

## 4.3 Lógica de Primeira Ordem

Na lógica proposicional as primitivas básicas são chamadas de átomos que expressam valores 0 ou 1. Através dos átomos são escritas as fórmulas, e estas são empregadas para expressar idéias complexas. O átomo é tratado como uma unidade indivisível, sendo ignorados sua composição e estrutura. Entretanto muitas idéias não podem ser tratadas deste modo simplificado. Considere, por exemplo, as seguintes sentenças:

Todo homem é mortal.

Uma vez que João é homem, então João é mortal.

O raciocínio intuitivo leva o leitor a reconhecer como correta a conclusão acima. Contudo, a primeira sentença não pode ser representada sob a ótica da lógica proposicional, pois não é uma sentença meramente declarativa. Neste sentido, através da lógica de primeira ordem, serão usadas mais três noções lógicas, a saber: termo, predicado e quantificador.

Suponha que se deseja representar “João ama Maria”. Primeiro define-se um predicado  $ama(x, y)$ , e em seguida, é feita sua instanciação,  $ama(João, Maria)$ . Além disto, se João for pai de Fulano, pode-se reescrever o predicado  $ama$  como sendo  $ama(pai - de(Fulano), Maria)$ .

Informalmente, dizemos que João, Maria e Fulano são *símbolos individuais*, ou *constantes*. As constantes representam qualquer objeto sobre o qual se deseja descrever algo. A noção de constante é uma representação ampla, podendo ser dividida em constantes concretas (a praia, o carro, ...), constantes abstratas (a amizade, o amor, ...) e constantes fictícias (o Papai Noel, o Homem-Aranha, ...).

As variáveis  $x$  e  $y$  são chamadas de *símbolos variáveis*. Estes símbolos variáveis permitem estabelecer fatos a respeito de objetos do Universo, sem que haja necessidade de torná-los explícitos. Desta forma, as variáveis  $x$  e  $y$  podem assumir um valor *verdadeiro* ou *falso* de acordo com a quantificação e a substituição que será feita.

A função  $pai - de$  e  $ama$  são chamados de *símbolos predicativos*. Eles são res-

ponsáveis por representar propriedades ou relações entre objeto do Universo. Deste modo, estas funções mapeiam a pessoa chamada *Fulano* sobre aquela que é seu pai. Além disto, observe que os símbolos predicativos recebem um certo número de argumentos. Assim, se um símbolo predicativo  $P$  possui  $n$  argumentos, chama-se  $P$  de símbolo predicativo de  $n$ -argumentos.

**Definição 4.7** [Termo] Termo é definido recursivamente como:

- (i) Uma constante é um termo;
- (ii) Uma variável é um termo;
- (iii) Se  $P$  é um predicado de  $n$ -argumentos, e  $t_1, \dots, t_n$  são termos, então  $P(t_1, \dots, t_n)$  é um termo;
- (iv) Todos os termos são obtidos a partir destas regras.

A constante *Fulano* é um termo, e o predicado *pai – de* é um símbolo predicativo de 1-argumento. Assim, *pai – de(Fulano)* é um termo, bem como *pai – de(pai – de(Fulano))*, denotando o avô de Fulano, também é um termo.

**Definição 4.8** [Átomo] Se  $P$  é um símbolo predicativo de  $n$ -argumentos e  $t_1, \dots, t_n$  são termos, então  $P(t_1, \dots, t_n)$  é um átomo.

Uma vez que um átomo se encontra definido, pode-se utilizar os mesmos conectivos lógicos da lógica proposicional. Além disto, dado que foram introduzidas variáveis à representação, também é possível utilizar o **quantificador universal**  $\forall$  e o **quantificador existencial**  $\exists$ . Se  $x$  for uma variável, então  $(\forall x)$  é lido com "para todo  $x$ " ou "para cada  $x$ ", enquanto que  $(\exists x)$  é dito "existe um  $x$ ", "para algum  $x$ ", "para ao menos um  $x$ ".

As sentenças listadas a seguir apresentam suas respectivas representações em lógica, chamadas *fórmulas*:

- (a) Todo número racional é um número real.

$$(\forall x) (\mathbb{Q}(x) \rightarrow \mathbb{R}(x))$$

(b) Existe um número que é primo.

$$(\exists x) \text{primo}(x)$$

(c) Para todo número  $x$ , existe um número  $y$  tal que  $x < y$ .

$$(\forall x)(\exists y) \text{menor}(x, y)$$

O escopo de um quantificador em uma fórmula corresponde ao trecho desta fórmula ao qual o quantificador se aplica. Por exemplo, o escopo de ambos os quantificadores universal e existencial na fórmula  $(\forall x)(\exists y) \text{menor}(x, y)$  é  $\text{menor}(x, y)$ . O escopo para o quantificador universal na fórmula  $(\forall x) (\mathbb{Q}(x) \rightarrow \mathbb{R}(x))$  é  $\mathbb{Q}(x) \rightarrow \mathbb{R}(x)$ . Observe que na primeira situação o escopo não está explicitamente definido pelos símbolos  $($  e  $)$ , enquanto na segunda situação este escopo encontra-se explícito.

**Definição 4.9** Uma ocorrência de uma variável em  $x$  em uma fórmula é *ligada* se  $x$  é uma variável de um quantificador na fórmula, ou se  $x$  está no escopo de um quantificador  $(\forall x)$  ou  $(\exists x)$  na fórmula. Caso contrário, a ocorrência de  $x$  é *livre*.

**Definição 4.10** [Fórmula Bem Formada] Uma fórmula bem formada, ou simplesmente fórmula, na lógica de primeira ordem é definida como:

- (i) Um átomo é uma fórmula;
- (ii) Se  $F$  e  $G$  são fórmulas, então  $(\neg F)$ ,  $(F \vee G)$ ,  $(F \wedge G)$ ,  $(F \rightarrow G)$  e  $(F \leftrightarrow G)$  são fórmulas;
- (iii) Se  $F$  é uma fórmula e  $x$  é uma variável livre de  $F$ , então  $(\forall x) F$  e  $(\exists x) F$  são fórmulas;
- (iv) Fórmulas são geradas a partir de um número finito de aplicações de (i), (ii) e (iii).

Sempre que possível, e baseando-se na simplificação da representação da fórmula, pode-se omitir alguns parêntesis. Por exemplo,  $((\exists x) A) \vee B$  pode ser simplificado para  $(\exists x) A \vee B$ .

A representação do conhecimento empregado predicados envolve o uso de quatro categorias de sentenças, chamadas *enunciados categóricos*:

- *Universal Afirmativo*: estabelece que um relacionamento é um subconjunto de outro, normalmente representado com uma fórmula similar a  $(\forall x) [P(x) \rightarrow Q(x)]$ .  
Sentença: Todos os políticos são ladrões.  
Representação:  $(\forall x) [político(x) \rightarrow ladrão(x)]$
- *Universal Negativo*: estabelece uma relação de disjunção entre relacionamentos, normalmente representado com uma fórmula similar a  $(\forall x) [P(x) \rightarrow \neg Q(x)]$ .  
Sentença: Nenhum político é honesto.  
Representação:  $(\forall x) [político(x) \rightarrow \neg honesto(x)]$
- *Particular Afirmativo*: estabelece que os relacionamentos  $P$  e  $Q$  têm uma intersecção não vazia, normalmente representado com uma fórmula similar a  $(\exists x) [P(x) \wedge Q(x)]$ .  
Sentença: Alguns políticos são honestos.  
Representação:  $(\exists x) [político(x) \wedge honesto(x)]$
- *Particular Negativo*: estabelece que existem elementos do relacionamento  $P$  que não estão em  $Q$ , normalmente representado com uma fórmula similar a  $(\exists x) [P(x) \wedge \neg Q(x)]$ .  
Sentença: Alguns políticos não são honestos.  
Representação:  $(\exists x) [político(x) \wedge \neg honesto(x)]$

**Exemplo 4.3** Inicialmente, considere que se deseja traduzir as sentenças "Todo homem é mortal. João é homem. Deste modo, João é mortal". Denote " $x$  é homem" por  $homem(x)$ , e " $x$  é mortal" por  $mortal(x)$ . Assim, a sentença "todo homem é mortal" pode ser representada por  $(\forall x)(homem(x) \rightarrow mortal(x))$ . Observe que até esse ponto nada foi modelado sobre a constante João. Assim, "João é homem" é representado por  $homem(João)$ , e ainda, "João é mortal" por  $mortal(João)$ . Desta forma, a sentença final é  $(\forall x)(homem(x) \rightarrow mortal(x)) \wedge homem(João) \rightarrow mortal(João)$ .

Na lógica proposicional, uma interpretação é uma atribuição de valores verdade para um átomo. Em lógica de primeira ordem, uma vez que existem variáveis envolvidas, as interpretações não são tão imediatas. A idéia central é definir um domínio  $D$  sobre o qual a fórmula é aplicada. Com base nos valores verdade assu-

midos pelos predicados instanciados por elementos deste domínio  $D$ , verifica-se qual o resultado é obtido para toda a fórmula.

**Exemplo 4.4** Considere as fórmulas  $(\forall x)P(x)$  e  $(\exists x)\neg P(x)$ . Uma interpretação pode ser descrita como se segue:

Domínio:  $D = 1, 2$

Valores assumidos por  $P$ :  $P(1) = 1 \mid P(2) = 0$

O valor de  $(\forall x)P(x)$  para esta interpretação é 0 porque  $P(x)$  não é 1 para ambos  $x = 1$  e  $x = 2$ . Por outro lado, uma vez que  $\neg P(2) = 1$ , então  $(\exists x)\neg P(x)$  é 1 para esta interpretação.

**Exemplo 4.5** Considere a fórmula  $(\forall x)(\exists y)P(x, y)$ . A interpretação é definida como se segue:

Domínio:  $D = 1, 2$

Valores assumidos por  $P$ :  $P(1, 1) = 1 \mid P(1, 2) = 0 \mid P(2, 1) = 0 \mid P(2, 2) = 1$

Se  $x = 1$ , então existe um  $y$ , no caso  $y = 1$ , que torna o predicado  $P(1, y) = 1$ .

Se  $x = 2$ , então também existe um  $y$ , neste caso  $y = 2$ , o que torna o predicado  $P(2, y) = 1$ .

Desta forma, para esta interpretação  $(\forall x)(\exists y)P(x, y) = 1$ .

**Exemplo 4.6** Considere a fórmula  $G : (\forall x)(P(x) \rightarrow Q(f(x), a))$ . Nesta fórmula existe uma constante  $a$ , uma função  $f$ , e dois predicados  $P$  e  $Q$ . A interpretação  $I$  de  $G$  é avaliada conforme a seguir:

Domínio:  $D = 1, 2$

Valores assumidos por  $a$ :  $a = 1$

Valores assumidos por  $f$ :  $f(1) = 2 \mid f(2) = 1$

Valores assumidos por  $P$ :  $P(1) = 0 \mid P(2) = 1$

Valores assumidos por  $Q$ :  $Q(1, 1) = 1 \mid Q(1, 2) = 1 \mid Q(2, 1) = 0 \mid Q(2, 2) = 1$

Se  $x = 1$ , então

$$\begin{aligned} P(x) \rightarrow Q(f(x), a) &= P(1) \rightarrow Q(f(1), a) \\ &= P(1) \rightarrow Q(2, 1) \\ &= 0 \rightarrow 0 = 1 \end{aligned}$$

Se  $x = 2$ , então

$$\begin{aligned} P(x) \rightarrow Q(f(x), a) &= P(2) \rightarrow Q(f(2), a) \\ &= P(2) \rightarrow Q(1, 1) \\ &= 1 \rightarrow 1 = 1 \end{aligned}$$

Uma vez que  $P(x) \rightarrow Q(f(x), a)$  é 1 para todos os elementos  $x$  no domínio  $D$ , a fórmula  $G = 1$  sob a interpretação  $I$ .

## 4.4 Forma Normal Prenex

Em lógica proposicional foram introduzidos os conceitos de formas normais conjuntivas e disjuntivas. Em lógica de primeira ordem também existe uma forma normal, neste caso conhecida como *forma normal prenex*. A razão principal para se considerar fórmulas escritas em forma normal prenex deve-se às tentativas de simplificar o processo de prova, conforme será visto mais adiante.

**Definição 4.11** [Forma Normal Prenex] A fórmula  $F$  em lógica de primeira ordem é dita esta em forma normal prenex se e somente a fórmula  $F$  encontra-se segundo o padrão  $(Q_1x_1) \dots (Q_nx_n)(M)$ , onde  $(Q_ix_i)$  pode ser tanto  $(\forall x_i)$  como  $(\exists x_i)$ , e  $M$  uma fórmula sem quantificador algum.  $(Q_1x_1) \dots (Q_nx_n)$  são chamados de prefixo e  $M$  de matriz da fórmula  $F$ .

Deste modo, na Fórmula Normal Prenex, o escopo dos quantificadores deve ser a fórmula inteira. Assim, a fórmula  $(\forall x)(\forall y)(P(x, y) \wedge Q(y))$  e a fórmula  $(\forall x)(\forall y)(\exists z)(Q(x, y) \rightarrow R(z))$  encontram-se na forma normal prenex. As regras de equivalência para normalização previstas na Tabela 4.5 de lógica proposicional continuam válidas para a lógica de primeira ordem. Entretanto, para a lógica de primeira ordem também existem algumas outras regras de equivalência válidas, tal como apresentado na Tabela 4.9, cujas as demonstrações são simples e deixadas para o leitor. Na tabela, considere que  $G$  é uma fórmula que não contém a variável  $x$ , e ainda,  $Q$  pode ser tanto o quantificador  $\forall$ , como o  $\exists$ .

Neste momento, é importante observar que o quantificador universal  $\forall$  e o

# 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](http://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. Sã 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<sup>a</sup> 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<sup>a</sup> 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.