



LFA - Aula 03

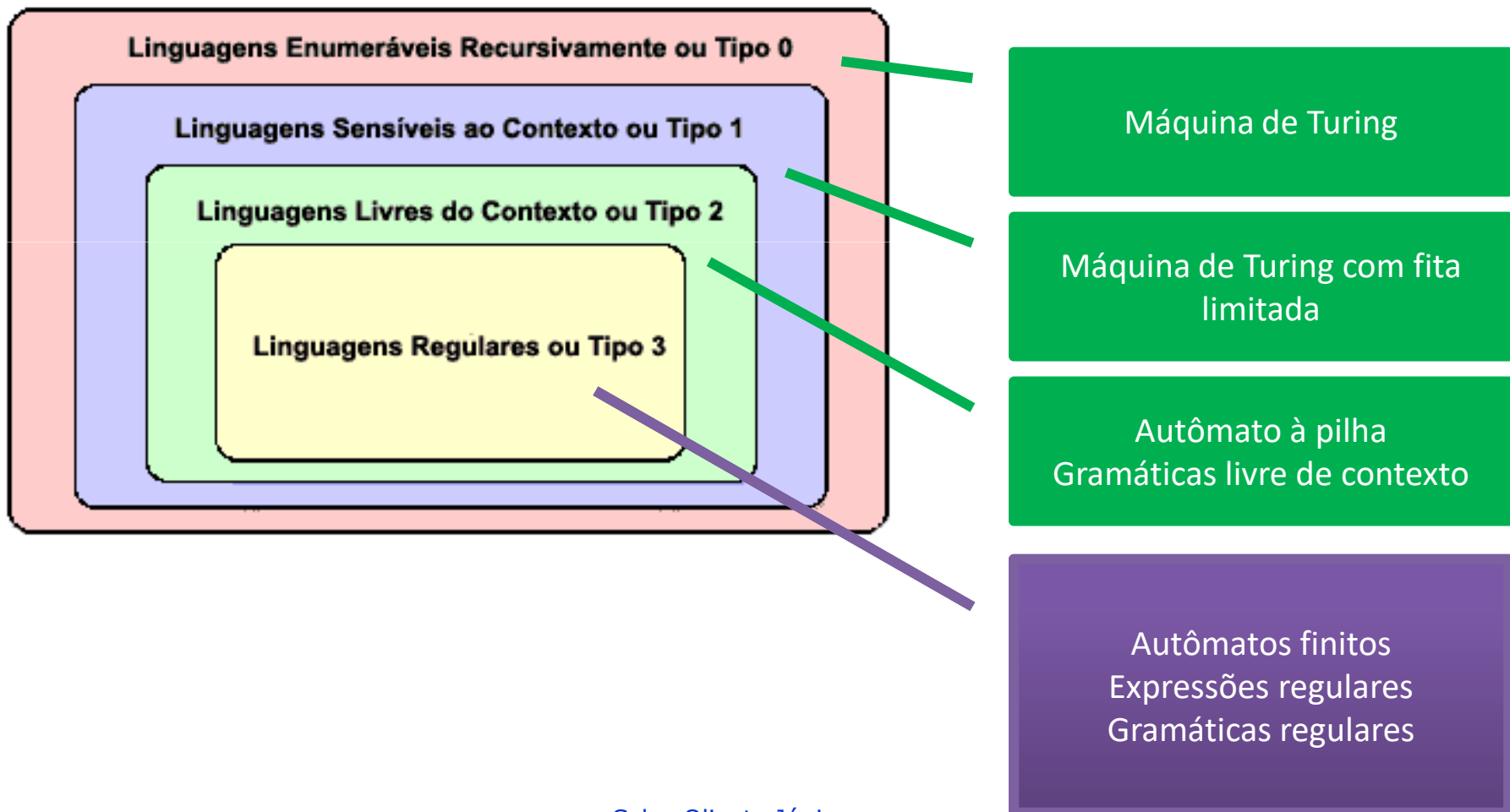
Linguagens regulares -
Gramáticas regulares

25-27/09/2017

Celso Olivete Júnior

olivete@fct.unesp.br

Classificação das Linguagens segundo Hierarquia de Chomsky





Na aula passada...

- Expressões regulares



Na aula de hoje...

- Linguagens regulares: **Gramáticas regulares**
- Referência bibliográfica

RAMOS, M.V.M.; NETO, J.J.; VEGA, I.S. *Linguagens Formais: Teoria, Modelagem e Implementação*. Editora Bookman 2009. → Capítulo 3



Relembrando...

- Uma linguagem regular é o conjunto de linguagens reconhecida/gerada pelos seguintes formalismos:
 - **Expressões regulares**
 - Gramáticas regulares
 - Autômatos finitos



Os operadores de expressões regulares

- Os tipos de operadores sobre as ER's são:
 - União (+)
 - Concatenação (.)
 - Fechamento (*)



Exemplos de expressões regulares

Exemplos	
ER para strings que são formadas por zero seguido por qualquer ocorrência de 1 (inclusive nenhuma)	$ER = 01^*$
ER formada por todas as palavras sobre (a,b) contendo aa como subpalavra	$ER = (a+b)^*aa(a+b)^*$



Relembrando...

- Uma linguagem regular é o conjunto de linguagens reconhecido/gerado pelos seguintes formalismos:
 - Expressões regulares
 - **Gramáticas regulares**
 - Autômatos finitos



Gramática

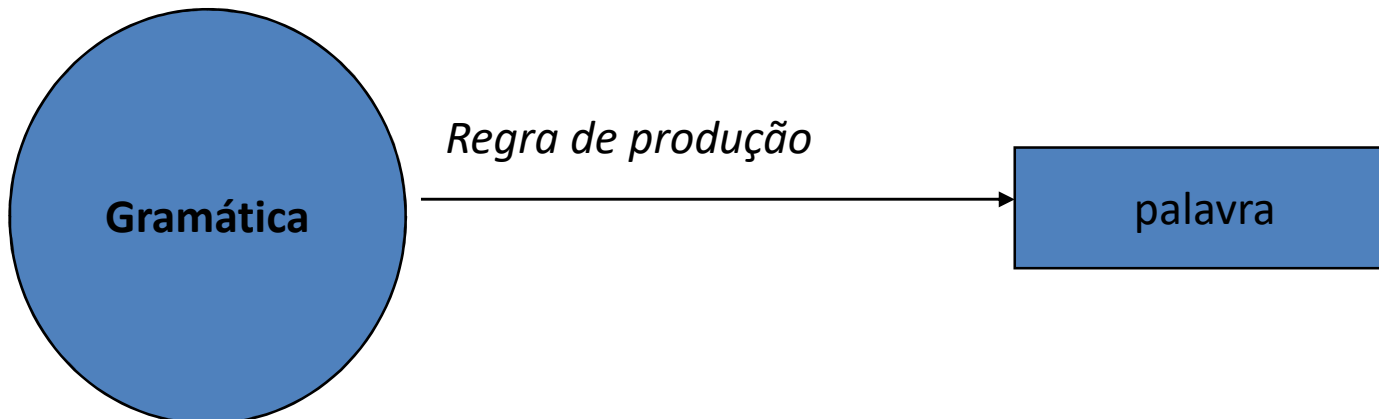
- Uma **gramática** consiste em **uma ou mais variáveis** que **representam linguagens**.

- Exemplo: linguagem dos palíndromos (permite a mesma leitura da esquerda para direita quanto da direita para esquerda). Ex: **Anita latina**
 - Se w é um palíndromo, então $0w0$ e $1w1$ são palíndromos.
 - Neste caso a linguagem é formada apenas por uma variável (**w**).



Gramáticas

- Uma **gramática** consiste em **uma ou mais variáveis** que **representam linguagens**.





Gramáticas

- Formalmente as gramáticas são caracterizadas como quádruplas ordenadas

$$G = (\{V\}, T, P, S)$$

• onde:

- V representa o vocabulário não terminal da gramática - **variáveis**.
- T é o vocabulário **terminal**, contendo os símbolos que constituem as sentenças da linguagem.
- P representa o conjunto de todas as leis de formação (**regras de produção**) utilizadas pela gramática para definir a linguagem.
- S representa o símbolo de **início**



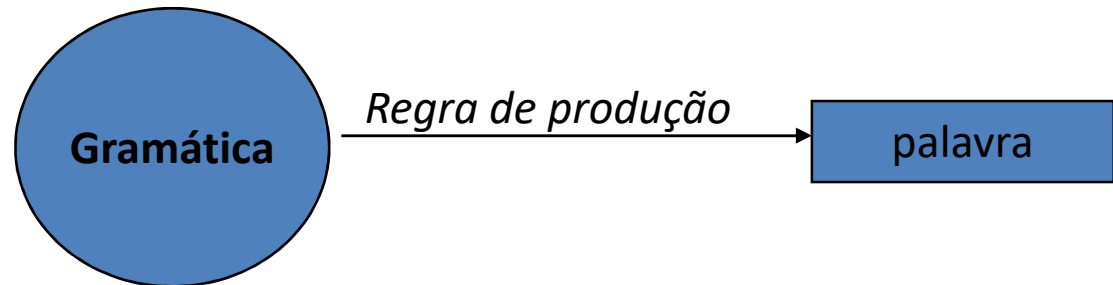
Gramáticas

• $G = (\{S,A,B\}, \{a,b\}, P, S)$

• $P = \{ S \rightarrow AB$

$A \rightarrow a \mid AB$

$B \rightarrow b \}$



• Passos para gerar a palavra: **abb**

• $S \rightarrow AB \rightarrow ABB \rightarrow aBB \rightarrow abB \rightarrow abb.$

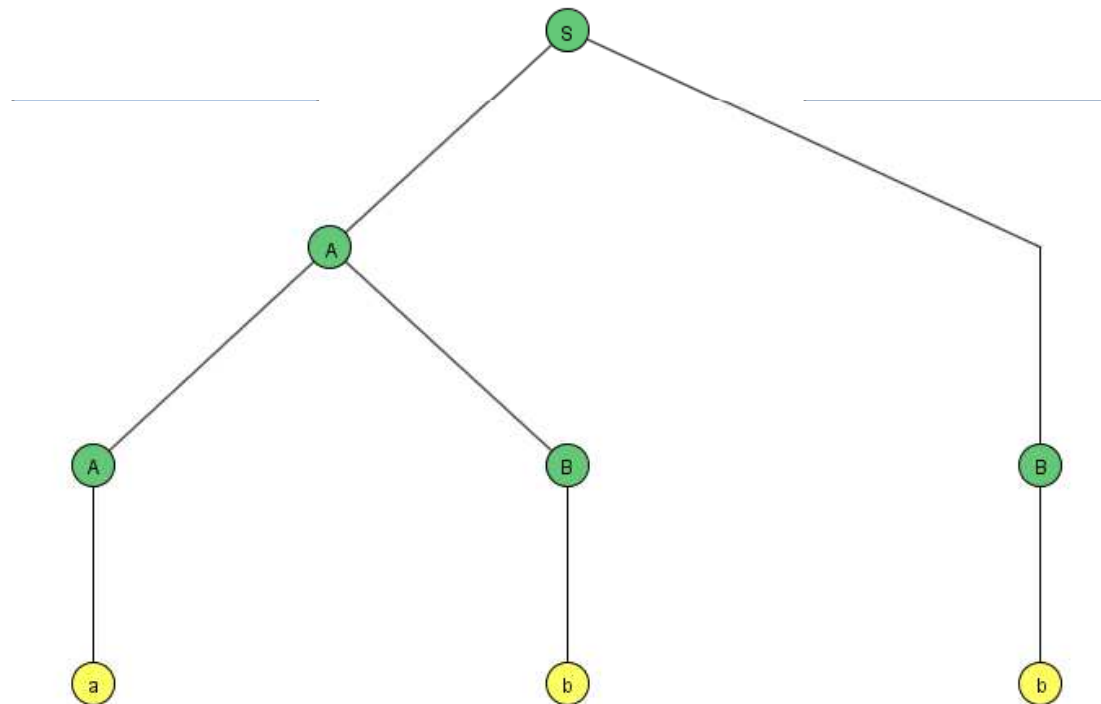


Gramática no JFlap

Input: `abb`

String accepted! 13 nodes generated.

LHS		RHS
S	→	AB
A	→	a
A	→	AB
B	→	b





Gramáticas

■ Notação / Convenções

- Variáveis: letras do alfabeto maiúsculas $\{A, B, \dots, Z\}$
- Terminais: letras do início do alfabeto minúsculas $\{a, b, c, \dots\}$, dígitos $\{0..9\}$ e outros caracteres como $+, -, *, /$
- Não-Terminais: letras do fim do alfabeto maiúsculas, como X ou Y , são terminais ou variáveis



Gramáticas

• podem ser classificadas em gramáticas lineares à direita (GLD) ou à esquerda (GLE), cujas regras $\alpha \rightarrow \beta$ são da forma:

- $\alpha \in V$ - α é um não terminal (Lado esquerdo deve ter apenas não terminais)
- GLD: $\beta \in (T \cup \{\epsilon\}) (V \cup \{\epsilon\})$ - $A \rightarrow wB$ ou $A \rightarrow w$
- GLE: $\beta \in (V \cup \{\epsilon\}) (T \cup \{\epsilon\})$ - $A \rightarrow Bw$ ou $A \rightarrow w$.



Gramáticas lineares obedecem a regra $\alpha \rightarrow \beta$

- $\alpha \in V$

- o lado esquerdo da regra é formado por um símbolo não terminal

- **GLD:** $\beta \in (T \cup \{\epsilon\}) (V \cup \{\epsilon\})$ - $A \rightarrow wB$ ou $A \rightarrow w$ com $|w| \geq 0$

- o lado direito da regra é formado por N símbolos terminais seguido de UM símbolo não terminal OU formado apenas por N símbolos terminais

- **GLE:** $\beta \in (V \cup \{\epsilon\}) (T \cup \{\epsilon\})$ - $A \rightarrow Bw$ ou $A \rightarrow w$ com $|w| \geq 0$

- o lado direito da regra é formado por UM símbolo não terminal seguido de N símbolos terminais OU formado apenas por N símbolos terminais



Gramáticas lineares

- *GLD* e *GLE* geram exatamente a mesma classe de linguagens. Portanto, é indiferente o emprego de uma ou outra dessas duas variantes de gramática, já que ambas possuem a mesma capacidade de representação de linguagens.
- As linguagens geradas por *GLD* e *GLE* são as linguagens regulares
 - Logo, *GLD* e *GLE* são equivalentes



Gramáticas regulares

- Ex: dada a linguagem representada pela ER

$(a+b)^*(aa+bb)$

qual a gramática que a reconhece?



Gramáticas regulares

- Ex: dada a linguagem representada pela ER

$(a+b)^*(aa+bb)$

qual a gramática que a reconhece?

$$G = (\{S,A\}, \{a,b\}, P, S)$$

$$P = \{ S \rightarrow aS \mid bS \mid A, \\ A \rightarrow aa \mid bb \}$$



Gramáticas regulares

- Ex: dada a linguagem representada pela ER

$(a+b)^*(aa+bb)$

qual a gramática que a reconhece?

$$G = (\{S,A\}, \{a,b\}, P, S)$$

$$P = \{ S \rightarrow aS \mid bS \mid A,$$

$$A \rightarrow aa \mid bb \}$$

- mostre os passos para reconhecer a palavra **babb**



Gramáticas regulares

- Ex: dada a linguagem representada pela ER

$(a+b)^*(aa+bb)$

qual a gramática que a reconhece?

$$G = (\{S,A\}, \{a,b\}, P, S)$$

$$P = \{ S \rightarrow aS \mid bS \mid A,$$

$$A \rightarrow aa \mid bb \}$$

- mostre os passos para reconhecer a palavra **babb**

$S \rightarrow bS \rightarrow baS \rightarrow baA \rightarrow babb$



Extensões para GLD e GLE

- Gramática Linear Unitária à Direita (**GLUD**)
 - como na gramática linear à direita.
 - Adicionalmente $|w| \leq 1$ no máximo um terminal do lado direito da regra
- Gramática Linear Unitária à Esquerda (**GLUE**)
 - como na gramática linear à esquerda.
 - Adicionalmente $|w| \leq 1$ no máximo um terminal do lado direito da regra



Gramáticas regulares

- Ex: dada a linguagem representada pela ER $a(ba)^*$ dê as GLD, GLE, GLUD e GLUE que as reconhece.

- mostre os passos para reconhecer a palavra **ababa**



Gramáticas regulares

- Ex: dada a linguagem representada pela ER $a(ba)^*$ dê as **GLD**, **GLE**, **GLUD** e **GLUE** que as reconhece

$$G = (\{S,A\}, \{a,b\}, P, S)$$

$$P = \{ S \rightarrow aA$$

$$A \rightarrow baA \mid \varepsilon \}$$

- mostre os passos para reconhecer a palavra **ababa**

$$S \rightarrow aA \rightarrow abaA \rightarrow ababaA \rightarrow ababa\varepsilon$$



Gramáticas regulares

- Ex: dada a linguagem representada pela ER $a(ba)^*$ dê as GLD, **GLE**, GLUD e GLUE que as reconhece

$$G = (\{S,A\}, \{a,b\}, P, S)$$
$$P = \{ S \rightarrow Sba \mid a \}$$

- mostre os passos para reconhecer a palavra **ababa**

$$S \rightarrow Sba \rightarrow Sbaba \rightarrow ababa$$



Gramáticas regulares

- Ex: dada a linguagem representada pela ER $a(ba)^*$ dê as GLD, GLE, **GLUD** e GLUE que as reconhece

$$G = (\{S,A\}, \{a,b\}, P, S)$$

$$P = \{ S \rightarrow aA$$

$$A \rightarrow bB \mid \varepsilon$$

$$B \rightarrow aA$$

- mostre os passos para reconhecer a palavra **ababa**

$$S \rightarrow aA \rightarrow abB \rightarrow abaA \rightarrow ababB \rightarrow ababaA \rightarrow ababa\varepsilon$$



Gramáticas regulares

- Ex: dada a linguagem representada pela ER $a(ba)^*$ dê as GLD, GLE, GLUD e **GLUE** que as reconhece

$$G = (\{S,A\}, \{a,b\}, P, S)$$

$$P = \{ S \rightarrow Aa \mid a \\ A \rightarrow Sb \}$$

- mostre os passos para reconhecer a palavra **ababa**

$$S \rightarrow Aa \rightarrow Sba \rightarrow Aaba \rightarrow Sbaba \rightarrow ababa$$



Exercícios

- Para cada uma das linguagens, construir a Gramática (indicar se é GLE, GLD, GLUE ou GLUD) e a Expressão Regular. As gramáticas podem ser resolvidas com o JFlap.

❖ Comprimento:

- Igual a ...
- Maior (ou igual) a ...
- Menor (ou igual) a ...
- Par
- Ímpar
- Múltiplo de ...

❖ Símbolos e subcadeias:

- Começa com ...
- Termina com ...
- Contém ...
- Contém exatamente tantas ocorrências ...
- Contém no mínimo tantas ocorrências ...
- Contém no máximo tantas ocorrências ...
- Justaposição

❖ Combinações:

- Negação
- E
- Ou
- Ou exclusivo

$\Sigma = \{a,b,c\}$

Cadeias de comprimento
qualquer, incluindo zero.

$\{\epsilon, a, b, c, aa, ab, ac, ba, bb, bc,$
 $ca, cb, cc, aaa, aab, \dots \}$

1

$\Sigma = \{a,b,c\}$

Cadeias de comprimento
qualquer, maior que zero.

{a, b, c, aa, ab, ac, ba, bb, bc,
ca, cb, cc, aaa, aab, ... }

2

$\Sigma = \{a,b,c\}$

Cadeias de comprimento 3.

{bca, aab, aca, bab, cab, acc, abb,
abc, acb, aaa, cbb, baa, ... }

3

$\Sigma = \{a,b,c\}$

Cadeias de comprimento
diferente de 3.

{a, bc, bbcc, bcabaab, bcaa, c,
 ϵ , acababab, acaacabbab,
cabacacb, aabc, babac, ba,
abaaa, bbcb, ... }

4

$\Sigma = \{a,b,c\}$

Cadeias de comprimento maior que 3.

{bbcc, bcabaab, bcaa,
cababab, acaacabbab,
cabacacb, aabc, babac, abaaa,
bbcb, ... }

5

$\Sigma = \{a,b,c\}$

Cadeias de comprimento
maior ou igual a 3.

{bbc, bcabaab, bcaa,
cababab, acaacabbab,
cabacacb, aabc, babac, abaaa,
bcb, aaa, ... }

6

$\Sigma = \{a,b,c\}$

Cadeias de comprimento
menor que 3.

$\{\epsilon, a, b, c, aa, ab, ac, ba, bb,$
 $bc, ca, cb, cc\}$

7

$\Sigma = \{a,b,c\}$

Cadeias de comprimento
múltiplo de 3.

{bca, acabababb, ϵ , acabab,
cabacacbb, aabcbabaccba,
baaaba, aaa, bbbbbbb,
aabaacbabb, aac, ... }

8

$\Sigma = \{a,b,c\}$

Cadeias de comprimento
múltiplo de 4.

{bcaa, acababab, ε ,
aabcbabaccba, aaac, ... }

9

$\Sigma = \{a,b,c\}$

Cadeias com uma quantidade
par de símbolos.

$\{\varepsilon, bb, ac, aabc, abac, abbc,$
 $abcc, acac, acbc, aaaacb,$
 $bababc, \dots \}$

10

$\Sigma = \{a,b,c\}$

Cadeias com uma quantidade ímpar de símbolos.

{bcb, acbbb, a, c, aabcbbb,
bbbacbbba, abc, cbabc, aaa, ... }

11

$\Sigma = \{a,b,c\}$

Cadeias iniciando com “abb”.

{abb, abba, abbab, abbabb,
abbcabbc, abbccbbb, ... }

12

$\Sigma = \{a,b,c\}$

Cadeias que não iniciam com
“aa”.

{abb, aba, abbb, bbabb,
bcabbc, babbccc, ... }

13

$\Sigma = \{a,b,c\}$

Cadeias terminando com 3
símbolos “b” consecutivos.

{bbb, acbbb, abcbbb,
abacbbb, abbb, bbbb,
acacbbb, bbbacbbb,
abababbbb, ... }

14

$\Sigma = \{a,b,c\}$

Cadeias terminando com 3
símbolos “b” consecutivos, e
não mais que isso.

{bbb, acbbb, abcbbb,
abacbbb, abbb, acacbbb,
bbbacbbb, abababbb, ... }

15

$\Sigma = \{a,b,c\}$

Cadeias que não terminam
com 2 símbolos “b”
consecutivos.

{a, b, c, acbba, aab, abacbbc,
abcc, acacbc, bbbacaa,
ababa, ccc, ... }

16

$\Sigma = \{a,b,c\}$

Cadeias iniciando com “a” e terminando com “c”.

{ac, abc, acc, aac, aabc, abac, abbc, abcc, acac, acbc, aaaac, aaabc, ... }

17

$\Sigma = \{a,b,c\}$

Cadeias que iniciam com “a” e não terminam com “c”.

{a, ab, acb, aaca, aabcb, aba, abb, abcca, acaca, acb, aaaa, aaab, ... }

18

$\Sigma = \{a,b,c\}$

Cadeias que não iniciam com
“a” e que terminam com “c”.

{c, bc, bac, bbc, , ccc, babcb,
babac, babbc, bbcc, cacacac,
cbc, ccccc, bbbbc, ... }

19

$\Sigma = \{a,b,c\}$

Cadeias que não iniciam com
“a” e não terminam com “c”.

{baca, cb, cacb, caaa, caabcb,
babacb, babbcb, cabccb, ca,
bb, bcab, bbbcb, ... }

20

$\Sigma = \{a,b,c\}$

Cadeias com exatamente 3
símbolos “b”.

{bcbb, acbbb, bbab, cbbb,
aabcbb, bacabccba, bbbc,
cbabcba, ababab, ... }

21

$\Sigma = \{a,b,c\}$

Cadeias com pelo menos 2
símbolos “a”.

{bcbaab, acbabab, aaabbab,
cababab, aabcbabaa,
bacabcaacba, aaabaaabbc,
cbabacaba, abbab, ... }

22

$\Sigma = \{a,b,c\}$

Cadeias com no máximo 4
símbolos “c”.

{bcbaab, acbabab, ccabbab,
cabacacb, aabcbabac,
baabaaba, aaabbc, ccabcac,
abcbab, ccc, ... }

23

$\Sigma = \{a,b,c\}$

Cadeias que contenham no mínimo 2 símbolos “a” ou no máximo 3 símbolos “c”, de forma não exclusiva.

{abccabc, abaccbcb,
aaabcc, acccbc, abcabcabc,
cababc, aa, ababbabca, ccc, ... }

24

$\Sigma = \{a,b,c\}$

Cadeias com no mínimo 3 e no máximo 5 símbolos “a”.

25

{bcabaab, acababab,
acaacabbab, cabacacb,
aabcbabac, baaba, aaabbc,
acacabcac, aabaacbab, aaaa, ... }

$\Sigma = \{a,b,c\}$

Cadeias que iniciam e terminam
com símbolos diferentes.

26

{abccabc, abaccbcb,
caabca,acccb, abc, bababc, ba,
bacacc, bcbabca, cca, ... }

$\Sigma = \{a,b,c\}$

Cadeias que não possuem
símbolos “a” à direita de
símbolos “b”, nem símbolos
“c” à direita de símbolos “b”.

{abcc, abbbbb, cccc, aabbcc,
abc, bbbc, b, aaa, aacccc, bc,
 ϵ , abc, ... }

27

$\Sigma = \{a,b,c\}$

28

Cadeias que possuem uma seqüência de um ou mais símbolos “b” imediatamente à direita de cada símbolo “a”.

{abccabc, abbabbbccbc, caabca, abcccb, abc, bababc, b, bacacc, bcbabca, ccabb, ... }

$\Sigma = \{a,b,c\}$

Cadeias que não contenham
símbolos “b” justapostos.

{abccabc, abaccbcb,
aaabcc, acccbc, abcabcabc,
cababc, aa, bacacc, ababcabca,
ccc, ... }

29

$\Sigma = \{a,b,c\}$

Cadeias com uma quantidade
par de símbolos “b”.

30

{bb, bcb, bbcc, bcababab, caa, c,
 ϵ , acbababab, acaacabba,
cabacacb, ababc, bbabbac,
babbb, aaaa, cbb, ... }

$\Sigma = \{a,b,c\}$

Cadeias com uma quantidade ímpar de símbolos “c”.

31

{bbc, bcb, cbbcc, bcababab, caa,
c, acbababab, acaacabcba,
cacbaccacb, ababc, bbabbac,
cbccabbb, acacaca, cbb, ... }

$\Sigma = \{a,b,c\}$

Cadeias com quantidade
par de símbolos “a” e
ímpar de símbolos “c”.

{cabccabcc, aaacacbc,
bccc,cb, aabcabacaabc,
cabccabcc, accca, bacacc,
aca, ccc, ... }

32

$\Sigma = \{a,b,c\}$

Cadeias que contenham a subcadeia “abc”.

{abcb, babcb, bbabcc, abc,
abcaabcb, cbabc, ababbabca, ... }

33

$\Sigma = \{a,b,c\}$

Cadeias que contenham pelo menos três símbolos iguais consecutivos.

{abbb, cacccbab,
bbbbbbccc, bbaaa, aaaaa,
ccccbabc, abaaabbabca, ... }

34

$\Sigma = \{a,b,c\}$

Cadeias que não contenham
dois símbolos consecutivos
iguais.

{abcb, cacbcbab,
bababababcacbcac, babababa,
acabacaca, cbabc, a, b ... }

35

$\Sigma = \{a,b,c\}$

Cadeias que não contenham o símbolo “a”.

{cbbbc, ccbc, bb,cb,
bbabaabb, babb, aaa, ϵ ,
aaabbb, aababa, baa, ... }

36

$\Sigma = \{a,b,c\}$

Cadeias que não contenham a subcadeia “ab”.

{cbacbc, acacbc, acacbb, caaaa, aacbbbacacbbc, cbbb, aaa, ϵ , aaacbbb, bbac, cccbaa, ... }

37

$\Sigma = \{a,b,c\}$

Cadeias que não contenham a subcadeia “abc”.

{cababbc, acacbc, b,
aabb, caaaba, aabbabacabbc,
cbabb, aaa, ϵ , aaabbb, aababac,
cccbaa, ... }

38

- Identificadores utilizados em linguagens de programação de alto nível qualquer
- Conjunto dos símbolos utilizados por uma linguagem de programação qualquer

39