

Uma proposta de aplicação da Tecnologia Adaptativa na Teoria Algorítmica do Aprendizado

R. I. Silva Filho, R. L. A. Rocha

1

Abstract— In this preliminary work, we propose an architecture of machine learning to use the adaptive technology and the model of identification in limit applied in inference processes.

I. INTRODUÇÃO

Este trabalho tem como objetivo apresentar o atual estágio de desenvolvimento da linha de pesquisa do autor, que é propor um modelo de aprendizado algorítmico que utilize a tecnologia adaptativa. De maneira mais específica, iremos estudar a viabilidade de uma descrição do modelo de identificação no limite utilizando o modelo dos autômatos adaptativos. O modelo de identificação no limite é um tipo particular de modelo de computação não convencional proposto por Mark Gold e que hoje faz parte da área de estudo da Teoria Algorítmica do aprendizado (*Algorithmic Learning Theory*).

O trabalho está organizado da seguinte forma. Na segunda sessão, temos uma apresentação das motivações em que se baseia este trabalho. Tais motivações têm um caráter multidisciplinar com origem em áreas como a Linguística, a Filosofia da Mente e a Inteligência Artificial.

A terceira sessão introduz a Teoria Algorítmica do Aprendizado. Trata-se de um sumário que contém o essencial para esclarecer o contexto teórico específico. Aqui serão apresentados os principais termos e definições relacionados com a inferência indutiva.

A quarta sessão trata da identificação no limite, o critério de aprendizagem assintótica para a identificação de regras desconhecidas. Uma verificação de sua relação com a Tecnologia Adaptativa é vista na quinta sessão, onde também é proposto um esboço para uma arquitetura que envolva as duas áreas de estudo. As consequências e possíveis desdobramentos futuros deste trabalho preliminar serão discutidos na conclusão.

A. Nomeclatura

Os conceitos básicos da teoria dos autômatos e dos autômatos adaptativos, bem como os aspectos gerais relacionados com a teoria das linguagens formais são considerados como

conhecidos. Como referência para esses temas, indicamos [7], [10]. Todo autômato é definido em função de um conjunto Σ finito de símbolos denominado alfabeto. Toda linguagem é definida na forma $L_i \subseteq \Sigma^*$, onde o símbolo "*" representa o fecho transitivo e reflexivo (também chamado de estrela de Kleene). A expressão $\{R_l\}_{l=1}^L$ representa o conjunto das regras genéricas (R_1, R_2, \dots, R_L) com $L \in \mathbb{N}$. O símbolo # representa uma pausa e não é um elemento do conjunto Σ . O conjunto \mathbb{N} é o conjunto dos números naturais

II. MOTIVAÇÃO

Nossa motivação parte de uma hipótese: o ser humano é uma máquina ótima para a busca de soluções[11]. Se um indivíduo é confrontado com um determinado problema, sua reação segue um padrão estabelecido por sua intuição, experiência e conhecimento formal. Partindo do pressuposto de que todo indivíduo possui uma representação interna do mundo que o cerca, e de que tal representação guarda fatos e permite ao indivíduo realizar previsões sobre os comportamentos futuros desse ambiente, assumimos que tal representação pode sofrer mudanças e estar sujeita a adaptações em função dos estímulos externos. Essas mudanças buscam proporcionar ao indivíduo uma "visão" mais completa dos comportamentos que regem o ambiente onde ele se insere. Assim, a representação do mundo disponível ao sujeito nunca (e este é um ponto crucial) será completa. Mais ainda: a representação será dinâmica e sujeita a um processo de construção ao longo da vida do indivíduo. Desta forma, o indivíduo procura "cobrir" a realidade que o cerca usando os recursos de representação interna que ele possui.

Inicialmente, vamos abstrair questões relacionadas à formação e a natureza da memória. Nossa abordagem consiste em ter uma representação do "conhecimento" como o objeto sujeito a ser alterado pelos comportamentos adaptativo e indutivo que veremos descritos nas sessões seguintes. É conveniente esclarecer que o propósito aqui não é criar um modelo de arquitetura cognitiva. Nosso objetivo é desenvolver uma arquitetura para o aprendizado de máquina que possua atributos semelhantes ao aprendizado humano. Porém, diferente do ponto de vista conexionista[2], não adotaremos uma abordagem estrutural e sim funcional[12], ou seja, não incorporaremos os atributos de interesse da aprendizagem humana propondo um modelo que imite as estruturas que compõem o cérebro. Nossa busca baseia-se em *modelos computacionais*. Nosso interesse é por representações comportamentais que nos disponibilizem as

¹ R. I. Santos Filho é pesquisador e doutorando no Laboratório de Linguagens e Técnicas Adaptativas.

R. L. A. Rocha é pesquisador e professor Doutor no Laboratório de Linguagens e Técnicas Adaptativas.

características que queremos ver incorporadas em nossa proposta de aprendizado de máquina, mesmo que estruturalmente tais modelos computacionais não lembrem qualquer elemento neurofisiológico relacionado com a cognição.

Assim, assumimos que já temos em mãos um objeto que se constitui um modelo daquilo que iremos acatar como o "conhecimento". No nosso caso, esse modelo serão as linguagens formais. Da mesma forma que não é nossa finalidade explicar a estrutura neurofisiológica envolvida no aprendizado, não nos preocuparemos em abordar como o conhecimento tem seu início partir do surgimento do indivíduo. Apenas assumimos que a nossa arquitetura já conta com um "conhecimento inato" e prosseguiremos analisando como a estrutura que representa esse conhecimento sofre as dinâmicas descritas ao longo deste trabalho.

Por enquanto, o importante é destacar que, por influência dessas dinâmicas, o indivíduo experimenta, ao longo de sua vida, a ação de inferir maneiras mais corretas de predição e construção do conjunto de regras $\{R_i\}_{i=1}^L$ que governa o ambiente onde ele está inserido. Assim, dependendo do momento ao qual o indivíduo foi exposto a um determinado problema, sua reação pode ser diferente da reação que seria tomada em relação a qualquer outro momento, tanto no passado quanto no futuro. Tais considerações estão ilustradas na figura abaixo. Neste diagrama conceitual, observamos uma linha diagonal em que o conhecimento \mathcal{M} do indivíduo sofre alterações em função dos estímulos (e_1, e_2, \dots, e_n) que ocorrem em diferentes momentos no tempo. \mathcal{M} guarda a representação \mathcal{R} das regras do ambiente, onde $\mathcal{R} \subseteq \{R_i\}_{i=1}^L$. Se pudéssemos "congelar" no tempo cada instância de \mathcal{M} antes desta mudar, veríamos as diferentes maneiras de como ele reagiria, em diferentes momentos da sua história. Assim, as linhas verticais indicam o comportamento de uma mesma instância de \mathcal{M} sujeita a estímulos diferentes, caso \mathcal{M} fosse estático, enquanto que as linhas horizontais indicam os comportamentos das diferentes instâncias de \mathcal{M} , sujeitas ao mesmo estímulo.

III. INFERÊNCIA INDUTIVA

Se uma criança cresce observando um número cada vez maior de sentenças da sua língua nativa, eventualmente desenvolve uma gramática que converge para um gerador de sentenças válidas dessa linguagem [14]. Do ponto de vista do diagrama ilustrado na Fig. 1, observaríamos que as

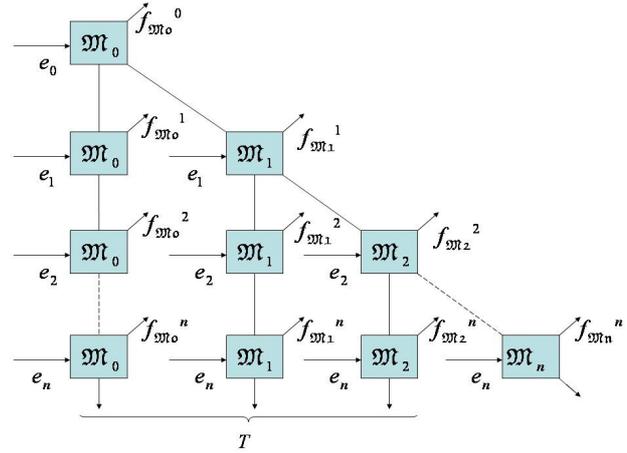


Figura 1: Comportamento do aprendizado de um indivíduo. As flechas diagonais representam a resposta do indivíduo frente a um estímulo e_i .

reações e o discurso da criança se aprimoram à medida que ela é submetida a novas construções sintáticas e a novos exemplos de vocabulário, de modo a inferir indutivamente as regras pertinentes à linguagem nativa.

O termo inferência indutiva denota a geração e a escolha de hipóteses sobre as regras gerais que governam um determinado domínio, a partir de um fluxo de exemplos disponíveis ao longo do tempo [1]. A Inferência Indutiva é um *framework* de aprendizado, em que "aprendizado" é entendido como a capacidade de identificar um determinado conjunto dentro de uma classe de escolhas possíveis [1].

A inferência indutiva centraliza sua atenção na figura do aprendiz. Podemos utilizar a criança mencionada anteriormente como exemplo de um aprendiz. No processo de aprendizado, o aprendiz recebe um fluxo de dados sobre uma linguagem desconhecida para ele. De posse desses dados, ele, de tempos em tempos, gera e modifica suas hipóteses acerca do funcionamento dessa linguagem.

Um problema de inferência indutiva é visto como a quádrupla $I = \langle \mathcal{R}, \mathcal{H}, \mathcal{T}, \mathcal{J} \rangle$ [13].

\mathcal{R} é representação das regras do ambiente. Para nós, toda linguagem formal L será uma representação válida.

$\mathcal{H} = (\phi_1, \phi_2, \phi_3, \dots)$ é o espaço de hipóteses, consistindo de um conjunto de representações tal que, para cada regra, existe pelo menos uma hipótese que a represente nesse espaço. O nosso espaço de hipóteses será constituído por sistemas programáveis fixados, ou seja, uma enumeração de todas Máquinas de Turing. Porém, também é possível utilizar programas LISP, Gramáticas, etc. Dessa forma, cada ϕ_i denota uma função recursiva parcial ou total.

O conjunto $\mathcal{T} = \{\eta_a \in L : a \in \mathbb{N}\}$ é o fluxo de dados ao qual o aprendiz tem acesso e representa os exemplos ou textos de uma linguagem L. Os exemplos são uma apresentação admissível da própria linguagem, desconhecida pelo aprendiz.

A Teoria Algorítmica do Aprendizado abstrai a figura do aprendiz na forma de um dispositivo recursivo \mathcal{A} que recebe

como entrada $(\eta_a \cup \{\#\})^*$, onde o símbolo $\#$ representa uma pausa e, de tempos em tempos, apresenta como uma saída um elemento do conjunto \mathfrak{H} . Um aprendiz \mathcal{A} pode ser uma função parcial ou total. Não é necessário que \mathcal{A} seja capaz de aprender toda e qualquer linguagem, mas ele deve ser capaz de lidar com uma coleção \mathcal{S} de classes de linguagens.

Uma classe de linguagens pode ser aprendida se e somente se existe um aprendiz tal que, para um fluxo de textos apresentados, \mathcal{A} gera as hipóteses corretas sobre o comportamento da linguagem pertencente à classe. Grande parte dos esforços realizados no âmbito do estudo da Teoria Algorítmica do Aprendizado está na análise das classes \mathcal{S} . Existem três tipos de classes envolvidas:

- **Classe recursiva.** A classe é descrita como uma lista $\mathbb{S}_{L_r} = (L_0, L_1, \dots)$ de linguagens onde $\{(e, x) : x \in L_e\}$ é recursiva.

- **Classe recursivamente enumerável.** A classe é descrita como uma lista $\mathbb{S}_{L_{re}} = (L_0, L_1, \dots)$ de linguagens onde $\{(e, x) : x \in L_e\}$ é recursivamente enumerável.

- **Classe geral.** A classe é descrita como uma lista $\mathbb{S}_{L_g} = (L_0, L_1, \dots)$ de linguagens.

Por fim, \mathcal{J} é um critério de aprendizagem e refere-se à estratégia de produções das hipóteses pelo aprendiz em relação aos exemplos apresentados. O critério adotado neste trabalho é aquele criado por Gold[6], denominado de Identificação no Limite.

IV. IDENTIFICAÇÃO O LIMITE

Critérios de Aprendizado. Um critério de aprendizagem se refere ao modo como a sequência de hipóteses de saída será construída em relação aos exemplos recebidos até então. A lista abaixo menciona os principais, pois a literatura apresenta outras variações[5].

- **Aprendizado Explanatório:** O aprendiz \mathcal{A} gera uma sequência $\Phi \subset \mathfrak{H}$ finita de hipótese que converge para um índice i , que se constitui na resposta correta. O tipo de convergência nesse critério é chamado de convergência sintática[6].

- **Aprendizado Comportamentalmente Correto,** ou Aprendizado no Limite: Este critério exige que a semântica (e não a sintaxe) das hipóteses convirjam corretamente no limite. O processo de inferência pode continuar modificando as suas hipóteses. \mathcal{A} gera uma sequência Φ infinita de hipóteses que, após um índice i , são todas descrições corretas de L . A diferença entre os dois critérios é que, no primeiro, as hipóteses param de mudar e a linguagem L é computada pela última hipótese, enquanto que, no segundo critério, as hipóteses não param de mudar; é necessário apenas que elas reconheçam L pelos seus exemplos[4].

Podemos verificar, portanto, que a garantia de um critério de convergência é extremamente importante qualquer que seja o critério de aprendizado envolvido. Assim, é necessário que o

aprendiz convirja sintaticamente ou semanticamente em função dos exemplos apresentados. Textos que pertencessem a linguagens fora da classe a ser aprendida fariam o aprendiz convergir para hipóteses erradas. As razões para isso residem no fato de existir um número incontável de linguagens diante de um número contável de índices.

V. IDENTIFICAÇÃO O LIMITE E A TECNOLOGIA ADAPTATIVA

Vamos tratar agora da questão do funcionamento do aprendiz. Apesar do aprendiz ser visto até agora como uma caixa-preta, existe um método universal para implementá-lo denominado de identificação por enumeração[5]. Ele se baseia na geração de uma lista da enumeração computável de todas as descrições d_1, d_2, d_3, \dots de todas as linguagens L_i de uma classe \mathbb{S} , de modo que cada linguagem da classe tem pelo menos uma ou mais descrições dentro da enumeração.

Dada uma coleção T de textos para essa linguagem, o método irá vasculhar a lista de descrições com o intuito de encontrar a primeira descrição d_i compatível com os exemplos dados (a verificação dessa compatibilidade também deve ser computável), transformando a descrição d_i em um elemento de Φ . O critério de convergência para esse método se baseia em duas condições. Primeiro, uma hipótese correta é sempre compatível com os exemplos dados. A segunda condição é a de que uma hipótese incorreta é incompatível com uma coleção suficientemente grande de exemplos para todos os textos dados[1]. A Fig. 2 mostra uma ilustração que representa a dinâmica desse método.

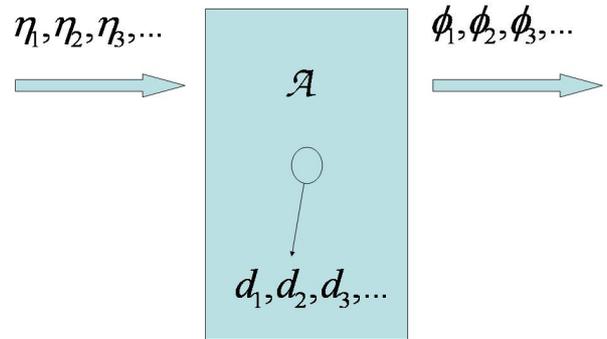


Figura 2: Método de identificação por enumeração. Uma nova hipótese escolhida é mantida enquanto não entrar em conflito com os exemplos.

Apesar de ser genérica e poderosa o bastante, a identificação por enumeração pode se tornar impraticável dependendo do tamanho da classe de linguagens envolvida. Em função desse fato, vamos buscar uma proposta alternativa para a identificação por enumeração. Façamos a seguinte conjectura: suponhamos a existência de uma estrutura U . Para um dado conjunto Σ , a estrutura U possui a capacidade de automodificar-se, de forma que, para cada

modificação efetuada, ela gera um novo reconhecedor R_i para uma linguagem $L_i \subset \Sigma^*$ diferente, pertencente a uma determinada classe \mathbb{S} .

Vamos supor também que esta estrutura U possui internamente um componente associado, um módulo o qual chamaremos de ζ . Entre suas atribuições, este módulo possui a capacidade de "ler" os elementos $\eta_a \in T$ e induzir as modificações em U .

Neste ponto, poderíamos fazer a seguinte pergunta: a estrutura U é um dispositivo adaptativo? A resposta envolve a questão da leitura dos exemplos pelo módulo ζ . Se afirmarmos que o módulo induz as alterações na estrutura U em função do texto lido, então, pela própria definição, a resposta é afirmativa, de acordo com [8] a estrutura U é um dispositivo adaptativo.

Vamos fazer agora uma pausa para discutir um pouco mais acerca das hipóteses geradas por um aprendiz genérico \mathcal{A} . Como foi visto antes, independentemente do critério de aprendizagem, da classe de linguagens ou do critério de aprendizagem e convergência, temos as hipóteses geradas definidas como uma lista de sistemas programáveis fixados. Assim, utilizando a equivalência entre Autômatos Adaptativos e as Máquinas de Turing mencionadas em [9], podemos definir, sem perda de generalidade, o conjunto de todas as hipóteses geradas por um aprendiz como sendo a lista, potencialmente infinita, denominada $\Phi_A = (a_1, a_2, a_3, \dots)$, composta por uma enumeração de Autômatos Adaptativos, ou seja: sem perda de generalidade, podemos utilizar os autômatos adaptativos como uma representação do nosso conjunto de hipóteses. Usaremos esse resultado adiante.

De volta à descrição sobre a estrutura U , foi dito que, entre duas automodificações consecutivas, a estrutura U é um reconhecedor para uma linguagem específica. O conjunto de automodificações poderia ser arbitrário, mas convencionou-se que toda modificação de U resultaria em reconhecedores para linguagens diferentes, porém dentro de uma mesma classe. Se o elemento do texto não for reconhecido pela atual configuração da estrutura, o módulo ζ requisita uma modificação de U que "se adapta" de forma que a nova configuração identifica o exemplo apresentado como pertencente à linguagem. Quando o processo de adaptação se completa, o módulo ζ apresenta como saída o reconhecedor R adaptado para reconhecer todos os exemplos anteriormente apresentados, mais o último que foi responsável pela sua transformação. A Fig. 3 apresenta um diagrama de blocos para a estrutura U .

Já sabemos que U é um dispositivo adaptativo *por definição*. Poderíamos então perguntar: é possível reproduzir o comportamento de U utilizando puramente o modelo dos Autômatos Adaptativos? Caso não seja possível, quais novos elementos teríamos que incluir para obter uma solução que se comporte como a estrutura U ?

Os autômatos adaptativos são um modelo de computação [9]. A partir do momento em que apresentamos a opção da adaptatividade para a identificação por enumeração, é fundamental possuímos um modelo formal, tanto para

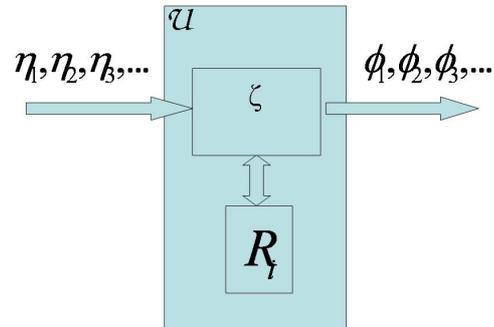


Figura 3: Diagrama de blocos da Estrutura U proposta

análise quanto para implementação.

Para respondermos se o modelo dos autômatos adaptativos pode ou não ser utilizado, vamos imaginar o seguinte cenário: um programador possui um arquivo em um determinado formato. Esse arquivo possui a seguinte série de *strings*:

```
1000000001
1100000011
1110000111
1111001111
1111111111
```

O programador é orientado pelo seu superior a desenvolver um reconhecedor para a linguagem representada pelas *strings*, utilizando para isso os Autômatos Adaptativos. O seu progresso será monitorado através de um repositório onde o programador guarda o código produzido. Caso a versão guardada lá funcione para todas as *strings* contidas no arquivo dado, então um novo conjunto de exemplos será fornecido em um novo arquivo (iremos admitir que o número de arquivos é finito). Para facilitar o trabalho do desenvolvedor, são dadas duas informações: todas as *strings* contidas nos arquivos são exemplos válidos da linguagem dada e a linguagem em questão é $L = 1^n 0^* 1^n$ com $n \geq 0$. Com essas informações, o programador parte para o desenvolvimento do reconhecedor. Obviamente, de posse dessas informações, ele irá conseguir projetar o Autômato Adaptativo para tratar dessa situação.

Porém, o que aconteceria se ele não recebesse a descrição da linguagem, de maneira a ter somente os exemplos dados no arquivo? Existiria uma maneira do programador descobrir a linguagem somente com base nos exemplos e assim construir o seu reconhecedor baseado em Autômatos Adaptativos?

O programador inicialmente analisaria, de alguma forma, os exemplos fornecidos, já essa é a única informação que ele possui. Em algum momento, ele faria uma suposição: "*provavelmente*, a linguagem representada pelas string é: $L = w w^{-1}$."

Ele percebe, ao implementar o protótipo do reconhecedor, que essa suposição foi muito boa, pois todas as *strings* do arquivo foram processadas com sucesso. Assim, ele armazena seu reconhecedor no repositório e aguarda alguma ação por parte do seu superior. Quando recebe a indicação de que o repositório foi atualizado, o superior libera a segunda massa de exemplos. Ele não avalia o código gerado, pois essa não é a sua função. Cabe ao programador essa tarefa.

O segundo arquivo chega com os novos dados:

```
0000000000
00000000
0000
00
0
```

O programador fornece as *strings* ao seu reconhecedor, porém, durante o processamento dos exemplos, a última *string* composta pelo símbolo "0" não é aceita. Caso ele não mude de suposição, ou seja, não implemente outro Autômato Adaptativo com um conjunto de funções adaptativas diferentes, seu programa não reconhecerá por completo todo o conjunto de novos exemplos dados.

Assim, com esse exemplo, ilustramos o seguinte fato: o modelo dos Autômatos Adaptativos por si só não pode alterar a linguagem para a qual ele está codificado para reconhecer. As funções adaptativas podem alterar o autômato subjacente de modo que este reconheça as cadeias das quais ele não possuía configurações de aceitação anteriormente, porém essa adaptação se dará apenas para as palavras de uma mesma linguagem fixada e codificada nas funções adaptativas. Mudar a linguagem aceita pelo Autômato Adaptativo implica, no mínimo, alterar o conjunto de funções adaptativas presente no autômato.

Assim, o programador do nosso exemplo teria que iniciar uma nova tentativa, mudando as funções adaptativas do seu autômato. A cada nova suposição, seu superior veria uma nova atualização no repositório, cada uma com sua data de criação. Ao final, quando o programador convergisse para linguagem representada pelas *strings*, seu superior teria uma enumeração de todas as versões de reconhedores criadas.

VI. CONCLUSÃO

Com esse cenário, podemos responder a pergunta feita anteriormente: de fato, o modelo dos Autômatos Adaptativos não é o suficiente para modelarmos toda a estrutura U . Para ser mais específico, não podemos reduzir toda a estrutura U ao modelo dos Autômato Adaptativo.

Embora o módulo ζ da arquitetura não pertença ao modelo dos Autômatos Adaptativos, as instâncias R_i que são as funções reconhedoras podem, sim, ser Autômatos Adaptativos. Dessa forma, as ações adaptativas em cada R_i também estariam sujeitas a um processo adaptativo, pois o módulo ζ teria a capacidade de alterar as funções

adaptativas em cada R_i com o objetivo de gerar novas hipóteses.

Assim, temos um conceito novo com que trabalhar: a Adaptatividade de Segunda Ordem. Nela, temos dispositivos adaptativos que também estão sujeitos a sofrer automodificações. A adaptatividade de segunda ordem pode auxiliar na formação de linguagens onde a única informação disponível são os exemplos apresentados a um aprendiz. Na adaptatividade de segunda ordem, cada automodificação gera um Autômato Adaptativo para uma nova linguagem, com o intuito de adaptar-se aos exemplos apresentados e convergir para a linguagem representada por esses mesmos exemplos.

Com esse trabalho, determinamos em que nível de adaptatividade devemos trabalhar. As etapas seguintes irão concentrar-se em detalhar da arquitetura aqui sugerida e obter resultados sobre sua validação. Esse rumo implicará uma séria de questões a serem respondidas, referentes às classes de linguagens que a arquitetura pode tratar, suas limitações e características.

VII. REFERÊNCIAS

- [1] Angluin, D. e Smith, C. H. 1983. *Inductive Inference: Theory and ethods*. ACM Comput. Surv. 15, 1983.
- [2] Arbib, M. *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1997.
- [3] Case, J. e Jain, S. e Manguelle, N. *Refinements of inductive inference by Popperian and reliable machines*. CJNM94, Kybernetika 30, pp. 12--52, 1994.
- [4] Case, J., Lynes, C. Machine inductive inference and language identification. *Proceedings of the ninth International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science 140:107--115, 1982.
- [5] Florencio, C.C. *Consistent identification in the limit of any of the classes k-valued is NP-hard*. In: Proceedings of the Conference on Logical Aspects of Computational Linguistics LACL, volume 2099 of LNCS/LNAI, Springer-Verlag, pp 125--138, 2001.
- [6] Gold, E. M. *Language identification in the limit*. Information and Control, 10, pp 447--474, 1967.
- [7] Lewis, H. and Papadimitriou, C. *Elements of the Theory of Computation*. Prentice-Hall, 1997.
- [8] Neto, J. J. *Adaptive Rule-Driven Devices - General Formulation and Case Study*. Lecture Notes in Computer Science. Implementation and Application of Automata 6th International Conference, CIAA 2001, Vol.2494, Pretoria, South Africa, Springer-Verlag, pp. 234--250, 2001.
- [9] Rocha, R. L. A. e Neto, J. J. *Autômato adaptativo, limites e complexidade em comparação com máquina de Turing*. In: Proceedings of the second Congress of Logic Applied to Technology - LAPTEC 2000. São Paulo: Faculdade SENAC de Ciências Exatas e Tecnologia, pp. 33--48, 2001.
- [10] Rozenberg G e Salomaa, A. editors. *Handbook of Formal Languages*. Berlin: Springer, 1997.
- [11] Searle, J. R. *The mystery of consciousness*. New York: New York Review of Books, 1997.
- [12] Stuart J. Russell e Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.
- [13] Zilles, S. On the Synthesis of Strategies Identifying Recursive Functions In: Proceedings of the 14th Annual Conference on Computational Learning theory and 5th European Conference on Computational Learning theory. Lecture Notes In Computer Science, vol. 2111. Springer-Verlag, London, pp 160-176, 2001.
- [14] Yang, C. D. Universal Grammar, statistics or both?, Trends in Cognitive Sciences. Volume 8, Issue 10, pp 451-456, 2004.