Linguagens Formais e Autômatos

Aula 18 - Determinismo em PDAs

Prof. Dr. Daniel Lucrédio
Departamento de Computação / UFSCar
Última revisão: ago/2015

Referências bibliográficas

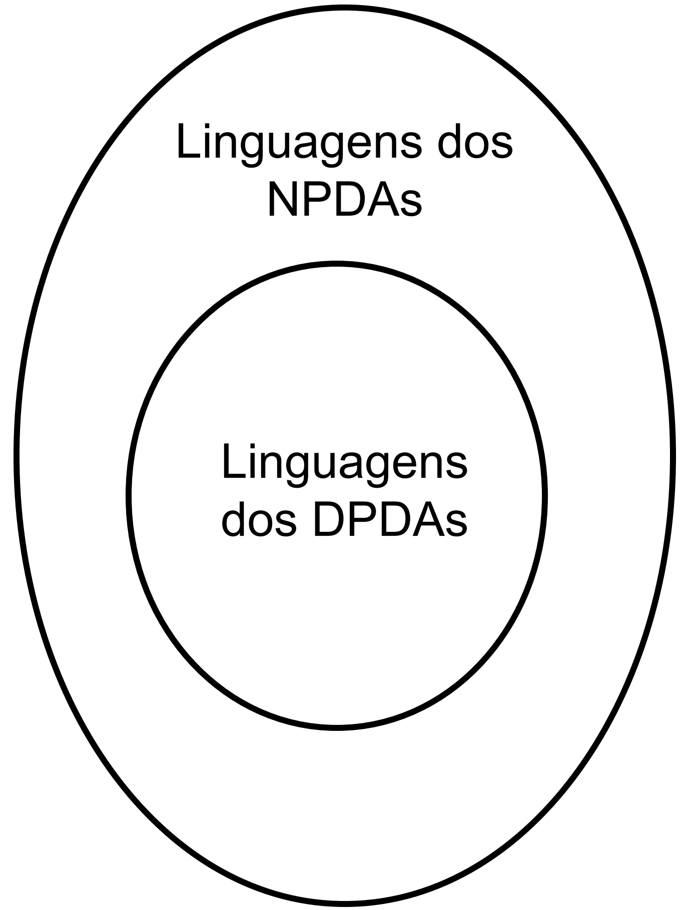
- **Introdução à teoria dos autômatos, linguagens e computação / John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman** ; tradução da 2.ed. original de Vandenberg D. de Souza. - Rio de Janeiro : Elsevier, 2002 (Tradução de: Introduction to automata theory, languages, and computation - ISBN 85-352-1072-5)
  - Capítulo 6 - Seção 6.4

Determinismo em PDAs

- O não-determinismo é um bom “truque de programação”, pois ajuda a projetar linguagens/autômatos
  - Para autômatos finitos, o não-determinismo não dá poder aos autômatos, ou seja:
    - NFAs reconhecem as mesmas linguagens que DFAs
    - É possível converter NFAs em DFAs e vice-versa
  - A escolha de quando converter (em tempo de execução ou pré-execução) fica a cargo do projetista, e depende do cenário de aplicação

Determinismo em PDAs

- Para autômatos de pilha, o mesmo não acontece
- O não-determinismo AUMENTA a capacidade reconhecedora de um PDA
  - Ou seja, PDAs determinísticos (DPDAs) reconhecem menos linguagens do que PDAs não-determinísticos (NPDAs)
    - Equivalente a dizer que existem linguagens reconhecidas por NPDAs que não são reconhecidas por DPDAs



PDAs determinísticos

- Definição informal é a mesma do que para autômatos finitos determinísticos
  - Ou seja, em um DPDA, nunca há alternativa de escolha para movimento, e sempre o autômato sabe o que fazer
  - Isso significa que, para toda combinação de entrada, estado e topo da pilha, há uma e somente uma possibilidade de transição
- Importante: a transição vazia (ε) aqui não é indicativo de não-determinismo!!
  - Pois pode haver uma decisão com base no topo da pilha.
  - O problema é haver conflito entre consumir a entrada ou não!

PDAs determinísticos

- O caso da entrada vazia
    - Para um determinado topo da pilha X, e uma entrada a
    - O DPDA:
        - Ou define uma transição com base em a (e a transição com base em ε fica vazia)
        - Ou define uma transição com base em ε (e a transição com base em a fica vazia)
- Na tabela, as células correspondentes às colunas ε nunca sobrepõem o que foi definido nas células das colunas das entradas

PDAs determinísticos

Na primeira linha, as colunas ε estão todas vazias, pois as combinações 0 x {0,1,$} e 1 x {0,1,$} foram definidas

Entrada:	0			1			ε		
Pilha:	0	1	$	0	1	$	0	1	$
q0	(q0,00)	(q0,01)	(q0,0$)	(q0,10)	(q0,11)	(q0,1$)	Ø	Ø	Ø
q1	(q1,ε)	Ø	(q2,11)	(q1,ε)	Ø	(q1,ε)	Ø	(q2,01)	Ø
q2	Ø	Ø	(q2,11)	Ø	Ø	(q1,0$)	(q1,0)	(q1,0)	Ø

PDAs determinísticos

Na segunda linha, a coluna (ε,1) não está vazia, pois
as combinações 0 x {1} e 1 x {1} não foram definidas

Entrada:	0			1			ε		
Pilha:	0	1	$	0	1	$		1	$
q0	(q0,00)	(q0,01)	(q0,0$)	(q0,10)	(q0,11)	(q0,1$)	Ø	Ø	Ø
q1	(q1,ε)	Ø	(q2,11)	(q1,ε)	Ø	(q1,ε)	Ø	(q2,01)	Ø
q2	Ø	Ø	(q2,11)	Ø	Ø	(q1,0$)	(q1,0)	(q1,0)	Ø

PDAs determinísticos

Entrada:	0			1			ε		
Pilha:	0	1	$	0	1	$	0	1	$
q0	(q0,00)	(q0,01)	(q0,0$)	(q0,10)	(q0,11)	(q0,1$)	Ø	Ø	Ø
q1	(q1,ε)	Ø	(q2,11)	(q1,ε)	Ø	(q1,ε)	Ø	(q2,01)	Ø
q2	Ø	Ø	(q2,11)	Ø	Ø	(q1,0$)	(q1,0)	(q1,0)	Ø

Na terceira linha, as colunas (ε,0) e (ε,1) não estão vazias, pois as combinações 0 x {0,1} e 1 x {0,1} não foram definidas

PDAs determinísticos

- Outra maneira de "enxergar" o determinismo é fazer o seguinte:
  - 1: "Recorte" as colunas do ε

0			1		
0	1	$	0	1	$
(q0,00)	(q0,01)	(q0,0$)	(q0,10)	(q0,11)	(q0,1$)
(q1,ε)		(q2,11)	(q1,ε)		(q1,ε)
		(q2,11)			(q1,0$)

ε		
0	1	$
	(q2,01)	
(q1,0)	(q1,0)	

PDAs determinísticos

- 2: Ctrl-C Ctrl-V no pedaço recortado, quantas vezes for necessário para obter o mesmo número de "cópias" do que entradas

0			1		
0	1	$	0	1	$
(q0,00)	(q0,01)	(q0,0$)	(q0,10)	(q0,11)	(q0,1$)
(q1,ε)		(q2,11)	(q1,ε)		(q1,ε)
		(q2,11)			(q1,0$)

ε		
0	1	$

ε		
0	1	$
	(q2,01)	
(q1,0)	(q1,0)	

Neste caso, criamos duas cópias do "pedaço" recortado, pois temos duas entradas, 0 e 1

PDAs determinísticos

- 3: Coloque cada cópia dos pedaços recortados sobre cada entrada

0 ε			1 ε		
0	1	$	0	1	$
(q0,00)	(q0,01)	(q0,0$)	(q0,10)	(q0,11)	(q0,1$)
(q1,ε)	(q2,01)	(q2,11)	(q1,ε)	(q2,01)	(q1,ε)
(q1,0)	(q1,0)	(q2,11)	(q1,0)	(q1,0)	(q1,0$)

ε		
0	1	$
	(q2,01)	
(q1,0)	(q1,0)	

ε		
0	1	$
	(q2,01)	
(q1,0)	(q1,0)	

PDAs determinísticos

- 4: A tabela resultante deve estar completamente preenchida
  - E nenhuma célula deve conter valores sobrepostos
- Caso contrário, existe um ponto de não-determinismo

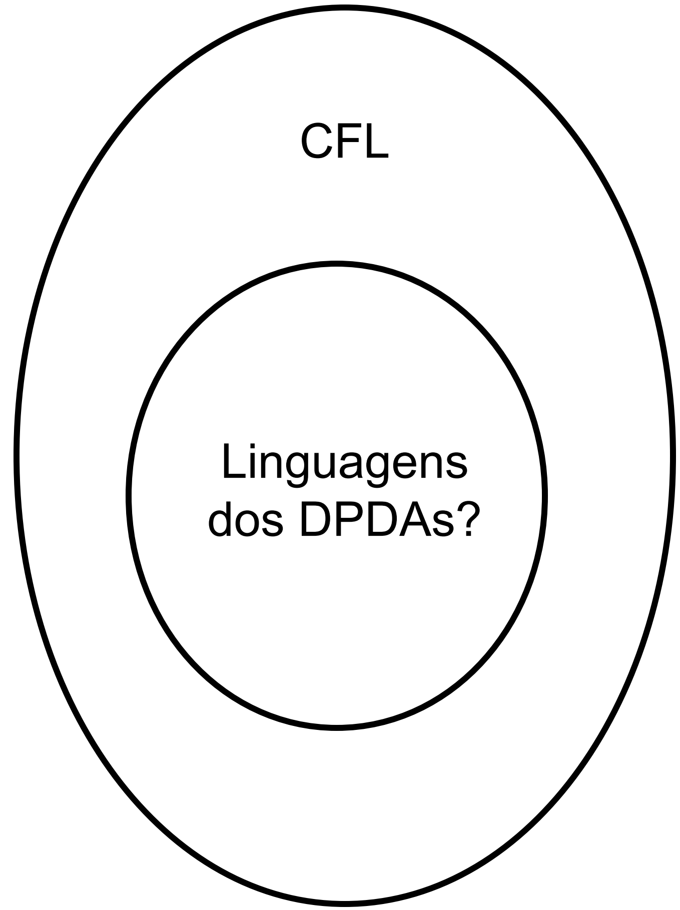
0 ε			1 ε		
0	1	$	0	1	$
(q0,00)	(q0,01)	(q0,0$)	(q0,10)	(q0,11)	(q0,1$)
(q1,ε)	(q2,01)	(q2,11)	(q1,ε)	(q2,01)	(q1,ε)
(q1,0)	(q1,0)	(q2,11)	(q1,0)	(q1,0)	(q1,0$)

PDAs determinísticos

- Definição formal:
- Um PDA $P = (Q,\Sigma,\Gamma,\delta_F,q0,Z_0,F)$ é determinístico se e somente se:
  - $\delta(q, a, X)$ tem no máximo um elemento para qualquer q em Q, a em Σ ou $a = \varepsilon$ e X em Γ
  - Se $\delta(q, a, X)$ é não vazio para algum a em Σ
    - então $\delta(q, \varepsilon, X)$ deve ser vazio

Determinismo em PDAs

- Até agora, vimos somente NPDAs
- Vimos que os NPDAs aceitam as linguagens livres de contexto
- Então quais seriam as linguagens dos DPDAs?

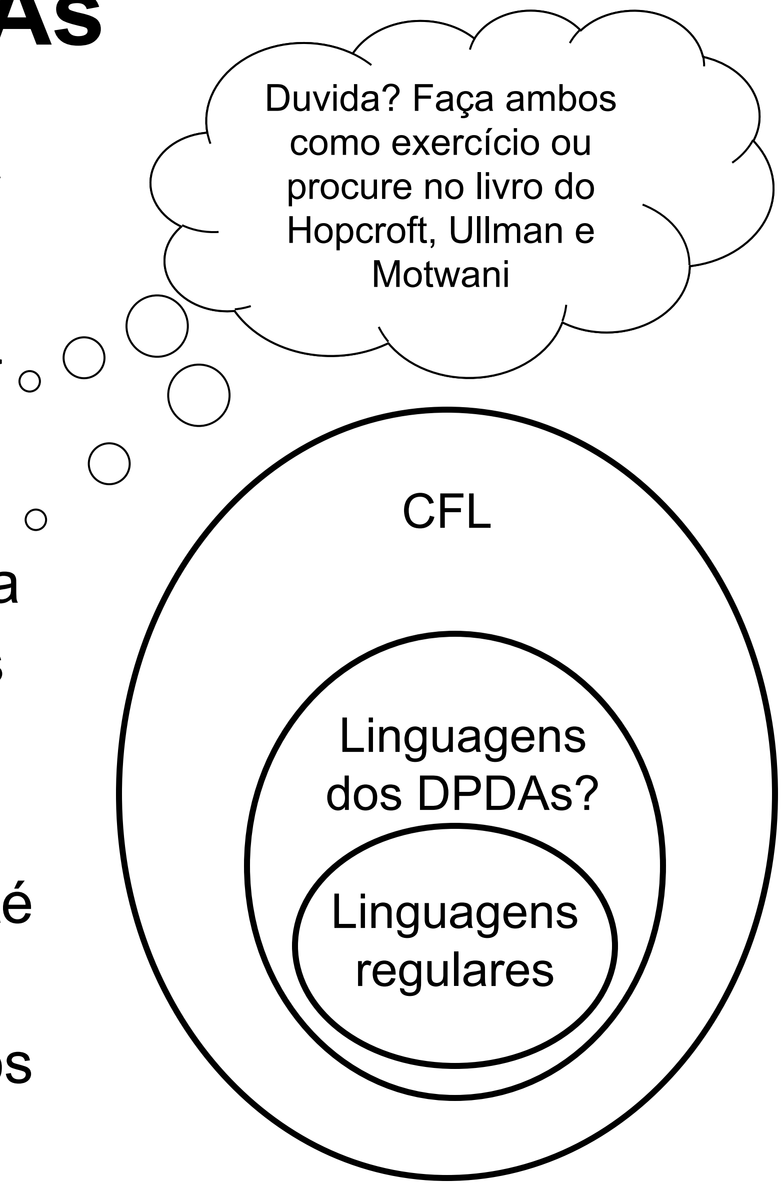


Linguagens dos DPDAs

- Bom, sabemos que as linguagens regulares são aceitas por DPDAs
  - Como? Prova por construção: basta construir um DPDA que sempre “ingora” a pilha
    - Na tabela, as colunas ε ficam vazias
  - O comportamento será exatamente o mesmo que um DFA

Linguagens dos DPDAs

- Existem linguagens aceitas por DPDAs que não são regulares
  - Ex: $L = \{wcw^R \mid w$ está em $(0+1)^*\}$
  - É possível construir um DPDA para essa linguagem
- É possível mostrar (usando o lema do bombeamento para linguagens regulares) que L não é regular
- Está acompanhando? O desenho à direita mostra o que sabemos até o momento sobre as linguagens aceitas pelos PDAs determinísticos



Linguagens dos DPDAs

- Além disso, sabemos que as linguagens aceitas pelos DPDAs não são inerentemente ambíguas
  - Ou seja, existe pelo menos uma gramática não-ambígua para suas linguagens
- A prova envolve o processo de conversão de PDAs para gramáticas
  - É possível mostrar que o processo de conversão de um DPDA para uma CFG produz uma gramática onde todas as derivações mais à esquerda que levam a cadeias da linguagem são únicas

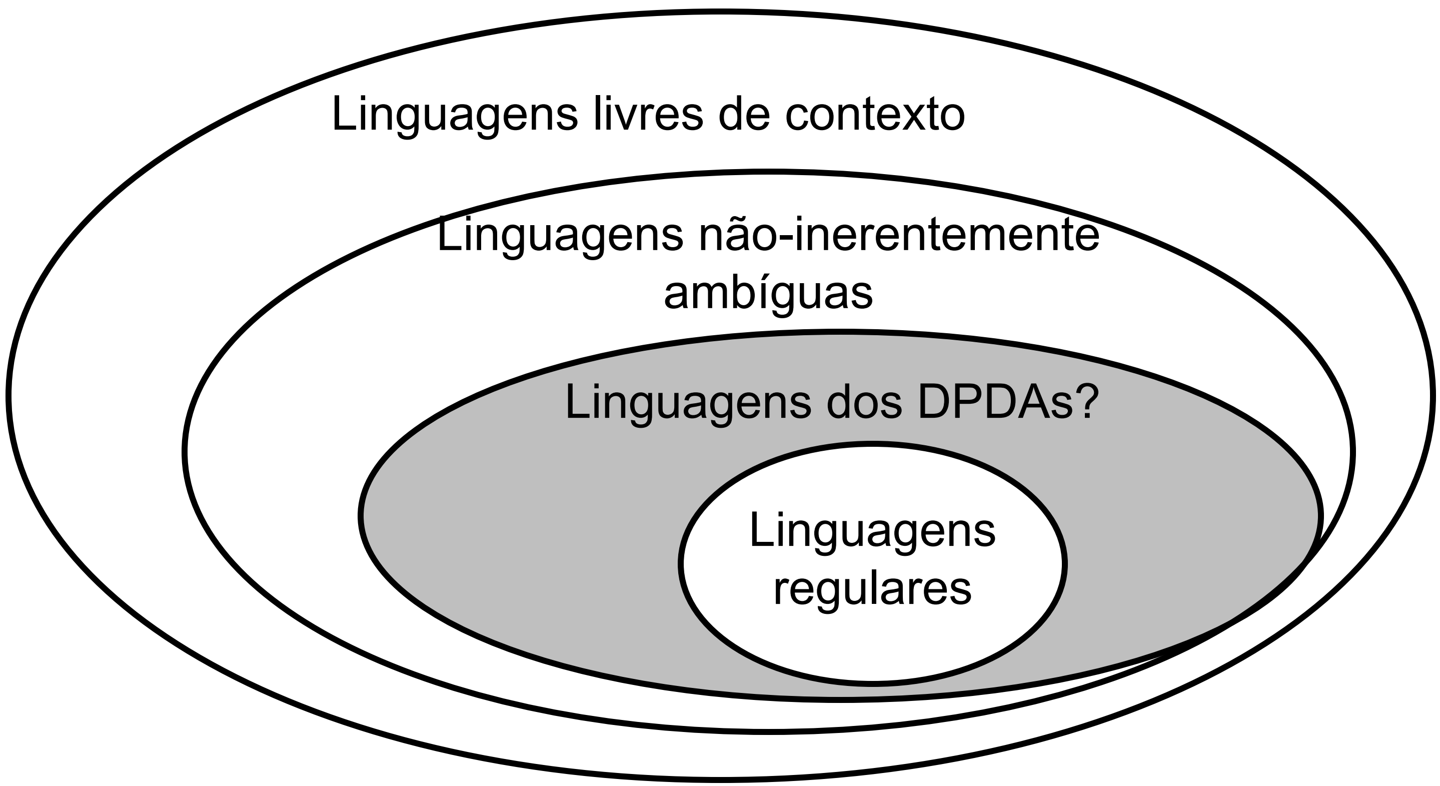
Não vimos essa conversão!!!

Linguagens dos DPDAs

- Mas as linguagens aceitas pelos DPDAs não são TODAS as linguagens não-inerentemente ambíguas
  - Ou seja, existem linguagens não-inerentemente ambíguas (que possuem pelo menos uma gramática não-ambígua) que NÃO são aceitas por um DPDA
  - Exemplo: $S \rightarrow 0S0 \mid 1S1 \mid \varepsilon$ (gramática dos palíndromos com número par de símbolos)
  - É necessário um ponto de não-determinismo para “adivinhar” quando terminou de ler a primeira metade da entrada
    - Ou seja, não existe um DPDA que aceita essa linguagem

Linguagens dos DPDAs

- Resumindo o que sabemos sobre as linguagens aceitas pelos DPDAs



Fim

Aula 18 - Determinismo em PDAs