

pilha antes e depois da aplicação da transição, respectivamente;

- s e s' representam o estado corrente da transição e o próximo estado, respectivamente;
- σ é o símbolo do alfabeto que será consumido da cadeia e σ' é o símbolo do alfabeto que será empilhado na cadeia (ambos podem ser iguais a cadeia vazia, representada pelo símbolo ϵ);
- α são meta-símbolos que representam o restante da cadeia não considerada pelo autômato (não influencia na aplicação da produção);
- $\mathcal{A}$ e $\mathcal{B}$ são chamadas a funções adaptativas a serem executadas antes e depois da mudança de estados determinada pela produção. As funções adaptativas podem ter n parâmetros e assumem a seguinte forma $\mathcal{A}(a_1, a_2, \dots, a_n)$, em que a_i são argumentos que assumirão valores a serem usados no corpo da função. As funções adaptativas são declaradas conforme descrito em [3].

III. MODELO PROPOSTO

No problema de emparelhamento de cadeias usando autômatos, para cada padrão a ser procurado no texto, um autômato que reconheça somente aquele padrão deve ser construído. A busca por um padrão em um texto é feita em duas etapas: a primeira etapa é a construção do autômato (pré-processamento); a segunda etapa é o uso do autômato construído para realizar a busca e encontrar todas as ocorrências do padrão no texto. A seguir, será descrito um modelo que utiliza autômatos adaptativos na fase de construção do autômato usado na busca. O modelo também possui uma função adaptativa que será usada somente durante a busca do padrão no texto.

Para facilitar a compreensão do modelo, o alfabeto foi limitado a apenas dois elementos. Suponha que o alfabeto seja $\Sigma = \{0, 1\}$ e que o padrão a ser buscado no texto seja $P = 01 \vdash$. O traço de asserção ($\vdash$) é usado para delimitar o final do padrão.

A idéia geral do modelo é que a cada símbolo lido no padrão, um novo estado seja criado e novas transições sejam inseridas para todos os símbolos do alfabeto. Para criar estas transições é necessário saber qual o estado destino para cada símbolo, pois é possível que haja sobreposição de símbolos de ocorrências sucessivas do padrão no texto. Por exemplo, se o padrão é 010 e o texto é 01010, então o último zero da primeira ocorrência do padrão é o primeiro zero da segunda ocorrência do padrão no texto.

Com o objetivo de criar um autômato determinístico para ser usado na busca, é necessário saber, para qualquer estado, qual o próximo estado para cada símbolo do alfabeto. Considerando-se as sobreposições possíveis, é preciso determinar se a função cria um novo estado ou cria uma transição para um estado já existente com cada símbolo do alfabeto.

Para solucionar este problema, é necessário encontrar o maior prefixo de P que é um sufixo da subcadeia que já é

reconhecida pelo autômato, concatenado com o símbolo para o qual será criada a transição. Uma cadeia w é um prefixo de uma cadeia x , se $x = wy$ para alguma cadeia $w \in \Sigma^*$ e alguma cadeia $y \in \Sigma^*$. Da mesma forma, uma cadeia w é um sufixo de uma cadeia x , se $x = yw$ para alguma cadeia $w \in \Sigma^*$ e alguma cadeia $y \in \Sigma^*$.

É necessário que a máquina conheça o comprimento do prefixo mais longo de P , que é um sufixo da subcadeia lida até o momento [1]. Este procedimento inicia-se com a passagem do símbolo lido no padrão para uma função adaptativa responsável por criar um novo estado em uma sub-máquina, que irá reconhecer o padrão lido até o momento. Em seguida, uma sub-máquina empilha na cadeia os símbolos anteriormente lidos em ordem invertida, que representam o prefixo de P , para fins de comparação com a sub-cadeia lida até o momento. Os símbolos são empilhados em ordem invertida porque é preciso verificar se este prefixo é um sufixo da cadeia lida até então.

Para separar o prefixo de P dos símbolos que ainda não foram lidos do padrão, é usado o símbolo especial $\#$. A sub-máquina que reconhece o padrão lido até o momento é executada consumindo os símbolos empilhados na cadeia até encontrar o símbolo $\#$. À medida que os símbolos são consumidos e a sub-máquina muda de estado, um estado equivalente no autômato final aponta para o estado de destino de acordo com o símbolo que se deseja inserir a transição. Portanto, cada vez que o sufixo é encontrado, uma máquina com um ponteiro para o estado correspondente indica qual estado deve receber a transição.

O autômato adaptativo para o problema de emparelhamento de cadeias será composto por cinco sub-máquinas. A sub-máquina principal responsável pelo reconhecimento do padrão no texto é Z . Nesta sub-máquina, $Z0$ é o estado inicial e ZF é o estado final. A definição de Z é mostrada abaixo.

$$\begin{aligned}
 (\epsilon, Z0, \epsilon) & \rightarrow (\epsilon, ZI, \epsilon) \\
 \{(\epsilon, ZI, \sigma\alpha) & \rightarrow (\epsilon, ZJ, \alpha), \mathcal{B}(\sigma) \mid \forall \sigma \in \Sigma\} \\
 (\epsilon, ZJ, \epsilon) & \rightarrow (\epsilon, ZK, \epsilon) \\
 (\gamma, ZK, \epsilon) & \rightarrow (\gamma ZQ, PK0, \epsilon) \\
 (\epsilon, ZQ, \epsilon) & \rightarrow (\epsilon, ZX, \epsilon), I(PK1, PC1) \\
 (\epsilon, ZX, \epsilon) & \rightarrow (\epsilon, ZU, \epsilon), J() \\
 (\gamma, ZU, \epsilon) & \rightarrow (\gamma ZT, PQ0, \epsilon) \\
 (\epsilon, ZT, \epsilon) & \rightarrow (\epsilon, ZW, \epsilon) \\
 (\epsilon, ZW, \vdash\alpha) & \rightarrow (\epsilon, ZV, \alpha), C() \\
 (\epsilon, ZW, \epsilon) & \rightarrow (\epsilon, ZI, \epsilon) \\
 (\epsilon, ZV, \epsilon) & \rightarrow (\epsilon, ZF, \epsilon) \\
 (\epsilon, PZ, \epsilon) & \rightarrow (\epsilon, PZ, \epsilon) \\
 (\epsilon, PO, \epsilon) & \rightarrow (\epsilon, PO, \epsilon)
 \end{aligned}$$

As sub-máquinas que representam, respectivamente, a sub-cadeia de P que já foi lida até o momento e que será usada para determinar o próximo estado de cada transição inserida, o prefixo de P que será inserido novamente na cadeia em ordem inversa, o contador de ocorrências do padrão no texto e o maior prefixo de P que é um sufixo da sub-cadeia lida, são PQ , PK , A e PC , conforme definidas abaixo. Os estados iniciais de PQ , PK , A e PC são $PQ0$, $PK0$, $A0$ e $PC0$, respectivamente. Da mesma forma, os estados finais são $PQ1$, $PK2$, $A1$ e $PC2$.

PQ:	$(\epsilon, PQ0, \epsilon) \rightarrow (\epsilon, PQ1, \epsilon)$ $(\epsilon, PQ0, \#\alpha) \rightarrow (\epsilon, PQ1, \alpha)$ $(\gamma\pi, PQ1, \epsilon) \rightarrow (\gamma, \pi, \epsilon)$	$[(\epsilon, PZ, \epsilon) \rightarrow (\epsilon, y, \epsilon)]$ $?[(\epsilon, z, \epsilon) \rightarrow (\epsilon, PC2, \epsilon)]$ $\{ - [(\epsilon, t, \epsilon) \rightarrow (\epsilon, z, \beta)] \mid \forall \beta \in \Sigma \}$ $+ [(\epsilon, t, \epsilon) \rightarrow (\epsilon, PC2, \epsilon)] \}$
PK:	$(\epsilon, PK0, \alpha) \rightarrow (\epsilon, PK1, \#\alpha)$ $(\epsilon, PK1, \epsilon) \rightarrow (\epsilon, PK2, \epsilon)$ $(\gamma\pi, PK2, \epsilon) \rightarrow (\gamma, \pi, \epsilon)$	$C() = \{g1^*, g2^*, g21^*, g22^*, g3^*, g4^*, g5^*, g6^*, g61^*, g62^*, g7^*, g8^*, x, y, z, w, t:$
A:	$(\epsilon, A0, \epsilon) \rightarrow (\epsilon, A1, \epsilon)$ $(\gamma\pi, A1, \epsilon) \rightarrow (\gamma, \pi, \epsilon)$	$- [(\epsilon, PZ, \epsilon) \rightarrow (\epsilon, x, \epsilon)]$ $- [(\epsilon, PO, \epsilon) \rightarrow (\epsilon, y, \epsilon)]$ $- [(\epsilon, z, \epsilon) \rightarrow (\epsilon, ZI, \epsilon)]$ $+ [(\epsilon, z, \epsilon) \rightarrow (\epsilon, ZF, \epsilon), \mathcal{G}()]$ $+ [(\epsilon, PZ, \epsilon) \rightarrow (\epsilon, z, \epsilon)]$ $+ [(\epsilon, PO, \epsilon) \rightarrow (\epsilon, z, \epsilon)]$ $- [(\epsilon, ZV, \epsilon) \rightarrow (\epsilon, ZF, \epsilon)]$ $+ [(\epsilon, ZV, \epsilon) \rightarrow (\epsilon, g1, \epsilon), \mathcal{D}(0)]$ $+ [(\gamma, g1, \epsilon) \rightarrow (\gamma g2, PK0, \epsilon)]$ $+ (\epsilon, g2, \epsilon) \rightarrow (\epsilon, g21, \epsilon), I(PK1,$
PC:	$(\epsilon, PC0, \alpha) \rightarrow (\epsilon, PC1, \#\alpha)$ $(\epsilon, PC1, \epsilon) \rightarrow (\epsilon, PCF, \epsilon)$ $(\gamma\pi, PC2, \epsilon) \rightarrow (\gamma, \pi, \epsilon)$	$+ (\epsilon, g21, \epsilon) \rightarrow (\epsilon, g22, \epsilon), J()$ $+ [(\gamma, g22, \epsilon) \rightarrow (\gamma g3, PQ0, \epsilon)]$ $+ [(\epsilon, g3, \epsilon) \rightarrow (\epsilon, g4, \epsilon), \mathcal{H}(0)]$ $+ [(\epsilon, g4, \epsilon) \rightarrow (\epsilon, g5, \epsilon), \mathcal{D}(1)]$ $+ [(\gamma, g5, \epsilon) \rightarrow (\gamma g6, PK0, \epsilon)]$ $+ (\epsilon, g6, \epsilon) \rightarrow (\epsilon, g61, \epsilon), I(PK1,$
As funções adaptativas $\mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}$ e $\mathcal{H}, I$ e J são declaradas da seguinte forma:		
$\mathcal{B}(\sigma) = \{g1^*, x, y, z:$	$- [(\epsilon, x, \epsilon) \rightarrow (\epsilon, ZI, \epsilon)]$ $+ [(\epsilon, x, \sigma\alpha) \rightarrow (\epsilon, g1, \alpha)]$ $+ [(\epsilon, g1, \epsilon) \rightarrow (\epsilon, ZI, \epsilon)]$ $- [(\epsilon, PZ, \epsilon) \rightarrow (\epsilon, y, \epsilon)]$ $- [(\epsilon, PO, \epsilon) \rightarrow (\epsilon, z, \epsilon)]$ $+ [(\epsilon, PZ, \epsilon) \rightarrow (\epsilon, x, \epsilon)]$ $+ [(\epsilon, PO, \epsilon) \rightarrow (\epsilon, x, \epsilon)]$ $- [(\epsilon, ZI, \epsilon) \rightarrow (\epsilon, ZK, \epsilon)]$ $\{ - [(\epsilon, ZJ, \epsilon) \rightarrow (\epsilon, ZK, \epsilon), \mathcal{D}(\theta)] \mid \forall \theta \in \Sigma \}$	PC1) $+ (\epsilon, g21, \epsilon) \rightarrow (\epsilon, g22, \epsilon), J()$ $+ [(\gamma, g22, \epsilon) \rightarrow (\gamma g3, PQ0, \epsilon)]$ $+ [(\epsilon, g3, \epsilon) \rightarrow (\epsilon, g4, \epsilon), \mathcal{H}(0)]$ $+ [(\epsilon, g4, \epsilon) \rightarrow (\epsilon, g5, \epsilon), \mathcal{D}(1)]$ $+ [(\gamma, g5, \epsilon) \rightarrow (\gamma g6, PK0, \epsilon)]$ $+ (\epsilon, g6, \epsilon) \rightarrow (\epsilon, g61, \epsilon), I(PK1,$
$\forall \rho \in \Sigma - \{\sigma\}$	$\{ + [(\epsilon, ZJ, \epsilon) \rightarrow (\epsilon, ZK, \epsilon), \mathcal{D}(\rho)]$ $- [(\epsilon, ZT, \epsilon) \rightarrow (\epsilon, ZW, \epsilon)]$ $\{ - [(\epsilon, ZT, \epsilon) \rightarrow (\epsilon, ZW, \epsilon), \mathcal{E}(\theta)] \mid \forall \theta \in \Sigma \}$ $\{ + [(\epsilon, ZT, \epsilon) \rightarrow (\epsilon, ZW, \epsilon), \mathcal{E}(\rho)]$	$+ (\epsilon, g61, \epsilon) \rightarrow (\epsilon, g62, \epsilon), J()$ $+ [(\gamma, g62, \epsilon) \rightarrow (\gamma g7, PQ0, \epsilon)]$ $+ [(\epsilon, g7, \epsilon) \rightarrow (\epsilon, g8, \epsilon), \mathcal{H}(1)]$ $+ [(\epsilon, g8, \epsilon) \rightarrow (\epsilon, ZF, \epsilon)]$ $?[(\epsilon, z, 0\alpha) \rightarrow (\epsilon, w, \alpha)]$ $+ [(\epsilon, ZF, 0\alpha) \rightarrow (\epsilon, w, \alpha)]$ $?[(\epsilon, z, 1\alpha) \rightarrow (\epsilon, t, \alpha)]$ $+ [(\epsilon, ZF, 1\alpha) \rightarrow (\epsilon, t, \alpha)] \}$
$\forall \rho \in \Sigma - \{\sigma\} \}$	$\mathcal{D}(\sigma) = \{g1^*, g2^*, g3^*, x:$	$\mathcal{H}(\sigma) = \{g1^*, g2^*, x, y, z, w:$
$\forall \rho \in \Sigma - \{\sigma\}$	$- [(\epsilon, PQ0, \epsilon) \rightarrow (\epsilon, x, \epsilon)]$ $+ [(\epsilon, PQ0, \epsilon) \rightarrow (\epsilon, g1, \epsilon)]$ $+ [(\epsilon, g1, \sigma\alpha) \rightarrow (\epsilon, x, \alpha)]$ $+ [(\epsilon, g1, \#\alpha) \rightarrow (\epsilon, PQ1, \alpha)]$ $\{ + [(\epsilon, g1, \rho\alpha) \rightarrow (\epsilon, g2, \alpha), \mathcal{F}()] \mid \forall \rho \in \Sigma - \{\sigma\} \}$	$- [(\epsilon, PQ0, \epsilon) \rightarrow (\epsilon, x, \epsilon)]$ $?[(\epsilon, x, \sigma\alpha) \rightarrow (\epsilon, y, \alpha)]$ $+ [(\epsilon, PQ0, \epsilon) \rightarrow (\epsilon, y, \epsilon)]$ $- [(\epsilon, PZ, \epsilon) \rightarrow (\epsilon, z, \epsilon)]$ $?[(\epsilon, PO, \epsilon) \rightarrow (\epsilon, w, \epsilon)]$ $+ [(\epsilon, w, \sigma\alpha) \rightarrow (\epsilon, z, \alpha)]$ $+ [(\epsilon, PZ, \epsilon) \rightarrow (\epsilon, w, \epsilon)] \}$
$\mathcal{E}(\sigma) = \{g1^*, g2^*, x, y, z, w, t, u:$	$\{ + [(\epsilon, g2, \beta\alpha) \rightarrow (\epsilon, g2, \alpha)] \mid \forall \beta \in \Sigma \}$ $+ [(\epsilon, g2, \#\alpha) \rightarrow (\epsilon, g3, \alpha)]$ $+ [(\gamma, g3, \epsilon) \rightarrow (\gamma PQ0, PC0, \epsilon)] \}$	$\mathcal{G}() = \{g1^*, x:$
$\{\sigma\}$	$- [(\epsilon, PQ0, \epsilon) \rightarrow (\epsilon, x, \epsilon)]$ $?[(\epsilon, x, \sigma\alpha) \rightarrow (\epsilon, y, \alpha)]$ $?[(\epsilon, PZ, \epsilon) \rightarrow (\epsilon, z, \epsilon)]$ $?[(\epsilon, PO, \epsilon) \rightarrow (\epsilon, w, \epsilon)]$ $+ [(\epsilon, w, \sigma\alpha) \rightarrow (\epsilon, z, \alpha)]$ $- [(\epsilon, t, \epsilon) \rightarrow (\epsilon, PK2, \epsilon)]$ $\{ + [(\epsilon, t, \alpha) \rightarrow (\epsilon, g1, \rho\alpha)] \mid \forall \rho \in \Sigma - \{\sigma\} \}$	$- [(\epsilon, x, \epsilon) \rightarrow (\epsilon, A1, \epsilon)]$ $+ [(\epsilon, x, +) \rightarrow (\epsilon, g1, \epsilon)]$ $+ [(\epsilon, g1, +) \rightarrow (\epsilon, A1, \epsilon)]$ $\}$
$\{\sigma\}$	$+ [(\epsilon, g1, \epsilon) \rightarrow (\epsilon, PK2, \epsilon)]$ $+ [(\epsilon, PQ0, \epsilon) \rightarrow (\epsilon, g2, \epsilon)]$ $\{ + [(\epsilon, g2, \rho\alpha) \rightarrow (\epsilon, y, \alpha)] \mid \forall \rho \in \Sigma - \{\sigma\} \}$	$I(\sigma1, \sigma2) = \{g1^*, x, y, z:$
$\forall \rho \in \Sigma - \{\sigma\}$	$+ [(\epsilon, g2, \#\alpha) \rightarrow (\epsilon, PQ1, \alpha)]$ $?[(\epsilon, x, \rho\alpha) \rightarrow (\epsilon, u, \alpha), \mathcal{F}()]$	$- [(\epsilon, \sigma2, \alpha) \rightarrow (\epsilon, y, \phi\alpha)] \mid \forall \phi \in \Sigma$ $- [(\epsilon, \sigma2, \epsilon) \rightarrow (\epsilon, y, \epsilon)]$ $+ [(\epsilon, \sigma2, \epsilon) \rightarrow (\epsilon, PCF, \epsilon)]$ $\{ ? [(\epsilon, \sigma1, \epsilon) \rightarrow (\epsilon, x, \beta)] \mid \forall \beta \in \Sigma \}$ $+ [(\epsilon, \sigma2, \epsilon) \rightarrow (\epsilon, g1, \beta)]$ $?[(\epsilon, z, \epsilon) \rightarrow (\epsilon, g1, \beta)]$ $- [(\epsilon, z, \epsilon) \rightarrow (\epsilon, PCF, \epsilon)]$ $+ [(\epsilon, g1, \epsilon) \rightarrow (\epsilon, PCF, \epsilon)]$ $I(x, g1) \}$
$\mathcal{F}() = \{x, y, z, t:$	$+ [(\epsilon, g2, \sigma\alpha) \rightarrow (\epsilon, u, \alpha), \mathcal{F}() \}$ $- [(\epsilon, PZ, \epsilon) \rightarrow (\epsilon, x, \epsilon)]$ $\{ ? [(\epsilon, y, \sigma\alpha) \rightarrow (\epsilon, x, \alpha)] \mid \forall \sigma \in \Sigma \}$	$J() = \{x, y:$ $?[(\epsilon, x, \epsilon) \rightarrow (\epsilon, PCF, \epsilon)]$ $\{ ? [(\epsilon, y, \epsilon) \rightarrow (\epsilon, x, \beta)] \mid \forall \beta \in \Sigma \cup \{\#\} \}$ $- [(\epsilon, x, \epsilon) \rightarrow (\epsilon, PCF, \epsilon)]$ $+ [(\epsilon, x, \epsilon) \rightarrow (\epsilon, PC2, \epsilon)] \}$

Esta solução foi criada para um alfabeto de apenas dois

símbolos. No entanto, é possível ampliar esta solução para qualquer alfabeto. Para cada símbolo do alfabeto seria necessário incluir uma função adaptativa para incluir uma transição para os demais símbolos. Portanto, quanto maior o tamanho do alfabeto, mais funções adaptativas são necessárias.

Após todo o padrão ter sido varrido e o autômato ter sido criado, o símbolo $\vdash$ é encontrado. Neste caso, uma função adaptativa é executada para remover as transições com a função que criava os estados e acrescentava as transições. Além disso, a função insere uma ação adaptativa na transição que chega ao estado final. Esta ação inserida será executada na etapa da busca para contar o número de ocorrências do padrão procurado no texto.

Uma vez que o autômato foi construído, começa a etapa de busca no texto. A Figura 2 ilustra o uso do autômato construído para efetuar a busca em um determinado texto T. A busca inicia no primeiro símbolo do texto e vai consumindo cada símbolo até atingir o final. O autômato muda de estado de acordo com cada símbolo lido no texto. Cada vez que o autômato alcança o estado final significa que o padrão foi encontrado no texto.

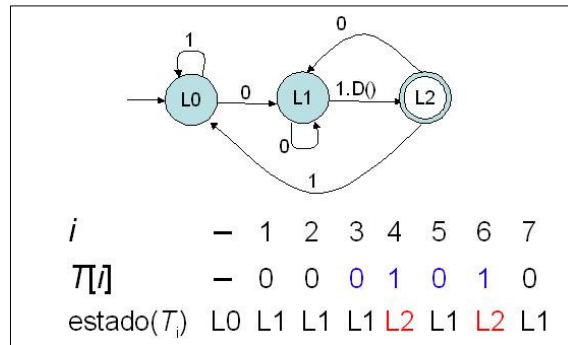


Fig. 2. Exemplo de busca em texto.

Para saber o deslocamento do padrão no texto, basta subtrair o valor da posição do texto em que o padrão terminou pelo tamanho do padrão. A função adaptativa que é executada cada vez que um padrão é encontrado no texto, cria um autômato que representa a quantidade de vezes que o padrão ocorreu no texto.

IV. TRABALHOS FUTUROS

Apesar do uso de autômatos adaptativos para o problema de emparelhamento de cadeias ter sido proposto neste artigo, esta solução ainda não foi implementada completamente. A versão implementada atualmente já conta com duas funções adaptativas: uma para a criação da sub-máquina principal e outra para a criação da máquina que representa a quantidade de ocorrências do padrão no texto.

Desta forma, pretende-se implementar completamente a solução para o problema de emparelhamento de cadeias proposta neste artigo.

Além do modelo apresentado, é possível criar outras funções adaptativas para a fase de busca. Esta é uma sugestão para trabalhos futuros.

Uma outra sugestão, é a criação de outras funções adaptativas que reaproveitem os estados e transições de um autômato para a busca de novos padrões.

Também faz-se necessário um estudo sobre a complexidade de tempo e espaço de um modelo baseado em autômatos adaptativos para o problema de emparelhamento de cadeias, bem como uma comparação com outras técnicas existentes.

V. CONCLUSÕES

A tecnologia adaptativa pode ser muito útil para a representação de diversos problemas. Isto porque, a capacidade de auto-modificação em sua estrutura, permite agregar conceitos desejáveis em um modelo, como por exemplo, a aprendizagem.

O modelo apresentado neste artigo é mais uma solução para o problema de emparelhamento de cadeias. Este trabalho contribuiu com mais uma aplicação dos autômatos adaptativos para a solução de problemas computacionais.

REFERÊNCIAS

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Algoritmos: Teoria e Prática*, Tradução da 2ª edição americana, Editora Campus, 2002.
- [2] J. J. Neto, *Solving Problems Efficiently with Adaptive Automata*, CIAA 2000 – Fifth International Conference on Implementation and Application of Automata, July 2000 – London, Ontario, Canada.
- [3] J. J. Neto, Adaptive automata for context-dependent languages, ACM SIGPLAN Notices, Vol 29, n. 9, pp 15 – 24, 1994.
- [4] 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, 2002, pp. 234-250.
- [5] R. L. A. Rocha e J. J. Neto, *Autômato adaptativo, limites e complexidade em comparação com máquina de Turing*, In: Proceedings of the Second Congress of Logic Applied to Technology – LAPTEC'2000, São Paulo: Faculdade SENAC de Ciências Exatas e Tecnologia, pp. 33-48, 2001.
- [6] J. J. Neto e C. A. B. Pariente, *Adaptive Automata – a reduced complexity proposal*, In: Seventh International Conference on Implementation and Application of Automata, CIAA 2002, Tours, Pre-proceedings Seventh International Conference on Implementation and Application of Automata, CIAA 2002, Tours – France: University of Tours, 2002, pp. 161 – 170.

Elisângela Rodrigues nasceu em Pelotas/RS, Brasil, em 18 de abril de 1978. Graduiu-se em Informática pela Universidade Federal de Pelotas em 2000 e obteve o título de Mestre em Informática pela Universidade Federal de Campina Grande em 2002. Iniciou suas atividades docentes em 2002, na Faculdade de Tecnologia e Ciências de Feira de Santana/BA, onde atuou até 2005. Neste mesmo ano, transferiu-se para a Faculdade ÁREA1 em Salvador/BA, onde atuou até 2007. Em 2006, participou como pesquisadora bolsista (com bolsa da Petrobrás) do Projeto Sistema de Gerenciamento de Poços Automatizados (SGPA), na Universidade Federal da Bahia. Atualmente, é aluna regular do Programa de Pós-graduação em Engenharia Elétrica da Escola Politécnica da Universidade de São Paulo, em regime integral.

Fabrizio Rodrigues nasceu em Mossoró/RN, Brasil, em 09 de agosto de 1973. Graduiu-se em Sistemas de Informação pela Universidade Potiguar em 2000 e obteve o título de Mestre em Informática pela Universidade Federal de Campina Grande em 2002. Iniciou suas atividades docentes em 2002, na Faculdade de Tecnologia e Ciências de Feira de Santana/BA, onde atuou até 2005. Neste mesmo ano, transferiu-se para a Faculdade ÁREA1 em Salvador/BA, onde atuou até 2007, além de atuar como Chefe do Núcleo de Computação. Atualmente, é aluno regular do Programa de Pós-graduação em Engenharia Elétrica da Escola Politécnica da Universidade de São Paulo, em regime integral.