TUTORIAT LFA 1: INTRODUCERE ÎN LIMBAJE FORMALE

RADU COSTACHE, MARIA PREDA

1. Breviar Teoretic

Definiţie 1.1. *Un alfabet* Σ *este o mulţime de simboluri.*

Definiţie 1.2. *Un cuvânt este o succesiune finită de simboluri dintr-un alfabet.*

Definiţie 1.3. *Fie* w *un cuvânt. Vom nota cu* $|w|$ *lungimea cuvântului, iar cu* $|w|_c$ *numărul de aparitii ale simbolului* c *în cuvântul* w*.*

Definiţie 1.4. *Vom defini cuvântul vid, pe care îl vom nota cu* λ*, având* $|\lambda| = 0$*.*

Definiţie 1.5. *Un limbaj formal este o mulţime de cuvinte finite asupra unui alfabet.*

Exemplu 1.6. *Limbajul formal* $L = \{a, ab, c\}$ *este un limbaj formal pentru alfabetul* $\Sigma = \{a, b, c\}$*.*

1.1. Operaţii cu cuvinte.

- **Concatenarea**: Concatenarea a două cuvinte reprezină cuvântul obţinut prin lipirea celor două. Utilizăm simbolul "$\cdot$".

  Exemplu 1.7. $ana \cdot maria = anamaria$

  Observaţie 1.8. *Concatenarea a două cuvinte* ***nu este comutativă****.*

- **Ridicarea la putere**: Reprezintă concatenarea cuvântului de la bază cu el însuşi, de numărul de ori dat de exponent. Cu alte cuvinte, având un cuvânt c şi $n \in \mathbb{N}$. Avem:

$$c^n = \underbrace{ccc\ldots c}_{n}$$

  Observaţie 1.9. *Pentru orice cuvânt* c *avem* $c^0 = \lambda$*.*

  Observaţie 1.10. *Pentru orice* $n \in \mathbb{N}$*, avem* $\lambda^n = \lambda$

- **Stelarea**: Reprezintă limbajul tuturor puterilor lui c. Astfel, $c^* = \{c^p \mid p \in \mathbb{N}\}$. Vom defini şi limbajul c^+, ce cuprinde puterile cu exponent pozitiv, adică $c^+ = \{c^p \mid p \in \mathbb{N}^*\}$.

1.2. Operaţii cu limbaje.

- **Reuniune**: Reuniunea a două limbaje reprezintă un nou limbaj, care conţine toate cuvintele distincte din cele două limbaje i.e. *reuniunea de mulţimi.*
  $L = L_1 \cup L_2 = \{w \mid w \in L_1 \textit{ sau } w \in L_2\}$

  Exemplu 1.11.

$$L_1 = \{radu, maria, fmi\}$$
$$L_2 = \{fmi, unibuc\}$$

$$L_1 \cup L_2 = \{radu, maria, fmi, unibuc\}$$

- **Intersecţie**: Intersecţia a doua limbaje este un nou limbaj, care contine toate cuvinte care se afla in ambele limbaje i.e. *intersecţia de mulţimi.*
  $L = L_1 \cap L_2 = \{w \mid w \in L_1 \textit{ si } w \in L_2\}$

Exemplu 1.12.

$$L_1 = \{radu, maria, fmi\}$$
$$L_2 = \{fmi, unibuc\}$$

$$L_1 \cap L_2 = \{fmi\}$$

- **Diferenţa**: Diferenţa a două limbaje este constituită de cuvintele care se află în primul limbaj, dar nu şi în al doilea i.e. *Diferenţa de mulţimi*
  $L = L_1 - L_2 = \{w \mid w \in L_1 \ si \ w \notin L_2\}$

Exemplu 1.13.

$$L_1 = \{radu, maria, fmi\}$$
$$L_2 = \{fmi, unibuc\}$$

$$L_1 - L_2 = \{radu, maria\}$$

- **Concatenare** Concatenarea a doua limbaje reprezintă un limbaj format din reuniunea "lipirii", pe rand, a fiecarui cuvânt din primul limbaj cu fiecare cuvânt din al doilea limbaj. Aceasta este notată cu "$\cdot$", dar simbolul se poate omite.
  $L = L_1 \cdot L_2 = \{w_1w_2 \mid w_1 \in L_1 \ si \ w_2 \in L_2\}$

Exemplu 1.14.

$$L_1 = \{fmi, unibuc\}$$
$$L_2 = \{Bucuresti, Romania\}$$

$$\begin{aligned} L &= L_1 \cdot L_2 \\ &= L_1L_2 \\ &= \{fmiBucuresti,\ unibucBucuresti,\ fmiRomania,\ unibucRomania\} \end{aligned}$$

Observaţie 1.15. *Cuvântul "fmiBucuresti" din limbajul L este diferit de cuvântul "Bucurestifmi", acesta fiind, de fapt, cuvânt al limbajului $L_2 \cdot L_1$. Aşadar, concatenarea nu este comutativa: $L_1L_2 \neq L_2L_1$.*

- **Ridicarea la putere**: Ridicarea la puterea n a unui limbaj reprezinta multimea formata din cuvinte obtinute prin concatenarea a n cuvinte din limbaj (acestea pot sa se repete, nu trebuie sa fie toate n distincte).
  $L_1^0 = \{\lambda\}$
  $L_2^n = \{w_1w_2 \cdots w_n | w_i \in L_2, i = \overline{1, n}\}$

Exemplu 1.16. $L = \{LFA, bc\}$
$L^3 = \{LFALFALFA,\ LFALFAbc,\ LFAbcLFA,\ bcLFALFA,\ LFAbcbc,\ bcLFAbc,\ bcbcLFA,\ bcbcbc\}$

:

- **Stelarea**: Stelarea unui limbaj reprezinta reuniunea limbajelor formate prin ridicarea la fiecare putere naturala a limbajului initial.
  $L^* = \{\lambda\} \cup L \cup L^2 \cup L^3 \cup \cdots$
  Operatia L^+ este asemanatoare, dar nu mai apare în reuniune şi limabjul ridicat la puterea 0.
  $L^+ = L \cup L^2 \cup L^3 \cup \cdots$

Observaţie 1.17. *Dacă $\lambda \in L$, atunci $\lambda \in L^+$.*

1.3. **Automate Finite.**

Definiţie 1.18 (Automate)**.** *Automatele sunt maşinării abstracte care acceptă cuvinte dintr-un limbaj formal.*

Definiţie 1.19 (DFA)**.** *Un automat finit determinist (deterministic finite automata sau DFA) este un tuplu* $(Q, \Sigma, q_0, F, \delta)$ *cu semnificaţia:*

- Q *este o mulţime de stări. Stările se vor reprezenta sub forma unor noduri într-un graf orientat. (Figura 1)*
- Σ *este alfabetul automatului (totalitatea simbolurilor pe care le va utiliza automatul).*
- q_0 *este starea iniţială a automatului. În reprezentare, va fi nodul de unde vom începe acceptarea unui cuvânt.* $q_0 \in Q$*. (Figura 2c)*
- F *este mulţimea stărilor finale ale automatului.* F *este o submulţime a mulţimii stărilor. Dacă din starea iniţială reuşim să ajungem intr-una dintre aceste stări, înseamnăcăam obţinut un cuvânt acceptat de automat. (Figura 3)*
- δ *se numeşte funcţie de tranziţie. Avem* $\delta : Q \times \Sigma \to Q$*. Practic,* δ *ne indică în ce stare ajungem dintr-o stare, dacă parcurgem un anumit simbol.*

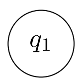


Figura 1. Stare

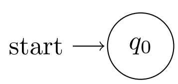


Figura 2. Stare iniţială

Figura 3. Stare finală

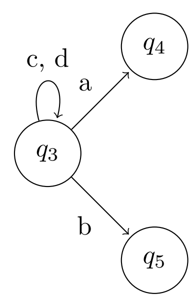
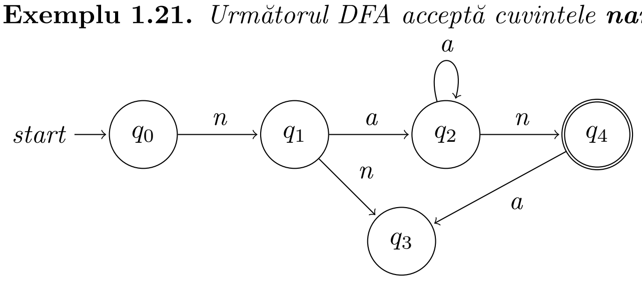


Figura 4. Tranziţii DFA

Definiţie 1.20 (Acceptarea cuvântului de DFA)**.** *Spunem că un DFA acceptă un cuvânt c dacă succesiunea de tranziţii începând din starea iniţială şi luând simbolurile din c în ordine, ajungem într-o stare finală.*

Exemplu 1.21. *Următorul DFA acceptă cuvintele* ***nan*** *şi* ***naaan****, dar nu acceptă cuvântul* ***nn****, sau* ***nana****:*



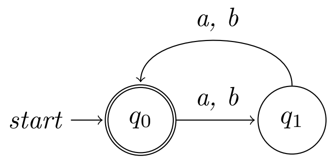
Demonstraţie. Vom verifica tranziţiile pentru fiecare dintre cuvinte până când ajungem la λ:

- $(q_0, nan) \vdash^n (q_1, an) \vdash^a (q_2, n) \vdash^n (q_4, \lambda), q_4 \in F \Rightarrow nan \in L$
- $(q_0, naaan) \vdash^n (q_1, aaan) \vdash^a (q_2, aan) \vdash^a (q_2, an) \vdash^a (q_2, n) \vdash^n (q_4, \lambda), q_4 \in F \Rightarrow nan \in L$
- $(q_0, nn) \vdash^n (q_1, n) \vdash^n (q_3, \lambda), q_3 \notin F \Rightarrow nn \notin L$
- $(q_0, nana) \vdash^n (q_1, ana) \vdash^a (q_2, na) \vdash^n (q_4, a), q_4 \vdash^a (q_3, \lambda), q_3 \notin F \Rightarrow nan \notin L$

□

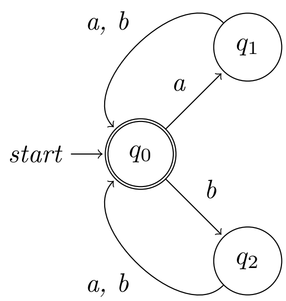
Observaţie 1.22. *Din definiţia DFA-ului se observă imediat că din fiecare stare pot ajunge cu un anumit simbol într-o singură altă stare.*

Exemplu 1.23. *Următorul DFA acceptă toate cuvintele formate cu alfabetul* $\Sigma = \{a, b\}$*, care au lungime pară i.e. modelează limbajul* $L = \{w \in \{a, b\}^* \mid |w|\ par\}$*:*



Observaţie 1.24. *Pot exista mai multe DFA-uri care sa accepte acelaşi limbaj.*

Exemplu 1.25. *Următorul DFA acceptă acelaşi limbaj ca cel din Exemplul 1.23, cele două fiind echivalente:*



Definiţie 1.26 (NFA)**.** *Un automat finit nedeterminist (nondeterministic finite automata sau NFA) este un tuplu* $(Q, \Sigma, q_0, F, \delta)$*.* Q, Σ, q_0 *şi* F *au aceeaşi semnificaţie ca în cazul DFA-urilor, dar pentru un NFA avem*

$$\delta : Q \times \Sigma \to \mathcal{P}(Q)$$

Observaţie 1.27. *Semnificaţia funcţiei de tranziţie a NFA-ului constă în faptul că putem avea mai multe tranziţii cu acelaşi simbol din aceeaşi stare (Figura 5)*

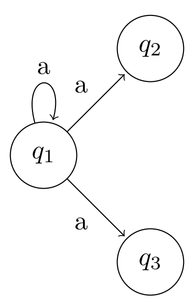
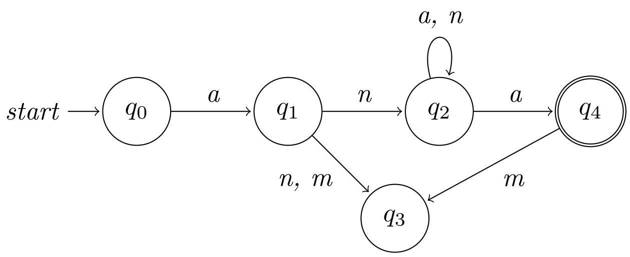


Figura 5. Tranziţii NFA. Nedeterminism

Observaţie 1.28 (Acceptare cuvânt NFA)**.** *Un cuvânt c este acceptat de un NFA daca exista o succesiune de tranziţii din starea iniţială care să parcurga în ordine toate simbolurile cuvântului c si sa ajungă intr-o stare finală.*

Exemplu 1.29. *Următorul NFA acceptă cuvintele* ***ana*** *şi* ***annana****, dar nu acceptăcuvintele* ***an*** *şi* ***anam****:*



Demonstraţie. Vom proceda similar acceptării de către DFA, dar vom lua în calcul toate stările în care vom putea ajunge pentru un anumit simbol:

- $\{(q_0, ana)\} \vdash^a \{(q_1, na)\} \vdash^n \{(q_2, a)\} \vdash^a \{(q_2, \lambda), (q_4, \lambda)\}, q_4 \in F \Rightarrow ana \in L$
- $\{(q_0, annana)\} \vdash^a \{(q_1, nnana)\} \vdash^n \{(q_2, nana)\} \vdash^n \{(q_2, ana)\} \vdash^a \{(q_2, na), (q_4, na)\} \vdash^n \{(q_2, a)\} \vdash^a \{(q_2, \lambda), (q_4, \lambda)\}, q_4 \in F \Rightarrow annana \in L$
- $\{(q_0, an)\} \vdash^a \{(q_1, n)\} \vdash^n \{(q_2, \lambda)\}, q_2 \notin F \Rightarrow an \notin L$
- $\{(q_0, anam)\} \vdash^a \{(q_1, nam)\} \vdash^n \{(q_2, am)\} \vdash^a \{(q_2, m), (q_4, m)\} \vdash^m \{(q_3, \lambda)\} q_3 \notin F \Rightarrow anam \notin L$

□

Definiţie 1.30 (λ-NFA)**.** *Un λ-NFA este un NFA care admite λ tranziţii i.e tranziţii care nu consumă simboluri din cuvânt. (Figura 6)*

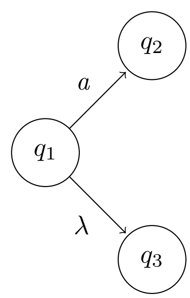
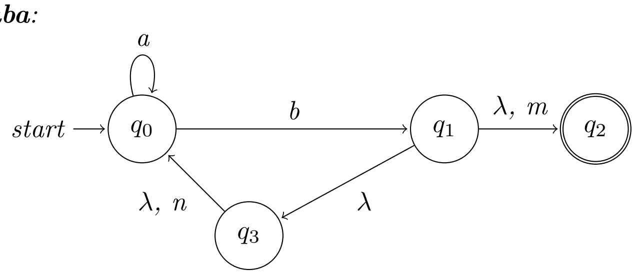


FIGURA 6. λ tranziţie

Exemplu 1.31. *Următorul λ-NFA acceptă cuvintele* ***aabaabm****,* ***bnbm****, şi* ***aab*** *dar nu acceptă cuvântul* ***aaba****:*



Demonstraţie. În plus faţă de NFA, trebuie să mai luăm în calcul şi λ-tranziţiile. Astfel, în loc să parcurgem doar stările în care putem ajunge doar consumând o literă din cuvânt, vom alterna o literă, cu o succesiune λ^* (oricâte tranziţii λ succesive). Pentru a realiza mai uşor acceptarea, vom începe prin a calcula λ-închiderile stărilor i. e. mulţimile de stări în care putem ajunge trecând doar prin λ-tranziţii. **Este important ca prima şi ultima tranziţie să fie** λ^*.

Avem următoarele λ-închideri:

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

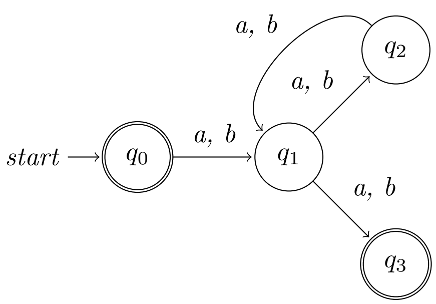
Aplicăm algoritmul:

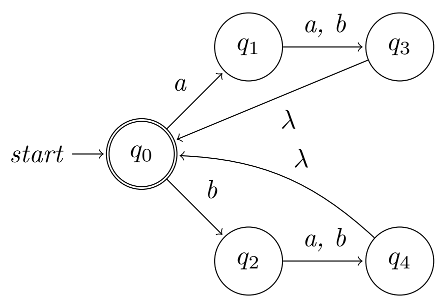
- $\{(q_0, aabaabm)\} \vdash^{\lambda^*} \{(q_0, aabaabm)\} \vdash^a \{(q_0, abaabm)\} \vdash^{\lambda^*} \{(q_0, abaabm)\} \vdash^a \{(q_0, baabm)\} \vdash^{\lambda^*} \{(q_0, baabm)\} \vdash^b \{(q_1, aabm)\} \vdash^{\lambda^*} \{(q_1, aabm), (q_2, aabm), (q_3, aabm), (q_0, aabm)\} \vdash^a \{(q_0, abm)\} \vdash^{\lambda^*} \{(q_0, abm)\} \vdash^a \{(q_0, bm)\} \vdash^{\lambda^*} \{(q_0, bm)\} \vdash^b \{(q_1, m)\} \vdash^{\lambda^*} \{(q_1, m), (q_2, m), (q_3, m), (q_0, m)\} \vdash^m \{(q_2, \lambda)\} \vdash^{\lambda^*} \{(q_2, \lambda)\}, q_2 \in F \Rightarrow aabaabm \in L$
- $\{(q_0, aaba)\} \vdash^{\lambda^*} \{(q_0, aaba)\} \vdash^a \{(q_0, aba)\} \vdash^{\lambda^*} \{(q_0, aba)\} \vdash^a \{(q_0, ba)\} \vdash^{\lambda^*} \{(q_0, ba)\} \vdash^b \{(q_1, a)\} \vdash^{\lambda^*} \{(q_1, a), (q_0, a), (q_3, a), (q_2, a)\} \vdash^a \{(q_0, \lambda)\} \vdash^{\lambda^*} \{(q_0, \lambda)\}, q_0 \notin F \Rightarrow aaba \notin L$

□

Teoremă 1.32. *Daca $\mathcal{L}_{DFA}$, $\mathcal{L}_{NFA}$ şi $\mathcal{L}_{\lambda-NFA}$ sunt mulţmile de limbaje pentru care se pot realiza DFA-uri, respectiv NFA-uri şi avem $\mathcal{L}_{DFA} = \mathcal{L}_{NFA} = \mathcal{L}_{\lambda-NFA}$.*

Exemplu 1.33. *Pentru limbajul din Exemplul 1.23 se pot construi şi următoarele automate NFA şi $\lambda-NFA$:*





2. Exerciţii

(1) Determinaţi urmatoarele limbaje:

(a) $L_a = L_{a1} \cdot L_{a2}$, unde:
$L_{a1} = \{a, ab, c\}$
$L_{a2} = \{ab, be, e\}$

Soluţie:
$L_a = \{aab, abe, ae, abab, abbe, cab, cbe, ce\}$

(b) $L_b = L_{b1} \cap \Sigma^*$

Soluţie:
$L_b = L_{b1}$

(c) $L_c = L_{c1} \cup L_{c1}^*$

Soluţie:
$L_c = L_{c1}^*$

(d) $L_d = L_{d1}^*$, unde $L_{d1} = \{a, b, c\}$

Soluţie:
$L_d = L_{d1}^* = L_{d1}^0 \cup L_{d1}^1 \cup L_{d1}^2 \cup \cdots = \{\lambda\} \cup \{a, b, c\} \cup \{aa, ab, ac, ba, bb, bc, ca, cb, cc\} \cup \cdots =$
$= \{\lambda, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, \cdots\}$

(e) $L_e = L_{e1}^+$, unde $L_{e1} = \{a, b, c\}$

Soluţie:
$L_e = L_{e1}^+ = L_{e1}^1 \cup L_{e1}^2 \cup \cdots = \{a, b, c\} \cup \{aa, ab, ac, ba, bb, bc, ca, cb, cc\} \cup \cdots =$

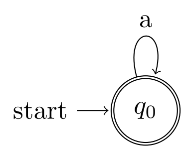
$= \{a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, \cdots\}$

(f) $L_f = L_{f1}^+$, unde $L_{f1} = \{\lambda, a, b\}$

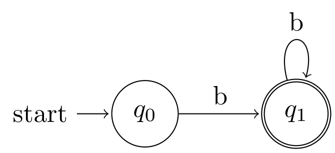
Soluţie:
$L_f = L_{f1}^+ = L_{f1}^1 \cup L_{f1}^2 \cup \cdots = \{\lambda, a, b\} \cup \{a, b, aa, ab, ba, bb\} \cup \cdots = \{\lambda, a, b, aa, ab, ba, bb, \cdots\}$

(2) Desenaţi DFA-urile pentru următoarele limbaje:

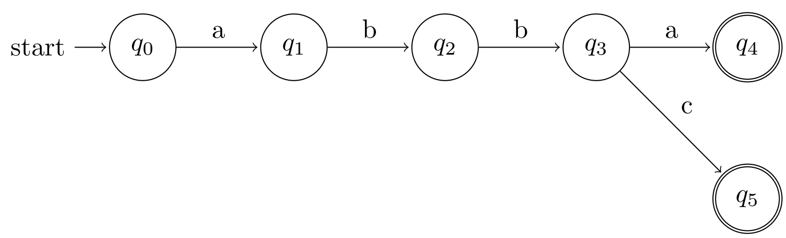
(a) $L_a = \{a\}^*$
Soluţie:



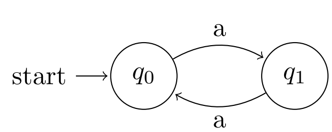
(b) $L_b = \{b\}^+$
Soluţie:



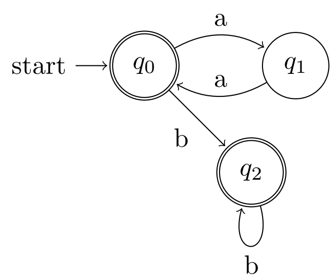
(c) $L_c = \{abba, abbc\}$
Soluţie:



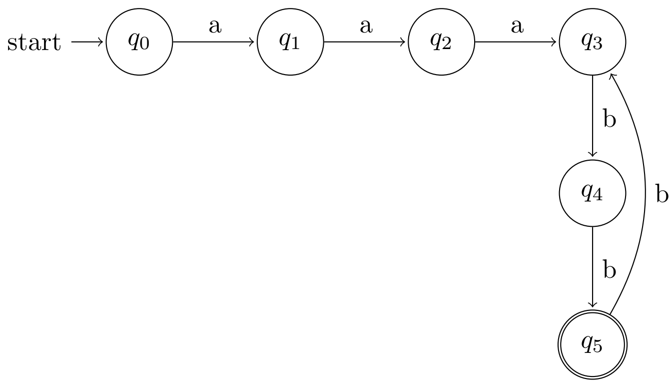
(d) $L_d = \{a^p b^t \mid p, t \in \mathbb{N},\ p\ par\}$
Soluţie:
Literele nu comută în cadrul unui cuvânt, deci trebuie sa ne asigurăm în prima parte a DFA-ului că sunt acceptate doar şiruri de a de lungime pară.



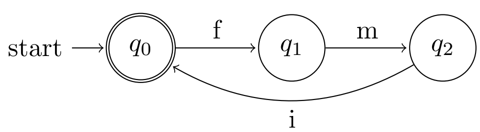
Pentru a fi in starea q_0, trebuie mereu sa fi procesat un număr par de caractere a (acest număr poate fi şi 0).
Acum trebuie să ne asigurăm că, după un număr par de a, poate accepta orice număr de caractere b. Pentru că p şi t nu sunt nenule:



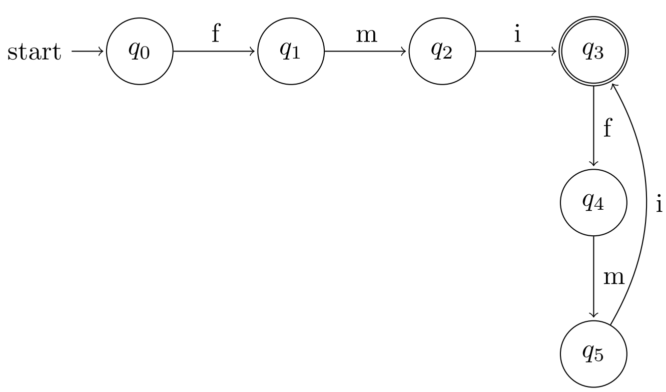
(e) $L_e = \{a^3b^k \mid k \in \mathbb{N},\ k = M_3 + 2\}$
Soluţie:



(f) $L_f = \{fmi\}^*$
Soluţie:

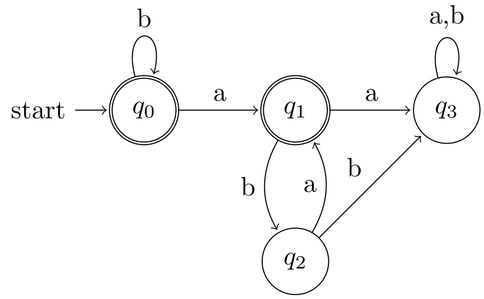


(g) $L_h = \{fmi\}^+$
Soluţie:



(3) Pentru fiecare subpunct, verificaţi dacă sunt acceptate de automate cuvintele.

(a) Cuvintele de verificat: aaba, bbbaba.



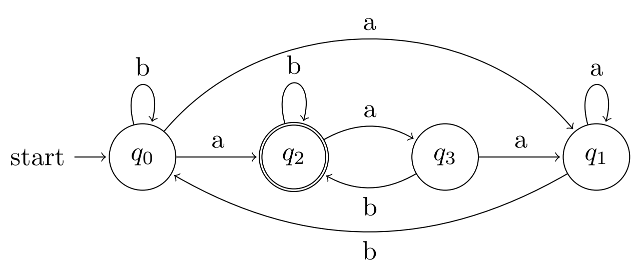
Avem tuplul $(Q, \Sigma, q_0, F, \delta)$, care reprezinta DFA-ul nostru, unde:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{a, b\}$
- $F = \{q_0, q_1\}$

Notăm cu L limbajul acceptat de automatul de mai sus.

- $(q_0, aaba) \vdash^a (q_1, aba) \vdash^a (q_3, ba) \vdash^b (q_3, b) \vdash^b (q_3, \lambda), q_3 \notin F \Rightarrow aaba \notin L$
- $(q_0, bbbaba) \vdash^b (q_0, bbaba) \vdash^b (q_0, baba) \vdash^b (q_0, aba) \vdash^a (q_1, ba) \vdash^b (q_2, a) \vdash^a (q_1, \lambda), q_2 \in F \Rightarrow bbbaba \in L$

(b) Cuvintele de verificat: babbab, aababaa.



- $(q_0, babbab) \vdash^b (q_0, abbab) \vdash^a (q_{12}, bbab) \vdash^b (q_{02}, bab) \vdash^b (q_{02}, ab) \vdash^a (q_{123}, b) \vdash^b (q_{02}, \lambda), q_2 \in F \Rightarrow babbab \in L$
- $(q_0, aababaa) \vdash^a (q_{12}, ababaa) \vdash^a (q_{13}, babaa) \vdash^b (q_{02}, abaa) \vdash^a (q_{123}, baa) \vdash^b (q_{02}, aa) \vdash^a (q_{123}, a) \vdash^a (q_{13}, \lambda), q_1 \notin F, q_3 \notin F \Rightarrow aababaa \notin L$