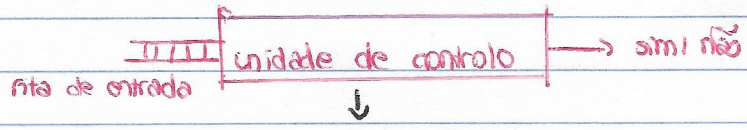
representação analítica

$M = (A, Q, q_0, \delta, F)$

3.2. Autómatos finitos

mecanismo reconhecedor das palavras de uma LR



com base no nº de estados (nº finito) e das transições entre eles

determinista AFD

transições associadas a símbolos individuais do alf-abeto

de cada estado sai uma e uma só transição por símbolo 

há um estado inicial → (A)

há zero ou + estados de aceitação (C)

palavra aceite consiste num caminho com início no estado inicial e fim num de aceitação

$M = (A, Q, q_0, \delta, F)$

A alfabeto de entrada

Q conjunto finito não vazio de estados

$q_0 \in Q$ estado inicial

δ $Q \times A \to Q$ é função que determina transições

F conjunto dos estados de aceitação

δ é dada por uma matriz $|Q| \times |A|$ com entradas $q \in Q$

Linguagem reconhecida por um AFD

dada a palavra $v = v_1 v_2 \dots v_n$, tem de existir

1. $s_0 = q_0$
2. $\forall i = 1, \dots, n \quad s_i = \delta(s_{i-1}, v_i)$
3. $s_n \in F$

sendo $s_i \in Q$

AFD aceita v se $\delta^*(q_0, v) \in F$

sendo $\delta^*(q, \varepsilon) = q$ e $\delta^*(q, av) = \delta^*(\delta(q, a), v)$

Seja M um AFD

$L(M) = \{ v \in A^* : \delta^*(q_0, v) \in F \}$

redução de AFD

1. dividir estados em aceitação e não aceitação (classes)
2. dentro das classes, identificar transições internas e dividir em subclasses
3. juntar subclasses num único estado

---

não determinista AFND

transições associadas a símbolos individuais ou palavra vazia ε
de cada estado saem $\emptyset$ ou + transições
há um estado inicial
há $\emptyset$ ou + estados de aceitação

passam assim a existir caminhos alternativos

$M = (A, \emptyset, q_0, \delta, F)$
δ difere dos AFD pois os estados são subconjuntos de $\emptyset$

Linguagem reconhecida por um AFND

dada a palavra $v = v_1 v_2 \dots v_n$ tem de existir
1. $S_0 = q_0$
2. $\forall i = 1, \dots, n \quad (S_{i-1}, v_i, S_i) \in \delta$
3. $S_n \in F$

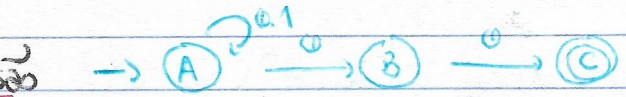
nota n pode ser maior que $|v|$ devido aos ε

Seja M um AFND
$L(M) = \{ v : q_0 \xrightarrow{v} q_F \wedge q_F \in F \}$

equivalência entre AFD e AFND

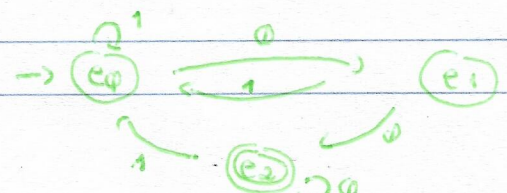
a classe de linguagens cobertas por um AFD é a mesma das cobertas por um AFND
$L(M') = L(M)$

conversão



	$\emptyset$	1
e_0 $\{A\}$	$\{A, B\}$ e_1	$\{A\}$ e_0 inicial
e_1 $\{A, B\}$	$\{A, B, C\}$ e_2	$\{A\}$ e_0
e_2 $\{A, B, C\}$	$\{A, B, C\}$ e_2	$\{A\}$ e_0 final pq $C \in F$

estados finais
todo conjunto que tenha estado final do AFND



operações sobre AFD e AFND

AF são fechados sobre as operações de reunião, concatenação, fecho, interseção, complementação

reunião criar estado novo com transição ε para iniciais

$Q = Q_1 \cup Q_2 \cup \{q_0\}$

q_0 novo estado com transições ε para iniciais q_1 e q_2

$\delta = \delta_1 \cup \delta_2 \cup \{(q_0, \varepsilon, q_1), (q_0, \varepsilon, q_2)\}$

$F = F_1 \cup F_2$

$L(M) = L(M_1) \cup L(M_2)$

concatenação dados AF M_1 e M_2 (~~estado inicial~~ ~~estados finais~~)

manter inicial de M_1 e finais de M_2

finais de M_1 transicionam a ε para inicial de M_2

$Q = Q_1 \cup Q_2$

$q_0 = q_1$

$\delta = \delta_1 \cup \delta_2 \cup (F_1 \times \{\varepsilon\} \times \{q_2\})$

$F = F_2$

$L(M) = L(M_1) \cdot L(M_2)$

fecho novo estado inicial e de aceitação com trans. ε

$Q = Q_1 \cup \{q_0\}$

$q_0 = q_0$

$\delta = \delta_1 \cup (F_1 \times \{\varepsilon\} \times \{q_0\}) \cup \{(q_0, \varepsilon, q_1)\}$

$F = \{q_0\}$

$L(M) = L(M_1)^*$

interseção criar AF para linguagens

criar estados que resultem de $Q_1 \times Q_2$

para cada estado $S_{i,j}$ e $\forall a \in A$:

$S_i \xrightarrow{a} S_k$ $S_j \xrightarrow{a} S_w$ então $S_{i,j} \xrightarrow{a} S_{k,w}$

$Q = Q_1 \times Q_2$

$q_0 = (q_1, q_2)$

$\delta \subseteq (Q_1 \times Q_2) \times A\varepsilon \times (Q_1 \times Q_2)$

$F = F_1 \times F_2$

complementação obter AFD e complemento (trocar) estados aceitação

equivalência entre ER e AF

a classe das linguagens cobertas por ER é a mesma das cobertas pelos AF

$L(e) = L(M) \quad (\forall e \in ER \; \exists M \in AF)$

conversão ER → AF

dadas as ER básicas e, e^*, $e_1 | e_2$, $e_1 e_2$

identificam-se os AF relativos às ER e fazem-se as operações sobre os AF

1º desconstruir ER em árvore

2º construir AF a partir das folhas

e $()$ → ○

ε → ◎

a → ○ —a→ ◎

e^* fecho do AF de e

$e_1|e_2$ reunião dos AF de e_1 e e_2

$e_1 . e_2$ concatenação dos AF de e_1 e e_2

AF Generalizado AFG

distingue-se dos restantes por as etiquetas de δ serem ER

$M = (A, Q, q_0, \delta, F)$

$\delta \subseteq (Q \times E \times Q)$ sendo E o conjunto das ER def sobre A

∴ os AFD e AFND são AFG

conversão AFG em ER

ER equivalente ao AFG

o objetivo é obter autómato → ○ —e→ ◎

estado inicial não pode ter transições a chegar

se necessário criar novo com transição ε para antigo

estado de aceitação único sem transições de saída

se necessário criar novo que é transicionado a ε desde os antigos

eliminar restantes estados mantendo equivalência

+ ver algoritmo de eliminação de um estado