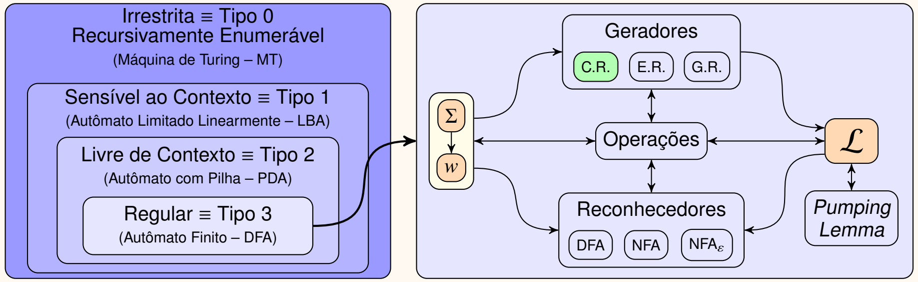
Linguagens Formais e Autômatos

Humberto Longo

Instituto de Informática
Universidade Federal de Goiás

Bacharelado em Ciência da Computação, 2021/1

Roteiro



Linguagens formais

- Dado um alfabeto Σ, uma linguagem em Σ é um conjunto de sequências de símbolos (palavras) do alfabeto.
- Se $\Sigma = \{a, b\}$, então são linguagens sobre Σ:
  - Finitas: o conjunto vazio e o conjunto formado pela palavra vazia. (Atenção: $\{\ \} \neq \{\varepsilon\} \neq \varepsilon$).
  - Finitas: $\{a, b, aa, ab, ba, bb\}, \{\varepsilon, aaa, bbb\}, \{aaa, aab, aba, abb\}$.
  - Infinitas: o conjunto $\{\varepsilon, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, \dots\}$ de palíndromos sobre Σ.
- Linguagem Σ^* : conjunto de todas as sequências de símbolos do alfabeto Σ.
  - $\varepsilon \in \Sigma^*$.
  - $\mathcal{L} \subseteq \Sigma^*$, se $\mathcal{L}$ é uma linguagem em Σ.

Linguagens formais

- Especificação de uma linguagem:
  - Descrição não ambígua das cadeias da linguagem.
- Linguagem finita:
  - Enumeração de suas cadeias.
- Linguagem infinita:
  - Definição recursiva das cadeias (para linguagens com estrutura sintática simples).
  - Construção a partir de conjuntos finitos através dos operadores de conjuntos.

Conjuntos definidos por indução

- Uma definição indutiva/recursiva de um conjunto C tem a seguinte forma:
  - Base: Especificação de um ou mais elementos "iniciais" de C.
  - Recursão: Uma ou mais regras para construção de "novos" elementos de C a partir de elementos "antigos" de C.
  - Fecho: O conjunto C consiste exatamente dos elementos que podem ser obtidos, começando-se com os elementos iniciais de C, aplicando-se as regras de recursão para a construção de novos elementos.
- **Obs.:** A condição de fechamento é frequentemente omitida, uma vez que é sempre assumida nas definições indutivas.

Definição recursiva de linguagens

Exemplo 1.8

- Seja Σ um alfabeto. A definição recursiva do conjunto Σ^*, das cadeias definidas sobre Σ, é:
  - Base: $\varepsilon \in \Sigma^*$.
  - Recursão: Se $w \in \Sigma^*$ e $a \in \Sigma$, então $wa \in \Sigma^*$.
  - Fecho: $w \in \Sigma^*$ se w pode ser obtida a partir de ε com um número finito de aplicações do passo recursivo.

Definição recursiva de linguagens

Exemplo 1.9

- Linguagem $\mathcal{L}$, sobre o alfabeto $\Sigma = \{a, b\}$, que contém cadeias de comprimento par e começam com a:
  - Base: $aa, ab \in \mathcal{L}$.
  - Recursão: Se $u \in \mathcal{L}$, então $uaa, uab, uba, ubb \in \mathcal{L}$.
  - Fecho: Uma cadeia $u \in \mathcal{L}$ se pode ser obtida a partir das cadeias básicas, com a aplicação um número finito de vezes da recursão.

Definição recursiva de linguagens

Exemplo 1.10

- Linguagem $\mathcal{L}$, sobre o alfabeto $\Sigma = \{a, b\}$, que contém cadeias em que cada ocorrência de um b é precedida de um a:
  - Base: $\varepsilon \in \mathcal{L}$.
  - Recursão: Se $u \in \mathcal{L}$, então $ua, uab \in \mathcal{L}$.
  - Fecho: Uma cadeia $u \in \mathcal{L}$ se pode ser obtida a partir das cadeias básicas, com a aplicação um número finito de vezes da recursão.

Especificação finita de linguagens

Definição 1.11

- A concatenação das linguagens $\mathcal{L}$ e $\mathcal{M}$ é:

$$\mathcal{L}\circ\mathcal{M} = \{xy \mid x \in \mathcal{L} \text{ e } y \in \mathcal{M}\}.$$

Definição 1.12 (Operações com Linguagens)

- Se $\mathcal{L}$ e $\mathcal{M}$ são linguagens no alfabeto Σ, então:

$$\begin{aligned}
\mathcal{L} \cup \mathcal{M} &= \{x \mid x \in \mathcal{L} \text{ ou } x \in \mathcal{M}\}.\\
\mathcal{L} \cap \mathcal{M} &= \{x \mid x \in \mathcal{L} \text{ e } x \in \mathcal{M}\}.\\
\mathcal{L} - \mathcal{M} &= \{x \mid x \in \mathcal{L} \text{ e } x \notin \mathcal{M}\}.\\
\overline{\mathcal{L}} &= \Sigma^* - \mathcal{L},\\
&= \{x \in \Sigma^* \mid x \notin \mathcal{L}\}.
\end{aligned}$$

Operações com linguagens

Exemplo 1.13

- Se $\mathcal{L} = \{a, bc, cb\}$, $\mathcal{M} = \{aa, bb, cc, bc, cb\}$ e $\Sigma = \{a, b, c\}$, então:

$$\begin{aligned}
\mathcal{L} \cup \mathcal{M} &= \{a, bc, cb, aa, bb, cc\}.\\
\mathcal{L} \cap \mathcal{M} &= \{bc, cb\}.\\
\mathcal{L} - \mathcal{M} &= \{a\}.\\
\mathcal{M} - \mathcal{L} &= \{aa, bb, cc\}.\\
\overline{\mathcal{L}} &= \{x \in \Sigma^* \mid x \neq a, x \neq bc, x \neq cb\}.\\
&= \{\varepsilon, b, aa, ab, ac, ba, bb, cb, cc, aaa, \dots\}.\\
\mathcal{L}\circ\mathcal{M} &= \{aaa, abb, acc, abc, acb, bcaa, bcbb, bccc,\\
&\qquad bcbc, bccb, cbaa, cbbb, cbcc, cbbc, cbcb\}.\\
\mathcal{M}\circ\mathcal{L} &= \dots
\end{aligned}$$

Operações com linguagens

Definição 1.14

$\mathcal{I} = \{\varepsilon\}$: elemento neutro na concatenação de linguagens:

- $\mathcal{L}\circ\mathcal{I} = \mathcal{I}\circ\mathcal{L} = \mathcal{L}$.

Definição 1.15

$\mathcal{L}^i$: potência da linguagem $\mathcal{L}$.

$$\begin{aligned}
\mathcal{L}^0 &= \{\varepsilon\}.\\
\mathcal{L}^1 &= \mathcal{L}.\\
\mathcal{L}^{i+1} &= \mathcal{L}\circ\mathcal{L}^i, && \text{para } i \in \mathbb{N}.\\
\mathcal{L}^i\circ\mathcal{L}^j &= \mathcal{L}^{i+j}, && \text{para } i, j \in \mathbb{N}.\\
\mathcal{L}^j\circ\mathcal{L}^i &= \mathcal{L}^{j+i}, && \text{para } i, j \in \mathbb{N}.
\end{aligned}$$

Operações com linguagens

Concatenação de potências de uma linguagem (Rafael Quirino – LFA 2017/1)

$$\begin{aligned}
\mathcal{L}^i\circ\mathcal{L}^j &= \mathcal{L}\circ\mathcal{L}^{i-1}\circ\mathcal{L}^j\\
&= \mathcal{L}\circ\mathcal{L}\circ\mathcal{L}^{i-2}\circ\mathcal{L}^j\\
&\ \ \vdots\\
&= \underbrace{\mathcal{L}\circ\mathcal{L}\circ\cdots\circ\mathcal{L}}_{i-2 \text{ cópias}}\circ\mathcal{L}^2\circ\mathcal{L}^j\\
&= \underbrace{\mathcal{L}\circ\mathcal{L}\circ\cdots\circ\mathcal{L}\circ\mathcal{L}}_{i-1 \text{ cópias}}\circ\mathcal{L}^1\circ\mathcal{L}^j\\
&= \underbrace{\mathcal{L}\circ\mathcal{L}\circ\cdots\circ\mathcal{L}\circ\mathcal{L}}_{i-1 \text{ cópias}}\circ\mathcal{L}^{1+j}\\
&= \underbrace{\mathcal{L}\circ\mathcal{L}\circ\cdots\circ\mathcal{L}}_{i-2 \text{ cópias}}\circ\mathcal{L}^{2+j}\\
&\ \ \vdots\\
&= \mathcal{L}\circ\mathcal{L}^{(i-1)+j}\\
&= \mathcal{L}^{i+j}.
\end{aligned}$$

$$\begin{aligned}
\mathcal{L}^j\circ\mathcal{L}^i &= \mathcal{L}\circ\mathcal{L}^{j-1}\circ\mathcal{L}^i\\
&= \mathcal{L}\circ\mathcal{L}\circ\mathcal{L}^{j-2}\circ\mathcal{L}^i\\
&\ \ \vdots\\
&= \underbrace{\mathcal{L}\circ\mathcal{L}\circ\cdots\circ\mathcal{L}}_{j-2 \text{ cópias}}\circ\mathcal{L}^2\circ\mathcal{L}^i\\
&= \underbrace{\mathcal{L}\circ\mathcal{L}\circ\cdots\circ\mathcal{L}\circ\mathcal{L}}_{j-1 \text{ cópias}}\circ\mathcal{L}^1\circ\mathcal{L}^i\\
&= \underbrace{\mathcal{L}\circ\mathcal{L}\circ\cdots\circ\mathcal{L}\circ\mathcal{L}}_{j-1 \text{ cópias}}\circ\mathcal{L}^{1+i}\\
&= \underbrace{\mathcal{L}\circ\mathcal{L}\circ\cdots\circ\mathcal{L}}_{j-2 \text{ cópias}}\circ\mathcal{L}^{2+i}\\
&\ \ \vdots\\
&= \mathcal{L}\circ\mathcal{L}^{(j-1)+i}\\
&= \mathcal{L}^{j+i}.
\end{aligned}$$

Operações com linguagens

Exemplo 1.16

- Se $\mathcal{L} = \{0, 11\}$ e $\Sigma = \{0, 1\}$, então:

$$\mathcal{L}^0 = \{\varepsilon\}.$$
$$\mathcal{L}^1 = \mathcal{L} \circ \mathcal{L}^0 = \{0, 11\} \circ \{\varepsilon\} = \{0, 11\}.$$
$$\mathcal{L}^2 = \mathcal{L} \circ \mathcal{L}^1 = \{0, 11\} \circ \{0, 11\} = \{00, 011, 110, 1111\}.$$
$$\mathcal{L}^3 = \mathcal{L} \circ \mathcal{L}^2 = \{0, 11\} \circ \{00, 011, 110, 1111\}.$$
$$\mathcal{L}^3 = \{000, 0011, 0110, 01111, 1100, 11011, 11110, 111111\}.$$

Operações com linguagens

Definição 1.17 (Fecho de Kleene)

- Fecho da linguagem $\mathcal{L}$:
  - $\mathcal{L}^* = \bigcup_{i=0}^{\infty} \mathcal{L}^i = \mathcal{L}^0 \cup \mathcal{L}^1 \cup \mathcal{L}^2 \cup \ldots$

Definição 1.18

- $\mathcal{L}^+ = \bigcup_{i=1}^{\infty} \mathcal{L}^i = \mathcal{L} \circ \mathcal{L}^* = \mathcal{L}^1 \cup \mathcal{L}^2 \cup \mathcal{L}^3 \cup \ldots$

Exemplo 1.19

- Se $\mathcal{L} = \{0, 11\}$ e $\Sigma = \{0, 1\}$, então:

$$\mathcal{L}^* = \{\varepsilon, 0, 11, 00, 011, 110, 1111, 000, 0011, 0110, \ldots\}.$$

Especificação finita de linguagens

- Especificação não ambígua das cadeias que pertencem à linguagem.
- Descrição informal não é rigorosa o suficiente para uma definição precisa.
- Ex: Linguagem $\mathcal{L}$ sobre o alfabeto $\Sigma = \{a, b\}$, tal que $\mathcal{L}$ contém cadeias com a subcadeia bb.
  - Uma cadeia $w \in \mathcal{L}$ pode conter mais de uma ocorrência da subcadeia bb?
- A precisão das operações em conjuntos pode ser usada para a descrição não ambígua de linguagens.
  - O resultado de uma operação unária em uma linguagem ou uma operação binária em duas linguagens define outra linguagem.

Especificação finita de linguagens

Exemplo 1.20

- $\Sigma = \{a, b\}$ e $\mathcal{L}$ é composta de todas as cadeias que contém a subcadeia bb:
  - $\mathcal{L} = \{a, b\}^* \circ \{bb\} \circ \{a, b\}^*$.
- A concatenação do conjunto $\{bb\}$ garante a presença de bb em toda cadeia de $\mathcal{L}$.
- Os conjuntos $\{a, b\}^*$ permitem qualquer número de a's e b's, em qualquer ordem, antes ou depois da cadeia bb.

Especificação finita de linguagens

Exemplo 1.21

- $\Sigma = \{a, b\}$ e $\mathcal{L}$ é composta de todas as cadeias que começam com aa ou terminam com bb:
  - $\{aa\}\circ\{a, b\}^* \rightarrow$ conjunto de cadeias com prefixo aa.
  - $\{a, b\}^*\circ\{bb\} \rightarrow$ conjunto de cadeias com sufixo bb.
- $\mathcal{L} = \{aa\}\circ\{a, b\}^* \cup \{a, b\}^*\circ\{bb\}$.

Exemplo 1.22

- Dadas linguagens $\mathcal{L}_1 = \{bb\}$ e $\mathcal{L}_2 = \{\varepsilon, bb, bbbb\}$ sobre o alfabeto $\Sigma = \{b\}$, então $\mathcal{L}_1^* = \mathcal{L}_2^*$ contém cadeias com número par de b's.

Especificação finita de linguagens

Exemplo 1.23

- $\mathcal{P}$ é o conjunto de cadeias de comprimento par definidas sobre o alfabeto $\Sigma = \{a, b\}$:
  - $\mathcal{P} = \{aa, bb, ab, ba\}^*$.
  - A repetição de concatenações constrói cadeias com o acréscimo de dois símbolos de cada vez.
- $\mathcal{I}$ é o conjunto de cadeias de comprimento ímpar definidas sobre o alfabeto $\Sigma = \{a, b\}$:
  - $\mathcal{I} = \{a, b\}^* - \{aa, bb, ab, ba\}^*$.
  - $\mathcal{I} = \{a, b\}\circ\{aa, bb, ab, ba\}^*$.

Conjuntos regulares

Definição 1.24

- Um conjunto regular sobre um alfabeto Σ é definido como:

Base: $\varnothing$, $\{\varepsilon\}$ e $\{a\}$, para todo $a \in \Sigma$, são conjuntos regulares sobre Σ.

Recursão: Se X e Y são conjuntos regulares sobre Σ, então $X \cup Y$, $X\circ Y$ e X^* também são conjuntos regulares sobre Σ.

Fecho: X é um conjunto regular sobre Σ se pode ser obtido, a partir dos conjuntos regulares básicos, com a aplicação da recursão um número finito de vezes.

Conjuntos regulares

Exemplo 1.25

- A linguagem $\mathcal{L} = \{a, b\}^*\circ\{bb\}\circ\{a, b\}^*$ é um conjunto regular sobre $\Sigma = \{a, b\}$:
  - $\{a\}$ e $\{b\}$ são conjuntos regulares (base da definição).
  - $\{a, b\} = \{a\} \cup \{b\}$ é regular (união).
  - $\{a, b\}^*$ é regular (fecho de Kleene).
  - $\{bb\} = \{b\}\circ\{b\}$ é regular (concatenação).

Exemplo 1.26

- O conjunto de cadeias, sobre o alfabeto $\{a, b\}$, que terminam com a e contêm pelo menos um b é regular.
  - $\{a, b\}^*\circ\{b\}\circ\{a, b\}^*\circ\{a\}$

Operações com conjuntos regulares

Lema 1.27

Se $\mathcal{L}$ é conjunto regular, então $\overline{\mathcal{L}}$ também é regular

Demonstração.

No decorrer do curso . . . □

Lema 1.28

Se $\mathcal{L}_1$ e $\mathcal{L}_2$ são conjuntos regulares, então $\mathcal{L}_1 - \mathcal{L}_2$ também é regular

Demonstração.

$$\mathcal{L}_1 - \mathcal{L}_2 = \mathcal{L}_1 \cap \overline{\mathcal{L}_2} = \overline{\overline{\mathcal{L}_1 \cap \overline{\mathcal{L}_2}}} = \overline{\overline{\mathcal{L}_1} \cup \overline{\overline{\mathcal{L}_2}}} = \overline{\overline{\mathcal{L}_1} \cup \mathcal{L}_2}$$

□

Operações com conjuntos regulares

Lema 1.29

Se $\mathcal{L}_1$ e $\mathcal{L}_2$ são conjuntos regulares, então $\mathcal{L}_1 \oplus \mathcal{L}_2$ também é regular

Demonstração.

$$\begin{aligned}
\mathcal{L}_1 \oplus \mathcal{L}_2 &= (\mathcal{L}_1 \cup \mathcal{L}_2) - (\mathcal{L}_1 \cap \mathcal{L}_2) \\
&= (\mathcal{L}_1 \cup \mathcal{L}_2) \cap \overline{(\mathcal{L}_1 \cap \mathcal{L}_2)} \\
&= (\mathcal{L}_1 \cup \mathcal{L}_2) \cap (\overline{\mathcal{L}_1} \cup \overline{\mathcal{L}_2}) \\
&= \overline{\overline{(\mathcal{L}_1 \cup \mathcal{L}_2) \cap (\overline{\mathcal{L}_1} \cup \overline{\mathcal{L}_2})}} \\
&= \overline{\overline{(\mathcal{L}_1 \cup \mathcal{L}_2)} \cup \overline{(\overline{\mathcal{L}_1} \cup \overline{\mathcal{L}_2})}}
\end{aligned}$$

□

Livros texto

R. P. Grimaldi
Discrete and Combinatorial Mathematics – An Applied Introduction.
Addison Wesley, 1994.

D. J. Velleman
How To Prove It – A Structured Approach.
Cambridge University Press, 1996.

J. E. Hopcroft; J. Ullman.
Introdução Ã Teoria de Autômatos, Linguagens e Computação.
Ed. Campus.

T. A. Sudkamp.
Languages and Machines – An Introduction to the Theory of Computer Science.
Addison Wesley Longman, Inc. 1998.

J. Carroll; D. Long.
Theory of Finite Automata – With an Introduction to Formal Languages.
Prentice-Hall, 1989.

M. Sipser.
Introduction to the Theory of Computation.
PWS Publishing Company, 1997.

H. R. Lewis; C. H. Papadimitriou
Elementos de Teoria da Computação.
Bookman, 2000.