



Figura 5.3: Autômato finito.

tem como reconhecedor o autômato finito da Figura 5.3

O estado S é o estado inicial e o estado F é o estado final, dada uma palavra, por exemplo $ababa$. Esta é lida da esquerda para a direita, símbolo a símbolo. Quando lê a a máquina vai para o estado A , quando lê b vai para o estado B , lê a vai para o estado A , lê b vai para B e finalmente lê a e termina no estado A . T é uma armadilha.

A organização das linguagens através desta caracterização sintática é a chamada **Hierarquia de Chomsky**, que é apresentada esquematicamente pela Figura 5.4.

5.5 Gramáticas Sensíveis ao Contexto

Quando Chomsky introduziu o conceito de gramáticas sensíveis ao contexto, ele queria capturar a idéia de que em linguagens naturais, certas palavras podem ser, ou não ser, apropriadas em determinadas posições, de acordo com o contexto onde elas se encontram, isto é, os strings vizinhos a estas palavras. Assim a substituição de um símbolo terminal depende de um string de contexto do seu lado esquerdo e de outro string de contexto do seu lado direito.

Estas mesmas gramáticas sensíveis ao contexto são suficientes para descrever

uma linguagem de programação. Entretanto, na prática elas não são empregadas porque é melhor usar as gramáticas livres de contexto acrescidas de alguns complementos, tais como regras para tipagem, para escopo e para mecanismos de restrição de acesso (métodos `public` \times `private`).

Na Seção 5.4 foi apresentado que gramáticas sensíveis ao contexto $\mathcal{G} = \langle \Sigma, \mathcal{L}, \mathcal{A}, \mathcal{R} \rangle$ são aquelas que possuem como característica possuir regras $\alpha \rightarrow \beta \in \mathcal{R}$ tais que $|\alpha| \leq |\beta|$, onde $|\alpha|$ e $|\beta|$ representam os comprimentos de α e β , respectivamente. Existe uma exceção a esta definição que são as regras do tipo $\alpha \rightarrow \Lambda$, cujo α não pode estar do lado direito de qualquer produção. As linguagens produzidas por este tipo de gramática são reconhecidas por Autômatos Linearmente Limitados. Nesta seção serão abordados alguns teoremas importantes das gramáticas sensíveis ao contexto.

Teorema 5.1 Toda gramática sensível ao contexto é decidível.

Seja $w \in \Sigma^*$ e $|w| = n$. Em qualquer derivação $\alpha_i, i \leq n$ pode-se afirmar com segurança que $\alpha_i \neq \alpha_j$ para qualquer $i < j$. Observe que $|\alpha_i| \leq n$, o que significa que existe uma quantidade finita de derivações, e ainda, que estas seqüências podem ser geradas por funções recursivas primitivas. Por fim, basta verificar dentre todas estas produções qual é aquela que é igual a w . Por outro lado, note que a quantidade de derivações cresce de modo incrivelmente exponencial, o que torna impraticável esta determinação na realidade.

O fechamento de uma operação sobre um conjunto indica que elementos de um conjunto operados por este operador resultam em elementos que também pertencem ao conjunto. Desta forma, torna-se importante conhecer aquelas operações que asseguram as propriedades de elementos de um conjunto tal como as gramáticas de modo geral.

Teorema 5.2 As gramáticas sensíveis ao contexto são fechadas para a união, interseção, complemento, concatenação, estrela de Kleene e reverso.

Uma linguagem $L(G)$ é definida por todas as sentenças que podem ser derivadas por uma combinação de passos começando a partir dos axiomas \mathcal{A} , tal como $\mathcal{A} = \{S\}$. Suponha também que $\mathcal{R} = \{S \rightarrow S\}$. Começando com S e aplicando a re-

gra de produção, obtém-se S . Aplicando a regra duas vezes, novamente obtém-se S . Por indução, aplicando um número finito de vezes a regra de produção, continua-se obtendo S . Desta forma, como nenhuma sentença é obtida, tem-se que $L(G) = \{\}$.

Observe que não é um requisito de uma linguagem que ela termine. Entretanto, quando se trata de uma produção feita por máquina, torna-se importante saber que a produção em algum momento irá parar. Uma forma alternativa para caracterizar que a construção de uma linguagem termina em algum momento é dizer que uma gramática produz linguagens não vazias. O teorema a seguir está inserido neste cenário.

Teorema 5.3 O problema de saber se uma gramática sensível ao contexto gera uma linguagem vazia é indecidível.

5.6 Gramáticas Livres de Contexto

Na Seção 5.4 foi apresentado que gramáticas livres de contexto $\mathcal{G} = \langle \Sigma, \mathcal{L}, \mathcal{A}, \mathcal{R} \rangle$ são aquelas onde $\alpha \rightarrow \beta \in \mathcal{R}$ tal que $\alpha \in N$, além de possuírem as propriedades de gramáticas tipo 1. Nesta seção serão observados alguns aspectos importantes sobre estas gramáticas.

Teorema 5.4 O problema de decidir se uma gramática livre de contexto é ambígua é insolúvel.

Tal como foi discutido na seção 5.5, é importante conhecer as propriedades de fechamento para as gramáticas livres de contexto. Neste sentido, os dois teoremas a seguir apontam para as operações que são fechadas em gramáticas livres de contexto, e as operações que não são fechadas.

Teorema 5.5 As linguagens livres de contexto são fechadas em relação às operações de união, concatenação e estrela de Kleene.

Teorema 5.6 As linguagens livres de contexto não são fechadas em relação às operações de interseção e complemento.

Algumas vezes é necessário ser capaz de avaliar se uma linguagem não é livre de contexto, algo que pode ser feito através do pumping lemma (lema do bombeamento). Este lema atesta que toda linguagem livre de contexto possui um valor especial, chamado de pumping length, tal que todas as cadeias com comprimentos maiores que este valor, podem ser obtidas através de uma espécie de operação de bombeamento. Uma cadeia é dividida em cinco partes (tipicamente $uvxyz$), de tal modo que a segunda (x) e a quarta (y) partes podem ser repetidas quantas vezes se desejar, resultando em uma cadeia que ainda pertence à linguagem.

Lema 5.1 [Pumping Lemma para Linguagens Livres de Contexto] Se $L(G)$ é uma linguagem livre de contexto, então existe um valor p (pumping length) onde, $w \in L(G)$, com $|w| \geq p$, então w pode ser dividido em cinco partes $w = uvxyz$ satisfazendo as condições:

- (a) para $i \geq 0$, $uv^i xy^i z \in L(G)$
- (b) $|vy| > 0$
- (c) $|vxy| \leq p$

Para entender este teorema, observe que o símbolo inicial S de uma gramática G , cujas regras são do tipo $\alpha \rightarrow \beta$, irá derivar uma cadeia com comprimento máximo $|\beta|$, isto é:

$$S \Rightarrow^1 \underbrace{\square}_{|\beta|}$$

Duas derivações a partir do símbolo inicial S irão produzir uma cadeia com comprimento máximo $|\beta|^2$, isto é:

$$S \Rightarrow^2 \underbrace{\underbrace{\square}_{|\beta|} \underbrace{\square}_{|\beta|} \dots \underbrace{\square}_{|\beta|}}_{|\beta| \text{ vezes}}$$

Por indução, h derivações a partir do símbolo inicial S irão produzir uma cadeia com comprimento máximo $|\beta|^h$. Desta forma, se uma cadeia possui comprimento $|\beta|^h + 1$ é porque ela foi derivada no passo $h + 1$.

Seja $\#(N)$ o número de símbolos não terminais de G . Além disso, assuma um valor conhecido com comprimento de bombeamento dado por $p = |\beta|^{\#(N)+1}$.

Se $w \in L(G)$, $|w| \geq p$, então a quantidade de passos para produzir w tem que ser maior que $\#(N) + 1$. Assim, w é derivado com pelo menos $\#(N) + 2$ passos. A produção começa com o símbolo inicial S , seguido de pelo menos $\#(N) + 2$ cadeias. A última cadeia é w e deve conter apenas símbolos terminais. Todas as demais $\#(N) + 1$ cadeias possuem ao menos um símbolo não terminal, caso contrário, não seria possível realizar a derivação da cadeia do passo seguinte. Ora, se G possui $\#(N)$ símbolos não terminais e existem $\#(N) + 1$ produções, então algum símbolo não terminal R tem que ser repetido nas produções de w ⁴.

Se o símbolo não terminal R será repetido é porque o conjunto \mathcal{R} de regras da gramática G precisa ter regras do tipo:

$$\begin{aligned} S &\rightarrow uRz \\ R &\rightarrow vRy \\ R &\rightarrow x \end{aligned}$$

A primeira regra de produção corresponde ao caso mais genérico possível de cadeia derivada, onde o símbolo R pode estar cercado por cadeias quaisquer u e z . Em seguida, se R deve ser repetido, então a regra mais genérica de produção é derivar uma nova cadeia, novamente com R cercado por duas outras cadeias, neste caso por v e y . Por fim, deve haver finalmente uma substituição de R por uma cadeia x . Assim, w deve ser da forma $w = uv^i xy^i z$, $i \geq 0$, como estabelece a condição (a) do lema.

Para obter a condição (b) é necessário que v e y não sejam Λ . Se eles fossem Λ , isso significaria que a produção de w não repetiu R . Isto é uma contradição visto que a construção de w , por definição, exige ao menos uma chamada à regra $R \rightarrow vRy$.

Por fim, observe que a quantidade de passos em que R produz vxy é no máximo $\#(N) + 1$ vezes. Logo, o comprimento máximo da cadeia produzido por essas derivações é no máximo $|\beta|^{\#(N)+1} = p$ e portanto $|vxy| \leq p$, o que satisfaz a condição (c).

⁴Este raciocínio utiliza o *pigeonhole principle* (casa de pombos), que em sua formulação mais simples diz que de $n + 1$ pombos ocupam uma casa de pombos com n buracos, então algum buraco possui ao menos dois pombos.

Exemplo 5.14 Seja $L(G) = \{w = 0^i 1^i 2^i | i \geq 0\}$ e seja p o comprimento de bombeamento. Tomando $w = 0^p 1^p 2^p$, $|w| \geq p$, divide-se w em cinco partes, isto é, $w = uvxyz$. É necessário que $|vxy| \leq p$, logo vxy não pode conter 0 e 2 ao mesmo tempo.

Supondo que o 0 não ocorra em vxy (o caso com o símbolo 2 é análogo) e fazendo $i = 0$, tem-se que o número de 1's e 2's em $w_0 = uv^0xy^0z = uxz$ é menor que $2p$. Além disso, o número de 0's em w_0 tem que ser p , logo $w_0 \notin L(G)$ e L não é livre de contexto.

Exemplo 5.15 Seja $L(G) = \{w = 0^i 10^i 10^i | i \geq 1\}$ e seja p o comprimento de bombeamento. Tomando $w = 0^p 10^p 10^p$, $|w| \geq p$, divide-se w em cinco partes, isto é, $w = uvxyz$. É necessário que $|vxy| \leq p$ e que $|vy| > 0$. Existe dois casos a serem analisados: (1) quando vy não possui 0's; (2) quando vy possui pelo menos um 0.

No primeiro caso, se vy não possui 0's e $|vy| > 0$, então ou $v = 1$, ou $y = 1$. Assumindo que $v = 1$ (para $y = 1$ a análise é análoga), então o bombeamento produziria 1's contíguos, fazendo com que $w \notin L(G)$.

No segundo caso quando existe ao menos um 0 em vy e $|vxy| \leq p$. Assim, vxy é pequeno demais para acomodar todos os três agrupamentos de 0's presentes em $0^p 10^p 10^p$. Assim, $w = uv^0xy^0z = uxz$ possui pelo menos um bloco com p 0's e pelo menos um bloco com menos de p 0's. Logo, $uwz \notin L(G)$.

5.7 Gramáticas Regulares

Na Seção 5.4 foi definido que gramáticas regulares $\mathcal{G} = \langle \Sigma, \mathcal{L}, \mathcal{A}, \mathcal{R} \rangle$ são aquelas cujas regras de produção são da forma $A \rightarrow a$, $a \in T$ e $A \in N$ e $A \rightarrow aB$, $a \in T$ e $A, B \in N$, além de possuírem as propriedades de gramáticas tipo 2.

Desta forma, pode-se contruir uma gramática regular $G3 = \langle \Sigma_3, \mathcal{L}_3, \mathcal{A}_3, \mathcal{R}_3 \rangle$ obtida a partir da concatenação das gramáticas $G1 = \langle \Sigma_1, \mathcal{L}_1, \mathcal{A}_1, \mathcal{R}_1 \rangle$ e $G2 = \langle \Sigma_2, \mathcal{L}_2, \mathcal{A}_2, \mathcal{R}_2 \rangle$ tal que $L(G3) = \{ab | a \in L(G1), b \in L(G2)\}$. Nesta gramática tem-se que:

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. 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^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.