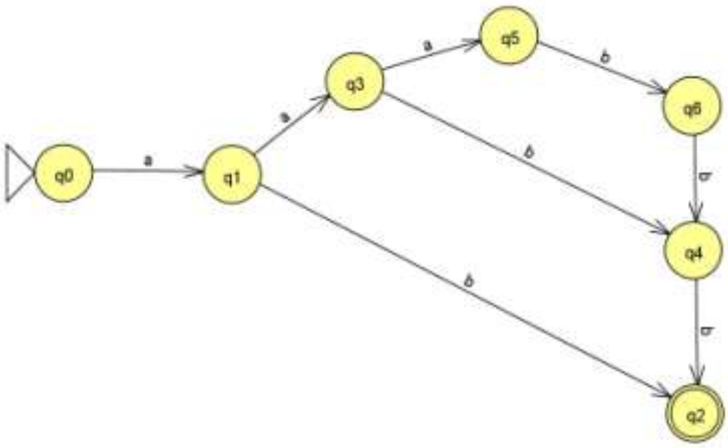
Exercício: faça um AFD ou AFND para a seguinte linguagem:

L = $\{a^n b^n \ / \ n > 0\}$

Para esta linguagem devemos ter quantidade n de a’s e a mesma quantidade n de b’s, sendo que as palavras que fazem parte da linguagem são = {ab, aabb, aaabbb, aaaabbbb, ... }.

Poderíamos começar o desenvolvimento do AFD para a linguagem acima tal como mostra a Figura abaixo, porém como temos uma linguagem com infinitos símbolos não é possível desenvolver o AFD pois o AFD não tem memória para guardar a quantidade de a da primeira metade da palavra e repetir na segunda metade com o símbolo b.



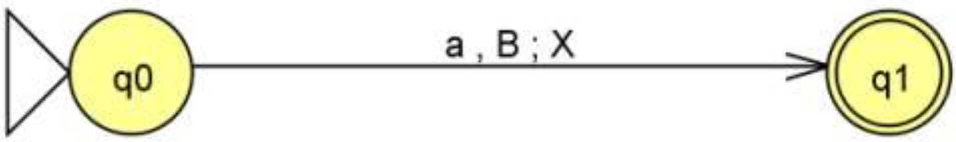
Como podemos ver este AFD não teria fim para aceitar todas as palavras da linguagem. Isto quer dizer que não é possível fazer um AFD para a linguagem acima, pois esta linguagem é de um Tipo mais complexo, que precisa ter uma “memória” para guardar uma quantidade n de símbolos.

As linguagens reconhecidas por um AFD ou AFND podem ser descritas como Linguagens Regulares, do tipo 3, como visto na Hierarquia de Chomsky. A linguagem $\{a^n b^n \ / \ n > 0\}$ pode ser descrita como uma Linguagem Livre de Contexto, do Tipo 2. Para estes tipos de linguagens precisamos da memória auxiliar e neste caso é utilizada uma pilha como memória auxiliar. Os mecanismos reconhecedores das Linguagens Livres de Contextos são os Autômatos a Pilha (AP)

Autômato a Pilha

Um autômato a pilha é similar a um AFD, com a diferença que em cada transição algo é retirado do topo da pilha e algo é inserido na pilha. Para processar e aceitar uma palavra no AP podem ter dois tipos, um AP que aceita quando processa toda a palavra e a pilha está vazia, não necessitando de estado final, e outro tipo de AP que aceita por estado final. Aqui veremos o autômato a pilha que aceita por pilha vazia.

O autômato a pilha usa uma estrutura de pilha como memória auxiliar, podendo inserir ou remover elementos na pilha. O processamento de uma palavra funciona de modo similar ao processamento no AFD, começando do símbolo inicial de uma palavra e lendo um símbolo por vez. A representação gráfica do autômato a pilha é representada como a figura abaixo, a representação de estado inicial e estado final são as mesmas de um AFD ou AFND, tendo como diferença a representação de uma transição, na qual são representadas três símbolos, o primeiro símbolo (a) representa o símbolo do alfabeto lido da palavra que está sendo processada. O segundo símbolo (B) representa o elemento que está no topo da pilha, e este elemento é retirado da pilha, e o terceiro elemento (X) é inserido na pilha.

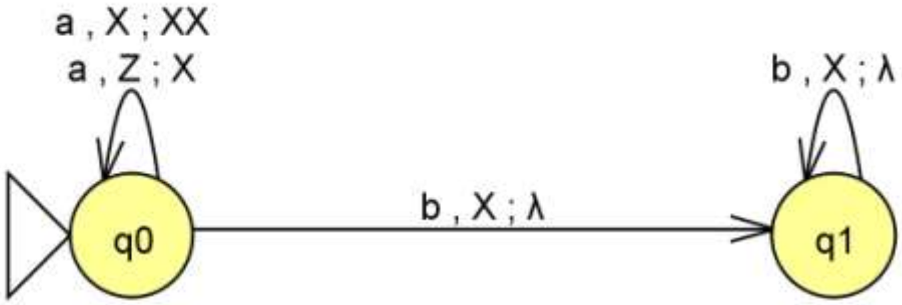


Para o autômato a pilha que reconhece uma palavra por pilha vazia, a descrição formal do autômato é formada por seis elementos, sendo eles $\{E, \Sigma, \delta, i, B, \Gamma\}$, e a representação de cada elemento é:

- E é o conjunto finito de estados
- Σ é o conjunto finito de símbolos alfabeto
- δ é a função de transição, tal que δ: $E \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow E \times \Gamma^*$
- i é o estado inicial ($i \in E$)
- B é a base da pilha (Base $\in \Gamma$)
- Γ são os símbolos auxiliares da pilha

No autômato a pilha o processamento de uma palavra, a estrutura de pilha utilizada no autômato a pilha começa com o símbolo Base da descrição formal na pilha, sendo o único elemento na pilha. O elemento Γ da descrição formal representa os símbolos auxiliares da pilha, que são os elementos que podem ser inseridos/retirados da pilha para auxiliar no processamento.

Para as linguagens livres de contexto, um exemplo a L = {$a^n b^n$ / n > 0}, no processamento por um autômato com pilha, o que pode ser utilizado na pilha é que para cada “a” processado em uma palavra é inserido um elemento na pilha (X), e para que a quantidade de “b” seja a mesma de “a”, cada “b” processado na palavra retirado o elemento inserido pelo “a”, no caso o X.



No exemplo acima, temos um AP com Z como base da pilha, e neste autômato temos quatro transições, a primeira δ(q0,a,Z) = (q0, X), que representa que se processarmos um “a” estando no estado inicial q0, e com o elemento Z no topo da pilha, este elemento Z é retirado e colocado um X no lugar dele. Para a segunda transição δ(q0,a,X) = (q0, XX) indica que estando no estado q0, processando um “a” e com o elemento X no topo da pilha, esse elemento X é retirado e inserido o elemento X duas vezes na pilha.

Para a terceira transição, δ(q0,b,X) = (q1, λ), temos o processamento de um “b” estando no estado q0, e com o elemento X no topo da pilha, este elemento X é retirado e é inserido o λ no lugar dele, ou seja, quando temos λ no terceiro símbolo de um transição quer dizer que o segundo símbolo (X) é retirado e não é inserido nenhum elemento no lugar dele. E para a quarta transição δ(q1,b,X) = (q1, λ), estando no estado q1, processando “b” e com X no topo da pilha, continuamos no estado q1, retiramos o elemento X do topo da pilha, e não inserimos nenhum elemento na pilha.

Este autômato representa a linguagem L = {$a^n b^n$ / n > 0} e o funcionamento para reconhecer as palavras destas linguagens é tal que para cada “a” processado é inserido um elemento X na pilha, para termos a quantidade igual de “a” e “b”, cada “b” processado retira o elemento inserido na pilha (X). Como para aceitar a palavra devemos terminar com a pilha vazia, se a pilha finalizar com vazia quer dizer que a quantidade de “a” e “b” são iguais e a palavra faz parte da linguagem.

Assim, para o primeiro “a” processado em uma palavra, teremos Z como base na pilha, e este símbolo é trocado por X. Para cada próximo “a” processado em uma palavra, teremos X como elemento no topo da pilha, este X é retirado da pilha e inserido dois XX, ou seja, para cada próximo “a” é inserido um X na pilha. Do mesmo modo, o primeiro “b” processado, estaremos no estado q0, e com X na pilha, sairemos de q0 e iremos para o estado q1 e retiramos um X do topo da pilha, e sempre retirando X da pilha ao processar um próximo “b”.

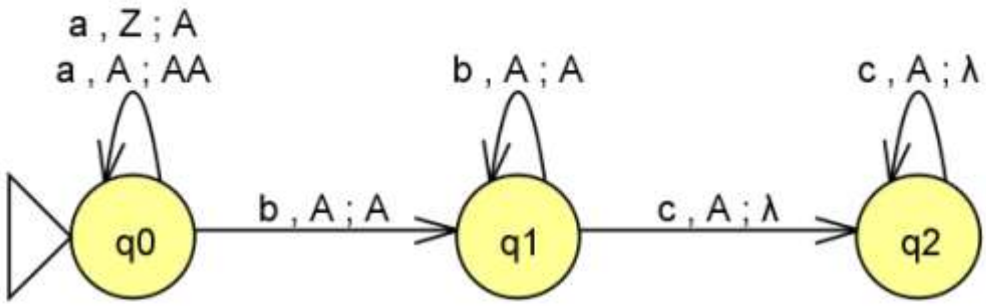
A descrição formal do autômato para a linguagem = $\{a^n b^n \ / \ n > 0\}$ segue como descrito:

- E = {q0, q1}
- Σ = {a, b}
- i = q0
- Γ = {Z, X}
- Base = Z
- Função de transição
    - δ(q0, a, Z) = (q0, X)
    - δ(q0, a, X) = (q0, XX)
    - δ(q0, b, X) = (q1, λ)
    - δ(q1, b, X) = (q1, λ)

Para a função de transição, em vez de representarmos como uma tabela usamos a representação acima que indica do lado esquerdo da igualdade, qual estado estamos, o símbolo do alfabeto a ser processado e o que está no topo da pilha, que é retirado; e do lado direito da igualdade, para qual estado é a transição e o que é inserido na pilha

Ao desenvolver um AP, o que podemos pensar para resolução de um exercício é qual símbolo irá inserir na pilha e qual símbolo irá retirar da pilha e em qual quantidade inserir ou retirar, por exemplo, $\{a^n b^m c^n \ / \ n > 0, m > 0\}$, como devemos ter a quantidade de “a” igual a quantidade de “c”, podemos pensar em um autômato a pilha no qual cada “a” processado insere um símbolo na pilha, e cada “c” processado retira o símbolo inserido na pilha. Entre os símbolos “a” e “c” temos uma quantidade m de “b”, para este AP ao processar o símbolo “b” não iremos inserir nada na pilha, nem retirar. Isto é feito repetindo o símbolo em uma transição, por exemplo, para a transição δ(q0, b, X) = (q1, X) é processada quando o estado ativo é o q0, processamos o símbolo “b” e temos X no topo da pilha. Este símbolo X do topo da pilha é removido, e no lugar dele colocamos X, ou seja, não alteramos nada na pilha e, mudamos de estado ativo de q0 para q1.

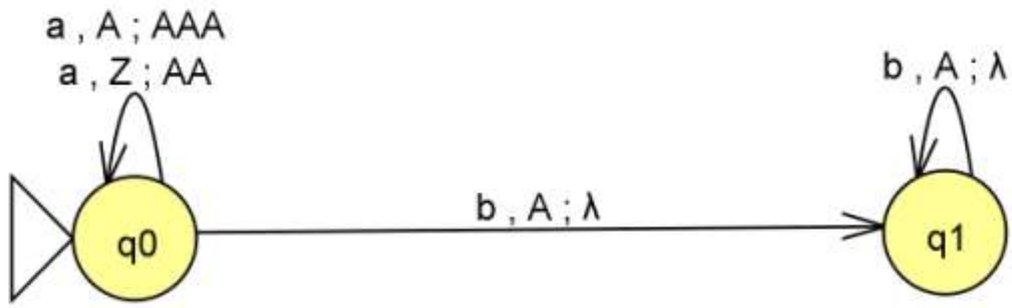
Para a linguagem = $\{a^n b^m c^n \ / \ n > 0, m > 0\}$, o autômato a pilha teria a seguinte representação gráfica e descrição formal:



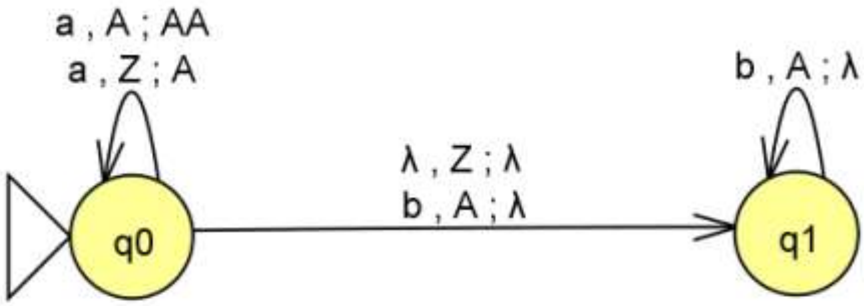
A descrição formal do autômato acima pode ser representada por:

- E = {q0, q1, q2}
- Σ = {a, b, c}
- i = q0
- Γ = {Z, A}
- Base = Z
- Função de transição
  - δ(q0, a, Z) = (q0, A)
  - δ(q0, a, A) = (q0, AA)
  - δ(q0, b, A) = (q1, A)
  - δ(q1, b, A) = (q1, A)
  - δ(q1, c, A) = (q2, λ)
  - δ(q2, c, A) = (q2, λ)

Para a linguagem $\{a^n b^{2n} / n > 0\}$ as seguintes palavras fazem parte {abb, aabbbb, aaabbbbbb, ...}, ou seja, para cada "a" no início da palavra, temos o dobro de "b" no final da palavra. Se utilizássemos a ideia dos exercícios anteriores na qual cada "a" processado insere um elemento na pilha e cada "b" processado remove o elemento da pilha somente iremos garantir que a quantidade de "b" é igual a quantidade de "a". Assim podemos inserir mais de um elemento na pilha quando necessário. Para a linguagem acima, se para cada "a" processado for inserido dois símbolos na pilha, em uma palavra qualquer da linguagem com quantidade n de "a", teríamos 2n símbolos na pilha, sendo que cada "b" retira um símbolo da pilha, terminando assim com a pilha vazia quando a palavra pertence a linguagem. O autômato com pilha para a linguagem acima pode ser visto abaixo.



Quando a palavra vazia pertence a uma linguagem temos um caso especial para realizar o processamento desta palavra. Como em um AP, a pilha inicia-se com o elemento base nela, e para aceitar uma palavra devemos processa-la e a pilha deve terminar vazia, devemos ter uma transição para processar a palavra vazia, retirando assim o elemento base da pilha e não inserindo nada na pilha. Isto é realizado por meio da transição $\delta(q0, \lambda, Z) = (q1, \lambda)$ no autômato abaixo, representando que a partir do estado inicial (q0), processando a palavra vazia λ, e estando o Z (base da pilha) no topo, este símbolo é retirado e é inserido λ na pilha, ou seja, não insere nada.



O autômato acima reconhece a linguagem $\{a^n b^n \ / \ n \geq 0\}$, processando a palavra vazia e retirando o símbolo base Z da pilha.

Essa transição com a palavra vazia é representada na função de transição de um AP, na qual, $E \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow E \times \Gamma^*$, que indica que a partir de um estado pertencente ao conjunto de estados E, processando um símbolo do conjunto de símbolos do alfabeto, ou a palavra vazia, e com algo do topo da pilha, pertencente ao alfabeto da pilha, que é retirado, vai para um estado do conjunto de estados e inserimos algo na pilha (inclusive nenhum elemento)

Bibliografia

VIEIRA, NEWTON JOSÉ. **Introdução aos Fundamentos da Computação**. São Paulo. Pioneira Thomson Learning. 2006;

MENEZES, P. B.; DIVERIO, T. A.; **Teoria da Computação: Máquinas Universais e Computabilidade**; 3ª edição Bookman 2011

MENEZES, P. B. ; **Linguagens Formais e Autômatos**. 6ª edição. Ed. Artmed. 2011

SIPSER M. **Introdução à Teoria da Computação**. 2 ed. Cengage Learning.2007

MORET, B. M. **"Theory of Computation".** Addison-Wesley, 1998.

HOPCROFT, JOHN E.; ULLMAN, JEFFREY D.; MOTWANI, RAJEEV **Introdução à Teoria de Autômatos, Linguagens e Computação** Ed.Campus 2002

SILVA, FLAVIO SOARES CORRÊA; MELO, ANA CRISTINA VIEIRA; **Modelos Clássicos de Computação** Ed. Thomson 1ª Edição 2006