Formalização da expansão de submáquinas do autômato de pilha estruturado

P. R. M. Cereda e J. José Neto

***Abstract*—Uma chamada de submáquina tradicional em um autômato de pilha estruturado pode ser substituída por uma expansão desta, tal que uma cópia da submáquina que originalmente seria chamada é incorporada à topologia corrente, com as devidas ligações. Neste caso, a expansão de submáquinas pode ser entendida como uma forma de adaptatividade mais restrita. Este artigo apresenta uma formalização da expansão de submáquinas do autômato de pilha estruturado, através de uma extensão do dispositivo original. A pilha sintática é preservada para chamadas convencionais. Adicionalmente, são apresentadas discussões acerca da semântica operacional.**

***Palavras-chave:*—Autômato de pilha estruturado, sistema de reescrita, expansão de submáquinas, adaptatividade.**

I. Introdução

Macros constituem um mecanismo significativo para representação de artefatos em um determinado nível de abstração, sem a necessidade da exposição excessiva de detalhes ou características particulares, viabilizando estruturas mais convenientes e aderentes às necessidades do usuário. Transformações simbólicas e algorítmicas conferem a tal mecanismo expressividade e poder computacional [1].

É importante destacar que o conceito de macro extrapola sua vertente textual. Em [2], Moraes substitui a chamada de submáquina tradicional em um autômato por uma expansão desta, dispensando o uso da pilha sintática. Em linhas gerais, uma cópia da submáquina que originalmente seria chamada é incorporada à topologia corrente, com as devidas ligações (por exemplo, transições em vazio dos estados de aceitação para o estado de destino da chamada). Neste caso, o conceito de macro pode ser entendido como uma forma de adaptatividade mais restrita [1]. Este artigo apresenta uma formalização da expansão de submáquinas do autômato de pilha estruturado, através de uma extensão do dispositivo original. A pilha sintática é preservada para chamadas convencionais.

A organização deste artigo é a seguinte: a Seção II apresenta uma breve revisão bibliográfica da teoria. A Seção III introduz formalmente um autômato de pilha estruturado com suporte a expansão de submáquinas. Discussões sobre a semântica operacional são contempladas na Seção IV. As considerações finais são apresentadas na Seção V.

II. Conceitos iniciais

Esta seção apresenta os conceitos relevantes à formalização da expansão de submáquinas do autômato de pilha estruturado através de um sistema de reescrita de propósito geral.

Os autores podem ser contatados através dos seguintes endereços de correio eletrônico: `paulo.cereda@usp.br` e `jjneto@usp.br`.

A. Autômato de pilha estruturado

O autômato de pilha estruturado [3], [4] é um tipo de autômato de pilha formado por um conjunto de autômatos, também chamados *submáquinas*, a cada um dos quais cabe a tarefa de efetuar o reconhecimento de uma das diferentes classes de subcadeias que compõem uma cadeia de entrada em análise [5]. Diferentemente do autômato de pilha tradicional, a pilha tem a finalidade exclusiva de armazenar estados de retorno a cada chamada de uma submáquina. As chamadas e retornos consistem em transferir o controle entre uma submáquina e outra; essa transição consiste em utilizar o símbolo de entrada apenas para a tomada de decisão do autômato em relação a qual transição executar, sendo o tal símbolo consumido na transição subsequente [4], [6].

Definição 1 (autômato de pilha estruturado)**.** Um *autômato de pilha estruturado* M é definido como $M = (Q, A, \Sigma, \Gamma, P, Z_0, q_0, F)$, em que Q é um conjunto finito não-vazio de estados, A é um conjunto de submáquinas (introduzidas formalmente na Definição 2), Σ é o alfabeto do autômato, correspondendo ao conjunto finito não-vazio dos símbolos de entrada, Γ é o conjunto finito não-vazio dos símbolos de pilha, a serem armazenados na memória auxiliar do autômato, P é a relação de transição de estados, $q_0 \in Q$ é o estado inicial (da primeira submáquina), Z_0 é o símbolo marcador de pilha vazia, e $F \subseteq Q$ é o conjunto dos estados de aceitação do autômato (da primeira submáquina) [3], [4]. □

Definição 2 (submáquina do autômato de pilha estruturado)**.** Uma *submáquina* $a_i \in A$ é definida como um autômato finito tradicional, da forma $a_i = (Q_i, \Sigma_i, P_i, q_{i,0}, F_i)$, no qual $Q_i \subseteq Q$ é o conjunto de estados de a_i, $\Sigma_i \subseteq \Sigma$ é o conjunto de símbolos de entrada de a_i, $q_{i,0} \in Q_i$ é o estado de entrada da submáquina a_i, $P_i \subseteq P$ é a relação de transição de estados de a_i, e $F_i \subseteq Q_i$ é o conjunto de estados de retorno da submáquina. □

Definição 3 (relação de transição do autômato de pilha estruturado)**.** A *relação de transição* P é definida como $P \subseteq \Gamma \times Q \times \Sigma \times \Gamma \times Q$, na forma $(\gamma g, q, s\alpha) \rightarrow (\gamma g', q', \alpha)$, na qual q, q' são os estados corrente e de destino, respectivamente, s é o símbolo consumido, α é o restante da cadeia de entrada, g é o topo da pilha, g' é o novo topo da pilha, e γ é o restante da pilha. Uma configuração é um elemento de $Q \times \Sigma^* \times \Gamma^*$, e uma relação entre configurações sucessivas $\vdash$ é definida como:

– *Consumo de símbolo:* $(q, \sigma w, uv) \vdash (q', w, xv)$, com $q, q' \in Q$, $u, x \in \Gamma$, $v \in \Gamma^*$, $\sigma \in \Sigma \cup \{\epsilon\}$, $w \in \Sigma^*$, se σ foi consumido pelo autômato, $x = u$, e $(\gamma, q, \sigma\alpha) \rightarrow (\gamma, q', \alpha) \in P$.
– *Chamada de submáquina:* $(q, w, uv) \vdash (q', w, xv)$, com

$q, q' \in Q$, $u \in \Gamma$, $v, x \in \Gamma^*$, $w \in \Sigma^*$, $x = pu$, com chamada da submáquina R, estado inicial q', retorno em p, e $(\gamma, q, \alpha) \to (\gamma p, q', \alpha) \in P$.

– *Retorno de submáquina:* $(q, w, uv) \vdash (q', w, v)$, com $q, q' \in Q$, $u, x \in \Gamma$, $v \in \Gamma^*$, $w \in \Sigma^*$, $u = q'$, com retorno de submáquina para q', e $(\gamma g, q, \alpha) \to (\gamma, g, \alpha) \in P$. $\square$

Definição 4 (linguagem reconhecida por um autômato de pilha estruturado)**.** A *linguagem reconhecida por um autômato de pilha estruturado* M é dada por $L(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (f, \epsilon, Z_0), f \in F\}$. $\square$

Notação 1 (representação gráfica de chamada de submáquina)**.** Uma chamada de submáquina pode ser representada graficamente através de uma transição com linhas duplas, conforme ilustra a Figura 1. Observe que, a partir do estado $q_{i,m}$ da submáquina a_i, a execução é transferida para a submáquina a_j e o endereço referente ao estado de retorno $q_{i,n}$ é inserido no topo da pilha. No exemplo, o estado corrente passa a ser $q_{i,0}$, que é o estado inicial da submáquina a_j. $\square$

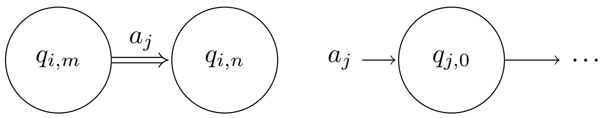


Figura 1. Exemplo de chamada da submáquina a_j.

É importante notar que, por uma questão de organização do modelo, admite-se que $a_i, a_j \in A$, $a_i = (Q_i, \Sigma_i, P_i, q_{i,0}, F_i)$, $a_j = (Q_j, \Sigma_j, P_j, q_{j,0}, F_j)$, $Q_i \cap Q_j = \emptyset$ e $P_i \cap P_j = \emptyset$, isto é, os conjuntos de estados e mapeamentos das submáquinas são disjuntos entre si.

B. Sistema abstrato de redução

Um sistema de reescrita (também chamado *sistema de redução* em sua forma geral) realiza transformações entre termos de acordo com um conjunto de *regras de substituição* (também chamadas *regras de reescrita*) [7]. Aplicações de sistemas de reescrita incluem especificações de tipos abstratos de dados (propriedades de consistência), teoria de computabilidade, decidibilidade de problemas de palavra, prova de teoremas, implementações de linguagens de programação funcionais e dedução automática [1].

Definição 5 (sistema abstrato de redução)**.** Um *sistema abstrato de redução* R é definido como $R = (A, I)$, no qual A é um conjunto de elementos e I é uma sequência de relações binárias $\to_\alpha$ sobre A, também chamadas de relações de redução ou reescrita [8]. Um sistema abstrato de redução com apenas uma relação de redução é chamado *sistema de substituição* [9] ou *sistema de transformação* [10]. Se $a, b \in A$ e $(a, b) \in \to_\alpha$, tal relação de redução pode ser escrita como $a \to_\alpha b$ e b é dita uma α-redução (de um passo) de a. Analogamente, $a \to_\alpha^* b$, sendo $\to_\alpha^*$ o fecho transitivo e reflexivo de $\to_\alpha$, se existe uma sequência finita, potencialmente vazia, de passos de redução $a \equiv a_0 \to_\alpha a_1 \to_\alpha \ldots \to_\alpha a_n \equiv b$, no qual $\equiv$ denota a identidade de elementos de A [11]. $\square$

Definição 6 (forma normal)**.** Um termo $a \in A$ é dito estar em sua *forma normal* quando este não pode ser reescrito, ou seja, $\nexists (a, b) \in I$, $a, b \in A$, tal que $a \to b$. $\square$

A terminação de reescrita de termos (isto é, quando não há mais regras de substituição que possam ser aplicadas na sequência de termos) é, em geral, um problema indecidível [12]. Entretanto, existem estudos no desenvolvimento de condições para garantia de terminação através de ordenações de redução [13], [14], [15], [16], [17]. Um sistema de reescrita é dito *noetheriano* (isto é, possui garantias de terminar a reescrita de termos) se cada termo possui uma forma normal [12].

III. Expansão de submáquinas

Esta seção apresenta a formalização da expansão de submáquinas do autômato de pilha estruturado, através de uma extensão do dispositivo original.

Definição 7 (autômato de pilha estruturado com suporte a expansão)**.** Um *autômato de pilha estruturado com suporte a expansão* M é definido como $M = (Q, A, \Sigma, \Gamma, P, Z_0, q_0, F, \Phi)$, em que $Q \subset H$ é um conjunto finito não-vazio de estados, H é o conjunto enumerável de todos os estados possíveis, A é um conjunto de submáquinas (introduzidas formalmente na Definição 8), Σ é o alfabeto do autômato, correspondendo ao conjunto finito não-vazio dos símbolos de entrada, Γ é o conjunto finito não-vazio dos símbolos de pilha, a serem armazenados na memória auxiliar do autômato, P é a relação de transição de estados, $q_0 \in Q$ é o estado inicial (da primeira submáquina), Z_0 é o símbolo marcador de pilha vazia, $F \subseteq Q$ é o conjunto dos estados de aceitação do autômato (da primeira submáquina), e Φ é o sistema de reescrita associado (introduzido formalmente na Definição 9). $\square$

Definição 8 (submáquina do autômato de pilha estruturado com suporte a expansão)**.** Uma *submáquina* $a_i \in A$ é definida como um autômato finito tradicional, da forma $a_i = (Q_i, \Sigma_i, P_i, q_{i,0}, F_i)$, no qual $Q_i \subseteq Q$ é o conjunto de estados de a_i, $\Sigma_i \subseteq \Sigma$ é o conjunto de símbolos de entrada de a_i, $q_{i,0} \in Q_i$ é o estado de entrada da submáquina a_i, $P_i \subseteq P$ é a relação de transição de estados de a_i, e $F_i \subseteq Q_i$ é o conjunto de estados de retorno da submáquina. $\square$

Definição 9 (sistema de reescrita do autômato de pilha estruturado com suporte a expansão)**.** O *sistema de reescrita* Φ *do autômato de pilha estruturado com suporte a expansão* é definido como $\Phi = (K, \phi_0, H, J)$, em que K é um conjunto enumerável de funções de mapeamento entre identificadores de estados, $K = \{\phi_i \mid \phi_i \colon H \mapsto H\}$, $\phi_0 \in K$ é a função de mapeamento inicial, H é o conjunto enumerável de todos os estados possíveis do autômato de pilha estruturado, e J é uma relação de reescrita $J \colon H \times \Sigma^* \times \Gamma^* \mapsto H \times \Sigma^* \times \Gamma^*$. A cada nova expansão de submáquina, o índice i da função $\phi_i \in K$ é adicionado em uma unidade, $i \in \mathbb{N}$, isto é, a i-ésima expansão está associada à função ϕ_i correspondente. $\square$

Definição 10 (relação de transição do autômato de pilha estruturado com suporte a expansão)**.** A *relação de transição* P é definida como $P \subseteq \Gamma \times H \times \Sigma \times \Gamma \times H$, na forma $(\gamma g, q, s\alpha) \to (\gamma g', q', \alpha)$, na qual q, q' são os estados corrente e de destino, respectivamente, s é o símbolo consumido, α é o restante da cadeia de entrada, g é o topo da pilha, g' é o novo topo da pilha, e γ é o restante da pilha. Uma configuração é um elemento de $H \times \Sigma^* \times \Gamma^*$, e uma relação entre configurações sucessivas $\vdash$ é definida como:

– *Consumo de símbolo:* $(q, \sigma w, uv) \vdash (q', w, xv)$, com $q, q' \in H$, $u, x \in \Gamma$, $v \in \Gamma^*$, $\sigma \in \Sigma \cup \{\epsilon\}$, $w \in \Sigma^*$, se σ foi consumido pelo autômato, $x = u$, e $(\gamma, q, \sigma\alpha) \rightarrow (\gamma, q', \alpha) \in P$.
– *Chamada de submáquina:* $(q, w, uv) \vdash (q', w, xv)$, com $q, q' \in H$, $u \in \Gamma$, $v, x \in \Gamma^*$, $w \in \Sigma^*$, $x = pu$, com chamada da submáquina R, estado inicial q', retorno em p, e $(\gamma, q, \alpha) \rightarrow (\gamma p, q', \alpha) \in P$.
– *Retorno de submáquina:* $(q, w, uv) \vdash (q', w, v)$, com $q, q' \in H, u, x \in \Gamma, v \in \Gamma^*, w \in \Sigma^*, u = q'$, com retorno de submáquina para q', e $(\gamma g, q, \alpha) \rightarrow (\gamma, g, \alpha) \in P$.
– *Expansão de submáquina:* $(q, w, uv) \vdash (q', w, xv)$, com $q, q' \in H$, $u, x \in \Gamma$, $v \in \Gamma^*$, $w \in \Sigma^*$, $x = u$, com expansão da submáquina R, estado inicial r, saída em t, $\phi_i(r) = q'$, e $\forall p_i \in P_R, (\gamma g, e, s\alpha, \gamma g', e', \alpha) \rightarrow (\gamma\phi_i(g), \phi_i(e), s\alpha, \gamma\phi_i(e'), \phi_i(e'), \alpha)$, e $P = P \cup \{(uv, q, w, xv, q', w)\} \cup \{p \mid p \in F_R, (uv, \phi_i(p), w, xv, t, w)\}$, e i é adicionado em uma unidade após a expansão. □

Definição 11 (linguagem reconhecida por um autômato de pilha estruturado com suporte a expansão)**.** A *linguagem reconhecida por um autômato de pilha estruturado com suporte a expansão* M é dada por $L(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (f, \epsilon, Z_0), f \in F\}$. □

Notação 2 (representação gráfica de expansão de submáquina)**.** Uma expansão de submáquina pode ser representada graficamente através de uma transição com linha ondulada, conforme ilustra a Figura 2. Observe que, a partir do estado $q_{i,m}$ da submáquina a_i, uma cópia da submáquina a_j é inserida na topologia corrente, com as devidas transições de entrada e saída. A execução do dispositivo, portanto, prossegue como em um autômato finito tradicional. O índice k da função ϕ_k indica que esta é a k-ésima expansão de submáquina. □

Observe que a representação gráfica de chamada de submáquina do autômato de pilha estruturado apresentada na Notação 1 mantém-se inalterada para o dispositivo estendido apresentado nesta seção.

IV. Semântica operacional

Esta seção contempla e discute alguns aspectos da semântica operacional envolvida na expansão de submáquinas do autômato de pilha estruturado.

A. Protótipo de submáquina

A expansão de submáquinas utiliza o conceito de *protótipos* estabelecidos em tempo de definição do autômato de pilha estruturado. Tal resolução determina que as expansões procedam somente nas definições de submáquinas e não em suas instâncias particulares potencialmente modificáveis em tempo de reconhecimento de cadeia. A Figura 3, a seguir, ilustra o conceito de protótipo de submáquina.

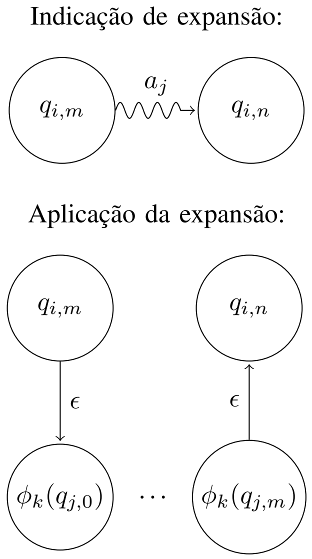


Figura 2. Exemplo de expansão da submáquina a_j.

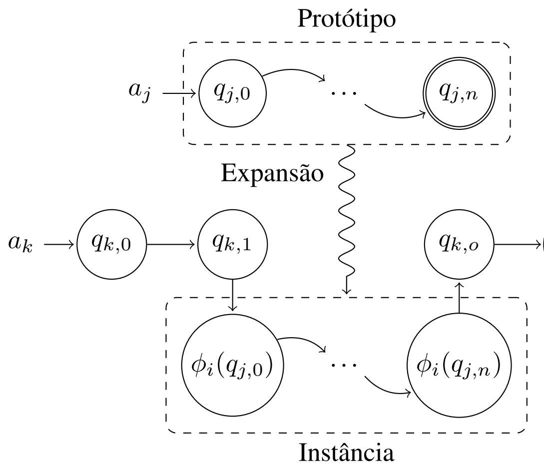


Figura 3. Exemplo de resolução de expansão fundamentada em protótipos de submáquinas. Uma vez expandida, a instância torna-se potencialmente modificável ao longo do tempo, não refletindo em seu protótipo correspondente.

Observe que, de acordo com a Figura 3, o estado de retorno $q_{j,n}$ da submáquina a_j perde tal propriedade quando representado na instância expandida. A razão para tal comportamento deve-se ao fato de que, por definição, o conjunto de estados finais do dispositivo é imutável. De acordo com a Definição 10, os estados de retorno da submáquina recebem transições explícitas em vazio ao estado de destino, tal qual o retorno implícito no formalismo original.

Exemplo 1. Considere um autômato de pilha estruturado M_1 que reconhece cadeias pertencentes à linguagem livre de contexto $L_1 = \{w \in \{a, +, (,)\}^* \mid w$ representa uma soma de termos com suporte a parênteses aninhados$\}$, de acordo com a Figura 4.

O reconhecimento da cadeia $a + a \in L_1(M_1)$ resulta em três expansões de submáquina, incorporando tais instâncias na topologia do autômato de pilha estruturado da Figura 4. A

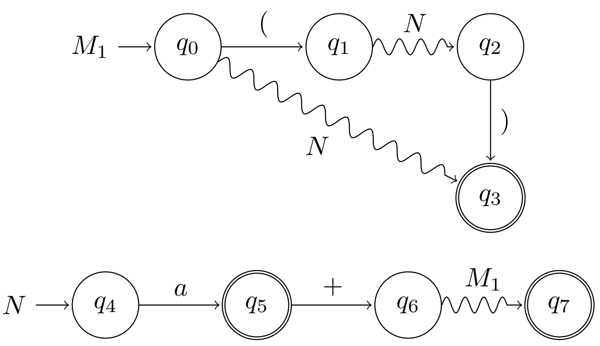


Figura 4. Autômato de pilha estruturado M_1 que reconhece cadeias pertencentes à linguagem livre de contexto $L_1 = \{w \in \{a, +, (,)\}^* \mid w$ representa uma soma de termos com suporte a parênteses aninhados$\}$.

topologia resultante é apresentada na Figura 5.

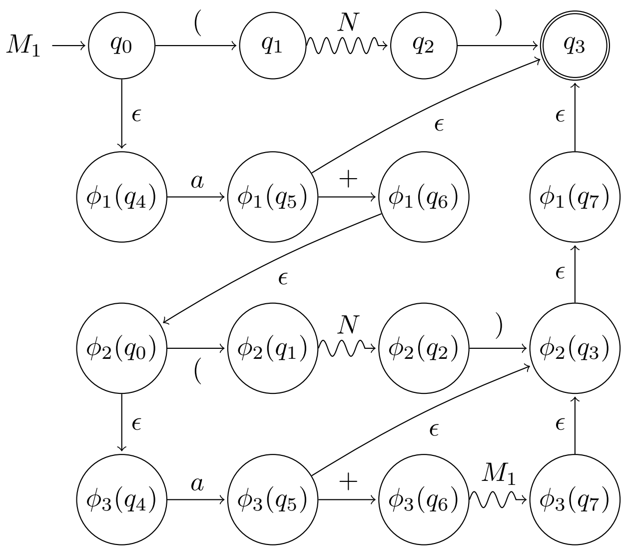


Figura 5. foo

Observe que, de acordo com a Figura 5, a expansão das submáquinas M_1 (nível 2) e N (níveis 1 e 3) procedeu de acordo com suas respectivas definições (protótipos) e não conforme suas instâncias particulares. □

É importante destacar que esta resolução evita que submáquinas recursivas expandam suas instâncias ao invés de seus protótipos. Observe que cada instância particular de submáquina efetua o reconhecimento de uma das diferentes classes de subcadeias que compõem a cadeia de entrada e, portanto, pode não representar uma classe arbitrária em um momento subsequente durante o reconhecimento.

B. *Referência ao ponto de expansão*

Sob a perspectiva teórica, a função de mapeamento ϕ_i referencia novos identificadores de estados para garantir a unicidade dos estados criados durante a i-ésima expansão. Entretanto, operacionalmente, a função ϕ_i pode manter referência explícita ao ponto de expansão da submáquina, conforme ilustra a Equação 1.

$$\phi_i(a_{j,0}) = \langle s, a_{j,0} \rangle \mid (s, w, uv) \vdash (\phi_i(a_{j,0}), w, uv) \qquad (1)$$

De acordo com a Equação 1, uma expansão indexada por i e iniciada no estado s o especifica como prefixo dos identificadores do mapeamento de ϕ_i para a renomeação de estados na submáquina a_i sendo expandida. A Equação 2 apresenta a forma geral, considerando p como prefixo de mapeamento e k como índice da expansão corrente.

$$\phi_k(a_{i,j}) = \langle p, a_{i,j} \rangle \mid (p, w, uv) \vdash (\phi_k(a_{i,j}), w, uv) \qquad (2)$$

É importante observar que identificadores de mapeamento podem conter prefixos aninhados. Tal característica permite o reconhecimento facilitado de trechos expandidos na topologia do autômato, incluindo seus pontos de origem.

Exemplo 2. Considere um autômato de pilha estruturado M_2 que reconhece cadeias pertencentes à linguagem livre de contexto $L_2 = \{w \in \{a, b\}^* \mid w = a^n b^n, n \in \mathbb{N}\}$, de acordo com a Figura 6.

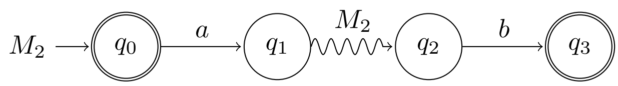


Figura 6. Autômato de pilha estruturado M que reconhece cadeias pertencentes à linguagem livre de contexto $L = \{w \in \{a, b\}^* \mid w = a^n b^n, n \in \mathbb{N}\}$.

O reconhecimento da cadeia $aaabbb \in L_2(M_2)$ resulta em três expansões de submáquina, incorporando tais instâncias na topologia do autômato de pilha estruturado da Figura 6. A topologia resultante é apresentada na Figura 7.

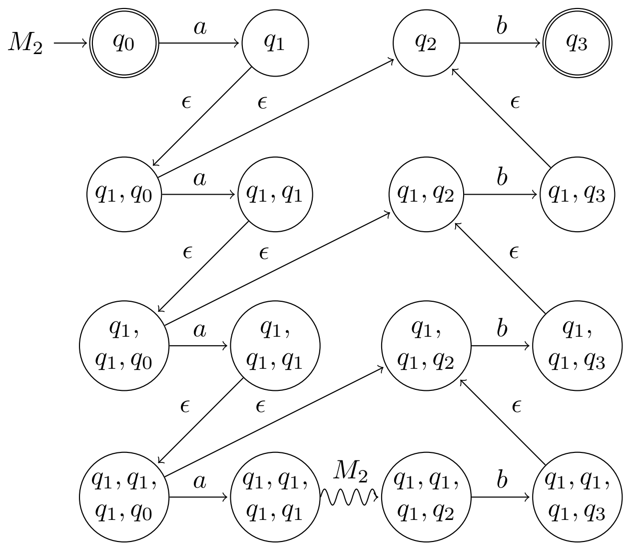


Figura 7. Reconhecimento da cadeia $aaabbb \in L(M)$ pelo autômato de pilha estruturado da Figura 6, resultando em três expansões de submáquina.

Observe que, de acordo com a Figura 7, é possível identificar os pontos de origem de expansão da submáquina M_2 na topologia do autômato e as instâncias propriamente ditas. □

A referência ao ponto de expansão nos identificadores de estados das instâncias de submáquinas oferece conveniências para análise dos trechos expandidos e, consequentemente, do reconhecimento das classes de subcadeias que compõem a cadeia de entrada. É possível, entretanto, usar outras estruturas de representação conforme o domínio de aplicação, desde que a unicidade dos identificadores de estados seja preservada.

V. Considerações finais

Este artigo apresentou uma formalização da expansão de submáquinas do autômato de pilha estruturado através de uma extensão do dispositivo original, como alternativa a desvios de controle entre autômatos mutuamente recursivos para reconhecimento da cadeia de entrada.

É importante observar que chamadas e expansões de submáquinas, de acordo com a relação de transição apresentada na Definição 10, representam operacionalmente os mesmos passos computacionais necessários para reconhecimento das classes particulares de subcadeias componentes da cadeia de entrada em análise. A distinção reside na substituição do desvio de controle convencional entre autômatos mutuamente recursivos, auxiliado por uma pilha sintática para armazenamento exclusivo do estado de retorno, por transições e estados explicitamente incorporados à topologia da submáquina corrente. A extensão apresentada neste artigo mantém a possibilidade de chamadas de submáquinas e preserva a pilha sintática. Por esta perspectiva, o formalismo pode ser simplificado para admitir apenas expansões de submáquinas.

Pragmaticamente, o conceito de expansão de submáquina dispensa o uso da pilha sintática, mas pode aumentar significativamente o espaço de estados e transições do autômato de pilha estruturado. É necessário considerar tais particularidades ao adotar um tratamento específico para submáquinas, conforme o domínio da aplicação. Uma solução híbrida (equilíbrio entre chamada e expansão) pode oferecer vantagens operacionais. Do ponto de vista de implementação, a expansão de submáquina permite que o dispositivo em execução atue como um autômato finito tradicional, sem manipulação de pilha.

A expansão de submáquina pode ser entendida como uma forma de adaptatividade mais restrita, dado que o autômato de pilha estruturado sofre modificações em sua topologia em tempo de reconhecimento de cadeia sem, entretanto, alterar a classe de linguagens que identifica.

Espera-se que tal formalização contribua para a especificação de autômatos com características de automodificação seletiva através da expansão controlada de submáquinas, conforme a conveniência e as características do domínio de aplicação.

Referências

[1] P. R. M. Cereda, “Macros como mecanismos de abstração em transformações textuais,” Tese de Doutorado, Escola Politécnica, Universidade de São Paulo, São Paulo, 2018.

[2] M. Moraes, “Alguns aspectos de tratamento de dependências de contexto em linguagem natural empregando tecnologia adaptativa,” Tese de Doutorado, Escola Politécnica, Universidade de são Paulo, São Paulo, 2006.

[3] J. José Neto and M. E. S. Magalhães, “Reconhecedores sintáticos: Uma alternativa didática para uso em cursos de engenharia,” in *Congresso Nacional de Informática*, 1981, pp. 171–181.

[4] J. José Neto, “Contribuições à metodologia de construção de compiladores,” Tese de Livre docência, Escola Politécnica, Universidade de São Paulo, São Paulo, 1993.

[5] M. V. M. Ramos, J. José Neto, and Í. S. Vega, *Linguagens formais: teoria, modelagem e implementação*. Porto Alegre: Bookman, 2009.

[6] J. José Neto, “Adaptive automata for context-sensitive languages,” *SIGPLAN Notices*, vol. 29, no. 9, pp. 115–124, set 1994.

[7] F. Baader and T. Nipkow, *Term rewriting and all that*. New York: Cambridge University, 1998.

[8] J. W. Klop, “Term rewriting systems,” Tech. Rep., 1992.

[9] J. Staples, “Church-Rosser theorem for replacement systems,” in *Algebra and Logic*, ser. Lecture Notes in Mathematics, J. Crosley, Ed. Springer-Verlag, 1975, no. 450.

[10] M. Jantzen, “Confluent string rewriting and congruences,” in *EATCS (European Association for Theoretical Computer Science) Monographs on Theoretical Computer Science*. Berlin: Springer-Verlag, 1988, vol. 14.

[11] G. Huet and D. C. Oppen, “Equations and rewrite rules: A survey,” Stanford University, Stanford, Technical report, 1980.

[12] M. Hermann, C. Kirchner, and H. Kirchner, “Implementations of term rewriting systems,” *The Computer Journal*, vol. 34, no. 1, pp. 20–33, feb 1991.

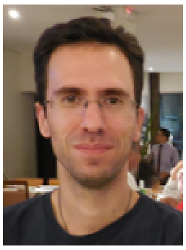
[13] N. Dershowitz, “Orderings for term-rewriting systems,” *Theoretical Computer Science*, no. 17, pp. 279–301, 1982.

[14] ——, “Termination of rewriting,” *Journal of Symbolic Computation*, vol. 3, no. 1–2, pp. 69–116, 1987.

[15] J.-P. Jouannaud, P. Lescanne, and F. Reinig, “Recursive decomposition ordering,” in *Formal Description of Programming Concepts 2*, D. Bjørner, Ed. Garmisch-Partenkirchen, Germany: North Holland, 1982, pp. 331–348.

[16] P. Lescanne, “Uniform termination of term rewriting systems,” in *Colloque les Arbres en Algebre et en Programmation*, B. Courcelle, Ed. Bordeaux: Cambridge University, 1984, pp. 182–194.

[17] G. Huet, “Confluent reductions: Abstract properties and applications to term rewriting systems: Abstract properties and applications to term rewriting systems,” *Journal of the ACM*, vol. 27, no. 4, pp. 797–821, oct 1980.



Paulo Roberto Massa Cereda é graduado em Ciência da Computação pelo Centro Universitário Central Paulista (2005), mestre em Ciência da Computação pela Universidade Federal de São Carlos (2008) e doutor em Engenharia de Computação pela Escola Politécnica da Universidade de São Paulo (2018). Tem experiência na área de Ciência da Computação, com ênfase em Teoria da Computação, atuando principalmente nos seguintes temas: tecnologia adaptativa, autômatos adaptativos, dispositivos adaptativos, sistemas de reescrita, cálculo lambda, complexidade computacional, linguagens de programação e construção de compiladores.

João José Neto é graduado em Engenharia de Eletricidade (1971), mestre em Engenharia Elétrica (1975), doutor em Engenharia Elétrica (1980) e livre-docente (1993) pela Escola Politécnica da Universidade de São Paulo. Atualmente, é professor associado da Escola Politécnica da Universidade de São Paulo e coordena o LTA – Laboratório de Linguagens e Técnicas Adaptativas do PCS – Departamento de Engenharia de Computação e Sistemas Digitais da EPUSP. Tem experiência na área de Ciência da Computação, com ênfase nos Fundamentos da Engenharia da Computação, atuando principalmente nos seguintes temas: dispositivos adaptativos, tecnologia adaptativa, autômatos adaptativos, e em suas aplicações à Engenharia de Computação, particularmente em sistemas de tomada de decisão adaptativa, análise e processamento de linguagens naturais, construção de compiladores, robótica, ensino assistido por computador, modelagem de sistemas inteligentes, processos de aprendizagem automática e inferências baseadas em tecnologia adaptativa.