Introdução

Linguagens Formais e Autômatos

Prof. Daniel Saad Nogueira Nunes

IFB – Instituto Federal de Brasília, Campus Taguatinga

Sumário

Introdução

Por que estudar linguagens formais e autômatos?

Autômatos finitos

Autômatos finitos são úteis para uma série de aplicações importantes.

- Softwares de projeto e verificação de circuitos digitais.
- Análise léxica de compiladores: quebra a entrada em unidades lógicas, como identificadores, palavras chaves e pontuação.
- Softwares para inspecionar textos e encontrar ocorrência de palavras, frases ou outros tipos de padrões.
- Softwares para verificação de sistemas que possuem um número finito e distinto de estados, como aqueles que utilizam protocolos de comunicação ou de troca segura de informação.

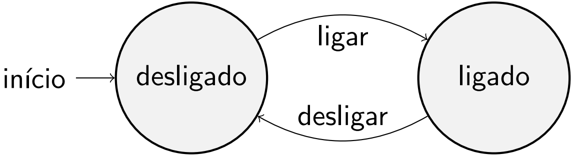
Autômatos finitos

Exemplo: interruptor

Modelaremos o comportamento de um interruptor de liga/desliga através de um autômato finito.

- O dispositivo possui dois estados, **ligado** e **desligado**.
- Quando o dispositivo está ligado e o usuário pressiona o interruptor, ele passa ao estado desligado.
- Quando o dispositivo está desligado e o usuário pressiona o interruptor, ele passa ao estado ligado.

Exemplo



Autômatos finitos: exemplo

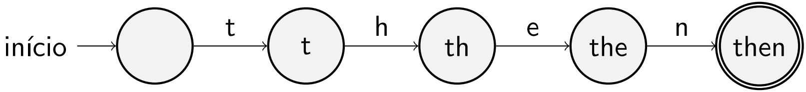
- No exemplo anterior, um dos estados foi designado como estado inicial,*i.e.*, o estado em que o sistema se encontra inicialmente.
- Muitas vezes, é necessário indicar um estado como *final* ou de *aceitação*.
- Ao entrar em um estado de *aceitação*, indica-se que a entrada, de alguma forma, atende alguma especificação

Autômatos finitos: exemplo

Exemplo: analisador léxico

Um analisador léxico é um componente de um compilador responsável por reconhecer padrões, como, por exemplo, palavras-chave.
O próximo autômato reconhecerá a palavra-chave *then*.

Autômatos finitos: analisador léxico



Exemplo: analisador léxico

- No exemplo anterior, temos 5 estados, cada um, representado um prefixo diferente da palavra *then*.
- O estado inicial representa a palavra vazia e cada estado possui uma transição com a próxima letra da palavra-chave para o próximo.
- Para uma entrada qualquer, chega-se no estado final somente quando essa palavra corresponde à palavra-chave.
- A tarefa desse autômato foi **reconhecer** a palavra-chave *then*.

Representações estruturais

- Outras notações, que não consistem de máquinas de estado, são muito importantes no estudo dos autômatos e suas aplicações, como por exemplo as **gramáticas** e as **expressões regulares**.

Gramáticas

- Gramáticas são extremamente úteis para processar dados que possuem estruturas recursivas.
- Um exemplo prático da aplicação de gramáticas é um *parser*, componente de um compilador que lida com a estrutura sintática de um programa.
- Um programa pode ter várias estruturas aninhadas, como laços de repetição, expressões aritméticas e condicionais.
- Gramáticas são extremamente úteis para capturar esse tipo de aninhamento.

Gramáticas: exemplo

Exemplo

Exemplo de gramática que descreve expressões aritméticas.

$$
\begin{aligned}
\mathrm{E} &\to \mathrm{E} + \mathrm{E} \\
\mathrm{E} &\to \mathrm{E} - \mathrm{E} \\
\mathrm{E} &\to \mathrm{E} \cdot \mathrm{E} \\
\mathrm{E} &\to \mathrm{E}/\mathrm{E} \\
\mathrm{E} &\to (\mathrm{E}) \\
\mathrm{E} &\to \langle \mathrm{num} \rangle
\end{aligned}
$$

Expressões regulares

- Expressões regulares, são extremamente úteis para descrever padrões de palavras.
- A expressão regular `[A-Z][a-z]*[ ][A-Z][A-Z]` (padrão UNIX), descreve o padrão do nome de uma cidade, seguida de espaço e duas letras, que podem representar o estado.
- Exemplo: `Brasilia DF`

Autômatos e complexidade

- Além de todas essas aplicações, autômatos são essenciais para o estudo dos limites da computação.
- Quais problemas são possíveis de resolver utilizando determinado modelo? E quais não são possíveis?

Sumário

Conceitos preliminares

- Estudaremos conceitos fundamentais para compreender a teoria dos autômatos.

Alfabeto

Definição (Alfabeto)

Um **alfabeto** é um conjunto finito e não vazio de símbolos.

- Normalmente usamos o símbolo Σ para denotar um alfabeto.

Alfabeto

Exemplos de alfabetos

- $\Sigma = \{0, 1\}$, o alfabeto binário.
- $\Sigma = \{a, b, \ldots, z\}$, o alfabeto das letras minúsculas.
- O conjunto de todos os caracteres ASCII também é um alfabeto.

Strings

Definição (String)

Uma string é uma **sequência finita** de símbolos escolhidas de algum alfabeto.

- Por exemplo, a string 01101 é formada por símbolos do alfabeto $\Sigma = \{0, 1\}$.

Strings

Definição (String vazia)

A string vazia, denotada por ϵ, é uma string que possui 0 símbolos.

Strings

Definição (Tamanho de uma string)

O tamanho de uma string w, denotado por $|w|$, representa a quantidade de símbolos presentes em w.

- $w = 01010$, $|w| = 5$
- $|\epsilon| = 0$

Potências de um alfabeto

Definição (Σ^k)

Se Σ é um alfabeto, definimos Σ^k como o conjunto de strings de tamanho k que podem ser formadas usando os símbolos de Σ. Em especial, $\Sigma^0 = \{\epsilon\}$.

- Se $\Sigma = \{0, 1\}$, então $\Sigma^0 = \{\epsilon\}$, $\Sigma^1 = \{0, 1\}$, $\Sigma^2 = \{00, 01, 10, 11\}$, $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\} \ldots$
- Não confundir Σ com Σ^1. Enquanto um é um conjunto de símbolos, o outro é um conjunto de palavras.

Potências de um alfabeto

Definição (Σ^*)

Para um alfabeto Σ, Σ^* é definido como o conjunto de todas as palavras que podem ser formadas com símbolos desse alfabeto. Formalmente:

$$\Sigma^* = \bigcup_{k \geq 0} \Sigma^k$$

- Para $\Sigma = \{0, 1\}$, $\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$

Potências de um alfabeto

Definição (Σ^+)

Denotamos por Σ^+ o conjunto de todas as palavras sobre o alfabeto Σ, com exceção da palavra vazia. Formalmente temos:

$$\Sigma^+ = \bigcup_{k>0} \Sigma^k$$

- Para $\Sigma = \{0, 1\}$, $\Sigma^+ = \Sigma^* \backslash \{\epsilon\}$, portanto, $\Sigma^+ = \{0, 1, 00, 01, 10, 11, 000, \ldots\}$.

Concatenação de palavras

Definição (Concatenação de palavras)

Sejam $x = x_1x_2 \ldots x_n$ e $y = y_1y_2 \ldots y_m$ palavras.
$xy = x_0x_1 \ldots x_ny_1y_2 \ldots y_m$ denota a concatenação de x com y.

- Em especial $\epsilon x = x\epsilon = x$, para qualquer palavra x.

Linguagens

Definição (Linguagem)

Uma linguagem é um conjunto de palavras escolhida a partir de Σ^*. Se Σ é um alfabeto, e $L \subseteq \Sigma^*$, então L é uma linguagem sobre o alfabeto Σ.

Linguagens

Exemplos

- $L = \{0^n1^n | n \geq 0\}$: linguagem que contém todas as palavras que começam com n 0s e seguidos por n 1s.
- $L = \{w | w \in \Sigma^*$ e w possui o mesmo número de zeros e uns$\}$
- $L = \{w | w$ é um programa sintaticamente correto na linguagem C$\}$
- $L = \emptyset$ é a linguagem vazia.

problemas

Definição (Problema)

Problemas no contexto de teoria de autômatos é a questão de responder se uma palavra pertence ou não à uma linguagem. Mais precisamente, se Σ é um alfabeto e L uma linguagem sobre Σ, então o problema L é:

Dada uma palavra w, determinar se ela está ou não em L.