. Assim pode-se escrever equivalentemente $P \vee Q \vee R$. Quando isto acontece, diz-se que se tem uma *disjunção* de P , Q e R . De forma análoga, $P \wedge Q \wedge R$ representa uma *conjunção* de P , Q e R .

Definição 4.3 [Literal] Um *literal* é um átomo, ou a negação de um átomo.

Definição 4.4 [Forma Normal Conjuntiva] Uma fórmula F é dita estar representada através de uma *forma normal conjuntiva* se e somente se $F \equiv F_1 \wedge F_2 \wedge \dots \wedge F_n$, onde $n \geq 1$ e F_1, F_2, \dots, F_n são literais disjuntos.

Definição 4.5 [Forma Normal Disjuntiva] Uma fórmula F é dita estar representada através de uma *forma normal disjuntiva* se e somente se $F \equiv F_1 \vee F_2 \vee \dots \vee F_n$, onde $n \geq 1$ e F_1, F_2, \dots, F_n são literais conjuntos.

Exemplo 4.1 Obtenha a forma normal disjuntiva de $(P \vee \neg Q) \rightarrow R$.

$$\begin{aligned} (P \vee \neg Q) \rightarrow R &\equiv \neg(P \vee \neg Q) \vee R && (r.2) \\ &\equiv (\neg P \wedge \neg \neg Q) \vee R && (r.10a) \\ &\equiv (\neg P \wedge Q) \vee R && (r.9) \end{aligned}$$

Exemplo 4.2 Obtenha a forma normal conjuntiva de $(P \wedge (Q \rightarrow R)) \rightarrow S$.

$$\begin{aligned} (P \wedge (Q \rightarrow R)) \rightarrow S &\equiv (P \wedge (\neg Q \vee R)) \rightarrow S && (r.2) \\ &\equiv \neg(P \wedge (\neg Q \vee R)) \vee S && (r.2) \\ &\equiv (\neg P \vee \neg(\neg Q \vee R)) \vee S && (r.10b) \\ &\equiv (\neg P \vee (\neg \neg Q \wedge \neg R)) \vee S && (r.10a) \\ &\equiv (\neg P \vee (Q \wedge \neg R)) \vee S && (r.9) \\ &\equiv ((\neg P \vee Q) \wedge (\neg P \vee \neg R)) \vee S && (r.5a) \\ &\equiv S \vee ((\neg P \vee Q) \wedge (\neg P \vee \neg R)) && (r.3a) \\ &\equiv (S \vee (\neg P \vee Q)) \wedge (S \vee (\neg P \vee \neg R)) && (r.5a) \\ &\equiv (S \vee \neg P \vee Q) \wedge (S \vee \neg P \vee \neg R) && (r.4a) \end{aligned}$$

Um desdobramento direto das formas normais é a questão da conseqüência lógica.

Definição 4.6 [Conseqüência Lógica] Dadas as fórmulas P_1, P_2, \dots, P_n e a fórmula Q , Q é dito ser uma *conseqüência lógica* de P_1, P_2, \dots, P_n se e somente se para cada

interpretação I em que $P_1 \wedge P_2 \wedge \dots \wedge P_n$ for verdadeira, Q também é verdadeira. Matematicamente, diz-se $P_1, P_2, \dots, P_n \models Q$. P_1, P_2, \dots, P_n são chamados *axiomas* (ou *postulados*, *premissas*) de Q .

Teorema 4.1 [Teorema da Dedução] Dadas as fórmulas F_1, \dots, F_n e a fórmula G , dizemos que G é uma consequência lógica de F_1, \dots, F_n se e somente se a fórmula $(F_1 \wedge \dots \wedge F_n) \rightarrow G$ é válida.

(\Rightarrow) Considere I como sendo uma interpretação arbitrária qualquer. Se I é verdadeira em F_1, \dots, F_n , então por definição de consequência lógica, I é verdadeira em G . Assim I também é verdadeira em $(F_1 \wedge \dots \wedge F_n) \rightarrow G$. Por outro lado, se I é falso em F_1, \dots, F_n então I continua sendo verdadeira em $(F_1 \wedge \dots \wedge F_n) \rightarrow G$. Assim $(F_1 \wedge \dots \wedge F_n) \rightarrow G$ é uma fórmula válida.

(\Leftarrow) Suponha que $(F_1 \wedge \dots \wedge F_n) \rightarrow G$ seja uma fórmula válida, Qualquer que seja uma interpretação I , se $(F_1 \wedge \dots \wedge F_n)$ é verdadeiro em I , então G também é verdadeiro em I . Deste modo G é uma consequência lógica de F_1, \dots, F_n .

Teorema 4.2 [Teorema da Dedução por Refutação] Dadas as fórmulas F_1, \dots, F_n e G , G é uma consequência lógica de F_1, \dots, F_n se e somente se a fórmula $(F_1 \wedge \dots \wedge F_n \wedge \neg G)$ é inconsistente.

Pelo Teorema 4.1, G é uma consequência lógica de F_1, \dots, F_n se e somente se $(F_1 \wedge \dots \wedge F_n) \rightarrow G$ é válida. Deste modo a negação $\neg((F_1 \wedge \dots \wedge F_n) \rightarrow G)$ precisa ser uma fórmula inconsistente.

$$\begin{aligned} \neg((F_1 \wedge \dots \wedge F_n) \rightarrow G) &\equiv \neg(\neg(F_1 \wedge \dots \wedge F_n) \vee G) \\ &\equiv \neg\neg(F_1 \wedge \dots \wedge F_n) \wedge \neg G \\ &\equiv (F_1 \wedge \dots \wedge F_n) \wedge \neg G \\ &\equiv F_1 \wedge \dots \wedge F_n \wedge \neg G \end{aligned}$$

de onde se conclui o Teorema da Dedução por Refutação.

Os Teoremas 4.1 (Dedução) e 4.2 (Dedução por Refutação) são importantes pois eles são utilizados diretamente na prova automática de teoremas. Para ilustrar

Referências Bibliográficas

- [1] HOPCROFT, J. E., *Theory os Machines and Compuatations*, chapter An n log n algorithm for minimizing states in a finite automaton, Academic Press, pp. 189–196, 1971.
- [2] MONTEIRO, A. A., PAULO, J. D. S., *Aritmética Racional*. Lisboa, Livraria Avelar Machado, 1945.
- [3] IFRAH, G., *Os Números: a história de uma grande invenção*. São Paulo, Globo S.A., 2005.
- [4] DEDEKIND, R., *Was sind und was sollen die Zahlen*, v. 3, *Gesammelte Mathematische Werke*. New York, Chelsea Publishing Company, 1969. pp. 335-391.
- [5] GÖDEL, K., *Collected Works*, v. 2, *Gesammelte Mathematische Werke*. Oxford, Oxford University Press, 1990.
- [6] ISRAEL, D., “Reflections on Gödel’s and Gandy’s Reflection on Turing’s Thesis”, *Minds and Machines*, v. 12, n. 2, pp. 181–201, 2002.
- [7] SOBRINHO, J. Z., “Aspectos da Tese de Church-Turing”, *Revista Matemática Universitária - USP*, v. 1, n. 6, pp. 1–23, 1987.
- [8] MCDERMOTT, D., “Artificial Intelligence Meets Natural Stupidity”, *SI-GART Newsletter*, v. 57, pp. 4–9, April 1976.
- [9] SETTI, M. D. O. G., *O Processo de Discretiza çã o do Raciocínio Matemático na Tradu çã o para o Raciocínio Computacional*, Report, Universidade Federal do Paraná, 2009.

- [10] GUERREIRO, G., “A Vanguarda Matemática e os Limites da Razão”, *Scientific American Brasil*, v. 5, n. 12, pp. 39–56, 2007. Coleção Gênios da Ciência.
- [11] SMULLYAN, R., *Gödel’s Incompleteness Theorems*. Oxford University Press, 1992.
- [12] GÖDEL, K., *The Undecidable*, chapter On Formally Undecidable Propositions of Principia Mathematica and Related Systems, New York, Raven Press, pp. 5–38, 1965.
- [13] WHITEHEAD, A. N., RUSSELL, B., *Principia Mathematica*. Londres, Cambridge University Press, 1913.
- [14] NAGEL, E., NEWMAN, J. R., *Gödel’s Proof*. USA, Routledge, 1989.
- [15] GÖDEL, K., “Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme - On Formally Undecidable Propositions of Principia Mathematica and Related Systems”, *Kurt Gödel: Collected Works*, v. 1, n. 1, pp. 144–195, 1986. Tradução para o inglês por Martin Hirzel. www.research.ibm.com/people/h/hirzel/papers/canon00-goedel.pdf [capturado em 13 de agosto de 2011].
- [16] MELO, A. C. V. D., SILVA, F. S. C. D., *Modelos Clássicos de Computação*, Coleção Schaum. São Paulo, Thomson, 2006.
- [17] KUBRUSLY, R. D. S., *Uma viagem informal ao Teorema de Gödel ou (O preço da matemática é o eterno matemático)*, Report, Universidade Federal do Rio de Janeiro, 2007. IM/UFRJ.
- [18] SMULLYAN, R., *Recursion Theory for Metamathematics*. Oxford University Press, 1993.
- [19] SMULLYAN, R., *What’s the Name of This Book*. Penguin Books, 1978.
- [20] SMULLYAN, R., *Forever Undecided: A Puzzle Guide to Gödel*. Oxford University Press, 1987.
- [21] GOLDSTEIN, R., *Incompleteness: The Proof and Paradox of Kurt Gödel*. W. W. Norton Company, Inc., 2005.

- [22] HOFSTADTER, D. R., *Gödel, Escher e Bach: an Eternal Golden Braid*. Nova Iorque, Basic Books, 1979.
- [23] Rodríguez-Consuegra, F. A. (ed.), *Kurt Gödel - Unpublished Philosophical Essays*. Berlin, Birkhäuser Verlag, 1995.
- [24] Feferman, S., *et al.* (eds.), *Kurt Gödel - Collected Works*, v. I, II, III. New York, Oxford University Press, 1986.
- [25] WANG, H., *Reflections on Kurt Gödel*. Cambridge, Massachusetts, The MIT Press, 1988.
- [26] SUPPES, P., *Axiomatic Set Theory*. New York, Dover, 1972.
- [27] LIPSCHUTZ, S., *Teoria Elementar dos Conjuntos*, Cole çã o Schaum. São o Paulo, McGraw-Hill do Brasil, 1990.
- [28] SUPPES, P., *Axiomatic Theory Set*. USA, Van Nostrand Company Inc., 1960.
- [29] MELLO, F. L. D., CARVALHO, R. L. D., “Knowledge Geometry”, *Journal of Information and Knowledge Management*, v. 14, pp. 1550028, 2015.
- [30] STOLL, R. R., *Set Theory and Logic*. USA, Dover Publications Inc., 1961.
- [31] MIRAGLIA, F., *Teoria dos Conjuntos: um mínimo*. Brasil, Edusp - Editora da Universidade de São Paulo, 1992.
- [32] HALMOS, P., *Teoria Ingênua dos Conjuntos*. Brasil, Edusp - Editora da Universidade de São Paulo, 1970.
- [33] GRATZER, G., *Universal Algebra*. USA, Van Nostrand Company Inc., 1968.
- [34] COHN, P. M., *Universal Algebra*. USA, Harper and Row, 1965.
- [35] GALLIER, J. H., *Logic for Computer Science*. USA, John Wiley and Sons Inc., 1987.
- [36] PREPARATTA, F. P., YEH, R. T., *Introduction to Discreate Structures for Computer Science and Engineering*. USA, Addison-Wesley Publishing Company, 1973.

- [37] SHOENFIELD, J. R., *Degrees of Unsolvability*. North-Holland Publishing Company, 1971.
- [38] BRAINERD, W. S., LANDWEBER, L. H., *Theory of Computation*. USA, John Wiley and Sons, 1974.
- [39] EILENBERG, S., ELGOT, C., *Recursiveness*. New York: Academic Press, 1970.
- [40] CARVALHO, R. L. D., OLIVEIRA, C. M. G. M. D., *Modelos de Computação e Sistemas Formais*, 11^a Escola de Computação. Rio de Janeiro, Universidade Federal do Rio de Janeiro, 1998.
- [41] BRAINERD, W. S., LANDWEBER, L. H., *Theory of Computation*. John Wiley & Sons, 1974.
- [42] KLEENE, S. C., *Introduction to Metamathematics*. D. Van Nostrand Company, Inc., 1952.
- [43] Rogers Jr., H., *Theory of Recursive Functions and Effective Computability*. USA, McGraw-Hill Book Company, 1967.
- [44] DAVIS, M., *Computability and Unsolvability*. New York, Dover, 1983.
- [45] BOOLOS, G. S., JEFFREY, R. C., *Computability and Logic*. Cambridge University Press, 1974.
- [46] MARTIN DAVIS, R. S., WEYUKER, E. J., *Computability Complexity and Languages*. New York, Academic Press, 1994.
- [47] MALLOZZI, J. S., LILLO, N. J. D., *Computability with Pascal*. New Jersey, Prentice-Hall, Inc., 1984.
- [48] TURING, A. M., *The Undecidable*, chapter On Computable Numbers, with an Application to the Entscheidungsproblem, New York, Raven Press, pp. 115–151, 1965.
- [49] ELGOT, C. C., ROBINSON, A., “Random-access stored-program machines, an approach to programming languages”, *Journal of the ACM*, v. 11, pp. 365–399, 1964.

- [50] PREPARATTA, F. P., YEH, R. T., *Introduction to Discrete Structures for Computer Science and Engineering*. Addison-Wesley Publishing Company, 1973.
- [51] HENNIE, F., *Introduction to Computability*. Addison-Wesley Publishing Company, 1977.
- [52] NELSON, R. J., *Introduction to Automata*. USA, Jonh Wiley & Sons, Inc., 1968.
- [53] CARVALHO, R. L. D., *Máquinas, Programas e Algoritmos*, 2^a Escola de Computação. Campinas, Universidade Estadual de Campinas, 1981.
- [54] MENDELSON, E., *Introduction to Mathematical Logic*, Cole Mathematics Series. 3 ed. The Wadsworth and Brooks, 1987.
- [55] MANIN, Y. I., *A Course in Mathematical Logic*, Graduate Texts in Mathematics 53. 1 ed. Springer-Verlag, 1977.
- [56] HOMER, S., SELMAN, A. L., *Computability and Complexity Theory*, Texts in Computer Science. 2 ed. Springer, 2011.
- [57] CUTLAND, N. J., *Computability: An introduction to recursive function theory*. Cambridge University Press, 1980.
- [58] SIPSER, M., *Introduction to The Theory of Computation*, Course Technology Series. 2 ed. Thomson, 2006.
- [59] WALTER CARNIELLI, R. L. E., *Computability: computable functions, logic and the foundations of mathematics*. Belmont, Wadsworth and Brooks, 1989.
- [60] TARSKI, A., *Logic, semantics, metamathematics*, chapter Fundamental concepts of the methodology of the deductive sciences, London, Oxford at the Clarendon Press, pp. 60–109, 1969.
- [61] ADAM YOUNG, M. Y., *Malicious Cryptography: Exposing Cryptovirology*. John Wiley and Sons Inc., 2004.

- [62] BONFANTE, G., KACZMAREK, M., MARION, J.-Y., *A Classification of Viruses through Recursion Theorems*, volume 4497 of Lecture Notes in Computer Science. 2 ed. CiE 2007, 2007.
- [63] MACHTEY, M., YOUNG, P., *An Introduction to the General Theory of Algorithms*. New York, North Holland, 1978.
- [64] ROBINSON, J. A., “A machine oriented logic based on the resolution principle”, *J. Assoc. Comput.*, v. 12, pp. 23–41, 1965.
- [65] ROBINSON, J. A., “Automatic deduction with hyper-resolution”, *Internat. J. Comput. Math.*, v. 1, pp. 227–234, 1965.
- [66] GILMORE, P. C., “A proof method for quantification theory: Its justification and realization”, *IBM Journal of Research and Development*, v. 4, n. 1, pp. 28–35, 1960.
- [67] DAVIS, M., PUTNAM, H., “A computing procedure for quantification theory”, *Journal of the ACM*, v. 7, n. 3, pp. 201–215, 1960.
- [68] CHANG, C.-L., LEE, R. C.-T., *Symbolic Logic and Mechanical Theorem Proving*. Academic Press Inc, 1973.
- [69] KLEENE, S. C., *Mathematical Logic*. Wiley, 1967.
- [70] HILBERT, D., ACKERMANN, W., *Prinmciples of Mathematical Logic*. Chelsea, 1950.
- [71] MCCAWLEY, J. D., *Everything That Linguists Have Always Wanted To Know About Logic*. 2 ed. The University of Chicago Press, 1993.
- [72] SUPPES, P., *Introduction to Logic*. D. van Nostrand, 1966.
- [73] RUSSELL, B., *A Filosofia do Atomismo Lógico*, *Lógica e Conhecimento*. 1 ed. Abril Cultural, 1974. (Os Pensadores, 42).
- [74] RUSSELL, B., *Significado e Verdade*. 1 ed. Zahar, 1978.
- [75] POPPER, K. R., *A Lógica da Pesquisa Científica*. 2 ed. Cultrix, 1974.

- [76] LAKATOS, I., , MUSGRAVE, A., *A Crítica e o Desenvolvimento do Conhecimento*. 1 ed. EDUSP, Cultrix, 1979. Tradução: M. O. Caiado.
- [77] WANG, H., *From Mathematics to Philosophy*. 1 ed. Cultrix, 1974.
- [78] GREEN, C. C., *The Application of Theorem Proving to Question-Answering Systems*. Ph.D. dissertation, Stanford, June 1969. AI Project MEMO AI-96.
- [79] LOVELAND, D. W., *Automated Theorem Proving: a Logical Basis*. 1 ed. North Holland, 1978.
- [80] HUGHES, G. E., LONDEY, D. G., *The Elements of Formal Logic*. USA, Methuen and Co Ltd, 1965.
- [81] BOOK, R. V., OTTO, F., *String-Rewriting Systems*. USA, Springer-Verlag, 1993.
- [82] HOPCROFT, J. E., ULLMAN, J. D., *Introduction to Automata Theory, Language and Computation*. USA, Addison-Wesley Publishing Company, 1979.
- [83] HOPCROFT, J. E., ULLMAN, J. D., *Formal Languages and their Relation to Automata*. USA, Addison-Wesley Publishing Company, 1969.
- [84] CURRY, H. B., *Foundations of Mathematical Logic*. New York, Academic Press, 1977.
- [85] SMULLYAN, R., *Theory of Formal Systems*. USA, Princeton, 1961.
- [86] BERSTEL, J., BOASSON, L., CARTON, O., *et al.*, *Handbook of Automata: from Mathematics to Applications*, chapter Minimization of automata, European Mathematical Society, pp. 189–196, 2010.
- [87] BEAL, M. P., CROCHEMORE, M., “Minimizing incomplete automata”, *Workshop on Finite State Methods and Natural Language Processing*, , september 2008. Ispra.
- [88] VALMARI, A., LEHTINEN, P., “Efficient minimization of DFAs with partial transition”, *Proc. 25th Symp. Theoretical Aspects of Comp. Sci.*, v. 08001, pp. 645–656, 2008. S. Albers and P. Weil, editors.

- [89] PAPADONIKOLAKIS, M., BOUGANIS, C.-S., CONSTANTINIDES, G.,
“Performance comparison of GPU and FPGA architectures for the SVM training problem”, *IEEE International Conference on FieldProgrammable Technology*, pp. 388–391, 2009.
- [90] MU, S., WANG, C., LIU, M., *et al.*, “Evaluating the potential of graphics processors for high performance embedded computing”, *Proc. IEEE Design, Automation and Test in Europe Conference and Exhibition*, pp. 709–714, 2011.
- [91] KAI HWANG, F. A. B., *Computer Architecture and Parallel Processing*. McGraw-Hill, 1984.
- [92] AGARWAL, P., KRISHNAN, S., MUSTAFA, N., *et al.*, *Algorithms - ESA 2003, Lecture Notes in Computer Science*, chapter Streaming Geometric Optimization Using Graphics Hardware, Springer Berlin-Heidelberg, pp. 115–151, 2003.
- [93] TANENBAUM, A. S., *Organização Estruturada de Computadores*. 3 ed. Prentice Hall do Brasil, 1997.
- [94] BACKUS, J., “Can Programming be Liberated from von Neumann Style? A functional style and its algebra of program”, *ACM Turing Award Lecture, Communications of the ACM*, v. 21, n. 8, pp. 613–641, 1978.
- [95] OWENS, J. D., LUEBKE, D., GOVINDARAJU, N., *et al.*, “A Survey of General-Purpose Computing on Graphics Hardware”, *Eurographics 2005, State of the Art Reports*, pp. 21–51, 2005.
- [96] GUSTAFSON, J. L., “Reevaluating Amdahl’s law”, *Communications of the ACM*, v. 5, n. 31, pp. 532, 1988.
- [97] HANDLER, W., *Parallel Processing Systems, an advanced course*, chapter Innovative computer architecture - how to increase parallelism but not complexity, Cambridge University Press, pp. 1–41, 1982.
- [98] LOBUR, J., NULL, L., *The Essentials of Computer Organization And Architecture*. Jones and Bartlett Pub, 2006.

- [99] LEWIS, H. R., PAPADIMITRIOU, C. H., *Elements of the Theory of Computation*. 2 ed. New York, Prentice-Hall, 1998.
- [100] DUNNE, P., *Computability Theory: Concepts and Applications*. Ellis Horwood, 1991.
- [101] AARONSON, S., “NP-complete Problems and Physical Reality”, *ACM SIGACT News*, , march 2005. Complexity Theory Column 46.