~ Seminar 2 ~

➢ *Algoritm:* Transformarea AFN → AFD

Se dă un automat AFN. Se cere să se construiască un automat AFD echivalent (care să accepte același limbaj).

Idee: Atunci când în AFN dintr-o stare cu un anumit simbol avem ramificare către mai multe stări-destinație, în AFD vom grupa toate aceste stări-destinație într-o singură stare pentru a elimina ramificarea.

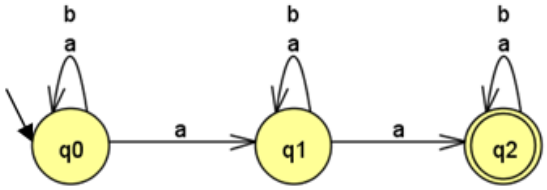
Algoritm: Pentru AFN-ul $(Q, \Sigma, q_0, F, \delta)$ construim AFD-ul $(Q', \Sigma, q_0, F', \delta')$ astfel:
- Stările AFD-ului vor reprezenta *submulțimi are mulțimii stărilor* AFN-ului ($Q' \subseteq \mathcal{P}(Q)$).
- Cele două automate vor avea același alfabet Σ și ***aceeași stare inițială*** q_0.
- Funcția de tranziție a AFD-ului este calculată astfel:

$$\delta'(R, x) = \bigcup_{q \in R} \delta(q, x), \qquad \forall R \in \mathcal{P}(Q), \forall x \in \Sigma$$

- Stările finale ale AFD-ului sunt mulțimile care ***conțin cel puțin o stare finală*** a AFN-ului $F' = \{R \mid R \in Q', R \cap F \neq \emptyset\}$.

Obs: La seminar vom folosi *metoda iterativă* de calcul pentru Q' (pornim din starea inițială și adăugăm stările AFD-ului pe măsură ce ajungem la ele calculând funcția de tranziție δ' pentru stările găsite anterior).

- ***Exemplu:*** Pentru AFN-ul următor construiți un AFD echivalent.

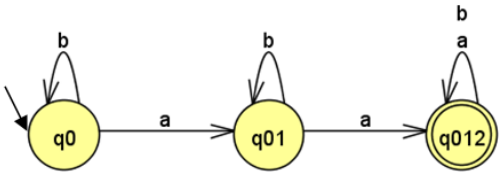


AFN

a) Completăm tabel_1 cu funcția de tranziție pentru AFN.
b) Completăm tabel_2 cu funcția de tranziție pentru AFD (pornim din starea inițială a AFN-ului și adăugăm pe rând stările obținute în interiorul tabel_2).
c) Desenăm graful pentru AFD conform tabel_2.

δ_AFN	a	b
q_0 init	$\{q_0, q_1\} = q_{01}$	$\{q_0\}$
q_1	$\{q_1, q_2\} = q_{12}$	$\{q_1\}$
$q_2 \in F$	$\{q_2\}$	$\{q_2\}$

δ_AFD	**a**	**b**
q_0 init	q_{01}	q_0
q_{01}	q_{012}	q_{01}
$q_{012} \in F$	q_{012}	q_{012}



AFD

➢ Verificare acceptare cuvânt de către AFD, AFN

Care este limbajul recunoscut de cele două automate din exemplul de mai sus?
Verificați (pentru AFD, apoi pentru AFN) dacă cuvintele **bba** și **babbaba** sunt acceptate sau respinse, folosind configurații.

→ AFD, cuv bba:
$(q_0, bba) \vdash^b (q_0, ba) \vdash^b (q_0, a) \vdash^a (q_{01}, \lambda)$, $q_{01} \notin F$ => bba respins

→ AFN, cuv bba:
$(q_0, bba) \vdash^b (q_0, ba) \vdash^b (q_0, a) \vdash^a \{(q_0, \lambda), (q_1, \lambda)\}$, $\{q_0, q_1\} \cap F = \emptyset$ => bba respins

→ AFD, cuv babbaba:
$(q_0, babbaba) \vdash^b (q_0, abbaba) \vdash^a (q_{01}, bbaba) \vdash^b (q_{01}, baba) \vdash^b (q_{01}, aba)$
$\vdash^a (q_{012}, ba) \vdash^b (q_{012}, a) \vdash^a (q_{012}, \lambda)$, $q_{012} \in F$ => babbaba acceptat

→ AFN, cuv babbaba:
$(q_0, babbaba) \vdash^b (q_0, abbaba) \vdash^a \{(q_0, bbaba), (q_1, bbaba)\}$
$\vdash^b \{(q_0, baba), (q_0, baba)\} \vdash^b \{(q_0, aba), (q_1, aba)\}$
$\vdash^a \{(q_0, ba), (q_1, ba), (q_2, ba)\} \vdash^b \{(q_0, a), (q_1, a), (q_2, a)\} \vdash^a \{(q_0, \lambda), (q_1, \lambda), (q_2, \lambda)\}$,
$\{q_0, q_1, q_2\} \cap F = \{q_2\} \neq \emptyset$ => babbaba acceptat

➢ *Automat Finit Nedeterminist cu λ–tranziţii*

AFN–λ = (Q, Σ, q_0, δ, F)
Q mulţimea de stări
Σ alfabetul de intrare („Sigma”)
$q_0 \in Q$ starea iniţială
$F \subseteq Q$ mulţimea de stări finale
$\delta : Q \times (\Sigma \cup \{\lambda\}) \to 2^Q$ funcţia de tranziţie („delta”)

Diferenţa faţă de AFN-ul simplu este că suntem într-o stare şi putem citi fie un caracter din alfabetul Σ, fie cuvântul vid λ. Deci se poate întâmpla să ajungem într-o stare nouă fără să fi citit nicio literă nouă din cuvântul de intrare, ci doar aplicând λ–tranziţii.

➢ λ–închiderea unei stări

Mulţimea de stări în care se poate ajunge plecând din starea q şi aplicând zero sau mai multe λ–tranziţii se numeşte „**λ–închiderea stării q**" şi se notează cu **<q>**.

Obs: Orice stare face parte din propria λ–închidere (pentru că $\delta(q,\lambda^0)=q$; practic putem presupune că orice stare are o λ–tranziţie *implicită* către ea însăşi).

$$<q>=\bigcup_{k\geq 0}\{r \mid r\in\hat{\delta}(q,\lambda^k)\}$$

$$<q>=\{q\}\cup\{q_i \mid q_i\in\hat{\delta}(q,\lambda^1)\}\cup\{q_{ij} \mid q_{ij}\in\hat{\delta}(q,\lambda^2)\}\cup\ldots$$

Observăm că mulţimile se pot calcula inductiv după puterea lui λ :

$$\{q_{ij} \mid q_{ij}\in\hat{\delta}(q,\lambda^2)\}=\{q_{ij} \mid q_i\in\hat{\delta}(q,\lambda^1), q_{ij}\in\delta(q_i,\lambda^1)\}.$$

Sau în general $\{r \mid r\in\delta(q,\lambda^k)\}=\{r \mid s\in\hat{\delta}(q,\lambda^{k-1}), r\in\delta(s,\lambda^1)\}$.

➢ Verificare acceptare cuvânt de către automat AFN–λ

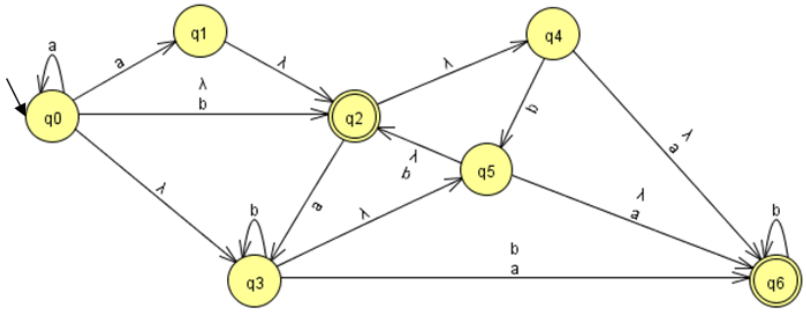
Pentru a verifica dacă un cuvânt este sau nu acceptat de un automat AFN–λ:
Se procedează analog ca în cazul AFN-ului, doar că înainte de a căuta toate stările posibile de continuare cu tranziţii cu simbolul curent, trebuie să facem λ–închiderea mulţimii curente de stări. Iar după ce a fost citit tot cuvântul de intrare, trebuie să facem o ultimă λ–închidere a stărilor curente, pentru a obţine mulţimea finală de stări în care poate ajunge automatul pentru cuvântul dat.

Obs: λ–închiderea unei mulţimi de stări este egală cu reuniunea λ–închiderilor acelor stări.

$$<\{q_{i1},q_{i2},\ldots,q_{in}\}>=<q_{i1}>\cup<q_{i2}>\cup\ldots\cup<q_{in}>$$

Obs: La AFN–λ, $\lambda\in L \Leftrightarrow <q_0>\cap F\neq\emptyset$ (cuvântul vid este acceptat de automat dacă și numai dacă λ-închiderea stării inițiale conține cel puțin o stare finală).

- ***Exemplu:*** Se dă următorul AFN- λ.



a) Calculăm λ–închiderile tuturor stărilor.

$<q_0>=\{q_0,q_2,q_3,q_4,q_5,q_6\}$
$<q_1>=\{q_1,q_2,q_4,q_6\}$
$<q_2>=\{q_2,q_4,q_6\}$
$<q_3>=\{q_3,q_5,q_2,q_6,q_4\}$
$<q_4>=\{q_4,q_6\}$
$<q_5>=\{q_5,q_2,q_6,q_4\}$
$<q_6>=\{q_6\}$

b) Verificăm dacă cuvântul **abbaa** este acceptat sau respins de acest AFN-λ, folosind configurații.
$(q_0, abbaa) \vdash^{\lambda^*} (q_{023456}, abbaa) \vdash^{a} (q_{0136}, bbaa) \vdash^{\lambda^*} (q_{0123456}, bbaa) \vdash^{b} (q_{2365}, baa)$
$\vdash^{\lambda^*} (q_{23456}, baa) \vdash^{b} (q_{3652}, aa) \vdash^{\lambda^*} (q_{23456}, aa) \vdash^{a} (q_{36}, a) \vdash^{\lambda^*} (q_{23456}, a)$
$\vdash^{a} (q_{36}, \lambda) \vdash^{\lambda^*} (q_{23456}, \lambda), \{q_2, q_3, q_4, q_5, q_6\} \cap F = \{q_2, q_6\} \neq \emptyset$ => abbaa acceptat

➢ *Algoritm:* Transformarea AFN-λ → AFN (***metoda 1***) *[vezi curs 3, pag 16 – 18]*

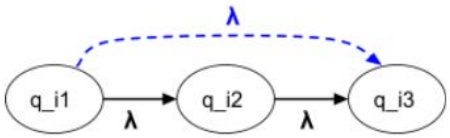
Se dă un automat AFN-λ. Se cere să se construiască un automat AFN echivalent.

Idee: Adăugăm tranziții cu litere din alfabet și stări finale astfel încât să simulăm comportamentul λ-tranzițiilor, pe care apoi le eliminăm.

→ Pas 1 („λ-completion"):

- Cât timp e posibil:

$$\forall q_{i_1}, q_{i_2}, q_{i_3} \in Q, \text{daca } \delta(\boldsymbol{q_{i_1}}, \boldsymbol{\lambda}) \ni \boldsymbol{q_{i_2}} \text{ si } (\boldsymbol{q_{i_2}}, \boldsymbol{\lambda}) \ni \boldsymbol{q_{i_3}} \Rightarrow (\boldsymbol{q_{i_1}}, \boldsymbol{\lambda}) \ni \boldsymbol{q_{i_3}}$$



Altfel spus, din fiecare stare **q** trebuie să adăugăm (dacă nu există deja) câte o λ-tranziție către fiecare stare **r** (cu $r \neq q$) din mulțimea <**q**> (λ-închiderea stării q).

(Adică dacă aveam de la starea q la starea r un drum format din două sau mai multe λ-tranziții, atunci trebuie să adaugăm o λ-tranziție directă de la q la r.)

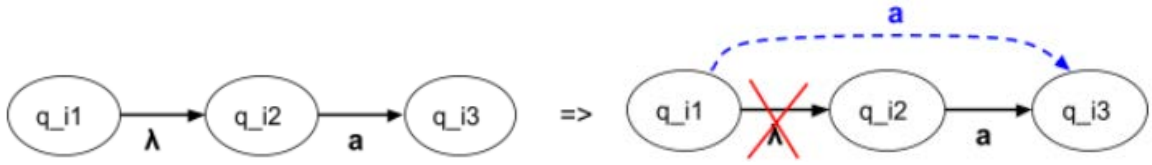
- Apoi trebuie să adăugăm la mulțimea stărilor finale acele stări din care cu λ ajungem într-o stare finală.

(Adică stările care conțineau în λ-închiderea lor cel puțin o stare finală vor deveni și ele stări finale.)

→ Pas 2 („λ-transition removal"):

- Adăugăm tranziții cu litere din alfabet care să înlocuiască efectul λ-tranzițiilor, astfel:

$$\forall q_{i_1}, q_{i_2}, q_{i_3} \in Q, \forall a \in \Sigma, \text{daca } \delta(\boldsymbol{q_{i_1}}, \boldsymbol{\lambda}) \ni \boldsymbol{q_{i_2}} \text{ si } (\boldsymbol{q_{i_2}}, \boldsymbol{a}) \ni \boldsymbol{q_{i_3}} \Rightarrow (\boldsymbol{q_{i_1}}, \boldsymbol{a}) \ni \boldsymbol{q_{i_3}}$$

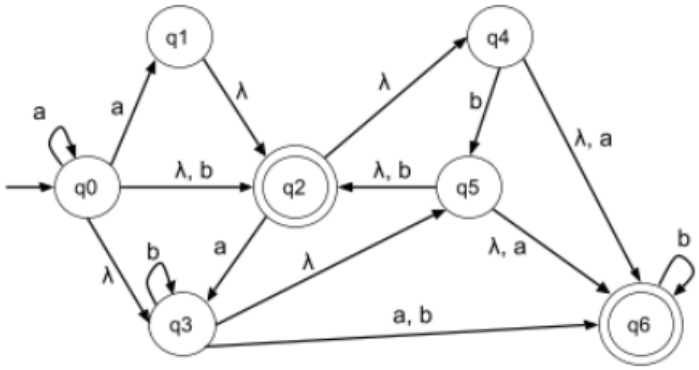


Adică dacă din starea **q** puteam ajunge în starea **r** citind ***λa*** ($\forall a \in \Sigma$), atunci adăugăm o tranziție cu litera **a** direct de la **q** la **r**.

- Apoi eliminăm toate λ-tranzițiile din automat.

- ***Exemplu:*** Se dă următorul AFN–λ.

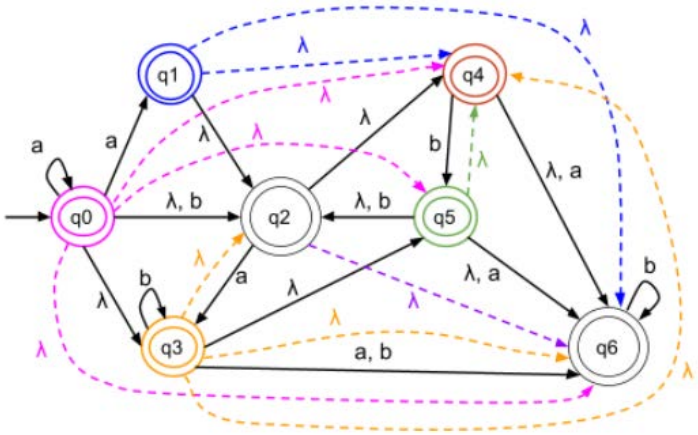
(Același de mai sus, pentru care calculasem deja λ-închiderile tuturor stărilor.)



→ Pas 1 („λ-completion”):

$< q_0 >= \{q_0, q_2, q_3, q_4, q_5, q_6\}$ => adaug λ de la q_0 spre q_4, q_5, q_6
$< q_1 >= \{q_1, q_2, q_4, q_6\}$ => adaug λ de la q_1 spre q_4, q_6
$< q_2 >= \{q_2, q_4, q_6\}$ => adaug λ de la q_2 spre q_6
$< q_3 >= \{q_3, q_5, q_2, q_6, q_4\}$ => adaug λ de la q_3 spre q_2, q_4, q_6
$< q_4 >= \{q_4, q_6\}$ => nu adaug nimic
$< q_5 >= \{q_5, q_2, q_4, q_6\}$ => adaug λ de la q_5 spre q_4
$< q_6 >= \{q_6\}$ => nu adaug nimic

Toate stările au în λ-închiderile lor cel puțin o stare finală, deci toate stările vor deveni finale.

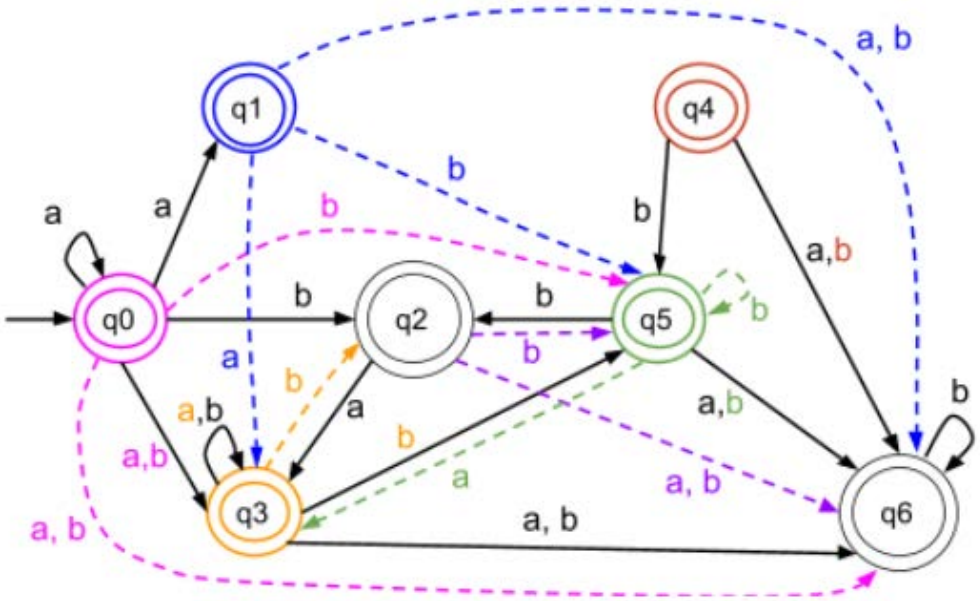


→ Pas 2 („λ-transition removal”):

Vom elimina: $\delta(q_{i1}, \lambda) \ni q_{i2}$	Avem: $\delta(q_{i2}, x) \ni q_{i3}, x \in \Sigma$	Adăugăm: $\delta(q_{i1}, x) \ni q_{i3}$
$\delta(q_0, \lambda) \ni q_2$	$\delta(q_2, a) \ni q_3$	$\delta(q_0, a) \ni q_3$
$\delta(q_0, \lambda) \ni q_3$	$\delta(q_3, a) \ni q_6$ $\delta(q_3, b) \supset \{q_3, q_6\}$	$\delta(q_0, a) \ni q_6$ $\delta(q_0, b) \supset \{q_3, q_6\}$
$\delta(q_0, \lambda) \ni q_4$	$\delta(q_4, a) \ni q_6$ $\delta(q_4, b) \ni q_5$	$\delta(q_0, a) \ni q_6$ $\delta(q_0, b) \ni q_5$
$\delta(q_0, \lambda) \ni q_5$	$\delta(q_5, a) \ni q_6$ $\delta(q_5, b) \ni q_2$	$\delta(q_0, a) \ni q_6$ $\delta(q_0, b) \ni q_2$
$\delta(q_0, \lambda) \ni q_6$	$\delta(q_6, b) \ni q_6$	$\delta(q_0, b) \ni q_6$

Vom elimina: $\delta(q_{i1}, \lambda) \ni q_{i2}$	Avem: $\delta(q_{i2}, x) \ni q_{i3}, x \in \Sigma$	Adăugăm: $\delta(q_{i1}, x) \ni q_{i3}$
$\delta(q_1, \lambda) \ni q_2$	$\delta(q_2, a) \ni q_3$	$\delta(q_1, a) \ni q_3$
$\delta(q_1, \lambda) \ni q_4$	$\delta(q_4, a) \ni q_6$ $\delta(q_4, b) \ni q_5$	$\delta(q_1, a) \ni q_6$ $\delta(q_1, b) \ni q_5$
$\delta(q_1, \lambda) \ni q_6$	$\delta(q_6, b) \ni q_6$	$\delta(q_1, b) \ni q_6$
$\delta(q_2, \lambda) \ni q_4$	$\delta(q_4, a) \ni q_6$ $\delta(q_4, b) \ni q_5$	$\delta(q_2, a) \ni q_6$ $\delta(q_2, b) \ni q_5$
$\delta(q_2, \lambda) \ni q_6$	$\delta(q_6, b) \ni q_6$	$\delta(q_2, b) \ni q_6$
$\delta(q_3, \lambda) \ni q_2$	$\delta(q_2, a) \ni q_3$	$\delta(q_3, a) \ni q_3$
$\delta(q_3, \lambda) \ni q_4$	$\delta(q_4, a) \ni q_6$ $\delta(q_4, b) \ni q_5$	$\delta(q_3, a) \ni q_6$ $\delta(q_3, b) \ni q_5$
$\delta(q_3, \lambda) \ni q_5$	$\delta(q_5, a) \ni q_6$ $\delta(q_5, b) \ni q_2$	$\delta(q_3, a) \ni q_6$ $\delta(q_3, b) \ni q_2$
$\delta(q_3, \lambda) \ni q_6$	$\delta(q_6, b) \ni q_6$	$\delta(q_3, b) \ni q_6$
$\delta(q_4, \lambda) \ni q_6$	$\delta(q_6, b) \ni q_6$	$\delta(q_4, b) \ni q_6$
$\delta(q_5, \lambda) \ni q_2$	$\delta(q_2, a) \ni q_3$	$\delta(q_5, a) \ni q_3$
$\delta(q_5, \lambda) \ni q_4$	$\delta(q_4, a) \ni q_6$ $\delta(q_4, b) \ni q_5$	$\delta(q_5, a) \ni q_6$ $\delta(q_5, b) \ni q_5$
$\delta(q_5, \lambda) \ni q_6$	$\delta(q_6, b) \ni q_6$	$\delta(q_5, b) \ni q_6$
$\delta(q_6, \lambda) = \emptyset$ =>		=> Nu avem ce adăuga din q_6.

Am obținut un AFN echivalent cu AFN-λ dat.
(Observăm că starea q_4 nu este accesibilă din starea inițială, deci ar putea fi eliminată împreună cu tranzițiile ei fără a afecta limbajul recunoscut de automat.)



➢ *Algoritm:* Transformarea AFN-λ → AFN / AFD (***metoda 2***) *[asemănător AFN → AFD, seminar]*

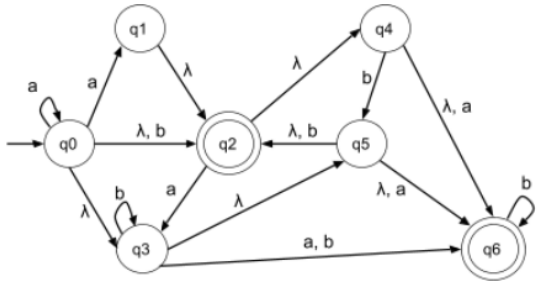
Idee: Dacă în AFN-λ din starea **q** citind $\boldsymbol{\lambda^* x \lambda^*}$ $(\forall x \in \Sigma)$ se ajunge în mulțimea de stări **R**, atunci în AFN din starea **q** citind litera **x** se va ajunge în mulțimea de stări **R**.

Starea inițială pentru AFN este aceeași ca la AFN-λ. Stările finale ale AFN-ului sunt cele ale căror λ-închideri în AFN-λ conțin cel puțin o stare finală.

Obs: Dacă dorim să obținem AFD, atunci starea inițială din AFD este λ-închiderea stării inițiale din AFN-λ. Stările finale ale AFD-ului sunt cele care conțin *cel puțin o stare* care în AFN-λ avea în λ-închidere cel puțin o stare finală.

- ***Exemplu:*** Se dă următorul AFN–λ.

(Același de mai sus, pentru care calculasem deja λ-închiderile tuturor stărilor.)



δ_AFN-λ	**a**	**b**	**λ**	**λ*** (λ-închiderea)
q0 init	{q0, q1}	{q2}	{q2, q3}	<q0> = {q0, q2, q3, q4, q5, q6}
q1	Ø	Ø	{q2}	<q1> = {q1, q2, q4, q6}
q2 in F	{q3}	Ø	{q4}	<q2> = {q2, q4, q6}
q3	{q6}	{q3, q6}	{q5}	<q3> = {q3, q5, q2, q6, q4}
q4	{q6}	{q5}	{q6}	<q4> = {q4, q6}
q5	{q6}	{q2}	{q2, q6}	<q5> = {q5, q2, q4, q6}
q6 in F	Ø	{q6}	Ø	<q6> = {q6}

în AFN-λ:	$\boldsymbol{\lambda^* a \lambda^*}$	$\boldsymbol{\lambda^* b \lambda^*}$
q0 init	$q_0 \xrightarrow{\lambda^*} q_{023456} \xrightarrow{a} q_{0136} \xrightarrow{\lambda^*} q_{0123456}$	$q_0 \xrightarrow{\lambda^*} q_{023456} \xrightarrow{b} q_{2356} \xrightarrow{\lambda^*} q_{23456}$
q1	$q_1 \xrightarrow{\lambda^*} q_{1246} \xrightarrow{a} q_{36} \xrightarrow{\lambda^*} q_{23456}$	$q_1 \xrightarrow{\lambda^*} q_{1246} \xrightarrow{b} q_{56} \xrightarrow{\lambda^*} q_{2456}$
q2 in F	$q_2 \xrightarrow{\lambda^*} q_{246} \xrightarrow{a} q_{36} \xrightarrow{\lambda^*} q_{23456}$	$q_2 \xrightarrow{\lambda^*} q_{246} \xrightarrow{b} q_{56} \xrightarrow{\lambda^*} q_{2456}$
q3	$q_3 \xrightarrow{\lambda^*} q_{23456} \xrightarrow{a} q_{36} \xrightarrow{\lambda^*} q_{23456}$	$q_3 \xrightarrow{\lambda^*} q_{23456} \xrightarrow{b} q_{2356} \xrightarrow{\lambda^*} q_{23456}$
q4	$q_4 \xrightarrow{\lambda^*} q_{46} \xrightarrow{a} q_6 \xrightarrow{\lambda^*} q_6$	$q_4 \xrightarrow{\lambda^*} q_{46} \xrightarrow{b} q_{56} \xrightarrow{\lambda^*} q_{2456}$
q5	$q_5 \xrightarrow{\lambda^*} q_{2456} \xrightarrow{a} q_{36} \xrightarrow{\lambda^*} q_{23456}$	$q_5 \xrightarrow{\lambda^*} q_{2456} \xrightarrow{b} q_{256} \xrightarrow{\lambda^*} q_{2456}$
q6 in F	$q_6 \xrightarrow{\lambda^*} q_6 \xrightarrow{a} \emptyset \xrightarrow{\lambda^*} \emptyset$	$q_6 \xrightarrow{\lambda^*} q_6 \xrightarrow{b} q_6 \xrightarrow{\lambda^*} q_6$

δ_AFN	***a***	***b***
q0 init, in F	$q_{0123456} = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$	$q_{23456} = \{q_2, q_3, q_4, q_5, q_6\}$
q1 in F	$q_{23456} = \{q_2, q_3, q_4, q_5, q_6\}$	$q_{2456} = \{q_2, q_4, q_5, q_6\}$
q2 in F	$q_{23456} = \{q_2, q_3, q_4, q_5, q_6\}$	$q_{2456} = \{q_2, q_4, q_5, q_6\}$
q3 in F	$q_{23456} = \{q_2, q_3, q_4, q_5, q_6\}$	$q_{23456} = \{q_2, q_3, q_4, q_5, q_6\}$
q4 in F	$\{q_6\}$	$q_{2456} = \{q_2, q_4, q_5, q_6\}$
q5 in F	$q_{23456} = \{q_2, q_3, q_4, q_5, q_6\}$	$q_{2456} = \{q_2, q_4, q_5, q_6\}$
q6 in F	Ø	$\{q_6\}$

δ_AFD	***a***	***b***
<q0> = q_{023456} init, in F	$q_{0123456}$	q_{23456}
$q_{0123456}$ in F	$q_{0123456}$	q_{23456}
q_{23456} in F	q_{23456}	q_{23456}

Am obținut un AFD echivalent cu automatele AFN și AFN-λ de mai sus.
Ce limbaj recunoaște acest AFD?