LINGUAGENS REGULARES - AUTOMATOS FINITOS DETERMINÍSTICOS

Prof. Alexandre Agustini
alexandre.agustini@pucrs.br

Material original desenvolvido pelos profs. Júlio Machado, Renata Vieira, Alexandre Agustini e outros

Máquinas de estados finitos

- As máquinas de Estados Finitos são máquinas abstratas que capturam as partes essenciais de algumas máquinas concretas (máquinas de vendas de refrigerantes, relógios digitais, elevadores, computador digital, programas, etc).
- Existem basicamente, dois tipos de máquinas de estados finitos: os TRANSDUTORES e os RECONHECEDORES (aceitadores) de linguagens.
- Os transdutores são máquinas com entrada e saída.
- Os reconhecedores são máquinas com apenas duas saídas possíveis: uma ACEITAÇÃO da entrada ou uma REJEIÇÃO da entrada.

Máquinas de estados finitos

- As linguagens reconhecidas por máquinas de estados finitos são denominadas de linguagens regulares.
- Existem duas notações úteis para a especificação de linguagens regulares: GRAMÁTICAS REGULARES e EXPRESSÕES REGULARES.
- Uma característica fundamental de uma máquina de estados finitos é que sua memória é limitada e exclusivamente organizada em torno do conceito de ESTADO.

Máquina de estados finitos como um reconhecedor

- Máquina abstrata baseada em estados e instruções primitivas que modificam os estados
  - A máquina representa uma linguagem
  - Analisa uma entrada (programa) e tenta reconhecê-la, o que só acontece se a linguagem foi usada corretamente
- Cada estado resume somente as informações do passado necessárias para determinar as ações para a próxima entrada

Autômatos finitos determinísticos

Um AUTÔMATO FINITO DETERMINÍSTICO (AFD) A é uma quíntupla $A = (E, \Sigma, \delta, i, F)$, onde:

- E é um conjunto finito de um ou mais elementos denominados estados;
- Σ é um alfabeto;
- δ: $E \times \Sigma \rightarrow E$ é uma funcão de transição (recebe um estado e um símbolo de entrada e retorna um estado);
- i : um estado de E, é o estado inicial;
- F um subconjunto de E, é o conjunto de estados finais.

Exemplo de AFD

Um AFD para reconhecer a linguagem
$L = \{w \in \{0,1\}^* \mid (|w|\%2) = 1\}$, isto é, todos os strings sobre $\Sigma=\{0,1\}$ com comprimento impar.

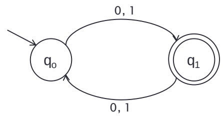
- E = {par,impar}
- Σ = {0,1}
- I = par
- F = {impar}

onde a função de transição δ: $E \times \Sigma \rightarrow E$ é definida como segue:

δ	0	1
par	impar	impar
impar	par	par

AFD – representação gráfica

- Na notação gráfica para AFD's, a função de transição é representada através de um grafo dirigido.
- Os estados são representados por círculos e as transições entre os estados por arcos rotulados (por exemplo par = q_0 e impar= q_1).
- O estado inicial é representado por uma seta, enquanto o estado final por dois círculos concêntricos.
- Exemplo: representação gráfica do AFD COMP-IMPAR:



Processamento do autômato (informal)

Processamento do Autômato Finito *A* para uma palavra de entrada *w* é realizado através de aplicações sucessivas da função de transição

- A *função de transição* é ativada para cada símbolo de *w* (da esquerda para a direita) até ocorrer uma *condição de parada*
- *Um AFD sempre para ao processar qualquer entrada visto que cada palavra é finita*
- A parada pode ser de duas maneiras: **ACEITANDO** ou **REJEITANDO** a entrada w

Condição de parada (informal)

1. Após processar o último símbolo da fita, o autômato assume um estado final
   - o autômato para e a palavra de entrada w é **aceita**;
2. Após processar o último símbolo da fita, o autômato assume um estado não-final
   - o autômato para e a entrada w é **rejeitada**;
3. A função de transição é indefinida para o argumento (estado corrente, símbolo lido)
   - a máquina para e a palavra w é **rejeitada**

Função de Transição Estendida

Seja um AFD $A = (E, \Sigma, \delta, i, F)$. A função de transição estendida para *A*, $\delta^*: E \times \Sigma^* \rightarrow E$, é definida recursivamente como segue:

1. $\delta^*(e, \varepsilon) = e$ para qualquer $e \in E$;
2. $\delta^*(e, aw) = \delta^*(\delta(e,a),w)$, para qualquer $e \in E$, $a \in \Sigma$, $w \in \Sigma^*$

Computando a Função de Transição Estendida - exemplo

Seja o AFD COMP-IMPAR, apresentado anteriormente. Seja o string 0101 em Σ^*. Então temos que:

$$
\begin{aligned}
\delta^*(par, 0101) &= \delta^*(\delta(par, 0), 101) \\
&= \delta^*(impar, 101) \\
&= \delta^*(\delta(impar, 1), 01) \\
&= \delta^*(par, 01) \\
&= \delta^*(\delta(par, 0), 1) \\
&= \delta^*(impar, 1) \\
&= \delta^*(\delta(impar, 1), \varepsilon) \\
&= \delta^*(par, \varepsilon) \\
&= \text{par}
\end{aligned}
$$

Linguagem definida por um AFD

Uma linguagem reconhecida (aceita) por um AFD $A = (E, \Sigma, \delta, i, F)$ é o conjunto $L(A) = \{w \in \Sigma^* \mid \delta^*(i,w) \in F\}$. Uma determinada palavra $w \in \Sigma^*$ é dita ser reconhecida, ou aceita por *A* se e somente se $\delta^*(i,w) \in F$.

Exemplos

- 1010 $\notin$ L(COMP-IMPAR) já que $\delta^*(par,1010) = par$ e $par \notin F$.
- 110 $\in$ L(COMP-IMPAR), já que $\delta^*(par,110) = impar$ e $impar \in F$

Execução de um AFD

- Dado um AFD $A = (E, \Sigma, \delta, i, F)$, um símbolo $a \in \Sigma$ e uma palavra $w \in \Sigma^*$, um passo de computação em A é uma dupla $[q_i, aw] \rightarrow [q_j, w]$, onde q_i e $q_j \in E$ e $\delta(q_i, a) = q_j$.

- Dada da uma palavra $w \in \Sigma^*$, uma computação (run, trace) é uma sequência de passos de computação

$$[i, w] \rightarrow .. \rightarrow [e, \varepsilon],$$

onde i é o estado inicial do automato e $\delta^*(i,w) = e$.

Exemplos de computação

Dado o autômato COMP-IMPAR

- Um exemplo de computação de aceitação:

$$[q_0, 101] \rightarrow [q_1, 01] \rightarrow [q_0, 1] \rightarrow [q_1, \varepsilon]$$

- Um exemplo de computação de rejeição:

$$[q_0, 1101] \rightarrow [q_1, 101] \rightarrow [q_0, 01] \rightarrow [q_1, 1] \rightarrow [q_0, \varepsilon]$$