

# Capítulo 6

## Autômatos e Linguagens

O entendimento dos autômatos trata da definição e das propriedades de modelos matemáticos de computação. Estes modelos possuem um papel importante em várias áreas aplicadas da ciência da computação. Um modelo, chamado de autômato finito, é empregado em processadores de texto, compiladores e desenvolvimento de hardware. Outro modelo, chamado de gramáticas livres de contexto, é utilizado em linguagens de programação e inteligência artificial. As teorias de computabilidade e de complexidade exigem a definição precisa do que é um computador. O estudo dos autômatos possibilita adquirir prática com as definições formais de computação introduzindo conceitos importantes neste contexto.

Um autômato finito, também chamado de máquina de estados finitos, é um formalismo que representa de forma inequívoca um processo qualquer composto por um conjunto de estados e de transições entre esses estados. O termo finito é empregado para caracterizar que o autômato pode conter apenas uma quantidade finita e limitada de informação em um dado momento. Esta condição de contorno impõe uma limitação significativa, pois limita seu emprego às linguagens regulares. Os autômatos finitos, conforme será observado nas seções a seguir, podem ser classificados como determinísticos ou não-determinísticos.

A Figura 6.1 apresenta de forma ilustrativa a representação clássica de um autômato finito como sendo uma máquina de estados finitos alimentada por um fita de entrada. Modernamente, pode-se também fazer uma analogia com uma CPU

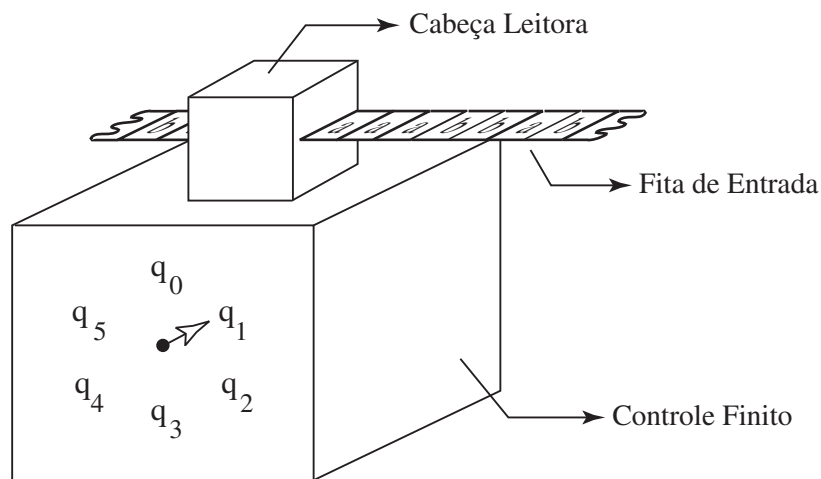


Figura 6.1: Autômato Finito ou Máquina de Estados Finita.

(Unidade Central de Processamento) de um único núcleo com uma só entrada e uma saída. No caso da máquina de fita, ela recebe como entrada uma string através de uma fita de entrada. Cada *slot* desta fita possui um símbolo de um alfabeto qualquer. Esta mesma máquina não produz saída alguma, exceto um indicativo de que a entrada foi aceita ou rejeitada. Além disto, atribui-se a máquina uma ausência permanente de memória volátil.

Inicialmente, o controle finito encontra-se configurado em um estado final chamado  $q_0$ . A cabeça leitora lê um símbolo da fita de entrada, desloca a fita para o próximo *slot*, e o controle finito muda seu indicativo de estado de acordo com o estado atual e o símbolo que acabou de ser lido. Por fim, a cabeça leitora chega ao final da string (fim da fita) e então indica sua aceitação, ou reprovação, quanto ao que leu. Se o estado de parada faz parte do conjunto de estados considerados como estados de aceitação, então se diz que a máquina aceitou a string.

## 6.1 Autômato Finito Determinístico - DFA

**Definição 6.1** Um autômato finito determinístico (*Deterministic Finite Automaton* - DFA) é uma quintupla  $M = \langle k, \Sigma, \delta, s, F \rangle$  onde:

- $k$  é o conjunto finito de ESTADOS;
- $\Sigma$  é um ALFABETO;
- $s \in k$  é o ESTADO INICIAL;
- $F \subseteq k$  é o conjunto de ESTADOS FINAIS;
- $\delta$  é uma FUNÇÃO DE TRANSIÇÃO de  $k \times \Sigma$ .

O termo determinístico é um indicativo de que a função de transição  $\delta$  determina de forma inequívoca o próximo estado  $q'$  a ser assumido quando a máquina  $M$  encontra-se em um estado  $q$  e lê o símbolo  $a$ . Dizemos que  $(q, w) \vdash_M (q', w')$ , isto é, que a máquina  $M$  no estado  $q$  e com uma string  $w$  produz um resultado onde o novo estado é  $q'$  sendo a sobra de string dada por  $w'$  se e somente se  $w = aw'$  para algum símbolo  $a \in \Sigma$  e  $\delta(q, a) = q'$ . Uma produção  $(q, \Lambda)$  significa que  $M$  consumiu todas as entradas, chegando ao fim da string a ser processada, e portanto sua produção acaba neste ponto.

**Definição 6.2** Uma string  $w \in \Sigma^*$  é aceita por  $M$ , se e somente se, há um estado  $q \in F$  tal que  $(s, w) \vdash_M^* (q, \Lambda)$ . Uma linguagem aceita por  $M$ ,  $L(M)$  é o conjunto de todas as strings aceitas por  $M$ .

**Exemplo 6.1** Suponha que  $M$  seja o autômato finito determinístico  $\langle k, \Sigma, \delta, s, F \rangle$  onde:

	$q$	$\sigma$	$\delta(q, \sigma)$
$k = \{q_0, q_1\}$	$q_0$	$a$	$q_0$
$\Sigma = \{a, b\}$	$q_0$	$b$	$q_1$
$s = q_0$	$q_1$	$a$	$q_1$
$F = \{q_0\}$	$q_1$	$b$	$q_0$

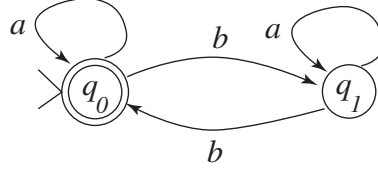


Figura 6.2: Diagrama de estados do Exemplo 6.1.

Seja  $w = aabba$  uma entrada para  $M$ . Assim, tem-se que:

$$\begin{aligned}
 (q_0, \underline{a}abba) &\vdash_M (q_0, \underline{a}bba) \\
 &\vdash_M (q_0, \underline{b}ba) \\
 &\vdash_M (q_1, \underline{b}a) \\
 &\vdash_M (q_0, \underline{a}) \\
 &\vdash_M (q_0, \Lambda)
 \end{aligned}$$

Portanto,  $(q_0, aabba) \vdash_M^* (q_0, \Lambda)$ , e assim a string  $aabba$  é aceita pela máquina  $M$ .

A representação tabular da função de transição não é a mais adequada descrição de uma máquina uma vez que a quantidade de transições pode crescer significativamente. Além disto, esta mesma representação contribui fracamente para o entendimento imediato da tarefa a qual o autômato  $M$  se destina. Deste modo, geralmente é empregada a representação gráfica chamada de diagrama de estados. A Figura 6.2 apresenta o diagrama representativo da mesma máquina  $M$  do Exemplo 6.1.

A Figura 6.2 apresenta algumas peças importantes na construção deste tipo de representação. Os estados  $q_0$  e  $q_1$  são representados por círculos, enquanto que a transição de um estado para outro, mediante a leitura de um símbolo, é indicada pelas setas etiquetadas com um símbolo correspondente. O estado final é caracterizado por dois círculos concêntricos e o estado inicial apresenta o indicativo  $>$ . No caso do exemplo em questão, o estado  $q_0$  coincide como sendo o estado inicial e o único estado final.

Observe que todas as transições propostas pela função de transição do Exemplo 6.1 são contempladas no diagrama da Figura 6.2. Através desta figura, é mais

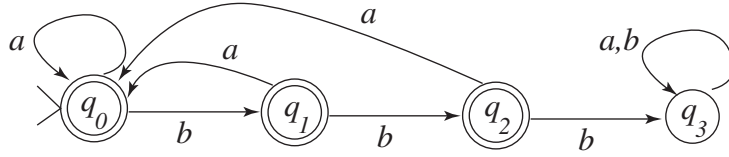


Figura 6.3: Autômato finito determinístico reconhecedor de strings que não contenham 3  $b$ 's consecutivos.

fácil afirmar que a máquina pára no estado final  $q_0$  quando houver um número par de  $b$ 's.

**Exemplo 6.2** Projetar um autômato finito determinístico  $M$  que reconheça a linguagem  $L(M) = \{w \in \{a, b\}^* \mid w \text{ não contém 3 } b\text{'s consecutivos}\}$

$$k = \{q_0, q_1, q_2, q_3\}$$

$$\Sigma = \{a, b\}$$

$$s = q_0$$

$$F = \{q_0, q_1, q_2\}$$

A Figura 6.3 apresenta o autômato correspondente.

**Definição 6.3** Uma linguagem aceita por um DFA é uma linguagem regular.

## 6.2 Autômato Finito Não Determinístico - NFA

Na seção anterior foi observado que os DFA's apresenta como característica mais marcante a transição inequívoca de um estado para outro, isto é,  $\delta(a, q) = q'$ . Os autômatos finitos não determinísticos (NFA), abordados nesta seção, apresentam uma transição que não pode ser definida de modo assertivo. De fato, a função de transição desta espécie de autômato pode apresentar mais de uma opção de estado de destino, ou seja,  $\delta(a, q) = q'$  ou  $q''$  ou  $q''' \dots$ .

Sob esta ótica, pode-se afirmar que o conjunto de todos os autômatos finitos não determinísticos é um super conjunto dos autômatos finitos determinísticos.

# 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 Discrete 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.