Uma proposta de método adaptativo para a seleção automática de soluções

Ricardo Luis de Azevedo da Rocha

Escola Politécnica da Universidade de São Paulo
Av. Luciano Gualberto, trav. 3, n.158 Cidade Universitária
CEP 05508-900 – São Paulo – Brasil
e-mail: rlarocha@usp.br

João José Neto

Escola Politécnica da Universidade de São Paulo
Av. Luciano Gualberto, trav. 3, n.158 Cidade Universitária
CEP 05508-900 – São Paulo – Brasil
e-mail: jjneto@pcs.usp.br

Uma proposta de método adaptativo para a seleção automática de soluções

Ricardo Luis de Azevedo da Rocha; João José Neto

Escola Politécnica da Universidade de São Paulo
Av. Luciano Gualberto, trav. 3, n.158 Cidade Universitária
CEP 05508 – São Paulo – Brasil
e-mail: rlarocha@usp.br; jjneto@pcs.usp.br

RESUMO

Esta pesquisa busca propor uma formulação para um método automatizado de escolha de soluções de problemas, para uso como um dispositivo computacional.
O substrato formal utilizado como base na proposição do método e do dispositivo é o autômato adaptativo.

ABSTRACT

This research proposes a formulation for an automated method for choosing among problem solutions, regarding its use as a computational tool.
The adaptive automaton has been chosen as the underlying theoretical framework for the method and for the tool suggested in this proposal.

1. INTRODUÇÃO

Resolver problemas é uma tarefa que aflige o ser humano desde quando este, ao tornar-se consciente de sua existência e da existência do mundo físico, deparou-se com a necessidade de sobrevivência em um ambiente hostil e competitivo. Segundo Darwin, a sobrevivência se dá entre os mais adaptados ao ambiente [11]. Entretanto, pode-se presumir que a tarefa de adaptação não seja estática e absolutamente inata, isto é, uma vez que um indivíduo de determinada espécie nasce, não há determinismo biológico sobre seu destino. Há portanto possibilidades de sobrevivência ou de morte, de acordo com os eventos associados à sua vida e com a sua capacidade de lidar com tais eventos.

1.1. Visão do problema e evolução

Em termos filosóficos, muito se tem estudado a respeito de consciência, e a maior dificuldade é conceitual, ontológica, já que é possível estudar os efeitos da consciência, embora não seja possível estudar sua origem, porque não é uma experiência vivida em terceira pessoa [9]. O filósofo Karl Popper propôs um modelo mental no qual a consciência desempenha papel preponderante, porém não fornece explicação ontológica para a mesma [6]. Os críticos da idéia de construir máquinas inteligentes apegam-se à dificuldade de se definir consciência, pela sua própria natureza, para negar a possibilidade de construção de tais máquinas. John Searle afirma ser a consciência uma capacidade oriunda de cérebros constituídos por neurônios biológicos, já que os processos cerebrais são bio-físico-químicos. Seria, portanto, vedada consciência a cérebros de silício, a não ser que se conseguisse reproduzir nestes os mesmos fenômenos bio-físico-químicos encontrados nos cérebros biológicos [9].

Após o estudo desses aspectos bastante intrigantes, a motivação para este trabalho tornou-se evidente: tentar construir (ainda que teoricamente) algum dispositivo que busque resolver problemas de maneira similar ao ser humano.

Inicialmente, buscou-se dentro do conjunto de teorias e conceitos computacionais algum modelo teórico que apresentasse, a priori, alguma ligação ou mesmo semelhança com a forma através da qual os processos mentais se desencadeiam. Os trabalhos de McCulloch e Pitts,

ampliados por Minsky [5] na área de redes neurais forneceram uma noção bastante clara do nível de dificuldade a ser enfrentado e fixaram mais um conceito que intuitivamente parecia necessário - a influência do grau de paralelismo do dispositivo sobre a eficiência do mesmo.

Aliada ao conceito de redes neurais e bastante comprometida com ele, está a teoria de autômatos, da qual o modelo mais simples (autômato finito) foi utilizado para implementar, em termos computacionais, as primeiras redes neurais [5]. Porém, trabalhos recentes trouxeram luz novamente à teoria, colocando-a em foco. Estes trabalhos formalizam um modelo de autômato que é capaz de modificar sua estrutura topológica, de maneira a resolver um problema (aceitar uma cadeia), adaptando-se às diferentes necessidades dele suscitadas [2]. O modelo de autômato, assim formulado, possui poder computacional equivalente à máquina de Turing [7]. Entretanto, a inserção de modificações estruturais nos modelos de autômato somente é possível se estiver devidamente prevista na estrutura inicialmente concebida, e se for implementada por meio de uma função dentro do modelo (não pode ter surgido no modelo de forma autônoma).

Na teoria da computação, dois outros modelos atualmente em voga trabalham com a possibilidade de adaptação: os algoritmos genéticos [11] (que reproduzem, em termos computacionais, o funcionamento do mecanismo de seleção natural dos mais aptos encontrado nos seres vivos, cujas células reprodutoras sofrem recombinação genética e mutação) e os agentes computacionais [12] [8] (sistema baseado em computador, que opera de forma autônoma, independente, sendo capaz de tomar decisões.). No caso dos algoritmos genéticos, é prevista a existência de uma função de medição (construída anteriormente à execução do modelo) que permite estabelecer um critério de comparação entre diversos modelos de solução gerados pelo algoritmo genético e, desta forma, proporcionando meios para a escolha do melhor. Apesar da construção dos modelos imitar alguns dos preceitos encontrados na natureza, como é o caso da recombinação e da mutação genética para construir novos modelos, a elaboração da função de medição ainda se mostra um problema difícil.

Por outro lado, os agentes computacionais têm por prerrogativa de existência o conhecimento, a crença. Desta forma, um agente pode, então, estabelecer autonomamente suas ações frente aos problemas que lhe são apresentados.

Ao observar os modelos computacionais, notou-se que para a proposta desta pesquisa, nenhum deles seria o modelo ideal, embora algumas características encontradas em cada um se mostrassem bastante úteis e, em alguns casos, necessárias (como a capacidade de auto-modificação). A primeira sugestão razoável foi a de constituir um novo modelo e um método de construção, a partir dos modelos observados e dos métodos de construção dos mesmos.

A partir desta sugestão, foi realizado um estudo comparativo dos modelos observados, buscando-se estabelecer o conjunto ideal de parâmetros para a geração do modelo final híbrido. Entretanto, o estudo comparativo não mostrou como resolver autonomamente o problema de geração de soluções, já que em todos os modelos a solução já deve estar embutida ou, ao menos inferida (através de uma função de medição). Em [5], há a afirmação de que um dispositivo é capaz de aprender tudo o que pode representar. A solução de problemas não é um caso diferente do aprendizado, é um problema inverso. Assim, neste ponto, o caminho seguido foi passar por um estudo de complexidade, um estudo sobre o valor de cada informação em si.

O raciocínio utilizado foi o seguinte: se um indivíduo é confrontado com determinado problema, sua reação segue um padrão estabelecido por seus pressupostos, pela sua história de vida, sua experiência, seu conhecimento e, claro, sua intuição e sua predisposição genética. Isto posto, dependendo do momento de sua vida no qual o indivíduo foi exposto ao problema, sua reação poderá ser diferente da que seria tomada em relação ao mesmo, em qualquer outro momento.

Entretanto, como não é possível ter controle sobre o instante de vida no qual um indivíduo é confrontado com um problema, sua resposta mais provável é fornecida em termos de uma escolha dentre as possíveis respostas no instante em questão, ou seja, leva em consideração todas as instâncias até o instante. Esta escolha é uma função probabilística das possíveis respostas de todas as suas instâncias até o momento do confronto, porque não se pode determinar que eventos ocorreram antes do instante de tempo. Como a duração é finita, e a experiência (memória dos eventos) ajuda a escolher, restava descobrir como determinar a probabilidade da resposta para que a função de medição estivesse teoricamente completa.

Assim, uma teoria que engloba algoritmos, complexidade, probabilidade e medida de informação, a chamada *probabilidade de algoritmos* de Ray Solomonoff [10], foi resgatada.

1.2. Objetivos

A proposta concreta desta pesquisa é o desenvolvimento de um método de construção de modelos e resolução de problemas complexos utilizando um formalismo adaptativo como base. O objetivo principal é encontrar soluções para problemas computacionais, e sendo mais preciso, para aqueles problemas computacionais que possam ser transformados em problemas de linguagem (Esta escolha deve-se à necessidade de limitar a abrangência das informações recebidas e tratadas pelo dispositivo gerado). Para isso, estuda-se comparativamente alguns dos métodos e técnicas de construção e resolução de modelos correntemente adotados, e utiliza-se características destes no intuito de propor um método alternativo, preservando as características estudadas e consideradas importantes.

Além desses objetivos imediatos, pode-se ressaltar também a utilização da teoria de autômatos, retomando um caminho percorrido anteriormente pelos pioneiros na área de inteligência artificial [8] [5], utilizando como base para o modelo de solução o autômato adaptativo no lugar do finito, já que propicia um tratamento mais adequado e uniforme às questões computacionais [2].

Outro objetivo é propor um mecanismo de busca de soluções, que se adeque à metodologia proposta. Este mecanismo deve ser simulado em um computador, isto é, deve ser um mecanismo de software. Um protótipo para o mecanismo é construído e testado posteriormente, de modo a validar as hipóteses da proposta. Assim, o dispositivo BSMA (Busca de Soluções por Máquina Adaptável) foi proposto, formalizado, e um protótipo experimental foi construído e exercitado, para que se completasse esta pesquisa. Maiores detalhes podem ser encontrados em [7].

2. MÉTODO PROPOSTO

O método de investigação de soluções é proposto a partir das características encontradas no modelo do dispositivo computacional BSMA definido, que, conforme mencionado, utiliza como base formal o autômato adaptativo, acrescido de características de outros modelos levantados, procurando-se dessa forma garantir a integridade, a integração e a uniformidade do método.

2.1. Definição do Método e Construção do Dispositivo BSMA

O método proposto é baseado em indução, da seguinte forma: Para um dado problema ou questão Q a ser solucionado, deve-se coletar o maior número possível de informações, como por exemplo teorias, modelos, exemplos de resultados. A partir destas informações, organiza-se o material colhido segundo a sua natureza, e trabalha-se com uma ordenação deste material operando teorias juntamente com modelos, como se fossem hipóteses em um processo indutivo, e com os exemplos, como se fossem dados observados. A partir daí, tem-se um

esquema adequado à utilização da teoria da previsão de Solomonoff, podendo-se aplicar a regra de Bayes, substituindo a probabilidade *a priori* por uma medida universal, como a complexidade de Kolmogorov para prefixos. Então, a resposta à questão é fornecida como uma previsão, se houver resposta alcançável pelo dispositivo utilizado [4].

A questão da resposta a ser gerada, quando não há qualquer informação sobre o problema, é resolvida através do uso do senso comum. Assim, busca-se uma resposta que seja aceita pela maioria dos indivíduos e, depois, verifica-se se tal resposta está correta. A forma de produzi-la através do senso comum foge ao escopo deste trabalho.

O método a ser utilizado aqui será limitado em seu aspecto de produzir solução para problemas novos, ou seja, somente se utiliza o método quando há alguma informação sobre o fenômeno ou o problema em estudo. A utilização do dispositivo BSMA, por um pesquisador, deve ser então realizada através do envio de modelos ou teorias, e de dados de exemplo que possam servir para treinar o dispositivo na busca de uma solução para o problema proposto. A execução do treinamento pode ser feita repetidas vezes, para a mesma massa de dados, já que o dispositivo tem limite de tempo de execução, não possui senso comum e nem consciência, portanto, pode não haver resposta até mesmo durante o treinamento.

A operação do dispositivo dá-se por meio de ciclos. Assim, um ciclo corresponde a uma execução completa do modelo, ou seja, é composto pelos passos de computação executados pelos modelos utilizados. Por outro lado, é necessário impor limites de tempo de execução dentro de um ciclo, bem como um limite para o número de ciclos. Isto é garantido pela existência de um dispositivo de medida de tempo (relógio), que controla todos os tempos de execução.

2.1.1. Especificação do Dispositivo BSMA

O autômato adaptativo, tal como definido, é composto por transições que podem ou não acionar funções adaptativas. Nesta pesquisa, ao se construir um modelo baseado neste autômato, as transições são enumeradas, isto é, são ordenadas e contabilizadas. Desta maneira, é sempre possível identificar partes estruturais do modelo de autômato, através do número de ordem de cada parte. Da mesma forma, os modelos de autômato adaptativo a serem executados, ou em execução, dentro do dispositivo BSMA, também são enumerados.

Porém, para se poder realizar tanto a modificação aleatória, quanto a mudança através da combinação de partes estruturais de diferentes modelos, estende-se o modelo de autômato adaptativo com a introdução de duas características: a primeira é a introdução de um oráculo (semelhante à máquina de Turing com oráculo [1]), com o objetivo de realizar as mudanças aleatórias pretendidas nos modelos. A segunda mudança é a introdução de uma função de combinação de partes estruturais (transições), que não faz parte do autômato, mas apenas do dispositivo aqui proposto.

O oráculo, na verdade, traduz-se na execução de uma ação especial que, através do uso de um número aleatório (valor numérico inteiro positivo, gerado por uma função externa aos modelos, presente no dispositivo), modifica a estrutura dos modelos de autômato existentes. Esta modificação é solicitada por um elemento controlador (portanto externo aos modelos), e dá-se através da troca de transições cujo número de ordem corresponda ao número aleatório utilizado. Desta maneira, seguindo a ordem dos modelos gerados, cada um dos modelos troca transições com o próximo, sempre gerando novos modelos (sem destruir os modelos já existentes). O último modelo troca transições com o primeiro modelo.

Para controlar os passos de computação e indicar em quais deles uma ação oráculo pode ser processada, há ainda um controlador de tempo único (relógio universal). Na realidade o relógio

conta os passos de computação, e também os ciclos de computação (um ciclo de computação é composto por um conjunto de passos, no qual pelo menos um modelo gerado terminou seu processamento, ou o tempo limite de processamento se esgotou [7]). Estabelece-se a seguinte regra: uma ação de oráculo não pode ser executada em dois ciclos de computação subseqüentes pela mesma função adaptativa, somente por outra função. Isto significa que deve haver um hiato entre a execução de uma ação de oráculo e a próxima execução.

O conceito de combinação de modelos de autômato também é definido a partir do modelo de autômato adaptativo original. A combinação entre dois modelos de autômato adaptativo somente é possível entre pares distintos de modelos presentes no dispositivo, e dar-se-á através da execução da ação de oráculo.

Uma combinação, ou uma modificação aleatória, não elimina os modelos originais, apenas acrescenta modelos novos aos já existentes. Os novos modelos introduzidos devem ser dispostos em ordem de construção, sem perturbar a ordem anterior de modelos. Especifica-se que ao fim de um ciclo de computação os modelos que não finalizaram são descartados e nova ordem é gerada.

É fundamental a existência, localmente ao dispositivo, de um elemento controlador, externo aos modelos de autômato, que possa gerenciar e computar os passos e ciclos de computação, que induza as combinações e que, finalmente, responda às questões propostas ao oráculo. Portanto, cada modelo de autômato tem funcionamento autônomo enquanto o controlador assim o permitir. Este controlador pode ser modelado de forma semelhante à de uma máquina de Turing, com uma fita infinita, toda preenchida com elementos numéricos aleatórios, da qual ele retira os valores a serem passados, indicando a resposta à ação de oráculo. Esta fita pode, com vantagens práticas, ser substituída por uma função que gere números aleatórios.

Porém, o papel do controlador não se encerra neste ponto. Ele deve avaliar os modelos quanto aos resultados por eles obtidos e, dessa maneira, reconhecer os mais aptos, descartando os demais. Os modelos selecionados servem de base para a geração dos modelos no próximo ciclo de computação.

Concluindo a definição do método, é necessário, estabelecer os requisitos do dispositivo BSMA:

1. O dispositivo deve possuir um controlador central;
2. O controlador central deve coordenar as ações a serem tomadas nos modelos gerados;
3. Os modelos devem ser estruturados como autômatos adaptativos;
4. Os modelos devem ser executados concorrentemente ou em ordem lexicográfica;
5. O controlador central deve ser capaz de combinar os modelos;
6. O controlador deve poder introduzir mudanças aleatórias na estrutura dos modelos, baseados em autômatos adaptativos, através de uma ação de oráculo;
7. Toda e qualquer mudança na estrutura dos modelos somente pode ser realizada antes de um ciclo de computação;
8. Os ciclos de computação devem ser compostos pelos passos de computação, e o controlador deve indicar quando os passos e os ciclos devem se iniciar e quando um ciclo deve terminar;
9. O controlador deve ser capaz de avaliar os modelos, de forma a escolher os mais aptos a encontrar uma solução;
10. A execução dos modelos é prevista para durar uma determinada quantidade previamente estabelecida de passos de computação;
11. O controlador deve verificar, ao final do total de passos-limite de processamento, quais computações ainda não terminaram e finalizá-las, eliminando em seguida os

modelos correspondentes.

A partir dessas características, define-se a estrutura do dispositivo computacional que será usado pelo método proposto. Cumpre ressaltar que um sistema computacional qualquer, proposto com estas características, pode ser usado também como um sistema de aprendizado que, através de um treinamento adequado, seja capaz de aprender a realizar uma determinada tarefa e posteriormente a ela responder.

2.2. Dispositivo BSMA Proposto

É fundamental, para o propósito de encontrar soluções, que o dispositivo possua um indicador que permita identificar se a trajetória computacional adotada é adequada, isto é, se ela conduz ou não a uma solução. Para a especificação do indicador, há três alternativas viáveis: Treinamento, Aprendizagem por exemplos e Especificação de uma função de verificação.

A alternativa escolhida foi a primeira. Com isso, para cada par entrada/saída fornecido, modelos de autômato são gerados e exercitados, tendo seus comportamentos comparados com o esperado para cada par. Assim, modelos que apresentem comportamentos diferentes do esperado são descartados. Cada modelo que consiga atingir a resposta esperada é utilizado para compor novos processos de solução.

Assim, o elemento de entrada do dispositivo deve especificar uma construção inicial, associada a um conjunto de triplas estipuladas pelo usuário do dispositivo:

$$\{(e_i, s_i, v_i) \mid e_i - \text{entrada}, s_i - \text{saída}, v_i - \text{valor}, i \geq 1\}^n, n \geq 1.$$

O indicador para a função de medição opera com um conjunto de dados para o qual se deseja encontrar uma hipótese consistente, e com o conjunto de modelos de autômato, que representa o conjunto das hipóteses levantadas. Esta função de medição determina, na realidade, se os modelos de solução estão ou não trilhando um caminho que leva ao objetivo a ser alcançado, desempenhando assim um papel de manutenção de meta.

Na Figura 1, o elemento controlador aparece como o item mais importante, já que ele centraliza todas as ações do dispositivo, disparando o exercício dos modelos de solução através da entrada e identificação dos problemas, realizada pelo elemento de entrada. Assim, o controlador induz alterações nas construções, de forma a tentar gerar novos modelos. Caso não se chegue a uma solução, pode ocorrer a necessidade de processamento cíclico e, mesmo assim, pode-se não alcançar uma solução. Neste caso, a resposta assumida é “Solução não-Exeqüível”, por inexistência ou impossibilidade de aplicação prática.

As construções, que representam os modelos de solução gerados pelo dispositivo, são representadas na figura através dos modelos gerados (M_1 .. M_n), e juntas compõem um agregado, isto é, um vetor de soluções possíveis, porém, não necessariamente, preenchido por completo. Os agregados, compostos pelas contruções (M_1 .. M_n), são independentes entre si, podendo com isso gerar diferentes soluções para os problemas. A quantidade de agregados existentes no dispositivo depende basicamente de seu parâmetro de limite da quantidade de espaço alocado, e também da quantidade de diferentes modelos de solução gerados. A combinação de ambos indica a quantidade total de agregados, embora o limite máximo seja estabelecido pelo parâmetro de limite da quantidade de espaço.

Pode-se observar que cada construção dentro de um agregado gera saídas, as quais não passam diretamente ao elemento controlador. A Figura 1 mostra as saídas sendo tratadas por um elemento “CP” (comparador). Este elemento simboliza, na realidade, que o controlador realiza uma comparação entre as construções exercitadas que permaneceram, e que permitem gerar os valores das medidas de complexidade utilizadas e não que exista um elemento específico

denominado comparador. Maiores detalhes em [7].

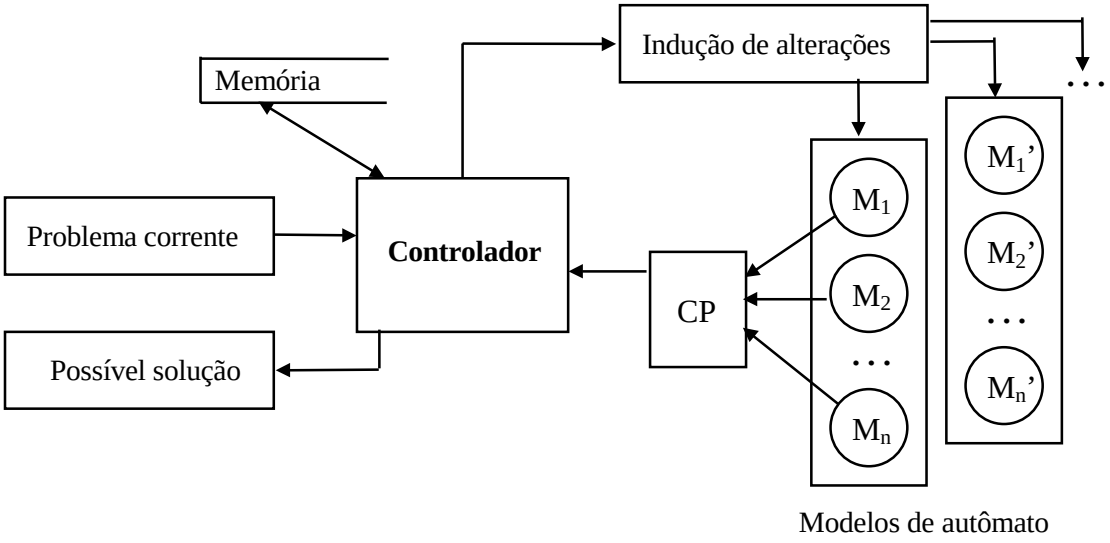


Figura 1 - Arquitetura do dispositivo BSMA proposto

2.2.1. Significado da Solução para o Dispositivo BSMA Proposto

A solução é, portanto, formada dentro de um agregado, pela construção mais adaptada, fazendo com que o dispositivo tenha sempre de escolher, dentro dos agregados, as construções que devem permanecer como possíveis modelos de solução, e descartar as demais. Conforme estipulado anteriormente, esta escolha é também realizada através do uso da função de medição, que desempenha o papel de manutenção de meta.

Novamente, deve-se observar que, neste dispositivo, as soluções encontradas deverão sempre estar presentes em alguma construção exercitada dentro de um agregado, isto é, supõe-se que a construção já estivesse apta, estruturada para encontrar a solução. A idéia central utilizada aqui é a possibilidade de introduzir modelos de solução aos agregados gerados pelo dispositivo, através de alterações inseridas nas construções dentro dos agregados – usando combinação ou mudanças probabilísticas produzidas pela ação de oráculo – de forma a encontrar uma configuração apta a tratar o problema em questão. A ação de oráculo tem, portanto, papel extremamente importante na busca de soluções, já que, neste dispositivo, é ela a responsável pela introdução de alterações não previstas (probabilísticas) nos modelos, podendo com isso dotá-los de características diversas das anteriormente encontradas e ampliar o espaço de busca de soluções.

Considera-se que a resposta do dispositivo pode ou representar uma solução ou então apenas uma resposta negativa, podendo, portanto, não ser a finalização da tarefa de busca de soluções. Isto ocorre porque, devido aos parâmetros utilizados ou mesmo devido à complexidade do problema, a solução pode não ser exeqüível. Entretanto, supondo-se que o problema em questão apresente solução computacional exeqüível, a saída do dispositivo será uma solução, representada através de um modelo de autômato adaptativo. Maiores detalhes em [7].

3. ASPECTOS DA IMPLEMENTAÇÃO

A escolha realizada para implementar o dispositivo BSMA foi utilizar uma linguagem de programação bastante difundida na comunidade científica da área da Inteligência Artificial, que é a linguagem LISP. O uso desta linguagem permite o desenvolvimento mais rápido dos

programas, já que seu nível de abstração é mais elevado, e opera naturalmente com processamento simbólico. Entretanto, o uso de um compilador para a linguagem LISP pura não é tão vantajoso neste trabalho, já que não se teriam as facilidades de depuração, de geração de elementos gráficos, que se encontram em compiladores mais novos, os quais introduziram funções para orientação a objeto e uso de interface gráfica.

O dispositivo BSMA proposto foi, portanto, implementado inicialmente como um protótipo, apoiado na plataforma de microcomputadores da família IBM-PC e com código escrito em linguagem LISP.

3.1. Construção do dispositivo BSMA

O dispositivo foi construído de forma a que seus componentes se comportem da seguinte maneira:

a. Há uma função de interpretação de autômato previamente codificada, que opera as construções de forma semelhante a uma máquina de Turing universal que executa programas e dados que descrevem o comportamento de uma máquina de Turing particular;
b. Os modelos de autômato vão sendo construídos e modificados à medida que o dispositivo computacional vai desenrolando seu processamento. Assim, a partir de uma entrada específica, os modelos são gerados para serem posteriormente interpretados. Este procedimento de geração–modificação e interpretação é realizado até se obter uma resposta.

Para que o dispositivo possa efetuar seu processamento, é necessário que receba as informações que lhe permitirão gerar os modelos de autômato. Desta maneira, o primeiro elemento estruturado é a entrada, depois o controlador com suas funções, e por último o elemento de saída [7].

3.1.1. Análise do Dispositivo BSMA

Ao analisar o dispositivo proposto, é possível, por simples inspeção, definir cinco classes funcionais fundamentais: a classe de entradas; a classe do controlador; a classe do interpretador de modelos de autômato; a classe dos modelos de autômato; e a classe das saídas.

Conforme estabelecido anteriormente, a proposta é de se usar uma ontologia específica de gramática, de linguagem. Isto não tira a generalidade do princípio, embora na prática reduza drasticamente a dificuldade de implementação. A generalidade é preservada devido à conhecida equivalência entre linguagens, gramáticas, autômatos, funções recursivas como formas alternativas de expressão de computações. Neste trabalho, que se concentra especialmente em mostrar a viabilidade de tal dispositivo, torna-se assim muito conveniente usar esta simplificação nesta fase do desenvolvimento desta pesquisa. Assim, o elemento de entrada pode capturar as especificações do usuário e construir um modelo genérico e possivelmente não-determinístico de autômato inicial [7]. Para a especificação do problema, deve ser enviado um conjunto de três listas de argumentos e um símbolo (o símbolo inicial da linguagem). As listas são as seguintes: Lista de Símbolos Terminais; Lista de Símbolos Não-Terminais; Lista de Produções.

Um exemplo de especificação é a linguagem regular composta por quantidades pares de elementos “a”, $L = (a^2)^*$, ou usando-se a notação adotada L = (a a)*, cuja especificação fica: Símbolo inicial: s; Lista de Símbolos Terminais: (a); Lista de Símbolos Não-Terminais: (s aa a*); Lista de Produções: ((s -> a*) (a* -> () (aa aa a*)) (aa -> a)).

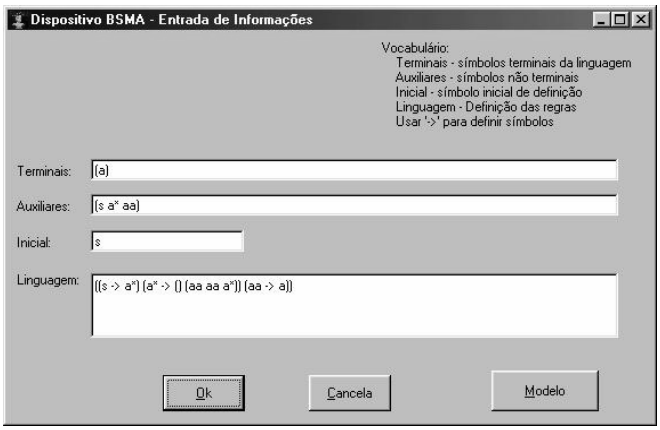


Figura 1 - Tela de Entrada (linguagem regular)

No exemplo em questão, a segunda regra da lista de regras de produção: (a* -> () (aa aa a*)) é interpretada como um "ou" entre o símbolo de regra vazia "()" e a lista (aa aa a*).

Os casos de teste (ou de treinamento), que permitirão exercitar os modelos aptos e, a partir do exercício, determinar quais são o(s) mais apto(s), o(s) melhor(es). Utilizando o mesmo exemplo anterior, as listas são do tipo: Casos válidos: ((a a) (a a a a)); Casos não-válidos: ((a) (a a a)).

Para os casos válidos foram especificadas duas cadeias: "aa" e "aaaa", enquanto que para os casos não-válidos foram especificadas: "a" e "aaa".

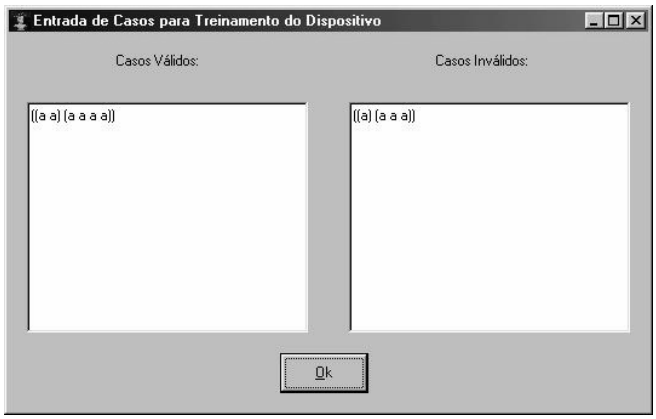
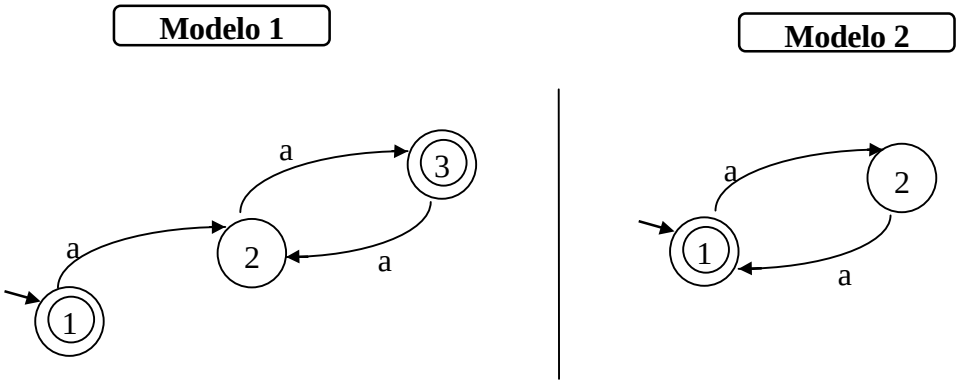


Figura 2 - Entrada de casos de treinamento (linguagem regular)

Através destas especificações, o dispositivo entrada captura as informações e as repassa ao elemento controle, que prossegue com a etapa de geração e os exercita. Neste caso:



Usando o critério adotado para a função de medição, obtém-se um valor de complexidade para

o primeiro modelo maior do que para o segundo modelo. Assim sendo, o segundo modelo é o melhor. Com isso, a resposta enviada ao usuário é o segundo modelo, da seguinte forma:

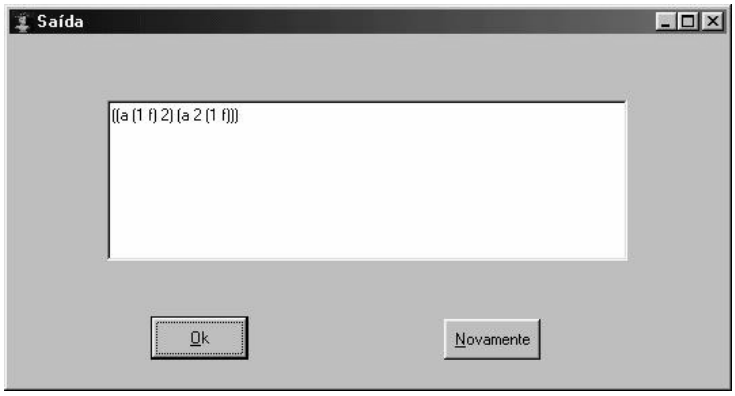


Figura 3 - Saída gerada pelo dispositivo

Desta maneira, o resultado gerado é o modelo de autômato mais simples para resolver o problema. Nesta tela, o usuário pode encerrar este processamento retornando à tela inicial, ou solicitar nova tentativa.

De forma idêntica podem ser consideradas as linguagens livres de contexto. Já no caso das dependentes de contexto são utilizadas funções adaptativas. Assim, ao se representar uma linguagem do tipo $a^n b^n$, deve-se especificar da seguinte forma: ((s -> a*) (a* -> () (aa a* bb)) (aa -> a) (bb -> b)). O dispositivo BSMA trata uma produção deste tipo através da criação de transições adaptativas. No caso exemplificado, ao se dar entrada na produção a*, através da regra aa, o dispositivo gera uma função adaptativa que, quando executada, gera uma nova transição que conecta o estado corrente a ele mesmo, consumindo o símbolo indicado pela regra aa. Ao sair da produção a*, através da regra bb, o dispositivo gera uma função adaptativa que, quando executada, elimina uma transição que conecta o estado anterior a ele mesmo, e consome o símbolo indicado pela regra bb.

3.2. Comentários sobre a Parte Prática

Os experimentos realizados foram simples, com baixo número de estados na maior parte dos casos, e com baixa complexidade. Porém, o objetivo foi exatamente mostrar a viabilidade de geração de soluções e avaliar o custo associado a essa tarefa.

Resultados

Para o dispositivo BSMA proposto, a forma utilizada para se limitar a quantidade de modelos é a escolha baseada na medida de complexidade. Nos casos estudados, ao limitar o espaço de busca, ainda assim, no espaço remanescente, os melhores estavam presentes. Isto sugere que esta medida de complexidade possa ser de fato um elemento direcionador, e que seu uso pelo dispositivo BSMA proposto possa auxiliar a melhor escolha, mesmo que se necessite eliminar alguns modelos por falta de espaço, ou limitar uma explosão combinatória.

Uma consideração interessante e clássica a respeito de linguagens regulares e autômatos finitos pode ser encontrada em [3]: um estudo a respeito do pior caso na geração de modelos de autômato determinísticos revela que, para o caso de haver um total de R regras de produção em uma gramática regular, para se gerar o melhor modelo determinístico, o algoritmo deveria dispender, em termos de tempo, valores da ordem $O(2^K)$, onde K representa o número de estados, ou seja $K \approx 2R$. Isto porque o conjunto potência de estados gerados é 2^K.

Ao se efetuar o mesmo cálculo para o pior caso do dispositivo BSMA proposto e, considerando

que não haverá restrições quanto ao número de transições e modelos gerados, tem-se o seguinte: o total de modelos gerados é de $2^{(k-1)}$, portanto, da ordem O[$2^{(k-1)}$]. Ao se comparar os dois valores, no pior caso, leva-se em consideração que os valores de K e de k são diferentes, porém comparáveis, já que a linguagem é a mesma. Portanto, tem-se: K = 2R; k = R + 2 $\Rightarrow$ k $\cong$ K / 2, para R >> 2 (R grande). Comparando-se os valores em ambos os casos, obtém-se que, se t representa o dispêndio em termos de tempo para o dispositivo, e T representa o dispêndio de tempo para o algoritmo de [3], tem-se que $t \approx \sqrt{T}$.

Este é um resultado melhor que o anterior, embora também seja da ordem exponencial. Entretanto, pode-se considerar que o algoritmo implementado pelo dispositivo BSMA proposto pode limitar bastante o espaço de busca em uma computação exeqüível, trabalhando em um espaço polinomial, à custa de diminuir a chance de encontrar uma solução, caso a redução seja drástica em relação ao espaço total de busca [7].

4. CONSIDERAÇÕES FINAIS

Nesta pesquisa, propõe-se um método para estruturar modelos de computação, utilizando os autômatos adaptativos como substrato. Outros modelos computacionais importantes, com características diferentes dos autômatos adaptativos, também foram estudados para compor o método e permitir a construção de um dispositivo que pudesse aprender e chegar a soluções de problemas complexos.

Dentre as características diferentes encontradas em outros modelos computacionais acham-se: a possibilidade de introdução de modificações aleatórias, semelhante a uma mutação genética (encontrada em algoritmos genéticos); a combinação de partes estruturais, transições, nos modelos (também encontrada em algoritmos genéticos); a possibilidade de treinamento e o exercício de elementos em paralelo (encontradas nas redes neurais), etc..

Com relação à ciência da computação, há contribuições interessantes nesta pesquisa, como por exemplo, a representação do problema do caixeiro viajante em termos de autômatos adaptativos. Desta forma, reduz-se este problema ao do ciclo hamiltoniano (“*Dado um grafo G, existe algum ciclo que passe por todos os nós de G exatamente uma vez?*” [2, p.282]), associado a uma escolha dos modelos de menor complexidade pelo dispositivo BSMA. Assim, o problema do caixeiro viajante fica formulado por meio de modelos de autômatos.

Uma outra contribuição na área computacional é o algoritmo utilizado para a geração de modelos de autômatos determinísticos, a partir de um modelo inicial, de tal forma que possa ser encontrado o melhor deles. Conforme exposto anteriormente, o número de alternativas a serem exploradas é menor (da ordem O[$\sqrt{\bullet}$]), porque o número de estados e de transições iniciais geradas é menor, diminuindo, portanto, o tempo de processamento e o espaço ocupados.

Para os problemas que envolvem otimização, busca da melhor alternativa, o dispositivo BSMA proposto também se mostra propício, já que realiza a busca sempre para o melhor modelo de solução, baseado na redução de complexidade. Basta formular o problema com este direcionamento, ou seja, como um problema de minimização de complexidade expresso em forma de linguagem, e utilizar o dispositivo.

Finalmente, o método e o dispositivo BSMA propostos permitem o uso dos autômatos adaptativos para resolver problemas de aprendizado e resolução de problemas, isto porque o método prevê treinamento. Assim um modelo, como o exemplo da árvore de decisão, pode “aprender” a resolver um problema e, a partir de então, inferir respostas a questões ou situações para as quais não foi treinado. Com isso, abre-se mais uma alternativa de estudo na área de inteligência artificial.

Comentários

Desde já, observa-se que uma ampliação possível e importante é a utilização de linguagem natural para formulação das solicitações do usuário, o que exigirá a elaboração de uma sofisticada interface homem-máquina adequada a esse propósito.

Em relação ao protótipo do dispositivo BSMA, algumas críticas podem ser tecidas. A principal delas diz respeito às construções de autômatos: tais construções podem ser ampliadas para melhor representar e explorar os autômatos adaptativos.

Tanto o método como o dispositivo podem ser utilizados, em termos práticos, nas seguintes áreas: Ensino–aprendizagem; Treinamento; Produção.

Sendo o método proposto totalmente genérico, dispensa o uso de ferramenta específica, podendo ser utilizado por um ser humano na busca de soluções.

Conclusão

Espera-se ter contribuído com esta pesquisa para a formulação de uma metodologia bem fundamentada de solução de problemas, que possa ser usada em simulações e também como modelo teórico para execução de programas em diversos ambientes. Espera-se, ainda, ter contribuído com a utilização prática do autômato adaptativo, através da proposição de um dispositivo de software nele baseado.

Dentro dos objetivos traçados, a forma utilizada para alcançá-los também foi interessante e profícua. Isto deve-se ao fato de buscar uma visão integradora entre áreas diversas do conhecimento humano e, dentro desta diversidade, tentar encontrar a resposta mais adequada às questões colocadas no início desta pesquisa.

O trabalho desenvolveu-se dentro das expectativas provando a viabilidade do projeto, e pode ser considerado um marco inicial nesta pesquisa, abrindo caminho para toda uma série de futuros experimentos e desenvolvimentos. Acredita-se ter com ele contribuído com a proposta de um novo ângulo de visão dentro da computação, com o uso do método e do dispositivo BSMA propostos.

5. REFERÊNCIAS BIBLIOGRÁFICAS

[1] BENNETT, Charles H. Logical depth and physical complexity. In: HERKEN, R. ed. *The universal Turing machine*. New York, Springer-Verlag, 1995 p. 207-235.

[2] JOSÉ NETO, J. Adaptive automata for context-dependent languages. *ACM SIGPLAN Notices*, v. 29, n. 9, p. 115-124, Sep. 1994.

[3] LEWIS, H. R.; PAPADIMITRIOU, C. H. *Elements of the theory of computation*. New Jersey, Prentice-Hall, Inc, 1998.

[4] LI, M; VITÁNYI, P. *An introduction to Kolmogorov complexity and its applications*. 2nd. ed., New York, Springer-Verlag, 1997.

[5] MINSKY, M; SEYMOUR P. *Perceptrons*. Cambridge (MA), MIT Press, 1988.

[6] POPPER, K.; ECCLES, J. C *O eu e seu cérebro*. Brasília, Editora Universidade de Brasília, 1995.

[7] ROCHA, R. L. A. *Um método de escolha automática de soluções usando tecnologia adaptativa*. São Paulo, 2000. 211p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo.

[8] RUSSELL, S. J.; NORVIG, P. *Artificial intelligence a modern approach*. New Jersey, Prentice-Hall, Inc., 1995.

[9] SEARLE, J.*O Mistério da Consciência.* São Paulo, Martins Fontes, 1998.

[10] SOLOMONOFF, R. J. A formal theory of inductive inference - Part I e Part II. *Information Control*, v.7, p.1-22; p.224-254, 1964.

[11] SPEARS, W. M.; DE JONG, K. A.; BÄCK, T.; FOGEL, D. B.; DE GARIS, H. An overview of evolutionary computation. In: European Conference on Machine Learning, 1993. *Proceedings*. p. 442-459.

[12] WOOLDRIDGE, M.; JENNINGS, N. Formalizing the cooperative problem solving process. In: Thirteenth International Workshop on Distributed Artificial Intelligence (IWDAI-94), Lake Quinalt, WA, 1994. *Proceedings*. p. 403-417.

Ricardo Luis de Azevedo da Rocha

Formado em Eng. Elétrica – modalidade Eletrônica pela PUC-RJ em 1982.
Mestre em Eng. Elétrica – Sistemas Digitais - EPUSP - 1995
Doutorando em Eng. Elétrica – Sistemas Digitais - EPUSP – 2000
Áreas de interesse: Autômatos e Linguagens formais, Engenharia de Software, Tecnologia Adaptativa, Inteligência Artificial, Métricas de Software.
Pesquisa Corrente em: metodologias para aplicação de tecnologias adaptativas à eng. de software
Atividade Didática: Prof. Universitário de Eng. de Software, Autômatos e linguagens formais.