AUTÓMATOS E LINGUAGENS FORMAIS

(LCC/LMAT)

4. Autómatos de Pilha

Departamento de Matemática

Universidade do Minho

2022/2023

Os autómatos de pilha (ou autómatos empilhadores) constituem um modelo de computação, mais complexo que os autómatos finitos, mas também mais expressivo, que permite reconhecer as linguagens independentes de contexto.

Contrariamente aos autómatos finitos, os autómatos de pilha dispõem de:

- memória (a pilha);
- uma cabeça de leitura/escrita para consultar e alterar o conteúdo da pilha (apenas o símbolo do topo da pilha).

Os autómatos de pilha dispõem ainda de uma fita de leitura, munida de um cursor, que percorre a fita da esquerda para a direita, lendo uma letra de cada vez, assemelhando-se neste aspeto aos autómatos finitos, que reconhecem uma palavra processando uma letra de cada vez, da esquerda para a direita.

Definição

Um autómato de pilha (ou autómato empilhador) é um septeto

$$\mathcal{E} = (Q, A, X, \delta, i, z, F)$$

onde

1. Q é um conjunto finito não vazio, chamado o conjunto de estados de $\mathcal{E}$;
2. A é um alfabeto finito, chamado o alfabeto (de entrada) de $\mathcal{E}$;
3. X é um alfabeto finito, dito o alfabeto da pilha;
4. $\delta : Q \times (A \cup \{\epsilon\}) \times X \to \mathcal{P}_{fin}(Q \times X^*)$ é uma função, designada a função transição de $\mathcal{E}$.

   Cada quíntuplo (p, a, x, q, α), em que $p, q \in Q$, $a \in A \cup \{\epsilon\}$, $x \in X$ e $\alpha \in X^*$ são tais que $(q, \alpha) \in \delta(p, a, x)$, diz-se uma transição de $\mathcal{E}$;
5. $i \in Q$, sendo chamado o estado inicial de $\mathcal{E}$;
6. $z \in X$, sendo chamado o símbolo inicial da pilha;
7. $F \subseteq Q$ é o conjunto de estados finais (ou de aceitação) de $\mathcal{E}$.

Exemplo

O septeto $$\mathcal{E}_1 = \big(\{1,2,3\},\{a,b,c\},\{z,a,b\},\delta,1,z,\{3\}\big)$$

é um autómato de pilha com

- três estados (1, 2 e 3);
- alfabeto de entrada $\{a, b, c\}$;
- alfabeto da pilha $\{z, a, b\}$;
- 1 como estado inicial;
- 3 como único estado final;
- onde δ é a função de $\{1,2,3\} \times \{a,b,c,\epsilon\} \times \{z,a,b\}$ em $\mathcal{P}_{fin}(\{1,2,3\} \times \{z,a,b\}^*)$, cujas imagens não vazias são dadas por:

$$\begin{array}{lll}
\delta(1,a,z) = \{(1,az)\}, & \delta(1,a,a) = \{(1,aa)\}, & \delta(1,a,b) = \{(1,ab)\}, \\
\delta(1,b,z) = \{(1,bz)\}, & \delta(1,b,a) = \{(1,ba)\}, & \delta(1,b,b) = \{(1,bb)\}, \\
\delta(1,c,z) = \{(3,z)\}, & \delta(1,c,a) = \{(2,a)\}, & \delta(1,c,b) = \{(2,b)\}, \\
\delta(2,a,a) = \{(2,\epsilon)\}, & \delta(2,b,b) = \{(2,\epsilon)\}, & \delta(2,\epsilon,z) = \{(3,z)\}.
\end{array}$$

Definição

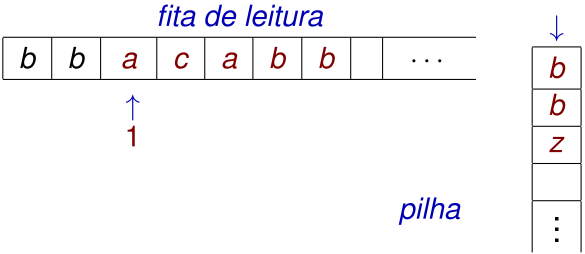
Seja $\mathcal{E} = (Q, A, X, \delta, i, z, F)$ um autómato de pilha (AP).

Uma configuração de $\mathcal{E}$ é um triplo $(q, w, \sigma) \in Q \times A^* \times X^*$, tal que:

1. $q \in Q$ e representará o estado atual do autómato;
2. $w \in A^*$ e representará a palavra que está escrita na fita para a direita do cursor;
3. $\sigma \in X^*$ e representará a palavra que está escrita na pilha (de cima para baixo).

Uma configuração permitirá, assim, representar a situação do autómato $\mathcal{E}$ num dado instante, nomeadamente, o estado, o conteúdo da fita e da pilha e a posição do cursor na fita.

Por exemplo, a configuração $(1, acabb, bbz)$ poderia representar uma situação do AP $\mathcal{E}_1$ do exemplo anterior num dado instante. Tal configuração pode ser visualizada da seguinte forma



Note-se que em $\mathcal{E}_1$ tem-se $\delta(1, a, b) = \{(1, ab)\}$. Então, diremos que se pode passar da configuração $(1, acabb, bbz)$ para a configuração $(1, cabb, abbz)$.

Definição

Seja $\mathcal{E} = (Q, A, X, \delta, i, z, F)$ um autómato de pilha.
A relação de computação direta (em configurações) é notada por $\underset{\mathcal{E}}{\longrightarrow}$ e, para $a \in A \cup \{\epsilon\}$ e $x \in X$, tem-se

$$(p, aw, x\sigma) \underset{\mathcal{E}}{\longrightarrow} (q, w, \alpha\sigma) \quad \text{quando } (q, \alpha) \in \delta(p, a, x),$$

dizendo-se, então, que o par (ordenado) é uma computação direta da 2^a configuração a partir da 1^a configuração.

Quando duas configurações (q_1, w_1, σ_1) e (q_2, w_2, σ_2) são iguais ou a segunda pode ser obtida da primeira através de uma sequência finita de computações diretas, escrevemos

$$(q_1, w_1, \sigma_1) \overset{*}{\underset{\mathcal{E}}{\longrightarrow}} (q_2, w_2, \sigma_2)$$

chamando computação de (q_2, w_2, σ_2) a partir de (q_1, w_1, σ_1) a tal sequência de computações diretas.

A relação binária $\overset{*}{\underset{\mathcal{E}}{\longrightarrow}}$ é chamada a relação de computação (em configurações).

Exemplo

Seja $\mathcal{E}_1 = \big(\{1,2,3\},\{a,b,c\},\{z,a,b\},\delta,1,z,\{3\}\big)$ o autómato de pilha do exemplo do slide 4, em que

$$
\begin{array}{lll}
\delta(1,a,z) = \{(1,az)\}, & \delta(1,a,a) = \{(1,aa)\}, & \delta(1,a,b) = \{(1,ab)\}, \\
\delta(1,b,z) = \{(1,bz)\}, & \delta(1,b,a) = \{(1,ba)\}, & \delta(1,b,b) = \{(1,bb)\}, \\
\delta(1,c,z) = \{(3,z)\}, & \delta(1,c,a) = \{(2,a)\}, & \delta(1,c,b) = \{(2,b)\}, \\
\delta(2,a,a) = \{(2,\epsilon)\}, & \delta(2,b,b) = \{(2,\epsilon)\}, & \delta(2,\epsilon,z) = \{(3,z)\}.
\end{array}
$$

Um exemplo de uma computação a partir da configuração $(1,abcba,z)$ é:

$$
\begin{aligned}
(1,abcba,z) &\underset{\mathcal{E}}{\longrightarrow} (1,bcba,az) \underset{\mathcal{E}}{\longrightarrow} (1,cba,baz) \\
&\underset{\mathcal{E}}{\longrightarrow} (2,ba,baz) \underset{\mathcal{E}}{\longrightarrow} (2,a,az) \underset{\mathcal{E}}{\longrightarrow} (2,\epsilon,z) \underset{\mathcal{E}}{\longrightarrow} (3,\epsilon,z)
\end{aligned}
$$

De facto, as configurações que ocorrem na computação anterior constituem a totalidade das configurações computáveis a partir da configuração $(1,abcba,z)$.

Definição

Seja $\mathcal{E} = (Q, A, X, \delta, i, z, F)$ um autómato de pilha.

1. A configuração inicial associada a uma palavra $u \in A^*$ é (i, u, z).
2. Uma configuração (f, ϵ, σ) em que $f \in F$, é chamada uma configuração final (ou uma configuração de aceitação).
3. Uma palavra $u \in A^*$ é reconhecida (ou aceite) por $\mathcal{E}$ quando para alguma configuração de aceitação (f, ϵ, σ) se tem

$$(i, u, z) \xrightarrow[\mathcal{E}]{*} (f, \epsilon, \sigma)\,,$$

ou seja, quando existe uma computação (dita bem sucedida) de uma configuração final a partir da configuração inicial associada à palavra u.

Exemplo

Consideremos de novo o AP $\mathcal{E}_1$ dos slides anteriores.

Por exemplo, a configuração inicial associada à palavra *abcba* é $(1, abcba, z)$.

Um exemplo de uma configuração final é $(3, \epsilon, z)$.

No slide 8 contruímos uma computação de $(3, \epsilon, z)$ a partir de $(1, abcba, z)$. Assim, podemos dizer que tal computação se trata de uma computação bem sucedida e concluir que a palavra *abcba* é reconhecida por $\mathcal{E}_1$.

Definição

Seja $\mathcal{E} = (Q, A, X, \delta, i, z, F)$ um autómato de pilha. A linguagem reconhecida por $\mathcal{E}$, representada por $L(\mathcal{E})$, é o conjunto das palavras reconhecidas por $\mathcal{E}$; ou seja,

$$L(\mathcal{E}) = \{u \in A^* : \exists f \in F\ \exists \sigma \in X^*,\ (i, u, z) \xrightarrow[\mathcal{E}]{*} (f, \epsilon, \sigma)\}.$$

Exemplo

O AP $\mathcal{E}_1 = \big(\{1,2,3\}, \{a,b,c\}, \{z,a,b\}, \delta, 1, z, \{3\}\big)$ considerado em slides anteriores, reconhece a linguagem

$$L_1 = \{wcw^I : w \in \{a,b\}^*\}.$$

A estratégia do autómato é explicada a seguir. Partindo da configuração inicial de uma palavra $u \in \{a,b,c\}^*$,

- $\mathcal{E}_1$ começa por ler a palavra u até chegar à primeira ocorrência da letra c, caso c ocorra em u, ou até ao fim, caso contrário. À medida que vai lendo a palavra, $\mathcal{E}_1$ vai fazendo uma cópia na pilha;
- se u não tem ocorrências da letra c, então $\mathcal{E}_1$ pára numa configuração não final e portanto não aceita a palavra u. Caso contrário, u é da forma $u = u_1 c u_2$ com $u_1 \in \{a,b\}^*$. Neste caso, quando a primeira ocorrência de c em u é atingida, o prefixo u_1 de u acabou de ser copiado para a pilha;
- depois de passar a primeira ocorrência de c, $\mathcal{E}_1$ compara o prefixo u_1 (lendo-o pela ordem inversa, na pilha) com o sufixo u_2 de u. Portanto, a palavra é aceite se e só se $u_1^I = u_2$.

Exemplo (continuação)

A estratégia do autómato $\mathcal{E}_1$ é concretizada da seguinte forma:

$$\left.\begin{array}{l}\delta(1,a,z)=\{(1,az)\}\\ \delta(1,b,z)=\{(1,bz)\}\\ \delta(1,c,z)=\{(3,z)\}\end{array}\right\}\text{ transições a partir da configuração inicial}$$

$$\left.\begin{array}{l}\delta(1,a,a)=\{(1,aa)\}\\ \delta(1,a,b)=\{(1,ab)\}\\ \delta(1,b,a)=\{(1,ba)\}\\ \delta(1,b,b)=\{(1,bb)\}\end{array}\right\}\text{ transições até à posição central}$$

$$\left.\begin{array}{l}\delta(1,c,a)=\{(2,a)\}\\ \delta(1,c,b)=\{(2,b)\}\end{array}\right\}\text{ transições na posição central}$$

$$\left.\begin{array}{l}\delta(2,a,a)=\{(2,\epsilon)\}\\ \delta(2,b,b)=\{(2,\epsilon)\}\end{array}\right\}\text{ transições após a posição central}$$

$$\delta(2,\epsilon,z)=\{(3,z)\}\ \}\text{ transição para uma configuração final.}$$

Teorema

Toda a linguagem regular é reconhecida por algum autómato de pilha.

Demonstração: Suponhamos que $L = L(\mathcal{A})$, onde

$$\mathcal{A} = (Q, A, \delta, i, F)$$

é um autómato finito. Seja $\mathcal{E}$ o autómato de pilha

$$\mathcal{E} = (Q, A, \{z\}, \delta', i, z, F)$$

em que, $\delta'(q, a, z) = \begin{cases} \emptyset & \text{se } \delta(q,a) = \emptyset \\ \{(p, z) : p \in \delta(q, a)\} & \text{se } \delta(q,a) \neq \emptyset \end{cases}$

para cada $q \in Q$ e $a \in A$. Então um caminho bem sucedido arbitrário

$$i \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \cdots q_{n-1} \xrightarrow{a_n} q_n$$

em $\mathcal{A}$ pode ser imitado por uma computação bem sucedida

$$(i, a_1 a_2 \cdots a_n, z) \to (q_1, a_2 \cdots a_n, z) \to (q_2, a_3 \cdots a_n, z) \to \cdots \to (q_{n-1}, a_n, z) \to (q_n, \epsilon, z)$$

em $\mathcal{E}$. Por outro lado, as únicas computações bem sucedidas em $\mathcal{E}$ são deste tipo, donde $L(\mathcal{A}) = L(\mathcal{E})$.

Exemplo

Neste exemplo apresentamos um autómato de pilha não determinista $\mathcal{E}_2$ que reconhece a linguagem

$$L_2 = \{ww^I : w \in \{a,b\}^*\}$$

das palavras capicua de comprimento par sobre o alfabeto $\{a,b\}$.
O AP $\mathcal{E}_2$ é dado por

$$\mathcal{E}_2 = \big(\{1,2,3\}, \{a,b\}, \{z,a,b\}, \delta, 1, z, \{3\}\big)$$

onde as imagens não vazias de δ são definidas por

$$\begin{aligned}
\delta(1,a,z) &= \{(1,az)\}, & \delta(1,a,b) &= \{(1,ab)\},\\
\delta(1,b,z) &= \{(1,bz)\}, & \delta(1,b,a) &= \{(1,ba)\},\\
\delta(1,a,a) &= \{(1,aa),(2,\epsilon)\}, & \delta(1,b,b) &= \{(1,bb),(2,\epsilon)\},\\
\delta(2,a,a) &= \{(2,\epsilon)\}, & \delta(2,b,b) &= \{(2,\epsilon)\},\\
\delta(1,\epsilon,z) &= \{(2,z)\}, & \delta(2,\epsilon,z) &= \{(3,z)\}.
\end{aligned}$$

Em $\mathcal{E}_2$ tem-se, por exemplo, a seguinte computação bem sucedida

$$(1, a^2b^2a^2, z) \to (1, ab^2a^2, az) \to (1, b^2a^2, a^2z) \to (1, ba^2, ba^2z)$$
$$\to (2, a^2, a^2z) \to (2, a, az) \to (2, \epsilon, z) \to (3, \epsilon, z).$$

Exemplo (continuação)

As explicações seguintes ajudam a compreender a estratégia do autómato $\mathcal{E}_2$:

$$\delta(1,\epsilon,z)=\{(2,z)\}\ \Big\}\ \text{transição para que a palavra vazia seja aceite}$$

$$\left.\begin{array}{l}\delta(1,a,z)=\{(1,az)\}\\ \delta(1,b,z)=\{(1,bz)\}\\ \delta(1,a,b)=\{(1,ab)\}\\ \delta(1,b,a)=\{(1,ba)\}\end{array}\right\}\ \text{antes da posição central}$$

$$\left.\begin{array}{l}\delta(1,a,a)=\{(1,aa),(2,\epsilon)\}\\ \delta(1,b,b)=\{(1,bb),(2,\epsilon)\}\end{array}\right\}\ \begin{array}{l}\text{antes da posição central ou}\\ \text{na posição central}\end{array}$$

$$\left.\begin{array}{l}\delta(2,a,a)=\{(2,\epsilon)\}\\ \delta(2,b,b)=\{(2,\epsilon)\}\end{array}\right\}\ \text{após a posição central}$$

$$\delta(2,\epsilon,z)=\{(3,z)\}\ \Big\}\ \text{transição para uma configuração final.}$$

Observação

O modelo de autómato de pilha que introduzimos admite algumas variantes, mas que não alteram a classe de linguagens que reconhecem.

Uma dessas variantes modifica o critério de aceitação: em vez de se aceitar uma palavra u para a qual existe uma computação

$$(i, u, z) \xrightarrow[\mathcal{E}]{*} (f, \epsilon, \sigma)$$

em que f é um estado final, pode aceitar-se u quando existe uma computação

$$(i, u, z) \xrightarrow[\mathcal{E}]{*} (q, \epsilon, \epsilon)$$

que acaba numa configuração em que a pilha está vazia.

O primeiro critério de aceitação é chamado critério dos estados finais e o segundo é chamado critério da pilha vazia.

Definição

Seja $\mathcal{E}$ um autómato de pilha. A linguagem reconhecida por $\mathcal{E}$ pelo critério da pilha vazia, é

$$L_{pv}(\mathcal{E}) = \{u \in A^* : \exists q \in Q,\ (i, u, z) \xrightarrow[\mathcal{E}]{*} (q, \epsilon, \epsilon)\}.$$

Os dois tipos de reconhecimento são equivalentes:

Teorema

Seja A um alfabeto finito e seja L uma linguagem sobre A.

1. Se $\mathcal{E}$ é um autómato de pilha tal que $L = L(\mathcal{E})$, então existe um autómato de pilha $\mathcal{E}'$ tal que $L = L_{pv}(\mathcal{E}')$.
2. Se $\mathcal{E}'$ é um autómato de pilha tal que $L = L_{pv}(\mathcal{E}')$, então existe um autómato de pilha $\mathcal{E}$ tal que $L = L(\mathcal{E})$.

Exemplo

Como vimos atrás, a linguagem $L_2 = \{ww^I : w \in \{a,b\}^*\}$ é reconhecida pelo AP $\mathcal{E}_2$ seguinte, usando o critério dos estados finais.

$\mathcal{E}_2 = \big(\{1,2,3\},\{a,b\},\{z,a,b\},\delta,1,z,\{3\}\big)$

$$\begin{aligned}
\delta(1,\epsilon,z) &= \{(2,z)\}\\
\delta(1,a,z) &= \{(1,az)\}\\
\delta(1,b,z) &= \{(1,bz)\}\\
\delta(1,a,b) &= \{(1,ab)\}\\
\delta(1,b,a) &= \{(1,ba)\}\\
\delta(1,a,a) &= \{(1,aa),(2,\epsilon)\}\\
\delta(1,b,b) &= \{(1,bb),(2,\epsilon)\}\\
\delta(2,a,a) &= \{(2,\epsilon)\}\\
\delta(2,b,b) &= \{(2,\epsilon)\}\\
\delta(2,\epsilon,z) &= \{(3,z)\}
\end{aligned}$$

$\mathcal{E}_2' = \big(\{1,2\},\{a,b\},\{z,a,b\},\delta',1,z,\emptyset\big)$

$$\begin{aligned}
\delta'(1,\epsilon,z) &= \{(2,z)\}\\
\delta'(1,a,z) &= \{(1,az)\}\\
\delta'(1,b,z) &= \{(1,bz)\}\\
\delta'(1,a,b) &= \{(1,ab)\}\\
\delta'(1,b,a) &= \{(1,ba)\}\\
\delta'(1,a,a) &= \{(1,aa),(2,\epsilon)\}\\
\delta'(1,b,b) &= \{(1,bb),(2,\epsilon)\}\\
\delta'(2,a,a) &= \{(2,\epsilon)\}\\
\delta'(2,b,b) &= \{(2,\epsilon)\}\\
\delta'(2,\epsilon,z) &= \{(2,\epsilon)\}
\end{aligned}$$

O AP $\mathcal{E}_2'$ reconhece L_2 pelo critério da pilha vazia.

O próximo objetivo é estabelecer:

Teorema

Uma linguagem é independente de contexto se e só se é reconhecida por um autómato de pilha.

Ou seja, pretende-se estabelecer a correspondência entre gramáticas independentes de contexto e autómatos de pilha:

- para cada GIC $\mathcal{G}$, define-se um AP $\mathcal{E}$ tal que $L_{pv}(\mathcal{E}) = L(\mathcal{G})$;
- para cada AP $\mathcal{E}$, define-se uma GIC $\mathcal{G}$ tal que $L(\mathcal{G}) = L_{pv}(\mathcal{E})$.

Tal será atingido adiante, com as duas proposições que concluem estes slides.

Como obter um AP que reconheça a linguagem gerada por uma GIC?

Seja $\mathcal{G} = (V, A, \mathcal{S}, P)$ uma GIC que gera uma linguagem L.

Pretende-se construir um autómato de pilha $\mathcal{E}$ cujas computações simulem derivações em $\mathcal{G}$. Para isso, bastará um único estado i e devem ser possíveis dois tipos de computações:

1. tirar um símbolo terminal do topo da pilha e se for igual ao símbolo lido na fita de leitura, avançar na fita de leitura.

$$(i, au, a) \underset{\mathcal{E}}{\longrightarrow} (i, u, \epsilon);$$

2. tirar uma variável $\mathcal{B}$ do topo da pilha e, se existir uma produção do tipo $\mathcal{B} \to \alpha$, substituir por α, sem avançar na fita de leitura,

$$(i, au, \mathcal{B}) \underset{\mathcal{E}}{\longrightarrow} (i, au, \alpha).$$

Exemplo

Seja $\mathcal{G} = (\{\mathcal{S},\mathcal{B},\mathcal{C},\mathcal{D}\},\{0,\ldots,9,(,),+,\times\},\mathcal{S},P)$ a GIC com produções

$$\begin{aligned}
\mathcal{S} &\to (\mathcal{S}) \mid \mathcal{S}+\mathcal{B} \mid \mathcal{B}\\
\mathcal{B} &\to \mathcal{B}\times\mathcal{C} \mid \mathcal{C}\\
\mathcal{C} &\to \mathcal{D}\\
\mathcal{D} &\to 0 \mid 1 \mid \ldots \mid 9
\end{aligned}$$

A seguinte derivação em $\mathcal{G}$,

$$\mathcal{S} \underset{\mathcal{G}}{\Rightarrow} (\mathcal{S}) \underset{\mathcal{G}}{\Rightarrow} (\mathcal{B}) \underset{\mathcal{G}}{\Rightarrow} (\mathcal{B}\times\mathcal{C}) \underset{\mathcal{G}}{\Rightarrow} (\mathcal{C}\times\mathcal{C}) \underset{\mathcal{G}}{\Rightarrow} (\mathcal{D}\times\mathcal{C}) \underset{\mathcal{G}}{\Rightarrow} (3\times\mathcal{C}) \underset{\mathcal{G}}{\Rightarrow} (3\times\mathcal{D}) \underset{\mathcal{G}}{\Rightarrow} (3\times 8)$$

pode ser simulada pela seguinte computação em $\mathcal{E}$,

$$(i,(3\times 8),\mathcal{S}) \underset{\mathcal{E}}{\longrightarrow} (i,(3\times 8),(\mathcal{S})) \underset{\mathcal{E}}{\longrightarrow} (i,3\times 8),\mathcal{S})) \underset{\mathcal{E}}{\longrightarrow} (i,3\times 8),\mathcal{B})) \underset{\mathcal{E}}{\longrightarrow} (i,3\times 8),\mathcal{B}\times\mathcal{C})) \underset{\mathcal{E}}{\longrightarrow} (i,3\times 8),\mathcal{C}\times\mathcal{C})) \underset{\mathcal{E}}{\longrightarrow} (i,3\times 8),\mathcal{D}\times\mathcal{C})) \underset{\mathcal{E}}{\longrightarrow} (i,3\times 8),3\times\mathcal{C})) \underset{\mathcal{E}}{\longrightarrow} (i,\times 8),\times\mathcal{C})) \underset{\mathcal{E}}{\longrightarrow} (i,8),\mathcal{C})) \underset{\mathcal{E}}{\longrightarrow} (i,8),\mathcal{D})) \underset{\mathcal{E}}{\longrightarrow} (i,8),8)) \underset{\mathcal{E}}{\longrightarrow} (i,),)) \underset{\mathcal{E}}{\longrightarrow} (i,\epsilon,\epsilon).$$

Proposição

Seja $\mathcal{G} = (V, A, \mathcal{S}, P)$ uma GIC. Então existe um autómato de pilha $\mathcal{E}$ tal que $L_{pv}(\mathcal{E}) = L(\mathcal{G})$.

Demonstração: Seja $\mathcal{E} = (\{i\}, A, V \cup A, \delta, i, \mathcal{S}, \emptyset)$ o AP cuja função transição é dada por:

$$\begin{array}{ll} \delta(i, \epsilon, \mathcal{B}) = \{(i, \alpha) \ : \ \mathcal{B} \to \alpha \in P\} & \text{para qualquer } \mathcal{B} \in V \\ \delta(i, a, a) = \{(i, \epsilon)\} & \text{para qualquer } a \in A \\ \delta(i, a, x) = \emptyset & \text{nos restantes casos.} \end{array}$$

Para quaisquer $\mathcal{B} \in V$ e $u \in A^*$, prova-se que,

$$\mathcal{B} \overset{+}{\underset{\mathcal{G}}{\Rightarrow}} u \quad \text{se e só se} \quad (i, u, \mathcal{B}) \xrightarrow[\mathcal{E}]{+} (i, \epsilon, \epsilon).$$

Em particular,

$$\mathcal{S} \overset{+}{\underset{\mathcal{G}}{\Rightarrow}} u \quad \text{se e só se} \quad (i, u, \mathcal{S}) \xrightarrow[\mathcal{E}]{+} (i, \epsilon, \epsilon),$$

ou seja, $L_{pv}(\mathcal{E}) = L(\mathcal{G})$.

Proposição

Seja $\mathcal{E} = (Q, A, X, \delta, i, z, \emptyset)$ um autómato de pilha. Então existe uma GIC $\mathcal{G}$ tal que $L(\mathcal{G}) = L_{pv}(\mathcal{E})$.

Demonstração: Seja $\mathcal{G} = (V, A, \mathcal{S}, P)$ a gramática em que:

1. $\mathcal{S}$ é um novo símbolo;
2. $V = \{\mathcal{S}\} \cup \{[p, x, q] \ : \ p, q \in Q, \ x \in X\}$;
3. P é constituído pelas produções:

$$\begin{aligned}
\mathcal{S} &\to [i, z, q], \qquad \text{para todo } q \in Q;\\
[p, x, q] &\to a, \quad \text{para todo } (q, \epsilon) \in \delta(p, a, x)\\
&\qquad\qquad (\text{com } p, q \in Q, x \in X, a \in A \cup \{\epsilon\});\\
[p, x, p_{m+1}] &\to a[q, x_1, p_2][p_2, x_2, p_3] \ldots [p_m, x_m, p_{m+1}],\\
&\qquad\qquad \text{para todo } (q, x_1 \ldots x_m) \in \delta(p, a, x)\\
&\qquad\qquad (\text{com } m \geq 1, \ a \in A \cup \{\epsilon\}, x_1, \ldots x_m \in X, p, q \in Q),\\
&\qquad\qquad \text{para todo } p_2, ..., p_{m+1} \in Q.
\end{aligned}$$

A demonstração ficaria concluída com a verificação de que $\mathcal{G}$ gera a linguagem reconhecida pelo autómato $\mathcal{E}$.