LINGUAGENS FORMAIS: Teoria, Modelagem e Implementação

Errata, versão do dia 14 de abril de 2018 às 10:22

1. Página 31
   onde se lê: “Um conjunto é uma coleção de símbolos, também denominados átomos ou elementos...”
   leia-se: “Um conjunto é uma coleção de elementos...”

2. Página 31, Exemplo 1.1
   onde se lê: “A inclusão do símbolo...”
   leia-se: “A inclusão do elemento...”

3. Página 31, inserir imediatamente depois do Exemplo 1.2
   “Símbolos podem ser agrupados na forma de um conjunto, caso em que o mesmo recebe o nome de alfabeto. Conjuntos, por outro lado, podem ser formados por elementos de outra natureza, e não apenas por símbolos. É o caso, por exemplo, de conjuntos formados por cadeias (seqüências finitas de símbolos) e conjuntos cujos elementos também são conjuntos.”

4. Página 32, Exemplo 1.7
   onde se lê: “$\{1,2,3,5,7,11,13,17...\}$”
   leia-se: “$\{2,3,5,7,11,13,17...\}$”

5. Página 36, Exemplo 1.20
   onde se lê: “$(1,1,1),(-1,-1,-1)$”
   leia-se: “$(2,2,0),(2,0,-2)$”

6. Página 37, Exemplo 1.23
   onde se lê: “$A_0 = \{0,0\}$”
   leia-se: “$A_0 = \{0\}$”

7. Página 48, Exemplo 1.45
   onde se lê: “Como $2x-1 < 3x, \forall x \geq 1$ (porque $3x = (2+1)x = 2x+2$, logo $2x-1 < 2x+2$ e ...”
   leia-se: “Como $2x-1 < 3x, \forall x \geq 1$ (porque $3x = (2+1)x = 2x+x$, logo $2x-1 < 2x+x$ e ...”

8. Página 50, último parágrafo
   onde se lê: “Dessa forma, a hipótese efetuada não pode ser considerada falsa, devendo, portanto, ser considerada verdadeira”
   leia-se: “Dessa forma, a hipótese negada não pode ser considerada verdadeira, devendo, portanto, ser considerada falsa, e, conseqüentemente, a hipótese original deve ser considerada verdadeira”

9. Página 53, Exemplo 1.55
   onde se lê: “O fato de que $|S| = |R|$ ...”
   leia-se: “O fato de que $|S| = |\mathbb{R}|$ ...”

10. Página 54, legenda Tabela 1.5
    onde se lê: “Bijeção hipotética entre $\mathbb{N}$ e $\mathbb{R}$”
    leia-se: “Bijeção hipotética entre $\mathbb{N}$ e S”

11. Página 54, primeira linha, primeira coluna da Tabela 1.5
    onde se lê: “$\mathbb{R}$”
    leia-se: “S”

12. Página 54, no meio da página
    onde se lê: “$x_j \neq d_{jj}$”
    leia-se: “$x_j \neq d_{j_j}$”

13. Página 56, Tabela 1.6
onde se lê: “Função f_1 para o Teorema 1.7”
leia-se: “Função f_1 para o Teorema 1.4”

14. Página 56, Tabela 1.7
onde se lê: “Função f_2 para o Teorema 1.7”
leia-se: “Função f_2 para o Teorema 1.4”

15. Página 56, Teorema 1.6
onde se lê: “Sejam A e B dois conjuntos quaisquer. Se $|A| = \aleph_0$ e $|B| = \aleph_0$, então $|A \cap B| = \aleph_0$”
leia-se: “Sejam A e B dois conjuntos quaisquer. Se $|A| = \aleph_0$ e $|B| = \aleph_0$, então $|A \cap B| \leq \aleph_0$”

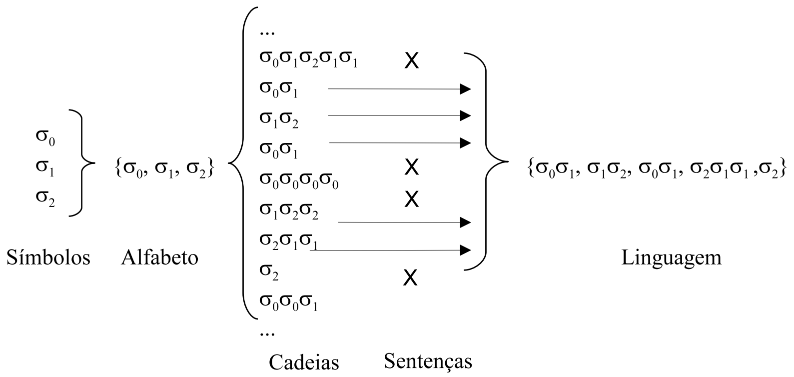
16. Página 56, Teorema 1.6
onde se lê: “Finalmente, se nenhuma dessas duas condições for verdadeira, então $A \cap B \subseteq A$ e, pelo Teorema 1.7, $|A \cap B| = \aleph_0$. Portanto, em qualquer caso que se considere, $|A \cap B| = \aleph_0$”
leia-se: “Finalmente, se nenhuma dessas duas condições for verdadeira, então $(A \cap B) \subseteq A$ e, pelo Teorema 1.4, $|A \cap B| \leq \aleph_0$. Portanto, em qualquer caso que se considere, $|A \cap B| \leq \aleph_0$”

17. Página 56, Tabela 1.8
onde se lê: “Composição de f_1 com f_2 para o Teorema 1.7”
leia-se: “Composição de f_1 com f_2 para o Teorema 1.4”

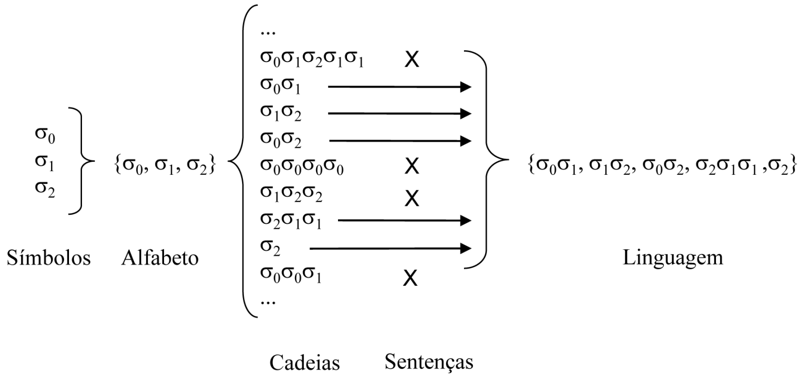
18. Página 57, Teorema 1.7
onde se lê: “Então, de acordo com o Teorema 1.7 ...”
leia-se: “Então, de acordo com o Teorema 1.5 ...”

19. Página 78, Exemplo 2.2
onde se lê: “Considerem-se as cadeias $\alpha = 1, \beta = 469, \chi = bce60, \phi = df$.”
leia-se: “Considerem-se as cadeias $\alpha = 1, \beta = 469, \chi = bce60$ e $\phi = df$, também construídas sobre o alfabeto Σ do Exemplo 2.1.”.

20. Página 81, Figura 2.2
onde se lê:



leia-se:



21. Página 81
    onde se lê: $\Sigma^i = \Sigma\Sigma^{i-1}, i \geq 0$
    leia-se: $\Sigma^i = \Sigma\Sigma^{i-1}, i > 0$

22. Página 88, Exemplo 2.19
    onde se lê: “$n * (n(n+n+n))$”
    leia-se: “$n * (n/(n+n+n))$”
    onde se lê: “$L_1 = \{n, n+n, (n*n), n*(n(n+n+n))\}$”
    leia-se: “$L_1 = \{n, n+n, (n*n), n*(n/(n+n+n))\}$”

23. Página 90, linha 5
    onde se lê: “$L_1 \subset L_2$”
    leia-se: “$L_2 \subset L_1$”

24. Página 93, Exemplo 2.22
    onde se lê: “$0S33 \Rightarrow 00A3333$”
    leia-se: “$0S33 \Rightarrow^* 00A3333$”

25. Página 146, Exemplo 3.4
    onde se lê: “N”
    leia-se: “$\mathbb{N}$” (cinco ocorrências)

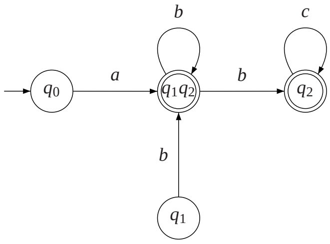
26. Página 146, Exemplo 3.4
    onde se lê: “$DD^*.D^* \cup D^*.DD^*$”
    leia-se: “$DD^*\{.\}D^* \cup D^*\{.\}DD^*$” (duas ocorrências)

27. Página 148, Exemplo 3.8
    onde se lê: “$\varepsilon|a|aaaa^*$ e $(aa)^*a(aaa)^*$”
    leia-se: “$a|aaaa^*$ e $(aa)^*a(aaa)^*$”

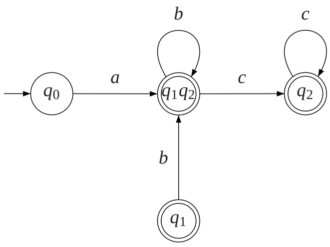
28. Página 153, Exemplo 3.9
    onde se lê: “Portanto, $(q_0, 0011222) \vdash^* (q_0, \varepsilon)$, e ...”
    leia-se: “Portanto, $(q_0, 0011222) \vdash^* (q_2, \varepsilon)$, e ...”

29. Página 161, Algoritmo 3.4, item 5
    onde se lê: “$\delta' \leftarrow \emptyset$”
    leia-se: “$\delta_2 \leftarrow \emptyset$”

30. Página 162, Figura 3.11
onde se lê:



leia-se:



31. Página 164, Primeiro parágrafo
onde se lê: "A Figura 3.13 apresenta o diagrama de estados do autômato determinístico obtido. O estado q_3, que é inacessível, não está mostrado na figura."
leia-se: "A Figura 3.13 apresenta o diagrama de estados do autômato determinístico obtido."

32. Página 165, Tabela 3.9
onde se lê:

	δ	a	b	c
$\rightarrow$	q_0	q_1q_2		
$\leftarrow$	q_1	q_0	q_0q_1	
$\leftarrow$	q_2			q_2
	q_3	q_2	q_1	
$\leftarrow$	q_1q_2	q_0	q_0q_1	q_2
$\leftarrow$	q_0q_1	$q_0q_1q_2$	q_0q_1	
$\leftarrow$	$q_0q_1q_2$	$q_0q_1q_2$	q_0q_1	$\{q_2, q_3\}$

leia-se:

	δ	a	b	c
$\rightarrow$	q_0	q_1q_2		q_3
$\leftarrow$	q_1	q_0	q_0q_1	
$\leftarrow$	q_2			q_2
	q_3	q_2	q_1	
$\leftarrow$	q_1q_2	q_0	q_0q_1	q_2
$\leftarrow$	q_0q_1	$q_0q_1q_2$	q_0q_1	q_3
$\leftarrow$	$q_0q_1q_2$	$q_0q_1q_2$	q_0q_1	$\{q_2,q_3\}$

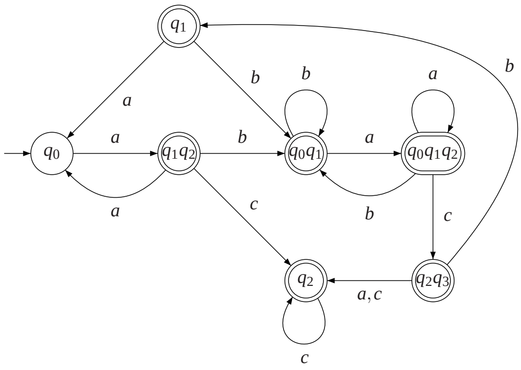
33. Página 165, Tabela 3.10
    onde se lê:

	δ	a	b	c
$\rightarrow$	q_0	q_1q_2		
$\leftarrow$	q_1	q_0	q_0q_1	
$\leftarrow$	q_2			q_2
	q_3	q_2	q_1	
$\leftarrow$	q_1q_2	q_0	q_0q_1	q_2
$\leftarrow$	q_0q_1	$q_0q_1q_2$	q_0q_1	
$\leftarrow$	$q_0q_1q_2$	$q_0q_1q_2$	q_0q_1	q_2q_3
$\leftarrow$	q_2q_3	q_2	q_1	q_2

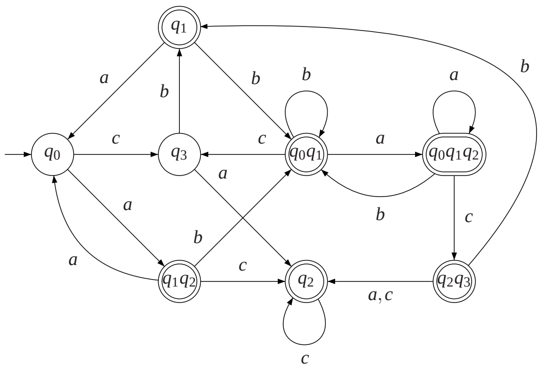
leia-se:

	δ	a	b	c
$\rightarrow$	q_0	q_1q_2		q_3
$\leftarrow$	q_1	q_0	q_0q_1	
$\leftarrow$	q_2			q_2
	q_3	q_2	q_1	
$\leftarrow$	q_1q_2	q_0	q_0q_1	q_2
$\leftarrow$	q_0q_1	$q_0q_1q_2$	q_0q_1	q_3
$\leftarrow$	$q_0q_1q_2$	$q_0q_1q_2$	q_0q_1	q_2q_3
$\leftarrow$	q_2q_3	q_2	q_1	q_2

34. Página 166, Figura 3.13
    onde se lê:



leia-se:



35. Página 167, Tabela 3.12
onde se lê:

	δ	a	b
$\rightarrow$	q_0	q_1q_2	
$\leftarrow$	q_1		
	q_2	q_0q_2	q_0q_1
$\leftarrow$	q_1q_2	q_0q_2	q_0q_1
$\leftarrow$	q_0q_1	q_1q_2	
	q_0q_2	q_1q_2, q_0q_2	q_0q_1
$\leftarrow$	$q_0q_1q_2$	$q_0q_1q_2$	q_0q_1

leia-se:

	δ	a	b
$\rightarrow$	q_0	q_1q_2	
$\leftarrow$	q_1		
	q_2	q_0q_2	q_0q_1
$\leftarrow$	q_1q_2	q_0q_2	q_0q_1
$\leftarrow$	q_0q_1	q_1q_2	
	q_0q_2	$q_0q_1q_2$	q_0q_1
$\leftarrow$	$q_0q_1q_2$	$q_0q_1q_2$	q_0q_1

36. Página 169, imediatamente antes do Teorema 3.4
acrescentar:
"Serão apresentados dois algoritmos para essa finalidade. O primeiro, mais simples e intuitivo, aceita como entrada qualquer autômato finito que não possua ciclos de transições em vazio. O segundo, mais robusto, aceita qualquer tipo de autômato finito como entrada."

37. Página 169, Algoritmo 3.5
onde se lê: "Entrada: um autômato finito com transições em vazio M;"
leia-se: ""Entrada: um autômato finito com transições em vazio M, porém isento de ciclos formados por transições em vazio;"

38. Página 177, última linha
onde se lê: "adotoda"
leia-se: "adotada"

39. Página 193
os parágrafos que iniciam com "Suponha-se..." e "Admita-se..." devem ser indentados da mesma forma que o parágrafo anterior "Por definição...", pois eles não estão vinculados ao item "$\{\sigma\}$ é uma linguagem linear à direita...".

40. Página 194
acrescentar, antes da frase "é tal que $L(G_z) = XY$", os itens:
"c) Se $A \rightarrow \varepsilon \in P_x$, então $A \rightarrow S_y \in P_z$"
"d) Se $A \rightarrow B \in P_x$, então $A \rightarrow B \in P_z$"
acrescentar, antes da frase "é tal que $L(G_z) = X^*$", os itens:
"c) Se $A \rightarrow \varepsilon \in P_x$, então $A \rightarrow S_z \in P_z$"
"d) Se $A \rightarrow B \in P_x$, então $A \rightarrow B \in P_z$"

41. Página 198, Exemplo 3.33
onde se lê: $G_0 = (\{a,b,c,X_0,X_1,X_2\},\{a,b,c,d\},P_0,X_0)$
leia-se: $G_0 = (\{a,b,c,d,X_0,X_1,X_2\},\{a,b,c,d\},P_0,X_0)$

42. Página 205, Algoritmo 3.15, item 3
remover a linha que contém o comando "$F \leftarrow \emptyset$"

43. Página 214, legenda da Figura 3.49
onde se lê: "$x^*yz^*(wx^*yz^*)$"
leia-se: "$x^*yz^*(wx^*yz^*)^*$"

44. Página 214, última linha
onde se lê: "$x^*yz^*(wx^*yz^*)$"
leia-se: "$x^*yz^*(wx^*yz^*)^*$"

45. Página 222, Tabela 3.41
substituir "(f)" por "$\leftarrow$" (duas ocorrências)

46. Página 224, terceiro parágrafo
onde se lê: "Seja, portanto, M um autômato finito com n estados."
leia-se: "Seja, portanto, M um autômato finito com $n+1$ estados."

47. Página 225, legenda da Tabela 3.42
onde se lê: "Representação dos pares de estados de um autômato M com n estados"
leia-se: "Representação dos pares de estados de um autômato M com $n+1$ estados"

48. Página 226, Tabela 3.43
substituir "(f)" por "$\leftarrow$" (três ocorrências)

49. Página 226, Exemplo 3.44
remover todas as referências ao estado q_5; especificamente

onde se lê:

	δ	a	b
$\rightarrow$	q_0	q_1	q_6
	q_1	q_2	q_3
(f)	q_2	q_2	q_3
	q_3	q_4	q_2
(f)	q_4	q_2	q_3
(f)	q_5	q_4	q_5
	q_6	q_4	q_4

leia-se:

	δ	a	b
$\rightarrow$	q_0	q_1	q_6
	q_1	q_2	q_3
(f)	q_2	q_2	q_3
	q_3	q_4	q_2
(f)	q_4	q_2	q_3
	q_6	q_4	q_4

onde se lê:

	q_1	q_2	q_3	q_4	q_5	q_6
q_0		$\neq$		$\neq$	$\neq$	
q_1	-	$\neq$		$\neq$	$\neq$	
q_2	-	-	$\neq$			$\neq$
q_3	-	-	-	$\neq$	$\neq$	
q_4	-	-	-	-		$\neq$
q_5	-	-	-	-	-	$\neq$

leia-se:

	q_1	q_2	q_3	q_4	q_6
q_0		$\neq$		$\neq$	
q_1	-	$\neq$		$\neq$	
q_2	-	-	$\neq$		$\neq$
q_3	-	-	-	$\neq$	
q_4	-	-	-	-	$\neq$

50. Página 227, Exemplo 3.44
remover todas as referências ao estado q_5; especificamente:

remover os dois últimos itens da lista;

no último parágrafo:

onde se lê:
"$\{q_3, q_6\}$ e $\{q_5\}$"
leia-se:
"$\{q_3, q_6\}$"

onde se lê:
"possui cinco estados"
leia-se:
"possui quatro estados"

onde se lê:
"$[q_3, q_6]$ e $[q_5]$"
leia-se:
"$[q_3, q_6]$"

51. Página 228, Exemplo 3.44
remover todas as referências ao estado q_5; especificamente:

onde se lê:

	q_1	q_2	q_3	q_4	q_5	q_6
q_0	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$
q_1	-	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$
q_2	-	-	$\not\equiv$	$\equiv$	$\not\equiv$	$\not\equiv$
q_3	-	-	-	$\not\equiv$	$\not\equiv$	$\equiv$
q_4	-	-	-	-	$\not\equiv$	$\not\equiv$
q_5	-	-	-	-	-	$\not\equiv$

leia-se:

	q_1	q_2	q_3	q_4	q_6
q_0	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$
q_1	-	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$
q_2	-	-	$\not\equiv$	$\equiv$	$\not\equiv$
q_3	-	-	-	$\not\equiv$	$\equiv$
q_4	-	-	-	-	$\not\equiv$

onde se lê:

	δ'	a	b
$\rightarrow$	$[q_0]$	$[q_1]$	$[q_3, q_6]$
	$[q_1]$	$[q_2, q_4]$	$[q_3, q_6]$
(f)	$[q_2, q_4]$	$[q_2, q_4]$	$[q_3, q_6]$
	$[q_3, q_6]$	$[q_2, q_4]$	$[q_2, q_4]$
(f)	$[q_5]$	$[q_2, q_4]$	$[q_5]$

leia-se:

	δ'	a	b
$\rightarrow$	$[q_0]$	$[q_1]$	$[q_3, q_6]$
	$[q_1]$	$[q_2, q_4]$	$[q_3, q_6]$
(f)	$[q_2, q_4]$	$[q_2, q_4]$	$[q_3, q_6]$
	$[q_3, q_6]$	$[q_2, q_4]$	$[q_2, q_4]$

52. Página 228, Tabela 3.46
substituir "(f)" por "$\leftarrow$" (duas ocorrências)

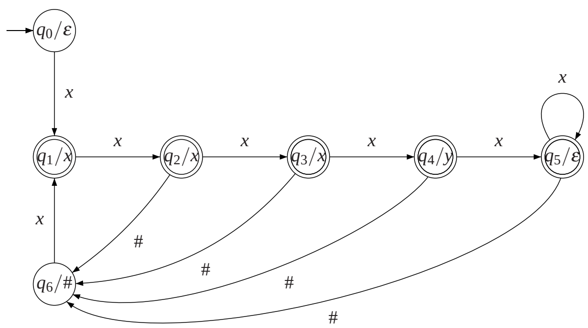
53. Página 230, imediatamente antes do Exemplo 3.46
acrescentar:
"Alguns autores consideram $\lambda : Q \rightarrow \Delta^*$ ao invés de $\lambda : Q \rightarrow \Delta$. Trata-se de uma flexibilização na definição da

função de transdução, de forma a permitir a geração de uma cadeia ao invés de um único símbolo em cada aplicação da mesma. De qualquer forma, é fácil perceber que transdutores que adotam a primeira definição podem ser mapeados em transdutores equivalentes baseados na segunda definição (que é a definição original). O Exemplo 3.46, apresentado a seguir, adota esta definição alternativa.”.

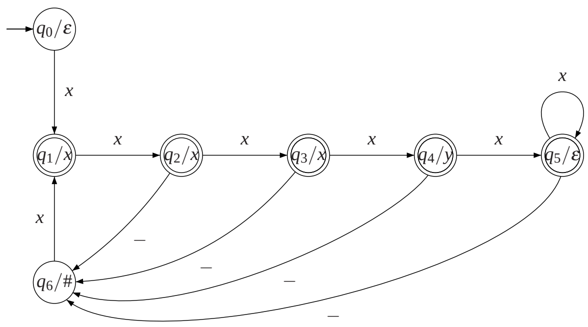
54. Página 231, imediatamente antes do Exemplo 3.47
acrescentar:
“Assim como no caso das Máquinas de Mealy, e pelos mesmos motivos, alguns autores consideram $\lambda : Q \times \Sigma \rightarrow \Delta^*$ ao invés de $\lambda : Q \times \Sigma \rightarrow \Delta$. Do mesmo modo, é fácil perceber que transdutores que adotam a primeira definição podem ser mapeados em transdutores equivalentes baseados na segunda definição (que é a definição original). O Exemplo 3.47, apresentado a seguir, adota esta definição alternativa.”.

55. Página 233, Seção 3.9, segundo parágrafo, segunda frase
onde se lê: “... verdadeira para toda e qualquer linguagen regular.”
leia-se: “... verdadeira para toda e qualquer linguagen regular infinita.”

56. Página 234, Figura 3.67
onde se lê:



leia-se:



57. Página 234, Item 2
onde se lê: “$(q_0, a_1...a_m) \vdash (q_n, a_{n+1}...a_m)$”
leia-se: “$(q_0, a_1...a_m) \vdash^* (q_n, a_{n+1}...a_m)$”

58. Página 235, Item 1 (segunda ocorrência)
onde se lê: “Se $(q_0, ab) \vdash (q_0, b) \vdash (q_0, \varepsilon)$, então pelo menos uma das seguintes possibilidades é verdadeira:”
leia-se: “Se $(q_0, ab) \vdash (q_0, b) \vdash (q_0, \varepsilon)$, então as três possibilidades seguintes são verdadeiras:”

59. Página 236, Exemplo 3.51
onde se lê: “$x = ab, y = b, z = c$. As cadeias $(ab)(b)^*(c)$ estão contidas em L.”
leia-se: “$x = ab, y = b, z = bc$. As cadeias $(ab)(b)^*(bc)$ estão contidas em L.”

60. Página 237, segundo bullet
onde se lê: “senteça”
leia-se: “sentença”

61. Página 245, Exemplo 3.64, última linha
onde se lê: “$L_1 = ...$”
leia-se: “$L_2 = ...$”

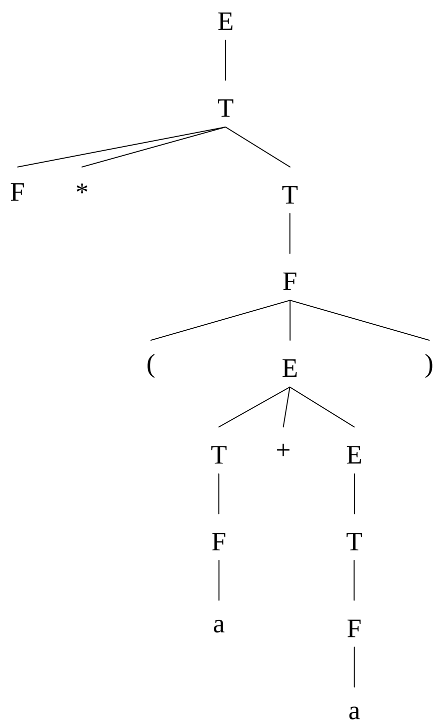
62. Página 248, terceiro parágrafo
onde se lê: “A condição suficiente do Teorema 3.11...”
leia-se: “A condição suficiente do Teorema 3.24...”

63. Página 249, penúltimo parágrafo
onde se lê: “A condição “somente se” do Teorema 3.11 ... (ver Teorema 3.11).”
leia-se: “A condição “somente se” do Teorema 3.25 ... (ver Teorema 3.24).”

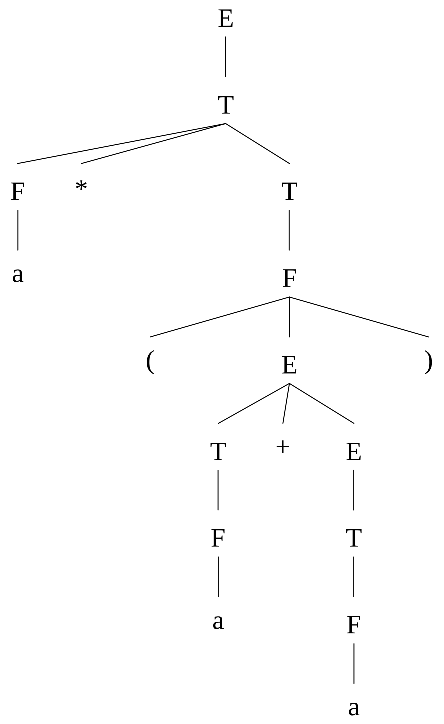
64. Página 250, Exemplo 3.68
onde se lê:
“Então, para saber se L é não-vazia...”
leia-se:
“Então, para saber se L é infinita...”

65. Página 250, Teorema 3.26, segundo parágrafo
onde se lê: “Decorre diretamente ... (Teorema 3.11).”
leia-se: “Decorre diretamente ... (Teorema 3.24).”

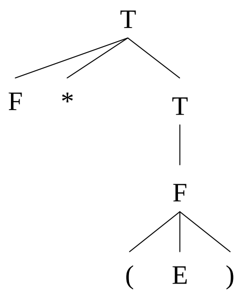
66. Página 309, Figura 4.1
onde se lê:



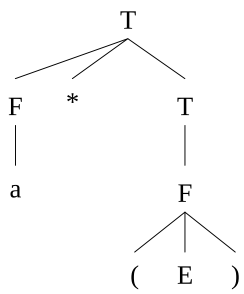
leia-se:



67. Página 310, Figura 4.2
onde se lê:



leia-se:



68. Página 310, legenda da Figura 4.2

onde se lê: “Árvore de derivação para $T \Rightarrow^* a * (a + a)$”
leia-se: “Árvore de derivação para $T \Rightarrow^* a * (E)$”

69. Página 311, linhas 2 e 3
onde se lê: “esquerda” (linha 2)
leia-se: “direita”
onde se lê: “direita” (linha 3)
leia-se: “esquerda”

70. Página 311, gramática do Exemplo 4.13
onde se lê: “`if` <*exp*> `else` <*com*> `else` <*com*>”
leia-se: “`if` <*exp*> `then` <*com*> `else` <*com*>”

71. Página 312, penúltimo parágrafo
onde se lê: “(e, portant,o também mais à direita)”
leia-se: “(e, portanto, também mais à direita)”

72. Página 317, Algoritmo 4.1, item 3
onde se lê: “$\alpha \in (N_{i-1} \cup \Sigma^*)$”
leia-se: “$\alpha \in (N_{i-1} \cup \Sigma)^*$”

73. Página 318, Algoritmo 4.2, item 3
onde se lê: “... e $A \in N_{i-1}$”
leia-se: “... e $A \in V_{i-1}$”

74. Página 318, último parágrafo
onde se lê:
“Símbolos inacessíveis e inúteis podem ser eliminados de uma gramática livre de contexto qualquer aplicando-se-lhes inicialmente o Algoritmo 4.5 e, ao resultado deste, o Algoritmo 4.5, ou vice-versa. A ordem de aplicação dos algoritmos não interfere no resultado final obtido, como demonstra o Teorema 4.5.”
leia-se:
“Símbolos inacessíveis e inúteis podem ser eliminados de uma gramática livre de contexto qualquer aplicando-se-lhes inicialmente o Algoritmo 4.1 e, ao resultado deste, o Algoritmo 4.2, uma única vez cada. A ordem de aplicação dos algoritmos interfere no resultado final obtido, como demonstra o Teorema 4.6.”

75. Página 319, Teorema 4.6
eliminar o texto completo desde “Teorema 4.6...” até “... em qualquer ordem”

76. Página 320, Exemplo 4.18, primeira linha
onde se lê: “$P' = \{S \rightarrow A | A \rightarrow bS \ldots$”
leia-se: “$P' = \{S \rightarrow A, A \rightarrow bS \ldots$”

77. Página 320, Exemplo 4.18, acrescentar, embaixo de $V_2 = \{S, A, b\}$:
$V_3 = \{S, A, b\}$

78. Página 320, Exemplo 4.18
onde se lê: “$V'' = \{S, A\}$”
leia-se: “$V'' = \{S, A, b\}$”

79. Página 320, Exemplo 4.18
onde se lê: “$P'' = \{S \rightarrow A | A \rightarrow bS \ldots$”
leia-se: “$P'' = \{S \rightarrow A, A \rightarrow bS \ldots$”

80. Página 320, Exemplo 4.18
onde se lê: “Observe-se a eliminação dos símbolos C e a de V''”
leia-se: “Observe-se a eliminação dos símbolos C e a de V''”

81. Página 320, depois do Exemplo 4.18 e antes de “Analisa-se...”
inserir:
Teorema 4.6 (Ordem de eliminação de símbolos inúteis e inacessíveis) “A obtenção de uma gramática livre de contexto equivalente G_A, isenta de símbolos inúteis e inacessíveis, a partir de uma gramática livre de contexto G qualquer, pode ser feita pela aplicação dos Algoritmos 4.1 e 4.2, uma única vez cada, desde que a aplicação ocorra nesta ordem (primeiro símbolos inúteis e depois símbolos inacessíveis).”

De fato, temos que (i) a eliminação de símbolos inúteis pode gerar símbolos inacessíveis, mas o contrário não é verdadeiro, ou seja, (ii) a eliminação de símbolos inacessíveis após a eliminação de símbolos inúteis não torna nenhum símbolo inútil. Logo, a eliminação da símbolos inúteis seguida da eliminação de símbolos inacessíveis garante que a gramática resultante será isenta de ambos os tipos de símbolos com uma única aplicação de cada algoritmo. A aplicação na ordem inversa (ou seja, primeiro a eliminação de símbolos inacessíveis e depois a eliminação de símbolos inúteis) não garante que a gramática resultante seja isenta de ambos os tipos de símbolos. Neste caso, seria necessária uma segunda execução do algoritmo de eliminação de símbolos inacessíveis para garantir o resultado pretendido.

As provas de (i) e (ii) podem ser feitas levando-se em conta que a acessibilidade dos símbolos se propaga, nas regras da gramática, da esquerda para direita, ao passo que a utilidade se propaga no sentido inverso, da direita para a esquerda.

Para provar (i), basta considerar qualquer gramática G com um símbolo útil e acessível Y tal que todas as regras de G em que Y comparece do lado direito contém um símbolo inútil e acessível do lado esquerdo (diferente de Y). Desta forma, a eliminação dos símbolos inúteis da esquerda implica a eliminação das regras onde os mesmos comparecem, o que por sua vez implica a inacessibilidade do símbolo Y. Ou seja, o símbolo acessível Y torna-se inacessível após a eliminação dos símbolos inúteis de G. Naturalmente, Y não deve ser acessível por nenhum outro caminho formado apenas por símbolos úteis. Em outras palavras, trata-se da situação muito particular em que a acessibilidade de um símbolo Y está garantida apenas por meio de caminhos que contêm símbolos inúteis. Logo, a eliminação destes caminhos implica a inacessibilidade de Y.

Exemplo: considere a gramática com as seguintes regras:

$S \to W \mid X$
$W \to aW \mid a$
$X \to YZ$
$Y \to bY \mid b$
$Z \to cZ$

Consideremos o que acontece quando a aplicação do Algoritmo 4.1 é feita em primeiro lugar. O resultado é:

$S \to W$
$W \to aW \mid a$
$Y \to bY \mid b$

Em seguida, a aplicação do Algoritmo 4.2 se encarrega da eliminação do símbolo inacessível Y, resultando em:

$S \to W$
$W \to aW \mid a$

Desta forma, com uma única execução de cada um dos algoritmos é possível obter uma gramática equivalente, isenta de símbolos inúteis e inacessíves. Suponhamos agora que o Algoritmo 4.2 seja aplicado em primeiro lugar. O resultado, neste caso, é uma gramática idêntica à original (pois todos os símbolos são acessíveis). Num segundo passo, a eliminação de símbolos inúteis (Z e X) por meio do Algoritmo 4.1 torna o símbolo Y inacessível:

$S \to W \mid X$
$W \to aW \mid a$
$Y \to bY \mid b$

Para resolver esta situação e eliminar Y, uma segunda execução do Algoritmo 4.1 é necessária. Logo, precisamos de duas execuções do Algoritmo 4.1 e uma do Algoritmo 4.2 para chegar no mesmo resultado.

Esta situação acontece porque o símbolo Y é útil e acessível, porém comparece numa regra ($X \to YZ$) cujo lado esquerdo (X) é um símbolo inútil (por causa do símbolo Z, que também é inútil) e acessível. Ou seja, a acessibilidade de Y é garantida apenas por um caminho que passa por um símbolo inútil. Logo, a eliminação do símbolo inútil Z

implica a eliminação da regra $X \to YZ$, o que por sua vez implica a inacessibilidade de Y (apesar de ele ser útil). As regras $Y \to bY|b$ não devem ser consideradas pois possuem o próprio símbolo Y do lado esquerdo.

Para provar (ii), considere que $G = (V, \Sigma, P, S)$ é a gramática original, $G_U = (V_U, \Sigma_U, P_U, S)$ é a gramática obtida a partir de G pela aplicação do Algoritmo 4.1 e $G_A = (V_A, \Sigma_A, P_A, S)$ é a gramática obtida pela aplicação do Algoritmo 4.2 à G_U. Queremos provar que G_U é isenta de símbolos inúteis e também de símbolos inacessíveis. Para tanto, suponha que X seja um símbolo qualquer de G_A, portanto $X \in V_A$. Como os Algoritmos 4.1 e 4.2 apenas eliminam símbolos e regras das respectivas gramáticas de entrada, segue que $X \in V_U$. Portanto, X é um símbolo útil em G_U e um símbolo acessível em G_A. A partir dessas informações, iremos provar que X é também um símbolo útil em G_A. Como resultado, poderemos concluir que todos os símbolos de G_A são acessíveis e úteis, e o Algoritmo 4.2 não gera símbolos inúteis.

Como X é útil em G_U, segue que existe uma cadeia de símbolos terminais w tal que $X \Rightarrow^*_{G_U} w$. Por outro lado, como X é acessível em G_A, segue que existem cadeias α e β tais que $S \Rightarrow^*_{G_A} \alpha X \beta$. Esta última observação, no entanto, permite concluir que todos os símbolos de $\alpha X \beta$ são acessíveis em G_A e, portanto, todos eles também o eram em G_U. Portanto, temos que $S \Rightarrow^*_{G_U} \alpha X \beta$. Como os símbolos de $\alpha X \beta$ aparecem em G_U, segue que todos eles são úteis. Portanto, existem cadeias de terminais w_1 e w_2 tais que $\alpha X \beta \Rightarrow^*_{G_U} w_1 w w_2$. Como todos os símbolos de $\alpha X \beta$ são acessíveis em G_U, segue que todos os demais símbolos utilizados nesta últma derivação são também acessíveis. Logo, todos eles, com as suas respectivas regras, devem pertencer também à G_A. Assim, é possível afirmar que $\alpha X \beta \Rightarrow^*_{G_A} w_1 w w_2$ e temos que X é um símbolo útil em G_A como queríamos provar.

Para concluir, cumpre provar que a gramática G', obtida por meio da aplicação sucessiva dos Algoritmos 4.1 e 4.2, nesta ordem, a uma gramática original G, é tal que $L(G) = L(G')$. De fato, toda sentença gerada por G é obtida por meio de derivações em que são utilizados apenas símbolos úteis e acessíveis, ou seja, por meio de regras que também pertencem à G'. Assim, toda e qualquer derivação de uma sentença em G é efetuada por meio de regras que também pertencem à G', o que prova que $L(G) \subseteq L(G')$. Por outro lado, os dois algoritmos citados não introduzem novas regras nas gramáticas resultantes, eles apenas eliminam regras que contêm símbolos inúteis e/ou inacessíveis. Ou seja, $L(G') \subseteq L(G)$. Logo, toda sentença gerada por G' é também uma sentença que pode ser gerada por G e temos que que $L(G) = L(G')$.

82. Página 321, item 2
onde se lê: "As produções $A \to \alpha \in P, \sigma = \sigma_1 \sigma_2 ...$"
leia-se: "As produções $A \to \alpha \in P, \alpha = \sigma_1 \sigma_2 ...$"

83. Página 322, Exemplo 4.20
onde se lê: "$S \to aBC|aB|aC$"
leia-se: "$S \to aBC|aB|aC|a$"

84. Página 324, antes de "Diz-se que..."
inserir:

Teorema 4.8a (Ordem de eliminação de regras vazias e unitárias) "A obtenção de uma gramática livre de contexto equivalente G_3, isenta de símbolos inúteis e inacessíveis, a partir de uma gramática livre de contexto G_1 qualquer, pode ser feita pela aplicação dos Algoritmos 4.3 e 4.4, uma única vez cada, desde que a aplicação ocorra nesta ordem (primeiro regras vazias e depois regras unitárias)."

De fato, uma simples inspeção dos algoritmos 4.3 e 4.4 revela que (i) a eliminação de regras vazias pode gerar regras unitárias, mas o contrário não é verdadeiro, ou seja, (ii) a eliminação de regras unitárias não gera regras vazias. Logo, a eliminação de regras vazias seguida da eliminação de regras unitárias garante que a gramática resultante será isenta de ambos os tipos de regras com uma única aplicação de cada algoritmo. A aplicação na ordem inversa (ou seja, primeiro a eliminação de regras unitárias e depois a eliminação de regras vazias) não garante que a gramática resultante seja isenta de ambos os tipos de símbolos. Neste caso, seria necessário uma segunda execução do algoritmo de eliminação de regras unitárias para garantir o resultado pretendido.

Os resultados vistos até o momento indicam que a eliminação de símbolos inúteis deve preceder a eliminação de símbolos inacessíveis, e também que a eliminação de regras vazias deve preceder a eliminação de regras unitárias. Resta analisar se os símbolos devem ser eliminados antes das regras, ou se as regras devem ser eliminadas antes do símbolos.

Por um lado, é importante notar que a eliminação de símbolos inúteis e inacessíveis não introduz novas regras na gramática (ao contrário, pode haver no máximo uma redução do conjunto de regras original). Portanto, não há risco de serem introduzidas novas regras vazias ou unitárias na gramática como resultado da aplicação dos algoritmos de eliminação de símbolos inúteis e inacessíveis.

Por outro lado, a eliminação de regras vazias e unitárias, em particular a eliminação de regras unitárias, pode introduzir novos símbolos inúteis na gramática original, conforme mostrado a seguir. Tal fato sugere, portanto, que a eliminação de regras deva preceder a eliminação de símbolos (conforme enunciado do Teorema 4.8b). Antes, no entanto, ilustramos o fato de que a aplicação do algoritmo de eliminação de regras unitárias pode introduzir símbolos inúteis não existentes na gramática original.

Exemplo: considere a gramática abaixo (isenta de símbolos inúteis) com as seguintes regras:

$S \to X$
$X \to a \,|\, b$

A aplicação do algoritmos de eliminação de regras unitárias gera como resultado:

$S \to a \,|\, b$
$X \to a \,|\, b$

Como é fácil perceber, a gramática resultante possui o símbolo inútil X. Logo, fica claro que, no caso geral, a aplicação do algoritmo de eliminação de regras unitárias pode introduzir novos símbolos inúteis que inexistiam na gramática original.

Teorema 4.8b (Ordem de eliminação de símbolos e regras) "A obtenção de uma gramática livre de contexto equivalente G_3, isenta de regras vazias e regras unitárias, e também de símbolos inúteis e inacessíveis, a partir de uma gramática livre de contexto G_1 qualquer, pode ser feita pela aplicação dos Algoritmos 4.3 (regras vazias), 4.4 (regras unitárias), 4.1 (símbolos inúteis) e 4.2 (símbolos inacessíveis), uma única vez cada, nesta ordem."

Uma demonstração formal deste teorema exigiria a prova prévia dos seguintes teoremas auxiliares:

(a) De que o Algoritmo 4.3 produz uma gramática equivalente isenta de regras vazias;

(b) De que o Algoritmo 4.4 produz uma gramática equivalente isenta de regras unitárias e não introduz regras vazias.

(c) De que o Algoritmo 4.1 produz uma gramática equivalente isenta de símbolos inúteis e não introduz regras vazias nem regras unitárias;

(d) De que o Algoritmo 4.2 produz uma gramática equivalente isenta de símbolos inacessíveis e não introduz símbolos inúteis nem regras vazias ou unitárias.

Tais demonstrações, no entanto, estão fora do escopo deste texto mas podem ser justificadas com base nos argumentos informais apresentados anteriormente. Sendo assim, a ordem sugerida para a realização de todas as etapas de simplificação de gramáticas livres de contexto é:

(a) Eliminação de regras vazias;

(b) Eliminação de símbolos unitárias;

(c) Eliminação de símbolos inúteis;

(d) Eliminação de símbolos inacessíveis.

85. Página 326, Algoritmo 4.6, item 7
onde se lê: "... com $n \geq 2$ e ..."
leia-se: "... com $n > 2$ e ..."

86. Página 326, Exemplo 4.23
onde se lê: "Da aplicação do algoritmo acima resulta $G' = (V', \Sigma, P', S)$ com:" e até o final do exemplo, no início da página 327

leia-se: “Eliminando-se as produções unitárias:

$$
\begin{aligned}
\{E &\rightarrow E+T|T*F|(E)|a \\
T &\rightarrow T*F|(E)|a \\
F &\rightarrow (E)|a
\end{aligned}
$$

Da aplicação do Algoritmo 4.6 resulta $G' = (V', \Sigma, P', S)$, com:

$$
\begin{aligned}
N' &= \{E, T, F, X_0, X_1, X_2, X_3, W_0, W_1, W_2\} \\
P' &= \{E \rightarrow EW_0|TW_1|X_2W_2|a, \\
&\quad W_0 \rightarrow X_0T, \\
&\quad W_1 \rightarrow X_1F, \\
&\quad W_2 \rightarrow EX_3, \\
&\quad T \rightarrow TW_1|X_2W_2|a, \\
&\quad F \rightarrow X_2W_2|a, \\
&\quad X_0 \rightarrow +, \\
&\quad X_1 \rightarrow *, \\
&\quad X_2 \rightarrow (, \\
&\quad X_3 \rightarrow)\}\text{”}
\end{aligned}
$$

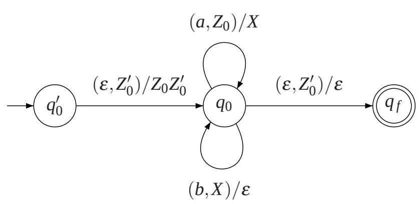
87. Página 327, terceiro parágrafo
onde se lê: “$X_i \rightarrow X_{j\alpha}$”
leia-se: “$X_i \rightarrow X_j\alpha$”

88. Página 342, Teorema 4.11
onde se lê: “O Algorimo 4.11 ...”
leia-se: “O Algoritmo 4.9 ...”

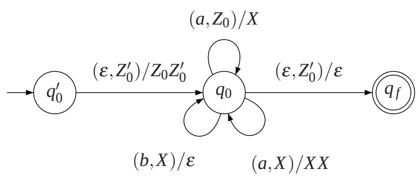
89. Página 344, Algoritmo 4.10
onde se lê: “Entrada: ... $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$...”
leia-se: “Entrada: ... $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset)$...”
onde se lê: “Saída: ... $M = (Q', \Sigma, \Gamma', \delta', q_0', Z_0', \emptyset)$...”
leia-se: “Saída: ... $M = (Q', \Sigma, \Gamma', \delta', q_0', Z_0', F)$...”

90. Página 345, entre os itens 2 e 3 do Algoritmo 4.10
acrescentar: “$F \leftarrow \{q_f\}$;”

91. Página 345, Figura 4.12
onde se lê:



leia-se:



92. Página 351, Algoritmo 4.12
onde se lê: "$M = (Q, \Sigma, \Gamma, q, S, F)$"
leia-se: "$M = (Q, \Sigma, \Gamma, \delta, q, S, F)$"

93. Página 353, Algoritmo 4.13
onde se lê:
"para qualquer seqüência de $q_j \in Q$, $2 \leq j \leq (k+1)$"
leia-se:
"para toda e qualquer seqüência de estados $q_2, q_3, ..., q_k, q_{k+1}$ que possa ser obtida a partir de Q (repetições são permitidas);"

94. Página 353
onde se lê:
"$(q_i, \alpha\beta, Z_0) \vdash^* (q_k, \beta, \mu)$"
leia-se:
"$(q_i, \alpha\beta, Z\mu) \vdash^* (q_k, \beta, \mu)$"

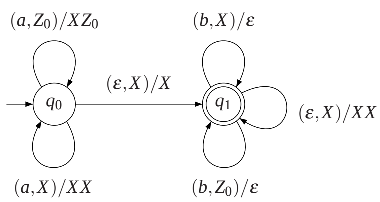
95. Página 353, antes de "O conjunto das cadeias..."
inserir:
Em outras palavras, é possível demonstrar que:

$$[q_i Z q_k] \Rightarrow^* w \text{ se e somente se } (q_i, w, Z) \vdash^* (q_k, \varepsilon, \varepsilon)$$

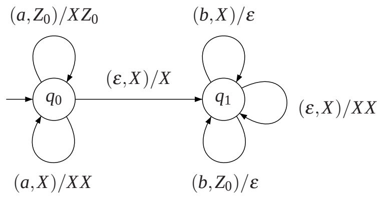
96. Página 354, última linha
onde se lê:
"Algoritmo 4.14"
leia-se:
"Algoritmo 4.13"

97. Página 354, depois da última linha
acrescentar:
O Exemplo 4.38 ilustra a aplicação do Algoritmo 4.13 de duas formas diferentes, porém com o mesmo resultado final.

98. Página 355, Exemplo 4.38
onde se lê:



leia-se:



99. Página 355, Exemplo 4.38, depois da equação 4.8 e antes de "Considerando-se..."
inserir:

Conforme o Algoritmo 4.13, passamos a considerar as transições do autômato individualmente, com o objetivo de determinar as regras gramaticais delas derivadas:

- Para $\delta(q_0,a,Z_0) = (q_0,XZ_0)$, considerar $[q_0Z_0\_\_] \to a[q_0X\_\_][\_\_Z_0\_\_]$ com as listas (q_0,q_0), (q_0,q_1), (q_1,q_0) e (q_1,q_1), gerando:

$$\begin{array}{rcl} [q_0Z_0q_0] & \to & a[q_0Xq_0][q_0Z_0q_0] \\ [q_0Z_0q_1] & \to & a[q_0Xq_0][q_0Z_0q_1] \\ [q_0Z_0q_0] & \to & a[q_0Xq_1][q_1Z_0q_0] \\ [q_0Z_0q_1] & \to & a[q_0Xq_1][q_1Z_0q_1] \end{array}$$

- Para $\delta(q_0,a,X) = (q_0,XX)$, considerar $[q_0X\_\_] \to a[q_0X\_\_][\_\_X\_\_]$ com as listas (q_0,q_0), (q_0,q_1), (q_1,q_0) e (q_1,q_1), gerando:

$$\begin{array}{rcl} [q_0Xq_0] & \to & a[q_0Xq_0][q_0Xq_0] \\ [q_0Xq_1] & \to & a[q_0Xq_0][q_0Xq_1] \\ [q_0Xq_0] & \to & a[q_0Xq_1][q_1Xq_0] \\ [q_0Xq_1] & \to & a[q_0Xq_1][q_1Xq_1] \end{array}$$

- Para $\delta(q_0,\varepsilon,X) = (q_1,X)$, considerar $[q_0X\_\_] \to [q_1X\_\_]$ com as listas (q_0) e (q_1), gerando:

$$\begin{array}{rcl} [q_0Xq_0] & \to & [q_1Xq_0] \\ [q_0Xq_1] & \to & [q_1Xq_1] \end{array}$$

- Para $\delta(q_1,b,X) = (q_1,\varepsilon)$:

$$[q_1Xq_1] \to b$$

- Para $\delta(q_1,\varepsilon,X) = (q_1,XX)$, considerar $[q_1X\_\_] \to [q_1X\_\_][\_\_X\_\_]$ com as listas (q_0,q_0), (q_0,q_1), (q_1,q_0) e (q_1,q_1), gerando:

$$\begin{array}{rcl} [q_1Xq_0] & \to & [q_1Xq_0][q_0Xq_0] \\ [q_1Xq_1] & \to & [q_1Xq_0][q_0Xq_1] \\ [q_1Xq_0] & \to & [q_1Xq_1][q_1Xq_0] \\ [q_1Xq_1] & \to & [q_1Xq_1][q_1Xq_1] \end{array}$$

- Para $\delta(q_1,b,Z_0) = (q_1,\varepsilon)$:

$$[q_1Z_0q_1] \to b$$

A renomeação dos símbolos não-terminais e o agrupamento das regras produz como resultado o conjunto:

$$\begin{array}{rcl} S & \rightarrow & A|B \\ A & \rightarrow & aCA|aDE \\ B & \rightarrow & aCB|aDF \\ C & \rightarrow & aCC|aDG|G \\ D & \rightarrow & aCD|aDH|H \\ F & \rightarrow & b \\ G & \rightarrow & GC|HG \\ H & \rightarrow & GD|b|HH \end{array}$$

A eliminação de símbolos inacessíveis e inúteis resulta em:

$$\begin{array}{rcl} S & \rightarrow & B \\ B & \rightarrow & aDF \\ D & \rightarrow & aDH|H \\ F & \rightarrow & b \\ H & \rightarrow & b|HH \end{array}$$

Finalmente, a gramática pode ser ainda simplificada para:

$$\begin{array}{rcl} S & \rightarrow & aDb \\ D & \rightarrow & aDb^+|b^+ \end{array}$$

ou seja, $L(G) = V(M) = \{a^i b^j | i \geq 1 \text{ e } j > i\}$.

Apresenta-se agora uma particular seqüência de movimentos efetuada por M durante o reconhecimento da sentença $aabbbb$:

$(q_0, aabbbb, Z_0) \vdash (q_0, abbbb, XZ_0) \vdash (q_0, bbbb, XXZ_0) \vdash (q_1, bbbb, XXZ_0) \vdash (q_1, bbb, XZ_0) \vdash (q_1, bbb, XXZ_0) \vdash (q_1, bb, XZ_0) \vdash (q_1, b, Z_0) \vdash (q_1, \varepsilon, \varepsilon)$

A correspondência entre G e M pode ser ilustrada através da análise da seqüência de derivações mais à esquerda obtida para esta mesma cadeia $aabbbb$, e de sua comparação com a seqüência de movimentos efetuados pelo autômato conforme apresentado acima:

$$\begin{array}{rcl} S & \Longrightarrow & [q_0Z_0q_1] \Longrightarrow a[q_0Xq_1][q_1Z_0q_1] \Longrightarrow aa[q_0Xq_1][q_1Xq_1][q_1Z_0q_1] \Longrightarrow \\ & & aa[q_1Xq_1][q_1Xq_1][q_1Z_0q_1] \Longrightarrow aab[q_1Xq_1][q_1Z_0q_1] \Longrightarrow \\ & & aab[q_1Xq_1][q_1Xq_1][q_1Z_0q_1] \Longrightarrow \\ & & aabb[q_1Xq_1][q_1Z_0q_1] \Longrightarrow \\ & & aabbb[q_1Z_0q_1] \Longrightarrow \\ & & aabbbb \end{array}$$

Observe-se que as formas sentenciais obtidas através de G, neste exemplo, estão diretamente relacionadas às configurações assumidas por G em cada etapa do reconhecimento da cadeia considerada. Note-se, portanto, que G "simula", através de uma seqüência de derivações mais à esquerda, a seqüência de movimentos que conduz M de sua configuração inicial até uma configuração final.

O resultado apresentado segue estritamente o modelo sugerido pelo Algoritmo 4.13, que constrói a gramática a partir da análise das transições do autômato, uma por uma. Uma outra forma de se chegar no mesmo resultado (apesar de a gramática obtida ser distinta) é apresentada a seguir.

100. Página 356, final da página:
inserir:

A gramática pode ser ainda simplificada para:

$$\begin{array}{rcl} S & \rightarrow & aBb \\ B & \rightarrow & aBb^+|b^+ \end{array}$$

101. Página 357 eliminar todo o texto entre "Apresenta-se..." (inclusive) e "configuração final." (idem)

102. Página 359, Teorema 4.17
onde se lê: "Seja L uma linguagem livre de contexto, com $\varepsilon \notin L$."
leia-se: "Seja L uma linguagem livre de contexto qualquer."
onde se lê: "$\gamma = uvwxy$;;"
leia-se: "$\gamma = uvwxy$;"
onde se lê: "$|vwx| \leq n$"
leia-se: "$|vwx| \leq (n-1) * 2$"

103. Página 359
onde se lê: "Se L é livre de contexto e não contém a cadeia vazia, então $L = L(G)$..." leia-se: "Se L é livre de contexto, então $L - \{\varepsilon\} = L(G)$..."

104. Página 359, Nota de rodapé 3
eliminar

105. Página 361
onde se lê:

- Como $vwx = \alpha_1$, então $2 \leq |vwx| \leq 2^k$;

leia-se:

- Como $vwx = \alpha_1$, então $2 \leq |vwx| \leq 2^k$. Por outro lado, como $n = 2^{k-1} + 1$, segue que $2^k = (n-1) * 2$. Logo, $2 \leq |vwx| \leq (n-1) * 2$;

106. Página 362, Exemplo 4.39
onde se lê: "$\{a^iba^i | i \geq 1\}$"
leia-se: "$\{a^iba^i | i \geq 0\}$"

107. Página 364, antes do Exemplo 4.40
inserir o parágrafo:
"Alguns autores adotam $n' = 2^k$ em substituição ao valor $n = 2^{k-1} + 1$ adotado neste texto. Suponha $\gamma \in L$, $|\gamma| \geq n'$. Então, como $n' \geq n$ é possível afirmar que $|\gamma| \geq n$. Pelo enunciado do Pumping Lemma, por outro lado, é possível afirmar que $\gamma = uvwxy$ de tal forma que $|vxy| \leq (n-1) * 2$. Mas como $n = 2^{k-1} + 1$, segue que $(n-1) * 2 = 2^k = n'$. Logo, temos que $|vxy| \leq n'$. A vantagem de se utilizar $n' = 2^k$ no lugar de $n = 2^{k-1} + 1$ é que o enunciado do Pumping Lemma torna-se uniforme, conforme mostrado a seguir:

Teorema 4.17a "Pumping Lemma" modificado Seja L uma linguagem livre de contexto qualquer. Então, existe uma constante inteira n, dependente apenas de L, que satisfaz às seguintes condições: (i) $\forall \gamma \in L, |\gamma| \geq n, \gamma = uvwxy$; (ii) $|vwx| \leq n$; (iii) $|vx| \geq 1$; (iv) $\forall i \geq 0, uv^iwx^iy \in L$.

Essa versão corresponde ao enunciado mais comum do presente teorema, sendo usada na maioria das aplicações, como é o caso dos exemplos seguintes, por causa da sua praticidade ao igualar o comprimento mínimo da sentença γ com o comprimento máximo da cadeia $|vwx|$. Ela, no entanto, desconsidera a possibilidade de utilização de sentenças de comprimento menor que 2^k (especificamente, de comprimento maior ou igual a $2^{k-1} + 1$ e menor que 2^k), o que é feito no enunciado original do Teorema 4.17 e na prova apresentada nesta seção.

Para concluir, cumpre observar que o Pumping Lemma para as Linguagens Livres de Contexto se aplica tanto para linguagens finitas quanto infinitas. Para atestar a validade da proposição para linguagens finitas basta considerar como valor da constante n o sucessor do comprimento da maior sentença de L. Outras opções são utilizar $n = 2^{k-1}+1$ ou mesmo $n = 2^k$, onde k é o número de símbolos não-terminais da gramática na Forma Normal de Chomsky que gera L. Como não existem sentenças em L com comprimento maior ou igual a n, qualquer que seja o caso (conforme explicado a seguir), o teorema é verificado automaticamente. Senão, vejamos:

- Caso 1: n é o sucessor da sentença de maior comprimento de L: trivial, pois a linguagem é finita;
- Caso 2: $n = 2^{k-1}+1$: basta observar que os caminhos numa árvore de derivação obtida a partir da gramática na Forma Normal de Chomsky podem conter no máximo k símbolos não-terminais (pois não deve haver repetição de símbolos não-terminais ao longo do mesmo, uma vez que a linguagem é finita); logo, o comprimento destes caminhos é no máximo $k+1$ e a altura máxima das respectivas árvores é k. Por conseqüência, o tamanho das maiores sentenças representadas por tais árvores é limitado em 2^{k-1}. Logo, não existem sentenças com comprimento maior ou igual a $2^{k-1}+1$;
- Caso 3: $n = 2^k$: trivial, levando-se em conta o resultado do item anterior.

Na seqüência são apresentados três exemplos de aplicação do Pumping Lemma para as Linguagens Livres de Contexto. Em todos eles, a aplicação é feita com o objetivo de provar que as respectivas linguagens não são livres de contexto.

108. Página 365, Exemplo 4.40, Item 4
onde se lê: "... e quantidades de símbolos "*a*" e "*b*" respectivamente menores..."
leia-se: "... e quantidades de símbolos "*a*" e "*b*" (pelo menos uma delas) respectivamente menores..."

109. Página 365, Exemplo 4.40, Item 5
onde se lê: "... e quantidades de símbolos "*b*" e "*c*" respectivamente menores..."
leia-se: "... e quantidades de símbolos "*b*" e "*c*" (pelo menos uma delas) respectivamente menores..."

110. Página 366, Exemplo 4.42
substituir o texto do exemplo por:
"A linguagem $L_3 = \{a^k | k \geq 1$ é um número primo$\}$ não é livre de contexto. Suponha-se que L_3 seja livre de contexto e considere-se a sentença $\gamma = a^p$, $p \geq n+2$, onde p é um número primo e n é o valor da constante definida pelo "Pumping Lemma" para linguagens livres de contexto. Se $\gamma = uvwxy$ pertence a L_3, então, de acordo com o "Lemma", a sentença uwy também deve pertencer. Seja $q = |uwy|$.

O comprimento das sentenças uv^iwx^iy pode ser calculado da seguinte forma: $|uv^iwx^iy| = |uwy| + i * |vx|$. Em particular, o comprimento da sentença $|uv^qwx^qy| = |uwy| + q * |vx| = q + q * |vx| = q * (1 + |vx|)$. Além disso,

- $q = |uwy| = |uvwxy| - |vx|$. Como, pelo "Pumping Lemma", $|vwx| \leq n$, segue que $|vx| \leq n$. Como $|uvwxy| \geq n+2$, segue que $q \geq 2$;
- Pelo "Pumping Lemma", $|vx| \geq 1$. Logo, $1 + |vx| \geq 2$.

Portanto, o comprimento de $|uv^qwx^qy|$ corresponde ao produto de dois números maiores que 1 (ou seja, ele não é primo) e isso prova que L_3 não pode ser livre de contexto.

Vale lembrar que, anteriormente ("Pumping Lemma" para as Linguagens Regulares), esta mesma linguagem foi demonstrada como sendo não-regular."

111. Página 368, último parágrafo
onde se lê: "inspecionando-se no máximo os k primeiros símbolos de γ"
leia-se: "inspecionando-se no máximo os k primeiros símbolos de β"

112. Página 446, exercício 18, última linha
onde se lê:
"produções unitárias"
leia-se:
"regras unitárias e regras vazias"

113. Página 447, exercício 21, definição de δ
onde se lê:
"$(q_0, b, Z_0) \rightarrow (q_0, Z_0B)$"
leia-se:
"$(q_0, b, Z_0) \rightarrow (q_0, BZ_0)$"

114. Página 585, sétimo parágrafo
onde se lê: "Os termos **solucionável**, **não-solucionável** (ou **insolúvel**) e **parcialmente solucionável**, também empregados quando se trata de problemas, significam, respectivamente, que as correspondentes linguagens são: (i) recursivas, ou seja, podem sempre ser decididas no caso geral; (ii) recursivamente enumeráveis, ou seja, não podem ser decididas no caso geral; e (iii) recursivamente enumeráveis, enfatizando o fato de que pode haver solução para algumas instâncias do problema (ainda que correndo o risco de se esperar indefinidamente por uma resposta).

Aplicada ao estudo dos problemas de decisão, a decidibilidade indica se os mesmos são solucionáveis, não-solucionáveis ou parcialmente solucionáveis."

leia-se: "Os termos (i) **solucionável**, (ii) **não-solucionável**, (iii) **parcialmente solucionável** e (iv) **completamente insolúvel**, empregados quando se trata de problemas, significam, respectivamente, que as correspondentes linguagens são: (i) recursivas; (ii) não recursivas; (iii) recursivamente enumeráveis e (iv) não recursivamente enumeráveis.

Aplicada ao estudo dos problemas de decisão, a decidibilidade indica se os mesmos são solucionáveis, não-solucionáveis, parcialmente solucionáveis ou completamente insolúveis."

115. Em todo o texto
os algoritmos que estão contidos dentro de teoremas estão referenciados de forma incorreta no texto principal. No lugar dos algoritmos, os números (como em "Algoritmo X.Y") estão se referindo aos teoremas onde eles estão contidos (no caso, "Teorema X.Y"). As identificações corretas dos algoritmos são aquelas que constam da Lista de Algoritmos, nas páginas 29 e 30, e também aquelas que constam no título (primeira linha) dos respectivos algoritmos.