Aplicação de Autômatos Adaptativos na avaliação de autômatos gerados por algoritmos genéticos

Victor D. Lopes, Ricardo L. A. Rocha

Abstract— **The language inference is a field of great importance in computer science. This work studies the application of genetic algorithms in the inference of regular languages, through the evolution of a population of finite automata that accept the strings belonged to it.**
In the automata evaluation obtained in each population, it is proposed the utilization of an adaptive automaton as base for this evaluation.

***Palavras chaves*—Autômato adaptativo, árvore de prefixo, algoritmos genéticos, inferência de autômato.**

I. Introdução

A inferência de linguagens é um campo de grande importância dentro da ciência da computação. No presente trabalho, será estudada a aplicação de algoritmos genéticos na inferência de linguagens regulares, por meio da evolução de uma população de autômatos finitos que aceitam as cadeias pertencentes a ela.
Na avaliação dos autômatos obtidos em cada população, propõe-se a utilização de um autômato adaptativo como base para essa avaliação.

II. Tecnologia Adaptativa

Como visto em [3], um dispositivo adaptativo possui estrutura interna que se modifica dinamicamente durante sua execução, em função de seus estímulos de entrada, sem a intervenção de agentes externos.
Esses dispositivos são divididos em duas partes; a camada subjacente, que são dispositivos usuais como autômatos, gramáticas ou tabelas de decisão. Esses dispositivos são tipicamente não adaptativos, determinísticos e definidos por um conjunto finito de regras. Considera-se como segunda parte o mecanismo adaptativo, definido como responsável pela formalização e execução das auto-modificações, que ocorrerão no conjunto de regras do dispositivo subjacente.
A execução do mecanismo adaptativo, como definido em [4], é feita através de ações adaptativas. Para cada passo na execução de um dispositivo adaptativo pode haver uma ação adaptativa anterior e uma posterior a esse passo. Essas ações são normalmente descritas como funções adaptativas, parametrizadas ou não, constituídas basicamente de uma lista de ações adaptativas elementares.
Há três tipos de ações adaptativas elementares: ação de consulta, que possibilita a busca de padrões na estrutura definida pelas regras da camada subjacente; ação de eliminação e ação de inclusão, que permitem a alteração da estrutura da camada subjacente.
Essas funções podem conter três tipos de variáveis: uma lista de parâmetros que são passados no momento de sua execução; uma lista de geradores, que são variáveis especiais automaticamente preenchidas a cada execução da função com valores não utilizados pelo dispositivo subjacente até a ocasião. Por último, uma lista de variáveis preenchidas através das ações adaptativas elementares de consulta, cujo valor, depois de determinado, permanece constante até o final da execução da função.

III. Autômato Adaptativo

Autômato adaptativo é uma instância de dispositivo adaptativo que tem como camada subjacente um autômato de estados finitos ou um autômato de pilha estruturado. Para este estudo será utilizado um autômato adaptativo de estados finitos simples.
Baseando-se em [4], um autômato adaptativo M é definido por uma máquina de estados inicial E_0 a qual será submetida uma cadeia W de comprimento n. E_n é a máquina de estados final, depois de reconhecer a cadeia W inteira. E_i é uma máquina de estados intermediária que será submetida a uma cadeia W_i, que é uma sub-cadeia de W, sem os i primeiros elementos.
Definindo W como sendo constituída pelos elementos $a_0a_1a_2...a_n$, pode-se dizer que a trajetória de reconhecimento do autômato adaptativo pela cadeia W é (E_0, a_0) -> (E_1, a_1) -> (E_2,a_2) ->...->(E_n,a_n).
Para um autômato de estados finitos simples, em um determinado momento da trajetória de reconhecimento, as transições do autômato são representadas pela seguinte notação ***(e, s) : A, -> e', B***, sendo que ***e*** é o estado de origem da transição, ***s*** é o elemento recebido pelo autômato, ***A*** é a ação adaptativa anterior da transição, ***e'*** é o estado destino e ***B*** é a ação adaptativa posterior.

V. D. Lopes é pesquisador do Laboratório de Linguagens e Técnicas Adaptativas, Escola Politécnica da USP, São Paulo/SP, Brasil; fone: 55 11-3091-5583; e-mail: victor.lopes@ poli.usp.br.
R. L. A. Rocha é pesquisador do Laboratório de Linguagens e Técnicas Adaptativas, Escola Politécnica da USP, São Paulo/SP, Brasil; fone: 55 11-3091-5583; e-mail: luis.rocha@ poli.usp.br.

IV. Autômato de Árvore de Prefixo

Com base em [5], um autômato de árvore de prefixo é um autômato em forma de árvore que aceita, de forma determinística, uma linguagem finita representando alguns exemplos positivos de uma linguagem mais ampla.

Implementando esse autômato através de um autômato adaptativo, este inicialmente possuirá apenas um estado, o inicial que também será o final. Esse estado apresentará apenas uma transição para ele mesmo com uma ação adaptativa associada a essa transição.

Essa ação adaptativa fará com que o autômato cresça guiado pelos elementos que recebe da cadeia de entrada e realize a memorização dessa cadeia, através de alterações no conjunto de regras que o definem.

V. Algoritmos Genéticos

Baseando-se em [7], algoritmos genéticos são algoritmos de busca ou classificação que se baseiam no processo de evolução natural de Darwin. Eles trabalham com populações de indivíduos, onde cada indivíduo é uma solução em potencial para o problema em estudo.

Cada indivíduo é representado por um cromossomo, normalmente uma cadeia de 0's e 1's. Durante o processo de recombinação genética, esse cromossomo sofre operações inspiradas nas que ocorrem nos núcleos celulares, como *crossover* e mutação.

Todos os indivíduos são submetidos a um ambiente, que representa o problema a ser resolvido, e recebem uma nota de acordo com seu desempenho. Assim os mais aptos, os que melhor resolveram o problema, recebem notas mais altas.

Depois de avaliados, são escolhidos alguns indivíduos de forma aleatória, para gerar a próxima população, sendo que os mais aptos têm probabilidade maior de serem escolhidos, podendo ser escolhidos até mais de uma vez.

Os indivíduos escolhido são agrupados em pares e submetidos ao processo de reprodução, seus descendentes são os indivíduos que irão constituir a próxima geração.

VI. Representação do cromossomo

Para o presente trabalho, a representação do indivíduo por meio do cromossomo se baseou em [2], primeiramente foi limitado o número máximo de estados e elementos distintos na cadeia de entrada, os autômatos têm no máximo sete estados e quatro elementos distintos na cadeia de entrada.

O formato do cromossomo é bastante simples, ele é dividido em sete partes, uma para cada estado, cada parte possui treze bits, a identificação dos estados é posicional, ou seja, o primeiro bloco de treze bits representa o estado 0, o segundo representa o estado 1 e assim por diante. O primeiro bit da representação do estado indica se o estado é de aceitação ou não, os doze bits restantes são divididos em quatro parte, uma para cada possível elemento de entrada, esses elementos serão identificados hipoteticamente como **a**,**b**,**c** e **d**, de forma que a primeira parte dos doze bits indica o estado destino (em binário) da transição recebendo o elemento **a**, a segunda parte indica o estado destino recebendo o elemento **b** e assim por diante, sendo que o número sete (111) indica que a transição não está ativa.

VII. Implementação do algoritmo genético

Para a implementação do algoritmo foi usado como mecanismo de seleção o *roulette wheel* padrão, sem utilização de *elitist strategy* para garantir a permanência dos indivíduos mais aptos. Como operadores genéticos, foram utilizados o *crossover* de ponto único com probabilidade de 90% e a mutação com probabilidade de 1%.

A população é de vinte indivíduos criados aleatoriamente ou algum padrão de cromossomo é forçado na primeira população. Esses indivíduos são submetidos a um determinado número de cadeias aleatórias de uma linguagem regular.

Ao final do processamento de cada uma das cadeias, os indivíduos têm seus desempenhos avaliados e armazenados. No final será utilizada a avaliação média de cada um deles na seleção.

O critério de parada foi definido empiricamente de forma a parar após executar mil gerações.

VIII. Procedimento de avaliação dos indivíduos de uma população

A avaliação será feita através da comparação da nota do autômato com a nota de um autômato adaptativo usado como referência implementando uma árvore de prefixo. Esse método foi baseado no método proposto em [1], com várias diferenças importantes, desde a representação do cromossomo dos indivíduos até a função de avaliação.

No lugar da função que calcula o comprimento mínimo de mensagem (MML) para cada autômato, proposta originalmente, foi usada uma função mais simples que não leva em consideração não determinismos, pois como é possível verificar, a estrutura proposta do cromossomo não permite a criação de autômatos não determinísticos. Além do mais, a nova função leva em consideração a eficiência das transições do autômato, pois considera a relação entre o número de transições utilizadas no processamento das cadeias de entrada e o número total de transições presentes nele. Considera também a quantidade de estados e a quantidade de estados finais.

A função é:

$$fitness = a + b * (c + d + e * f)$$

sendo que:

a é a relação entre a quantidade de elementos que o autômato conseguiu computar com a quantidade de elementos da cadeia multiplicada por um determinado peso.

b é 1 se autômato conseguiu computar todos os elementos da cadeia e 0 no caso contrário.

c é a relação entre a quantidade de estados do autômato com a quantidade máxima de estado do autômato multiplicada por um determinado peso. [2]

d é a relação entre a quantidade de transições utilizadas pelo autômato com a quantidade total de transições no autômato multiplicada por um determinado peso.

e é 1 se o autômato processou a cadeia toda e parou em um estado de aceitação e 0 no caso contrário.

f é a relação entre a quantidade de estados de aceitação com a quantidade máxima de estados multiplicada por um determinado peso.[3]

Finalmente, a função que realiza a comparação com o autômato de referência e retorna a avaliação dos indivíduos é a mesma exponencial proposta em [1].

IX. EXPERIMENTO

Foram realizados alguns experimentos com o algoritmo e como base de comparação entre os autômatos obtidos foi utilizada suas relações de tamanho. O cálculo da relação de tamanho é feito através da fórmula:

r = 1 – (T/Max), sendo T a quantidade de estados do autômato e Max a quantidade máxima de estados que o autômato pode ter, nesse caso é sete, devido às restrições na representação dos cromossomos dos indivíduos da população.

Foram utilizadas 12 linguagens regulares escolhidas de [1] para a execução do algoritmo. Os resultados obtidos são a média de 10 execuções para cada linguagem, segue a tabela 1 com eles.

Na coluna "Alvo" estão os dados obtidos de [1] dos autômatos esperados para as linguagens. Na coluna "GA20" estão as relações obtidas dos autômatos gerados pelo algoritmo genético com as configurações descritas anteriormente utilizando 20 cadeias na avaliação da população. Por último, na coluna "GA5", estão as relações obtidas dos autômatos gerados com apenas 5 cadeias usadas na avaliação da população.

Nos dois casos, fixou-se um padrão de cromossomo que faz com que os indivíduos da primeira população tenham todas as transições desativadas e todos os estados sejam de não aceitação. Iniciando a primeira população aleatoriamente obtém-se resultados bem abaixo do esperado, com relações iguais a 0 em praticamente todos os casos.

	Alvo	**GA20**	**σ**	**GA5**	**σ**
a*	0,86	0,53	0,16	0,43	0,21
(ab)*	0,71	0,57	0,19	0,49	0,25
s/ mais de 2 a's seg.	0,57	0,39	0,19	0,53	0,16
N# par de a's e b's	0,43	0,39	0,16	0,4	0,28
a*b*a*b*	0,43	0,28	0,2	0,33	0,13
a*b	0,71	0,44	0,22	0,4	0,26
(a* + c*)b	0,43	0,39	0,23	0,26	0,15
(aa)*(bbb)*	0,29	0,27	0,25	0,29	0,19
a(aa)*b	0,57	0,3	0,23	0,33	0,16
N# par de a's	0,71	0,64	0,09	0,64	0,11
(aa)*ba*	0,57	0,31	0,19	0,24	0,18
bc*b+ac*a	0,43	0,36	0,27	0,33	0,18
Médias	0,56	0,41	0,20	0,39	0,19

Tabela 1: Média e desvio padrão das relações de tamanho dos autômatos gerados pelo algoritmo.

X. RESULTADOS OBTIDOS

Os resultados obtidos, comparados com os autômatos alvos, não foram satisfatórios. Primeiramente, analisando a execução do algoritmo com a primeira população iniciada aleatoriamente, a baixa qualidade dos resultados indica que esse método não é bom em aperfeiçoar soluções já existentes, pois não conseguiu diminuir a quantidade de estados dos autômatos.

Para as outras duas configurações GA20 e GA5, há dois exemplos que deixam as deficiências do algoritmo bastante evidentes. O primeiro é o caso da linguagem com a quantidade par de ***a***'s, analisando mais detalhadamente os resultados é possível observar que o algoritmo inferiu autômatos com uma quantidade de estados menor que o autômato alvo. Isso aconteceu porque além das cadeias com quantidade par de ***a***'s eles aceitam cadeias com quantidade impar. Isso é devido a não utilização de exemplos negativos na avaliação da população, não houve pressão evolutiva alguma garantindo que os autômatos reconhecedores apenas das cadeias com quantidade par de ***a***'s predominassem no ambiente.

O segundo exemplo é o da linguagem ***a*b***, o autômato alvo para essa linguagem é bastante trivial e possui apenas dois estados, analisando os resultados é possível verificar que o algoritmo nunca conseguiu gerar um autômato com dois estados apenas. Isso ocorre pelo mesmo motivo que fez com que a configuração do algoritmo com a primeira população iniciada aleatoriamente tivesse um desempenho tão baixo, o algoritmo não consegue aperfeiçoar a solução. Ele encontrou uma solução com três estados, mas ficou preso nesse ótimo local.

Por último, foi possível observar que os resultados obtidos na configuração GA5 tiveram um desvio padrão maior que os obtidos com a configuração GA20. Isso ocorre porque a variação da quantidade de cadeias na avaliação dos indivíduos altera a avaliação do autômato adaptativo de referência. Utilizando mais cadeias este autômato perde eficiência na utilização das transições, quantidade de estados e estados de aceitação. Como a avaliação dos indivíduos é feita em relação à do autômato de referência, com a configuração GA5, que utiliza uma quantidade menor de cadeias, a avaliação fica mais rigorosa. Uma avaliação mais rigorosa varia menos, isto

[2] Nesse caso, a relação é inversa, pois quanto menor mais eficiente é o autômato, por tanto, foi utilizado o seguinte cálculo: 1- v/T, sendo v o valor que pode variar na relação (numero de estado e o número de estados finais) e T é o valor total, que é a base da relação (no caso o número máximo de estados).

[3] Idem nota 1.

dificulta ainda mais que o algoritmo escape de ótimos locais, pois evita a convergência das medidas em um único ponto, aumentando a variação dos resultados obtidos.

XI. CONCLUSÃO

A utilização de algoritmos genéticos na inferência de autômatos finitos não é trivial, isso porque, no espaço de soluções de autômatos existe uma quantidade considerável de ótimos locais, onde o algoritmo acaba preso. Além disso, ficou evidente que é muito importante a utilização de exemplos negativos na avaliação da população.

Para obter melhores resultados será necessário melhorar o mecanismo de seleção dos indivíduos, utilizar a *rolette whell* simples não pareceu ser suficiente, principalmente quanto a questão dos ótimos locais. Também a utilização de *crossover* de dois pontos pode melhorar a recombinação dos cromossomos. Por último, é interessante melhorar os métodos de avaliação dos indivíduos, por exemplo, no lugar de usar simplesmente uma Arvore de prefixo, pode ser mais proveitoso utilizar o autômato gerado pelo método proposto em [5], que é um autômato mais eficiente e próximo do ótimo.

REFERÊNCIAS

[1] P. Hingston, “A Genetic Algorithm for Regular Inference”, 2001 Genetic and Evolutionary Computation Conference. GECCO 2001, pp. 1299-1306. Morgan Kaufmann, July 2001.

[2] V. Fabera, “Automata Construct with genetic algorithm”, 2006 9th EUROMICRO Conference on Digital System Design (DSD'06) pp. 460-463

[3] J. J. Neto, “Adaptive Rule-Driven Devices - General Formulation and Case Study. Lecture Notes in Computer Science”, Watson, B.W. and Wood, D. (Eds.): Implementation and Application of Automata 6th International Conference, CIAA 2001, Vol.2494, Pretoria, South Africa, July 23-25, Springer-Verlag, 2001, pp. 234-250.

[4] J. J. Neto, “Contribuições à metodologia de construção de compiladores”, Tese de Livre Docência, USP, São Paulo,1993.

[5] J. J. Neto e M. K. Iwai, “Adaptive Automata for Syntax Learning”, CLEI 98 - XXIV Conferencia Latinoamericana de Informatica, MEMORIAS. pp. 135-149, Quito, Equador, 1998

[6] J. H. Holland, “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, 1975

[7] J. Kennedy e R. Eberhart, “Swarm Intelligence”, San Francisco, California – USA: Morgan Kaufmann Publishers, 2001, 512p.