AUTÔMATO ADAPTATIVO, LIMITES E COMPLEXIDADE EM COMPARAÇÃO COM A MÁQUINA DE TURING

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

Neste trabalho é realizado um estudo a respeito do autômato adaptativo, buscando encontrar limites e equivalências em termos de processamento computacional com a máquina de Turing. É apresentado o modelo de autômato adaptativo segundo [2]. A partir desta apresentação, é traçada a estratégia de simular o funcionamento de uma máquina de Turing qualquer, e, a partir da simulação, comparar os resultados obtidos com os resultados produzidos pela máquina de Turing original. Esta estratégia permite encontrar limites para as classes de complexidade em tempo e espaço ocupado. Os resultados obtidos mostram que o autômato adaptativo, simulando uma máquina de Turing que apresente resposta a uma cadeia de comprimento n em tempo T(n) e utilizando espaço S(n), tem sua complexidade em tempo limitada por O[T(n)], e em espaço limitada por $O[S^2(n)]$.

ABSTRACT

This paper presents the result of a study on adaptive automata, regarding limits and equivalences to Turing Machines. Adaptive automata are represented following their original definition. Then a strategy is designed to simulate general Turing Machines with adaptive automata. Results are compared, allowing time and space-complexity limits to be searched for. For an adaptive automaton simulating a Turing Machine with time-response T(n) and demanding space S(n) when processing an input with length n, the results achieved were respectively O[T(n)] and $O[S^2(n)]$.

PALAVRAS-CHAVE

Teoria da Computação, Modelos computacionais, Complexidade Computacional, Autômatos, Máquina de Turing.

INTRODUÇÃO

Um modelo de autômato representado como um formalismo adaptável introduz novas modificações sobre o conceito de autômato, dotando-o da capacidade de se automodificar conforme as necessidades encontradas, o que lhe concede o poder de representação da Máquina de Turing [1], [2] e [4]. Este formalismo pode ser também compreendido como um complemento à clássica teoria de autômatos [1] e [2]. Assim, este formalismo apresenta-se como uma formulação hierárquica do caso mais geral do autômato.

Um formalismo adaptável, apresentado em termos de autômatos, pode ser visto como uma máquina de estados representada por um grafo orientado inicial, que inicialmente se apresenta como um autômato finito ou de pilha, mas que, à medida em que lhe são impostas alterações durante sua operação (alterações que são geradas pela própria movimentação do autômato e que resultam da aplicação de suas regras de transição), o grafo inicial pode ser modificado.

Com isso, estados e transições podem ser eliminados ou acrescentados ao modelo de autômato, em decorrência de cada um dos passos executados pelo mesmo durante sua operação.

Durante a execução de uma transição adaptativa, o autômato pode sofrer mudanças em sua própria topologia. Isto faz com que uma nova máquina de estados substitua a anterior, caracterizando, para o autômato, o percurso de um passo adicional, em uma trajetória no espaço das máquinas de estado.

A trajetória em um espaço de máquinas de estado é gerada porque a máquina inicial não está mais presente, portanto, o autômato começa seu trabalho em uma máquina específica e vai continuá-lo em outra máquina, descrevendo uma trajetória em um espaço de trabalho que contempla todas as configurações de autômato possíveis.

Autômatos adaptativos

O modelo de autômato adaptativo representa um caso especial de formalismo adaptável [1], que serve como base para esta pesquisa.

Um autômato adaptativo *M* é constituído de:

- ω - Fita de entrada
- E_0 - Autômato inicial que implementa *M*
- E_m - Autômato final que implementa *M* após a aceitação da cadeia de entrada $\omega = \alpha_0\alpha_1 \ldots \alpha_m$ $(m \geq 0)$, onde cada α_i, $0 \leq i \leq m$, representa uma sub-cadeia de ω
- E_i - Autômato após a i^ésima transição adaptativa, $0 \leq i \leq m$

Na aceitação de uma cadeia de entrada ω, composta de sub-cadeias α_k, $0 \leq k \leq m$, a operação do autômato *M* pode ser visualizada macroscopicamente como uma seqüência de reconhecimentos das sub-cadeias α_i pelas sub-máquinas correspondentes E_i $(0 \leq i \leq m)$.

Assim, *M* descreve um caminho $< E_0, \alpha_0 > \rightarrow < E_1, \alpha_1 > \rightarrow \ldots \rightarrow < E_m, \alpha_m >$, em que cada elemento $< E_i, \alpha_i >$ representa a aceitação de α_i pela sub-máquina correspondente E_i e a disposição de elementos $< E_i, \alpha_i >$, um após o outro, denota a seqüência de execução das transições adaptativas [2]. Uma transição adaptativa pode conter ações adaptativas de adição, remoção ou inspeção de transições.

Os tipos possíveis de transições são os seguintes:

- Transições entre submáquinas: Chamada/retorno de submáquina, são transições, executadas sem consumo de símbolos da cadeia de entrada, que permitem transferir o controle entre uma submáquina e outra;
- Transições internas: São transições que ocorrem dentro da mesma submáquina, com ou sem consumo de símbolos da cadeia de entrada;
- Transição em vazio: São transições internas sem consumo de símbolos;
- Transições adaptativas: São quaisquer das transições anteriores, às quais estejam associadas ações adaptativas especificando a alteração da do conjunto de transições do autômato adaptativo.

Uma transição adaptativa é descrita por uma quádrupla [2] $P_i = (t_i, A_i, u_i, B_i)$, em que:

t_i - representa a situação do autômato, anterior à transição

A_i - representa a ação adaptativa a ser executada antes da transição

u_i - representa a situação do autômato, posterior à transição

B_i - representa a ação adaptativa a ser executada depois da transição.

Operação do autômato adaptativo

A operação do autômato adaptativo dá-se através de sucessivas transformações em sua topologia, de forma que, partindo da topologia inicial, o autômato adaptativo pode sofrer modificações de ampliação ou de redução em seu conjunto de estados e transições iniciais. Estas transformações são provocadas pela execução das transições adaptativas.

O autômato adaptativo pode exibir um comportamento similar ao autômato finito ou de pilha estruturado, quando os problemas apresentados puderem ser representados e tratados por estes, isto é, para o autômato adaptativo a utilização de transições adaptativas não é compulsória. Entretanto, existe a possibilidade de aparecimento, na cadeia de entrada, de uma construção que não possa ser tratada pelos autômatos finitos ou de pilha, como é o caso, por exemplo, de uma construção dependente de contexto (no caso de reconhecedores de linguagens dependentes de contexto). Neste caso, o autômato adaptativo é capaz de resolver o problema, com o auxílio do recurso da alteração dos seus próprios conjuntos de estados e de regras de transição, de forma que o autômato modificado seja capaz de reconhecer a construção encontrada [1].

Algumas ações básicas podem ser executadas no autômato adaptativo, tais como, por exemplo, eliminação e inserção de transições, modificando assim o comportamento do autômato. Outra ação básica é a de inspeção, que permite obter a característica de um elemento específico na topologia corrente do autômato (após sucessivas modificações na topologia, pode ser necessário verificar uma determinada característica, que pode se encontrar diferente em relação à topologia inicial). Maiores detalhes podem ser encontrados em [1] e [2].

DESENVOLVIMENTO

Inicialmente, introduz-se algumas restrições ao comportamento do autômato adaptativo, de forma a facilitar a verificação de seus limites superiores em termos de ocupação de espaço e de tempo. Essas restrições possibilitam uma comparação mais simples entre um modelo de autômato adaptativo e um modelo de máquina de Turing.

As restrições impostas são:

1. O modelo de autômato adaptativo a ser estudado não utiliza a pilha nas transições adaptativas.
2. Considera-se que a execução de cada transição adaptativa representa um passo de computação.
3. Todos os estados iniciais e os gerados por ações adaptativas permanecem, mesmo após uma ação de remoção. A ação de remoção apenas elimina as transições sem remover os estados por elas referenciados.

Através destas restrições, é possível acompanhar com maior facilidade a trajetória descrita por um modelo de autômato adaptativo, bem como estimar o total de estados gerados pelo processamento de funções adaptativas.

Seja M uma máquina de Turing, na qual se tenha: $M = \langle Q, Z, T, \delta, q_0 \rangle$. Supõe-se, sem perda de generalidade, que a fita da máquina de Turing é limitada à esquerda e infinita à direita, que a cadeia a ser aceita está à esquerda da cabeça de leitura e gravação, e que:

- Q representa o conjunto de estados da máquina, excluindo o estado de parada (h)
- Z representa o alfabeto da fita da máquina de Turing, e inclui o símbolo #
- $T \subseteq Z - \{\#\}$ representa o alfabeto de entrada da máquina
- $\delta: Q \times Z \rightarrow (Q \cup \{h\}) \times \{L, R, W(\sigma)\}$, $\sigma \in Z$, δ representa a função de transferência da máquina
- $q_0 \in Q$, representa o estado inicial da máquina

Os símbolos L, R, W(σ) representam respectivamente as ações da máquina de Turing para mover a cabeça de leitura e gravação uma posição à esquerda, mover a cabeça de leitura e gravação uma posição à direita, e gravar o símbolo σ na fita, na posição indicada pela posição da cabeça de leitura e gravação. A fita está totalmente preenchida com "#", fora da posição em que está gravada a cadeia de entrada.

Para a demonstração de alguns Lemas e Teoremas foi utilizada uma ação especial "??", que não foi definida originalmente para o autômato adaptativo [2], e aqui representa uma ação condicional. Caso a condição referida seja verdadeira, a ação tem efeito. Pode-se realizar a mesma função sem o auxílio desta ação, entretanto, é necessário que toda e qualquer ação de movimentação à direita acrescente um símbolo "#" na cadeia. Portanto, assume-se aqui a existência de uma macro "?? [⟨condição⟩]" que realiza tal ação, de forma a simplificar o tratamento e também diminuir o número de estados ao final da execução.

Seguindo o raciocínio a ser desenvolvido adiante, através do qual se utiliza construtivamente um modelo de autômato adaptativo para simular uma fita contendo símbolos quaisquer, depois a fita de uma máquina de Turing, posteriormente mostrar que as ações básicas de uma máquina de Turing podem ser simuladas por funções adaptativas, pode-se demonstrar que o autômato adaptativo tem poder computacional equivalente à máquina de Turing.

Lema A.1: É possível se construir um autômato adaptativo que seja capaz de identificar o final de uma cadeia de entrada, contida em sua fita.

Demonstração: Suponha a existência de um modelo de autômato adaptativo qualquer, que opera uma cadeia de entrada, colocada sobre sua fita. Pode-se então, sem alterar o funcionamento deste modelo de autômato, acrescentar um símbolo ao alfabeto de entrada, "#" que tem por objetivo indicar fim de cadeia. Sem perda de generalidade, admite-se que todas as cadeias de entrada terminam com o símbolo acrescentado. Assim, basta acrescentar uma transição aos estados finais do modelo de autômato, uma transição que levando ao próprio estado final, consuma o símbolo "#".

■

Lema A.2: É possível construir um autômato adaptativo que, tendo lido toda uma cadeia de entrada antes de processá-la, ainda assim possua todas as informações sobre ela, e com isso possa operar através dela.

Demonstração: Suponha a existência de um modelo de autômato adaptativo qualquer, que opera uma cadeia de entrada, colocada sobre sua fita. Partindo desse modelo de autômato, constrói-se um outro modelo semelhante, porém, com a seguinte característica: Toda a cadeia é lida em um estado especial, colocado antes do estado inicial do modelo anterior de autômato. Pelo lema anterior, pode-se deterministicamente consumir todos os símbolos da cadeia e identificar que esta foi completamente lida.

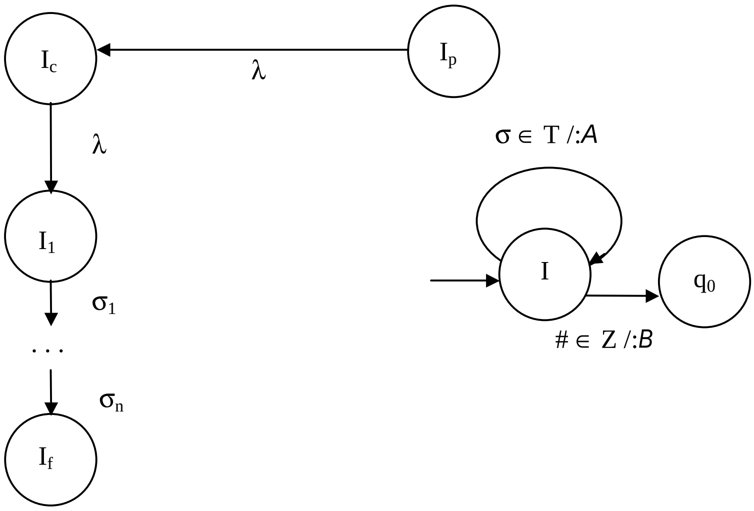
Seja I este estado especial e, além deste, acrescente-se três outros estados: o estado apontador de cadeia, I_p, o estado inicial de cadeia, I_c, e o estado final de cadeia I_f. O estado inicial de cadeia está ligado por uma transição vazia, transição-λ (sem consumo de símbolos) ao estado final de cadeia e o estado apontador de cadeia também está ligado inicialmente ao estado inicial de cadeia, por uma transição-λ.

No estado I, tem-se as seguintes transições:

a) Todo símbolo $\sigma \neq \#$ é consumido, e há uma função adaptativa que conecta o estado anteriormente criado (através de uma ação adaptativa de inspeção, descobre-se qual é o estado cujo destino é o estado I_f desejado) a um estado criado através de um gerador, cujo índice inicial é 1, e este com I_f, usando o símbolo consumido (supõe-se que já esteja criada a estrutura entre I_p, I_c e I_f);

b) O símbolo # é consumido através de uma transição adaptativa cuja ação conecta o estado I com o estado q_0, o primeiro estado do modelo original.

Esquematicamente tem-se:



As funções adaptativas têm o seguinte formato (σ representa o símbolo lido):

A (σ, x, s): { j* :
? [(x, s) $\rightarrow$ (I_f, s)] // 1. Descobre o último estado de cadeia
+ [(j, $\sigma\alpha$) $\rightarrow$ (I_f, α)] // 2. Adiciona um novo estado à cadeia
- [(x, s) $\rightarrow$ (I_f, s)] // 3. Remove a conexão com o fim da cadeia
+ [(x, s) $\rightarrow$ (j, s)] } // 4. Adiciona nova conexão com o fim

B (x, s): { j* :
? [(I_c, s) $\rightarrow$ (x, s)] // 1. Descobre conexão com o 1º estado da cadeia
- [(I_p, s) $\rightarrow$ (I_c, s)] // 2. Remove a conexão com o marcador
+ [(I_p, s) $\rightarrow$ (x, s)] } // 3. Adiciona conexão com o primeiro símbolo

O símbolo j* simboliza a geração de um estado novo, único [2]. Assim, toda a cadeia está disposta de forma ordenada em termos de transições simples, não adaptativas, interligando estados cujos índices indicam os valores de ordem dos próprios elementos da cadeia, iniciando no estado I_c e finalizando no estado I_f, com seu primeiro elemento apontado por I_p. Isto completa a demonstração, já que o lema solicita apenas a possibilidade de conservação dos valores da cadeia lida.

■

<u>Lema A.3</u>: É possível se construir um autômato adaptativo que simule o funcionamento da fita de uma máquina de Turing.

<u>Demonstração</u>: Usando o Lema anterior, produz-se, de forma semelhante, uma estrutura de transições para armazenar toda a cadeia de entrada. Entretanto, deve-se permitir o aumento da estrutura que representa a cadeia, já que uma máquina de Turing possui, como uma de suas possíveis ações, o movimento de uma posição à direita. As ações de movimentar à esquerda, bem como de escrever sobre a fita, são ações simples adaptativas que removem uma transição e inserem outra em seu lugar. Portanto, basta criar uma função que simule o movimento à direita da cadeia, designada por *R*.

R (x, y, s, t): { j* :
? [(I_p, s) $\rightarrow$ (x, s)] // 1. Descobre o estado da cadeia com o símbolo
?? [x = I_f] // 2. Caso o estado seja o último marcador:

$\{ ? [(y, t\alpha) \rightarrow (I_f, \alpha)]$ // 2.1. Descobre o último estado da cadeia
$+ [(j, \#\alpha) \rightarrow (I_f, \alpha)]$ // 2.2. Adiciona uma transição consumindo #
$- [(y, t\alpha) \rightarrow (I_f, \alpha)]$ // 2.3. Remove transição do último estado
$+ [(y, t\alpha) \rightarrow (j, \alpha)] \}$ // 2.4. Adiciona transição ao último estado
$?? [x \neq I_f]$ // 3. Caso o estado não seja o último marcador:
$\{ ? [(x, t\alpha) \rightarrow (y, \alpha)]$ // 3.1. Descobre a transição (estado) da cadeia
$- [(I_p, s) \rightarrow (x, s)]$ // 3.2. Remove o apontador antigo da cadeia
$+ [(I_p, s) \rightarrow (y, s)] \} \}$ // 3.3. Adiciona novo apontador à cadeia

Com isso, o lema fica demonstrado.

■

Lema A.4: É possível se construir um autômato adaptativo que simule e rastreie as ações de um modelo de autômato qualquer, através de funções adaptativas.

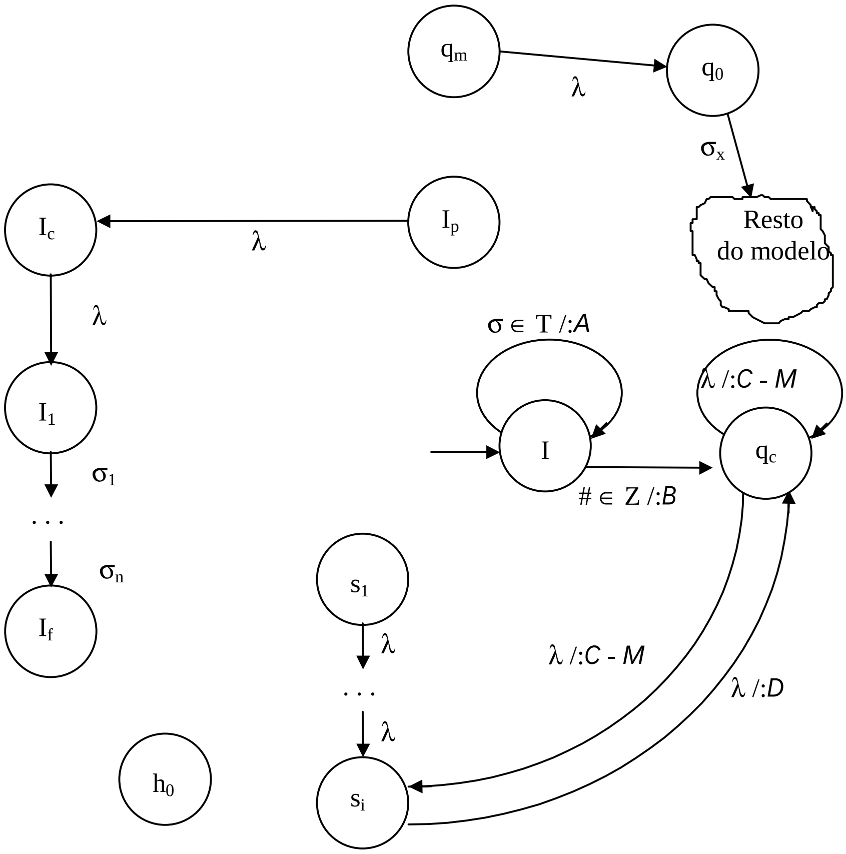
Demonstração: Suponha a existência de um modelo de autômato adaptativo qualquer, que opera uma cadeia de entrada, colocada sobre sua fita. Partindo desse modelo de autômato e sem manipulá-lo diretamente, constrói-se um outro modelo semelhante, porém com a seguinte característica: toda a cadeia é lida em um estado especial, colocado antes do estado inicial do modelo anterior de autômato.

Pelo Lema anterior, pode-se deterministicamente consumir todos os símbolos da cadeia e identificar que esta foi completamente lida. Além disso, há outras duas características especialmente importantes: existe um estado especial, marcador (identificador) de estados no autômato original e um estado de parada (h_0), e um outro estado especial com duas funções adaptativas de controle de transições a serem simuladas e acompanhadas.

A primeira suposição realizada para a demonstração deste lema utiliza exatamente a mesma construção proposta no Lema A.2. Isto garante a possibilidade de se percorrer a fita de entrada tantas vezes quantas se fizerem necessárias. Suponha ainda que há um conjunto de estados finais F, no autômato original a ser simulado, que é conhecido. A segunda suposição é realizada para garantir que é possível acompanhar a seqüência de estados e transições em um modelo de autômato qualquer e, através de uma ação adaptativa, construir-se um caminho equivalente ao traçado pelo autômato original, que, entretanto não será executado.

Seja q_m o estado marcador. A fim de que este estado aponte sempre para um estado no modelo de autômato a ser simulado, constrói-se uma transição vazia (transição-λ), apontando primeiramente para o estado inicial do modelo, o estado q_0. Há, então, um outro estado, suponha-se q_c, o estado de controle das ações e *C* a função adaptativa de controle. Através do estado q_c constrói-se e acompanha-se a execução simulada do modelo original. A outra função adaptativa, *D*, remove a transição anterior que parte de q_c, construindo uma nova, que liga o estado de controle ao estado corrente.

Através da função adaptativa de controle *C*, os estados simulados (s_i) são construídos e sua seqüência é determinística, sem ciclos. A execução é então realizada por estes estados. Desta forma, a construção do modelo de autômato que simula e acompanha um outro modelo qualquer pode ser representada esquematicamente como:



Observa-se então, que a execução será sempre através dos estados simulados, os quais retornam o controle sempre ao estado q_c, após sua execução, através de uma transição-λ. Para garantir a execução de somente uma transição, a fita de entrada é acompanhada pela função especial *M*, enquanto a funções especiais *C* e *D* constróem e acompanham os estados simulados. As funções ficam assim:

C (σ, x, s, y, t, z, w, m, k, n, l, *F*, *G*): { j* :
- ? [(q_m, s) $\rightarrow$ (x, s)] // 1. Descobre o próximo estado real (x)
- ? [(I_p, α) $\rightarrow$ (m, α)] // 2. Descobre qual o próximo símbolo na fita
- ? [(m, $\sigma\alpha$) $\rightarrow$ (n, α)] // e armazena em σ
- ?? [$\sigma \neq \lambda$ <u>OU</u> x $\notin$ F] // 3. Caso o estado não seja o último:
- ? [(x, σs): *F* $\rightarrow$ (y, s): *G*] // 3.1. Descobre próxima transição (salva *F* e *G*)
- ? [(q_c, w): *C* $\rightarrow$ (z, w): *M*] // 3.2. Descobre o próximo estado simulado
- \+ [(z, $\sigma\alpha$): *F* $\rightarrow$ (j, $\sigma\alpha$): *G*] // 3.3. Cria nova transição simulando a real
- \+ [(j, α): *D* $\rightarrow$ (q_c, α)] // 3.4. Cria transição retorno partindo do novo
- \- [(z, α): *D* $\rightarrow$ (q_c, α)] // 3.5. Remove a transição de retorno anterior
- ?? [$\sigma = \lambda$ <u>E</u> x $\in$ F] // 4. Caso o estado seja o último:
- ? [(q_c, w): *C* $\rightarrow$ (z, w): *M*] // 4.1. Descobre o próximo estado simulado
- \- [(z, α): *D* $\rightarrow$ (q_c, α)] // 4.2. Remove a transição de retorno anterior
- \+ [(z, α) $\rightarrow$ (h_0, α)] } // 4.3. Cria nova transição simulando a real

D (x, s, y, t, *F*, *G*): {
- ? [(q_c, s): *C* $\rightarrow$ (x, s): *M*] // 1. Descobre última transição p/ estado simulado
- \- [(q_c, s): *C* $\rightarrow$ (x, s): *M*] // 2. Remove a transição

? [(x, s): *F* → (y, t): *G*] // 3. Descobre o próximo estado simulado
+ [(q_c, s): *C* → (y, s): *M*] } // 4. Cria nova transição para estado simulado

M (x, s, y, t): {
? [(I_p, s) → (x, s)] // 1. Descobre conexão com estado da cadeia
- [(I_p, s) → (x, s)] // 2. Remove a conexão com estado
? [(x, s) → (y, t)] // 3. Descobre a conexão com o próximo estado
+ [(I_p, s) → (y, s)] } // 4. Adiciona conexão com o próximo estado

Por sua vez, a função *B* deve ser modificada, para garantir que o estado q_c aponte sempre para um estado inicial e que os símbolos da cadeia possam assim ser encontrados. Tem-se, portanto:

B (x, s): {
? [(I_c, s) → (x, s)] // 1. Descobre conexão com o 1º estado da cadeia
- [(I_p, s) → (I_c, s)] // 2. Remove a conexão com marcador
+ [(I_p, s) → (x, s)] // 3. Adiciona conexão com o primeiro símbolo
+ [(q_c, s) → (q_0, s)] } // 4. Adiciona conexão com o primeiro estado

Desta forma, fica marcado o caminho determinístico (no sentido de que somente uma transição é possível por estado) traçado pelo autômato adaptativo simulado, e é possível acompanhar sua trajetória até sua parada. Com isso, o lema fica demonstrado.

■

Lema A.5: É possível se construir um autômato adaptativo que, lendo toda uma cadeia de entrada e possuindo a informação sobre os estados de uma máquina de Turing, é capaz de acompanhar e rastrear o funcionamento da máquina, através da criação de transições que mostrem a seqüência de passos de configuração da máquina de Turing, através de funções adaptativas.

Demonstração: Usando-se os Lemas anteriores, A.3 e A.4, pode-se construir as funções adaptativas que faltam para simular completamente as ações da máquina de Turing.

Seja R a ação da máquina de Turing que movimenta o cabeçote de leitura e gravação uma posição à direita, L à esquerda, W(σ) a ação de escrita do símbolo σ na fita da máquina, e seja h o estado final da máquina de Turing. A fita da máquina de Turing, como visto no lema A.3, pode ser simulada. Vai-se, então, construir apenas as funções adaptativas *R*, *L* e *W*, que simulam as ações da máquina.

A função *R* já foi descrita anteriormente, no lema A.4. Cabe, portanto, construir as funções *L* e *W*. A função *L* movimenta o cabeçote à esquerda, até o limite da fita, ou seja, pelo modelo de fita utilizado na simulação, há um limite à esquerda conhecido, o estado I_c. Desta forma, basta acompanhar as transições, em sentido contrário, até encontrar o limite estabelecido.

Já a função *W* escreve sobre a fita da máquina de Turing um símbolo, independentemente do valor do símbolo que estava preenchido anteriormente no mesmo espaço. Utilizando a fita simulada, basta substituir a transição original por uma nova, na qual o símbolo corrente seja substituído pelo novo. Assim tem-se:

L (x, y, s, t): {
? [(I_p, s) → (x, s)] // 1. Descobre o estado da cadeia com o símbolo

```
    ?? [ x ≠ Ic]                    // 2. Caso estado não seja primeiro marcador:
      { ? [ (y, tα) → (x, α)]       // 2.1. Descobre o estado anterior na cadeia
        - [ (Ip, s) → (x, s)]       // 2.3. Remove transição ao estado
        + [ (Ip, s) → (y, s)] } }   // 2.4. Adiciona transição ao estado anterior
W (x, y, s, t, σ):        {
    ? [ (Ip, s) → (x, s)]           // 1. Descobre estado da cadeia com o símbolo
    ? [ (x, tα) → (y, α)]           //  e armazena em tal símbolo em t
    - [ (x, tα) → (y, α)]           // 2. Remove a transição anterior
    + [ (x, σα) → (y, α)] }         // 3. Adiciona a transição com o novo símbolo
```

Com isso, todas as ações da máquina de Turing podem ser simuladas sobre a fita construída, completando a demonstração.

■

Teorema A.6: Um autômato adaptativo pode simular a operação de uma máquina de Turing qualquer.

Demonstração: Usando-se os Lemas A.3, A.4 e A.5 anteriores e, sabendo-se que somente há um estado final para a máquina de Turing – h – pode-se construir uma réplica da máquina de Turing, em forma de funções adaptativas. Tal réplica servirá de base para a simulação, por um autômato adaptativo, procedendo exatamente como no Lema A.4, além de construir uma função especial *H*, para indicar a condição de parada (último estado).

Para isso será necessário construir a réplica, partindo de sua função de transição, assim tem-se (supondo aqui que, se X representa uma das ações da máquina de Turing, então *X* representa a função adaptativa correspondente, e que a função *H* é executada após a transição correspondente):

Máquina de Turing	**Réplica de Autômato**
$\delta(\sigma, R)$	$\delta(\sigma, R)$
$\delta(\sigma, L)$	$\delta(\sigma, L)$
$\delta(\sigma, W(\sigma_1))$	$\delta(\sigma, \mathit{W}(\sigma_1))$
$\delta(\sigma, X) \Rightarrow h$	$\delta(\sigma, \mathit{X})$: H

Através desta tabela, constrói-se o modelo de autômato que tem a função de servir de cópia para a simulação, e então, através dos Lemas anteriores, pode-se garantir que o autômato construído é capaz de simular uma máquina de Turing qualquer. Entretanto, é necessário introduzir algumas modificações, já que a função *M* não é mais necessária, que a função *C* precisaria encontrar apenas um estado final h, e que a função *H* constrói o caminho para o estado de parada. Tem-se, então, as seguintes funções adaptativas:

```
A (σ, x, s):     {        j* :
    ? [ (x, s) → (If, s)]           // 1. Descobre o último estado de cadeia
    + [ (j, σα) → (If, α)]          // 2. Adiciona um novo estado à cadeia
    - [ (x, s) → (If, s)]           // 3. Remove a conexão com o fim da cadeia
```

$\quad + [\ (x, s) \rightarrow (j, s)]\ \}$	// 4. Adiciona nova conexão com estado j adicionado
$B\ (x, s): \quad \{$	
$\quad ?\ [\ (I_c, s) \rightarrow (x, s)]$	// 1. Descobre conexão com o 1º estado da cadeia
$\quad -\ [\ (I_p, s) \rightarrow (I_c, s)]$	// 2. Remove a conexão com marcador
$\quad + [(I_p, s) \rightarrow (x, s)]$	// 3. Adiciona conexão com o primeiro símbolo
$\quad + [(q_c, s) \rightarrow (q_0, s)]\ \}$	// 4. Adiciona conexão com o primeiro estado
$C\ (\sigma, x, s, y, t, z, w, m, k, n, l, F, G): \{ \quad j^*:$	
$\quad ?\ [\ (q_m, s) \rightarrow (x, s)]$	// 1. Descobre o próximo estado real (x)
$\quad ?\ [\ (I_p, \alpha) \rightarrow (m, \alpha)]$	// 2. Descobre qual o próximo símbolo na fita
$\quad ?\ [\ (m, \sigma\alpha) \rightarrow (n, \alpha)]$	// e armazena em σ
$\quad ?\ [\ (x, \sigma t): F \rightarrow (y, t): G]$	// 3. Descobre a próxima transição (salva *F* e *G*)
$\quad ?\ [\ (q_c, w): C \rightarrow (z, w): M]$	// 4. Descobre o próximo estado simulado
$\quad + [\ (z, \sigma\alpha): F \rightarrow (j, \sigma\alpha): G]$	// 5. Cria nova transição simulando a real
$\quad + [\ (j, \alpha): D \rightarrow (q_c, \alpha)]$	// 6. Cria transição de retorno a partir do novo estado
$\quad -\ [\ (z, \alpha): D \rightarrow (q_c, \alpha)]\ \}$	// 7. Remove a transição de retorno anterior
$D\ (x, s, y, t, F, G): \quad \{$	
$\quad ?\ [\ (q_c, s): C \rightarrow (x, s): M]$	// 1. Descobre a última transição p/ estado simulado
$\quad -\ [\ (q_c, s): C \rightarrow (x, s): M]$	// 2. Remove a transição
$\quad ?\ [\ (x, s): F \rightarrow (y, t): G]$	// 3. Descobre o próximo estado simulado
$\quad + [(q_c, s): C \rightarrow (y, s): M]\ \}$	// 4. Cria nova transição para o estado simulado
$M\ (x, s, y, t): \ \{\ \}$	
$R\ (x, y, s, t): \ \{ \quad j^*:$	
$\quad ?\ [\ (I_p, s) \rightarrow (x, s)]$	// 1. Descobre o estado da cadeia com o símbolo
$\quad ??\ [\ x = I_f]$	// 2. Caso o estado seja o último marcador:
$\quad\ \{\ ?\ [\ (y, t\alpha) \rightarrow (I_f, \alpha)]$	// 2.1. Descobre o último estado da cadeia
$\quad\quad + [\ (j, \#\alpha) \rightarrow (I_f, \alpha)]$	// 2.2. Adiciona uma transição consumindo #
$\quad\quad -\ [\ (y, t\alpha) \rightarrow (I_f, \alpha)]$	// 2.3. Remove a transição do último estado
$\quad\quad + [\ (y, t\alpha) \rightarrow (j, \alpha)]\ \}$	// 2.4. Adiciona a transição ao último estado
$\quad ??\ [\ x \neq I_f]$	// 3. Caso estado não seja o último marcador:
$\quad\ \{\ ?\ [\ (x, t\alpha) \rightarrow (y, \alpha)]$	// 3.1. Descobre transição (estado) da cadeia
$\quad\quad -\ [\ (I_p, s) \rightarrow (x, s)]$	// 3.2. Remove o apontador antigo da cadeia
$\quad\quad + [\ (I_p, s) \rightarrow (y, s)]\ \}\ \}$	// 3.3. Adiciona novo apontador à cadeia
$L\ (x, y, s, t): \ \{$	
$\quad ?\ [\ (I_p, s) \rightarrow (x, s)]$	// 1. Descobre estado da cadeia com o símbolo
$\quad ?\ [\ (y, t\alpha) \rightarrow (x, \alpha)]$	// 2. Descobre o estado anterior na cadeia
$\quad -\ [\ (I_p, s) \rightarrow (x, s)]$	// 3. Remove transição ao estado
$\quad + [\ (I_p, s) \rightarrow (y, s)]\ \}\ \}$	// 4. Adiciona transição ao estado anterior

Comentário: Caso se tente ir à esquerda do limite da fita, perder-se-á o contato com a mesma. Este efeito indica falha e é semelhante a se tentar ir à esquerda do limite da fita da máquina de Turing. Isto indica que os efeitos, apesar de diferentes, mostram a mesma ocorrência: um erro no algoritmo.

$W\ (x, y, s, t, \sigma): \quad \{$	
$\quad ?\ [\ (I_p, s) \rightarrow (x, s)]$	// 1. Descobre o estado da cadeia com o símbolo
$\quad ?\ [\ (x, t\alpha) \rightarrow (y, \alpha)]$	// e armazena em t
$\quad -\ [\ (x, t\alpha) \rightarrow (y, \alpha)]$	// 2. Remove a transição anterior

$+ [(x, \sigma\alpha) \to (y, \alpha)] \}$ // 3. Adiciona a transição com o novo símbolo

H (x, y, s, t, F, G): {

$? [(q_c, s): C \to (x, s): M]$ // 1. Descobre o próximo estado simulado

$? [(x, t): F \to (y, t): G]$ // 2. Descobre a próxima transição simulada

$- [(y, t): D \to (q_c, t)]$ // 3. Remove a transição de retorno anterior

$+ [(y, \alpha) \to (h_0, \alpha)] \}$ // 4. Cria nova transição simulando a real

Convém lembrar aqui o comentário introduzido na demonstração do Lema A.3, dando conta da possibilidade de não se utilizar a ação "??", que não foi definida no modelo original do autômato adaptativo. Como esta ação é usada somente no contexto da função adaptativa *R*, ela não é necessária, podendo ser substituída pelo acréscimo de algumas ações, que aumentam a cadeia original de um símbolo "#", cada vez que a função *R* é utilizada. Portanto, com esta construção é possível simular uma máquina de Turing qualquer, através de um modelo de autômato adaptativo, completando a demonstração.

■

<u>Teorema A.7</u>: Um autômato adaptativo, ao simular a operação de uma máquina de Turing qualquer e, desde que tal máquina dispenda em termos de espaço, em função do tamanho "n" da cadeia de entrada, S(n), então o autômato dispenderá em termos de espaço no máximo da ordem $O[S^2(n)]$.

<u>Demonstração</u>: Usando-se os Lemas A.3 e A.5 e o Teorema A.6, pode-se concluir que o autômato adaptativo simulador dispende um estado para cada símbolo da cadeia de entrada e, portanto, um estado para cada movimentação de ampliação efetuada na cadeia, na tentativa de acompanhá-la (Lema A.3). Logo, nesta tarefa, o autômato dispende um número de estados igual a S(n). Para cada estado da máquina de Turing há um correspondente, cuja função é servir de base para o acompanhamento da execução. Se C corresponde ao total de estados originais, então tem-se um total de S(n) + C.

No acompanhamento da máquina de Turing, o espaço ocupado, em termos de estados, das transições que de fato a simulam, é proporcional ao espaço ocupado, em termos de estados, da simulação da cadeia. Tem-se, assim, $2 \times S(n) + C$, correspondendo ao total de estados.

Entretanto, o espaço ocupado pode ser melhor comparado em função das transições. Supondo que cada estado possa, no limite, conter transições para todos os demais estados (basta observar a matriz de conectividade do grafo orientado do autômato), pode-se concluir, então, que o espaço ocupado é proporcional ao quadrado do valor anterior. Assim, o espaço ocupado é proporcional a $(2 \times S(n) + C)^2$. Logo, o espaço ocupado pelo autômato no limite superior é proporcional a $O[S^2(n)]$.

■

<u>Teorema A.8</u>: Um autômato adaptativo, ao simular a operação de uma máquina de Turing qualquer, e desde que tal máquina dispenda em tempo, em função do tamanho "n" da cadeia de entrada, T(n), então o autômato dispende tempo da ordem O[T(n)].

<u>Demonstração</u>: Usando-se os Lemas A.3 e A.5 e o Teorema A.6, pode-se concluir que o autômato adaptativo simulador dispende um passo de computação (uma transição adaptativa) para construir um novo estado ao simular cada estado da máquina de Turing, cuja função é servir de base para o acompanhamento da

execução. Então, se há T(n) passos de computação, haverá T(n) passos para construção de um estado correspondente (Teorema A.6).

Porém, além de construir, deve-se executar a transição gerada. Logo, dispende-se T(n) passos na execução, o que perfaz um total de $2 \times T(n)$ passos de computação. Portanto, o autômato adaptativo simulador dispende tempo da ordem O[T(n)] na simulação.

■

CONSIDERAÇÕES FINAIS

O autômato adaptativo é um modelo computacional poderoso (equivalente à máquina de Turing), e, a despeito das suposições formuladas, possui limites superiores em tempo e espaço bastante próximos dos encontrados na máquina de Turing. Os resultados obtidos mostram que o autômato adaptativo, simulando uma máquina de Turing que apresente resposta a uma cadeia de comprimento n em tempo T(n) e utilizando espaço S(n), tem como complexidade em tempo limitada por O[T(n)], e em espaço limitada por $O[S^2(n)]$.

Este resultado deve ser entendido como um estudo de situação extrema, em que o autômato adaptativo procura imitar o comportamento da máquina de Turing, simulando passo a passo seus movimentos. Na prática, porém, é possível constatar com facilidade que, por meio de autômatos adaptativos, é possível construir modelos de resolução de problemas que se mostram muito superiores em eficiência do que as máquinas de Turing clássicas, usualmente empregadas na mesma situação. Um conjunto significativo de casos em que os autômatos adaptativos mostram plenamente o seu valor como modelos de implementação eficiente de soluções para problemas complexos pode ser encontrado em [3].

Com isso, pode-se concluir que o modelo adaptativo representa uma alternativa à máquina de Turing como base de um processo computacional, o que amplia as opções existentes, como pode ser encontrado, por exemplo, em [7], [8] e em [9].

REFERÊNCIAS BIBLIOGRÁFICAS

[1] JOSÉ NETO, J. *Contribuições à metodologia de construção de compiladores*. São Paulo, 1993. 272p. Tese (Livre-Docência) - Escola Politécnica - Universidade de São Paulo.

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

[3] JOSÉ NETO, J. Solving Complex Problems Efficiently with Adaptive Automata - paper submitted to the *5th International Conference on Implementation and Application of Automata - CIAA 2000* - London - Ontario - Canada - July 24-26, 2000.

[4] SHUTT, J. N. Self-modifying finite automata - power and limitations. *Computer Science Technical Report Series*, WPI-CS-TR-95-4, Worcester Polytechnic Institute, Dec. 1995.

[5] KLEENE, S. C. Turing's analysis of computability, and major applications of it. In: HERKEN, R. ed. *The universal Turing machine*. New York, Springer-Verlag, 1995 p. 15-50.
[6] LEWIS, H. R.; PAPADIMITRIOU, C. H. *Elements of the theory of computation*. New Jersey, Prentice-Hall, Inc, 1998.
[7] PEREIRA, J. C. D.; JOSÉ NETO, J. Um ambiente de desenvolvimento de reconhecedores sintáticos baseados em autômatos adaptativos. Apresentado no *SBLP´97, II Simpósio Brasileiro em Linguagens de Programação*, Instituto de Computação, Universidade Estadual de Campinas, 3-5, p. 139-150, Set., 1997.
[8] 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.
[9] ROCHA, R. L. A.; NETO, J. J. Uma proposta de método adaptativo para a seleção automática de soluções. In VI International Congress on Information Engineering (ICIEY2K), Buenos Aires, Argentine, 2000, *Proceedings*, [CD-ROM].
[10] RUBINSTEIN, R.; SHUTT, J. N. Self-modifying finite automata. In: 13th IFIP World Computer Congress, Amsterdam, 1994. *Proceedings.* Technology and Foundations: Information Processing ´94, v. I., p. 493-498, 1994.
[11] SUDKAMP, T. A. L*anguages and Machines: an Introduction to the Theory of computer science*. Addison-Wesley, 1988.