Expressões Regulares

MCTA015-13 - Linguagens Formais e Automata

Prof. Maycon Sambinelli

m.sambinelli@ufabc.edu.br

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC

UFABC

Aula de Hoje

Objetivos de Aprendizagem

- Aprender a descrever linguagens com expressões regulares
- Compreender qual linguagem pode ser descrita por uma expressão regular
- Aprender a converter uma expressão regular em um AFN
- Aprender a converter um AFD em uma expressão regular

Introdução à Expressões Regulares

- Uma **expressão regular (regex)** é um formalismo matemático que nos permite descrever uma linguagem de forma concisa.
  - Usando as operações regulares: união, concatenação e estrela.
  - descrições algébricas
- O valor de uma expressão regular é uma linguagem.
  - O valor de uma expressão aritmética é um número.

Exemplo de uma expressão regular

✎ Exemplo

Seja γ a expressão regular definida como

$$\gamma = (\mathtt{0} \cup \mathtt{1})\mathtt{0}^*,$$

Então

$$L(\gamma) = \{\omega \in \{0,1\}^* : \text{ começam com } 0 \text{ ou } 1 \text{ e são seguidas por um número de qualquer de } 0\text{'s } \}$$

Expressão Regular

Definição

Dizemos que γ é uma **expressão regular** sobre um alfabeto Σ, descrevendo a linguagem $L(\gamma)$, se

- $\gamma = \mathsf{a}$, para algum $a \in \Sigma$, onde $L(\gamma) = \{a\}$
- $\gamma = \boldsymbol{\varepsilon}$, onde $L(\gamma) = \{\varepsilon\}$
- $\gamma = \boldsymbol{\emptyset}$, onde $L(\emptyset) = \emptyset$
- Sejam α e β expressões regulares
  - $\gamma = \alpha \cup \beta$, onde $L(\gamma) = L(\alpha) \cup L(\beta)$
  - $\gamma = \alpha\beta$, onde $L(\gamma) = L(\alpha)L(\beta)$
  - $\gamma = \alpha^*$, onde $L(\gamma) = L(\alpha)^*$

Precedência: estrela tem precedência sobre concatenação que, por sua vez, tem precedência sobre a união.

Precedência

A ordem de precedência pode ser alterada através da parentização de subexpressões regulares.

- $\gamma = (\alpha)$, onde $L(\gamma) = L(\alpha)$

✎ Exemplo

Qual a linguagem descrita por $\gamma = 0(a \cup b)^*$?

A das cadeias que começam com 0 e que são seguidas de símbolos a's e b's.

Equivalência de Expressões Regulares

Duas expressões regulares α e β são **equivalentes**, denotado por $\alpha \equiv \beta$, se $L(\alpha) = L(\beta)$.

✎ Exemplo

- $(\mathtt{0} \cup \mathtt{1})^* \equiv (\mathtt{0}^*\mathtt{1}^*)^*$
- $(\mathtt{0} \cup \varepsilon)\mathtt{1} \equiv \mathtt{01} \cup \mathtt{1}$

Abuso de notação

As vezes vamos denotar uma regex como sendo a linguagem descrita por ela:

$$\mathtt{0}(\mathtt{0} \cup \mathtt{1}) = \{\omega \in \{0,1\}^* : \omega = 0\beta \text{ e } \beta \in \{0,1\}^*\}$$

Onde o correto seria:

$$L(\mathtt{0}(\mathtt{0} \cup \mathtt{1})) = \{\omega \in \{0,1\}^* : \omega = 0\beta \text{ e } \beta \in \{0,1\}^*\}$$

Açúcar Sintático

Definição

Adendo à definição de regex

- $\gamma = \alpha^+$, onde $L(\gamma) = L(\alpha)L(\alpha)^*$
- $\gamma = \mathbf{\Sigma}$, onde $L(\gamma) = \Sigma$

Exercícios 🏋

Exercícios

Qual a linguagem descrita pelas seguintes regex?

1. $\mathtt{0}^*\mathtt{10}^*$
2. $\mathbf{\Sigma}^*\mathtt{001}\mathbf{\Sigma}^*$
3. $\mathtt{1}^*(\mathtt{01})^*$
4. $\mathtt{01} \cup \mathtt{10}$
5. $\mathtt{0}\mathbf{\Sigma}^*\mathtt{0} \cup \mathtt{1}\mathbf{\Sigma}^*\mathtt{1} \cup \mathtt{0} \cup \mathtt{1}$

Exercícios

Qual a linguagem reconhecida pelas seguintes regex?

1. $\mathtt{0}^*\mathtt{10}^* = \{\omega : \omega = \alpha 1 \beta \text{ e } \alpha, \beta \in \{0\}^*\}$
2. $\Sigma^*\mathtt{001}\Sigma^* = \{\omega : \omega \text{ contém } 001 \text{ como subcadeia }\}$
3. $\mathtt{1}^*(\mathtt{01})^* = \{\omega : \text{ todo } 0 \text{ em } \omega \text{ é seguido de ao menos um } 1\}$
4. $\mathtt{01} \cup \mathtt{10} = \{01, 10\}$
5. $\mathtt{0}\Sigma^*\mathtt{0} \cup \mathtt{1}\Sigma^*\mathtt{1} \cup \mathtt{0} \cup \mathtt{1} = \{\omega : \omega \text{ começa e termina com o mesmo símbolo }\}$

Exercícios

Escreva uma regex que descreva as seguintes linguagens

1. $\{\omega \in \Sigma : \omega$ tem comprimento par $\}$
2. $\{\varepsilon, 0, 1, 01\}$
3. Constante numérica de ponto flutuante válida na linguagem C
4. Cadeias que alternam entre 0 e 1
5. Cadeias onde toda posição ímpar contém o número 1
6. Cadeias com pelo menos um a e um b
7. Cadeias com no máximo um par de 11's consecutivos
8. Cadeias com símbolos 0 e 1 cujo um número de zeros é divisível por 3
9. Cadeias que não contém 110 como subcadeia

Exercícios

Escreva uma regex que descreva as seguintes linguagens

1. $\{\omega \in \Sigma : \omega$ tem comprimento par $\}$

$$(\mathbf{\Sigma\Sigma})^*$$

2. $\{\varepsilon, 0, 1, 01\}$

$$(\mathtt{0} \cup \boldsymbol{\varepsilon})(\mathtt{1} \cup \boldsymbol{\varepsilon})$$

3. Número de ponto flutuante com na linguagem C

$(+ \cup - \cup \boldsymbol{\varepsilon})(D^+ \cup D^+.D^* \cup D^*.D^+)$, onde $D = (0 \cup 1 \cup \cdots 9)$

4. Cadeias que alternam entre 0 e 1

$$(\mathtt{01})^* \cup (\mathtt{10})^* \cup \mathtt{0}(\mathtt{10})^* \cup \mathtt{1}(\mathtt{01})^* \equiv (\boldsymbol{\varepsilon} \cup \mathtt{1})(\mathtt{01})^*(\boldsymbol{\varepsilon} \cup \mathtt{0})$$

5. Cadeias onde toda posição ímpar contém o número 1

$$1(\mathbf{\Sigma}1)^* \cup \boldsymbol{\varepsilon}$$

Exercícios (cont.)

6. Cadeias com pelo menos um a e um b

$$\Sigma^*\mathtt{a}\Sigma^*\mathtt{b}\Sigma^* \cup \Sigma^*\mathtt{b}\Sigma^*\mathtt{a}\Sigma^*$$

7. Cadeias com no máximo um par de 11's consecutivos

$$(\mathtt{0} \cup \mathtt{10})^*(\mathtt{11} \cup \varepsilon)(\mathtt{0}^+\mathtt{1})^*\mathtt{0}^*$$

8. Cadeias com símbolos 0 e 1 cujo um número de zeros é divisível por 3

$$(\mathtt{1}^*\mathtt{0}\mathtt{1}^*\mathtt{0}\mathtt{1}^*\mathtt{0}\mathtt{1}^*)^*$$

9. Cadeias que não contém 110 como subcadeia

$$(\mathtt{0} \cup \mathtt{10})^*(\varepsilon \cup \mathtt{1} \cup \mathtt{11}^+)$$

Propriedades Algébricas

Propriedades algébricas

Nos slides que seguem, sejam α, β e γ expressões regulares sob um alfabeto Σ.

Operador: $\cup$

A operação binária <u>união</u> $\cup\colon \Sigma^* \times \Sigma^* \to \Sigma^*$ é

- $\alpha \cup (\beta \cup \gamma) \equiv (\alpha \cup \beta) \cup \gamma$ — associativa
- $\alpha \cup \beta \equiv \beta \cup \alpha$ — comutativa
- $\alpha \cup \alpha \equiv \alpha$ — idempotente
- $\alpha \cup \emptyset \equiv \alpha$ — $\emptyset$ é a identidade para $\cup$

Operador: $\circ$

A operação binária concatenação $\circ : \Sigma^* \times \Sigma^* \to \Sigma^*$ é

- $\alpha(\beta\gamma) \equiv (\alpha\beta)\gamma$ — associativa
- $\boldsymbol{\varepsilon}\alpha \equiv \alpha\boldsymbol{\varepsilon} \equiv \alpha$ — $\boldsymbol{\varepsilon}$ é a identidade para $\circ$
- $\boldsymbol{\emptyset}\alpha \equiv \alpha\boldsymbol{\emptyset} \equiv \boldsymbol{\emptyset}$ — $\boldsymbol{\emptyset}$ é o aniquilador para $\circ$
- $\alpha(\beta \cup \gamma) \equiv \alpha\beta \cup \alpha\gamma$ — distributividade à esquerda
- $(\alpha \cup \beta)\gamma \equiv \alpha\gamma \cup \beta\gamma$ — distributividade à direita

Operador: $*$

A operação binária estrela $*\colon \Sigma^* \times \Sigma^* \to \Sigma^*$ é

- $\emptyset^* \equiv \varepsilon$
- $\varepsilon \cup \alpha\alpha^* \equiv \alpha^*$
- $\varepsilon \cup \alpha^*\alpha \equiv \alpha^*$

Linguagens descritas por expressões regulares

Teorema

A linguagem descrita por uma expressão regular é uma linguagem regular

Demonstração.

- Seja γ uma expressão regular e seja ℓ o número de operações regulares presentes em γ
- Vamos provar por indução em ℓ que $L(\gamma)$ é regular
- Base ($\ell = 0$):
  - Se $\gamma = \mathtt{a}$, então $L(\gamma) = \{a\}$
  - Se $\gamma = \boldsymbol{\varepsilon}$, então $L(\gamma) = \{\varepsilon\}$
  - Se $\gamma = \boldsymbol{\emptyset}$, então $L(\gamma) = \{\}$
  - Em todos os casos temos que $L(\gamma)$ é regular

Teorema

A linguagem descrita por uma expressão regular é uma linguagem regular

Demonstração (cont.).

- Passo ($\ell > 0$): γ tem ao menos uma operação regular.
- Se $\gamma = \alpha \diamond \beta$, onde α e β são expressões regulares e $\diamond$ é a operação de união ou concatenação.
  - Pela hipótese de indução, $L(\alpha)$ e $L(\beta)$ são linguagens regulares.
  - Pelos teoremas vistos na aula sobre as propriedade das linguagens regulares, sabemos que $L(\alpha) \diamond L(\beta)$ é regular.
- Se $\gamma = \alpha^*$, onde α é uma expressão regular
  - Pela hipótese de indução, $L(\alpha)$ é uma linguagem regulares.
  - Pelos teoremas vistos na aula, sabemos que $L(\alpha)^*$ é regular. □

Algoritmo para converter regex em AFN

- A demonstração anterior usa as propriedades de fechamento das linguagens regulares.
  - Tais resultados fornecem algoritmos para construir autômatos finitos.
  - Geralmente, provas por indução fornecem algoritmos

✎ Exemplo

$$(\mathsf{a} \cup \mathsf{b})^*\mathsf{aba}$$

Exercício

Escreva um programa que, dado uma expressão regular γ, construa um autômato finito que reconheça $L(\gamma)$

Hierarquia de linguagens

Seja

- L_{regex} o conjunto de todas as linguagens que podem ser descritas por uma expressão regular.
- L_{reg} o conjunto de todas as linguagens regulares.
- L_{∞} o conjunto de todas as linguagens.

Sabemos que

$$L_{\text{regex}} \subseteq L_{\text{reg}} \subsetneq L_{\infty}$$

Agora, vamos demonstrar que

$$L_{\text{reg}} \subseteq L_{\text{regex}} \implies L_{\text{reg}} = L_{\text{regex}}$$

Teorema

Uma linguagem regular pode ser descrita por uma expressão regular.

- Se L é uma linguagem regular, então existe um AFN N que reconhece L.
- Vamos projetar um algoritmo que converte N em um expressão regular γ_N tal que $L(\gamma_N) = L$.
  - Convertemos N em um autômato finito não determinístico generalizado N^* equivalente.
  - Convertemos N^* na expressão regular γ_N.