3 Регуларни јазици и конечни автомати

Во овој дел ќе дефинираме специјална класа јазици, ќе ги викаме регуларни јазици, ќе дефинираме специјални зборови со чија помош ќе можеме наполно да ги определуваме регуларните јазици и нив ќе ги викаме регуларни изрази, потоа ќе дефинираме конечни автомати, детерминистички и недетерминистички, и ќе покажеме дека регуларни јазици се точно оние препознаени со конечни автомати.

3.1 Регуларни јазици

3.1.1 Регуларни изрази

Во одделот за основни поими воведовме поим за подзбор, како и прво, второ, $\cdots$,n- то појавување на даден подзбор во даден збор; исто така во примерот 7 од поглавието за нормални алгоритми воведовме поим за инверзија на збор w која и натаму ќе ја означуваме со w^R. Притоа $(uv)^R = v^R u^R$. Да забележиме дека ако $\Sigma \neq \emptyset$ е конечно множество, тогаш Σ^* е преброиво, што е покажано во курсот "математичка логика и алгебра" од втора година. Ако $\Sigma = \emptyset$, тогаш $\Sigma^* = \{\lambda\}$.

Нека Σ е дадена азбука. За секое подмножество $L \subseteq \Sigma^*$ велиме дека е *јазик* во Σ.

Значи, Σ, $\emptyset$ и Σ^* се јазици во Σ.

Секој конечен јазик може да се зададе со наведување на сите негови зборови. Од интерес се јазици кои не се конечни; нив можеме да ги зададеме со некое својство кое го задоволуваат зборовите што се негови елементи. Така, на пример, еден бесконечен јазик во дадена азбука Σ е и јазикот $L = \{w; w \in \Sigma^*, w = w^R\}$.

Бидејќи јазиците над дадена азбука Σ се множества, можеме да воведеме операции меѓу јазици преку операции на множества: унија, пресек, разлика, комплемент (во однос на Σ^*). Покрај овие операции, можат да се воведат и други, имено:

- *Конкатенација* на јазици.

  Нека L_1 и L_2 се два јазика над азбуката Σ, т.е. $L_1, L_2 \subseteq \Sigma^*$. Тогаш конкатенацијата $L = L_1 \circ L_2$ (или само $L = L_1 L_2$) се дефинира со :

  $$L = \{w | w = x \circ y, x \in L_1, y \in L_2\}, \qquad (7)$$

  каде што со $x \circ y$ е означена конкатенацијата на зборовите x и y.

- Унарна операција над јазик $L \subseteq \Sigma^*$ е и таканаречената операција *затворач* или *Клиниева ѕвезда*. Јазикот добиен од L со оваа операција ќе го означуваме со L^*. Тој се дефинира на следниов начин:

  $$L^* = \{w \in \Sigma^* | (\exists k \geq 0) w = w_1 \circ w_2 \circ \cdots \circ w_k, \ \& \ w_1, w_2, \cdots, w_k \in L\}. \qquad (8)$$

Ако $L \neq \emptyset$, тогаш ја употребуваме ознаката L^+ за јазикот LL^*, т.е.

$$L^+ = \{w | (\exists k > 0, \ \& \ w_1, \cdots, w_k \in L) w = w_1 \circ w_2 \circ \cdots \circ w_k\}. \quad (9)$$

Да наведеме некои својства на дефинираните операции.

Својство 3.1.1 *За секои $L, L_1, L_2, L_3 \subseteq \Sigma^*$, точни се следниве тврдења:*

(а) $(L_1 L_2) L_3 = L_1 (L_2 L_3)$;

(б) *Ако $L_1 \subseteq L_2$, тогаш*

- (1) $L_1 L_3 \subseteq L_2 L_3$;
- (2) $L_3 L_1 \subseteq L_3 L_2$;
- (3) ${L_1}^* \subseteq {L_2}^*$.

(в)
- (1) $(L_1 \cup L_2) L_3 = L_1 L_3 \cup L_2 L_3$;
- (2) $L_1 (L_2 \cup L_3) = L_1 L_2 \cup L_1 L_3$.

(г)
- (1) $L_1 L_3 \setminus L_2 L_3 \subseteq (L_1 \setminus L_2) L_3$;
- (2) $L_3 L_1 \setminus L_3 L_2 \subseteq L_3 (L_1 \setminus L_2)$.

(д) $L^* L^* = L^*, \ (L^*)^* = L^*$.

(ѓ)
- (1) $L_1 \cup L_2)^* = ({L_1}^* \cup {L_2}^*)^*$;
- (2) $(L_1 \cap L_2)^* \subseteq ({L_1}^* \cap {L_2}^*)^*$;
- (3) ${L_1}^* \setminus {L_2}^* \subseteq (L_1 \setminus L_2)^*$.

Централна улога во теоријата на пресметливост игра претставувањето на јазици со конечни спецификации. Конечното претставување на јазици е специјално важно кај бесконечните јазици.

Бидејќи азбуката Σ е конечно множество, а Σ^* преброиво, множеството од сите јазици над Σ е непреброиво, (нив ги има 2^{Σ^*}), па само пребороиво многу бесконечни јазици ќе можат да се зададат со конечно претставување. Покрај тоа, идејата е едно конечно претставување да определува точно еден јазик, што претставува доста јако ограничување.

Ќе наведеме повеќе начини на конечни претставувања на јазици, при што ќе уочиме дека некои овозможуваат претставувања на јазици кои не се претставуваат со претходно наведените методи. На тој начин ќе добиеме и класификација на јазиците кои дозволуваат конечно претставување.

На почетокот ќе разгледуваме изрази–низи симболи, кои обезбедуваат апарат за градење јазици со помош на операциите: пресек, унија, комплемент, конкатенација и Клиниева ѕвезда.

Нека е зададена азбука Σ и нека $\Sigma_1 = \Sigma \cup \{), (, \emptyset, \cup, *\}$. Специјални зборови над азбуката Σ_1 ќе велиме дека се *регуларни изрази* и ќе ги дефинираме индуктивно на следниов начин:

(i) $\emptyset$ и секој симбол од Σ се регуларни изрази;

(ii) Ако α и β се регуларни изрази, тогаш и $(\alpha\beta)$, $(\alpha \cup \beta)$ и α^* се регуларни изрази;

(iii) Еден израз од азбуката Σ_1 е регуларен ако и само ако може да се добие со конечна примена на (i) и (ii).

Со секој регуларен израз е претставен еден јазик L над Σ. Имено, дефинираме пресликување L од множеството регуларни изрази во множеството јазици над Σ, така што за секој регуларен израз α, $L(\alpha)$ ќе биде јазикот претставен со регуларниот израз α. L е дефинирано на следниов начин:

(i) $L(\emptyset) = \emptyset$ и $L(a) = \{a\}$, за секој $a \in \Sigma$;

(ii) Ако α и β се регуларни изрази, тогаш
$L((\alpha\beta)) = L(\alpha)L(\beta)$;
$L((\alpha \cup \beta)) = L(\alpha) \cup L(\beta)$;
$L(\alpha^*) = L(\alpha)^*$.

Пример:
Да го определиме јазикот $L(((a \cup b)^* a))$.

$$L(((a \cup b)^* a)) = L((a \cup b)^*)L(a) = L(a \cup b)^* \{a\} =$$
$$= (L(a) \cup L(b))^* \{a\} = (\{a\} \cup \{b\})^* \{a\} =$$
$$= \{a, b\}^* \{a\} =$$
$$= \{w \in \{a, b\}^* | w \text{ завршува со } a\}.$$

Пример:
Да го определиме јазикот $L(c^*(a \cup (bc^*))^*)$.

$$L(c^*(a \cup (bc^*))^*) = L(c)^* L(a \cup (bc^*))^* =$$
$$= \{c\}^* (L(a) \cup L(bc^*))^* =$$
$$= \{c\}^* (\{a\} \cup (\{b\}\{c\}^*))^* =$$
$$= \{w \in \{a, b, c\}^* | w = c^n(a^m(bc^k))^r \vee w = c^n a^m \vee w = c^n(a^m(bc^k))^r, n, m, k, r \geq 0\}.$$

Јасно е дека ако $w \in L$, тогаш w не содржи подзбор ac. Нека w е збор што не содржи подзбор ac. Ако a се појавува во w, тогаш, бидејќи w не содржи подзбор ac, по a може да следува или a или b, што значи дека $w \in L$.

Бидејќи и унијата и конкатенацијата се асоцијативни, во регуларните изрази може да се отфрлат некои загради. Покрај тоа, еден јазик претставен со еден регуларен израз α може да биде претставен и со други регуларни изрази. Исто така, бидејќи не интересираат јазиците, често пати нема да правиме разлика помеѓу регуларен израз и јазикот претставен со тој регуларен израз.

Јасно е дека регуларните изрази не можат да ги претставуваат сите јазици над дадена азбука Σ. Да ги определиме својствата на јазиците кои што можат да се претстават со регуларен израз.

3.1.2 Регуларни јазици

Ќе дефинираме специјална класа јазици (*регуларни јазици*) , а потоа ќе докажеме дека тие се точно оние јазици што можат да се претстават со регуларни изрази.

Класата *регуларни јазици* $\mathcal{R}$ над азбуката Σ е подмножество од 2^{Σ^*}, со следниве својства:

(i) $\emptyset \in \mathcal{R}$; $a \in \Sigma \Rightarrow \{a\} \in \mathcal{R}$;

(ii) $L_1, L_2 \in \mathcal{R} \Rightarrow L_1 \circ L_2, L_1 \cup L_2, L_1^* \in \mathcal{R}$;

(iii) Ако $\mathcal{S}$ е класа јазици над Σ што ги задоволува условите (i) и (ii), тогаш $\mathcal{R} \subseteq \mathcal{S}$.

Забелешка Поради (iii) од горната дефиниција, класата регуларни јазици еднозначно определена.

Од дефиницијата на регуларен израз и јазик определен од регуларен израз како и од самата дефиниција на регуларен јазик следува следново својство:

Својство 3.1.2 *Еден јазик L над азбуката Σ е регуларен ако и само ако постои регуларен израз α над Σ, таков што $L = L(\alpha)$.* $\square$

Со регуларни изрази може да се претстави релативно "мала" класа јазици. Постојат релативно "едноставни" јазици, како на пример:

$$\{a^n b^n | n \geq 1\}, \text{каде што } \Sigma = \{a, b\},$$

кои не можат да се претстават со регуларни изрази. Ова тврдење ќе го покажеме подолу, откако ќе изградиме соодветен апарат за определување кога еден јазик не е регуларен.

За претставување јазици можеме да го употребиме и начинот на претставување множества (подмножества од Σ^*) како решение на исказна функција над Σ^*. На пример:

$$L = \{w \in \Sigma^* | (\exists x, y, z \in N \setminus \{0\}) x^{|w|} + y^{|w|} = z^{|w|}\}$$

но, од гледна точка на пресметливост, наидуваме на тешкотии. Идеата е да се определи јазик L над Σ таков што да постои точно определена "постапка" (алгоритам) којашто по конечен број чекори ќе даде одговор на прашањето дали збор од Σ^* е збор од L.

Алгоритмот којшто е специјално составен за некој јазик L и проверува дали $w \in \Sigma^*$ е елемент од L се вика *алгоритам за препознавање на јазик.*

Наредниот чекор е дефинирање на алгоритми за препознавање на јазици.

3.1.3 Задачи

1. Да се докаже дека:

   (а) $(w^R)^R = w$, за кој било збор w од дадена азбука Σ;

   (б) v е подзбор од w ако и само ако v^R е подзбор од w^R;

   (в) $(\forall i \in N)(w^i)^R = (w^R)^i$, каде што w^i е зборот што се состои од i пати допишан зборот w еден по друг, т.е. е i-ти степен од w.

2. Да се докаже дека:

   (а) Ако $a \neq b$, тогаш

   $$\{a, b\}^* = \{a\}^*(\{b\}\{a\}^*)^*.$$

   (б) Ако Σ е азбука и $L_1, L_2 \subseteq \Sigma^*$, при што $\lambda \in L_1, \lambda \in L_2$, тогаш $\Sigma^* = (L_1 \sigma^* L_2)^* = (L_1 \Sigma L_2)^*$;

   (в) За секој јазик L точни се равенствата:

   $$\emptyset L = L\emptyset = \emptyset.$$

3. Нека Σ е азбука. Тогаш $(\Sigma^*; \circ)$ е моноид со кратење, каде што $\circ$ е операцијата конкатенација.

4. Кој јазик е претставен со регуларниот израз:

   $$(((a^*a)b) \cup b).$$

5. Да се упростат следниве регуларни изрази:

(а) $\emptyset^* \cup a^* \cup b^*$;

(б) $((a^*b^*)^*(b^*a^*)^*)^*$;

(в) $(a^*b)^* \cup (b^*a)^*$;

(г) $(a \cup b)^*a(a \cup b)^*$.

6. Нека $\Sigma = \{a, b\}$. Да се запишат регуларни изрази што ги претставуваат следниве јазици:

(а) $\{w \in \Sigma^* | w$ не содржи повеќе од три појавувања на $a\}$;

(б) $\{w \in \Sigma^* |$бројот на појавувања на a во w е делив со три$\}$;

(в) $\{w \in \Sigma^* |$подзборот aaa се појавува во w точно еднаш$\}$;

(г) $\{w \in \Sigma^* |$секое a во w е претходено и следено со $b\}$;

(д) $\{w \in \Sigma^* | abab$ е подзбор од $w\}$;

(ѓ) $\{w \in \Sigma^* |$ни aa ни bb не се подзборови од $w\}$;

(е) $\{w \in \Sigma^* | ab$ и ba се подзборови од $w\}$.

7. Еден регуларен израз е во дисјунктивна нормална форма ако е запишан во форма

$$w_1 \cup w_2 \cup \cdots \cup w_k,$$

каде што симболот $\cup$ не се појавува во $w_i, i = 1, 2, \cdots, k$.

Да се докаже дека за секој регуларен израз постои еквивалентен регуларен израз (регуларен израз којшто го определува истиот јазик) во дисјунктивна нормална форма.

3.2 Конечни автомати

Во овој оддел ќе разгледаме некои модели на *конечен автомат.* Тој има "централен процесор" со фиксиран конечен капацитет што зависи од основниот проблем. Покрај ова тој нема меморија. Влезот е низа симболи запишани на лента. Излезот само дава информација дали автоматот го препознава зборот од влезната лента или не, т.е. тоа е само направа за препознавање јазици. Всушност, ќе разгледаме два вида конечни автомати–детерминистички и недетерминистички, и ќе покажеме дека класата јазици препознаена од секој од овие два вида автомати е точно класата регуларни јазици.

3.2.1 Детерминистички конечни автомати

Конечниот детерминистички автомат се состои од *влезна лента* поделена на полиња со по еден симбол впишан во секое поле. Главниот дел од автоматот е "црна кутија" која може да се наоѓа во една од конечно многу *внатрешни состојби.* Црната кутија се вика *конечна контрола* и со помош на *подвижна глава* може да препознае кој симбол е запишан во ќелијата на лентата.

Во почетокот главата е поставена да ја гледа првата најлева ќелија од лентата и е поставена во *почетна состојба.* Во правилни интервали главата чита симбол од лентата и преминува во нова состојба, која зависи од состојбата во која се наоѓа и симболот што го чита, и потоа се подвижува за едно поле на десно. По конечно чекори автоматот стигнува до последната ќелија од лентата на која е запишан влезниот збор и преминува во нова состојба. Ако оваа последна состојба е една од множеството *завршни состојби* се смета дека автоматот го *препознава зборот*, во спротивно, се смета дека тој *не го препознава зборот. Јазикот препознаен од овој конечен автомат* е множеството зборови што тој ги препознава.

Да дадеме формална дефиниција на детерминистички конечен автомат.

Детерминистички конечен автомат е подредена петорка

$$M = (K, \Sigma, \delta, s, F),$$

каде што:

- K е конечно множество состојби,
- Σ е надворешна азбука,
- $s \in K$ е почетна состојба,
- $F \subseteq K$ е множество завршни состојби,

- δ е *функција на премин*, т.е. пресликување од $K \times \Sigma$ во K.

Функцијата на премин еднозначно ја определува состојбата во која ќе премине конечниот автомат M ако се наоѓа во состојба q и го чита симболот a. Таа може да биде зададена на различни начини:

- со табилица (заради конечноста на K и Σ),

  или

- графички, со помош на дијаграм.

Секоја функција на премин може да се прошири до пресликување $\delta^1 : K \times \Sigma^* \to K$ на следниов начин:
$\delta^1(q, a) = \delta(q, a)$, за секој $(q, a) \in K \times \Sigma$.
Ако $q \in K, a \in \Sigma$ и $w = aw' \in \Sigma^*$, тогаш

$$\delta^1(q, aw') = \delta^1(\delta(q, a), w') = \delta^1(q', w'),$$

каде што $\delta(q, a) = q'$.

Ако е зададен зборот $w = a_1 a_2 \cdots a_k \in \Sigma^*$, тогаш секој чекор на работата на конечниот автомат при влез w може да се опише со состојбата во која е тој во моментот кој се разгледува и подзборот што се уште не е прочитан. Овој начин на опишување на состојбата на конечен автомат се вика *конфигурација*. Конфигурација на конечен автомат се состои од низа што почнува со една од состојбите на машината (онаа во која машината се наоѓа во тој момент) и преостанатиот непрочитан подзбор. Така, низата $q_r a_i a_{i+1} \cdots a_n$ означува дека во дадениот момент конечниот автомат е во состојба q_r, го гледа симболот a_i, и останува да се прочита подзборот $a_i a_{i+1} \cdots a_n$

Пример 1.:

Нека $K = \{q_0, q_1\}$, $\Sigma = \{a, b\}$, $s = q_0$ и $F = \{q_0\}$, а

q	σ	$\delta(q\sigma)$
q_0	a	q_0
q_0	b	q_1
q_1	a	q_1
q_1	b	q_0

Ако $w = aabba$, тогаш $q_0 aabba$ е почетната конфигурација на лентата. По читање на буквата a машината останува во состојбата q_0 и преминува едно поле на десно, на кое повторно е запишана буквата a, што значи, дека машината и натаму останува во состојбата q_0 и ја чита наредната буква b.

Кога машината е во состојба q_0 и ја чита буквата b, според функцијата на премин, таа преминува во состојба q_1 и преминува едно поле на десно. Ако машината е во состојба q_1 и ја чита буквата b, тогаш преминува во состојба q_0 и едно поле на десно, во кое е запишана буквата a. Според функцијата на премин, машината останува во состојбата q_0, којашто е завршна, а на лентата нема запишано други симболи, што значи дека машината го препознава зборот w.

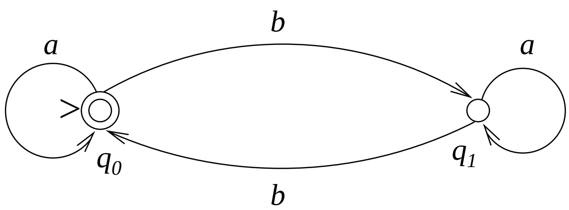
Описот на работата на машината од овој пример со помош на конфигурации може да се запише пократко на следниов начин:

$$q_0aabba \vdash_M q_0abba \vdash_M q_0bba \vdash_M q_1ba \vdash_M q_0a \vdash_M q_0,$$

каде што $\vdash_M$ означува директен премин на машината M од една конфигурација во наредната. Доколку однапред се знае за кој конечен автомат станува збор, тогаш индексот M на $\vdash_M$ може да се изостави. Ако, пак, сакаме со симболи да прикажеме што се добива при работа на зададен конечен автомат почнувајќи од дадена конфигурација по одреден број чекори извршени од машината, тогаш ја користиме ознаката $\models_M$, при што може да го запишеме и бројот на чекорите кои се изведени од почетната разгледувана конфигурација до последната. Така, на пример, ако ја земеме точно почетната конфигурација на автоматот од овој пример, и сакаме да прикажеме дека зборот $w = aabba$ е препознаен од автоматот, тогаш, знаејќи дека q_0 е и завршна состојба, пишуваме $q_0aabba \models_M q_0$.

Во суштина $\vdash$ е релација во множеството конфигурации на даден конечен автомат, додека $\models$ е транзитивното затворање на $\vdash$.

Функцијата на премин може да се прикаже и со дијаграм. Секоја состојба се означува со кругче, завршните се заокружени со поголемо кругче, додека почетната е означена со симболот $>$. За секоја буква од азбуката Σ и дадени состојби $q_1, q_2 \in K$ е поставена стрелка од q_1 до q_2 означена со буквата a, ако $\delta(q_1, a) = q_2$. На тој начин функцијата на премин на автоматот даден во овој пример е прикажана со следнава сл.3.1.



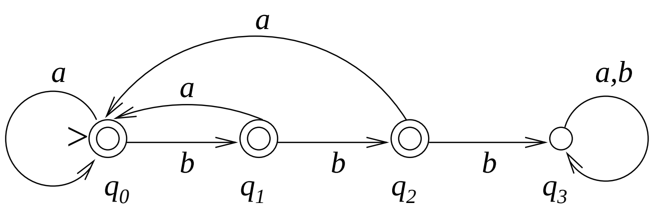
Слика 3.1:

Пример 2.:
Нека $K = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a, b\}$, $s = q_0$, а $F = \{q_0, q_1, q_2\}$.

Функцијата на премин е зададена со следнава таблица:

q	σ	$\delta(q\sigma)$
q_0	a	q_0
q_0	b	q_1
q_1	a	q_0
q_1	b	q_2
q_2	a	q_0
q_2	b	q_3
q_3	a	q_3
q_3	b	q_3

или со дијаграмот прикажан на сл.3.2.



Слика 3.2:

Ако е зададен зборот $w = abbba$, тогаш добиваме:

$$q_0abbba \vdash q_0bbba \vdash q_1bba \vdash q_2ba \vdash q_3a \vdash q_3,$$

што значи дека автоматот не го препознава дадениот збор w.

За јазикот L над азбуката Σ велиме дека е *препознаен од автоматот* M, го означуваме со $L(M)$, ако тој се состои точно од оние зборови над Σ што тој ги препознава, т.е.

$$L(M) = \{w | w \in \Sigma^*, sw \models q,\ q \in F\}.$$

Јазикот $L(M)$ препознаен од автоматот од Примерот 1 е

$$L(M) = a^*ba^*ba^*,$$

додека во Примерот 2,

$$L(M) = \{w | w \in \{a,b\}^*, w \text{ не содржи подзбор } b^n, n \geq 3\}.$$

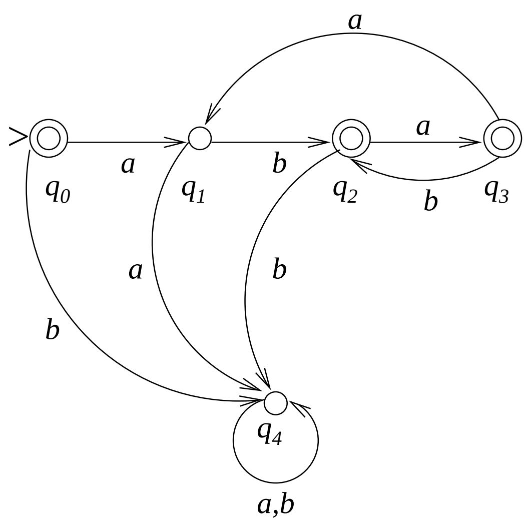
3.2.2 Недетерминистички конечни автомати

Во овој дел ќе разгледуваме малку поинаков вид автомати кај кои ќе дозволуваме од дадена состојба при читање даден симбол автоматот да може да премине во една од повеќе состојби по сопствен избор. Значи нема да постои точно определен премин од состојба при даден симбол (детерминираност на работата на автоматот), туку тој ќе има повеќе можности за избор на премин (недетерминираност на автоматот). Иако во пракса сметачките машини го немаат ова својство, се покажува погодно за работа со нив и поедноставно прикажување на работата на ваков автомат.

Да забележиме дека секој детерминистички конечен автомат е недетерминистички. Во наредниот дел ќе покажеме дека за секој недетерминистички конечен автомат M постои детерминистички конечен автомат M' којшто го препознава истиот јазик како и M, т.е таков што $L(M) = L(M')$.

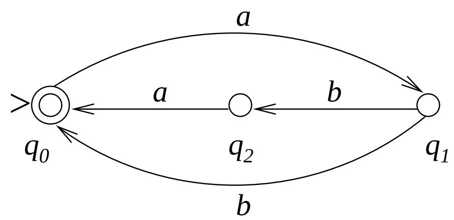
Пример 1.:

Нека $L = (ab \cup aba)^*$. Тогаш детерминистичкиот автомат прикажан на сл.3.3. го препознава јазикот L.



Слика 3.3:

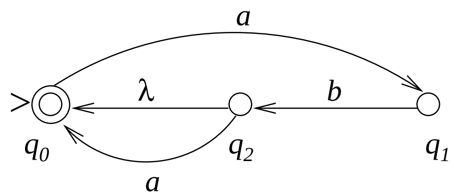
Меѓутоа, и автоматот прикажан на сл.3.4. којшто е недетерминистички, го препознава истиот јазик. Во овој случај, ако влезниот збор е aba автоматот може од q_0 да премине во состојбата q_1, па во q_2 и назад во q_0, но може да го избере и "погрешниот" пат q_0, q_1, q_0, q_1 и да заврши



Слика 3.4:

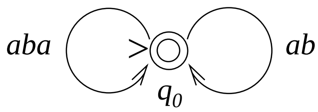
со работа во состојба којашто не е завршна. Ова не е битно, бидејќи доволно е за машината да го препознае влезниот збор да постои барем еден начин (пат) при "читање" на зборот од почетната состојба да стигне во завршна.

Некогаш е погодно стрелка да биде означена и со празниот збор λ, па горниот недетерминистички автомат да го има и следниов, поедноставен, графички приказ: Може да се дозволи уште едно обопш-



Слика 3.5:

тување: стрелките да можат да се означуваат и со цели зборови, при што уште повеќе се поедноставува дијаграмот на конечен недетерминистички автомат. Ако се искористи ова обопштување, автоматот од нашиов пример може да се прикаже и со следниов дијаграм:



Слика 3.6:

Да дадеме формална дефиниција на недетерминистички конечен автомат.

Недетерминистички конечен автомат е подредена петорка

$$M = (K, \Sigma, \Delta, s, F),$$

каде што:

- K е конечно множество состојби,
- Σ е конечна азбука од влезни симболи,
- $s \in K$ е почетна состојба,
- $F \subseteq K$ е множество завршни состојби, а
- Δ е *релација за премин* и е конечно подмножество од $K \times \Sigma^* \times K$.

Значењето на $(q, u, p) \in \Delta$ е дека ако M е во состојба q, тогаш тој може да го "прочита" зборот u и да премине во состојба p. Ова графички се прикажува со $\circ_q \xrightarrow{u} \circ_p$.

Секоја тројка $(q, u, p) \in \Delta$ се вика *премин во* M.

Конфигурација се дефинира по аналогија на дефиницијата кај детерминистички конечни автомати. За поедноставно опишување на работата на недетерминистички конечен автомат и во овој случај се дефинира релација $\vdash_M$ во $K \times \Sigma^*$ со:

$$(q, w) \vdash_M (q', w')$$

ако и само ако постои $u \in \Sigma^*$ такво што $uw' = w$ и $(q, u, q') \in \Delta$. Нејзиното транзитивно проширување се означува со $\models_M$.

Зборот $w \in \Sigma^*$ е *препознаен* од M ако и само ако постои состојба $q \in F$ таква што $(s, w) \models_M (q, \lambda)$.

Јазикот $L(M)$ *препознаен од автоматот* M се состои од сите зборови над Σ препознаени од M.

Лема 3.2.1 *Нека $M = (K, \Sigma, \Delta, s, F)$ е недетерминистички конечен автомат, нека $q, r \in K$ и $x, y \in \Sigma^*$. Ако за некој $p \in K$, $(q, x) \models_M (p, \lambda)$ и $(p, y) \models_M (r, \lambda)$, тогаш $(q, xy) \models_M (r, \lambda)$.*

Доказ. Нека $(q, x) \models_M (p, \lambda)$ и $(p, y) \models_M (r, \lambda)$. Тогаш постојат

$$n \geq 0, q_0, q_1, \cdots, q_n \in K$$

и

$$x_0, \cdots, x_n \in \Sigma^*,$$

такви што

$$(q, x) = (q_0, x_0) \vdash_M (q_1, x_1) \vdash_M \cdots \vdash_M (q_n, x_n) = (p, \lambda).$$

Тврдиме дека за $i = 0, 1, \cdots, n-1$, $(q_i, x_i y) \vdash_M (q_{i+1}, x_{i+1} y)$.

Бидејќи $(q_i, x_i) \vdash_M (q_{i+1}, x_{i+1})$, од дефиницијата на $\vdash_M$ постои $u_i \in \Sigma^*$ такво што $x_i = u_i x_{i+1}$ и $(q_i, u_i, q_{i+1}) \in \Delta$. Истотака, $x_i y = u_i x_{i+1} y$, па, од дефиницијата на $\vdash_M$ добиваме дека важи $(q_i, x_i y) \vdash_M (q_{i+1}, x_{i+1} y)$.

Значи,

$$(q, xy) = (q_0, x_0 y) \vdash_M (q_1, x_1 y) \vdash_M \cdots \vdash_M (q_n, x_n y) = (p, y),$$

па $(q, xy) \models_M (p, y)$. Но, од $(p, y) \models_M (r, \lambda)$ и фактот дека $\models_M$ е транзитивна релација, добиваме $(q, xy) \models_M (r, \lambda)$. □

3.2.3 Еквивалентност на детерминистички и недетерминистички конечни автомати

За автоматите M_1 и M_2 велиме дека се *еквивалентни* ако $L(M_1) = L(M_2)$.

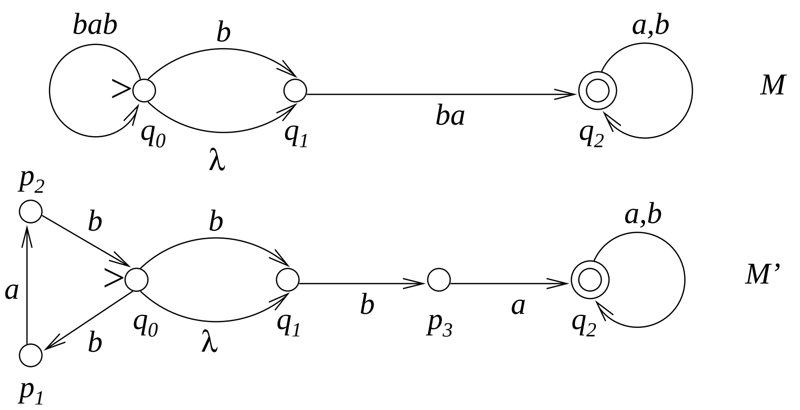
Теорема 3.2.1 *За секој недетерминистички конечен автомат постои еквивалентен со него детерминистички конечен автомат.*

Доказ: Нека $M = (K, \Sigma, \Delta, s, F)$ е недетерминистички конечен автомат. За секој премин $(q, u, q') \in \Delta$, таков што $|u| > 1$ и $u = a_1 a_2 \cdots a_k$ дефинираме нови незавршни состојби $p_1, \cdots, p_{k-1}$; во Δ тројката (q, u, q') ја заменуваме со тројките (p_{i-1}, a_i, p_i), $i = 1, \cdots, k$, каде што $p_0 = q$, $p_k = q'$. Со тоа добиваме нова релација на премин Δ'. Јасно е дека новиот автомат $M' = (K', \Sigma, \Delta', s, F)$ е недетерминистички автомат еквивалентен со M и $|u| \leq 1$ за секој премин (q, u, q') од M'. Оваа постапка е илустрирана со сл.3.7

Сега ќе конструираме детерминистички конечен автомат $M'' = (K'', \Sigma, \delta'', s'', F'')$ еквивалентен на M', што ќе биде доволно за да се докаже теоремата. Идејата е недетерминистичкиот автомат да се разгледува како автомат кој во секој момент не е во единствена состојба туку во множество состојби. Значи, ако M' има пет состојби $\{q_0, q_1, \cdots, q_4\}$ и, после читање на влезен симбол може да биде во состојба q_0, q_2, q_3 но не и во q_1 или q_4, за состојба може да се разгледува множеството $\{q_0, q_2, q_3\}$, наместо поединечен член од тоа множество. Ако следниот влезен симбол може да го доведе автоматот M' од q_0 во q_1 или q_2, од q_2 во q_0 и од q_3 во q_2, тогаш за наредна состојба на M' може да се смета множеството $\{q_0, q_1, q_2\}$.

Оваа конструкција ќе ја формализираме, за да го добиеме детерминистичкиот автомат M''.

Множеството состојби на M'' ќе биде $2^{K'}$, т.е. булеанот од множеството состојби на M'. Множеството завршни состојби ќе се состои од оние подмножества од K' кои содржат барем една завршна состојба од M'. Дефиницијата на функцијата на премин на M'' ќе биде малку



Слика 3.7:

покомплицирана. Основната идеја е преминот во M'' при читање на влезен симбол $a \in \Sigma$ да го имитира преминот од M' за влезниот симбол a, следено со одреден број премини од M' за кои не е прочитан никаков влезен симбол. За формализација на оваа идеја неопходна ни е специјална дефиниција.

За секоја состојба $q \in K'$, нека $E(q)$ биде множеството состојби од M' кои можат да се достигнат од q без читање на некаков влез. Значи:

$$E(q) = \{p \in K' | (q, \lambda) \models_{M'} (p, \lambda)\}.$$

Ако M' има премин од една во друга состојба без примање на влез, тогаш тој не зависи од самиот влез. Значи, друг начин на дефинирање на $E(q)$ би бил да се избере произволна низа $w \in \Sigma^*$ и да се запише:

$$E(q) = \{p \in K' | (q, w) \models_{M'} (p, w)\}.$$

Сега дефинираме конечен автомат

$$M'' = (K'', \Sigma, \delta'', s'', F''),$$

каде што

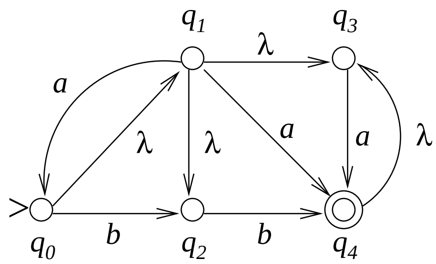
$$K'' = 2^{K'}$$

$$s'' = E(s)$$

$$F'' = \{Q \subseteq K' | Q \cap F' \neq \emptyset\}$$

и за секој $Q \subseteq K'$ и секој симбол $a \in \Sigma$,

$$\delta''(Q, a) = \bigcup \{E(p) | p \in K' \text{и} (\exists q \in Q)(q, a, p) \in \Delta'\}.$$



Слика 3.8:

На пример, ако M' е автоматот прикажан на сл.3.8. тогаш $s'' = E(q_0) = \{q_0, q_1, q_2, q_3\}$. Бидејќи $(q_1, a, q_0) \in \Delta'$ и $(q_1, a, q_4) \in \Delta'$, следува дека

$$\delta''(\{q_1\}, a) = E(q_0) \cup E(q_4) = \{q_0, q_1, q_2, q_3, q_4\}.$$

Слично се проверува дека од $(q_0, b, q_2) \in \Delta'$ и $(q_2, b, q_4) \in \Delta'$ следува дека

$$\delta''(\{q_0, q_2\}, b) = E(q_2) \cup E(q_4) = \{q_2, q_3, q_4\}.$$

Останува да се докаже дека M'' е детерминистички конечен автомат еквивалентен со M'. Доказот дека M'' е детерминистички следува од тоа што по конструкција δ'' е пресликување. Да забележиме дека $\emptyset \in K$", па е возможно $\delta''(Q, a) = \emptyset$, за некој $Q \in K''$ и $a \in \Sigma$. Исто така $\delta''(\emptyset, a) = \emptyset$, за секој $a \in \Sigma$.

Останува да се докаже дека за секоја низа $w \in \Sigma^*$, и секои $q, p \in K'$, точно е тврдењето

$$(q, w) \models_{M'} (p, \lambda) \text{ ако и само ако} (E(q), w) \models_{M''} (P, \lambda)$$

за некое множество P што го содржи p. Од тука следува тврдењето на теоремата. Имено, нека $w \in \Sigma^*$ е произволна низа. Тогаш $w \in L(M')$ ако и само ако $(s, w) \models_{M'} (f', \lambda)$ за некој $f' \in F'$ (по дефиниција на $L(M')$), а ова е точно ако и само ако $(E(s), w) \models_{M''} (Q, \lambda)$ за некој Q што го содржи f', ако и само ако $(s'', w) \models_{M''} (Q, \lambda)$ за некој $Q \in F''$. Последниов услов е дефиницијата за $w \in L(M'')$.

Горното тврдење ќе го докажеме со индукција по $|w|$.

Основа на индукцијата. За $|w| = 0$, т.е. за $w = \lambda$ треба да покажеме дека

$$(q, \lambda) \models_{M'} (p, \lambda) \Leftrightarrow (E(q), \lambda) \models_{M''} (P, \lambda)$$

за некое множество P што го содржи p. Првото тврдење е еквивалентно со тврдењето дека $p \in E(q)$. Бидејќи M'' е детерминистички и притоа мора да чита влезен симбол при секој чекор, второто тврдење е еквивалентно со $P = E(q)$ и P го содржи p, т.е. $p \in E(q)$.

Индуктивна претпоставка. Нека за некој $k \geq 0$ тврдењето е точно за секој збор w со должина еднаква или помала од k.

Чекор на индукцијата. Ќе го докажеме тврдењето за произволен збор w со должина $k+1$. Нека $w = va$, каде што $a \in \Sigma$ и $v \in \Sigma^*$.

Нека, прво, $(q, w) \models_{M'} (p, \lambda)$. Тогаш постојат состојби r_1 и r_2 такви што

$$(q, va) \models_{M'} (r_1, a) \vdash_{M'} (r_2, \lambda) \models_{M'} (p, \lambda).$$

Значи, M' ја достигнува состојбата p од состојбата q преку некој број чекори при кои влезот v е прочитан, следен со еден чекор при кој буквата a е прочитана, следен со некој број чекори при кој не е прочитан никаков влез. Бидејќи $(q, va) \models_{M'} (r_1, a)$, тогаш $(q, v) \models_{M'} (r_1, \lambda)$, па бидејќи $|v| = k$, според индуктивната претпоставка $(E(q), v) \models_{M''} (R_1, \lambda)$, за некој R_1 што го содржи r_1. Од друга страна, бидејќи $(r_1, a) \vdash_{M'} (r_2, \lambda)$, тројката $(r_1, a, r_2) \in \Delta'$; тогаш, според конструкцијата на M'' имаме $E(r_2) \subseteq \delta''(R_1, a)$. Но, бидејќи $(r_2, \lambda) \models_{M'} (p, \lambda)$, $p \in \delta''(R_1, a)$. Значи, $(R_1, a) \vdash_{M''} (P, \lambda)$, за некој P што го содржи p и $(E(q), va) \models_{M''} (R_1, a) \vdash_{M''} (P, \lambda)$.

Од друга страна, нека $(E(q), va) \models_{M''} (R_1, a) \vdash_{M''} (P, \lambda)$, за некој P, таков што $p \in P$ и R_1, таков што $\delta''(R_1, a) = P$. Според дефиницијата на δ'' сега $\delta''(R_1, a)$ е унија од сите множества $E(r_2)$, каде што за некоја состојба $r_1 \in R_1$, (r_1, a, r_2) е премин во M'. Бидејќи $p \in P = \delta''(R_1, a)$, постои определено r_2 такво што $p \in E(r_2)$ и за некое $r_1 \in R_1, (r_1, a, r_2)$ е премин во M'. Тогаш, од дефиницијата на $E(r_2)$, $(r_2, \lambda) \models_{M'} (p, \lambda)$. Исто така, од индуктивната претпоставка $(q, v) \models_{M'} (r_1, \lambda)$, па според тоа $(q, va) \models_{M'} (r_1, a) \vdash_{M'} (r_2, \lambda) \models_{M'} (p, \lambda)$. □

Да продолжиме со примерот прикажан на слика 3.8. Нека M' е автоматот прикажан на сл. 3.8. Бидејќи M' има 5 состојби, M'' ќе има $2^5 = 32$ состојби. Меѓутоа само неколку од нив ќе бидат битни за операцијата на премин на M'', имено оние состојби кои ќе можат да се достигнат од s'' читајќи некој влезен збор. Ќе го конструираме овој дел од M'' почнувајќи од s'' и додавајќи нова состојба само кога е неопходна како вредност на $\delta''(q, a)$ за некоја веќе воведена состојба q и некој $a \in \Sigma$.

Порано ги дефиниравме $E(q)$ за секоја состојба q од M'. Бидејќи $s'' = E(q_0) = \{q_0, q_1, q_2, q_3\}$, имаме дека

$$(q_1, a, q_0)$$

$$(q_1, a, q_4)$$

$$(q_3, a, q_4)$$

се сите премини (q, a, p) за некое $q \in s'$. Значи:

$$\delta''(s'', a) = E(q_0) \cup E(q_4) = \{q_0, q_1, q_2, q_3, q_4\}.$$

На ист начин постапуваме и за буквата $b \in \Sigma$. Имено,

$$(q_0, b, q_2)$$

$$(q_2, b, q_4)$$

се сите премини (q, b, p) за некој $q \in s''$. Значи

$$\delta''(s'', b) = E(q_2) \cup E(q_4) = \{q_2, q_3, q_4\}.$$

Повторувајќи ја истата постапка за новодобиените состојби, имаме:

$$\delta''(\{q_0, q_1, q_2, q_3, q_4\}, a) = \{q_0, q_1, q_2, q_3, q_4\}$$

$$\delta''(\{q_0, q_1, q_2, q_3, q_4\}, b) = \{q_2, q_3, q_4\}$$

$$\delta''(\{q_2, q_3, q_4\}, a) = E(q_4) = \{q_3, q_4\}$$

$$\delta''(\{q_2, q_3, q_4\}, b) = E(q_4) = \{q_3, q_4\}$$

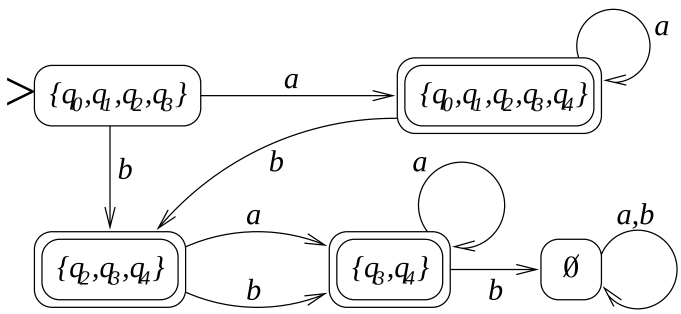
На крајот,

$$\delta''(\{q_3, q_4\}, a) = E(q_4) = \{q_3, q_4\}$$

$$\delta''(\{q_3, q_4\}, b) = \emptyset$$

и

$$\delta''(\emptyset, a) = \delta''(\emptyset, b) = \emptyset.$$



Слика 3.9:

Битниот дел од M'' е илустриран на сл.3.9. Множеството завршни состојби F'' ги содржи сите множества состојби во кои е член q_4, бидејќи $F' = \{q_4\}$. Значи $F'' = \{\{q_0, q_1, q_2, q_3, q_4\}, \{q_2, q_3, q_4\}, \{q_3, q_4\}\}$ е множеството завршни состојби за M''.

3.2.4 Својства на јазиците препознаени од конечен автомат

Основниот резултат во претходниот дел е дека класата јазици препознаени од конечни автомати останува иста дури и ако е воведен недетерминистички автомат. Во овој дел ќе покажеме дека комбинирање на јазици препознаени од конечни автомати со познатите операции со јазици не доведува до јазици надвор од спомнатата класа, од класата јазици препознаени со конечни автомати. Натаму, овие резултати ќе бидат докажани со техника со која од еден или повеќе автомати се конструира автомат кој го препознава јазикот добиен со дадената операција. На овој начин се добива алгоритам за конструкција на соодветен конечен автомат, која ќе ни помогне при воспоставување на врската помеѓу регуларните изрази и јазиците препознаени со конечни автомати.

Низ целиот овој дел Σ ќе биде зададена азбука, а главен резултат на овој дел ќе биде следнава теорема:

Теорема 3.2.2 *Класата јазици препознанени од конечни автомати е затворена во однос на:*

(*а*) *унија;*

(*б*) *конкатенација;*

(*в*) *Клиниева ѕвезда;*

(*г*) *комплемент;*

(*д*) *пресек.*

Доказ: Во секој од дадените случаи ќе конструираме автомат кој го распознава дадениот јазик од еден или повеќе зададени автомати.

(a) *Унија.* Нека L_1 и L_2 се јазици кои се препознаени од автоматите M_1 и M_2, соодветно. Нека

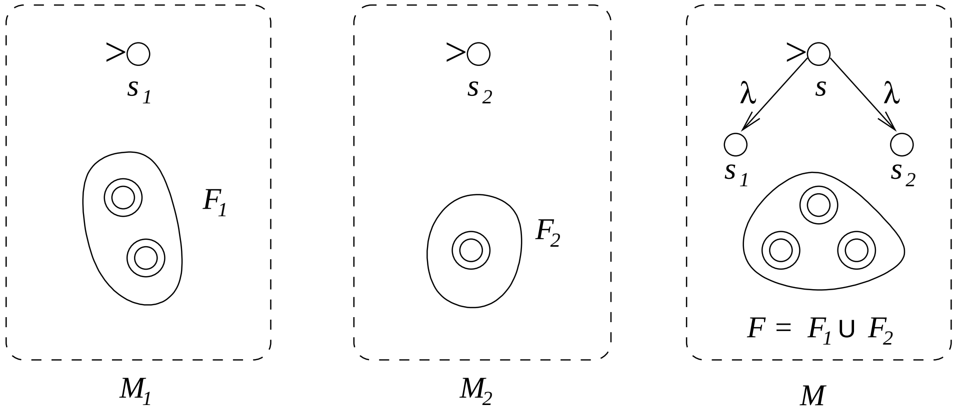
$$M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1) \text{ и } M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2).$$

Без губење на општоста на резултатот, можеме да претпоставиме дека K_1 и K_2 се дисјунктни множества.

Конструираме недетерминистички конечен автомат M кој го препознава јазикот $L(M_1) \cup L(M_2)$ на следниов начин: $M = (K, \Sigma, \Delta, s, F)$, каде што s е нова состојба што не припѓа во $K_1 \cup K_2$,

$$K = K_1 \cup K_2 \cup \{s\},$$

$$F = F_1 \cup F_2,$$



Слика 3.10:

$$\Delta = \Delta_1 \cup \Delta_2 \cup \{(s, \lambda, s_1), (s, \lambda, s_2)\}.$$

Значи, M започнува со проверка дали препознава некој збор од почетната состојба s, од која преминува или во s_1 или во s_2 и продолжува да работи како соодветниот автомат M_1 или M_2. Од тука се добива дека M го препознава точно јазикот $L(M_1) \cup L(M_2)$.

(б) *Конкатенација.* Повторно, нека M_1 и M_2 се недетерминистички конечни автомати; конструираме недетерминистички конечен автомат M што ќе го препознава јазикот $L(M_1) \circ L(M_2)$. Конструкцијата е прикажана шематски на сл.11; имено автоматот M почнува со работа како автоматот M_1, а потоа од завршна состојба на M_1 преминува во почетна состојба на M_2 и продолжува да работи како и автоматот M_2.

Формално, ако M_1, M_2 и M се означени како и во (а), тогаш:

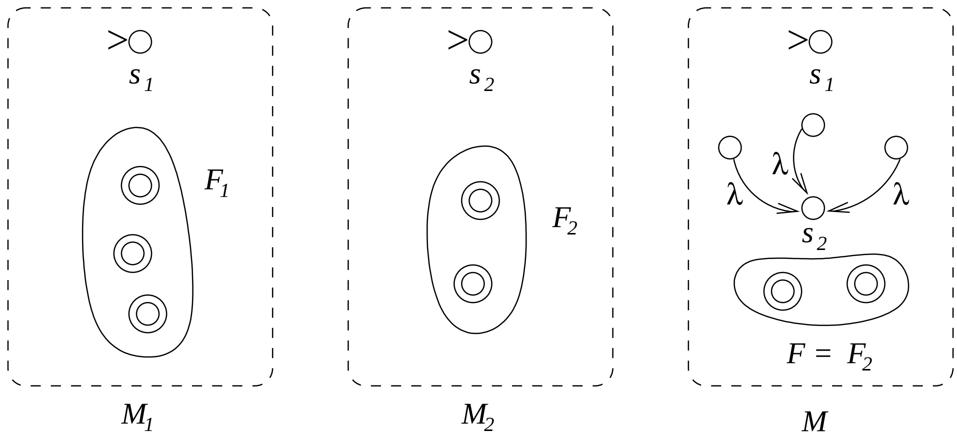
$$K = K_1 \cup K_2, \; s = s_1, \; F = F_2,$$

и

$$\Delta = \Delta_1 \cup \Delta_2 \cup (F_1 \times \{\lambda\} \times \{s_2\}).$$

Јасно е дека во овој случај автоматот M ќе го препознава точно јазикот $L(M_1) \circ L(M_2)$.

(в) *Клиниева ѕвезда.* Нека M_1 е недетерминистички конечен автомат. Ќе конструираме недетерминистички конечен автомат M, таков што $L(M) = L(M_1)^*$. Конструкцијата е слична на онаа за конкатенација и е илустрирана на слика 12. M се состои од сите состојби



Слика 3.11:

на M_1 и сите премини од M_1. Секоја завршна состојба на M_1 е завршна состојба и на M. Покрај ова M има нова почетна состојба s_1'. Оваа нова почетна состојба е истотака и завршна, така што M го препознава празниот збор λ и со него може да се достигне почетната состојба s_1 на M_1. Притоа релациите од M_1 можат да се изведуваат откако е иницирана машината M. Конечно, додадени се премини преку λ од секоја завршна состојба на M_1 назад до s_1. Значи, откако е отпочната работата на машината M_1, таа може да се иницира повторно од добиениот збор преку завршната состојба од M_1 која преку празниот збор се враќа назад во почетната состојба s_1 на M_1.

Формално, ако $M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$, тогаш $M = (K, \Sigma, \Delta, s_1', F)$ е таква што:

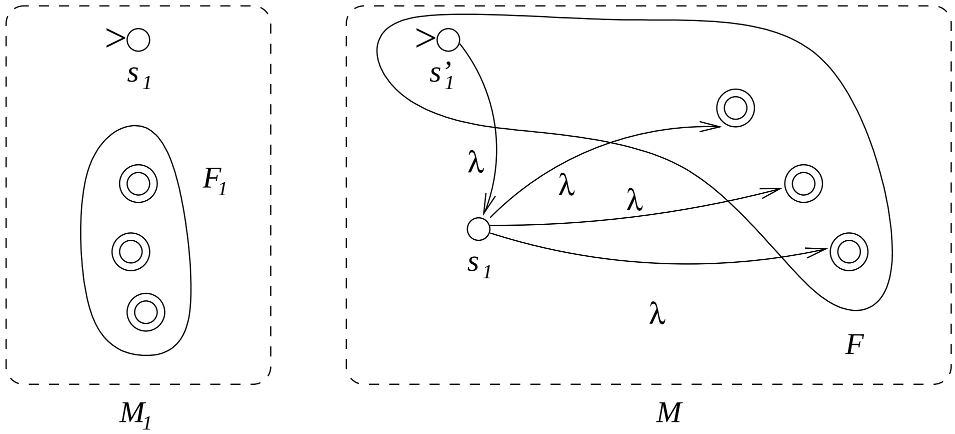
$$K = K_1 \cup \{s_1'\},$$

$$s_1' \text{ е нова состојба што не се содржи во } K_1,$$

$$F = F_1 \cup \{s_1'\},$$

$$\Delta = \Delta_1 \cup (F \times \{\lambda\} \times \{s_1\}) \cup \{(s_1', \lambda, s_1)\}.$$

Со директна проверка се добива дека ако $w \in L(M)$, тогаш или $w = \lambda$ или $w = w_1 \cdots w_k$ за некој $k \geq 1$, каде што за $i = 1, \cdots k$, постои $f_i \in F$такво што $(s_i, w_i) \models^*_{M_1} (f_i, \lambda)$, т.е. $w \in L(M_1)^*$. Обратно, ако $w \in L(M_1)^*$, тогаш или $w = \lambda$ или $w = w_1 \cdots w_k$, за некои $w_1, \cdots, w_k \in L(M_1)$. Во првиот случај $w \in L(M)$ затоа што s_1' е



Слика 3.12:

завршна состојба, а во вториот $w \in L(M)$, затоа што за некои $f_1, \cdots, f_k \in F_1$, $(s_1', w_1 \cdots w_k) \vdash_M (s_1, w_1 \cdots w_k) \models_M (f_1, w_2 \cdots w_k) \models_M (s_1, w_2 \cdots, w_k) \models_M (f_2, w_3 \cdots w_k) \models_M \cdots \models_M (f_k, \lambda)$. Значи $L(M) = L(M_1)$.

(г) *Комплемент.* Нека $M = (K, \Sigma, \delta, s, F)$ е детерминистички автомат. Тогаш јазикот $\Sigma^* \setminus L(M)$ е препознаен од автоматот

$$\overline{M} = (K, \Sigma, \delta, s, K \setminus F).$$

(д) *Пресек.* Ако L_1 и L_2 се јазици препознаени од конечни автомати M_1 и M_2, соодветно, тогаш

$$L_1 \cap L_2 = \Sigma^* \setminus ((\Sigma^* \setminus L_1) \cup (\Sigma^* \setminus L_2))$$

и $L_1 \cap L_2$ е препознаен од конечен автомат според (а) и (г) од ова својство. □

Да забележиме дека иако и двата автомати M_1 и M_2 на почетокот можат да бидат детерминистички автомати, во општ случај добиениот конечен автомат е недетерминистички

Горниве својства можат да се искористат за да се решат некои алгоритамски проблеми за конечни автомати. Наредното својство е, всушност, својство кое обезбедува алгоритам за одговарање на одредени прашања во врска со јазици препознаени со конечни автомати.

Теорема 3.2.3 *Нека M, M_1, M_2 се конечни автомати и $w \in \Sigma^*$. Постојат алгоритми за одговарање на следниве прашања:*

(*а*) *Дали $w \in L(M)$?*

(*б*) *Дали $L(M) = \emptyset$?*

(*в*) *Дали $L(M) = \Sigma^*$?*

(*г*) *Дали $L(M_1) \subseteq L(M_2)$?*

(*д*) *Дали $L(M_1) = L(M_2)$?*

Доказ: Доказот се спроведува со помош на претходната теорема. Претпоставуваме дека во секој од бараните случаи работиме со детерминистички автомати, затоа што покажавме дека за секој недетерминистичи автомат, постои детерминистички еквивалентен со него.

Во случајот (а) доволно е да ја следиме работата на даден збор $w \in \Sigma^*$, и после $|w|$ чекори да видиме дали последната добиена состојба при работата на автоматот е завршна. Бидејќи станува збор за детерминистички автомат, секој чекор е еднозначно определен од влезната буква и состојбата на автоматот во тој момент.

За да провериме дали $L(M) = \emptyset$ доволно е да провериме дали постои низа од последователни стрелки во дијаграмот на автоматот која води од почетната состојба до завршна состојба, што е возможно да се направи поради конечноста на автоматот.

За да одговориме на прашањето под (в), конструираме автомат M', таков што $L(M') = \Sigma^* \setminus L(M)$, и проверуваме дали $L(M') = \emptyset$.

За прашањето под (г), конструираме конечен детерминистичи автомат кој што го препознава јазикот $(\Sigma^* \setminus L(M_2)) \cap L(M_1)$ и проверуваме дали јазикот препознаен од овој автомат е празното множество.

На крај, за да одговориме на прашањето под (д), проверуваме дали $L(M_1) \subseteq L(M_2)$ и дали $L(M_2) \subseteq L(M_1)$. $\square$

3.2.5 Конечни автомати и регуларни јазици

На крајот од овој дел да ја разгледаме врската помеѓу конечните автомати и регуларните изрази. Бидејќи секој регуларен израз определува точно определен регуларен јазик, и обратно, секој регуларен јазик определува регуларен израз, всушност ќе ја разгледаме врската помеѓу јазиците препознаени од конечен автомат и регуларните јазици.

Теорема 3.2.4 (*Cleeny) Еден јазик е регуларен акко е препознаен од конечен автомат.*

Доказ: Да се потсетиме дека класата регуларни јазици е минималната класа јазици што ги содржи $\emptyset$, едноелементните подмножества од азбуката Σ и е затворена во однос на унија, конкатенација и Клиниева ѕвезда. Според Теоремата 3.2.2., класата јазици препознаени од конечен автомат го содржи празниот јазик, секој конечен јазик и е затворена во однос на унија, конкатенација и Клиниева ѕвезда. Значи, секој регуларен јазик е препознаен од конечен автомат.

Нека $M = (K, \Sigma, \delta, s, F)$ е детерминистички конечен автомат. Доволно е да покажеме дека постои регуларен јазик R, таков што $R = L(M)$. $L(M)$ ќе го претставиме како конечна унија од едноставни јазици и ќе покажеме дека секој од овие јазици е регуларен. Нека $K = \{q_1, \cdots, q_n\}$, $s = q_1$. За $i, j = 1, \cdots, n$ и $k = 1, \cdots, n+1$, нека со $R(i,j,k)$ биде означено множеството од зборови од Σ^* кое автоматот M го води од q_i во q_j без да помине низ состојба q_r, за $r \geq k$. Формално:

$$R(i,j,k) = \{x \in \Sigma^* | (q_i, x) \models_M (q_j, \lambda), \text{и ако}$$

$$(q_i, x) \models_M (q_l, y) \text{ за некој } y \in \Sigma^*, \text{ тогаш}$$

$$l < k, \text{или } y = \lambda \text{ и } l = j, \text{ или } y = x \text{ и } l = i\}.$$

Кога $k = n + 1$, тогаш следува дека

$$R(i,j,n+1) = \{x \in \Sigma^* | (q_i, x) \models_M (q_j, \lambda)\}.$$

Поради тоа

$$L(M) = \bigcup \{R(1,j,n+1) | q_j \in F\}.$$

Битно е да наведеме дека секое множество од облик $R(i,j,k)$ е регуларен јазик, па и $L(M)$ е регуларен, како унија од регуларни јазици.

Тврдењето дека $R(i,j,k)$ е регуларен за секој $k = 1, \cdots, n+1$ ќе го докажеме со индукција по k.

За $k = 1$ имаме

$$R(i,j,1) = \begin{cases} \{\sigma \in \Sigma | \delta(q_i, \sigma) = q_j\} & \text{ако } i \neq j \\ \{\lambda\} \cup \{\sigma \in \Sigma | \delta(q_i, \sigma) = q_j\} & \text{ако } i = j \end{cases}$$

Нека тврдењето е точно за секој $k = 1, \cdots, s$, т.е. секое од множествата $R(i,j,k)$ се дефинирани и претставуваат регуларен јазик. Тогаш

$$R(i,j,k+1) = R(i,j,k) \cup R(i,k,k)R(k,k,k)^*R(k,j,k).$$

Ова равенство покажува дека ако сакаме да стигнеме од q_i до q_j без да поминеме низ состојба q_r, за r поголемо k, M може или

- да премине од q_i во q_j без да премине низ состојба нумерирана со број поголем од $k-1$; или
- да премине
  (а) од q_i во q_k; потоа
  (б) од q_k во q_k со повторување на овој премин; и потоа
  (в) од q_k во q_j;

  во секој од случаите без да помине низ состојба нумерирана со број поголем од $k-1$.

Значи, секој од јазиците $R(i,j,k)$ е регуларен, па и јазикот $L(M)$, како унија од регуларни јазици, е регуларен. □

Ние спомнавме дека постојат јазици кои не се регуларни. Горниве техники ни овозможуваат да покажеме дека некој јазик е регуларен, но немаме начин да покажеме дека некој јазик *не е регуларен.* Наредната теорема, наречена *Теорема за пумпање*, ќе обезбеди техника за докажување дека некој јазик не е регуларен.

Теорема 3.2.5 (Теорема за пумпање) *Нека L е бесконечен регуларен јазик. Тогаш постојат зборови x, y и z, такви што $y \neq \lambda$ и за секој $n \geq 0$, $xy^nz \in L$.*

Доказ: Бидејќи L е регуларен, L е препознаен од конечен детерминистички автомат M. Нека M има n состојби. Од тоа што L е бесконечен, M препознава некој збор w со должина поголема или еднаква на n. Нека $l = |w|, w = \sigma_1\sigma_2 \cdots \sigma_l$, каде што $\sigma_i \in \Sigma$. Да ја разгледаме работата на M врз зборот w:

$$(q_0, \sigma_1 \cdots \sigma_l) \vdash_M \cdots \vdash_M (q_{l-1}, \sigma_l) \vdash_M (q_l, \lambda)$$

каде што q_0 е почетната состојба, а q_l е завршна состојба на M. Бидејќи $l \geq n$ и M има само n состојби, постојат i и j, $0 \leq i < j \leq l$, такви што $q_i = q_j$. Значи низата $\sigma_{i+1} \cdots \sigma_j$ ја доведува машината M од состојба q_i повторно во истата состојба q_i, и оваа низа е непразна, бидејќи $i+1 \leq j$. Но, во овој случај, овој подзбор може да се отстрани од w или произволен број копии на овој подзбор да се внесат последователно точно по

j-тиот симбол во w, и M ќе го препознава новодобиениот збор. Значи M го препознава зборот $\sigma_1 \cdots \sigma_i (\sigma_{i+1} \cdots \sigma_j)^n \sigma_{j+1} \cdots \sigma_l$, за секој $n \geq 0$; па ако ставиме $x = \sigma_1 \cdots \sigma_i$, $y = \sigma_{i+1} \cdots \sigma_j$ и $z = \sigma_{j+1} \cdots \sigma_l$, тогаш

$$(q_0, xy^n z) \models_M (q_i, y^{n-1} z) \models_M \cdots \models_M (q_i, z) \models_M (q_l, \lambda). \ \square$$

Наредните примери ќе ја илустрираат примената на оваа теорема.

Пример 1.

$L_1 = \{a^n b^n | n \geq 0\}$ не е регуларен.

Бидејќи L_1 е бесконечен јазик, на него може да се примени теоремата за пумпање. Значи, постојат зборови x, y и z, такви што $y \neq \lambda$ и $xy^n z \in L_1$. Возможни се три случаи:

Случај 1. y се состои од низа која го содржи само симболот a. Тогаш $x = a^p$, $y = a^q$ и $z = a^r b^s$, каде што $p, r \geq 0$ и $q, s > 0$. Но, тогаш, L мора да го содржи и зборот $xy^n z = a^{p+qn+r} b^s$ за секој $n \geq 0$, и најмногу една од овие низи има еднаков број на симболи a и b.

Случај 2. Ако y се состои од низа која го содржи само симболот b, тогаш, како и во случајот 1., се докажува дека L мора да содржи и зборови во кои бројот на појавувања на a и b не е еднаков.

Случај 3. Нека y содржи и симболи a и симболи b. Тогаш, за $n > 0$, $xy^n z$ ќе има појавување на b што му претходи на симболот a, што значи, $xy^n z \notin L$.

Пример 2.

$L_2 = \{a^n | n \text{ е прост број}\}$ не е регуларен јазик.

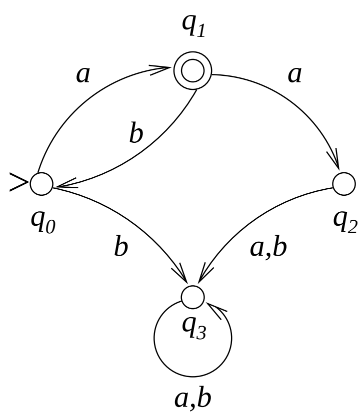
Нека L_2 е регуларен. Тогаш постојат x, y и z како во теоремата за пумпање, $x = a^p$, $y = a^q$ и $z = a^r$, каде што $p, r \geq 0$ и $q > 0$. Во тој случај, според теоремата, и $a^{p+nq+r} \in L$, за секој $n \geq 0$, што значи дека за секој $n \geq 0$ бројот $p + nq + r$ е прост, што не е возможно. Имено, за $n = p + 2q + r + 2$ имаме $p + nq + r = (q + 1) \cdot (p + 2q + r)$, т.е. е производ на два броја поголеми од 1.

3.2.6 Задачи

1. Нека M е детерминистички конечен автомат. Под точно кои услови $\lambda \in L(M)$? Да се докаже добиеното тврдење.

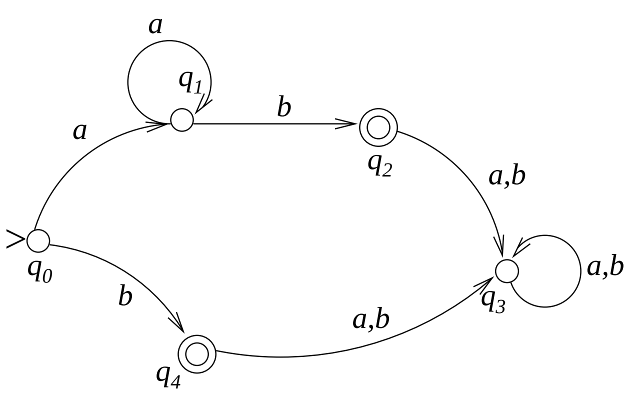
2. Кои јазици се препознаени со автоматите прикажани со следниве дијаграми:

(а)



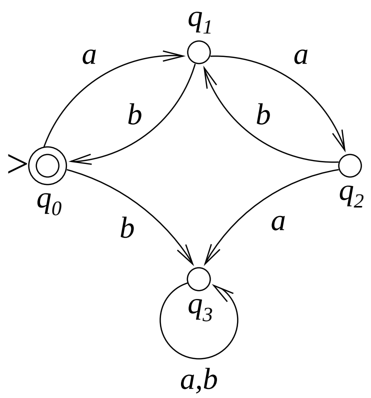
Слика 3.13:

(б)



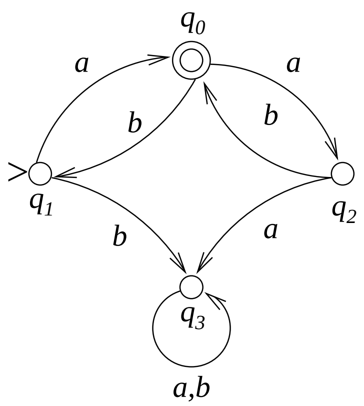
Слика 3.14:

(в)



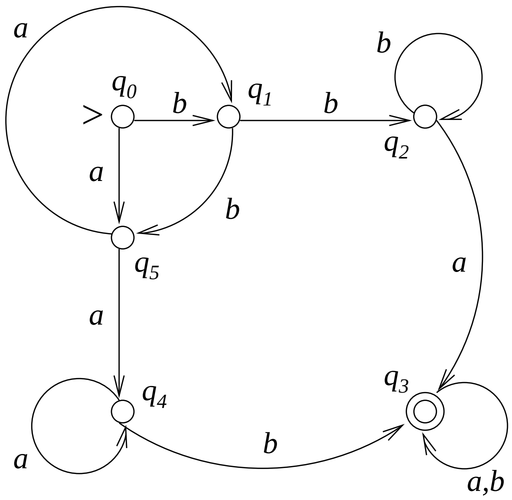
Слика 3.15:

(г)



Слика 3.16:

(д)



Слика 3.17:

3. Да се конструира детерминистички конечен автомат кој ќе ги препознава следниве јазици:

   (а) $\{w \in \{a,b\}^* |$ на секој a во w непосредно му претходи и по него следува b $\}$.

   (б) $\{w \in \{a,b\}^* | w$ го содржи подзборот $abab\}$.

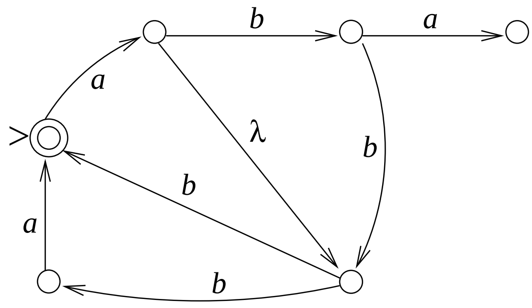
   (в) $\{w \in \{a,b\}^* | w$ не го содржи ни подзборот aa ни подзборот $bb\}$.

   (г) $\{w \in \{a,b\}^* | w$ непарен број појавувања на a и парен број појавувања на $b\}$.

   (д) $\{w \in \{a,b\}^* | w$ ги содржи и ab и ba како подзборови $\}$.

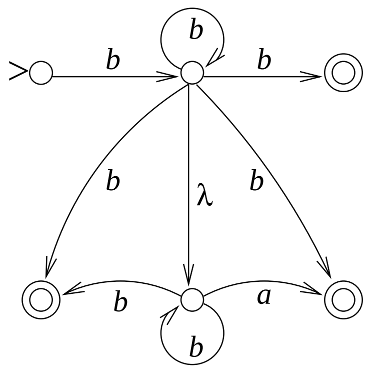
4. Кој од следниве зборови е препознаен од наведениот недетерминистички конечен автомат:

(а) aa; aba; abb; ab; $abab$, каде што автоматот е прикажан со дијаграмот на сл.3.18.



Слика 3.18:

(б) ba; ab; bb; b; bba, каде што автоматот е прикажан со дијаграмот на сл.3.19.



Слика 3.19:

5. Конструирај недетерминистички конечни автомати (прикажани со нивните дијаграми) кои ќе ги прифаќаат следниве јазици:

   (а) $(ab)^*(ba)^* \cup aa^*$;

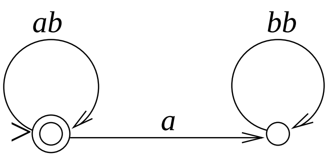
   (б) $((ab \cup aab)^*a^*)^*$;

   (в) $((a^*b^*a^*)^*b)^*$;

   (г) $(ba \cup b)^* \cup (bb \cup a)^*$.

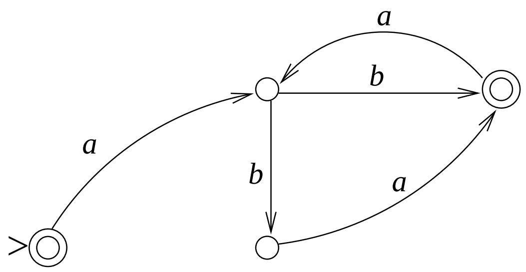
6. Да се напишат регуларните изрази за јазиците препознаени од следниве недетерминистички конечни автомати:

(а)



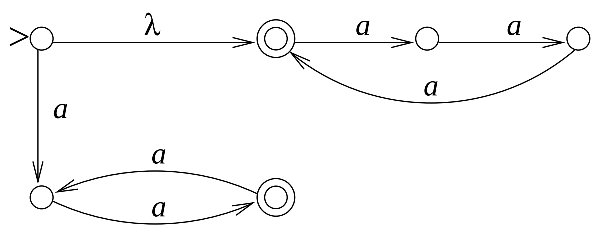
Слика 3.20:

(б)



Слика 3.21:

(в)



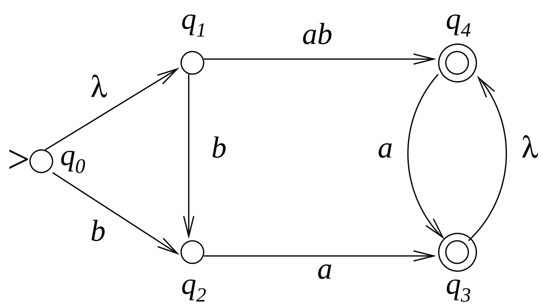
Слика 3.22:

7. Нека $\Sigma = \{0, 1, \cdots, 9\}$. Да се состави конечен автомат кој ги препознава јазиците:

   (а) $\{w \in \Sigma^* | \overline{w}$ е број делив со 2 $\}$;
   (б) $\{w \in \Sigma^* | \overline{w}$ е број делив со 3 $\}$;
   (в) $\{w \in \Sigma^* | \overline{w}$ е број делив со 4 $\}$;

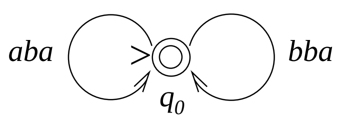
8. Конструирај детерминистички автомат еквивалентен со недетерминистичкиот прикажан со својот дијаграм на премин:

   (а)



Слика 3.23:

   (б)



Слика 3.24:

   Објасни што се случува кога алгоритмот за премин од недетерминистичи автомат во детерминистички ќе се примени на детерминистички автомат.

9. Нека $L, L' \subseteq \Sigma^*$. Да се определи конечен автомат кој го препознава јазикот:

   (а) $Pref(L) = \{w \in \Sigma^* | x = wy$ за некои $x \in L, y \in \Sigma^*\}$. (множество *префикси* на L).

(б) $Suf(L) = \{w \in \Sigma^* | x = yw$ за некои $x \in L, y \in \Sigma^*\}$. (множество *суфикси* на L).

(в) $Subseq(L) = \{w_1 w_2 \cdots w_k | k \in N, w_j \in \Sigma^*$ за $j = 1, 2, \cdots, k$ и постои низа $x = x_0 w_1 x_1 w_2 \cdots w_k x_k$ во $L\}$. (множество *поднизи* на L).

(г) $L/L' = \{w \in \Sigma^* | wx \in L$ за некој $x \in L'\}$. (множество *десни комплекси од* L *по* L').

(д) $Max(L) = \{w \in L | x \neq \lambda$ повлекува $wx \notin L\}$.

(ѓ) $L^R = \{w^R | w \in L\}$.

Покажи дека ако L е препознаен од накој конечен автомат, тогаш препознаен е и секој од:

- $Pref(L)$;
- $Suf(L)$;
- $Subseq(L)$;
- L/L', каде што и L' е препознаен од некој конечен автомат;
- L/L', каде L' е произволен јазик;
- $Max(L)$;
- L^R.

10. Образложи дали постојат алгоритми кои одговараат на следниве прашања за конечните автомати M_1 и M_2:

   (а) Дали $L(M_1)$ и $L(M_2)$ се дисјунктни.

   (б) Дали $L(M_1)$ и $L(M_2)$ се комплементарни, т.е. дали $L(M_1) = \Sigma^* \setminus L(M_2)$.

   (в) Дали $L(M_1)^* = L(M_2)$.

11. Нека $M = (K, \Sigma, \delta, s, F)$. Дефинираме серија еквивалентности на K: $\equiv_0, \equiv_1, \cdots, \equiv_n, \cdots$ со:

$p \equiv_0 q \iff p$ и q се и двете завршни или и двете незавршни состојби.

$$p \equiv_{k+1} q \iff p \equiv_k q \text{ и за секој } a \in \Sigma, \delta(p, a) \equiv_k \delta(q, a).$$

Кога ќе добиеме некој k, таков што $\equiv_k = \equiv_{k+1}$, запираме со конструкцијата на еквивалентностите и го формираме конечниот автомат $M' = (K', \Sigma, \delta', s', F')$, каде што

$$K' = K / \equiv_n$$

$$\delta'(p^{\equiv_n}, a) = \delta(p, a)^{\equiv_n}$$

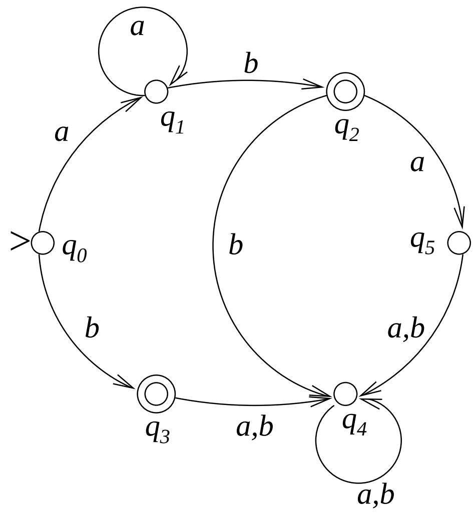
$$s' = s^{\equiv_n}$$

$$F' = \{q^{\equiv_n} | q \in F\}.$$

Да се провери дека важи $L(M) = L(M')$. Притоа автоматот M' е автомат со минимален број на состојби кој што го препознава јазикот $L(M)$.

За автоматот M' велиме дека е *минимален* кој што го препознава јазикот $L(M)$, а самата конструкција ја викаме постапка за *минимизација* на даден автомат.

Да се минимизира автоматот:



Слика 3.25:

12. (а) Да се состави недетерминистички конечен автомат кој го препознава јазикот $(baa)^*(baa^*)^*(abb^*)$.
    (б) Добиениот автомат да се претвори во детерминистички конечен автомат.
    (в) Автоматот добиен под (б) да се минимизира.
13. (а) Да се состави недетерминистички конечен автомат кој го препознава јазикот: $(a \cup b)^*aabab$;
    (б) Добиениот автомат да се претвори во детерминистички конечен автомат.
    (в) Автоматот добиен под (б) да се минимизира.

14. (а) Да се состави недетерминистички конечен автомат кој го препознава јазикот: $(ab \cup aab \cup aba)^*$.
    (б) Да се конструира детерминистички конечен автомат еквивалентен со недетерминистичкиот автомат добиен под (а).
    (в) Добиениот детерминистички конечен автомат да се минимизира.
15. Да се даде директна конструкција на автомат којшто ќе прифаќа пресек на два регуларни јазика дадени со соодветни автомати.
16. Да се конструира дијаграм на премин на конечен автомат кој ќе го прифаќа комплементот на јазикот a^*b.
17. Докажи дека јазикот $L = \{a^n b^m a^{n+m} | n, m \geq 1\}$ не е регуларен.
18. Да се докаже дека ниту еден од следниве јазици не е регуларен:
    (а) $L = \{ww^R | w \in \{a, b\}^*\}$.
    (б) $L = \{ww | w \in \{a, b\}^*\}$.
    (в) $L = \{w\overline{w} | w \in \{a, b\}^*\}$ и $\overline{w}$ е збор добиен од w со замена на секое појавување на a со b, а на b со a.
19. Да се провери точноста на следниве искази:
    (а) Секое подмножество од регуларен јазик е регуларен јазик.
    (б) Секој регуларен јазик содржи регуларен јазик како вистинско подмножество.
    (в) Ако L е регуларен јазик, тогаш и $\{xy | x \in L, y \notin L\}$ е регуларен.
    (г) $\{w | w = w^R\}$ е регуларен јазик.
    (д) Произволна унија од регуларни јазици е регуларен јазик.