LFA CURS 2

AUTOMATE FINITE NEDETERMINISTE CU λ-TRANZIȚII (AFN_λ)

<u>Definiție 1</u> Un AFN_λ are o structură de forma

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

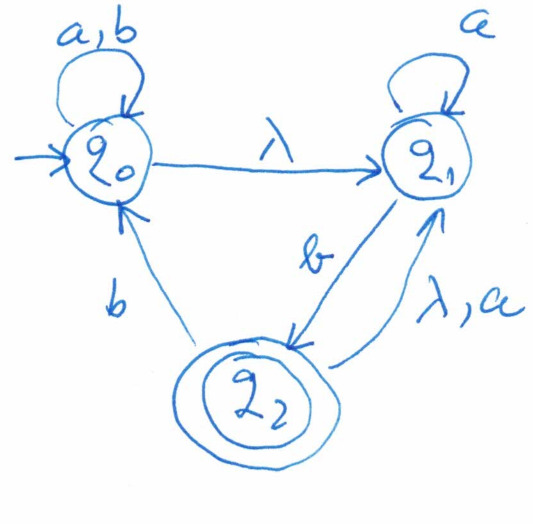
unde Q, Σ, q_0, F au aceeași semnificație ca în cazul AFD sau AFN, iar funcția de tranziție este

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \to 2^Q$$

În acest caz:

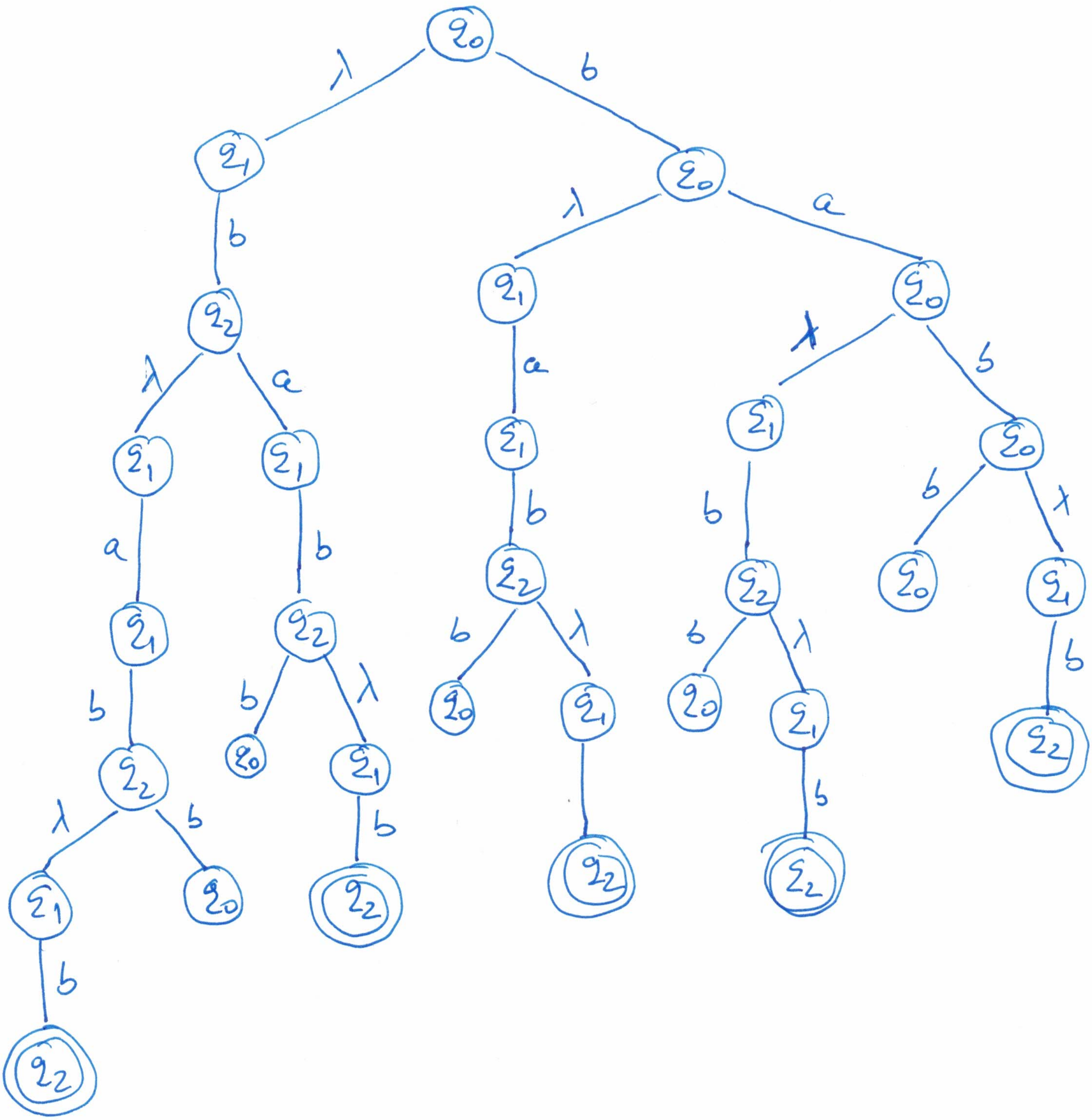
- automatul poate să își schimbe starea fără să citească niciun simbol din Σ. Aceasta este numită o λ-tranziție (automatul citește λ)
- pentru o anumită stare și un anumit simbol (posibil λ) pot exista: cel puțin două tranziții, o tranziție sau niciuna.

<u>Exemplu</u> $A = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$



δ	a	b	λ
q_0	$\{q_0\}$	$\{q_0\}$	$\{q_1\}$
q_1	$\{q_1\}$	$\{q_2\}$	$\{\}$
q_2	$\{q_1\}$	$\{q_0\}$	$\{q_1\}$

Drumuri posibile în graful automatului etichetate cu babb



Definiție 2 Fie $A=(Q,\Sigma,\delta,q_0,F)$ un AFNλ, $T\subseteq Q$.
Numim λ-închidere a lui T submulțimea lui Q notată cu $\langle T\rangle$ definită prin

$$\langle T\rangle = \{ p\in Q \mid \exists\, r_0\in T, r_1,\dots,r_m\in Q,\ m\geq 0,\ r_m=p, \text{ astfel încât } r_i\in\delta(r_{i-1},\lambda),\ i=1,\dots,m\}$$

Observații

1) $\forall\, p\in T$, $p\in\langle T\rangle$ (în explicitarea lui $\langle T\rangle$ luăm $m=0$)
2) În $\langle T\rangle$ apar toate stările ce provin din stări din T prin λ-tranziții.
3) $T\subseteq\langle T\rangle$ (rezultă din 1))
4) $\langle\langle T\rangle\rangle=\langle T\rangle$
5) $\langle T\cup T'\rangle=\langle T\rangle\cup\langle T'\rangle$

ALGORITM PENTRU DETERMINAREA LUI $\langle T\rangle$

Input $A=(Q,\Sigma,\delta,q_0,F)$ AFNλ, $T\subseteq Q$

Output $\langle T\rangle$

Metoda. Se utilizează o stivă STIVA care se inițializează cu T. Pentru fiecare q din vârful stivei (cât stiva nu este vidă) stările din $\delta(q,\lambda)$ ce nu au fost introduse în $\langle T\rangle$ se adaugă în $\langle T\rangle$ și în stivă.

```
R = T; STIVA = T;
while (STIVA ≠ ∅) {
    q = STIVA.pop(); // Se extrage q din stivă
    S = δ(q, λ);
    if (S ≠ ∅) {
        for (p ∈ S) {
            if (p ∉ R) {
                R = R ∪ {p}
                STIVA.push(p); // Se adaugă p în stivă
            } // endif
        } // endfor
    } // endif
} // endwhile
<T> = R;
```

<u>Definiție 3</u> Extinderea funcției de tranziție pentru AFN$_\lambda$ $A = (Q, \Sigma, \delta, q_0, F)$:

$$\hat{\delta} : Q \times \Sigma^* \to 2^Q$$

$$\hat{\delta}(q, \lambda) = \langle q \rangle, \ \forall q \in Q$$

$$\hat{\delta}(q, wa) = \left\langle \bigcup_{p \in \hat{\delta}(q,w)} \delta(p, a) \right\rangle$$

Observații

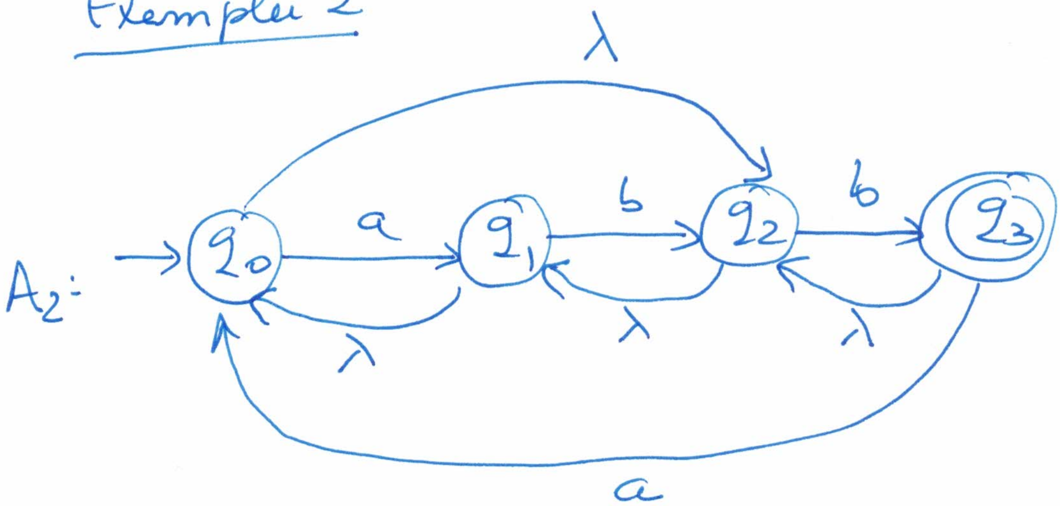
6) $\hat{\delta}(q, wa) = \left\langle \bigcup_{p \in \hat{\delta}(q,w)} \delta(p,a) \right\rangle = \bigcup_{p \in \hat{\delta}(q,w)} \langle \delta(p,a) \rangle$

7) Este posibil să existe $q \in Q$, $a \in \Sigma$ pentru care $\hat{\delta}(q, a) \neq \delta(q, a)$

Definiție 3 Limbajul acceptat de AFN$_\lambda$ $A=(Q,\Sigma,\delta,q_0,F)$ este definit prin:

$$L(A)=\{w\in\Sigma^* \mid \hat{\delta}(q_0,w)\cap F\neq\emptyset\}$$

Exemplu 2



$\langle q_0\rangle=\{q_0,q_1,q_2\}$

$\langle q_1\rangle=\{q_0,q_1,q_2\}$

$\langle q_2\rangle=\{q_0,q_1,q_2\}$

$\langle q_3\rangle=\{q_0,q_1,q_2,q_3\}$

Definiție 4 Fie AFN$_\lambda$ $A=(Q,\Sigma,\delta,q_0,F)$, $T\subseteq Q$, $a\in\Sigma$.

Notăm cu move(T,a) mulțimea stărilor definită prin move$(T,a)=\{p\in Q \mid \exists r\in T, p\in\delta(r,a)\}=\bigcup_{r\in T}\delta(r,a)$

Observații

8. move$(\emptyset,a)=\emptyset$, $\forall a\in\Sigma$

9. $\langle\emptyset\rangle=\emptyset$

<u>Teoremă</u> Pentru fiecare AFN, $A=(Q,\Sigma,\delta,q_0,F)$ există un AFD A' astfel încât $L(A')=L(A)$

<u>Demonstrație</u> Fie $A'=(Q',\Sigma,\delta',q_0',F')$, unde $Q'\subseteq 2^Q$, $q_0'=\langle q_0\rangle$, Q' și δ' sunt construite cu algoritmul următor:

```
Q' = {q'0}, q'0 stare nemarcată
while (există T ∈ Q' stare nemarcată) {
    marchează T;
    for (a ∈ Σ) {
        U = move(T, a);
        if (U ∉ Q') adaugă U la Q' ca stare nemarcată;
        δ'(T, a) = U
    } // end for
} // endwhile
F' = {T ∈ Q' | T ∩ F ≠ φ}
```

Automatul A' este complet determinist, deoarece pentru fiecare stare din Q' (care este o submulțime a lui Q) și pentru fiecare $a\in\Sigma$ se adaugă o unică tranziție în A'.

Arătăm că $\forall w \in \Sigma^*$, $\hat{\delta}(q_0, w) = \delta'(q_0', w)$

Facem inducție după $n = |w|$.

$\boxed{n = 0}$ $w = \lambda$; $\hat{\delta}(q_0, \lambda) \underset{\substack{\uparrow \\ \text{definiție}}}{=} \langle q_0 \rangle = q_0' = \delta'(q_0', \lambda)$

$\boxed{n \to n+1}$ Presupunem că $\hat{\delta}(q_0, w) = \delta'(q_0', w)$, $\forall w \in \Sigma^*$, $|w| = n$; δ' este definită pe cuvinte (este de fapt extinderea lui δ').

Fie șirul wa, $w \in \Sigma^*$, $|w| = n$, $a \in \Sigma$. Avem:

$$\hat{\delta}(q_0, wa) \underset{\substack{\uparrow \\ \text{definiție } \hat{\delta}}}{=} \left\langle \bigcup_{p \in \hat{\delta}(q, w)} \delta(p, a) \right\rangle$$

$$\underset{\substack{\uparrow \\ \text{definiție move}}}{=} \langle move(\hat{\delta}(q_0, w), a) \rangle$$

$$\underset{\substack{\uparrow \\ \text{algoritm}}}{=} \delta'(\hat{\delta}(q_0, w), a)$$

$$\underset{\substack{\uparrow \\ \text{ipoteza de ind.}}}{=} \delta'(\delta'(q_0, w), a) \underset{\substack{\uparrow \\ \text{definiție} \\ \text{extensie } \delta'}}{=} \delta'(q_0, wa)$$

Atunci:

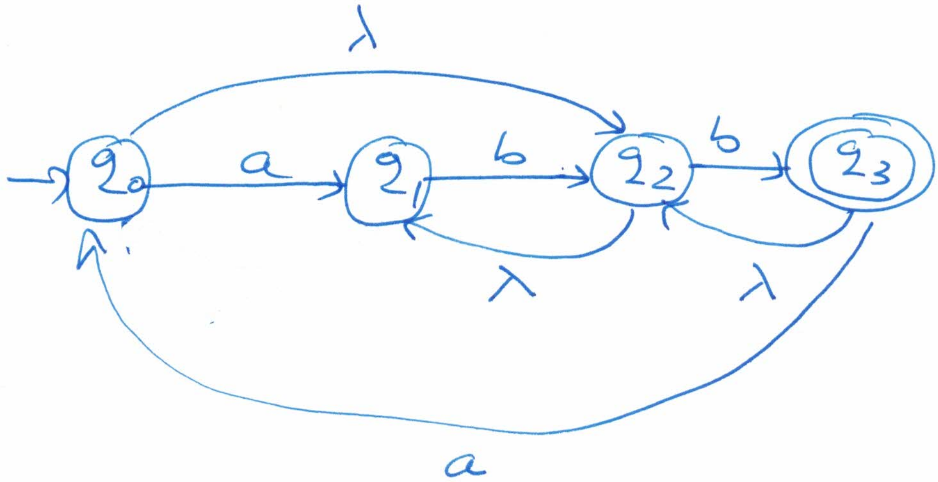
$w \in L(A') \Leftrightarrow \delta'(q_0', w) \in F' \Leftrightarrow \delta'(q_0', w) \cap F \neq \emptyset$

$\Leftrightarrow \hat{\delta}(q_0, w) \cap F \neq \emptyset \Leftrightarrow w \in L(A)$.

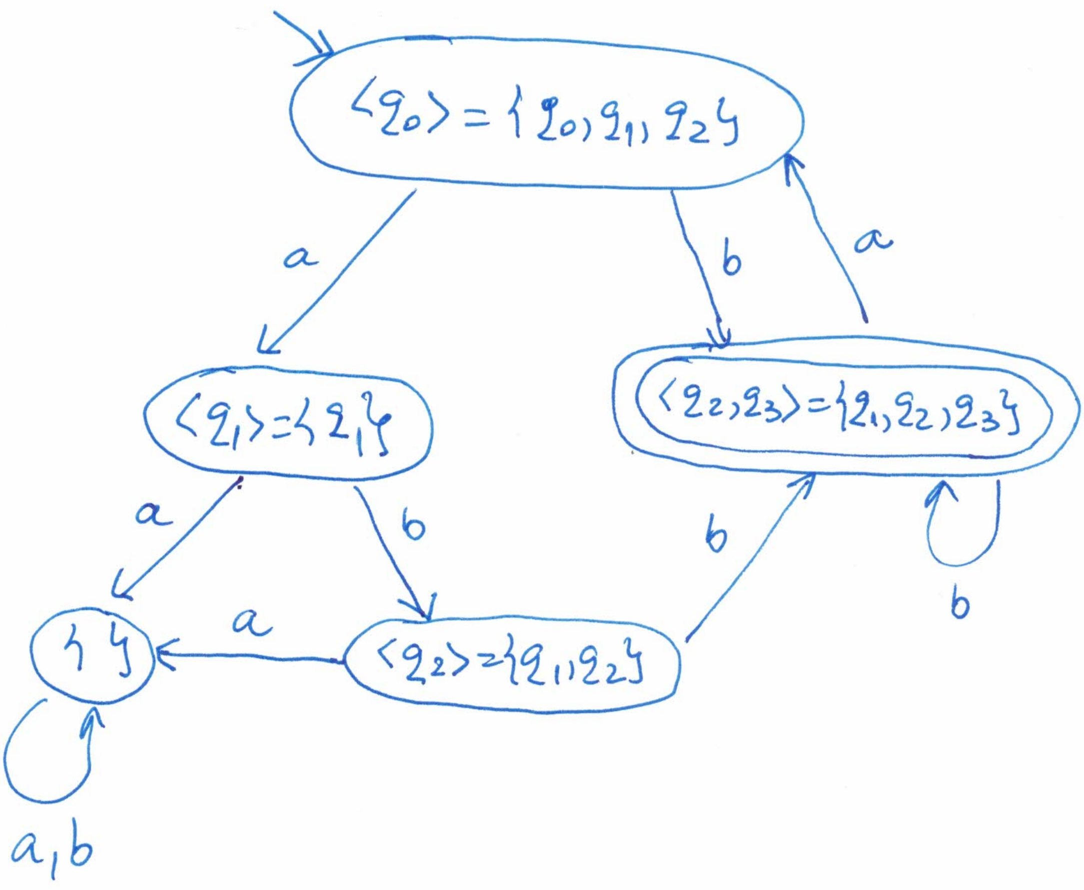
Rezultă că $L(A) = L(A')$

q.e.d.

Exemplul 3 Se dă automatul cu λ-tranziții:



AFD care se obține cu algoritmul din Teorema anterioară este:



Definiție 4 Spunem că AFN$_\lambda$ $A=(Q,\Sigma,\delta,q_0,F)$ este determinist dacă:

$\forall p\in Q, \forall a\in\Sigma, \ |\delta(p,a)|+|\delta(p,\lambda)|\le 1$

În acest caz, pentru orice stare $p\in Q$, $\forall a\in\Sigma$ există cel mult o tranziție cu a ce pleacă din p, iar în cazul în care o astfel de tranziție există, atunci din p nu avem tranziție cu λ. Pe de altă parte, dacă din p există o tranziție cu λ, atunci aceasta este unica tranziție.

Într-un AFN$_\lambda$ determinist, drumurile de la o stare la alta sunt unice (dacă există).

DEFINIREA LIMBAJULUI ACCEPTAT DE UN AFN$_\lambda$ CU AJUTORUL UNEI RELAȚII CE DESCRIE SCHIMBĂRILE DE CONFIGURAȚII ALE AUTOMATULUI

Definiție 5 Fie AFN$_\lambda$ $A=(Q,\Sigma,\delta,q_0,F)$.
Numim descriere instantanee (instanță) a lui A o pereche (p,w), unde $p\in Q$ este starea curentă a lui A iar $w\in\Sigma^*$ este șirul de pe banda de intrare care a rămas de scanat; capul de citire al automatului este poziționat pe primul simbol al lui w, dacă $w\neq\lambda$.
Numim o instanță a lui A și configurație.

Numim mișcare a lui A (schimbare de configurații) o relație $\vdash$ definită pe mulțimea configurațiilor lui A în felul următor:

$(p, aw) \vdash (q, w)$ ddacă $q \in \delta(p, a)$,

pentru $p, q \in Q$, $w \in \Sigma^*$, $a \in \Sigma \cup \{\lambda\}$

Observații

10. Pentru A AFN, avem $a \in \Sigma$
11. Pentru A AFD, avem $a \in \Sigma$, $q = \delta(p, a)$

Notăm cu $\vdash^*$ închiderea reflexivă și tranzitivă a relației $\vdash$ ($\vdash^*$ înseamnă zero sau mai multe mișcări ale lui A). Astfel

$(p, w) \vdash^* (q, w') \iff$

a) fie: $q = p$, $w' = w$ (zero mișcări)

b) fie: $\exists\, q_1, \dots, q_n \in Q$, $n \geq 1$, $q_n = q$, $\exists\, a_1, \dots, a_n \in \Sigma \cup \{\lambda\}$ astfel încât $w = a_1 \dots a_n w'$, $q_i \in \delta(q_{i-1}, a_i)$, unde $q_0 = p$.

Limbajul acceptat de A

$$L(A) = \{ w \in \Sigma^* \mid (q_0, w) \vdash^* (q, \lambda),\ q \in F \}$$

Arătăm că această definire a limbajului este consistentă cu

$$L(A) = \{ w \in \Sigma^* \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset \}.$$

Notăm cu $M(q,w)=\{s\in Q \mid (q,w)\vdash^* (s,\lambda)\}\subseteq Q$.

<u>Lemă</u> Pentru orice $q\in Q$ și pentru orice $w\in\Sigma^*$, $\hat{\delta}(q,w)=M(q,w)$.

<u>Demonstrație</u> Facem inducție după $n=|w|$

$\boxed{n=0}$ $w=\lambda$ $\hat{\delta}(q,\lambda)=\langle q\rangle$

$M(q,\lambda)=\{s\in Q \mid (q,\lambda)\vdash^*(s,\lambda)\}=$

$=\{s\in Q \mid \exists\, q_0,q_1,\dots,q_n\in Q,\ n\geq 0,\ q_0=q,\ q_n=s,$
$q_i\in\delta(q_{i-1},\lambda),\ i=1,\dots,n\}=\langle q\rangle=\hat{\delta}(q,\lambda)$

$\boxed{n\to n+1}$ Presupunem că $\forall q\in Q,\ \forall w\in\Sigma^*,\ |w|=n,$
$\hat{\delta}(q,w)=M(q,w)$

Fie $x=wa,\ w\in\Sigma^*,\ |w|=n,\ a\in\Sigma,\ |x|=n+1.$

$M(q,wa)=\{s\in Q \mid (q,wa)\vdash^*(s,\lambda)\}=$

$=\{s\in Q \mid \exists\, p,r\in Q,\ (q,wa)\vdash^*(p,a)\vdash(r,\lambda)\vdash^*(s,\lambda)\}$

$=\{s\in Q \mid \exists\, q_0,q_1,\dots,q_m\in Q,\ m\geq 0,\ q_0=q,\ q_m=p,$
$\exists\, x_1,\dots,x_m\in\Sigma\cup\{\lambda\},\ w=x_1\dots x_m,\ q_i\in\delta(q_{i-1},x_i),$
$r\in\delta(p,a)=\delta(q_m,a),\ \exists\, r_0,r_1,\dots,r_n\in Q,\ n\geq 0,$
$r_0=r,\ r_n=s,\ r_j\in\delta(r_{j-1},\lambda),\ j=1,\dots,n\}$
$=\{s\in Q \mid \exists\, p\in M(q,w),\ s\in\langle\delta(p,a)\rangle\}$

$$\hat{\delta}(q,wa)=\Big\langle\bigcup_{p\in\hat{\delta}(q,w)}\delta(p,a)\Big\rangle=\bigcup_{p\in\hat{\delta}(q,w)}\langle\delta(p,a)\rangle$$

$=\{s\in Q \mid \exists\, p\in\hat{\delta}(q,w),\ s\in\langle\delta(p,a)\rangle\}\ \underset{\substack{\uparrow\\ \text{ipoteza de inducție}}}{=}$

$= \{ s \in Q \mid \exists\, p \in M(q,w),\ s \in \langle \delta(p,a) \rangle \} = M(q,wa)$

q.e.d.

Corolar $w \in L(A) \Leftrightarrow \hat{\delta}(q_0, w) \cap F \neq \emptyset \Leftrightarrow$

$\Leftrightarrow M(q_0, w) \cap F \neq \emptyset \Leftrightarrow \exists\, q \in F,\ (q_0, w) \vdash^* (q, \lambda)$

$\Leftrightarrow w \in \overline{L}(A)$

Relația $\vdash^*$ ne oferă o modalitate mai practică și intuitivă de a defini limbajul acceptat de un AFNλ. Deoarece AFN și AFD sunt particularizări ale AFNλ, putem defini limbajul acceptat de un AFN sau un AFD în termeni de relația $\vdash^*$, particularizată pentru aceste tipuri de automate.

OPERAȚII CU LIMBAJE ACCEPTATE DE AUTOMATE FINITE

În cele ce urmează vom considera următoarele operații cu limbaje
$\cup$, $\cap$, $-$ (diferența), $\overline{}$ (complementara).

Teorema 2 Fie $A_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ și $A_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ două AFD care acceptă limbajele $L_1, L_2 \subseteq \Sigma^*$.

Fie AFD $A = (Q, \Sigma, \delta, q_0, F)$ definit prin $Q = Q_1 \times Q_2$, $q_0 = (q_1, q_2)$, $\delta : (Q_1 \times Q_2) \times \Sigma \to Q_1 \times Q_2$
$\delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$

1. Dacă $F = \{(p, q) \mid p \in F_1$ sau $q \in F_2\}$
   atunci $L(A) = L_1 \cup L_2$
2. Dacă $F = \{(p, q) \mid p \in F_1$ și $q \in F_2\}$
   atunci $L(A) = L_1 \cap L_2$
3. Dacă $F = \{(p, q) \mid p \in F_1$ și $q \notin F_2\}$
   atunci $L(A) = L_1 - L_2$

Demonstrație Arătăm ~~prin~~ inducție după $n = |x|$, $x \in \Sigma^*$, că $\hat{\delta}(q_0, x) = (\hat{\delta}_1(q_1, x), \hat{\delta}_2(q_2, x))$

$\boxed{n=0}$ $x = \lambda$; $\hat{\delta}(q_0, \lambda) = q_0 = (q_1, q_2) = (\hat{\delta}_1(q_1, \lambda), \hat{\delta}_2(q_2, \lambda))$

$\boxed{n \to n+1}$ Presupunem că $\hat{\delta}(q_0, x) = (\hat{\delta}_1(q_1, x), \hat{\delta}_2(q_2, x))$, $\forall x \in \Sigma^*, |x| = n$.

Fie acum $y = xa$, $a \in \Sigma$, $|x| = n$, $|y| = n+1$.

$$\hat{\delta}(q_0, y) = \hat{\delta}(q_0, xa) = \delta(\hat{\delta}(q_0, x), a) \underset{\text{ip. de inducție}}{=}$$
$$= \delta((\hat{\delta}_1(q_1, x), \hat{\delta}_2(q_2, x)), a) =$$
$$\underset{\text{definiție } \delta}{=} (\delta_1(\hat{\delta}_1(q_1, x), a), \delta_2(\hat{\delta}_2(q_2, x), a)) =$$

$= (\hat{\delta}_1(q_1, xa), \hat{\delta}_2(q_2, xa)) = (\hat{\delta}_1(q_1, y), \hat{\delta}_2(q_2, y))$

1. $x \in L(A) \Leftrightarrow \hat{\delta}(q_0, x) \in F \Leftrightarrow (\hat{\delta}_1(q_1, x), \hat{\delta}_2(q_2, x)) \in F$

$\Leftrightarrow \hat{\delta}_1(q_0, x) \in F_1$ sau $\hat{\delta}_2(q_2, x) \in F_2 \Leftrightarrow$

$\Leftrightarrow x \in L_1 \cup L_2$

2. Exerciţiu
3. Exerciţiu

<u>Corolar</u> Fie AFD $A = (Q, \Sigma, \delta, q_0, F)$. Atunci $\overline{L(A)} = \Sigma^* - L(A)$ poate fi recunoscut de un AFD.

<u>Demonstraţie</u>. Deoarece Σ^* poate fi recunoscut cu AFD $A' = (\{q_0'\}, \Sigma, \delta', q_0', \{q_0'\})$ definit prin $\delta'(q_0', a) = q_0'$, $\forall a \in \Sigma$, conform teoremei 2 rezultă că $\Sigma^* - L(A) = \overline{L(A)}$ este recunoscut de un AFD

<u>Direct</u> AFD $A'' = (Q, \Sigma, \delta, q_0, Q - F)$ recunoaşte $\Sigma^* - L(A)$.