

X.2.2/13

RECONHECEDORES SINTÁTICOS UMA ALTERNATIVA DIDÁTICA PARA USO EM CURSOS DE ENGENHARIA

Dr. João José Neto e Mario Eduardo S. Magalhães

Escola Politécnica da Universidade de São Paulo - Caixa Postal 11.455 - São Paulo - SP

Palavras-chave: Reconhecedores Sintáticos, Gramáticas, Linguagens Livres de Contexto, Compiladores, Autômatos, Expressões Regulares, Máquinas de Estados Finitos, Linguagens Formais.

Resumo: As linguagens de programação usuais exigem, para seu reconhecimento sintático, técnicas relativamente complexas, embora muito bem conhecidas (1, 2, 5).

Este artigo explora um reconhecedor sintático extremamente simples e potente baseado no antigo modelo utilizado por Conway (3) e formalizado por Lomet (7).

Apresenta-se inicialmente o resumo da teoria em que se baseia o método proposto de reconhecimento, identificando-se a seguir o seu funcionamento com o de reconhecedores elementares (máquinas de estados finitos) auxiliados por pilhas de controle.

Finalmente, um pequeno exemplo de aplicação da técnica é apresentado.

INTRODUÇÃO

Durante sua vida profissional um engenheiro se defronta frequentemente com problemas da área de compilação, que se apresentam de diversas formas, tais como:

- . implantação de processadores de linguagens já prontos.
- . modificação dos mesmos para adaptá-los a aplicações específicas.
- . necessidade de dotar programas de aplicação com linguagens de entrada de fácil compreensão e utilização.
- . manutenção de processadores de linguagens já implantados.
- . construção de programas com características abstratas semelhantes às dos processadores de linguagens.
- . (excepcionalmente) projeto e implementação de processadores de linguagens tradicionais em máquinas novas.

Isto motiva a inclusão do ensino de técnicas de compilação nos cursos de engenharia voltados para computadores.

Devido aos objetivos almejados, este ensino deve ser feito com o cuidado de, embora apresentando as técnicas de maneira rigorosa, evitar excesso de formalismos: deve-se, sempre que viável, escolher métodos que resolvam os problemas do engenheiro da maneira menos complexa possível. (Observe-se que, em vários casos, como no aqui apresentado, apesar de ser o método de aplicação extremamente simples, as justificativas teóricas em que se apoiam não apresentam a mesma simplicidade).

Com esta finalidade em mente, relembra-se neste trabalho os fundamentos teóricos que

suportam uma técnica de construção de reconhecedores sintáticos baseada em máquinas sequenciais, ferramenta de ampla utilização em engenharia (2, 5, 6, 9).

A idéia de utilizar máquinas de estados finitos para o reconhecimento sintático de linguagens é antiga. Já em 1963, Conway apresentava um compilador COBOL utilizando esta técnica (5). Em 1973, surge uma formalização deste modelo, devida a Lomet (9), que utiliza o conceito de autômatos de pilha de terminísticos encadeados.

O modelo aqui apresentado assemelha-se ao proposto por Conway e Lomet, consistindo de um conjunto de submáquinas elementares de estados finitos interligadas através de uma pilha. Qualitativamente, pode-se visualizar o funcionamento do modelo associando-se a cada submáquina a tarefa de reconhecer uma dada classe de formas setenciais, operando uma única submáquina de cada vez. Em cada instante, a submáquina correntemente ativa analisa um carácter de entrada e decide entre três opções:

a) movimentação interna, permanecendo a submáquina em atividade. Este é o caso em que ainda não foi completado o reconhecimento associado à submáquina, e que, além disso, não há necessidade de cooperação de outra submáquina para este reconhecimento. Neste caso, a pilha não é modificada;

b) ativação de outra submáquina, permanecendo a submáquina corrente em estado de espera. É o caso em que, tendo completado o reconhecimento a ela associado, a submáquina necessita da cooperação de outra submáquina para continuar o seu reconhecimento. A pilha é neste caso alterada, guardando informações necessárias ao prosseguimento

do reconhecimento pela submáquina corrente, na ocasião do término da operação da submáquina ativada.

c) término da operação da submáquina. Neste caso, a submáquina corrente acaba de efetuar o reconhecimento a ela associado. O prosseguimento da operação do modelo fica determinado pelo conteúdo da pilha: Se a pilha estiver vazia pode-se afirmar que a entrada até então analisada foi reconhecida pelo modelo, caso contrário, o topo da pilha contém informações suficientes para que a submáquina corrente devolva o controle àquela que a ativou, removendo-se então o conteúdo do topo da pilha.

Pode-se ainda, com facilidade, incluir um quarto tipo de movimento, correspondente à detecção de erros de sintaxe na cadeia de entrada, o qual poderia levar ao acionamento de uma submáquina destinada especificamente à recuperação do erro detetado.

Neste artigo, porém, tais aspectos não serão explorados por não ser do escopo do trabalho. Na máquina apresentada será detetado um erro quando, na situação corrente da máquina, não houver nenhuma especificação de movimento para o carácter de entrada em análise. Nesta situação a máquina interrompe o seu movimento, indicando o não reconhecimento da cadeia de entrada.

Quando aplicado ao reconhecimento de construção sintáticas muito simples, como expressões regulares, é possível, embora não necessário, simplificar bastante o modelo, reduzindo-o, em essência, a um autômato finito, sendo necessária nesta situação uma única submáquina, ocorrendo, portanto, movimentos do primeiro tipo (a) apenas.

No caso mais geral de gramáticas livres de contexto, as construções sintáticas, como será visto adiante, podem ser agrupadas em duas categorias: regulares e "self-embedded". Quando regulares, recai-se no caso anterior, quando "self-embedded" será necessário a aplicação de movimentos dos tipos (b) e (c). Tais movimentos ocorrem necessariamente quando do tratamento de construções que delimitam, respectivamente, à esquerda e à direita uma construção "self-embedded". A esta corresponde uma submáquina que deve ser ativada quando do movimento (b) e desativada quando do movimento (c), podendo este procedimento repetir-se recursivamente. O controle de retorno é, como foi citado acima, gerenciado pela pilha. Observe-se que, exceto nos movimentos em que se opera a pilha, todas as etapas do reconhecimento correspondem à análise de partes regulares da cadeia de entrada. Cumpre notar que, obrigatoriamente, deverá existir pelo menos uma submáquina correspondente a cada construção "self-embedded" da gramática.

Os conceitos citados informalmente no texto acima, serão, quando necessários, considerados de maneira mais precisa no resumo teórico a seguir.

RESUMO TEÓRICO*

No estudo de reconhecedores sintáticos pode-se constatar a importância dos conceitos de linguagens e suas classes.

Segundo a hierarquia de Chomsky as linguagens se dividem em quatro classes: tipos 0, 1, 2 ou 3 (8, 10). No âmbito deste artigo considerar-se-á somente linguagens dos tipos 2 e 3, que compõem as classes de linguagens livres de contexto e regulares respectivamente. Em engenharia e em inúmeros outros ambientes profissionais, as linguagens utilizadas, embora não sejam estruturalmente tão simples, são comodamente representadas através de versões simplificadas livres de contexto e/ou regulares. A inclusão de dependências de contexto costuma ser considerada, neste caso, fora do dispositivo de reconhecimento sintático da linguagem, simplificando assim tais reconhecimentos.

Pode-se representar as linguagens através de dois mecanismos básicos:

- dispositivos geradores ou gramáticas
- dispositivos reconhecedores

As classes de linguagem surgem quando da imposição de restrições sobre o funcionamento de tais dispositivos.

Dispositivos Geradores

Um *dispositivo gerador* ou *gramática* é definido como uma quádrupla:

$$G = (V, \Sigma, P, S)$$

onde:

V é o conjunto finito não vazio dito *Vocabulário*.

$\Sigma \subset V$ é o conjunto não vazio de *Terminais*.

$S \in V - \Sigma = N$ é a raiz da gramática, onde N é o conjunto de *não terminais*.

P é o conjunto finito não vazio de *Produção* do tipo:

$$\alpha \rightarrow \beta;$$

se $\alpha \in V^* N V^* \beta \in V^*$ tem-se uma *gramática de estrutura de frase geral*.

Definições - Diz-se que, para $\alpha', \beta' \in V^*$, α' *deriva diretamente* β' (denotado por $\alpha' \Rightarrow \beta'$) se existir $\alpha_1, \alpha_2, \beta \in V^*$ e $\alpha \in N$ tais que:

$$\begin{aligned} \alpha' &= \alpha_1 \alpha \alpha_2 \\ \beta' &= \alpha_1 \beta \alpha_2 \end{aligned}$$

$$\text{com } \alpha \rightarrow \beta \in P$$

- Diz-se que $\alpha' \in \Sigma^*$ *pertence à linguagem gerada* por G (ou que α' é uma *sentença gerada* por G) se existir uma cadeia de derivações diretas com início na raiz da gramática e levando o α' (denotado por $S \Rightarrow^* \alpha'$).

* Por motivos de concisão, quando não prejudicar a clareza do texto, todos os símbolos, uma vez introduzidos, manterão seus significados exceto quando ocorrer citação explícita em contrário, evitando-se assim repetição desnecessária de definições e relações de pertinência.

— Denomina-se *linguagem gerada* pela gramática G ao conjunto de todas as sentenças geradas por esta gramática.

Se a uma gramática de estrutura de frase forem impostas as restrições:

$\alpha \in N$ e $\beta \in V^*$ obter-se-á uma *gramática livre de contexto* usualmente descrita em notação BNF (1, 3, 4, 7).

Como se sabe (8, 10) gramáticas livres de contexto geram linguagens que, na classificação de Chomsky, pertencem ao *tipo 2*.

Se, por outro lado, todas as produções tiverem uma das formas:

$$\begin{aligned}\alpha &\rightarrow a & \text{com } \alpha \in N, a \in \Sigma^* \\ \alpha &\rightarrow a\beta & \text{com } \alpha, \beta \in N, a \in \Sigma^*\end{aligned}$$

tem-se uma *gramática linear à direita*. Define-se de modo análogo gramáticas *lineares à esquerda* (substituindo-se a segunda forma das produções por $\alpha \rightarrow \beta a$).

Observe-se que obviamente estas gramáticas estão incluídas entre as livres de contexto.

É sabido (8, 10) que tanto as gramáticas lineares à direita como as lineares à esquerda geram o mesmo conjunto de linguagens (*tipo 3* segundo a hierarquia de Chomsky).

— Diz-se que uma gramática $G = (V, \Sigma, P, S)$ é "self-embedded" se e somente se existir algum não terminal $\alpha \in N$ tal que:

$$\alpha \Rightarrow^* \beta \alpha \gamma, \text{ com } \beta, \gamma \in V^+$$

Obviamente uma gramática "self-embedded" nunca será do tipo 3.

— Diz-se que uma linguagem L , de tipo 2, é "self-embedded" se e somente se todas as gramáticas livres de contexto que a geram forem "self-embedded".

É resultado bastante conhecido (8, 10) ser condição necessária e suficiente para se afirmar de tipo 3 uma dada linguagem, que esta seja de tipo 2 e não "self-embedded".

DISPOSITIVOS RECONHECEDORES

A outra maneira clássica utilizada na representação de linguagens é o emprego de dispositivos reconhecedores. De modo análogo ao que foi visto para dispositivos geradores, aos dispositivos reconhecedores podem ser aplicadas restrições gradativas que os levem a reconhecer os diferentes tipos de linguagens. Assim, conta-se para linguagens de estrutura de frases gerais, com os dispositivos reconhecedores denominados "máquinas de Turing", cujo formalismo será omitido por não pertencer ao escopo desta publicação.

Para as gramáticas livres de contexto, os dispositivos reconhecedores correspondentes são os denominados *autômatos de pilha*, definidos como sendo sétuplas:

$$\hat{A} = (\hat{Q}, \hat{e}, \hat{\Gamma}, \hat{P}, \hat{q}_0, \hat{Z}_0, \hat{F})$$

onde:

$\hat{Q}$ é um conjunto finito não vazio de *estados*.

$\hat{\Sigma}$ é um conjunto finito não vazio de *terminais*.
 $\hat{\Gamma}$ é um conjunto não vazio de *símbolos de pilha*.
 $\hat{q}_0 \in \hat{Q}$ é denominado *estado inicial* de $\hat{A}$.
 $\hat{Z}_0 \in \hat{\Gamma}$ é o *símbolo inicial da pilha*.
 $\hat{F} \subseteq \hat{Q}$ é o conjunto de *estados finais* de $\hat{A}$.
 $\hat{P}$ é um conjunto finito não vazio de produções, que são expressões com a seguinte forma:

$$\hat{\gamma} \hat{Z} \hat{q}_i \hat{\beta} \rightarrow \hat{\gamma} \hat{\delta} \hat{q}_j \hat{\alpha}$$

onde:

$\hat{\gamma} \in \hat{\Gamma}^*$; representa o conteúdo não visível da pilha.

$\hat{Z} \in \hat{\Gamma}$; é o antigo conteúdo do topo da pilha.

$\hat{\delta} \in \hat{\Gamma}^*$; representa a alteração da pilha.

$\hat{q}_i, \hat{q}_j \in \hat{Q}$; são estados do autômato $\hat{A}$.

$\hat{\beta} \in \hat{\Sigma}^*$; representa a cadeia de entrada antes da aplicação da produção.

$\hat{\alpha} \in \hat{\Sigma}^*$; representa a cadeia de entrada após a aplicação da produção.

$\hat{\beta} \in [\hat{\delta} \hat{\alpha}, \hat{\alpha}]$ onde

$\hat{\delta} \in \hat{\Sigma}$; é o carácter de entrada consumido pela aplicação da produção.

— Define-se *situação i do autômato de pilha $\hat{A}$* como sendo a tripla:

$$i = (\hat{\gamma}, \hat{q}, \hat{\alpha}\phi), \text{ onde:}$$

$\hat{\gamma} \in \hat{\Gamma}^*$ é o conteúdo da pilha;

$\hat{q} \in \hat{Q}$ é o estado do autômato da pilha;

$\hat{\alpha} \in \hat{\Sigma}^*$ é a cadeia de terminais apresentada ao autômato de pilha, para reconhecimento, nesta situação;

$\phi \notin \hat{\Sigma}$ é um símbolo que indica fim de cadeia.

Para o caso particular:

$$i_0 = (\hat{Z}_0, \hat{q}_0, \hat{\alpha}_0 \phi), \text{ em que}$$

$\hat{\alpha}_0 \in \hat{\Sigma}^*$ é a cadeia de terminais a ser reconhecida por $\hat{A}$.

i_0 é denominada *situação inicial* do autômato de pilha $\hat{A}$.

— Define-se *transição do autômato de pilha $\hat{A}$* como sendo uma aplicação qualquer, sobre sua situação:

$$i_1 = (\hat{\gamma}_u, \hat{q}_u, \hat{\alpha}_u \phi)$$

de uma produção $\hat{p} \in \hat{P}$, da forma:

$$\hat{\gamma}_u \hat{q}_u \hat{\alpha}_u \rightarrow \hat{\gamma}_v \hat{q}_v \hat{\alpha}_v$$

onde:

$$\hat{\gamma}_k \in \hat{\Gamma}^*;$$

$$\hat{q}_k \in \hat{Q};$$

$$\hat{\alpha}_k \in \hat{\Sigma}^*; \text{ com } k = u, v$$

levando o autômato de pilha $\hat{A}$ à nova situação:

$$\hat{t}_{i+1} = (\hat{\gamma}_v, \hat{q}_v, \hat{\alpha}_v \phi)$$

Denota-se esta transição por: $\hat{t}_i \Rightarrow \hat{t}_{i+1}$

ou seja, $(\hat{\gamma}_u, \hat{q}_u, \hat{\alpha}_u \phi) \Rightarrow (\hat{\gamma}_v, \hat{q}_v, \hat{\alpha}_v \phi)$

- Diz-se que um autômato de pilha $\hat{A}$ reconhece $\hat{\alpha} \in \hat{\Sigma}^*$ por esgotamento da cadeia de entrada e esvaziamento da pilha se e somente se existir uma sequência de transições:

$$\hat{t}_0 \Rightarrow \hat{t}_1 \Rightarrow \hat{t}_2 \Rightarrow \dots \Rightarrow \hat{t}_{n-1} \Rightarrow \hat{t}_n$$

ou simplesmente $\hat{t}_0 \Rightarrow^* \hat{t}_n$

onde $\hat{t}_k = (\hat{\gamma}_k, \hat{q}_k, \hat{\alpha}_k \phi)$, $\hat{\gamma}_k \in \hat{\Gamma}^*$; $\hat{q}_k \in \hat{Q}$;

$$\hat{\alpha}_k \in \hat{\Sigma}^*, 0 < k < n$$

$$\hat{t}_n = (\hat{Z}_0, \hat{q}_F, \phi), \text{ com } \hat{q}_F \in \hat{F};$$

Nestas condições diz-se que $\hat{t}_n$ é uma situação final do autômato de pilha $\hat{A}$ para a cadeia de entrada $\hat{\alpha}_0$.

Observe-se que existem outras definições de reconhecimento de cadeias por autômatos de pilha: reconhecimento por esgotamento da cadeia de entrada apenas e reconhecimento por esvaziamento da pilha apenas. Demonstra-se (8) que, independentemente do critério escolhido, não se altera o conjunto de linguagens reconhecíveis pelo autômato de pilha $\hat{A}$.

Do mesmo modo, para as gramáticas lineares à direita (esquerda), isto é, Chomsky tipo 3, os dispositivos reconhecedores correspondentes são denominados autômatos finitos. Para facilidade de comparação com os autômatos de pilha, aqueles serão descritos a seguir através de notação similar.

Um autômato finito pode ser definido através de uma quintupla:

$$\bar{A} = (\bar{Q}, \bar{\Sigma}, \bar{P}, \bar{q}_0, \bar{F})$$

onde: $\bar{P}$ é um conjunto finito não vazio de produções, que são expressões com o seguinte formato:

$$\bar{q}_i \bar{\beta} \rightarrow \bar{q}_j \bar{\alpha}$$

Definições - Denomina-se situação $\bar{t}$ do autômato finito $\bar{A}$ como sendo a dupla:

$$\bar{t} = (\bar{q}, \bar{\alpha} \phi) \text{ onde:}$$

$\bar{q} \in \bar{Q}$ é o estado do autômato finito;

$\bar{\alpha} \in \bar{\Sigma}^*$ é a cadeia de terminais que está sendo apresentada ao autômato finito, para reconhecimento, nesta situação;

$\phi \notin \bar{\Sigma}$ é um símbolo que indica fim de cadeia

Para o caso particular:

$$\bar{t}_0 = (\bar{q}_0, \bar{\alpha}_0 \phi) \text{ em que:}$$

$\bar{\alpha}_0 \in \bar{\Sigma}^*$ é a cadeia de terminais a ser reconhecido por $\bar{A}$

$\bar{t}_0$ é denominada situação inicial do autômato finito $\bar{A}$

- Define-se transição do autômato finito $\bar{A}$ como sendo uma aplicação qualquer, sobre a situação:

$$\bar{t}_i = (\bar{q}_u, \bar{\alpha}_u \phi),$$

de uma produção $\bar{p} \in \bar{P}$ da forma:

$$\bar{q}_u \bar{\alpha}_u \Rightarrow \bar{q}_v \bar{\alpha}_v$$

onde: $\bar{q}_k \in \bar{Q}$;

$$\bar{\alpha}_k \in \bar{\Sigma}^*; \text{ com } k = u, v$$

levando o autômato finito $\bar{A}$ à nova situação:

$$\bar{t}_{i+1} = (\bar{q}_v, \bar{\alpha}_v \phi)$$

Denota-se esta transição por:

$$\bar{t}_i \Rightarrow \bar{t}_{i+1}$$

$$(\bar{q}_u, \bar{\alpha}_u \phi) \Rightarrow (\bar{q}_v, \bar{\alpha}_v \phi)$$

- Diz-se que um autômato finito $\bar{A}$ reconhece $\bar{\alpha} \in \bar{\Sigma}^*$ se e somente se existir uma sequência de transições:

$$\bar{t}_0 \Rightarrow \bar{t}_1 \Rightarrow \bar{t}_2 \Rightarrow \dots \Rightarrow \bar{t}_{n-1} \Rightarrow \bar{t}_n$$

ou simplesmente $\bar{t}_0 \Rightarrow^* \bar{t}_n$

onde: $\bar{t}_k = (\bar{q}_k, \bar{\alpha}_k \phi)$, $\bar{q}_k \in \bar{Q}$; $\bar{\alpha}_k \in \bar{\Sigma}^*$, $0 < k < n$

$$\bar{t}_n = (\bar{q}_F, \phi), \bar{q}_F \in \bar{F}$$

Nestas condições diz-se que $\bar{t}_n$ é uma situação final do autômato finito $\bar{A}$ para a cadeia de entrada $\bar{\alpha}_0$

O RECONHECEDOR PROPOSTO

O reconhecedor aqui apresentado pode ser definido por uma óctupla:

$$M = (Q, A, \Sigma, \Gamma, P, Z_0, q_0, F)$$

onde: A é o conjunto finito não vazio de submáquinas de M

Uma submáquina $a_i \in A$ é uma quintupla

$$a_i = (Q_i, \Sigma_i, P_i, E_i, S_i)$$

onde: $Q_i = [q_{ij} \in Q] \subset Q$ é o conjunto finito não vazio de estados de a_i

$\Sigma_i \subset \Sigma$ é o conjunto finito não vazio de terminais correspondentes ao alfabeto de entrada de a_i

$P_1 \subset P$ é o conjunto finito não vazio de produção de a_1 , que são expressões que podem ser de uma das seguintes formas:

- i) $\gamma q_{ik} \beta \rightarrow \gamma q_{ij} \alpha$
- ii) $\gamma q_{ik} \beta \rightarrow \gamma (i, \ell) q_{m,n} \alpha$
- iii) $\gamma (m,n) q_{ik} \rightarrow \gamma q_{mn} \alpha$

onde: $\gamma \in \Gamma^*$; representa o conteúdo não visível da pilha

$(m,n) \in \Gamma$; é o antigo conteúdo do topo da pilha

$$\forall (m,n) \in \Gamma$$

$(i, \ell) \in \Gamma$; representa a alteração da pilha

$q_{xy} \in Q_x$; são estados da submáquina $a_x \in A$

$\beta \in \Sigma^*$; representa a cadeia de entrada antes da aplicação da produção

$\alpha \in \Sigma^*$; representa a cadeia de entrada após a aplicação da produção

$\beta \in [\sigma \alpha, \alpha]$, onde

$\sigma \in \Sigma_i$; é o caracter de entrada consumido pela aplicação da produção

$E_i \subset Q_i$; é o conjunto de estados de entrada de a_i

$S_i \subset Q_i$; é o conjunto de estados de saída de a_i

Q é o conjunto finito não vazio de estados de M

$$Q = [q_{ij} \in Q_i \mid a_i = (Q_i, \Sigma_i, P_i, E_i, S_i) \in A]$$

$q_0 \in Q$ é o estado inicial de M com:

$q_0 = q_{00} \in E_0$ onde $a_0 = (Q_0, \Sigma_0, P_0, E_0, F_0) \in A$ é a submáquina inicial de M .

Σ é o conjunto finito não vazio de terminais correspondente ao alfabeto de entrada de M :

$$\Sigma = [\sigma \in \Sigma_i \mid a_i = (Q_i, \Sigma_i, P_i, E_i, S_i) \in A]$$

Γ é o conjunto finito não vazio de símbolos da pilha de M , isto é, é o alfabeto de pilha de M :

$$\Gamma = [Z_0] \cup [\gamma_{ij} = (i, j) \mid q_{ij} \in Q_i \text{ para algum } a_i = (Q_i, \Sigma_i, P_i, E_i, S_i) \in A]$$

$Z_0 \in \Gamma$ é o símbolo inicial da pilha

P é o conjunto finito não vazio de produções de M , dado por:

$$P = [p_{ij} \in P_i \mid a_i = (Q_i, \Sigma_i, P_i, E_i, S_i) \in A]$$

F é o conjunto finito não vazio de estados

finais de M :

$$F \subset [q_{0j} \in S_0 \mid a_0 = (Q_0, \Sigma_0, P_0, E_0, S_0) \in A]$$

— Define-se situação t da máquina proposta M como sendo a tripla:

$$t = (\gamma, q, \alpha\phi), \text{ onde:}$$

$\gamma \in \Gamma^*$ é o conteúdo da pilha;

$q \in Q$ é o estado da máquina M ;

$\alpha \in \Sigma^*$ é a cadeia de terminais apresentada à máquina para reconhecimento nesta situação;

$\phi \notin \Sigma$ é um símbolo que indica fim da cadeia.

Para o caso particular:

$$t_0 = (Z_0, q_0, \alpha_0\phi), \text{ em que:}$$

$\alpha_0 \in \Sigma^*$ é a cadeia de terminais a ser reconhecida por M

t_0 é denominado *situação inicial* da máquina M

— Define-se transição da máquina M como a aplicação de uma produção $p \in P$ qualquer, sobre sua situação:

$$t_i = (\gamma_u, q_u, \alpha_u\phi)$$

levando M à nova situação

$$t_{i+1} = (\gamma_v, q_v, \alpha_v\phi)$$

— Denota-se esta transição por $t_i \Rightarrow t_{i+1}$, ou seja,

$$(\gamma_u, q_u, \alpha_u\phi) \Rightarrow (\gamma_v, q_v, \alpha_v\phi)$$

— Diz-se que a máquina M reconhece $\alpha \in \Sigma^*$ se e somente se existir uma seqüência de transições:

$$t_0 \Rightarrow t_1 \Rightarrow t_2 \Rightarrow \dots \Rightarrow t_{n-1} \Rightarrow t_n$$

ou simplesmente $t_0 \Rightarrow^* t_n$, onde

$$t_k = (\gamma_k, q_k, \alpha_k\phi), \gamma_k \in \Gamma^*; q_k \in Q,$$

$$\alpha_k \in \Sigma^*, 0 \leq k < n$$

$$t_n = (Z_0, q_F, \phi), \text{ com } q_F \in F.$$

Nestas condições t_n diz-se uma *situação final* da máquina M para a cadeia de entrada α_0 .

COMPARAÇÃO COM O AUTÔMATO DE PILHA

Estuda-se aqui a equivalência dos modelos visando demonstrar a sua equipotência com o autômato de pilha.

1ª CONDIÇÃO: O autômato de pilha simula a máquina proposta no seu caso geral. Seja a máquina proposta M :

$$M = (Q, A, \Sigma, \Gamma, P, Z_0, q_0, F)$$

Constrói-se um autômato de pilha

$\hat{A} = (\hat{Q}, \hat{\Sigma}, \hat{\Gamma}, \hat{P}, \hat{q}_0, \hat{Z}_0, \hat{F})$ que simula a máquina proposta fazendo-se:

$$1. \hat{\Sigma} = \Sigma \quad (0)$$

$$2. \hat{Q} = \{\hat{q}_{ij} \mid q_{ij} \in Q_i \text{ para todo } a_i \in A\} \quad (1)$$

3. Como, no autômato de pilha, $\hat{\Gamma}$ pode ser qualquer, pode-se fazer: $\hat{\Gamma} = \Gamma$, ou seja,

$$\begin{aligned} \hat{\Gamma} &= [\hat{Z}_{ij} \mid \hat{Z}_{ij} = (i, j) = \\ &= \gamma_{ij} \in \Gamma - [Z_0]] \cup [\hat{Z}_0] \end{aligned} \quad (2)$$

4. Considerando-se que $q_0 = q_{00}$, então por causa de (1), tem-se:

$$\hat{q}_0 = q_{00}$$

5. Devido a (2), tem-se:

$$\hat{Z}_0 = Z_0$$

6. Como (1), para cada $q_{ij} \in Q$ existe um e só um $\hat{q}_{ij} \in \hat{Q}$. Pode-se então construir $\hat{F}$ tal que, para cada $q_{ij} \in F$, existe um e só um $\hat{q}_{ij} \in \hat{F}$:

$$\hat{F} = \{\hat{q}_{ij} \in \hat{Q} \mid q_{ij} \in F\}$$

Mantém-se a relação

$$\hat{F} \subset \hat{Q}$$

7. No autômato proposto há 3 tipos de produção:

$$(I) \quad \gamma \ q_{ik} \ \beta \rightarrow \gamma \ q_{ik} \ \alpha$$

$$(II) \quad \gamma \ q_{ik} \ \beta \rightarrow \gamma \ (i, l) \ q_{mn} \ \alpha$$

$$(III) \quad \gamma \ (m, n) \ q_{ik} \ \beta \rightarrow \gamma \ q_{mn} \ \alpha$$

No autômato de pilha, tem-se:

$$(IV) \quad \hat{\gamma} \ \hat{Z} \ \hat{q}_i \ \hat{\beta} \rightarrow \hat{\gamma} \ \hat{\delta} \ \hat{q}_j \ \hat{\alpha}$$

Considerando-se que as produções da máquina proposta não especificam em nenhum caso o topo da pilha como fator de decisão de transição, e que no autômato de pilha as produções exigem tal comparação (ambos por definição), conclui-se que é suficiente para obter o mapeamento das produções (I), (II) e (III) expandir cada uma delas em $|\Gamma|$ produções do tipo (IV) de acordo com as seguintes regras de mapeamento:

— caso I: criar um conjunto de produções da forma

$$\hat{\gamma} \ \hat{Z}_{uv} \ \hat{q}_{ik} \ \hat{\beta} \rightarrow \hat{\gamma} \ \hat{Z}_{uv} \ \hat{q}_{ij} \ \hat{\alpha}$$

obtido pela substituição de γ por $\hat{\gamma} \ \hat{Z}_{uv}$ para todo $\hat{Z}_{uv} \in \hat{\Gamma}$; q_{ik} por $\hat{q}_{ik}$; q_{ij} por $\hat{q}_{ij}$; β por $\hat{\beta}$ e α por $\hat{\alpha}$

— caso II: criar um conjunto de produções da forma:

$$\hat{\gamma} \ \hat{Z}_{uv} \ \hat{q}_{ik} \ \hat{\beta} \rightarrow \hat{\gamma} \ \hat{Z}_{uv} \ \hat{Z}_{il} \ \hat{q}_{mn} \ \hat{\alpha}$$

obtido pela substituição de γ por $\hat{\gamma} \ \hat{Z}_{uv}$ para todo $\hat{Z}_{uv} \in \hat{\Gamma}$; (i, l) por $\hat{Z}_{il}$; q_{ik} por $\hat{q}_{ik}$; q_{mn} por $\hat{q}_{mn}$; β por $\hat{\beta}$ e α por $\hat{\alpha}$

— caso III: criar um conjunto de produções da forma:

$$\hat{\gamma} \ \hat{Z}_{mn} \ \hat{q}_{ik} \ \hat{\beta} \rightarrow \hat{\gamma} \ \hat{q}_{mn} \ \hat{\alpha}$$

obtido pela substituição de $\gamma \ (m, n)$ por $\hat{\gamma} \ \hat{Z}_{mn}$ para todo $\hat{Z}_{mn} \in \hat{\Gamma}$; q_{ik} por $\hat{q}_{ik}$; q_{mn} pelo $\hat{q}_{mn}$ correspondente a cada $\hat{Z}_{mn}$ utilizado; β por $\hat{\beta}$ e α por $\hat{\alpha}$.

O autômato de pilha assim constituído simula portanto a máquina proposta devido à correspondência utilizada na sua construção.

2ª CONDIÇÃO: A máquina proposta simula o autômato de pilha.

Seja um autômato de pilha.

$$\pi = (\hat{Q}, \hat{\Sigma}, \hat{P}, \hat{q}_0, \hat{Z}_0, \hat{F})$$

Sem perda de generalidade, pode-se considerar que este autômato reconhece uma cadeia de entrada $\hat{\alpha} \in \hat{\Sigma}^*$ por esgotamento da cadeia de entrada e simultâneo esvaziamento da pilha, isto é, ocorre a sequência de transições

$$(\hat{Z}_0, \hat{q}_0, \hat{\alpha}\phi) \Rightarrow^* (\hat{Z}_0, \hat{q}_F, \phi)$$

Ainda sem perda de generalidade é possível, a partir de π , construir um novo autômato de pilha $\pi' = (\hat{Q}', \hat{\Sigma}', \hat{\Gamma}', \hat{P}', \hat{q}'_0, \hat{Z}'_0, \hat{F}')$, em que cada produção de $\hat{P}'$ tem uma das formas seguintes apenas:

$$1) \quad \hat{\gamma} \ \hat{Z} \ \hat{q}_i \ \hat{\beta} \rightarrow \hat{\gamma} \ \hat{Z} \ \hat{q}_i \ \hat{\alpha}$$

$$2) \quad \hat{\gamma} \ \hat{Z} \ \hat{q}_i \ \hat{\beta} \rightarrow \hat{\gamma} \ \hat{Z} \ \hat{Z}_u \ \hat{q}_j \ \hat{\alpha}$$

$$3) \quad \hat{\gamma} \ \hat{Z} \ \hat{q}_i \ \hat{\beta} \rightarrow \hat{\gamma} \ \hat{q}_j \ \hat{\alpha}$$

onde $\hat{Z}, \hat{Z}_u \in \hat{\Gamma}$

Note-se que, sendo as produções de $\hat{P}'$ da forma geral:

$$4) \quad \hat{\gamma} \ \hat{Z} \ \hat{q}_i \ \hat{\beta} \rightarrow \hat{\gamma} \ \hat{\delta} \ \hat{q}_j \ \hat{\alpha}$$

e considerando-se que as únicas possíveis formas de $\hat{\delta}$ podem ser resumidas em:

$$\hat{\delta} = \mu \hat{\delta}' \text{ onde } \mu \text{ ou } \epsilon \text{ vazio, ou } \epsilon \hat{Z}$$

e $\hat{\delta}' \in \hat{\Gamma}^*$, com $|\hat{\delta}'| = n$, ou seja, $\hat{\delta}'$ é uma cadeia (eventualmente vazia) — de n símbolos de $\hat{\Gamma}$:

$$\hat{\delta}' = \hat{Z}_1 \hat{Z}_2 \dots \hat{Z}_n, \text{ com}$$

$$\hat{Z}_i \in \hat{\Gamma}^*, i = 1, \dots, n; n \geq 0$$

é possível obter π' a partir de π através da aplicação do seguinte método:

— Para cada produção $\hat{p}_k \in \hat{P}$ da forma geral:

$$\hat{p}_k: \hat{\gamma} \hat{Z} \hat{q}_i \hat{\beta} \rightarrow \hat{\gamma} \mu \hat{\delta}' \hat{q}_j \hat{\alpha},$$

cria-se um conjunto $\hat{P}_k$ de produções p_k^{mq} da forma:

$$p_k^{mq}: \gamma_m^q S_k^m \alpha \rightarrow \gamma_{m+1}^p S_k^{m+1} \alpha_{m+1}$$

onde para:

$m = 0$ tem-se: $q = 0$;

$$p = w, \text{ se } \mu = \hat{Z} = \hat{Z}_w \in \hat{\Gamma}$$

ou $p = 0$, se μ é vazio;

$$\gamma_0^0 = \hat{\gamma} \hat{Z}$$

$$\gamma_{m+1}^p = \hat{\gamma} \mu$$

$m = 1$ tem-se: γ_1^q é $\hat{\gamma} \hat{Z}$ se $\mu = \hat{Z}$ com

$$q = w \text{ para } \hat{Z} = \hat{Z}_w \in \hat{\Gamma}$$

caso contrário será $\hat{\gamma}' \hat{Z}_q$ para todo

$$\hat{Z}_q \in \hat{\Gamma}, 1 \leq q \leq |\hat{\Gamma}|, \hat{\gamma}' \in \hat{\Gamma}^*$$

$$\text{e } \gamma_{m+1}^p = \gamma_m^q \hat{Z}_t$$

$$\gamma_{m+1}^q = \gamma_{m+1}^p, \text{ com}$$

$$p = w, \hat{Z}_t = \hat{Z}_w \in \hat{\Gamma}, \hat{Z}_w \text{ da}$$

cadeia $\hat{\delta}'$,

e

$$\alpha_0 = \beta$$

$$\alpha_t = \alpha \text{ para } t > 0$$

$$\hat{Q}_k = [S_k^t \mid 0 \leq t \leq n+1, \text{ com } S_k^0 = \hat{q}_i,$$

$$\text{e } S_k^{n+1} = \hat{q}_j]$$

onde $0 \leq m \leq n$

Obtém-se o novo autômato $\pi' =$

$$= (\hat{Q}', \hat{\Sigma}, \hat{\Gamma}, \hat{P}', \hat{q}_0', \hat{Z}_0, \hat{F}') \text{ fazendo-se:}$$

$$\hat{Q}' = [q \mid q = S_k^t \in \hat{Q}_k, \forall p_k \in \hat{P}]$$

$$\hat{P}' = [p \mid p = p_k^{mq} \in \hat{P}_k, \forall p_k \in \hat{P}]$$

$$\text{onde } \hat{q}_0' = \hat{q}_0$$

$$\text{e } \hat{F}' = \hat{F}, \text{ ambos por construção}$$

A partir do novo autômato assim obtido:

$$\pi' = (\hat{Q}', \hat{\Sigma}, \hat{\Gamma}, \hat{P}', \hat{q}_0', \hat{Z}_0, \hat{F}')$$

obtem-se a seguir, por construção, uma máquina proposta capaz de simular o autômato de pilha genérico inicial.

$$M = (Q, A, \Sigma, \Gamma, P, Z_0, q_0, F)$$

1. Faz-se $\Sigma = \hat{\Sigma}$ identificando-se os alfabetos de entrada

2. Escolhe-se A como sendo um conjunto de submáquinas a_i , cada qual correspondente a um elemento do alfabeto $\hat{\Gamma}$ de pilha do autômato π' .

O conjunto A conterá pois $|\hat{\Gamma}|$ submáquinas:

(Observe-se que: a_0 corresponde a $\hat{Z}_0 \in \hat{\Gamma}$)

$$A = [a_i = (Q_i, \Sigma_i, P_i, E_i, S_i) \mid \hat{Z}_i \in \hat{\Gamma}]$$

onde para cada a_i faz-se:

$$- \Sigma_i = \hat{\Sigma}$$

$$- Q_i = [q_{ij} \mid \hat{Z}_i \in \hat{\Gamma}, \hat{q}_j \in \hat{Q}]$$

— $E_i \subset Q_i$ é o conjunto finito não vazio de estados de entrada da submáquina a_i

— $S_i \subset Q_i$ é o conjunto finito não vazio de estados de saída da submáquina a_i

Os elementos de E_i e S_i surgirão na construção de P_i adiante.

3. Obtém-se o conjunto Q de estados da máquina M a partir da união de $|\hat{\Gamma}|$ conjuntos Q_i :

$$Q = [q \mid q \in Q_i, \text{ com } \hat{Z}_i \in \hat{\Gamma}]$$

4. $F = S_0 = [q_{0k} \in Q \mid \hat{q}_k \in \hat{F}]$ é o conjunto de estados finais da máquina M correspondente ao conjunto de estados finais do autômato π'

5. Γ é o alfabeto de pilha de M definido como

$$\Gamma = Z = [(i, j) \mid q_{ij} \in Q] \cup [Z_0]$$

onde:

6. $Z_0 = \hat{Z}_0$ é o símbolo da pilha de M

7. q_0 é o estado inicial de M .

$$q_0 = q_{00} \in E_0 \subset Q_0$$

Observe-se que entre as pilhas do autômato π' e da máquina M pode ser imposta uma correspondência biunívoca entre cada possível topo da pilha do autômato π' dada por $\hat{Z}_i \in \hat{\Gamma}$ e a submáquina $a_i \in A$ da máquina M . Assim sendo, a um empilhamento de $\hat{Z}_j \in \hat{\Gamma}$ em π' , estando este previamente com $\hat{Z}_k \in \hat{\Gamma}$ no topo da pilha, corresponderá um empilhamento em M , (operando previamente na submáquina $a_k \in A$), acompanhado de uma mudança de submáquina para $a_j \in A$.

8. Obtém-se o conjunto P de produções da máquina M a partir das produções de $\hat{P}'$ do autômato de pilha π' através de mapeamento, considerando-se cada um dos três possíveis formatos das produções de $\hat{P}'$:

CASO 1) A produção $\hat{p}_k \in \hat{P}'$ tem a forma (1)

$$\hat{p}_k: \hat{\gamma} \hat{Z}_t \hat{q}_i \hat{\beta} \rightarrow \hat{\gamma} \hat{Z}_t \hat{q}_j \hat{\alpha}$$

neste caso constrói-se a produção:

$$p_{tk}^0: \gamma q_{ti} \beta \rightarrow \gamma q_{tj} \alpha$$

CASO 2) A produção $\hat{p}_k \in \hat{P}'$ tem a forma (2),

constrói-se para cada:

$$\hat{p}_k: \hat{\gamma} \hat{Z}_t \hat{s}_i \hat{\beta} \rightarrow \hat{\gamma} \hat{Z}_t \hat{Z}_u \hat{s}_j \hat{\alpha} \text{ em } \hat{P}',$$

um conjunto C_k de produções da máquina M , definido como:

$$C_k = [p_{tk}^w \mid p_{tk}^w: \gamma q_{ti} \beta \rightarrow$$

$$\rightarrow \gamma(t, j_w) q_{uj} \alpha \text{ para todo}$$

$$\hat{p}_w: \hat{\gamma} \hat{Z}_u \hat{q}_{iw} \hat{\beta}^w \rightarrow$$

$$\rightarrow \hat{\gamma} \hat{q}_{jw} \hat{\alpha} \text{ em } \hat{P}', w \neq k]$$

Obtém-se, a partir deste conjunto, os estados de entrada da máquina a_u , uma vez que as produções p_{tk}^w correspondem a transições com empilhamento e portanto mudança para estado de entrada de outra máquina:

$$E_u = [q_{uj} \mid p_{tk}^w: \gamma q_{ti} \beta \rightarrow$$

$$\rightarrow \gamma(t, j_w) q_{uj} \alpha, \text{ em } C_k]$$

CASO 3) A produção $\hat{p}_k \in \hat{P}'$ tem a forma 3.

$$\hat{p}_k: \hat{\gamma} \hat{\gamma}_t \hat{q}_i \hat{\beta} \rightarrow \hat{\gamma} \hat{q}_j \hat{\alpha}$$

Neste caso constrói-se a produção:

$$p_{tk}^0: \gamma(m, n) q_{ti} \beta \rightarrow \gamma q_{mj} \alpha$$

Obtém-se os estados de saída da máquina a_i como o conjunto:

$$S_i = [q_{ti} \in Q_t \mid p_{tk}^0: \gamma(m, n) q_{ti} \beta \rightarrow$$

$$\rightarrow \gamma q_{mj} \alpha \text{ em } P,$$

$$\text{com } \hat{p}_k: \hat{\gamma} \hat{Z}_t \hat{q}_i \hat{\beta} \rightarrow \hat{\gamma} \hat{q}_j \hat{\alpha} \text{ em } \hat{P}']$$

Embora aparentemente $\hat{q}_j$ não esteja sendo explicitamente utilizado neste mapeamento, é importante notar que, por construção das produções de M oriundas do tratamento de produções de π' so segundo caso, o caracter de pilha (m, n) é sempre compatível com $\hat{q}_j$.

Conclui-se assim que a máquina proposta é, por construção, capaz de simular autômato de pilha, inferindo-se desta maneira a equipotência dos dois modelos.

OBSERVAÇÕES FINAIS

No modelo anteriormente apresentado, é permitido que uma submáquina seja atingida através de qualquer dos seus estados. Tais submáquinas podem ser interpretadas como sendo uma família de submáquinas, todas iguais, cada qual podendo ser atingida por um único estado, respectivamente correspondentes a cada um dos possíveis estados pelos quais a submáquina inicial poderia ser atingida.

Demonstrou-se que o modelo tem aplicabilidade no reconhecimento de linguagens Chomsky tipo 2, ou seja, livres de contexto. Isto decorre do fato de ser o modelo proposto e o autômato de pilha formas equipotentes de construção de reconhecedores para tais linguagens. Deve-se observar, entretanto, que, na maioria dos casos, é possível, através de uma construção criteriosa, obter-se reconhecedores baseados no modelo, e que apresentem grande simplicidade estrutural e clareza de comportamento. Este fato torna-se evidente quando tais reconhecedores são produzidos diretamente a partir de uma gramática da linguagem desejada. Esta construção pode ser obtida em duas etapas: inicialmente são separados os não terminais "self-embedded", os quais determinam, juntamente com a raiz da gramática, as submáquinas essenciais. Em seguida são efetuadas divisões funcionais no restante da gramática, com o que são separados outros não terminais que facilitem a obtenção de reconhecedores que apresentem boa clareza estrutural. O reconhecedor assim

obtido exibe um número pequeno de submáquinas, o que reduz sobremaneira sua complexidade. Nesta construção é possível impor que cada submáquina tenha apenas um estado inicial, sendo, desta maneira, possível comparar-se as diversas submáquinas a autômatos finitos correspondentes, cuja operação é sincronizada pela pilha do modelo. Observe-se a invariância do conteúdo da pilha de controle antes da ativação e depois da desativação de qualquer submáquina, o que, associado ao fato de que cada submáquina, por construção, corresponde ao reconhecedor de um dado não terminal, é possível construir-se, de maneira bastante simplificada, um programa capaz de simular o comportamento do modelo nestas circunstâncias. Este programa simula os autômatos finitos que correspondem às diversas submáquinas pela chamada recursiva de uma rotina única de simulação que trata apenas a leitura do próprio carácter da cadeia de entrada e a transição de estado, interna à submáquina. A pilha, neste caso, reduz-se à própria pilha que implementa a recursão, não aparecendo explicitamente seus movimentos. Com estas restrições obtém-se um programa que implementa reconhecedores extremamente simples e de grande aplicabilidade para uma grande gama de problemas da área de compilação que se apresentam tipicamente na vida profissional do engenheiro.

Por outro lado, não se deve esquecer que é possível a utilização dos recursos ainda não explorados do modelo nesta versão simplificada o que leva à construção de reconhecedores mais potentes, capazes de reconhecer linguagens

mais complexas, por exemplo, não determinísticas (como no caso mostrado adiante). É possível, ainda em tais casos, a construção de programas que simulam o modelo, com relativa facilidade, baseados na mesma técnica acima citada.

Deve-se ressaltar ainda a característica didática inerente ao modelo, o qual pode ser apresentado como uma máquina sequencial composta de uma parte que realiza decisões a partir de um carácter da cadeia de entrada e utiliza informações de uma memória organizada em pilha (LIFO) que compõe sua segunda parte. Estes conceitos são bastante simples e de domínio do estudante de engenharia. Por outro lado, pelas próprias características do modelo, a teoria aqui apresentada, bem como a maioria dos conceitos formais correlatos podem ser omitidos em uma primeira apresentação sem no entanto restringir sua aplicabilidade, que pode ser facilmente suportada por argumentação intuitiva.

EXEMPLO

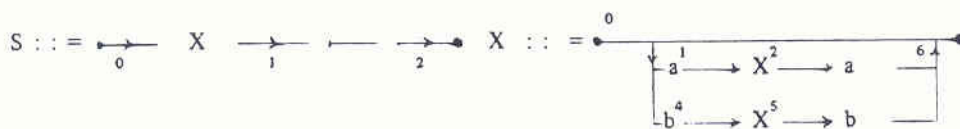
Para finalizar, mostra-se aqui uma pequena aplicação do modelo, baseado em um exemplo encontrado em Harrison (8). Trata-se de um reconhecedor de palíndromos pares (não centradas) sobre o alfabeto $[a, b]$

Uma possível gramática para esta linguagem é:

$S \rightarrow X$

$X \rightarrow a X a \mid b X b \mid \epsilon$

É imediata a construção dos seguintes diagramas de sintaxe para esta gramática:



de onde são extraídas facilmente as seguintes produções do modelo proposto:

- | | | | |
|---------|---|-----------------|--|
| $p_1 :$ | $Z_0 (S, 0) \alpha \rightarrow Z_0 Z_{S,1}$ | $(X, 0) \alpha$ | (empilha retorno, muda para submáquina X) |
| $p_2 :$ | $Z_0 (S, 1) \vdash \alpha \rightarrow Z_0$ | $(S, 2) \alpha$ | (evolui para estado final, de reconhecimento) |
| $p_3 :$ | $\gamma (X, 0) a \alpha \rightarrow \gamma$ | $(X, 1) \alpha$ | (transição normal dentro de submáquina X) |
| $p_4 :$ | $\gamma (X, 0) b \alpha \rightarrow \gamma$ | $(X, 4) \alpha$ | |
| $p_5 :$ | $\gamma (X, 1) \alpha \rightarrow \gamma Z_{X,2}$ | $(X, 0) \alpha$ | (empilha retorno, muda para submáquina X) |
| $p_6 :$ | $\gamma (X, 4) \alpha \rightarrow \gamma Z_{X,5}$ | $(X, 0) \alpha$ | |
| $p_7 :$ | $\gamma Z_{u,v} (X, 2) a \alpha \rightarrow \gamma$ | $(u, v) \alpha$ | (retorna para endereço empilhado: estado final de X) |
| $p_8 :$ | $\gamma Z_{u,v} (X, 5) b \alpha \rightarrow \gamma$ | $(u, v) \alpha$ | |
| $p_9 :$ | $\gamma Z_{u,v} (X, 0) \alpha \rightarrow \gamma$ | $(u, v) \alpha$ | (reconhece ϵ na submáquina X) |

A máquina proposta fica portanto:

$M = (Q, A, \Sigma, \Gamma, P, Z_0, q_0, F)$, onde

$$A = [S, X], \text{ com } S = (Q_s, \Sigma_s, P_s, E_s, S_s)$$

$$X = (Q_x, \Sigma_x, P_x, E_x, S_x)$$

$$\text{onde: } Q_s = [(S, i) \mid i = 0, 1, 2], \quad Q_x = [(X, i) \mid i = 0, 1, 2, 3, 4, 5]$$

$$\Sigma_s = \Sigma_x = \Sigma = [a, b]$$

$$P_s = [p_1, p_2], \quad P_x = [p_3, p_4, p_5, p_6, p_7, p_8, p_9]$$

$$E_s = [(S, 0)], \quad E_x = [(X, 0)]$$

$$S_s = [(S, 2)], \quad S_x = [(X, 6)]$$

$$Q = Q_s \cup Q_x$$

$$\Gamma = [Z_0] \cup [Z_{uv} \mid (u, v) \in Q_s \cup Q_x]$$

$$P = P_s \cup P_x$$

$$q_0 = (S, 0)$$

$$F = [(S, 2)]$$

Observe-se que a máquina M é um reconhecedor não determinístico, e que as produções de retorno p_7 , p_8 e p_9 correspondem à compactação das seguintes produções:

$$\gamma(X, 2) a \alpha \rightarrow \gamma(X, 6) \alpha$$

$$\gamma(X, 5) b \alpha \rightarrow \gamma(X, 6) \alpha$$

$$\gamma(X, 0) \alpha \rightarrow \gamma(X, 6) \alpha$$

$$\gamma Z_{uv}(X, 6) \alpha \rightarrow \gamma(u, v) \alpha$$

as quais explicitam o estado final (X, 6) da submáquina X.

Observe-se o movimento da máquina M no reconhecimento da cadeia a b b b b a | - :

possíveis produções
a aplicar:

Z_0	(S, 0)	a b b b b a -	$\underline{p_1}$
$Z_0 \quad Z_{s,1}$	(S, 0)	a b b b b a -	$\underline{p_3}, p_9$
$Z_0 \quad Z_{s,1}$	(X, 1)	b b b b a -	$\underline{p_5}$
$Z_0 \quad Z_{s,1} \quad Z_{x,2}$	(X, 0)	b b b b a -	$\underline{p_4}, p_9$
$Z_0 \quad Z_{s,1} \quad Z_{x,2}$	(X, 4)	b b b a -	$\underline{p_6}$
$Z_0 \quad Z_{s,1} \quad Z_{x,2} \quad Z_{x,5}$	(X, 0)	b b b a -	$\underline{p_4}, p_9$
$Z_0 \quad Z_{s,1} \quad Z_{x,2} \quad Z_{x,5}$	(X, 4)	b b a -	$\underline{p_6}$
$Z_0 \quad Z_{s,1} \quad Z_{x,2} \quad Z_{x,5} \quad Z_{x,5}$	(X, 0)	b b a -	$\underline{p_4}, p_9$
$Z_0 \quad Z_{s,1} \quad Z_{x,2} \quad Z_{x,5}$	(X, 5)	b b a -	$\underline{p_8}$
$Z_0 \quad Z_{s,1} \quad Z_{x,2}$	(X, 5)	b a -	$\underline{p_8}$
$Z_0 \quad Z_{s,1}$	(X, 2)	a -	$\underline{p_7}$
Z_0	(S, 1)	-	$\underline{p_2}$
Z_0	(S, 2)	-	estado final