Expressões regulares

1. Histórico e motivação
2. Definição
   a) Sintaxe
   b) Semântica
   c) Precedência dos operadores
3. Exemplos
4. Leis algébricas
5. Dialetos
6. Aplicações
7. Exercícios

Pré-requisito: básico de conjuntos (principais operações) e de teoria de linguagens (símbolos, alfabetos, cadeias e linguagens como conjuntos)

Histórico e motivação

- São uma alternativa, juntamente com os autômatos finitos e as gramáticas lineares à direita ou à esquerda, para a representação formal de linguagens regulares (tipo 3);
- Pela sua natureza declarativa, normalmente são concisas e fáceis de serem manipuladas;
- Foram definidas por Kleene em 1956, no artigo “Representation of events in nerve nets and finite automata”;
- Demonstra-se a equipotência das expressões regulares com os autômatos finitos e as gramáticas lineares à esquerda ou à direita;
- São largamente utilizadas em linguagens de programação, ferramentas de busca de padrões e geradores automáticos de analisadores léxicos.

Definição

Expressão regular X sobre um alfabeto Σ:

1. ∅ é uma expressão regular sobre Σ e representa a linguagem vazia;
2. ε é uma expressão regular sobre Σ e representa a linguagem formada pela cadeia vazia {ε};
3. $\forall \sigma \in \Sigma$, σ é uma expressão regular sobre Σ e representa a linguagem formada pela cadeia unitária {σ}

Sejam X e Y são duas expressões regulares quaisquer sobre Σ. Então:

7. **Concatenação**: X.Y, ou simplesmente XY, é uma expressão regular sobre Σ, e representa a linguagem $\{w=xy \mid x \in X \text{ e } y \in Y\}$;
8. **União**: X|Y é uma expressão regular sobre Σ e representa a linguagem $\{w \mid w \in X \text{ ou } w \in Y\}$;
9. **Fecho recursivo e transitivo**: X* é uma expressão regular sobre Σ e representa a linguagem $X^0 \cup X^1 \cup \ldots \cup X^n \cup \ldots$

Precedência dos operadores

1. Fecho recursivo e transitivo (maior)
2. Concatenação
3. União (menor)

Portanto...

a|bcd* representa a mesma linguagem que (a)|(bc(d)*)

⇨ Como nas expressões aritméticas, os parêntesis podem ser usados livremente para modificar, temporariamente, a precedência original dos operadores:

a|bcd*, (a|b)cd*, (a|bc)d*, a|b(cd)*, a|(bcd)* representam linguagens diferentes.

Exemplos

Expressões regulares sobre o alfabeto Σ={a}:

- ∅, denotando a linguagem {}
- ε, denotando a linguagem {ε}
- a, denotando a linguagem {a}
- a|ε, denotando a linguagem {a, ε}
- aa, denotando a linguagem {aa}
- a*, denotando a linguagem {ε, a, aa, aaa, ...}
- aa*, denotando a linguagem {a, aa, aaa, aaaa, ...}, ou seja, cadeias de comprimento mínimo 1
- a(aa)*, denotando a linguagem {a, aaa, aaaaa, aaaaaaa, ...}, ou seja, cadeias de comprimento ímpar

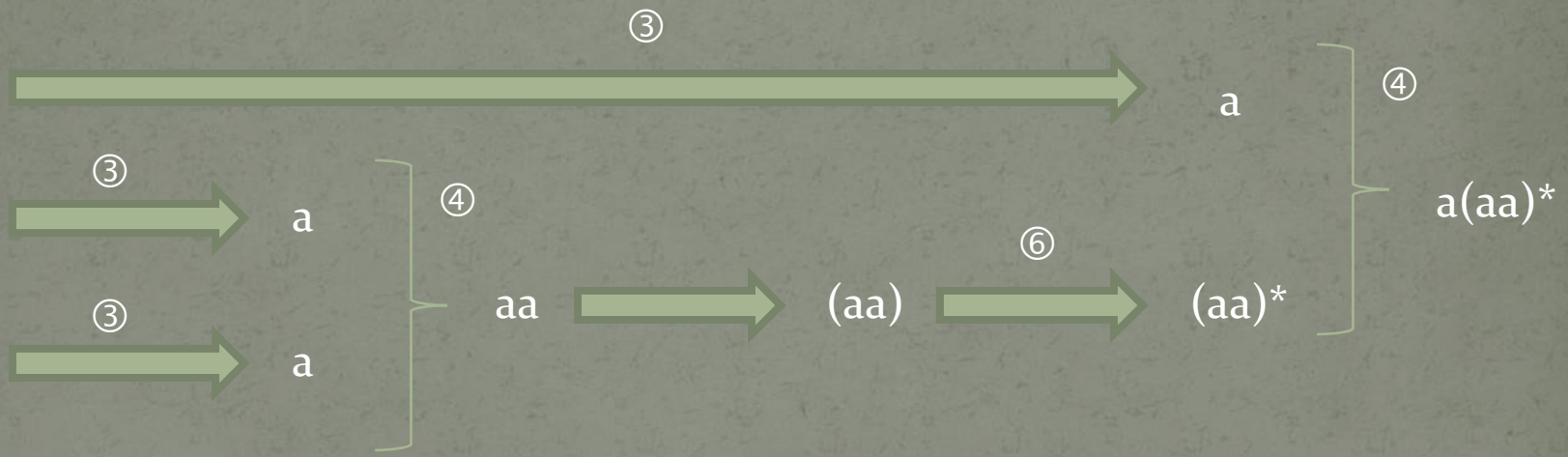
Análise estrutural

Expressão regular sobre o alfabeto {a}: a(aa)*

- “a” é uma expressão regular (regra número 3);
- Se “a” é uma expressão regular, então “aa” também é (regra número 4);
- Se “aa” é uma expressão regular, então (aa) também é;
- Se (aa) é uma expressão regular, então (aa)* também é (regra número 6);
- Se “a” e (aa)* são expressões regulares, então a(aa)* também é (regra número 4 novamente).

Análise estrutural

Expressão regular sobre o alfabeto {a}: a(aa)*



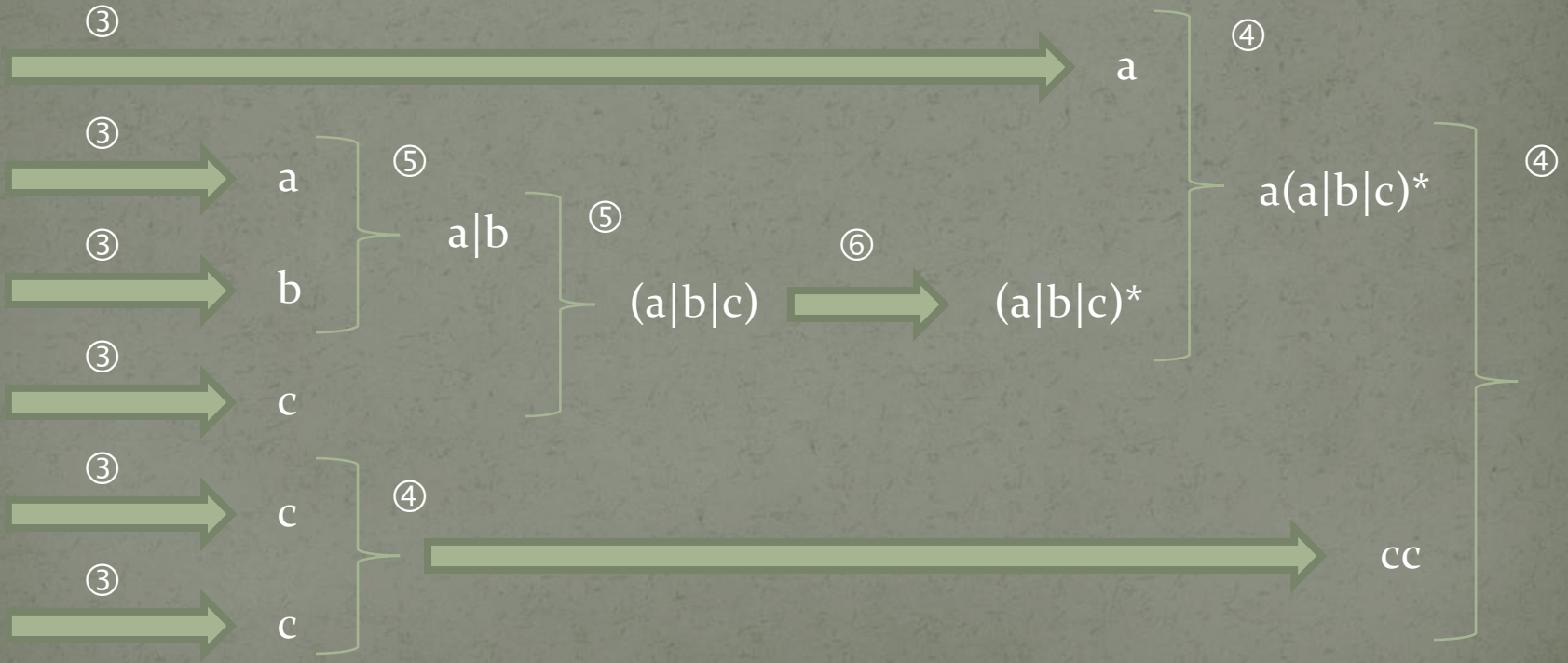
Exemplos

Expressões regulares sobre o alfabeto Σ={a,b,c}:

- ∅
- ε
- a
- b
- c
- a|b
- cb
- a(a|b|c)*cc, denotando a linguagem formadas pelas cadeias que iniciam com o símbolo “a” e terminam com dois símbolos “c” consecutivos;
- (a|b|c)*a(a|b|c)*a(a|b|c)*, denotando a linguagem formada pelas cadeias com pelo menos dois símbolos “a”;
- ((b|c)*a(b|c)*a)* (b|c)*a(b|c)*a(b|c)*, denotando a linguagem formada pelas cadeias com uma quantidade par, maior que zero, de símbolos “a”.

Análise estrutural

Expressão regular sobre o alfabeto {a,b,c}: a(a|b|c)*cc



Leis algébricas

Associatividade

- União:
  (X | Y) | Z = X | (Y | Z)
  ({a} | {b}) | {c} = {a} | ({b} | {c}) = {a,b,c}

- Concatenação:
  (X Y) Z = X (Y Z)
  ({a} {b}) {c} = {a} ({b} {c}) = {abc}

Leis algébricas

Comutatividade

- União:
  X | Y = Y | X
  {a,b} | {c,d} = {c,d} | {a,b} = {a,b,c,d}

- Concatenação:
  Não se aplica
  {a,b} {c,d} ≠ {c,d} {a,b}

Leis algébricas

Elemento neutro

- União:
  $X \mid \varnothing = \varnothing \mid X = X$
  $\{a,b\} \mid \varnothing = \varnothing \mid \{a,b\} = \{a,b\}$

- Concatenação:
  $X\varepsilon = \varepsilon X = X$
  $\{a,b\}\ \varepsilon = \varepsilon\ \{a,b\} = \{a,b\}$

Leis algébricas

Distributividade da concatenação sobre a união

- Esquerda:
  X (Y | Z) = X Y | X Z
  {a} ({b} | {c}) = {a}{b} | {a}{c} = {ab,ac}

- Direita:
  (Y | Z) X = Y X | Z X
  ({b} | {c}) {a} = {b}{a} | {c}{a} = {ba,ca}

Leis algébricas

Outras identidades

- $\emptyset X = X \emptyset = \emptyset$
- $X \mid X = X$
- $(X^*)^* = X^*$
- $\emptyset^* = \varepsilon$
- $\varepsilon^* = \varepsilon$
- $\emptyset \mid \varepsilon = \varepsilon$
- $X \mid X^* = X^*$

grep	Notação algébrica original
[Aa]	(A|a)
[0-3]	(0|1|2|3)
[Aa0-3]	(A|a|0|1|2|3)
a*	a*
a+	aa*
a?	(a|ε)
a{2}	aa
a{2,}	aaa*
a{2,4}	aa|aaa|aaaa
\(ab\)*	(ab)*
.	Qualquer caracter
.*	Qualquer cadeia
abc|cba	abc|cba

Aplicações

- Ferramentas e linguagens de script:
  - grep '\(abc\)*' arquivo
  - awk '/\(abc\)*/ {print $1} ' arquivo
- Geradores de analisadores léxicos:
  - [A-Za-z][A-Za-z0-9]* (lex, C)
  - JavaLex (Java), Coco-R (Java, C++, C#) etc
- Sistemas de busca de padrões:
  a) Bancos de dados, Editores de texto etc
  b) Biologia, genética
- Linguagens de programação:
  a) Java (java.util.regex)
  b) C++, C# etc

Esta é a teoria.

Agora vamos aos exercícios.

Dúvidas?

Exercícios

Para cada uma das expressões regulares sobre o alfabeto {a,b,c} apresentadas a seguir, (i) faça uma análise estrutural da mesmas, e (ii) apresente 5 cadeias pertencentes às respectivas linguagens.

- abb(ccc)*
- (a|bb|ccc)*
- a*|a*bb*|(ab*|(ab)*)*

Exercícios

Descreva, informalmente, da forma mais clara e concisa possível, e com o auxílio de exemplos, as linguagens representadas pelas seguintes expressões regulares sobre o alfabeto {a,b}:

- (a|b)(a|b)(a|b)*
- (b*ab*a)*b*|a*b(a*ba*ba*)*
- b*|b*ab*|b*ab*ab*|b*ab*ab*ab*a(a|b)*

Exercícios

Obtenha expressões regulares que representem as seguintes linguagens definidas de maneira informal:

- Conjunto de todas as cadeias sobre o alfabeto {a,b,c} tais que elas contém apenas 3 símbolos "b", todos eles consecutivos (exemplos: *acbbb*, *abbbccaa*);
- Conjunto de todas as cadeias sobre o alfabeto {a,b,c} tais que elas contém apenas 3 símbolos "b", não consecutivos (exemplos: *babccb*, *cabcbaabca*);
- Conjunto de todas as cadeias sobre o alfabeto {a,b,c} tais que elas contém apenas 3 símbolos "b", sendo que o primeiro "b" deve ser separado do segundo por pelo menos um símbolo e o segundo do terceiro por pelo menos dois símbolos (exemplos: *babccbcc*, *acabacabccbaa*).