Fundamentos Teóricos da Computação

– Autômato de Pilha –

Sumário

Introdução à Autômatos de Pilha

Arquitetura de AF

Principais componentes da arquitetura de um AF:

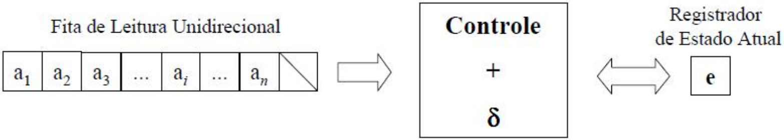
- Fita de Leitura Unidirecional
- Controle + Função de Transição
- Registrador de Estado Atual

Introdução à Autômatos de Pilha

Arquitetura de AF

Principais componentes da arquitetura de um AF:

- Fita de Leitura Unidirecional
- Controle + Função de Transição
- Registrador de Estado Atual



Introdução à Autômatos de Pilha

Arquitetura de AP

Principais componentes da arquitetura de um AP:

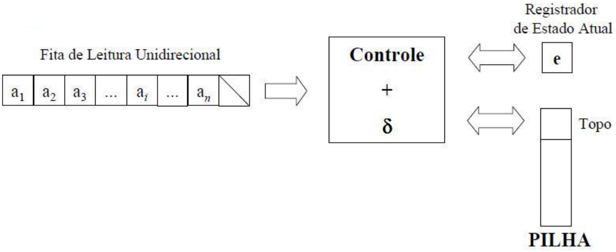
- Fita de Leitura Unidirecional
- Controle + Função de Transição
- Registrador de Estado Atual
- Pilha

Introdução à Autômatos de Pilha

Arquitetura de AP

Principais componentes da arquitetura de um AP:

- Fita de Leitura Unidirecional
- Controle + Função de Transição
- Registrador de Estado Atual
- Pilha



Autômatos de Pilha

Autômato de Pilha Determinístico

Um **Autômato de Pilha Determinístico** (APD) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto E \times \Gamma^*$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ não deve possuir transições compatíveis !!!

Autômatos de Pilha

Autômato de Pilha Determinístico

Um **Autômato de Pilha Determinístico** (APD) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto E \times \Gamma^*$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ **não deve possuir transições compatíveis !!!**

Autômatos de Pilha

Autômato de Pilha Determinístico

Um **Autômato de Pilha Determinístico** (APD) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto E \times \Gamma^*$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ **não deve possuir transições compatíveis !!!**

Autômatos de Pilha

Autômato de Pilha Determinístico

Um **Autômato de Pilha Determinístico** (APD) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto E \times \Gamma^*$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ não deve possuir transições compatíveis !!!

Autômatos de Pilha

Autômato de Pilha Determinístico

Um **Autômato de Pilha Determinístico** (APD) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto E \times \Gamma^*$$

- $i \equiv$ estado inicial $(i \in E)$
- $F \equiv$ conjunto de estados finais $(F \subseteq E)$

OBS: δ não deve possuir transições compatíveis !!!

Autômatos de Pilha

Autômato de Pilha Determinístico

Um **Autômato de Pilha Determinístico** (APD) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto E \times \Gamma^*$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ não deve possuir transições compatíveis !!!

Autômatos de Pilha

Autômato de Pilha Determinístico

Um **Autômato de Pilha Determinístico** (APD) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto E \times \Gamma^*$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ não deve possuir transições compatíveis !!!

Autômatos de Pilha

Autômato de Pilha Determinístico

Um **Autômato de Pilha Determinístico** (APD) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto E \times \Gamma^*$$

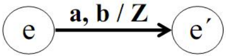
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ não deve possuir transições compatíveis !!!

Autômatos de Pilha

Representação de uma Transição

Uma transição do estado e para o estado e' quando o conteúdo da fita for a e o topo da pilha for b – desempilha b e empilha Z e pode ser representada da seguinte forma: $\delta(e, a, b) = [e', Z]$



Se $a = \lambda$, o símbolo da entrada não é consumido e a transição é dita vazia (transição λ).

Se $b = \lambda$, a transição ocorre sem consulta à pilha (e nada é desempilhado).

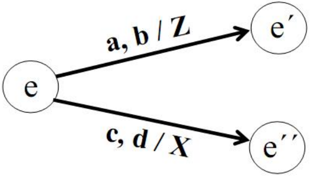
Se $Z = \lambda$, nada será empilhado.

Autômatos de Pilha

Transição Compatível

Seja uma função de transição $\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto E \times \Gamma^*$. Duas transições $\delta(e, a, b)$ e $\delta(e, c, d)$ são ditas compatíveis se, e somente se:

$$(a = c \text{ ou } a = \lambda \text{ ou } c = \lambda) \text{ e } (b = d \text{ ou } b = \lambda \text{ ou } d = \lambda).$$



Duas transições $\delta(e, a, b)$ e $\delta(e, c, d)$ são **não** compatíveis se, e somente se:

$$(a \neq c \text{ e } a \neq \lambda \text{ e } c \neq \lambda) \text{ ou } (b \neq d \text{ e } b \neq \lambda \text{ e } d \neq \lambda).$$

Autômatos de Pilha

Exemplo de Representação Gráfica de APD

Utilizando-se do lema do bombeamento, pode-se mostrar que o conjunto $\{a^n b^n \mid n \geq 0\}$ não é uma linguagem regular.

Contudo, ela pode ser reconhecida pelo APD

$$M = (\{q_0, q_1\}, \{a, b\}, \{X\}, \delta, q_0, \{q_0, q_1\}),$$

em que δ é dada por:

1. $\delta(q_0, a, \lambda) = [q_0, X]$;
2. $\delta(q_0, b, X) = [q_1, \lambda]$;
3. $\delta(q_1, b, X) = [q_1, \lambda]$.

Representação Gráfica

Autômatos de Pilha

Exemplo de Representação Gráfica de APD

Utilizando-se do lema do bombeamento, pode-se mostrar que o conjunto $\{a^n b^n \mid n \geq 0\}$ não é uma linguagem regular.

Contudo, ela pode ser reconhecida pelo APD

$$M = (\{q_0, q_1\}, \{a, b\}, \{X\}, \delta, q_0, \{q_0, q_1\}),$$

em que δ é dada por:

1. $\delta(q_0, a, \lambda) = [q_0, X]$; $\Rightarrow$ **armazena uma "lembrança" para cada "a"**
2. $\delta(q_0, b, X) = [q_1, \lambda]$; $\Rightarrow$ **retira uma "lembrança" para o 1º "b"**
3. $\delta(q_1, b, X) = [q_1, \lambda]$. $\Rightarrow$ **retira uma "lembrança" para cada "b"**

Representação Gráfica

Autômatos de Pilha

Exemplo de Representação Gráfica de APD

Utilizando-se do lema do bombeamento, pode-se mostrar que o conjunto $\{a^n b^n \mid n \geq 0\}$ não é uma linguagem regular.

Contudo, ela pode ser reconhecida pelo APD

$$M = (\{q_0, q_1\}, \{a, b\}, \{X\}, \delta, q_0, \{q_0, q_1\}),$$

em que δ é dada por:

1. $\delta(q_0, a, \lambda) = [q_0, X]$; $\Rightarrow$ **armazena uma "lembrança" para cada 'a'**
2. $\delta(q_0, b, X) = [q_1, \lambda]$; $\Rightarrow$ **retira uma "lembrança" para o 1º "b"**
3. $\delta(q_1, b, X) = [q_1, \lambda]$. $\Rightarrow$ **retira uma "lembrança" para cada "b"**

Representação Gráfica

Autômatos de Pilha

Exemplo de Representação Gráfica de APD

Utilizando-se do lema do bombeamento, pode-se mostrar que o conjunto $\{a^n b^n \mid n \geq 0\}$ não é uma linguagem regular.

Contudo, ela pode ser reconhecida pelo APD

$$M = (\{q_0, q_1\}, \{a, b\}, \{X\}, \delta, q_0, \{q_0, q_1\}),$$

em que δ é dada por:

1. $\delta(q_0, a, \lambda) = [q_0, X]$; $\Rightarrow$ **armazena uma "lembrança" para cada 'a'**
2. $\delta(q_0, b, X) = [q_1, \lambda]$; $\Rightarrow$ retira uma "lembrança" para o 1º 'b'
3. $\delta(q_1, b, X) = [q_1, \lambda]$. $\Rightarrow$ retira uma "lembrança" para cada 'b'

Representação Gráfica

Autômatos de Pilha

Exemplo de Representação Gráfica de APD

Utilizando-se do lema do bombeamento, pode-se mostrar que o conjunto $\{a^n b^n \mid n \geq 0\}$ não é uma linguagem regular.

Contudo, ela pode ser reconhecida pelo APD

$$M = (\{q_0, q_1\}, \{a, b\}, \{X\}, \delta, q_0, \{q_0, q_1\}),$$

em que δ é dada por:

1. $\delta(q_0, a, \lambda) = [q_0, X]$; $\Rightarrow$ **armazena uma "lembrança" para cada 'a'**
2. $\delta(q_0, b, X) = [q_1, \lambda]$; $\Rightarrow$ retira uma "lembrança" para o 1º 'b'
3. $\delta(q_1, b, X) = [q_1, \lambda]$.

Representação Gráfica

Autômatos de Pilha

Exemplo de Representação Gráfica de APD

Utilizando-se do lema do bombeamento, pode-se mostrar que o conjunto $\{a^n b^n \mid n \geq 0\}$ não é uma linguagem regular.

Contudo, ela pode ser reconhecida pelo APD

$$M = (\{q_0, q_1\}, \{a, b\}, \{X\}, \delta, q_0, \{q_0, q_1\}),$$

em que δ é dada por:

1. $\delta(q_0, a, \lambda) = [q_0, X]$; $\Rightarrow$ **armazena uma “lembrança” para cada 'a'**
2. $\delta(q_0, b, X) = [q_1, \lambda]$; $\Rightarrow$ **retira uma “lembrança” para o 1° 'b'**
3. $\delta(q_1, b, X) = [q_1, \lambda]$. $\Rightarrow$ retira uma “lembrança” para cada 'b'

Representação Gráfica

Autômatos de Pilha

Exemplo de Representação Gráfica de APD

Utilizando-se do lema do bombeamento, pode-se mostrar que o conjunto $\{a^n b^n \mid n \geq 0\}$ não é uma linguagem regular.

Contudo, ela pode ser reconhecida pelo APD

$$M = (\{q_0, q_1\}, \{a, b\}, \{X\}, \delta, q_0, \{q_0, q_1\}),$$

em que δ é dada por:

1. $\delta(q_0, a, \lambda) = [q_0, X]$; $\Rightarrow$ **armazena uma "lembrança" para cada 'a'**
2. $\delta(q_0, b, X) = [q_1, \lambda]$; $\Rightarrow$ **retira uma "lembrança" para o 1° 'b'**
3. $\delta(q_1, b, X) = [q_1, \lambda]$. $\Rightarrow$ retira uma "lembrança" para cada 'b'

Representação Gráfica

Autômatos de Pilha

Exemplo de Representação Gráfica de APD

Utilizando-se do lema do bombeamento, pode-se mostrar que o conjunto $\{a^n b^n \mid n \geq 0\}$ não é uma linguagem regular.

Contudo, ela pode ser reconhecida pelo APD

$$M = (\{q_0, q_1\}, \{a, b\}, \{X\}, \delta, q_0, \{q_0, q_1\}),$$

em que δ é dada por:

1. $\delta(q_0, a, \lambda) = [q_0, X]$; $\Rightarrow$ **armazena uma "lembrança" para cada 'a'**
2. $\delta(q_0, b, X) = [q_1, \lambda]$; $\Rightarrow$ **retira uma "lembrança" para o 1° 'b'**
3. $\delta(q_1, b, X) = [q_1, \lambda]$. $\Rightarrow$ **retira uma "lembrança" para cada 'b'**

Representação Gráfica

Autômatos de Pilha

Exemplo de Representação Gráfica de APD

Utilizando-se do lema do bombeamento, pode-se mostrar que o conjunto $\{a^n b^n \mid n \geq 0\}$ não é uma linguagem regular.

Contudo, ela pode ser reconhecida pelo APD

$$M = (\{q_0, q_1\}, \{a, b\}, \{X\}, \delta, q_0, \{q_0, q_1\}),$$

em que δ é dada por:

1. $\delta(q_0, a, \lambda) = [q_0, X]$; $\Rightarrow$ **armazena uma "lembrança" para cada 'a'**
2. $\delta(q_0, b, X) = [q_1, \lambda]$; $\Rightarrow$ **retira uma "lembrança" para o 1° 'b'**
3. $\delta(q_1, b, X) = [q_1, \lambda]$. $\Rightarrow$ **retira uma "lembrança" para cada 'b'**

Representação Gráfica

Autômatos de Pilha

Exemplo de Representação Gráfica de APD

Utilizando-se do lema do bombeamento, pode-se mostrar que o conjunto $\{a^n b^n \mid n \geq 0\}$ não é uma linguagem regular.

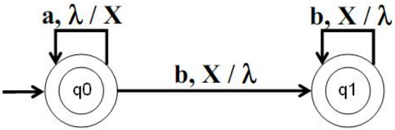
Contudo, ela pode ser reconhecida pelo APD

$$M = (\{q_0, q_1\}, \{a, b\}, \{X\}, \delta, q_0, \{q_0, q_1\}),$$

em que δ é dada por:

1. $\delta(q_0, a, \lambda) = [q_0, X]$; $\Rightarrow$ **armazena uma "lembrança" para cada 'a'**
2. $\delta(q_0, b, X) = [q_1, \lambda]$; $\Rightarrow$ **retira uma "lembrança" para o 1° 'b'**
3. $\delta(q_1, b, X) = [q_1, \lambda]$. $\Rightarrow$ **retira uma "lembrança" para cada 'b'**

Representação Gráfica



Autômatos de Pilha

Configuração Instantânea de AP

A configuração instantânea de um AP é dada pela tripla $[e, w, p]$ em que:

- $e \equiv$ estado atual;
- $w \equiv$ palavra de entrada (ou o que restar durante as transições); e
- $p \equiv$ conteúdo da pilha (topo $\rightarrow$ fundo).

Seja um APD $M = (E, \Sigma, \Gamma, \delta, i, F)$. A relação $\vdash \subseteq (E \times \Sigma^* \times \Gamma^*)^2$, para M, é tal que todo $e, e' \in E$, $a \in \Sigma \cup \{\lambda\}$, $b \in \Gamma \cup \{\lambda\}$ e $w \in \Gamma^*$:

$$[e, ay, bz] \vdash [e', y, wz], \forall y \in \Sigma^*, z \in \Gamma^* \iff \delta(e, a, b) = [e', w].$$

Linguagem de um APD

Seja um APD $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M é

$$L(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, \lambda] \text{ para algum } f \in F\}.$$

Uma palavra w tal que $[i, w, \lambda] \vdash^* [f, \lambda, \lambda]$, em que $f \in F$, é dita ser reconhecida (ou aceita) por M.

Autômatos de Pilha

Configuração Instantânea de AP

A configuração instantânea de um AP é dada pela tripla $[e, w, p]$ em que:

- $e \equiv$ estado atual;
- $w \equiv$ palavra de entrada (ou o que restar durante as transições); e
- $p \equiv$ conteúdo da pilha (topo $\rightarrow$ fundo).

Seja um APD $M = (E, \Sigma, \Gamma, \delta, i, F)$. A relação $\vdash \subseteq (E \times \Sigma^* \times \Gamma^*)^2$, para M, é tal que todo $e, e' \in E$, $a \in \Sigma \cup \{\lambda\}$, $b \in \Gamma \cup \{\lambda\}$ e $w \in \Gamma^*$:

$$[e, ay, bz] \vdash [e', y, wz], \forall y \in \Sigma^*, z \in \Gamma^* \iff \delta(e, a, b) = [e', w].$$

Linguagem de um APD

Seja um APD $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M é

$$L(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, \lambda] \text{ para algum } f \in F\}.$$

Uma palavra w tal que $[i, w, \lambda] \vdash^* [f, \lambda, \lambda]$, em que $f \in F$, é dita ser reconhecida (ou aceita) por M.

Autômatos de Pilha

Configuração Instantânea de AP

A configuração instantânea de um AP é dada pela tripla $[e, w, p]$ em que:

- $e \equiv$ estado atual;
- $w \equiv$ palavra de entrada (ou o que restar durante as transições); e
- $p \equiv$ conteúdo da pilha (topo $\rightarrow$ fundo).

Seja um APD $M = (E, \Sigma, \Gamma, \delta, i, F)$. A relação $\vdash \subseteq (E \times \Sigma^* \times \Gamma^*)^2$, para M, é tal que todo $e, e' \in E$, $a \in \Sigma \cup \{\lambda\}$, $b \in \Gamma \cup \{\lambda\}$ e $w \in \Gamma^*$:

$$[e, ay, bz] \vdash [e', y, wz], \forall y \in \Sigma^*, z \in \Gamma^* \iff \delta(e, a, b) = [e', w].$$

Linguagem de um APD

Seja um APD $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M é

$$L(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, \lambda] \text{ para algum } f \in F\}.$$

Uma palavra w tal que $[i, w, \lambda] \vdash^* [f, \lambda, \lambda]$, em que $f \in F$, é dita ser reconhecida (ou aceita) por M.

Autômatos de Pilha

Simulação de APD

algoritmo
  Entrada: APD $M = (E, \Sigma, \Gamma, \delta, i, F)$
    palavra de entrada w dada por $prox()$ e termina
    com EOS (= "End of Sequence")
  $Estado \leftarrow i$
  empilha("MarcaFundo")
  $Simbolo \leftarrow prox()$
  seja $a \in \{Simbolo, \lambda\}$ e $b \in \{topo(), \lambda\}$
  enquanto $\delta(Estado, a, b)$ for definida faça
    seja $\delta(Estado, a, b) = [Estado', z]$
    se $a \neq \lambda$ então
      $Simbolo \leftarrow prox()$
    fim se
    se $b \neq \lambda$ então
      desempilhe()
    fim se
    empilhe(z)
    $Estado \leftarrow Estado'$
  fim enquanto
  se $Simbolo = EOS$ e $topo() = \text{``}MarcaFundo''$ e $Estado \in F$ então
    retorne "Entrada foi aceita"
  senão
    retorne "Entrada não foi aceita"
  fim se
fim algoritmo

Autômatos de Pilha

Simulação de APD

```
algoritmo
   Entrada:  APD M = (E, Σ, Γ, δ, i, F)
             palavra de entrada w dada por prox() e termina
             com EOS (= "End of Sequence")
   Estado ← i
   empilha("MarcaFundo")
   Simbolo ← prox()
   seja a ∈ {Simbolo, λ} e b ∈ {topo(), λ}
   enquanto δ(Estado, a, b) for definida faça
      seja δ(Estado, a, b) = [Estado′, z]
      se a ≠ λ então
         Simbolo ← prox()
      fim se
      se b ≠ λ então
         desempilhe()
      fim se
      empilhe(z)
      Estado ← Estado′
   fim enquanto
   se Simbolo = EOS e topo() = "MarcaFundo″ e Estado ∈ F então
      retorne "Entrada foi aceita"
   senão
      retorne "Entrada não foi aceita"
   fim se
fim algoritmo
```

Autômatos de Pilha

Simulação de APD

algoritmo
  Entrada: APD $M = (E, \Sigma, \Gamma, \delta, i, F)$
          palavra de entrada w dada por $prox()$ e termina
          com EOS (= "End of Sequence")
  $Estado \leftarrow i$
  empilha("MarcaFundo")
  $Simbolo \leftarrow prox()$
  seja $a \in \{Simbolo, \lambda\}$ e $b \in \{topo(), \lambda\}$
  enquanto $\delta(Estado, a, b)$ for definida faça
    seja $\delta(Estado, a, b) = [Estado', z]$
    se $a \neq \lambda$ então
      $Simbolo \leftarrow prox()$
    fim se
    se $b \neq \lambda$ então
      desempilhe()
    fim se
    empilhe(z)
    $Estado \leftarrow Estado'$
  fim enquanto
  se $Simbolo = EOS$ e $topo() = ``MarcaFundo''$ e $Estado \in F$ então
    retorne "Entrada foi aceita"
  senão
    retorne "Entrada não foi aceita"
  fim se
fim algoritmo

Autômatos de Pilha

Simulação de APD

algoritmo
  Entrada: APD $M = (E, \Sigma, \Gamma, \delta, i, F)$
      palavra de entrada w dada por $prox()$ e termina
      com EOS (= "End of Sequence")
  $Estado \leftarrow i$
  empilha("MarcaFundo")
  $Simbolo \leftarrow prox()$
  seja $a \in \{Simbolo, \lambda\}$ e $b \in \{topo(), \lambda\}$
  enquanto $\delta(Estado, a, b)$ for definida faça
    seja $\delta(Estado, a, b) = [Estado', z]$
    se $a \neq \lambda$ então
      $Simbolo \leftarrow prox()$
    fim se
    se $b \neq \lambda$ então
      desempilhe()
    fim se
    empilhe(z)
    $Estado \leftarrow Estado'$
  fim enquanto
  se $Simbolo = EOS$ e $topo() = ``MarcaFundo''$ e $Estado \in F$ então
    retorne "Entrada foi aceita"
  senão
    retorne "Entrada não foi aceita"
  fim se
fim algoritmo

Autômatos de Pilha

Simulação de APD

algoritmo
  Entrada: APD $M = (E, \Sigma, \Gamma, \delta, i, F)$
    palavra de entrada w dada por $prox()$ e termina
    com EOS (= "End of Sequence")
  $Estado \leftarrow i$
  empilha("MarcaFundo")
  $Simbolo \leftarrow prox()$
  seja $a \in \{Simbolo, \lambda\}$ e $b \in \{topo(), \lambda\}$
  enquanto $\delta(Estado, a, b)$ for definida faça
    seja $\delta(Estado, a, b) = [Estado', z]$
    se $a \neq \lambda$ então
      $Simbolo \leftarrow prox()$
    fim se
    se $b \neq \lambda$ então
      desempilhe()
    fim se
    empilhe(z)
    $Estado \leftarrow Estado'$
  fim enquanto
  se $Simbolo = EOS$ e $topo() =$ "$MarcaFundo$" e $Estado \in F$ então
    retorne "Entrada foi aceita"
  senão
    retorne "Entrada não foi aceita"
  fim se
fim algoritmo

Autômatos de Pilha

Simulação de APD

algoritmo
  Entrada: APD $M = (E, \Sigma, \Gamma, \delta, i, F)$
    palavra de entrada w dada por $prox()$ e termina
    com EOS (= "End of Sequence")
  $Estado \leftarrow i$
  empilha("MarcaFundo")
  $Simbolo \leftarrow prox()$
  seja $a \in \{Simbolo, \lambda\}$ e $b \in \{topo(), \lambda\}$
  enquanto $\delta(Estado, a, b)$ for definida faça
    seja $\delta(Estado, a, b) = [Estado', z]$
    se $a \neq \lambda$ então
      $Simbolo \leftarrow prox()$
    fim se
    se $b \neq \lambda$ então
      desempilhe()
    fim se
    empilhe(z)
    $Estado \leftarrow Estado'$
  fim enquanto
  se $Simbolo = EOS$ e $topo() = "MarcaFundo''$ e $Estado \in F$ então
    retorne "Entrada foi aceita"
  senão
    retorne "Entrada não foi aceita"
  fim se
fim algoritmo

Autômatos de Pilha

Simulação de APD

```
algoritmo
   Entrada:  APD M = (E, Σ, Γ, δ, i, F)
             palavra de entrada w dada por prox() e termina
             com EOS (= "End of Sequence")
   Estado ← i
   empilha("MarcaFundo")
   Simbolo ← prox()
   seja a ∈ {Simbolo, λ} e b ∈ {topo(), λ}
   enquanto δ(Estado, a, b) for definida faça
      seja δ(Estado, a, b) = [Estado′, z]
      se a ≠ λ então
         Simbolo ← prox()
      fim se
      se b ≠ λ então
         desempilhe()
      fim se
      empilhe(z)
      Estado ← Estado′
   fim enquanto
   se Simbolo = EOS e topo() = "MarcaFundo" e Estado ∈ F então
      retorne "Entrada foi aceita"
   senão
      retorne "Entrada não foi aceita"
   fim se
fim algoritmo
```

Autômatos de Pilha

Simulação de APD

```
algoritmo
   Entrada:  APD M = (E, Σ, Γ, δ, i, F)
             palavra de entrada w dada por prox() e termina
             com EOS (= "End of Sequence")
   Estado ← i
   empilha("MarcaFundo")
   Simbolo ← prox()
   seja a ∈ {Simbolo, λ} e b ∈ {topo(), λ}
   enquanto δ(Estado, a, b) for definida faça
      seja δ(Estado, a, b) = [Estado′, z]
      se a ≠ λ então
         Simbolo ← prox()
      fim se
      se b ≠ λ então
         desempilhe()
      fim se
      empilhe(z)
      Estado ← Estado′
   fim enquanto
   se Simbolo = EOS e topo() = "MarcaFundo" e Estado ∈ F então
      retorne "Entrada foi aceita"
   senão
      retorne "Entrada não foi aceita"
   fim se
fim algoritmo
```

Autômatos de Pilha

Simulação de APD

```
algoritmo
   Entrada:  APD M = (E, Σ, Γ, δ, i, F)
             palavra de entrada w dada por prox() e termina
             com EOS (= "End of Sequence")
   Estado ← i
   empilha("MarcaFundo")
   Simbolo ← prox()
   seja a ∈ {Simbolo, λ} e b ∈ {topo(), λ}
   enquanto δ(Estado, a, b) for definida faça
      seja δ(Estado, a, b) = [Estado', z]
      se a ≠ λ então
         Simbolo ← prox()
      fim se
      se b ≠ λ então
         desempilhe()
      fim se
      empilhe(z)
      Estado ← Estado'
   fim enquanto
   se Simbolo = EOS e topo() = "MarcaFundo" e Estado ∈ F então
      retorne "Entrada foi aceita"
   senão
      retorne "Entrada não foi aceita"
   fim se
fim algoritmo
```

Autômatos de Pilha

Simulação de APD

```
algoritmo
   Entrada:  APD M = (E, Σ, Γ, δ, i, F)
             palavra de entrada w dada por prox() e termina
             com EOS (= "End of Sequence")
   Estado ← i
   empilha("MarcaFundo")
   Simbolo ← prox()
   seja a ∈ {Simbolo, λ} e b ∈ {topo(), λ}
   enquanto δ(Estado, a, b) for definida faça
      seja δ(Estado, a, b) = [Estado', z]
      se a ≠ λ então
         Simbolo ← prox()
      fim se
      se b ≠ λ então
         desempilhe()
      fim se
      empilhe(z)
      Estado ← Estado'
   fim enquanto
   se Simbolo = EOS e topo() = "MarcaFundo" e Estado ∈ F então
      retorne "Entrada foi aceita"
   senão
      retorne "Entrada não foi aceita"
   fim se
fim algoritmo
```

Autômatos de Pilha

Simulação de APD

```
algoritmo
   Entrada:  APD M = (E, Σ, Γ, δ, i, F)
             palavra de entrada w dada por prox() e termina
             com EOS (= "End of Sequence")
   Estado ← i
   empilha("MarcaFundo")
   Simbolo ← prox()
   seja a ∈ {Simbolo, λ} e b ∈ {topo(), λ}
   enquanto δ(Estado, a, b) for definida faça
      seja δ(Estado, a, b) = [Estado′, z]
      se a ≠ λ então
         Simbolo ← prox()
      fim se
      se b ≠ λ então
         desempilhe()
      fim se
      empilhe(z)
      Estado ← Estado′
   fim enquanto
   se Simbolo = EOS e topo() = “MarcaFundo′′ e Estado ∈ F então
      retorne "Entrada foi aceita"
   senão
      retorne "Entrada não foi aceita"
   fim se
fim algoritmo
```

Autômatos de Pilha

Simulação de APD

```
algoritmo
   Entrada:  APD $M = (E, \Sigma, \Gamma, \delta, i, F)$
             palavra de entrada $w$ dada por $prox()$ e termina
             com $EOS$ (= "End of Sequence")
   $Estado \leftarrow i$
   empilha("MarcaFundo")
   $Simbolo \leftarrow prox()$
   seja $a \in \{Simbolo, \lambda\}$ e $b \in \{topo(), \lambda\}$
   enquanto $\delta(Estado, a, b)$ for definida faça
      seja $\delta(Estado, a, b) = [Estado', z]$
      se $a \neq \lambda$ então
         $Simbolo \leftarrow prox()$
      fim se
      se $b \neq \lambda$ então
         desempilhe()
      fim se
      empilhe(z)
      $Estado \leftarrow Estado'$
   fim enquanto
   se $Simbolo = EOS$ e $topo() = “MarcaFundo''$ e $Estado \in F$ então
      retorne "Entrada foi aceita"
   senão
      retorne "Entrada não foi aceita"
   fim se
fim algoritmo
```

Autômatos de Pilha

Simulação de APD

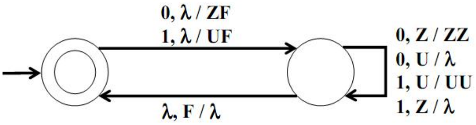
```
algoritmo
   Entrada:  APD M = (E, Σ, Γ, δ, i, F)
             palavra de entrada w dada por prox() e termina
             com EOS (= "End of Sequence")
   Estado ← i
   empilha("MarcaFundo")
   Simbolo ← prox()
   seja a ∈ {Simbolo, λ} e b ∈ {topo(), λ}
   enquanto δ(Estado, a, b) for definida faça
      seja δ(Estado, a, b) = [Estado′, z]
      se a ≠ λ então
         Simbolo ← prox()
      fim se
      se b ≠ λ então
         desempilhe()
      fim se
      empilhe(z)
      Estado ← Estado′
   fim enquanto
   se Simbolo = EOS e topo() = “MarcaFundo″ e Estado ∈ F então
      retorne "Entrada foi aceita"
   senão
      retorne "Entrada não foi aceita"
   fim se
fim algoritmo
```

Autômatos de Pilha

Autômatos de Pilha Determinísticos

Exercício.

(a) Qual a linguagem reconhecida pelo seguinte APD ?



(b) Construa o diagrama de um APD para a linguagem $\{0^m1^n \mid m \leq n\}$

(c) Construa o diagrama de um APD para a linguagem $\{0^m1^n\# \mid m \geq n\}$

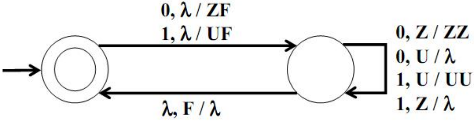
(d) Existe um APD para a linguagem $\{0^m1^n \mid m \geq n\}$?

Autômatos de Pilha

Autômatos de Pilha Determinísticos

Exercício.

(a) Qual a linguagem reconhecida pelo seguinte APD ?



(b) Construa o diagrama de um APD para a linguagem $\{0^m1^n \mid m \leq n\}$

(c) Construa o diagrama de um APD para a linguagem $\{0^m1^n\# \mid m \geq n\}$

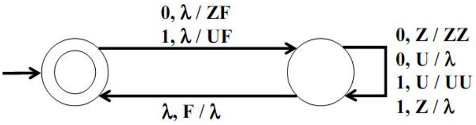
(d) Existe um APD para a linguagem $\{0^m1^n \mid m \geq n\}$?

Autômatos de Pilha

Autômatos de Pilha Determinísticos

Exercício.

(a) Qual a linguagem reconhecida pelo seguinte APD ?



(b) Construa o diagrama de um APD para a linguagem $\{0^m 1^n \mid m \leq n\}$

(c) Construa o diagrama de um APD para a linguagem $\{0^m 1^n \# \mid m \geq n\}$

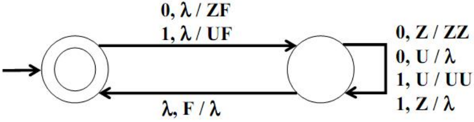
(d) Existe um APD para a linguagem $\{0^m 1^n \mid m \geq n\}$?

Autômatos de Pilha

Autômatos de Pilha Determinísticos

Exercício.

(a) Qual a linguagem reconhecida pelo seguinte APD ?



(b) Construa o diagrama de um APD para a linguagem $\{0^m1^n \mid m \leq n\}$

(c) Construa o diagrama de um APD para a linguagem $\{0^m1^n\# \mid m \geq n\}$

(d) Existe um APD para a linguagem $\{0^m1^n \mid m \geq n\}$?

Autômatos de Pilha

Autômato de Pilha Não Determinístico

Um **Autômato de Pilha Não Determinístico** (APN) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto \mathcal{P}(E \times \Gamma^*)$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ pode possuir transições compatíveis ou não !!!

Autômatos de Pilha

Autômato de Pilha Não Determinístico

Um **Autômato de Pilha Não Determinístico** (APN) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto \mathcal{P}(E \times \Gamma^*)$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ pode possuir transições compatíveis ou não !!!

Autômatos de Pilha

Autômato de Pilha Não Determinístico

Um **Autômato de Pilha Não Determinístico** (APN) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto \mathcal{P}(E \times \Gamma^*)$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ pode possuir transições compatíveis ou não !!!

Autômatos de Pilha

Autômato de Pilha Não Determinístico

Um **Autômato de Pilha Não Determinístico** (APN) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto \mathcal{P}(E \times \Gamma^*)$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ pode possuir transições compatíveis ou não !!!

Autômatos de Pilha

Autômato de Pilha Não Determinístico

Um **Autômato de Pilha Não Determinístico** (APN) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto \mathcal{P}(E \times \Gamma^*)$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ pode possuir transições compatíveis ou não !!!

Autômatos de Pilha

Autômato de Pilha Não Determinístico

Um **Autômato de Pilha Não Determinístico** (APN) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto \mathcal{P}(E \times \Gamma^*)$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ pode possuir transições compatíveis ou não !!!

Autômatos de Pilha

Autômato de Pilha Não Determinístico

Um **Autômato de Pilha Não Determinístico** (APN) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto \mathcal{P}(E \times \Gamma^*)$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ pode possuir transições compatíveis ou não !!!

Autômatos de Pilha

Autômato de Pilha Não Determinístico

Um **Autômato de Pilha Não Determinístico** (APN) é uma sêxtupla $M = (E, \Sigma, \Gamma, \delta, i, F)$ em que:

- $E \equiv$ conjunto de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\Gamma \equiv$ alfabeto de pilha
- $\delta \equiv$ função de transição:

$$\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \mapsto \mathcal{P}(E \times \Gamma^*)$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

OBS: δ pode possuir transições compatíveis ou não !!!

Autômatos de Pilha

Linguagem Reconhecida por um APN

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M é

$$L(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, \lambda] \text{ para algum } f \in F\}.$$

Exemplo. APN para a linguagem $\{0^m 1^n \mid m \geq n\}$

Autômatos de Pilha

Linguagem Reconhecida por um APN

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M é

$$L(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, \lambda] \text{ para algum } f \in F\}.$$

Exemplo. APN para a linguagem $\{0^m 1^n \mid m \geq n\}$

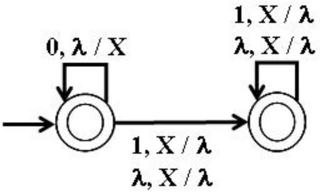
Autômatos de Pilha

Linguagem Reconhecida por um APN

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M é

$$L(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, \lambda] \text{ para algum } f \in F\}.$$

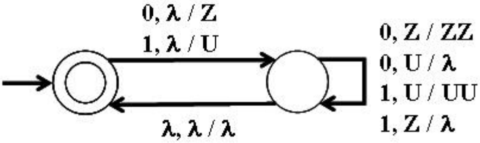
Exemplo. APN para a linguagem $\{0^m 1^n \mid m \geq n\}$



Autômatos de Pilha

Exemplos de APNs

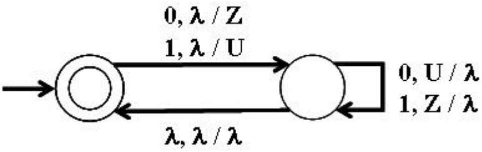
Seja $L = \{w \in \{0,1\}^* \mid \text{número de 0s em } w \text{ é igual ao número de 1s}\}$.
Um APN para L:



Autômatos de Pilha

Exemplos de APNs

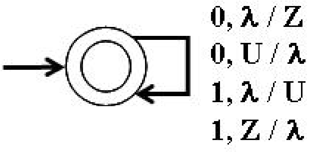
Seja $L = \{w \in \{0, 1\}^* \mid \text{número de 0s em } w \text{ é igual ao número de 1s}\}$.
Um segundo APN para L:



Autômatos de Pilha

Exemplos de APNs

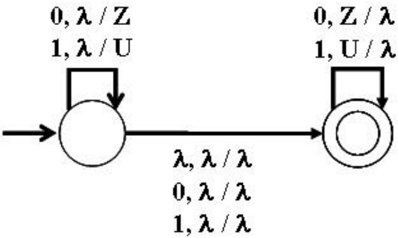
Seja $L = \{w \in \{0,1\}^* \mid \text{número de 0s em } w \text{ é igual ao número de 1s}\}$.
Um terceiro APN para L:



Autômatos de Pilha

Exemplos de APNs

Seja $L_1 = \{w \in \{0,1\}^* \mid w = w^R\}$:
Um APN para L_1:



Autômatos de Pilha

Linguagem Reconhecida por um APN por Pilha Vazia e Estado Final

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M por pilha vazia e estado final é

$$L(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, \lambda] \text{ para algum } f \in F\}.$$

Linguagem Reconhecida por um APN por Estado Final

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M por estado final é

$$L_F(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, y] \text{ para algum } f \in F,\ y \in \Gamma^*\}.$$

Linguagem Reconhecida por um APN por Pilha Vazia

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i)$. A linguagem reconhecida por M por pilha vazia é

$$L_V(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [e, \lambda, \lambda] \text{ para algum } e \in E\}.$$

Autômatos de Pilha

Linguagem Reconhecida por um APN por Pilha Vazia e Estado Final

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M por pilha vazia e estado final é

$$L(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, \lambda] \text{ para algum } f \in F\}.$$

Linguagem Reconhecida por um APN por Estado Final

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M por estado final é

$$L_F(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, y] \text{ para algum } f \in F,\ y \in \Gamma^*\}.$$

Linguagem Reconhecida por um APN por Pilha Vazia

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i)$. A linguagem reconhecida por M por pilha vazia é

$$L_V(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [e, \lambda, \lambda] \text{ para algum } e \in E\}.$$

Autômatos de Pilha

Linguagem Reconhecida por um APN por Pilha Vazia e Estado Final

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M por pilha vazia e estado final é

$$L(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, \lambda] \text{ para algum } f \in F\}.$$

Linguagem Reconhecida por um APN por Estado Final

Seja um APN $M = (E, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida por M por estado final é

$$L_F(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [f, \lambda, y] \text{ para algum } f \in F,\ y \in \Gamma^*\}.$$

Linguagem Reconhecida por um APN por Pilha Vazia

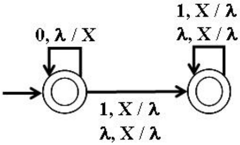
Seja um APN $M = (E, \Sigma, \Gamma, \delta, i)$. A linguagem reconhecida por M por pilha vazia é

$$L_V(M) = \{w \in \Sigma^* \mid [i, w, \lambda] \overset{*}{\vdash} [e, \lambda, \lambda] \text{ para algum } e \in E\}.$$

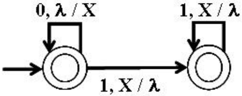
Autômatos de Pilha

Exemplos de Linguagens Reconhecidas por APNs

Seja $L = \{0^m1^n \mid m \geq n\}$.
Um APN M tal que $L(M) = L$:



Um APN M' tal que $L_F(M') = L$:

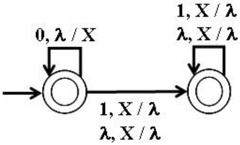


OBS: $L_F(M) = L$ $\qquad L(M') = \{0^n1^n \mid n \geq 0\}$

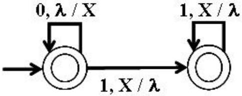
Autômatos de Pilha

Exemplos de Linguagens Reconhecidas por APNs

Seja $L = \{0^m1^n \mid m \geq n\}$.
Um APN M tal que $L(M) = L$:



Um APN M' tal que $L_F(M') = L$:

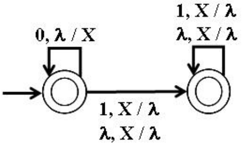


OBS: $L_F(M) = L$ $\qquad L(M') = \{0^n1^n \mid n \geq 0\}$

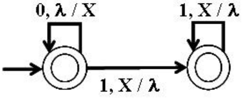
Autômatos de Pilha

Exemplos de Linguagens Reconhecidas por APNs

Seja $L = \{0^m 1^n \mid m \geq n\}$.
Um APN M tal que $L(M) = L$:



Um APN M' tal que $L_F(M') = L$:

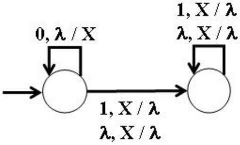


OBS: $L_F(M) = L$ $\qquad L(M') = \{0^n 1^n \mid n \geq 0\}$

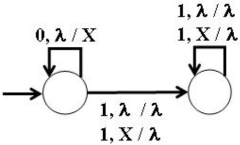
Autômatos de Pilha

Exemplos de Linguagens Reconhecidas por APNs

Um APN M tal que $L_V(M) = \{0^m 1^n \mid m \geq n\}$:



Um APN M' tal que $L_V(M') = \{0^m 1^n \mid m \leq n\}$:



Autômatos de Pilha

Linguagens Reconhecidas por APNs

Seja L uma linguagem reconhecida por um APN, então as seguintes afirmações são equivalentes:

1. L pode ser reconhecida por pilha vazia e estado final;
2. L pode ser reconhecida estado final; e
3. $L \cup \{\lambda\}$ pode ser reconhecida por pilha vazia.