

Linguagens Formais e Autômatos

Autômatos Finitos Determinísticos (AFD)

Cristiano Lehrer, M.Sc.

Linguagens Regulares

- A teoria da computação começa com uma pergunta:
 - O que é um computador?
 - É, talvez, uma questão tola, pois todos sabem que essa coisa sobre a qual estou teclando é um computador.
 - Porém, esses computadores reais são bastante complicados – demasiadamente complicados para nos permitir construir uma teoria matemática manejável sobre eles diretamente.
 - Em vez disso, usamos um computador idealizado chamado **modelo computacional**.
 - Como qualquer modelo em ciência, um modelo computacional pode ser preciso em alguns detalhes, mas talvez não em outros.

Introdução (1/2)

- Um Autômato Finito Determinístico (AFD) ou máquina de estados finitos é um modelo computacional significativamente restrito de um computador real:
 - O AFD compartilha com um computador real o fato de que tem uma unidade central de processamento de capacidade fixa e finita.
 - Recebe sua entrada como um texto, entregue em uma fita de entrada.
 - Não pode produzir nenhuma saída, exceto uma indicação informando se a entrada foi ou não considerada aceitável.
 - É, em outras palavras, um dispositivo de reconhecimento de linguagem.
 - O que torna o AFD um modelo tão restrito dos computadores reais é a completa falta de memória fora de seu processador central fixo.

Introdução (2/2)

- Uma razão para estudar AFD é que eles permitem projetar vários tipos comuns de computadores, algoritmos e programas:
 - Por exemplo, a fase de análise léxica de um compilador (no qual unidades de programa como `begin` e `+` são identificados) é frequentemente baseada na simulação de um AFD.
 - Além disso, o problema de encontrar uma ocorrência de um texto dentro de outro – por exemplo, se quisermos verificar se qualquer um dos textos *ar*, *água*, *terra* e fogo ocorrem no texto **Elementos de Teoria da Computação** – também pode ser resolvida eficientemente por métodos originados na teoria dos AFD.

Definição (1/5)

- Um Autômato Finito Determinístico ou simplesmente Autômato Finito pode ser visto como uma máquina composta, basicamente por:
 - **Fita** → dispositivo de entrada que contém a informação a ser processada.
 - **Unidade de controle** → reflete o estado corrente da máquina. Possui uma unidade de leitura (cabeça da fita) a qual acessa uma célula da fita de cada vez e movimenta-se exclusivamente para a direita.
 - **Programa ou Função de Transição** → função que comanda as leituras e define o estado da máquina.

Definição (2/5)

Autômato Finito como uma máquina com controle finito.

Fita →



Definição (3/5)

- A fita é finita, sendo dividida em células, onde cada uma armazena um símbolo:
 - Os símbolos pertencem a um alfabeto de entrada.
 - Não é possível gravar sobre a fita (e não existe memória auxiliar).
 - Inicialmente, a palavra a ser processada (a informação de entrada para a máquina) ocupa toda a fita.

Definição (4/5)

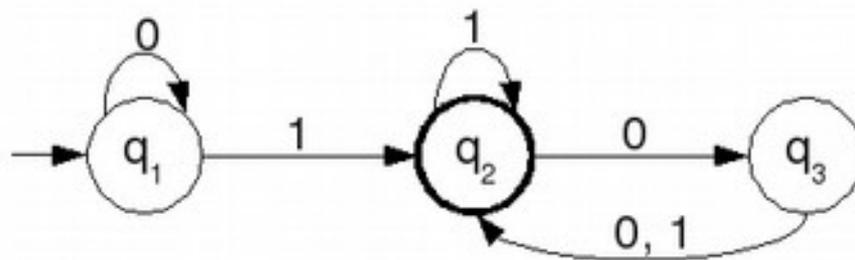
- A unidade de controle possui um número finito e predefinido de estados:
 - A unidade de leitura lê o símbolo de uma célula de cada vez.
 - Após a leitura, a cabeça da fita move-se uma célula para a direita.
 - Inicialmente, a cabeça está posicionada na célula mais à esquerda da fita.

Definição (5/5)

- O programa é uma função parcial que, dependendo do estado corrente e do símbolo lido, determina o novo estado do autômato:
 - Deve-se reparar que o Autômato Finito não possui memória de trabalho.
 - Portanto, para armazenar as informações passadas necessárias ao processamento, deve-se usar o conceito de estado.

Exemplo (1/6)

- Para exemplificar, vamos desenvolver um Autômato Finito Determinístico M_1 , que reconheça a linguagem:
 - $L_1 = \{w \mid w \text{ contém pelo menos um símbolo } 1 \text{ e um número par de símbolos } 0\text{'s seguem o último símbolo } 1\}$ sobre o alfabeto $\Sigma = \{0, 1\}$.



Exemplo (2/6)

- A figura anterior é denominada **diagrama de estados** de M_1 :
 - O autômato tem três **estados**, rotulados q_1 , q_2 e q_3 .
 - O **estado inicial**, q_1 , é indicado pela seta apontando para ele a partir do nada.
 - O **estado de aceitação** ou **estado final**, q_2 , é aquele com um círculo duplo.
 - As setas saindo de um estado para outro são chamadas **transições**.

Exemplo (3/6)

- Quando esse autômato recebe um texto de entrada tal como 1101, ele processa esse texto e produz uma saída:
 - A saída será **aceita** ou **rejeita**.
 - O processamento começa no estado inicial de M_1 .
 - O autômato recebe os símbolos do texto de entrada um por um da esquerda para a direita.
 - Após ler cada símbolo, M_1 move-se de um estado para outro ao longo da transição que tem aquele símbolo como seu rótulo.
 - Quando lê o último símbolo, M_1 produz sua saída.
 - A saída será aceita se M_1 estiver no estado de aceitação e rejeita se não estiver.

Exemplo (4/6)

Execução da entrada 1101 em M_1

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
|---|---|---|---|

q1

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
|---|---|---|---|

q2

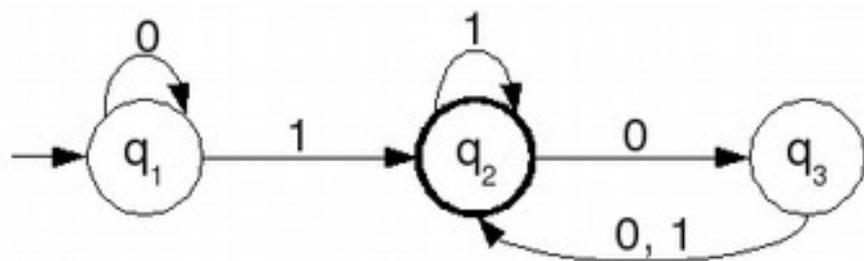
| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
|---|---|---|---|

q2

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
|---|---|---|---|

q3



q2

ACEITA

Exemplo (5/6)

Execução da entrada 0110 em M_1

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
|---|---|---|---|

| |
|----|
| q1 |
|----|

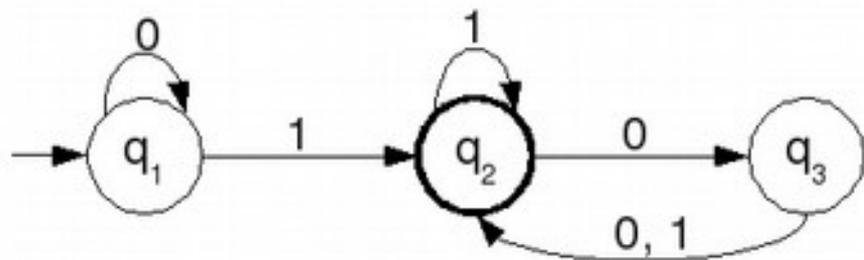
| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
|---|---|---|---|

| |
|----|
| q2 |
|----|

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
|---|---|---|---|

| |
|----|
| q3 |
|----|

REJEITA



| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
|---|---|---|---|

| |
|----|
| q1 |
|----|

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
|---|---|---|---|

| |
|----|
| q2 |
|----|

Exemplo (6/6)

- A experimentação com essa máquina sobre uma variedade de textos de entrada revela que ela aceita os textos 1, 01, 11 e 0101010101:
 - Na realidade, M_1 aceita qualquer texto que termine com o símbolo 1, pois ela vai para seu estado de aceitação q_2 sempre que ela lê o símbolo 1.
 - Além disso, ela aceita os textos 100, 0100, 110000 e 0101000000, e qualquer texto que termine com um número par de 0's seguindo o último 1.
 - Ela rejeita outros textos, como 0, 10 e 101000.

Ponto de Parada (1/2)

- Note-se que um Autômato Finito Determinístico sempre parará ao processar qualquer entrada pois, como qualquer palavra é finita e como um novo símbolo da entrada é lido a cada aplicação da função programa, não existe a possibilidade de ciclo (*loop*) infinito.
- A parada de um processamento pode ser de duas maneiras:
 - Aceitando ou rejeitando uma entrada w .

Ponto de Parada (2/2)

- As condições de parada são as seguintes:
 - Após processar o último símbolo da fita, o Autômato Finito Determinístico assume um estado final: o autômato é encerrado e a entrada w é **aceita**.
 - Após processar o último símbolo da fita, o Autômato Finito Determinístico assume um estado não final: o autômato é encerrado e a entrada w é **rejeitada**.
 - A função programa é indefinida para o argumento (estado corrente e símbolo lido): o autômato é encerrado e a entrada w é **rejeitada**.

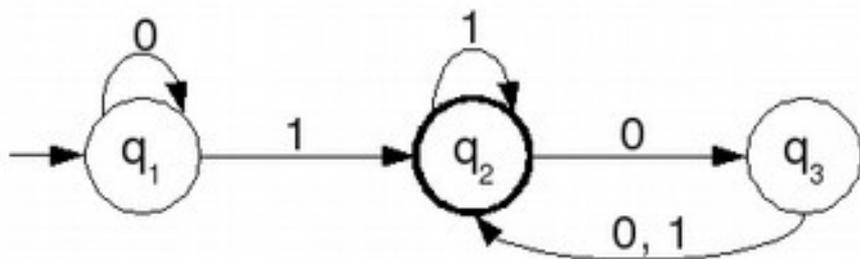
Formalismo (1/2)

- Um Autômato Finito Determinístico é definido como uma quintupla $M = (\Sigma, Q, \delta, q_0, F)$, onde:
 - Σ é um conjunto finito chamado de **alfabeto**.
 - Q é um conjunto finito conhecido como os **estados**.
 - $\delta: Q \times \Sigma \rightarrow Q$ é a **função de transição**.
 - $q_0 \in Q$ é o **estado inicial**.
 - $F \subseteq Q$ é o **conjunto de estados finais**.

Formalismo (2/2)

- A função de transição (δ) pode ser representada através de um grafo ou através de uma tabela:

$$M_1 = (\{0, 1\}, \{q_1, q_2, q_3\}, \delta, q_1, \{q_2\})$$



| δ_1 | 0 | 1 |
|----------------------|----------------|----------------|
| q₁ | q ₁ | q ₂ |
| q₂ | q ₃ | q ₂ |
| q₃ | q ₂ | q ₂ |