Aula 7 – Linguagens Regulares

7.1 Autômato Finito Não Determinístico

O não determinismo é uma importante generalização dos modelos de máquinas, sendo de fundamental importância no estudo dos Modelos para Concorrência, da Teoria da Computação e das Linguagens Formais, entre outros.

Em oposição ao que acontece com o AFD, a função de transição de um AFND não precisa determinar exatamente qual deve ser o próximo estado. Em vez disso, a função de transição fornece uma lista (um conjunto) de estados para os quais a transição poderia ser feita. Esta lista pode ser vazia, ou ter um número qualquer positivo de elementos.

A facilidade de não-determinismo para autômatos finitos é expressa no programa, que é uma função parcial tal que:

dependendo do estado corrente e do símbolo lido,

determina um conjunto de estados do autômato.

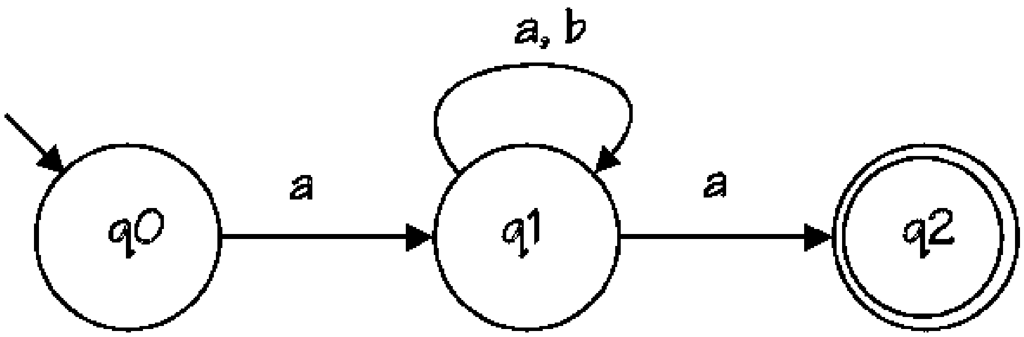
Essa possibilidade de escolha entre vários caminhos a serem seguidos nos leva a modificar a definição de aceitação.

Um AFD aceita se "**o último estado atingido é final**"; mas um AFND aceita se "**existe uma sequência de escolhas tal que o último estado atingido é final**".

Podemos alternativamente imaginar que o AFND "escolhe", "adivinha", o caminho certo para a aceitação, uma vez que a existência de escolhas erradas, que não levam a um estado final, é irrelevante.

Visto como uma máquina composta por fita, unidade de controle e programa, um Autômato Finito Não-Determinísticos assume um conjunto de estados alternativos, como se houvesse uma multiplicação da unidade de controle, uma para cada alternativa, processando independentemente, sem compartilhar recursos com as demais. Assim, o processamento de um caminho não influi no estado, símbolo lido e posição da cabeça dos demais caminhos alternativos.

Considere o AFND dado pelo diagrama abaixo e a cadeia de entrada ababa.



A cadeia ababa é aceita, porque uma das possibilidades é a sequência de estados q0, q1, q1,q1,q1,q2. Naturalmente, com a mesma cadeia, poderíamos escolher a sequência q0,q1,q1,q1,q1,q1, que não leva a um estado final. Ou ainda, a sequência q0,q1,q1,q2 interrompida, porque q2 não prevê uma transição com o segundo b.

Mas estes casos em que o “autômato adivinhou errado” não criam problemas para a aceitação, porque “existe um caminho certo”.

Este AFND aceita a linguagem das cadeias (de comprimento maior ou igual a 2), cujo primeiro e último símbolos são a, sendo os restantes quaisquer.

Definição: Formalmente, um Autômato Finito Não Determinístico (AFND) M, sobre um alfabeto Σ é um sistema (Q, Σ, δ, q0, F), onde:

Q é um conjunto (finito, não vazio) de estados possíveis

Σ é um alfabeto de entrada (finito)

δ: Q x ($\Sigma \cup \{\varepsilon\}$) $\rightarrow$ P(Q) é uma função de transição, ou função programa ou programa

q0 é o estado inicial (ou representado também por i)

F é o conjunto de estados finais.

A notação P(Q) indica o conjunto “partes” de Q (conjunto potência de Q), o conjunto de todos os subconjuntos de Q.

Pela definição, portanto, δ é uma função que aceita como argumentos q e a, onde q é um estado e a pode ser um símbolo de Σ ou a cadeia vazia ε. Em qualquer caso, δ(q, a) é sempre um conjunto de estados, ou seja, um subconjunto de Q.

Se tivermos δ(q,a) = {p1, p2, ... pq}, entendemos que o autômato M, a partir do estado q, pode escolher um dos estados p1, p2, ..., pq para ser o próximo estado. Se a = ε, nenhum símbolo de entrada é lido; se a $\neq \varepsilon$, o símbolo a de entrada é lido.

Podemos considerar o caso a = ε como correspondente a transições espontâneas: M muda de estado sem estímulo de entrada. Se tivermos δ(q, a) = $\emptyset$, não há transições possíveis a partir do estado q como símbolo a.

Definimos configurações para o caso do AFND da mesma forma que anteriormente. A mudança de configuração é dada pela relação |—, definida abaixo:

(q, ax) |— (p, x) se e somente se p $\in \delta$(q, a)

Note que a pode ser a cadeia vazia, caso em que temos, particularizando,

(q, x) |— (p, x) se e somente se p $\in \delta$(q, ε)

Podemos agora definir a linguagem L(M) por

L(M) = {x $\in \Sigma^*$ | (i, x) |— *(f, ε), com f $\in$ F}

Exemplo (continuação): Temos, para a mesma cadeia ababa de entrada,

(q0, ababa) |— (q1, baba) |— (q1, aba) |— (q1, ba) |— (q1, a) |— (q2, ε)

e, portanto, ababa $\in$ L(M). Temos também o “caminho errado”

(q0, ababa) |— (q1, baba) |— (q1, aba) |— (q1, ba) |— (q1, a) |— (q1, ε)

Que leva a configuração não final (q1, ε), e não permite nenhuma conclusão.

Cadeias como bab e abab não levam a configurações finais e não são aceitas. Da configuração (q0, bab) nenhuma configuração é atingível; para abab temos:

(q0, abab) |— (q1, bab) |— (q1, ab) |— (q1, b) |— (q1, ε)

Adicionalmente, temos um outro caminho

(q0, abab) |— (q1, bab) |— (q1, ab) |— (q2, b)

Que também não atinge nenhuma configuração final. Assim, as cadeias bab e abab não são aceitas e não fazem parte de L(M).

Exemplo1: Considere a:

L = {x y z | x, z $\in$ {a, b}* e (y = aaa ou y = bb)}

Construa um AFND M que aceite L.

Sugestão: M adivinha se a cadeia de entrada contém aaa ou bb, e apenas verifica esse fato.

Exercícios de Fixação:

1. Desenvolva Autômatos Finitos Não-Determinísticos (AFND) ou Autômatos Finitos Determinísticos (AFD), conforme o enunciado solicita, que reconheçam as seguintes Linguagens:

a) Sobre o Alfabeto Σ = {a, b, c}

L = {w | aa ou bb é subpalavra e cccc é sufixo de w}

- Construa o AFD e o AFND para a Linguagem acima.

b) Sobre o Alfabeto Σ = {a, b}, construa o AFND para a Linguagem abaixo:

L = {$w_1w_2w_1$ | $w_2 \in$ {a,b}e $|w_1|$ = 3}

2. Sejam as linguagens na forma L = {xyx | x, y $\in$ {a,b}* e x = n}. Determine o menor número de estados para um AFND e para um AFD, que reconheçam L, nos seguintes casos:

a) n = 1;

b) n = 2;

c) n arbitrário.