Teoria de Linguagem

Expressões Regulares

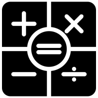
Vinicius H. S. Durelli
✉ durelli@ufsj.edu.br

UFSJ
UNIVERSIDADE FEDERAL
DE SÃO JOÃO DEL-REI

Organização

1. Contextualização
2. Definição
   - União
   - Concatenação
   - Concatenação sucessiva
3. Exemplos
4. Parênteses, precedência, operador extra e leis algébricas das ERs
5. Qual a classe de linguagens denotada por expressões regulares?
6. Considerações finais

Expressões regulares (ERs)...



Em aritmética, operações como $+$ e $\times$ podem ser utilizadas para construir expressões como $(5 + 3) \times 4$.

De forma similar, *expressões regulares* podem ser **usadas para criar expressões que descrevem linguagens**. Por exemplo, considere a expressão regular abaixo:

$$(0 \cup 1)0^*$$

Da mesma forma que o resultado da expressão aritmética acima é 32, o valor de uma expressão regular é uma linguagem (Sipser 2012).

Estamos falando sobre que tipo de formalismo?

ERs são um **formalismo denotacional**, também denominado formalismo funcional, pois:

- Definem um domínio que permite a caracterização do conjunto de palavras admissíveis em linguagens;
- Tratam-se de funções **composicionais** (horizontalmente).

Visto que a partir de uma ER é possível **inferir ("gerar")** as palavras da linguagem denotada, esse formalismo é algumas vezes referido como **formalismo gerador**.

Definição formal: expressões regulares

Definição → *Expressão Regular*

➤ Uma expressão regular (ER) sobre um alfabeto Σ é indutivamente definida como segue (Menezes 2011):

✢ **Base**:

- $\emptyset$ é uma ER e denota a linguagem vazia;
- ε é uma ER e denota a linguagem contendo somente a palavra vazia, ou seja, $\{\varepsilon\}$;
- Qualquer símbolo x pertencente a Σ é uma ER e denota a linguagem contendo uma única palavra, ou seja, $\{x\}$;

✢ **Passo de indução**:

- Se r e s são ERs e denotam as linguagens R e S, respectivamente, então:
  - $(r \cup s)$ é uma ER e denota a linguagem $R \cup S$;
  - $(r \cdot s)$ é uma ER e denota a linguagem $RS = \{uv | u \in R$ e $v \in S\}$;
  - (r^*) é uma ER e denota a linguagem R^*.

❑

União

Conforme mencionado, $(r \cup s)$ é uma ER e denota a linguagem $R \cup S$, ou seja, o conjunto de palavras que aparecem em R, S ou em ambos.

Exemplo: se $R = \{001, 10, 111\}$ e $S = \{\varepsilon, 001\}$, então

$$R \cup S = \{\varepsilon, 10, 001, 111\}$$

✍ Crie uma ER que gere a seguinte linguagem sobre $\Sigma = \{a, b\}$:

$$L = \{w \mid \text{w é } a,\ ab \text{ ou } aba\}$$

Conforme mencionado, $(r \cdot s)$ é uma ER e denota a linguagem $R \cdot S$, ou seja, o conjunto de palavras que podem ser formadas pela concatenação de qualquer palavra em R com qualquer palavra em S.[1]

Exemplo: se $R = \{001, 10, 111\}$ e $S = \{\varepsilon, 001\}$, então

$$R \cdot S = \{001, 10, 111, 001001, 10001, 111001\}$$

✍ Crie uma ER que gere a seguinte linguagem sobre $\Sigma = \{a, b, c\}$:

$L = \{w \mid$ w contém a, ab ou aba como prefixo e cc como sufixo$\}$

Omitindo o operador ·

Note que, por questões de simplicidade, o operador $\cdot$ pode ser omitido.

[1] Conceitualmente semelhante ao produto cartesiano.

Concatenação sucessiva (fechamento de Kleene)

Conforme mencionado, r^* é uma ER e denota a linguagem R^*, ou seja, o conjunto de palavras que podem ser formadas pela concatenação de todas as palavras em R. ➟ Mais formalmente, R^* é a união infinita $\cup_{i \geqslant 0} R^i$, onde $R^0 = \{\varepsilon\}$, $R^1 = R$ e R^i para $i \geqslant 1$ é $R \cdot R \cdot R \cdot R \ldots R$ (Hopcroft et al. 2006).

Exemplo: se $R = \{0, 1\}$ então

$$R^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, \ldots\}$$

✍ Crie uma ER que gere a seguinte linguagem sobre $\Sigma = \{a, b, c\}$:

$$L = \{w \mid \text{w contém } aa \text{ como subpalavra}\}$$

Exemplos de ER e suas respectivas linguagens

A tabela a seguir apresenta alguns exemplos de ERs.

ER	**Linguagem Representada**
aa	Somente a palavra aa
ab^*	Palavras que iniciam com a, seguido por zero ou mais bs
$(a \cup b)^*$	Todas as palavras sobre $\{a, b\}$
$a^*ba^*ba^*$	Todas as palavras contendo exatamente dois bs
$(a \cup b)^*(aa \cup bb)$	Todas as palavras que terminam com aa ou bb

Exercícios (1)

✍ Dado $\Sigma = \{a, b\}$, crie ERs que gerem as linguagens descritas a seguir.

Exercício ①: $L_{e1} = \{w \in \Sigma^* \;:\;$ w possui pelo menos um par de a consecutivos$\}$

Exercício ②: $L_{e2} = \{a^n b^m \;:\; n \geqslant 3$ e m é par$\}$

Exercício ③: Qual a linguagem denotada por $r = (00)^* \cdot (11)^* \cdot 1$?

Exercícios (1)

✍ Dado $\Sigma = \{a, b\}$, crie ERs que gerem as linguagens descritas a seguir.

Exercício ①: $L_{e1} = \{w \in \Sigma^* \ : \ $ w possui pelo menos um par de a consecutivos$\}$

$$(a \cup b)^* aa(a \cup b)^*$$

Exercício ②: $L_{e2} = \{a^n b^m \ : \ n \geqslant 3$ e m é par$\}$

Exercício ③: Qual a linguagem denotada por $r = (00)^* \cdot (11)^* \cdot 1$?

Exercícios (1)

✍ Dado $\Sigma = \{a, b\}$, crie ERs que gerem as linguagens descritas a seguir.

Exercício ①: $L_{e1} = \{w \in \Sigma^* : \text{w possui pelo menos um par de } a \text{ consecutivos}\}$

$$(a \cup b)^* aa(a \cup b)^*$$

Exercício ②: $L_{e2} = \{a^n b^m : n \geqslant 3 \text{ e } m \text{ é par}\}$

$$aaaa^*(bb)^*$$

Exercício ③: Qual a linguagem denotada por $r = (00)^* \cdot (11)^* \cdot 1$?

Exercícios (1)

✍ Dado $\Sigma = \{a, b\}$, crie ERs que gerem as linguagens descritas a seguir.

Exercício ①: $L_{e1} = \{w \in \Sigma^* \ : \ $ w possui pelo menos um par de a consecutivos$\}$

$$(a \cup b)^* aa (a \cup b)^*$$

Exercício ②: $L_{e2} = \{a^n b^m \ : \ n \geqslant 3 \text{ e } m \text{ é par}\}$

$$aaaa^*(bb)^*$$

Exercício ③: Qual a linguagem denotada por $r = (00)^* \cdot (11)^* \cdot 1$?

$$L = \{0^{2n}1^{2m+1} \mid n \geqslant 0, m \geqslant 0\} \text{ ou}$$

O conjunto de todas as palavras com um número par de 0 e ímpar de 1.

Exercícios (2)

✍ Dado $\Sigma = \{0, 1\}$, crie ERs que gerem as linguagens descritas a seguir.

Exercício ④: $L_{e4} = \{w \in \Sigma^* \ : \ \text{o número de 0 é divisível por 5}\}$

Exercício ⑤: $L_{e5} = \{w \ : \ \text{w contém pelo menos um } \mathit{0} \text{ e um } \mathit{1}\}$

Exercícios (2)

✍ Dado $\Sigma = \{0, 1\}$, crie ERs que gerem as linguagens descritas a seguir.

Exercício ④: $L_{e4} = \{w \in \Sigma^* \ : \ \text{o número de 0 é divisível por 5}\}$

$$(1^*01^*01^*01^*01^*01^*)^* \cup 1^*$$

Exercício ⑤: $L_{e5} = \{w \ : \ \text{w contém pelo menos um } \mathit{0} \text{ e um } \mathit{1}\}$

Exercícios (2)

✍ Dado $\Sigma = \{0, 1\}$, crie ERs que gerem as linguagens descritas a seguir.

Exercício ④: $L_{e4} = \{w \in \Sigma^* \;:\; \text{o número de 0 é divisível por 5}\}$

$$(1^*01^*01^*01^*01^*01^*)^* \cup 1^*$$

Exercício ⑤: $L_{e5} = \{w \;:\; \text{w contém pelo menos um } \mathit{0} \text{ e um } \mathit{1}\}$

$$00^*1 \cdot (0 \cup 1)^* \cup 11^*0 \cdot (0 \cup 1)^*$$

Parênteses, precedência e um operador extra...

Normalmente, parênteses são omitidos em ERs, respeitando as seguintes convenções: a concatenação sucessiva (i.e., $*$) tem precedência sobre a concatenação e a união, a concatenação tem precedência sobre a união.

Precedência mais alta	
Concatenação sucessiva	*
Concatenação	$\cdot$
União	$\cup$
Precedência mais baixa	

R^+

R^+ é usado como uma **abreviação** para $R \cdot R^*$. R^* inclui todas as palavras formadas por 0 ou mais concatenações de palavras in R, enquanto R^+ contém palavras formadas por uma ou mais concatenações de palavras em R, i.e., $R^+ \cup \varepsilon = R^*$.

Principais leis algébricas das ERs...

Sejam, r, s, e t três ERs quaisquer. Então:

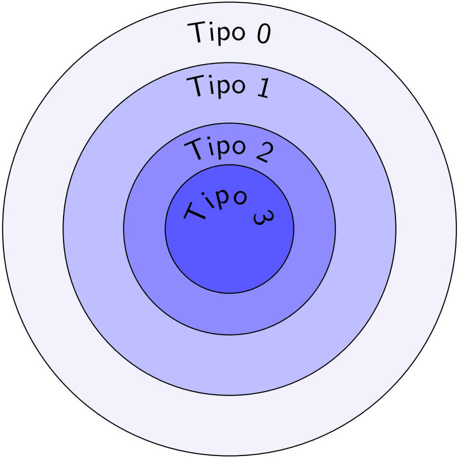
- Associatividade:
  - da união: $(r \cup s) \cup t = r \cup (s \cup t)$;
  - da concatenação: $(r \cdot s) \cdot t = r \cdot (s \cdot t)$;
- Comutativa:
  - da união: $r \cup s = s \cup r$;
- Elemento neutro:
  - da união: $r \cup \emptyset = \emptyset \cup r = r$;
  - da concatenação: $r \cdot \emptyset = \emptyset \cdot r = r$;
- Distribuição da concatenação sobre a união:
  - à esquerda: $r \cdot (s \cup t) = r \cdot s \cup r \cdot t$;
  - à direita: $(r \cup s) \cdot t = r \cdot t \cup s \cdot t$;

ERs denotam exatamente a classe das linguagens regulares.

Se r é uma ER, a linguagem denotada é dita a **linguagem gerada** por r, sendo representada por

$$L(r) \text{ ou } GERA(r)$$

- É possível denotar as mesmas linguagens reconhecidas por autômatos (Sipser 2012).



Teorema

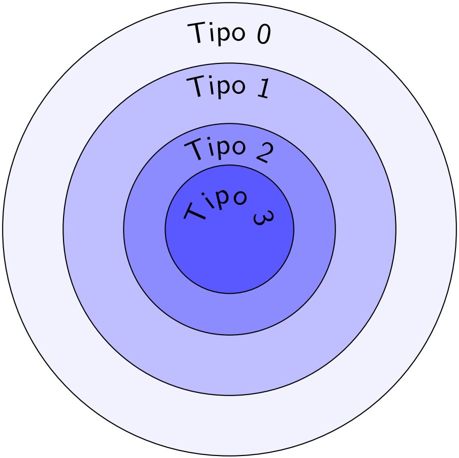
Se r é uma ER, então GERA(r) é uma linguagem regular.

ERs denotam exatamente a classe das linguagens regulares.

Se r é uma ER, a linguagem denotada é dita a **linguagem gerada** por r, sendo representada por

$$L(r) \text{ ou } GERA(r)$$

- É possível denotar as mesmas linguagens reconhecidas por autômatos (Sipser 2012).



Teorema

Se r é uma ER, então GERA(r) é uma linguagem regular.

Prova (1)

Se r é uma ER, então GERA(r) é uma linguagem regular.

Por definição, uma linguagem é regular se é possível construir um autômato que reconheça tal linguagem.

Ideia central

É necessário mostrar que dada uma ER r qualquer, é possível construir um autômato finito $\mathcal{M}$ tal que:

$$\text{ACEITA}(\mathcal{M}) = \text{GERA}(r)$$

Prova (2)

Se r é uma ER, então GERA(r) é uma linguagem regular.

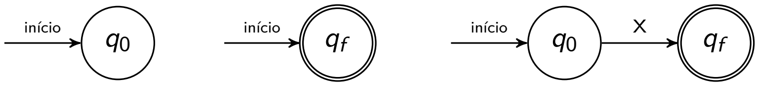
✢ **Base**: Seja r uma ER com **zero operadores**. Então r só pode ser da forma:

$$r = \emptyset$$
$$r = \varepsilon$$
$$r = x \ (\ x \in \Sigma \ )$$

Os autômatos finitos:

- $\mathcal{M}_1 = (\emptyset, \{q_0\}, \delta_1, q_0, \emptyset)$
- $\mathcal{M}_2 = (\emptyset, \{q_f\}, \delta_2, q_f, \{q_f\})$
- $\mathcal{M}_3 = (\{x\}, \{q_0, q_f\}, \delta_3, q_0, \{q_f\})$

ilustrados abaixo, aceitam as linguagens acima, respectivamente.



Prova (3)

- **Hipótese de indução**: Suponha que para algum $n \in \mathbb{N}$, e para qualquer $u \in \mathbb{N}$ tal que $u \leqslant n$, se o número de operadores de r é u, é possível definir um autômato finito que aceita a linguagem gerada por r.
- **Passo de indução**: Seja r uma ER com $n+1$ operadores. Então r pode ser representada por um dos seguintes casos, nos quais r_1 e r_2 individualmente, possuem no máximo n operadores e, conjuntamente, possuem $n+1$:

$$r = r_1 \cup r_2$$
$$r = r_1 \cdot r_2$$
$$r = r_1^*$$

Prova (4)

Portanto, por hipótese de indução é possível construir os autômatos:

- $\mathcal{M}_1 = (\Sigma_1, Q_1, \delta_1, q_{0_1}, \{q_{f_1}\})$ e
- $\mathcal{M}_2 = (\Sigma_2, Q_2, \delta_2, q_{0_2}, \{q_{f_2}\})$ j

tais que:

- ACEITA$(\mathcal{M}_1) =$ GERA(r_1) e
- ACEITA$(\mathcal{M}_2) =$ GERA(r_2)

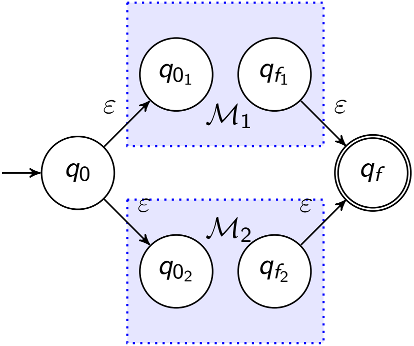
Nota-se que $\mathcal{M}_1$ e $\mathcal{M}_2$ possuem somente um estado final. Adicionalmente, suponha que os conjuntos de estados desses autômatos sejam disjuntos.

Prova (5)

Para $r = r_1 \cup r_2$ é possível criar o autômato abaixo (suponha $q_0 \notin Q_1 \cup Q_2$ e $q_f \notin Q_1 \cup Q_2$):

$$\mathcal{M} = (\Sigma_1 \cup \Sigma_2, Q_1 \cup Q_2 \cup \{q_0, q_f\}, \delta, q_0, \{q_f\})$$

$\mathcal{M}$ (ilustrado ao lado) é tal que a partir do estado inicial q_0, realiza transições vazias para os estados q_{0_1} e q_{0_2}. Assim, M_1 e M_2 processam de forma não-determinística e, portanto, é suficiente um dos "módulos" aceitar a entrada para o autômato M aceitar;

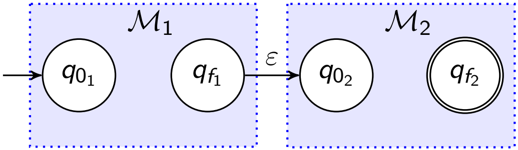


Prova (6)

Para $r = r_1 \cdot r_2$ é possível criar o autômato abaixo

$$\mathcal{M} = (\Sigma_1 \cup \Sigma_2, Q_1 \cup Q_2, \delta, q_{0_1}, \{q_{f_2}\})$$

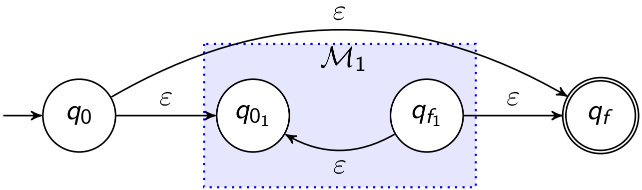
$\mathcal{M}$ (ilustrado abaixo), ao processar os módulos $\mathcal{M}_1$ e $\mathcal{M}_2$ em sequência, aceita a entrada se, e somente se, o prefixo pertencer a ACEITA($\mathcal{M}_1$) e o sufixo a ACEITA($\mathcal{M}_2$);



Para $r = r_1^*$ é possível criar o autômato abaixo

$$\mathcal{M} = (\Sigma_1, Q_1 \cup \{q_0, q_f\}, \delta, q_0, \{q_f\})$$

$\mathcal{M}$ (ilustrado abaixo), é tal que a transição vazia de q_0 para q_f garante a aceitação da palavra vazia e a transição de q_{f_1} para q_{0_1} permite o sucessivo processamento de $\mathcal{M}_1$ para assegurar o reconhecimento de duas ou mais concatenações sucessivas. ❑



Considerações finais. . .

Na aula de hoje nós vimos:

- Expressões regulares (ERs).
  - Precedência;
  - Leis algébricas.
- Equivalência entre as linguagens denotadas por ERs e autômatos finitos.

Na **próxima aula**: gramáticas livres de contexto.

Referências

Hopcroft, John E., Rajeev Motwani, & Jeffrey D. Ullman (2006). *Introduction to Automata Theory, Languages, and Computation*. 3rd ed. Pearson, p. 750.

Menezes, Paulo Blauth (2011). *Linguagens Formais e Autômatos*. 6th ed. Livros Didáticos Informática da UFRGS. Bookman, p. 256.

Sipser, Michael (2012). *Introduction to the Theory of Computation*. 3rd ed. Cengage Learning, p. 480.

☺**Próxima aula: exercício(s) sobre o conteúdo da aula de hoje!** ☺