

# Máquinas de estados finitos

Tendo visto a importância das linguagens formais para o estudo de problemas computáveis, passaremos ao estudo de máquinas (modelos) que são capazes de processar essas linguagens. Chamamos tais máquinas de **reconhecedores** de linguagens de uma forma geral.

Iniciaremos nossos estudos a partir de modelos bastante simples, mas, que ainda assim, são de grande importância não só teórica quanto prática, os chamados **autômatos finitos** (AF), ou **máquinas de estados finitos**.

Essas máquinas possuem memória limitada, mas podem ser usadas em diversas tarefas como na construção de compiladores, em algoritmos para manipulação de sequências, e até mesmo em máquinas físicas mais simples como as máquinas de venda de alimentos.

☆ O conjunto de linguagens que tais máquinas reconhecem é chamado de **Linguagens Regulares**. Do ponto de vista teórico, além dos AFs, existem outros formalismos para expressar linguagens dessa classe. Estudaremos alguns deles durante as próximas aulas.

Para entender melhor o funcionamento dessas máquinas, começaremos por um exemplo antes de apresentarmos os formalismos correspondentes.

Considere a seguinte situação: Queremos construir uma máquina (modelo) para controlar a abertura automática de uma porta. A porta conta com dois tapetes que possuem sensores para detectar pessoas do lado de dentro e fora do recinto. A porta possui duas bandas que abrem para dentro do recinto. Ao detectar pessoas em um dos lados, a máquina deve agir conforme estado atual da porta. Por exemplo, se estiver fechada e uma pessoa se aproximar do lado de fora, então a máquina deve abrir a porta. De forma análoga, se a porta estiver aberta e não forem detectadas pessoas do lado de dentro ou fora, então a máquina deve fechar a porta. Porém, se houver pessoas em qualquer dos lados, ela deve manter a porta aberta. Além disso, ao abrir a porta, devemos cuidar para que uma pessoa do lado de dentro não seja atingida por ela. Assim, se a porta estiver fechada e uma pessoa do lado de fora se aproximar, tendo outra do lado de dentro, então a porta não deve ser aberta.

☆ Note que nessa situação precisamos essencialmente armazenar o **estado** em que a porta se encontra para atuarmos de acordo. A memória da nossa máquina deve então armazenar esses estados. Ou seja, em uma máquina de estados finitos, os estados são situações que necessitamos armazenar; eles são a memória da nossa máquina. Sendo assim, os AFs são máquinas de memória finita (limitada). Especificamente, nossa máquina possui o conjunto de estados  $Q = \{Aberto, Fechado\}$ , representando os possíveis estados da porta.

As ações que essa máquina deve executar dependem da atuação dos sensores; isto é, ela deve reagir a pessoas se aproximando/afastando da porta e alterar seu estado

conforme a descrição acima. Em outras palavras, as ações dependem de 4 situações:

- Uma pessoa se encontra do lado de fora (F)
- Uma pessoa se encontra do lado de dentro (D)
- Há pessoas em ambos os lados (A)
- Não há pessoas em qualquer dos lados (N)

☆ Do ponto de vista formal, o alfabeto da linguagem para nos comunicarmos com essa máquina envolve 4 símbolos representando as possíveis ações  $\Sigma = \{F, D, A, N\}$ . Tais símbolos são como instruções para a máquina, que deve reagir e alterar seu estado conforme planejado.

☆ As alterações estado são chamadas de **transições**. Podemos representar as transições da máquina através de uma função. Tal função possui dois argumentos: um símbolo do alfabeto; e um estado da máquina.

	F	D	A	N
aberto				
fechado				

Uma **palavra** construída com o alfabeto da máquina indica uma sequência de transições que ela fará para efetuar a computação. No nosso exemplo, se iniciarmos no estado fechado, a palavra FDNDAN indica a sequência de ações: fechado->aberto->aberto->fechado->aberto->aberto->fechado. Cada transição consome um símbolo da palavra da esquerda para a direita, modificando o estado atual da máquina. Assim, a computação da máquina depende de duas informações para seguir:

- O estado atual; e
- O restante da palavra a processar y.

Chamamos esse par de **configuração instantânea** da máquina. As configurações instantâneas nos permitem acompanhar a computação de uma palavra. Para isso, usamos a existência de uma relação "resulta em", denotada por  $\vdash$ , indicando que uma configuração resulta em outra se há uma transição daquele estado para o outro com o primeiro símbolo da palavra (mais à esquerda). Por exemplo:

- $[fechado, FDNDAN] \vdash [aberto, DNDAN] \vdash [aberto, NDAN] \dots \vdash [fechado, \lambda]$

Agora que entendemos o funcionamento básico das máquinas de estados finitos,

podemos formalizar as definições.