Autômatos de Pilha (AP)

Material extraído do livro e slides do Prof. Newton Vieira (http://dcc.ufmg.br/~nvieira) e do Prof. Andrei Rimsa Álvares

Introdução

- Apesar das inúmeras aplicações de linguagens regulares, existem aplicações que requerem linguagens mais sofisticadas

- Exemplo: linguagens que contêm expressões aritméticas com balanceamento de parênteses

$$(^n t_1 + t_2) + t_3) \cdots) + t_{n+1})$$

  - onde $n \geq 0$, cada t_i é uma subexpressão, e o número de *(*'s é igual ao número de *)*'s

Intuitivamente, um autômato finito não possui memória poderosa o suficiente para lembrar que leu *n* ocorrências (arbitrárias) de (’s

É possível provar que essa linguagem não é regular (usando o lema do bombeamento)?

Autômatos de Pilha

- Os **autômatos de pilha** são uma extensão dos autômatos finitos que adicionam uma memória organizada como pilha

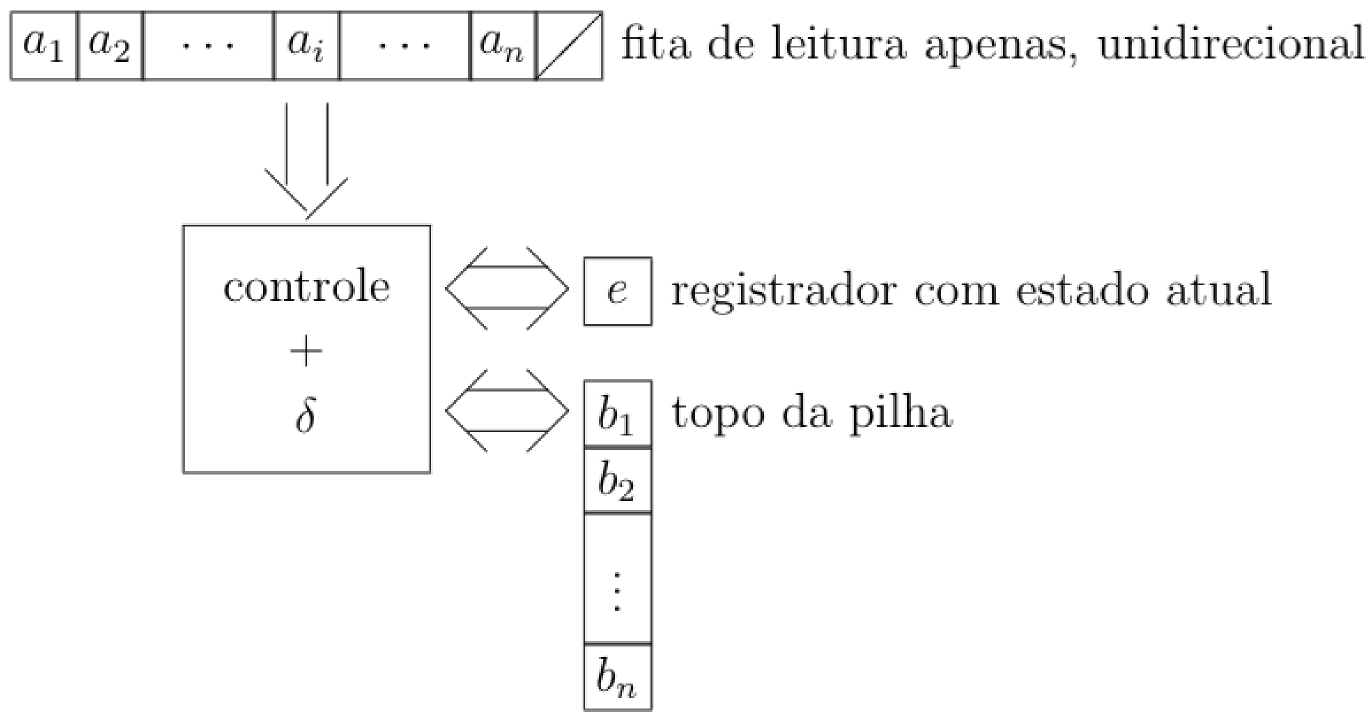
- São máquinas reconhecedoras para muitas linguagens que ocorrem na prática, como linguagens de programação

- Ao contrário de autômatos finitos, a versão não-determinística tem um poder de reconhecimento maior que a determinística
  - Contudo, autômatos de pilha determinísticos possuem implementação eficiente

- Os autômatos de pilha reconhecem a classe de **linguagens livres de contexto**

Arquitetura de um Autômato de Pilha

- A arquitetura é similar a de um autômato finito, mas contém adicionalmente **uma pilha**
  - A pilha é dividida em células, onde cada uma comporta apenas um símbolo
  - O cabeçote de leitura da pilha sempre aponta para o topo



Transições de um Autômato de Pilha

- Suponha um Autômato de Pilha com

  E: conjunto de estados

  Σ: alfabeto de entrada (da fita)

  Γ: alfabeto da pilha

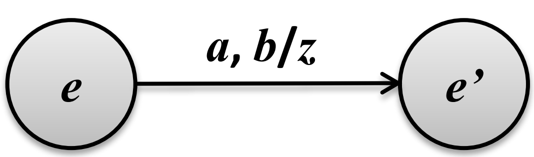
  – Cada transição é da forma

$$\delta(e, a, b) = [e', z]$$

  onde ***e***, ***e'*** **∈ E**, **a ∈ Σ ∪ {λ}**, ***b*** **∈ Γ ∪ {λ}** e ***z*** **∈ Γ***

  – Representado pelo diagrama de estados

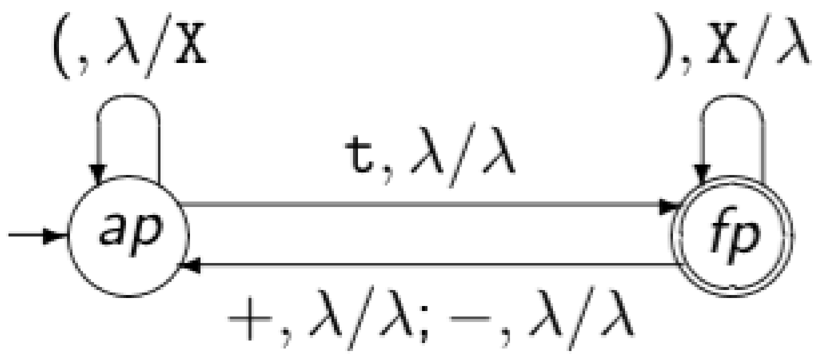
Dica: ***a*** e/ou ***b*** podem ser **λ**



Do estado ***e***, se o símbolo na fita é ***a*** e o símbolo no topo da pilha for ***b***, há uma transição para ***e'***, onde ***b*** é desempilhado e ***z*** empilhado

Exemplo

- Seja o conjunto EA das expressões aritméticas com parênteses e as operações de soma e subtração, definido recursivamente por
  a) ***t* ∈ EA;**
  b) se ***x*,*y* ∈ EA,** então **(*x*) ∈ EA, *x* + *y* ∈ EA** e ***x* –*y* ∈ EA**



Alfabetos:

- $\Sigma = \{ t, (,), +, - \}$
- $\Gamma = \{ X \}$

t representa expressões aritméticas mais básicas, como números inteiros ou de ponto flutuante

Computação

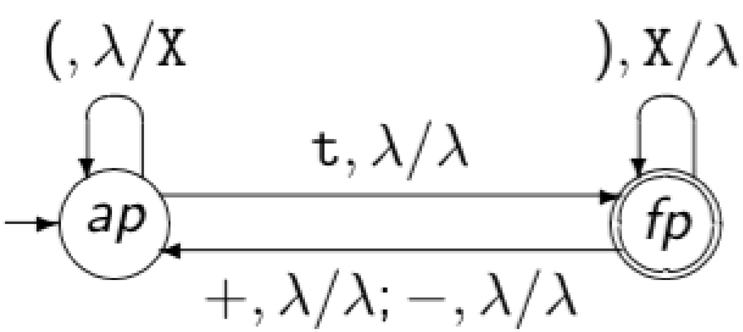
- A configuração instantânea de um autômato de pilha é dado por

$$[e, w, p]$$

onde ***e*** é o estado atual, ***w*** é o restante da palavra de entrada, e ***p*** é a pilha ($p \in \Gamma^*$)

- Exemplo para **(*t* - (*t* + *t*))**

$$\begin{aligned}
[ap, (\mathtt{t} - (\mathtt{t} + \mathtt{t})), \lambda] &\vdash [ap, \mathtt{t} - (\mathtt{t} + \mathtt{t})), X] \\
&\vdash [fp, -(\mathtt{t} + \mathtt{t})), X] \\
&\vdash [ap, (\mathtt{t} + \mathtt{t})), X] \\
&\vdash [ap, \mathtt{t} + \mathtt{t})), XX] \\
&\vdash [fp, +\mathtt{t})), XX] \\
&\vdash [ap, \mathtt{t})), XX] \\
&\vdash [fp,)), XX] \\
&\vdash [fp,), X] \\
&\vdash [fp, \lambda, \lambda]
\end{aligned}$$



Critérios de Reconhecimento

- Um autômato pode parar
  - sem consumir toda a palavra de entrada

$$[ap, \mathtt{t}), \lambda] \vdash [fp,), \lambda]$$

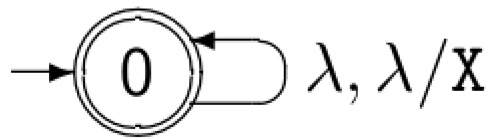
  - em um estado final com a pilha não vazia

$$\begin{aligned}[ap, (\mathtt{t}, \lambda] &\vdash [ap, \mathtt{t}, \mathtt{X}] \\ &\vdash [fp, \lambda, \mathtt{X}]\end{aligned}$$

- Um autômato de pilha **reconhece** uma palavra se
  - A palavra é totalmente consumida
  - A máquina para em estado final
  - A pilha termina vazia

Um Exemplo Estranho

- Considere o autômato de pilha com $\Sigma = \{1\}$, $\Gamma = \{X\}$, com diagrama de estado mostrado a seguir



  – Para toda palavra $\{1\}^+$, o autômato de pilha não para

$$[0, 1, \lambda] \vdash [0, 1, X] \vdash [0, 1, XX] \ldots$$

Para a entrada **λ:**

1) A máquina para ou não?
2) É reconhecida ou não?

AUTÔMATOS DE PILHA DETERMINÍSTICOS (APD)

Transições Compatíveis

- Para que haja **determinismo**, não pode haver duas transições **$\delta(e, a, b)$** e **$\delta(e, a', b')$** definidas para uma mesma configuração instantânea

- Seja $\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \rightarrow E \times \Gamma^*$, as transições $\delta(e, a, b)$ e $\delta(e, a', b')$ são
  - **Transições compatíveis** se, somente se,

$$(a = a' \text{ ou } a = \lambda \text{ ou } a' = \lambda) \text{ e } (b = b' \text{ ou } b = \lambda \text{ ou } b' = \lambda)$$

  - **Transições incompatíveis** se, somente se,

$$(a \neq a' \text{ e } a \neq \lambda \text{ e } a' \neq \lambda) \text{ ou } (b \neq b' \text{ e } b \neq \lambda \text{ e } b' \neq \lambda)$$

Qual é mais fácil de entender?

Autômato de Pilha Determinístico

- Um autômato de pilha determinístico (APD) é uma sêxtupla $(E, \Sigma, \Gamma, \delta, i, F)$, onde
  - **E** é um conjunto finito de um ou mais estados
  - $\boldsymbol{\Sigma}$ é o alfabeto de entrada
  - $\boldsymbol{\Gamma}$ é o alfabeto de pilha
  - $\boldsymbol{\delta}$ é uma função parcial de $E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\})$ para $E \times \Gamma^*$**, sem transições compatíveis**
  - $\mathbf{i \in E}$ é o estado inicial
  - $\mathbf{F \subseteq E}$ é o conjunto de estados finais

Linguagem Reconhecida por APD

- Seja um APD $M = (E, \Sigma, \Gamma, \delta, i, F)$, a relação $\vdash \subseteq (E \times \Sigma^* \times \Gamma^*)^2$ é tal que, $\forall e,e' \in E$, $a \in \Sigma \cup \{\lambda\}$, $b \in \Gamma \cup \{\lambda\}$ e $x \in \Gamma^*$

$$\forall y \in \Sigma^*, z \in \Gamma^*: [e, ay, bz] \vdash [e', y, xz] \boxed{\leftrightarrow} \delta(e, a, b) = [e', x]$$

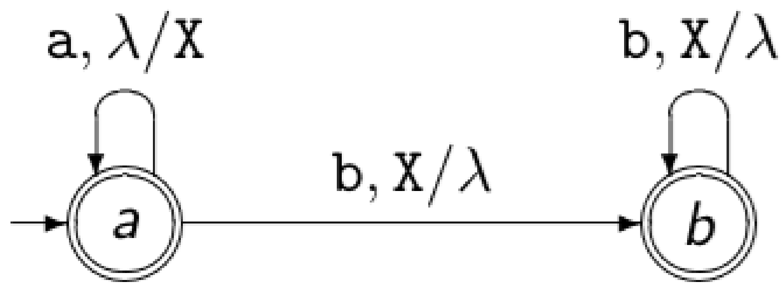
- A linguagem reconhecida pelo APD M é dada por

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

A relação $\overset{*}{\vdash}$ é o fecho transitivo e reflexivo de $\vdash$

Exemplo

- $L = \{ a^n b^n \mid n \in N \}$



  – APD M = ({a, b}, {a, b}, {X}, δ, a, {a, b}), onde as transições **δ** são dadas por

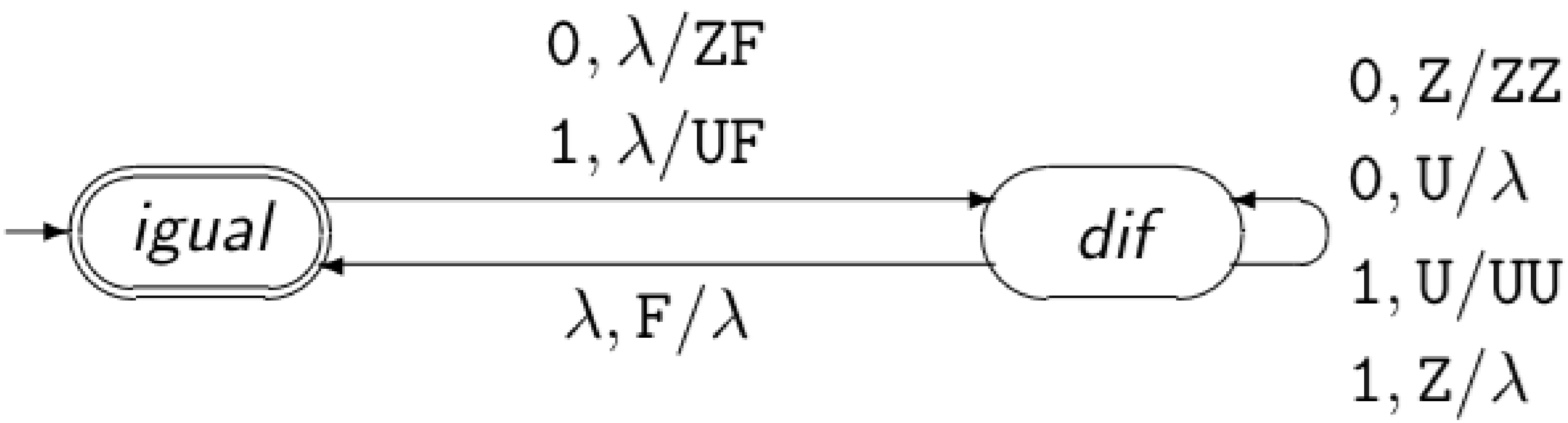
    $\delta(a, \mathrm{a}, \lambda) = [a, \mathrm{X}]$

    $\delta(a, \mathrm{b}, \mathrm{X}) = [b, \lambda]$

    $\delta(b, \mathrm{b}, \mathrm{X}) = [b, \lambda]$

Outro Exemplo

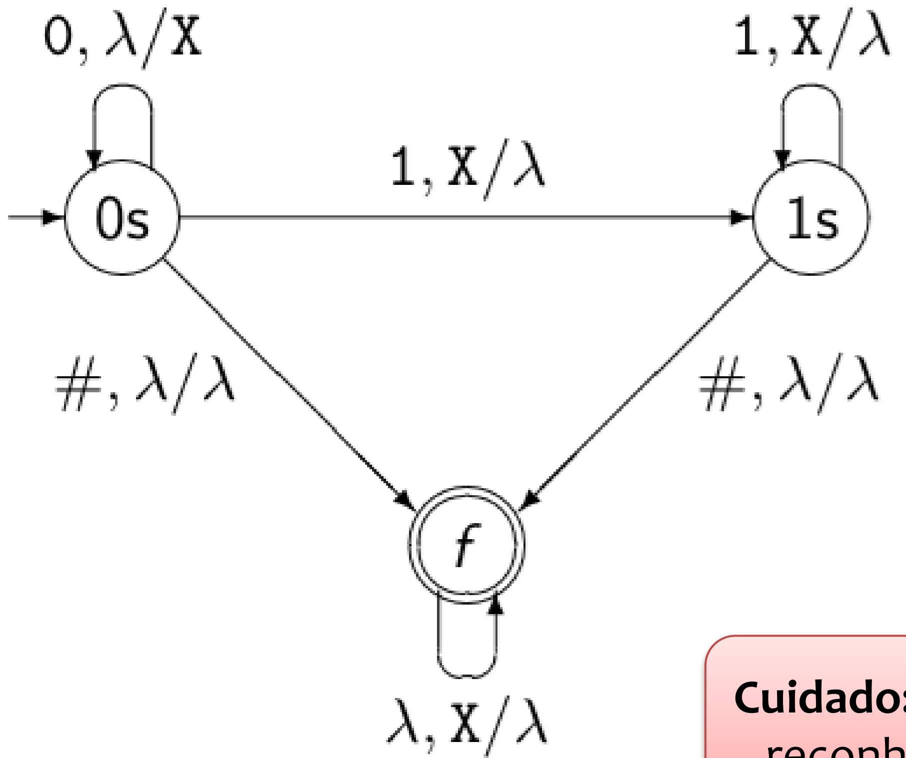
- $L = \{ w \in \{0, 1\}^* \mid$ o número de 0s é igual ao número de 1s em $w \}$



Esse APD utiliza uma técnica para marcar o fundo da pilha (***F***)

Mais um Exemplo

- $L = \{\ 0^m 1^n \# \mid m \geq n\ \}$



Cuidado: APD não é capaz de reconhecer $\{\ 0^m 1^n \mid m \geq n\ \}$

AUTÔMATOS DE PILHA NÃO DETERMINÍSTICOS (APN)

Autômato de Pilha Não Determinístico

- Um autômato de pilha não determinístico (APN) é uma sêxtupla
  $(E, \Sigma, \Gamma, \delta, I, F)$, onde

  – **E, Σ, Γ** e **F** são como em APD's
  – $\boldsymbol{\delta}$ é uma a função parcial de $E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\})$ para D, sendo D constituído dos subconjuntos finitos de $\mathbf{E \times \Gamma^*}$
  – $\mathbf{I \subseteq E}$ é o conjunto de estados iniciais

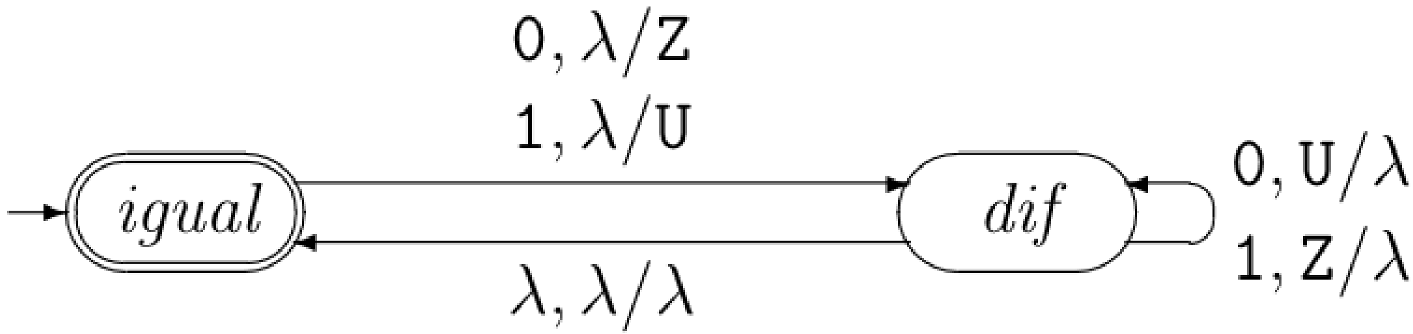
Linguagem Reconhecida por APN

- Seja um APN $M = (E, \Sigma, \Gamma, \delta, I, F)$, a linguagem reconhecida pelo M é dada por

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

Um Mesmo Exemplo

- $L = \{ w \in \{0, 1\}^* \mid$ o número de 0s é igual ao número de 1s em $w \}$

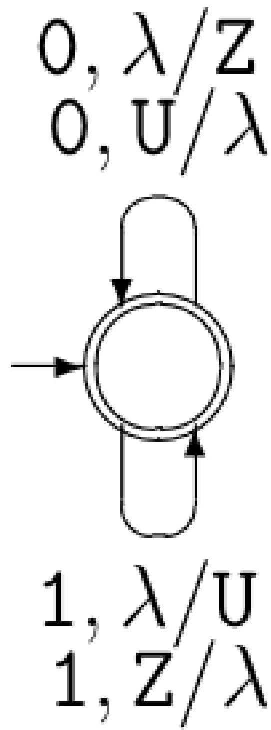


Não foi preciso marcar o fundo da pilha

Tem jeito de fazer melhor?

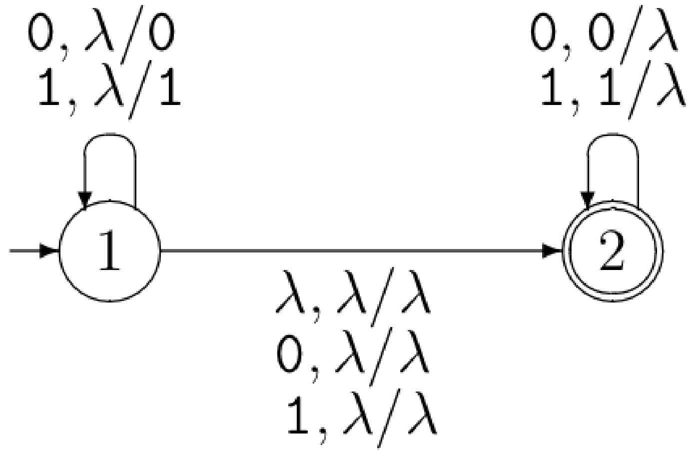
Um Mesmo Exemplo

- $L = \{ w \in \{0, 1\}^* \mid$ o número de 0s é igual ao número de 1s em $w \}$



Exemplo de Palíndromo

- $L = \{ w \in \{0,1\}^* \mid w = w^R\}$



É possível reconhecer essa linguagem por APD?

  - se |w| for par, é percorrida a transição de 1 para 2 sob λ
  - se |w| for ímpar e o símbolo do meio for
    - 0, é percorrida a transição de 1 para 2 sob 0
    - 1, é percorrida a transição de 1 para 2 sob 1

Critérios de Reconhecimento

- Seja um APN $M = (E, \Sigma, \Gamma, \delta, I, F)$, a linguagem $L(M)$ pode ser reconhecida usando os seguintes critérios

  - Por **estado final**

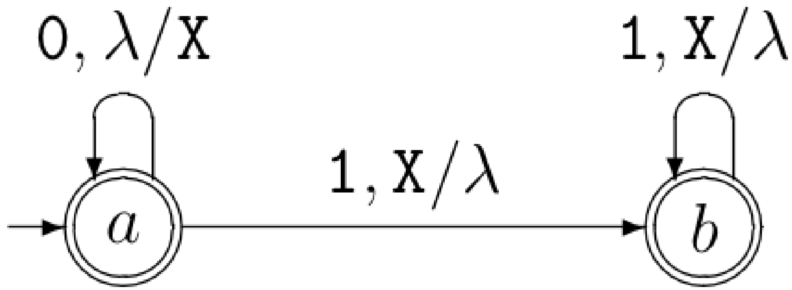
    $L_F(M) = \{ w \in \Sigma^* \mid [i, w, \lambda] \vdash [e, \lambda, y]$,
    onde $y \in \Gamma^*$, e $e \in F$ para algum $i \in I \}$

  - Por **pilha vazia**

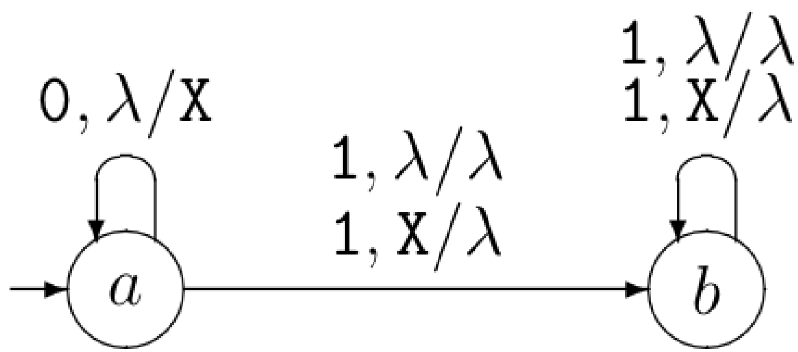
    $L_V(M) = \{ w \in \Sigma^* \mid [i, w, \lambda] \vdash [e, \lambda, \lambda]$ para algum $i \in I \}$

Exemplo

- $L = \{\ 0^m1^n \mid m \geq n\}$, por **estado final**



- $L = \{\ 0^m1^n \mid m \leq n\}$, por **pilha vazia**



Equivalência de Critérios de Reconhecimento

- Seja L uma linguagem, as seguintes afirmativas são **equivalentes**

  a) L pode ser reconhecida por estado final e pilha vazia
  b) L pode ser reconhecida por estado final
  c) $L \cup \{\lambda\}$ pode ser reconhecida por pilha vazia

Será demonstrado que as seguintes transformações são possíveis:
(a) $\rightarrow$ **(b)**, **(b)** $\rightarrow$ **(c)** e **(c)** $\rightarrow$ **(a)**

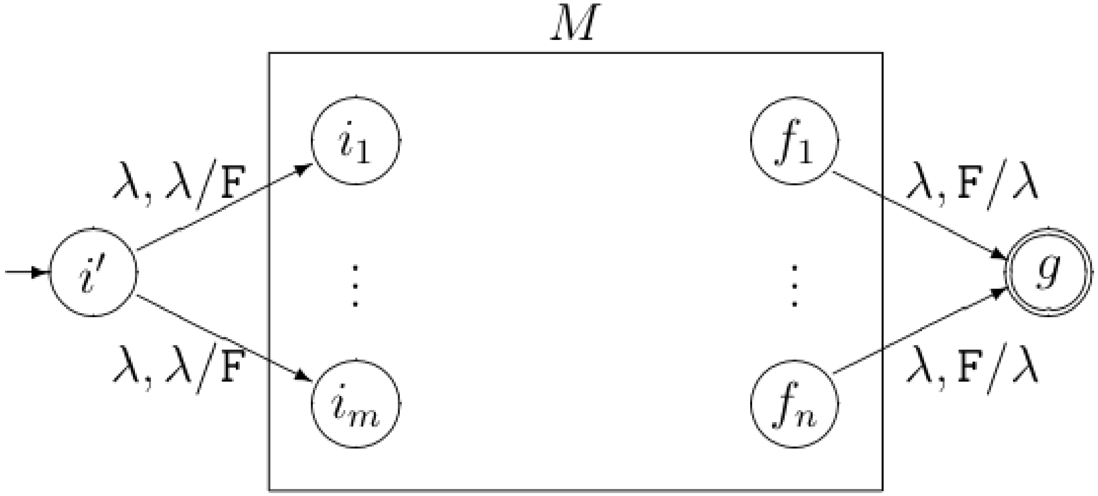
(a) → (b)

- Seja um APN $M = (E, \Sigma, \Gamma, \delta, I, F)$, é possível obter um APN M' tal que $L_F(M') = L(M)$

  $M' = (E \cup \{i', g\}, \Sigma, \Gamma \cup \{F\}, \delta', \{i'\}, \{g\})$

  - $i', g \notin E$ e $F \notin \Gamma$
  - δ' é como δ, mas com as transições
    - $\forall i_k \in I, \delta'(i', \lambda, \lambda) = \{[i_k, F]\}$
    - $\forall f_j \in F, \delta'(f_j, \lambda, F) = \{[g, \lambda]\}$

A ideia é utilizar um símbolo para marcar o fundo da pilha

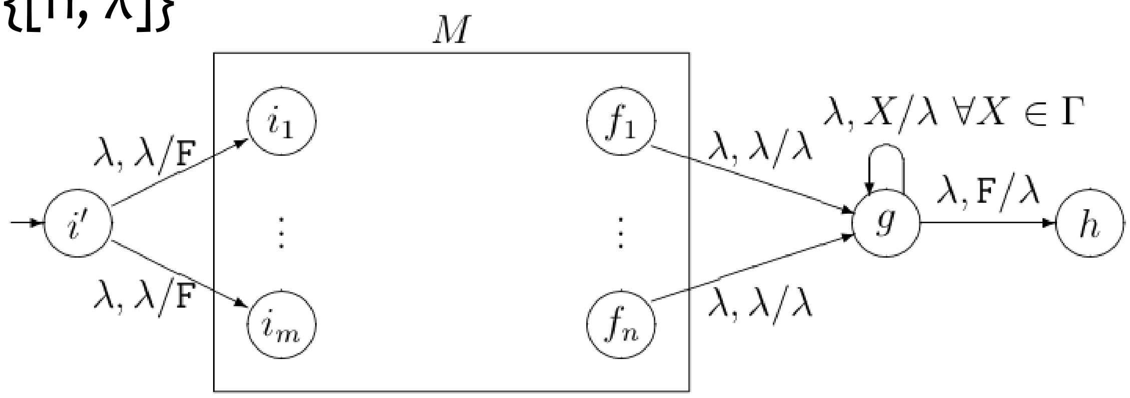


(b) → (c)

- Seja um APN $M = (E, \Sigma, \Gamma, \delta, I, F)$, é possível obter um APN M' tal que $L_V(M') = L_F(M) \cup \{\lambda\}$

  $M' = (E \cup \{i', g, h\}, \Sigma, \Gamma \cup \{F\}, \delta', \{i'\})$

  - $i', g, h \notin E$ e $F \notin \Gamma$
  - δ' é como δ, mas com as transições
    - $\forall i_k \in I, \delta'(i', \lambda, \lambda) = \{[i_k, F]\}$
    - $\forall f_j \in F, \delta'(f_j, \lambda, F) = \{[g, \lambda]\}$
    - $\forall X \in \Gamma, \delta'(g, \lambda, X) = \{[g, \lambda]\}$
    - $\delta(g, \lambda, F) = \{[h, \lambda]\}$



(c) $\rightarrow$ (a)

- Um APN que reconhece por pilha vazia é um APN que reconhece por pilha vazia e estado final, basta considerar todos seus estados como estados finais

- $L(M') = L_V(M)$

$$\text{Se } M = (E, \Sigma, \Gamma, \delta, I), \text{ então } M' = (E, \Sigma, \Gamma, \delta, I, E)$$