Introdução à Teoria da Computação

Autômatos com Pilha - Parte 2

Professor Murilo V. G. da Silva

Departamento de Informática

Universidade Federal do Paraná

2025/2

Diagrama de APs

Diagrama de APs

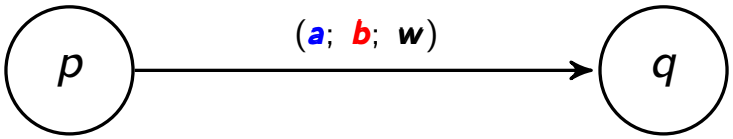
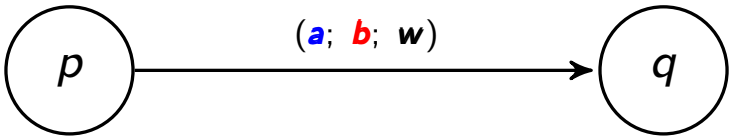
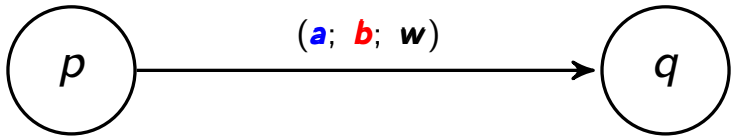


Diagrama de APs



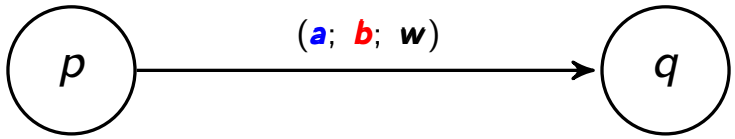
- Símbolo da string de entrada: ***a***

Diagrama de APs



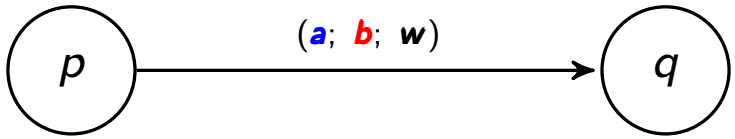
- Símbolo da string de entrada: ***a***
- Símbolo do topo da pilha: ***b***

Diagrama de APs



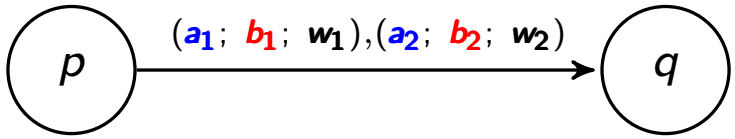
- Símbolo da string de entrada: ***a***
- Símbolo do topo da pilha: ***b***
- String empilhada após transição: ***w***

Diagrama de APs



- Símbolo da string de entrada: a
- Símbolo do topo da pilha: b
- String empilhada após transição: w

Pode haver vários rótulos em uma transição



APs: Exemplo de diagrama

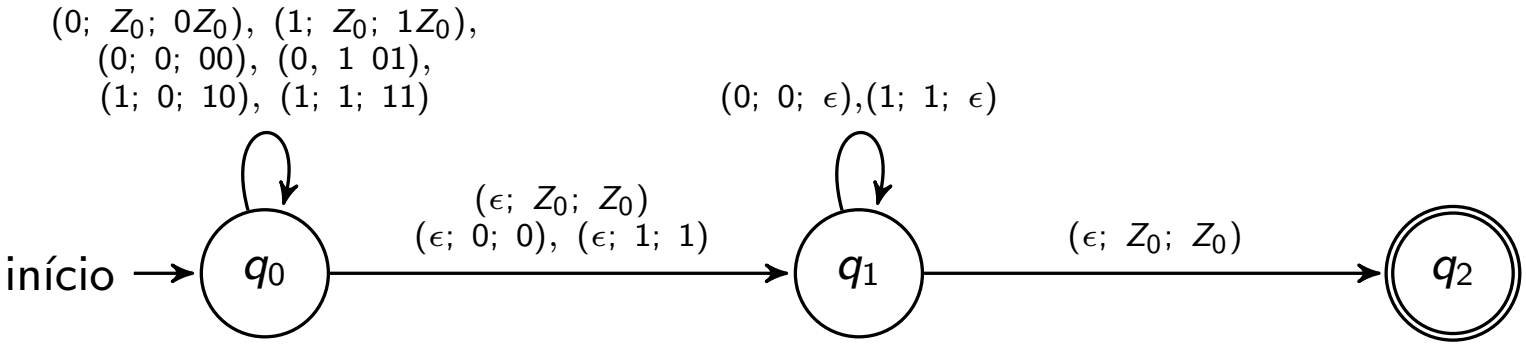
Segue um AP para $L_{\mathrm{RR}} = \{ww^R : w \in \Sigma^*\}$:

APs: Exemplo de diagrama

Segue um AP para $L_{\mathrm{RR}} = \{ww^R : w \in \Sigma^*\}$:

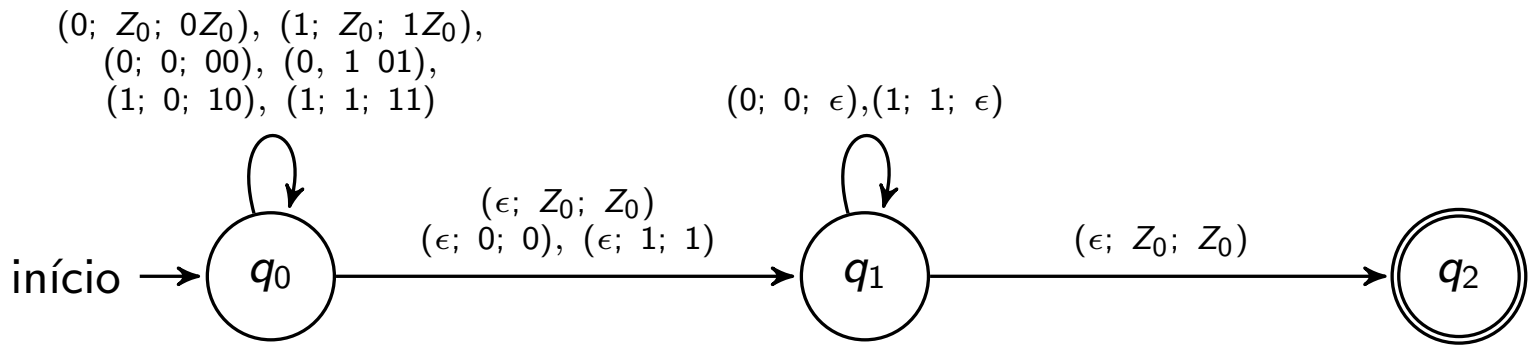
APs: Exemplo de diagrama

Segue um AP para $L_{\mathrm{RR}} = \{ww^R : w \in \Sigma^*\}$:



APs: Exemplo de diagrama

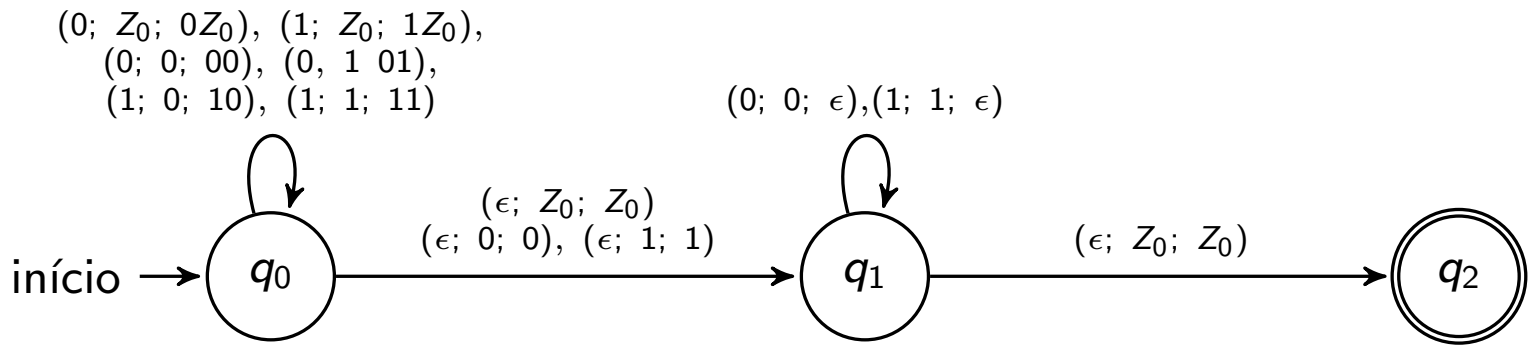
Segue um AP para $L_{\mathrm{RR}} = \{ww^R : w \in \Sigma^*\}$:



Exemplo de string aceita: 011110

APs: Exemplo de diagrama

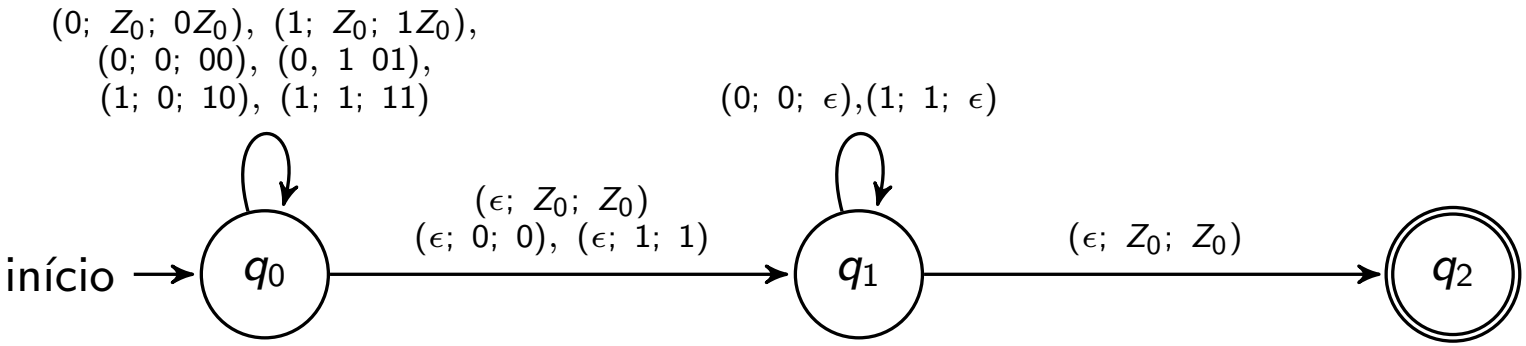
Segue um AP para $L_{\mathrm{RR}} = \{ww^R : w \in \Sigma^*\}$:



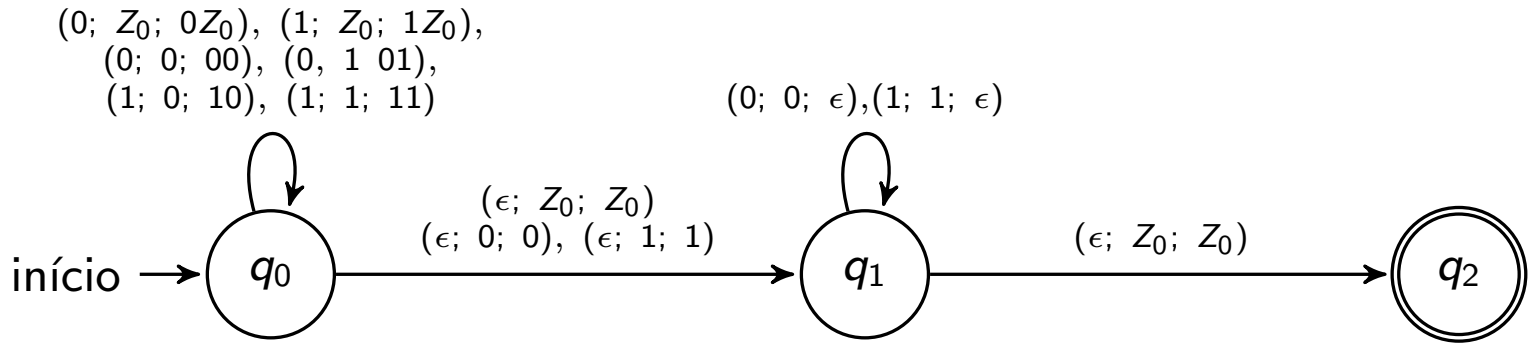
Exemplo de string aceita: 011110

Exemplo de string rejeitada: 10

O processo de computação com APs

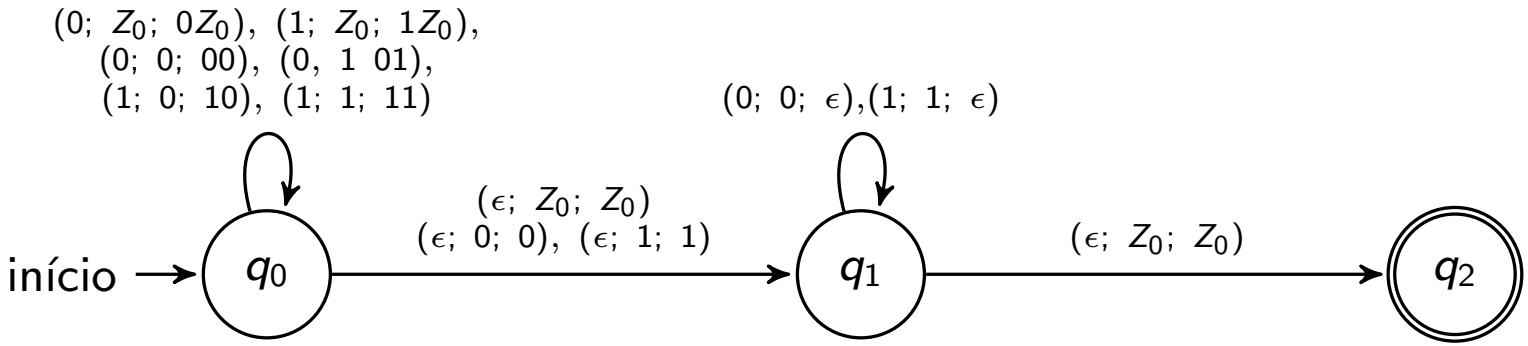


O processo de computação com APs



A computação passo a passo:

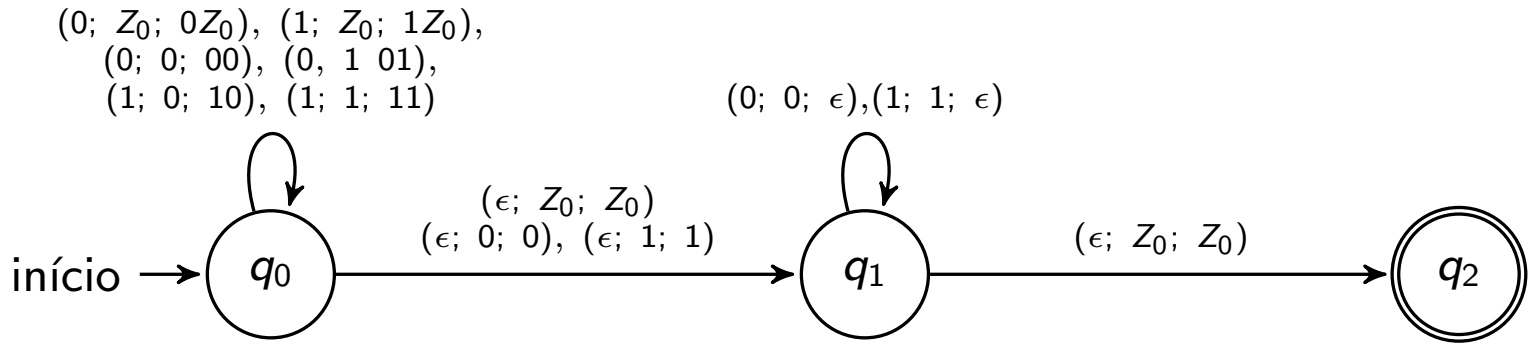
O processo de computação com APs



A computação passo a passo:

- entrada: 011110 estado: q_0 pilha: Z_0

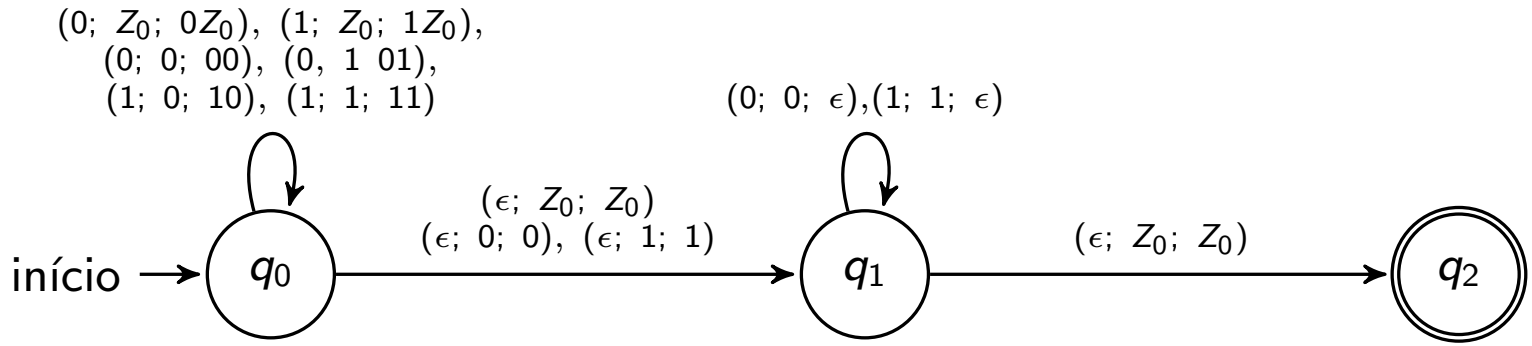
O processo de computação com APs



A computação passo a passo:

- entrada: 011110 estado: q_0 pilha: Z_0
- entrada: 11110 estado: q_0 pilha: $0Z_0$

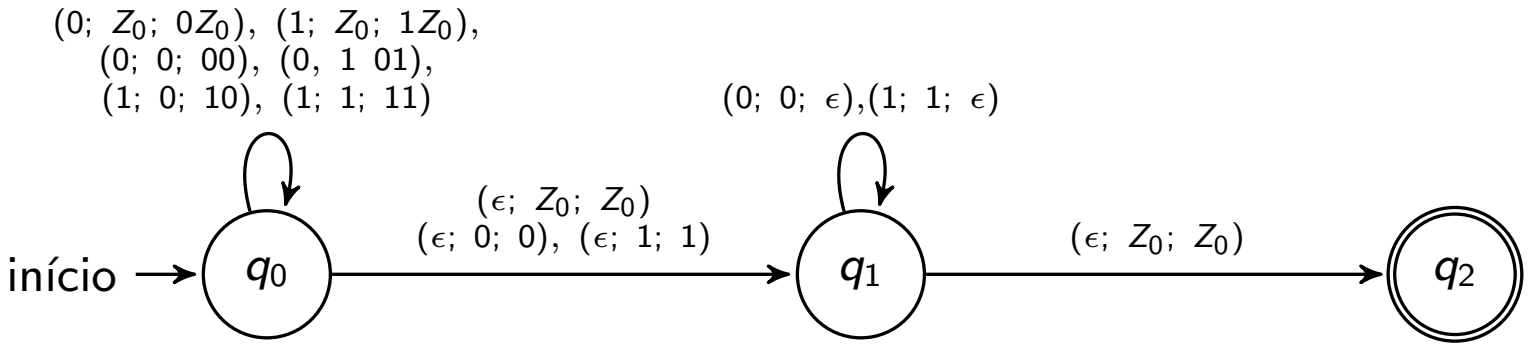
O processo de computação com APs



A computação passo a passo:

- entrada: 011110 estado: q_0 pilha: Z_0
- entrada: 11110 estado: q_0 pilha: $0Z_0$
- entrada: 1110 estado: q_0 pilha: $10Z_0$

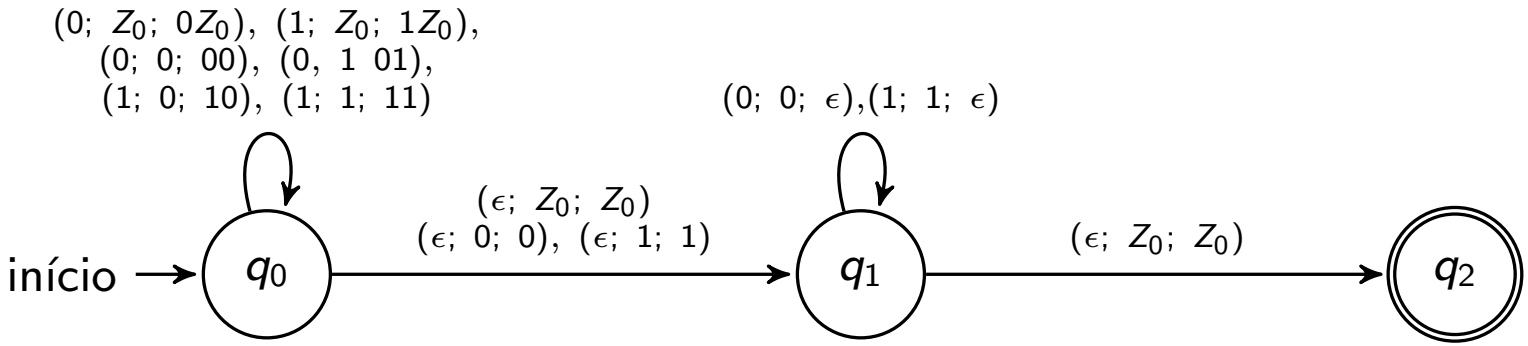
O processo de computação com APs



A computação passo a passo:

- entrada: 011110 estado: q_0 pilha: Z_0
- entrada: 11110 estado: q_0 pilha: $0Z_0$
- entrada: 1110 estado: q_0 pilha: $10Z_0$
- entrada: 110 estado: q_0 pilha: $110Z_0$

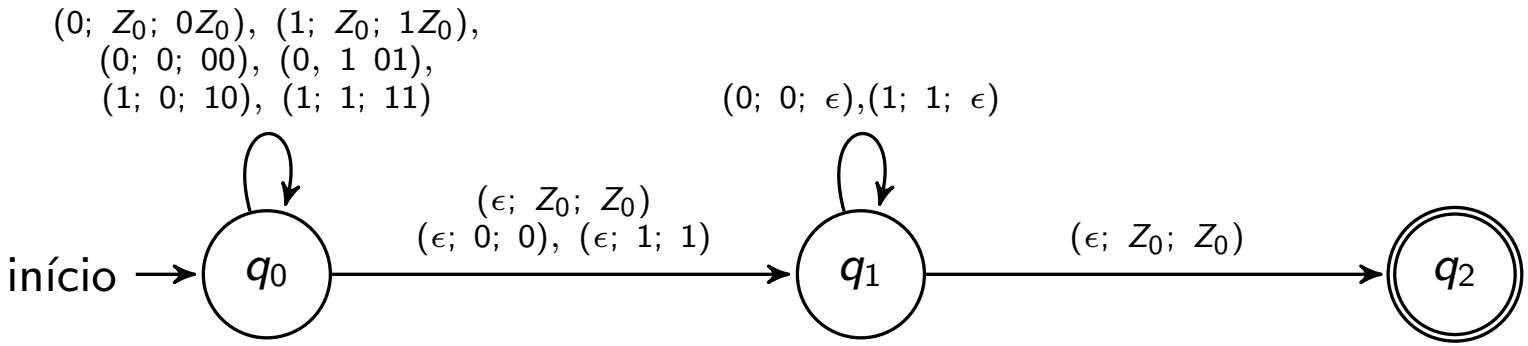
O processo de computação com APs



A computação passo a passo:

- entrada: 011110 estado: q_0 pilha: Z_0
- entrada: 11110 estado: q_0 pilha: $0Z_0$
- entrada: 1110 estado: q_0 pilha: $10Z_0$
- entrada: 110 estado: q_0 pilha: $110Z_0$
- entrada: 110 estado: q_1 pilha: $110Z_0$

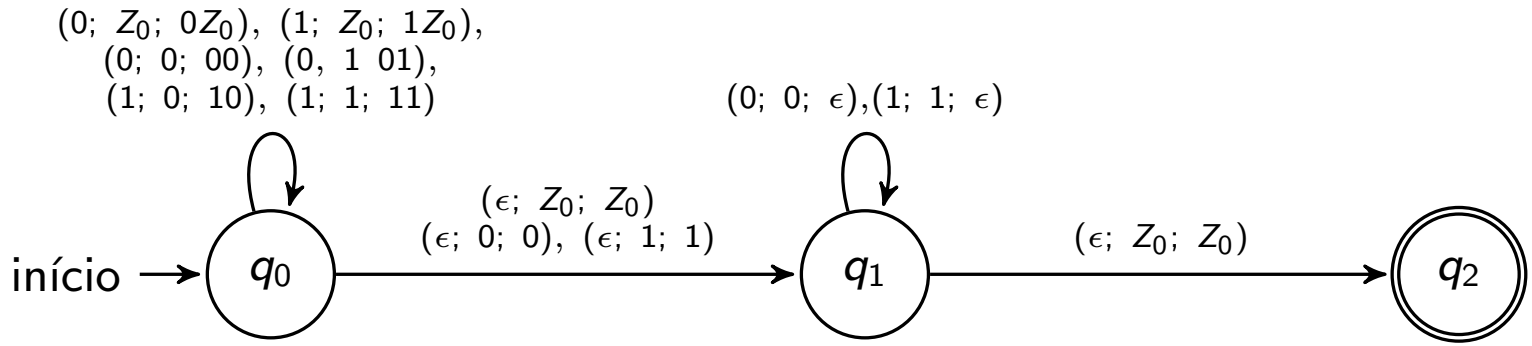
O processo de computação com APs



A computação passo a passo:

- entrada: 011110 estado: q_0 pilha: Z_0
- entrada: 11110 estado: q_0 pilha: $0Z_0$
- entrada: 1110 estado: q_0 pilha: $10Z_0$
- entrada: 110 estado: q_0 pilha: $110Z_0$
- entrada: 110 estado: q_1 pilha: $110Z_0$
- entrada: 10 estado: q_1 pilha: $10Z_0$

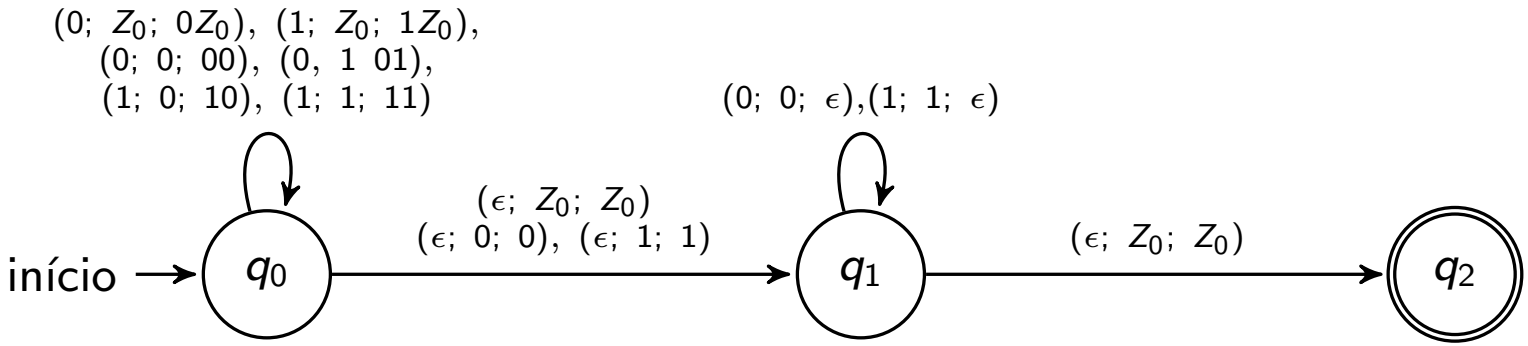
O processo de computação com APs



A computação passo a passo:

- entrada: 011110 estado: q_0 pilha: Z_0
- entrada: 11110 estado: q_0 pilha: $0Z_0$
- entrada: 1110 estado: q_0 pilha: $10Z_0$
- entrada: 110 estado: q_0 pilha: $110Z_0$
- entrada: 110 estado: q_1 pilha: $110Z_0$
- entrada: 10 estado: q_1 pilha: $10Z_0$
- entrada: 0 estado: q_1 pilha: $0Z_0$

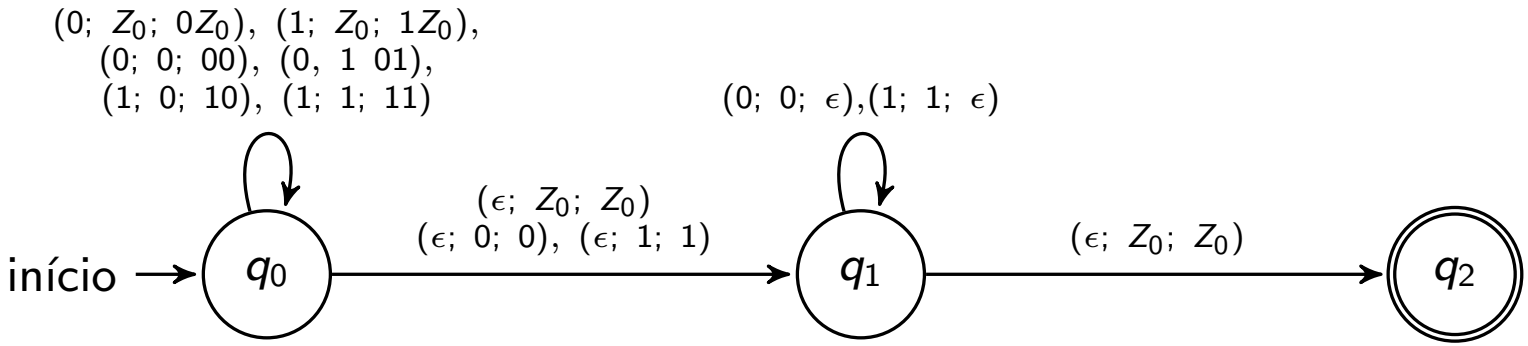
O processo de computação com APs



A computação passo a passo:

- entrada: 011110 estado: q_0 pilha: Z_0
- entrada: 11110 estado: q_0 pilha: $0Z_0$
- entrada: 1110 estado: q_0 pilha: $10Z_0$
- entrada: 110 estado: q_0 pilha: $110Z_0$
- entrada: 110 estado: q_1 pilha: $110Z_0$
- entrada: 10 estado: q_1 pilha: $10Z_0$
- entrada: 0 estado: q_1 pilha: $0Z_0$
- entrada: ϵ estado: q_1 pilha: Z_0

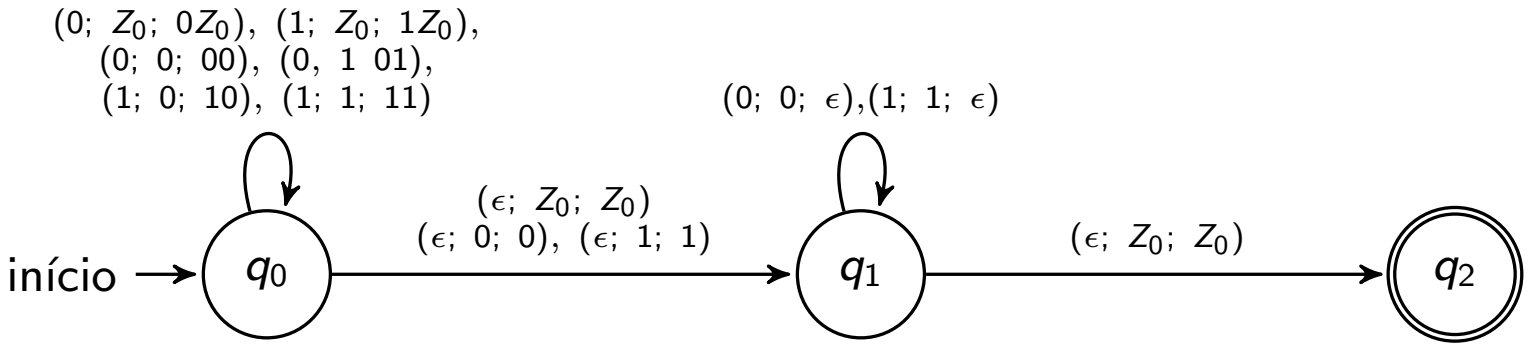
O processo de computação com APs



A computação passo a passo:

- entrada: 011110 estado: q_0 pilha: Z_0
- entrada: 11110 estado: q_0 pilha: $0Z_0$
- entrada: 1110 estado: q_0 pilha: $10Z_0$
- entrada: 110 estado: q_0 pilha: $110Z_0$
- entrada: 110 estado: q_1 pilha: $110Z_0$
- entrada: 10 estado: q_1 pilha: $10Z_0$
- entrada: 0 estado: q_1 pilha: $0Z_0$
- entrada: ϵ estado: q_1 pilha: Z_0
- entrada: ϵ estado: q_2 pilha: Z_0

O processo de computação com APs



A computação passo a passo:

- entrada: 011110 estado: q_0 pilha: Z_0
- entrada: 11110 estado: q_0 pilha: $0Z_0$
- entrada: 1110 estado: q_0 pilha: $10Z_0$
- entrada: 110 estado: q_0 pilha: $110Z_0$
- entrada: 110 estado: q_1 pilha: $110Z_0$
- entrada: 10 estado: q_1 pilha: $10Z_0$
- entrada: 0 estado: q_1 pilha: $0Z_0$
- entrada: ϵ estado: q_1 pilha: Z_0
- entrada: ϵ estado: q_2 pilha: Z_0

A cada momento a situação é definida por uma tripla (w, q, γ)

O processo de computação com APs

O processo de computação com APs

Def.: Configuração de um AP

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP,

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$,

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$,

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

- Uma tripla (q, w, γ) é uma configuração de P.

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

- Uma tripla (q, w, γ) é uma configuração de P.

Saindo de uma configuração C_1 para outra configuração C_2:

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

- Uma tripla (q, w, γ) é uma configuração de P.

Saindo de uma configuração C_1 para outra configuração C_2: $C_1 \vdash C_2$

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

- Uma tripla (q, w, γ) é uma configuração de P.

Saindo de uma configuração C_1 para outra configuração C_2: $C_1 \vdash C_2$

Def.: Passo computacional

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

- Uma tripla (q, w, γ) é uma configuração de P.

Saindo de uma configuração C_1 para outra configuração C_2: $\boldsymbol{C_1 \vdash C_2}$

Def.: Passo computacional

Seja um AP $P = (Q, \Sigma, \Gamma, \boldsymbol{\delta}, q_0, Z_0, F)$

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

- Uma tripla (q, w, γ) é uma configuração de P.

Saindo de uma configuração C_1 para outra configuração C_2: $\boldsymbol{C_1 \vdash C_2}$

Def.: Passo computacional

Seja um AP $P = (Q, \Sigma, \Gamma, \boldsymbol{\delta}, q_0, Z_0, F)$ e sejam $q \in Q$, $a \in \Sigma$, $X \in \Gamma$

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

- Uma tripla (q, w, γ) é uma configuração de P.

Saindo de uma configuração C_1 para outra configuração C_2: $C_1 \vdash C_2$

Def.: Passo computacional

Seja um AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ e sejam $q \in Q$, $a \in \Sigma$, $X \in \Gamma$ e $w \in \Sigma^*$, $\beta \in \Gamma^*$.

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

- Uma tripla (q, w, γ) é uma configuração de P.

Saindo de uma configuração C_1 para outra configuração C_2: $\boldsymbol{C_1 \vdash C_2}$

Def.: Passo computacional

Seja um AP $P = (Q, \Sigma, \Gamma, \boldsymbol{\delta}, q_0, Z_0, F)$ e sejam $q \in Q$, $a \in \Sigma$, $X \in \Gamma$ e $w \in \Sigma^*$, $\beta \in \Gamma^*$.

Se $\boldsymbol{\delta(q, a, X)}$ contém (p, α),

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

- Uma tripla (q, w, γ) é uma configuração de P.

Saindo de uma configuração C_1 para outra configuração C_2: $\boldsymbol{C_1 \vdash C_2}$

Def.: Passo computacional

Seja um AP $P = (Q, \Sigma, \Gamma, \boldsymbol{\delta}, q_0, Z_0, F)$ e sejam $q \in Q$, $a \in \Sigma$, $X \in \Gamma$ e $w \in \Sigma^*$, $\beta \in \Gamma^*$.

Se $\boldsymbol{\delta(q, a, X)}$ contém (p, α), então escrevemos

O processo de computação com APs

Def.: Configuração de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP, $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$.

- Uma tripla (q, w, γ) é uma configuração de P.

Saindo de uma configuração C_1 para outra configuração C_2: $\boldsymbol{C_1 \vdash C_2}$

Def.: Passo computacional

Seja um AP $P = (Q, \Sigma, \Gamma, \boldsymbol{\delta}, q_0, Z_0, F)$ e sejam $q \in Q$, $a \in \Sigma$, $X \in \Gamma$ e $w \in \Sigma^*$, $\beta \in \Gamma^*$.

Se $\boldsymbol{\delta(q, a, X)}$ contém (p, α), então escrevemos

$$\boldsymbol{(q, aw, X\beta) \vdash_P (p, w, \alpha\beta)}$$

O processo de computação com APs

O processo de computação com APs

Def.: Sequência de passos computacionais

O processo de computação com APs

Def.: Sequência de passos computacionais

$C_i \vdash_P^* C_j$: A partir C_i o autômato P atinge C_j com uma quantidade arbitrária de passos. (zero ou mais passos)

O processo de computação com APs

Def.: Sequência de passos computacionais

$C_i \vdash_P^* C_j$: A partir C_i o autômato P atinge C_j com uma quantidade arbitrária de passos. (zero ou mais passos)

O processo de computação com APs

Def.: Sequência de passos computacionais

$C_i \vdash_P^* C_j$: A partir C_i o autômato P atinge C_j com uma quantidade arbitrária de passos. (zero ou mais passos)

Def.: Linguagem de um AP

O processo de computação com APs

Def.: Sequência de passos computacionais

$C_i \vdash_P^* C_j$: A partir C_i o autômato P atinge C_j com uma quantidade arbitrária de passos. (zero ou mais passos)

Def.: Linguagem de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP. Definimos a linguagem de P como

$$L(P) = \{w \ : \ (q_0, w, Z_0) \vdash_P^* (q, \epsilon, \alpha) \text{ para } q \in F \text{ e } \alpha \in \Gamma^* \text{ qualquer}\}.$$

O processo de computação com APs

Def.: Sequência de passos computacionais

$C_i \vdash_P^* C_j$: A partir C_i o autômato P atinge C_j com uma quantidade arbitrária de passos. (zero ou mais passos)

Def.: Linguagem de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP. Definimos a linguagem de P como

$$L(P) = \{w \ : \ (q_0, w, Z_0) \vdash_P^* (q, \epsilon, \alpha) \text{ para } q \in F \text{ e } \alpha \in \Gamma^* \text{ qualquer}\}.$$

Def.: Linguagem livre de contexto

O processo de computação com APs

Def.: Sequência de passos computacionais

$C_i \vdash_P^* C_j$: A partir C_i o autômato P atinge C_j com uma quantidade arbitrária de passos. (zero ou mais passos)

Def.: Linguagem de um AP

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP. Definimos a linguagem de P como

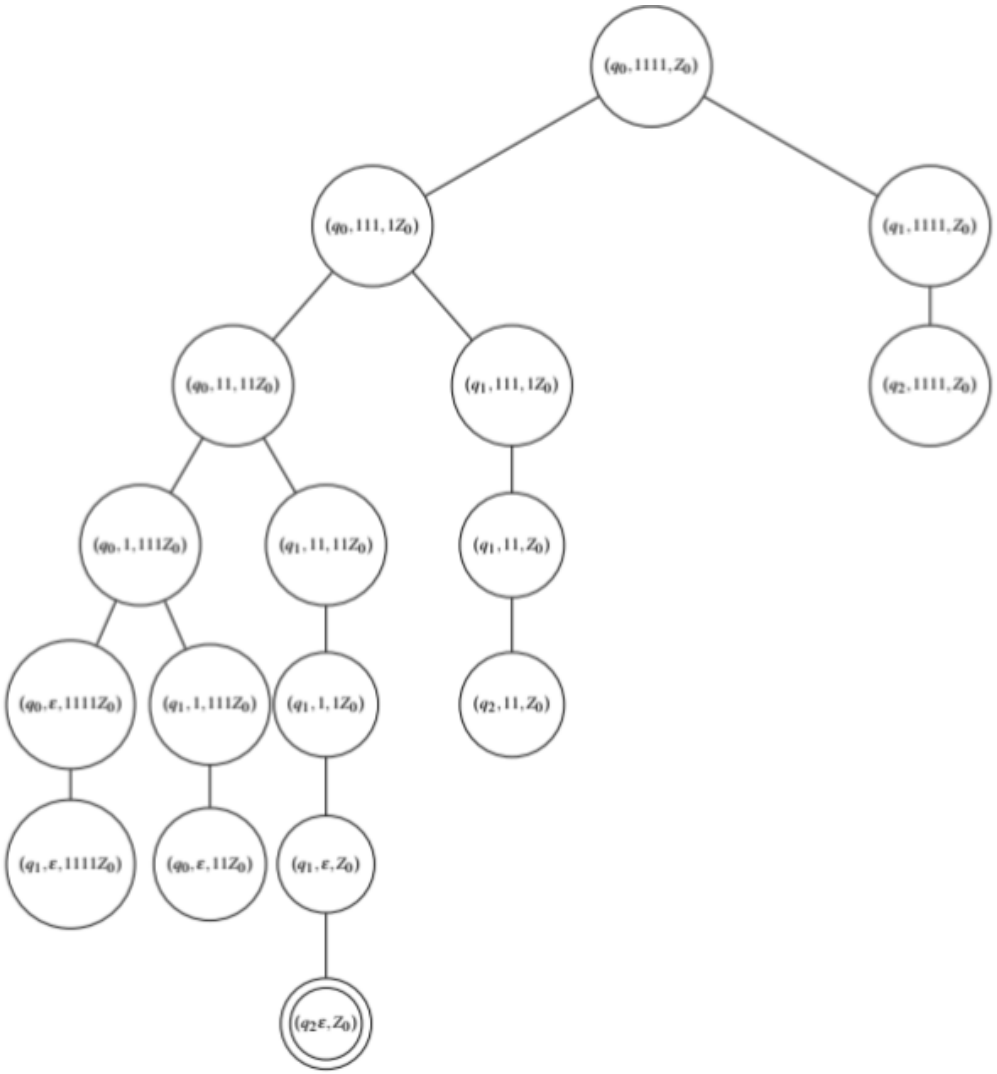
$$L(P) = \{w \ : \ (q_0, w, Z_0) \vdash_P^* (q, \epsilon, \alpha) \text{ para } q \in F \text{ e } \alpha \in \Gamma^* \text{ qualquer}\}.$$

Def.: Linguagem livre de contexto

Dada uma linguagem L, se existe um AP P tal que $L = L(P)$, então L é dita uma *Linguagem Livre de Contexto*.

Árvore de computações possíveis

Árvore do autômato P_{RR} com a string 1111:



Strings que fazem o autômato esvaziar a pilha:

Definição de N(P)

Strings que fazem o autômato esvaziar a pilha:

Definição de N(P)

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP.

Strings que fazem o autômato esvaziar a pilha:

Definição de N(P)

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP.

$$N(P) = \{w : (q_0, w, Z_0) \vdash_P^* (q, \epsilon, \epsilon) \text{ para } q \in Q \text{ qualquer}\}.$$

Strings que fazem o autômato esvaziar a pilha:

Definição de N(P)

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP.

$$N(P) = \{w : (q_0, w, Z_0) \vdash_P^* (q, \epsilon, \epsilon) \text{ para } q \in Q \text{ qualquer}\}.$$

Exemplo: Como o AP P_{RR} nunca esvazia a pilha, então $N(P_{\mathrm{RR}}) = \varnothing$.

Strings que fazem o autômato esvaziar a pilha:

Definição de N(P)

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP.

$$N(P) = \{w : (q_0, w, Z_0) \vdash_P^* (q, \epsilon, \epsilon) \text{ para } q \in Q \text{ qualquer}\}.$$

Exemplo: Como o AP P_{RR} nunca esvazia a pilha, então $N(P_{\mathrm{RR}}) = \varnothing$.

Teorema: Seja P uma AP qualquer. Então existe um AP P' tal que $L(P') = N(P)$.

Strings que fazem o autômato esvaziar a pilha:

Definição de N(P)

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP.

$$N(P) = \{w : (q_0, w, Z_0) \vdash_P^* (q, \epsilon, \epsilon) \text{ para } q \in Q \text{ qualquer}\}.$$

Exemplo: Como o AP P_{RR} nunca esvazia a pilha, então $N(P_{\mathrm{RR}}) = \varnothing$.

Teorema: Seja P uma AP qualquer. Então existe um AP P' tal que $L(P') = N(P)$.

Teorema: Seja P uma AP qualquer. Então existe um AP P' tal que $N(P') = L(P)$.

Strings que fazem o autômato esvaziar a pilha:

Definição de N(P)

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP.

$$N(P) = \{w : (q_0, w, Z_0) \vdash_P^* (q, \epsilon, \epsilon) \text{ para } q \in Q \text{ qualquer}\}.$$

Exemplo: Como o AP P_{RR} nunca esvazia a pilha, então $N(P_{\mathrm{RR}}) = \varnothing$.

Teorema: Seja P uma AP qualquer. Então existe um AP P' tal que $L(P') = N(P)$.

Teorema: Seja P uma AP qualquer. Então existe um AP P' tal que $N(P') = L(P)$.

- Pode ser mais fácil projetar um AP que esvazia a pilha para strings de interesse.

Strings que fazem o autômato esvaziar a pilha:

Definição de N(P)

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP.

$$N(P) = \{w : (q_0, w, Z_0) \vdash_P^* (q, \epsilon, \epsilon) \text{ para } q \in Q \text{ qualquer}\}.$$

Exemplo: Como o AP P_{RR} nunca esvazia a pilha, então $N(P_{\mathrm{RR}}) = \varnothing$.

Teorema: Seja P uma AP qualquer. Então existe um AP P' tal que $L(P') = N(P)$.

Teorema: Seja P uma AP qualquer. Então existe um AP P' tal que $N(P') = L(P)$.

- Pode ser mais fácil projetar um AP que esvazia a pilha para strings de interesse.
- E.g., mostrar autômatos que esvaziam a pilha para linguagem de *gramáticas*.

Strings que fazem o autômato esvaziar a pilha:

Definição de N(P)

Seja $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ um AP.

$$N(P) = \{w : (q_0, w, Z_0) \vdash_P^* (q, \epsilon, \epsilon) \text{ para } q \in Q \text{ qualquer}\}.$$

Exemplo: Como o AP P_{RR} nunca esvazia a pilha, então $N(P_{\mathrm{RR}}) = \varnothing$.

Teorema: Seja P uma AP qualquer. Então existe um AP P' tal que $L(P') = N(P)$.

Teorema: Seja P uma AP qualquer. Então existe um AP P' tal que $N(P') = L(P)$.

- Pode ser mais fácil projetar um AP que esvazia a pilha para strings de interesse.
- E.g., mostrar autômatos que esvaziam a pilha para linguagem de *gramáticas*.
  (assunto que veremos adiante)

Autômatos com pilha determinísticos

Definição de APDs

Autômatos com pilha determinísticos

Definição de APDs

Um *Autômato com Pilha Determinístico (APD)*

Autômatos com pilha determinísticos

Definição de APDs

Um *Autômato com Pilha Determinístico (APD)* é AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

Autômatos com pilha determinísticos

Definição de APDs

Um *Autômato com Pilha Determinístico (APD)* é AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ cuja função de transição δ tem as seguintes restrições:

Autômatos com pilha determinísticos

Definição de APDs

Um *Autômato com Pilha Determinístico (APD)* é AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ cuja função de transição δ tem as seguintes restrições:

1. Para quaisquer, $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$ e $X \in \Gamma$, temos $|\delta(q, a, X)| \leq 1$,

Autômatos com pilha determinísticos

Definição de APDs

Um *Autômato com Pilha Determinístico (APD)* é AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ cuja função de transição δ tem as seguintes restrições:

1. Para quaisquer, $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$ e $X \in \Gamma$, temos $|\delta(q, a, X)| \leq 1$,
2. Se existe $q \in Q$, $a \in \Sigma$ e $X \in \Gamma$ tal que $\delta(q, a, X) \neq \varnothing$, então $\delta(q, \epsilon, X) = \varnothing$.

Autômatos com pilha determinísticos

Definição de APDs

Um *Autômato com Pilha Determinístico (APD)* é AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ cuja função de transição δ tem as seguintes restrições:

1. Para quaisquer, $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$ e $X \in \Gamma$, temos $|\delta(q, a, X)| \leq 1$,
2. Se existe $q \in Q$, $a \in \Sigma$ e $X \in \Gamma$ tal que $\delta(q, a, X) \neq \varnothing$, então $\delta(q, \epsilon, X) = \varnothing$.

Exercício: Mostre que toda linguagem regular pode ser decidida por um APD.

Autômatos com pilha determinísticos

Definição de APDs

Um *Autômato com Pilha Determinístico (APD)* é AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ cuja função de transição δ tem as seguintes restrições:

1. Para quaisquer, $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$ e $X \in \Gamma$, temos $|\delta(q, a, X)| \leq 1$,
2. Se existe $q \in Q$, $a \in \Sigma$ e $X \in \Gamma$ tal que $\delta(q, a, X) \neq \varnothing$, então $\delta(q, \epsilon, X) = \varnothing$.

Exercício: Mostre que toda linguagem regular pode ser decidida por um APD.

Exercício: Considere o alfabeto $\{0, 1, M\}$

Autômatos com pilha determinísticos

Definição de APDs

Um *Autômato com Pilha Determinístico (APD)* é AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ cuja função de transição δ tem as seguintes restrições:

1. Para quaisquer, $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$ e $X \in \Gamma$, temos $|\delta(q, a, X)| \leq 1$,
2. Se existe $q \in Q$, $a \in \Sigma$ e $X \in \Gamma$ tal que $\delta(q, a, X) \neq \varnothing$, então $\delta(q, \epsilon, X) = \varnothing$.

Exercício: Mostre que toda linguagem regular pode ser decidida por um APD.

Exercício: Considere o alfabeto $\{0, 1, M\}$ e seja $L_{\mathrm{RMR}} = \{wMw^R \ : \text{tal que } w \in \{0,1\}^* \ \}$.

Autômatos com pilha determinísticos

Definição de APDs

Um *Autômato com Pilha Determinístico (APD)* é AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ cuja função de transição δ tem as seguintes restrições:

1. Para quaisquer, $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$ e $X \in \Gamma$, temos $|\delta(q, a, X)| \leq 1$,
2. Se existe $q \in Q$, $a \in \Sigma$ e $X \in \Gamma$ tal que $\delta(q, a, X) \neq \varnothing$, então $\delta(q, \epsilon, X) = \varnothing$.

Exercício: Mostre que toda linguagem regular pode ser decidida por um APD.

Exercício: Considere o alfabeto $\{0, 1, M\}$ e seja $L_{\mathrm{RMR}} = \{wMw^R \ : \text{tal que } w \in \{0,1\}^* \}$.

- Use o LB para mostrar que L_{RMR} não é regular
- Mostre um APD que decida L_{RMR}.

Autômatos com pilha determinísticos

Definição de APDs

Um *Autômato com Pilha Determinístico (APD)* é AP $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ cuja função de transição δ tem as seguintes restrições:

1. Para quaisquer, $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$ e $X \in \Gamma$, temos $|\delta(q, a, X)| \leq 1$,
2. Se existe $q \in Q$, $a \in \Sigma$ e $X \in \Gamma$ tal que $\delta(q, a, X) \neq \varnothing$, então $\delta(q, \epsilon, X) = \varnothing$.

Exercício: Mostre que toda linguagem regular pode ser decidida por um APD.

Exercício: Considere o alfabeto $\{0, 1, M\}$ e seja $L_{\mathrm{RMR}} = \{wMw^R \ : \text{tal que } w \in \{0,1\}^* \}$.

- Use o LB para mostrar que L_{RMR} não é regular
- Mostre um APD que decida L_{RMR}.

Teorema: Não existe APD que decida a linguagem L_{RR}.