Autômatos finitos determinísticos

Um autômato finito determinístico (AFD) é uma quíntupla $(Q, \quad \Sigma, \quad \delta, \quad i, \quad F)$ em que:

- Q é um conjunto finito de **estados**;
- Σ é o **alfabeto** da máquina;
- $\delta: Q \times \Sigma \rightarrow Q$ é uma **função de transição**, uma função total;
- i é o **estado inicial** da máquina $i \in Q$;
- F é um **conjunto de estados finais** (de aceitação) da máquina, $F \subseteq Q$.

Como dito anteriormente, os **estados** fazem o papel de memória de um AF. Embora a única restrição seja de que esse conjunto seja finito, normalmente são usados mnemônicos na identificação desses estados. Isso com o simples intuito de facilitar a leitura do diagrama, de maneira similar ao que normalmente se faz em programação.

O **alfabeto** da máquina define quais símbolos podem ser reconhecidos por ela. Ele define o alfabeto da linguagem reconhecida pelo autômato.

A **função de transição** define as ações da máquina. No nosso exemplo, ela definia a mudança de estado da porta entre aberto e fechado, dependendo do seu estado atual e de um símbolo do alfabeto (uma pessoa posicionada em um dos lados). O resultado dessa função é um novo estado que a máquina se encontrará após executar a ação. <u>Note que ela deve ser uma função total. Portanto, para todo par (estado, símbolo), deve haver um estado no qual a máquina entrará</u>. Isso confere o caráter determinístico dessa máquina; dada uma configuração instantânea e a função de transição, sabemos exatamente o próximo estado em que a máquina se encontrará.

O **estado inicial**, como o próprio nome diz, marca o estado em que a máquina inicia a computação. As computações serão sempre iniciadas nesse estado. O fato de sabermos exatamente em qual estado a máquina inicia sua computação também é uma característica de seu determinismo. Como a máquina inicia sempre em um mesmo estado e possui uma única transição possível para cada estado/símbolo, sabemos exatamente a sequência de computações que ela fará, isto é, a sequência de configurações instantâneas.

O **conjunto de estados finais** marca os estados de aceitação da máquina. Isto é, caso a computação se encerre em um desses estados, dizemos que a máquina aceita/reconhece a palavra como pertencente à linguagem para a qual foi desenhada. Discutiremos a definição formal mais à frente, mas, intuitivamente, uma palavra é reconhecida pela máquina se essa terminar sua computação (consumir todos os símbolos da palavra) em um estado final.

O diagrama representando o AF é chamado de **diagrama de estados**. No diagrama de estados, as transições são representadas por arestas direcionadas entre estados (círculos). O estado inicial é marcado por uma aresta sem origem, enquanto os finais marcados por duplo círculos. Em alguns casos, alguns estados e transições são omitidos do diagrama somente para clareza. Esses estados e transições são relacionadas a erros de computação, então, como a função de transição é total, caso a transição sob algum símbolo não esteja visível no diagrama, deve-se entender que aquela é uma transição para um estado de erro que foi omitido do diagrama. Veremos alguns exemplos mais a frente, onde reforçaremos esse conceito.

Exemplo: Considere o seguinte diagrama de estados de um AFD:

Esse autômato é definido formalmente por $M = (Q, \Sigma, \delta, i, F)$:

- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $\delta = \{(q_1, 0, q_1), (q_1, 1, q_2), (q_2, 0, q_3), (q_2, 1, q_2), (q_3, 0, q_2), (q_3, 1, q_2)\}$
- $i = q_1$
- $F = \{q_2\}$

Intuitivamente, qual a linguagem reconhecida por esse autômato?

Percebe-se que, para ser aceita, uma palavra precisa ter ao menos um 1. Uma vez que a máquina alcance o estado q_2, ela se mantém nele sempre que vem um 1, e, caso saia dele, só retorna com um número par de 0. Assim, percebemos que

$$L(M) = \{w \in \{0,1\}^* \mid o\ \text{último}\ 1\ em\ w\ \text{é}\ seguido\ de\ um\ \text{número}\ par\ de\ 0's\}$$

Veja que:

- $w = 0010$ não é aceita já que $[q_1, 0010] \vdash [q_1, 010] \vdash [q_1, 10] \vdash [q_2, 0] \vdash [q_3, \lambda]$
  Ou seja, ela termina sua computação em um estado não final

- $w = 100$ é aceita já que $[q_1, 100] \vdash [q_2, 00] \vdash [q_3, 0] \vdash [q_2, \lambda]$, ou seja ela termina sua computação em estado final

Para definir formalmente a linguagem reconhecida por um AFD, vamos definir o conceito de função de transição estendida. A **função de transição estendida** $\hat{\delta}: Q \times \Sigma^* \to Q$ determina o estado em que o AFD se encontra ao processar uma palavra qualquer a partir de um estado. Formalmente:

- $\hat{\delta}(e_1, \lambda) = e_1$;
- $\hat{\delta}(e_1, ay) = \hat{\delta}(\delta(e_1, a), y)$, para todo $a \in \Sigma$ e $y \in \Sigma^*$

Exemplo: Considerando o autômato acima, temos:

- $\hat{\delta}(q_1, 0010) =$