

Езици и автомати

Иво Стратев

7 февруари 2018 г.

Съдържание

1	Твърдение. За всеки нетотален ДКА съществува еквивалентен на него тотален ДКА	2
2	Затвореност на езиците разпознавани от ДКА относно операцията обединение	3
3	Затвореност на езиците разпознавани от ДКА относно операцията сечение	3
4	Допълнението на език разпознаван от ДКА е също език разпознаван от ДКА	4
5	За всеки НКА съществува еквивалентен на него НКА със само едно финално състояние	5
6	За всеки език L разпознаван от краен автомат съществува НКА, който разпознава L^{rev}	6
7	За всеки НКА съществува еквивалентен на него ДКА	6
8	Еквивалентност на класа от автоматни езици с класа на регулярни езици	7
8.1	Теорема на Клини (За всеки автоматен език съществува регулярен израз, който го описва)	7
8.2	За всеки регулярен израз съществува краен автомат, който описва същия език	9
8.2.1	Изконни езици	9
8.2.2	Затвореност относно конкатенация	9
8.2.3	Затвореност относно обединение	9
8.2.4	Затвореност относно операцията звезда на Клини	9
8.2.5	Заклучение	9
9	Лема за покачването (за регулярни езици)	9
9.1	Следствия	9
9.1.1	Проверка дали един регулярен език е празен	9
9.1.2	Проверка дали два регулярни езика съвпадат	9
9.1.3	Проверка дали един регулярен език е краен или безкраен	9
10	Теорема на Майхил-Нероуд	9
11	Алгоритъм за минимализация на автомат	9

1 Твърдение. За всеки нетотален ДКА съществува еквивалентен на него тотален ДКА

Нека $A = \langle Q, \Sigma, \delta, s, F \rangle$ е нетотален ДКА и нека $q_e \notin Q$. Тогава $A' = \langle Q \cup \{q_e\}, \Sigma, \delta', s, F \rangle$ където $\delta' = \delta \cup \{((q_e, a), q_e) \mid a \in \Sigma\} \cup \{((q, a), q_e) \mid q \in Q, a \in \Sigma : \forall p \in Q ((q, a), p) \notin \delta\}$ е тотален и еквивалентен на A , тоест $\mathcal{L}(A') = \mathcal{L}(A)$.

Първо ще докажем, че δ' е тотална функция, тоест $\forall q \in Q \cup \{q_e\}, \forall a \in \Sigma \exists! p \in Q \cup \{q_e\} : \delta'(q, a) = p$.

Нека $q \in Q \cup \{q_e\}$ и нека $a \in \Sigma$.

Ако $q = q_e$ то от дефиницията на δ' имаме, че ако $p \in Q \cup \{q_e\} \wedge ((q_e, a), p) \in \delta'$ то $p = q_e$, защото $q_e \notin Q \implies ((q_e, a), q_e) \notin \delta$ и единствените преходи от q_e , са дефинирани от $\{((q_e, a), q_e) \mid a \in \Sigma\}$.

Ако $q \in Q$ то за всяко p от $Q \cup \{q_e\}$ има два случая $((q, a), p) \in \delta$ или $((q, a), p) \notin \delta$. Ако $((q, a), p) \in \delta$ тогава $\delta'(q, a) = p$, което лесно се вижда от дефиницията на δ' . Ако $((q, a), p) \notin \delta$ тогава $p = q_e$, което отново лесно се съобразява от дефиницията на δ' , следователно $\delta'(q, a) = p$. Следователно δ' е тотална функция и значи A' е тотален ДКА. Остава да докажем, че $\mathcal{L}(A) = \mathcal{L}(A')$.

Първо ще докажем, че $\mathcal{L}(A) \subseteq \mathcal{L}(A')$. Нека $\omega \in \mathcal{L}(A)$ е произволна дума. Тогава

$$\exists a_1, \dots, a_{|\omega|} \in \Sigma : \omega = \prod_{i=1}^{|\omega|} a_i \in \mathcal{L}(A) \implies \delta^*(s, \omega) \in F \implies$$

$$\exists q_1, \dots, q_{|\omega|} \in Q : \delta(s, a_1) = q_1, \forall i \in \{1, \dots, |\omega| - 1\} \delta(q_i, a_{i+1}) = q_{i+1},$$

$$q_{|\omega|} \in F \implies \delta'(s, a_1) = q_1, \forall i \in \{1, \dots, |\omega| - 1\} \delta'(q_i, a_{i+1}) = q_{i+1}$$

$$\implies \delta^*(s, \omega) \in F \implies \omega \in \mathcal{L}(A') \implies \forall \alpha \in \mathcal{L}(A) \alpha \in \mathcal{L}(A') \implies \mathcal{L}(A) \subseteq \mathcal{L}(A').$$

Нека сега $\beta \in \mathcal{L}(A')$ е произволна дума. Тогава

$$\exists b_1, \dots, b_{|\beta|} \in \Sigma : \beta = \prod_{i=1}^{|\beta|} b_i \in \mathcal{L}(A') \implies \delta^*(s, \beta) \in F \implies$$

$$\exists q_1, \dots, q_{|\beta|} \in Q \cup \{q_e\} : \delta'(s, b_1) = q_1, \forall i \in \{1, \dots, |\beta| - 1\} \delta'(q_i, b_{i+1}) = q_{i+1},$$

$$q_{|\beta|} \in F. \text{ Да допуснем, че } i \in \{1, \dots, |\beta| - 1\} : q_i = q_e, \forall b \in \Sigma \delta'(q_e, b) = q_e$$

$$\implies \forall j \in \{i, \dots, |\beta|\} q_j = q_e$$

$$\implies q_{|\beta|} = q_e \notin Q \implies q_{|\beta|} \notin F \implies \beta \notin \mathcal{L}(A) \implies \nexists$$

$$\implies \forall k \in \{1, \dots, |\beta|\} q_k \in Q \implies \delta(s, b_1) = q_1, \delta(q_k, b_{k+1}) = q_{k+1}$$

$$\implies \delta^*(s, \beta) = q_{|\beta|} \in F \implies \beta \in \mathcal{L}(A) \implies \forall \gamma \in \mathcal{L}(A') \gamma \in \mathcal{L}(A)$$

$$\implies \mathcal{L}(A') \subseteq \mathcal{L}(A) \implies \mathcal{L}(A') = \mathcal{L}(A) \quad \square$$

Следствие: БОО можем да разглеждаме само тотални ДКА.

2 Затвореност на езиците разпознавани от ДКА относно операцията обединение

Нека Σ е азбука и $A_1 = \langle Q_1, \Sigma, \delta_1, s_1, F_1 \rangle$ и $A_2 = \langle Q_2, \Sigma, \delta_2, s_2, F_2 \rangle$ са тотални ДКА. Тогава нека $A = \langle Q_1 \times Q_2, \Sigma, \delta, (s_1, s_2), (Q_1 \times F_2) \cup (F_1 \times Q_2) \rangle$. Където $\delta : (Q_1 \times Q_2) \times \Sigma \rightarrow Q_1 \times Q_2$ и

$$\delta = \{(((q_1, q_2), a), (\delta_1(q_1, a), \delta_2(q_2, a))) \mid q_1 \in Q_1, q_2 \in Q_2, a \in \Sigma\}$$

Идеята на конструкцията е да симулираме двата автомата едновременно, тоест правим паралелни изчисления, за това налагаме изискването автоматите да са тотални. Автомата разпознава дума, точно когато поне едно от двете изчисления завърши във финално състояние.

Ще докажем, че $\forall q_1 \in Q_1, \forall q_2 \in Q_2, \forall \alpha \in \Sigma^* \delta^*((q_1, q_2), \alpha) = (\delta_1^*(q_1, \alpha), \delta_2^*(q_2, \alpha))$

$$\begin{aligned} \forall q_1 \in Q_1, \forall q_2 \in Q_2, \forall a \in \Sigma \delta^*((q_1, q_2), a) &= \delta((q_1, q_2), a) = \\ &= (\delta_1(q_1, a), \delta_2(q_2, a)) = (\delta_1^*(q_1, a), \delta_2^*(q_2, a)) \end{aligned}$$

Нека $\exists n \in \mathbb{N}^+ : \forall q_1 \in Q_1, \forall q_2 \in Q_2, \forall \alpha \in \Sigma^* : |\alpha| = n$
 $\delta^*((q_1, q_2), \alpha) = (\delta_1^*(q_1, \alpha), \delta_2^*(q_2, \alpha))$

Нека $q_1 \in Q_1, q_2 \in Q_2, \beta \in \Sigma^* : |\beta| = n+1 \implies \exists \alpha \in \Sigma^*, a \in \Sigma : \beta = \alpha.a \implies$
 $\delta^*((q_1, q_2), \beta) = \delta^*((q_1, q_2), \alpha.a) = \delta(\delta^*((q_1, q_2), \alpha), a) = \delta((\delta_1^*(q_1, \alpha), \delta_2^*(q_2, \alpha)), a) =$
 $(\delta_1(\delta_1^*(q_1, \alpha), a), \delta_2(\delta_2^*(q_2, \alpha), a)) = (\delta_1^*(q_1, \alpha.a), \delta_2^*(q_2, \alpha.a)) = (\delta_1^*(q_1, \beta), \delta_2^*(q_2, \beta))$

$$\implies \forall t_1 \in Q_1, \forall t_2 \in Q_2, \forall \gamma \in \Sigma^* \delta^*((t_1, t_2), \gamma) = (\delta_1^*(t_1, \gamma), \delta_2^*(t_2, \gamma)) \quad \square$$

Ще покажем, че $\mathcal{L}(A) = \mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.

Нека $\omega \in \mathcal{L}(A)$ е произволна дума следователно от дефиницията за език на ДКА

$$\delta^*((s_1, s_2), \omega) = (\delta_1^*(s_1, \omega), \delta_2^*(s_2, \omega)) \in (Q_1 \times F_2) \cup (F_1 \times Q_2) \iff$$

$$(\delta_1^*(s_1, \omega), \delta_2^*(s_2, \omega)) \in Q_1 \times F_2 \vee (\delta_1^*(s_1, \omega), \delta_2^*(s_2, \omega)) \in F_1 \times Q_2 \iff$$

$$\delta_2^*(s_2, \omega) \in F_2 \vee \delta_1^*(s_1, \omega) \in F_1 \iff \omega \in \mathcal{L}(A_2) \vee \omega \in \mathcal{L}(A_1) \iff \omega \in \mathcal{L}(A_1) \cup \mathcal{L}(A_2)$$

$$\implies \forall \alpha \in \mathcal{L}(A) \iff \alpha \in \mathcal{L}(A_1) \cup \mathcal{L}(A_2) \implies \mathcal{L}(A) = \mathcal{L}(A_1) \cup \mathcal{L}(A_2) \quad \square$$

3 Затвореност на езиците разпознавани от ДКА относно операцията сечение

Нека Σ е азбука и $A_1 = \langle Q_1, \Sigma, \delta_1, s_1, F_1 \rangle$ и $A_2 = \langle Q_2, \Sigma, \delta_2, s_2, F_2 \rangle$ са тотални ДКА. Тогава нека $A = \langle Q_1 \times Q_2, \Sigma, \delta, (s_1, s_2), F_1 \times F_2 \rangle$.

$$\delta = \{(((q_1, q_2), a), (\delta_1(q_1, a), \delta_2(q_2, a))) \mid q_1 \in Q_1, q_2 \in Q_2, a \in \Sigma\}$$

Конструкцията е същата, като на автомата разпознаващ обединението на двата езика с изключение на финалните състояния, тук ще искаме думата да бъде разпозната и от двата автомата едновременно за да кажем, че е разпозната от конструируания автомат.

Ще покажем, че $\mathcal{L}(A) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

Нека $\omega \in \mathcal{L}(A)$ е произволна дума следователно от дефиницията за език на ДКА

$$\delta^*((s_1, s_2), \omega) = (\delta_1^*(s_1, \omega), \delta_2^*(s_2, \omega)) \in F_1 \times F_2 \iff$$

$$\delta_1^*(s_1, \omega) \in F_1 \wedge \delta_2^*(s_2, \omega) \in F_2 \iff \omega \in \mathcal{L}(A_1) \wedge \omega \in \mathcal{L}(A_2) \iff \omega \in \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$$

$$\implies \forall \alpha \in \mathcal{L}(A) \iff \alpha \in \mathcal{L}(A_1) \cap \mathcal{L}(A_2) \implies \mathcal{L}(A) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2) \quad \square$$

4 Допълнението на език разпознаван от ДКА е също език разпознаван от ДКА

Нека $A = \langle Q, \Sigma, \delta, s, F \rangle$ е тотален ДКА, знаем че БОО можем да го изискам, налагаме го като изискване, защото всяка дума, за която не е дефиниран преход считаме, че не е в езика на автомата, но тя е в допълнението на езика и ни трябва начин, по който да разпознаем цялата дума, това може да стане като си поискаме автомата да е тотален, защото стигайки до състоянието на грешка, автомата след това автоматично прочита остатъка на дума и не променя своето състояние.

Нека $A' = \langle Q, \Sigma, \delta, s, Q \setminus F \rangle$. Ще докажем, че $\mathcal{L}(A') = \overline{\mathcal{L}(A)}$.

Идеята е на построеният автомат е че за думите, за които изчислението не завършва във финално състояние не се разпознават от автомата, тоест не са в неговия език, значи са в допълнителния език. На строго математически $\overline{\mathcal{L}(A)} = \{\omega \in \Sigma^* \mid \delta^*(s, \omega) \notin F\} = \{\omega \in \Sigma^* \mid \delta^*(s, \omega) \in Q \setminus F\}$

Първо нека $\omega \in \mathcal{L}(A')$ е произволна дума. $\omega \in \mathcal{L}(A') \implies \delta^*(s, \omega) \in Q \setminus F$

$$\implies \delta^*(s, \omega) \notin F \implies \omega \notin \mathcal{L}(A) \implies \omega \in \overline{\mathcal{L}(A)} \implies \forall \alpha \in \mathcal{L}(A') \alpha \in \overline{\mathcal{L}(A)}$$

$$\implies \mathcal{L}(A') \subseteq \overline{\mathcal{L}(A)}$$

След това нека вземем произволна дума $\beta \in \overline{\mathcal{L}(A)} \implies \beta \notin \mathcal{L}(A) \implies$

$$\neg(\delta^*(s, \beta) \in F) = \delta^*(s, \beta) \notin F \implies \delta^*(s, \beta) \in Q \setminus F \implies \beta \in \mathcal{L}(A') \implies$$

$$\forall \gamma \in \overline{\mathcal{L}(A)} \gamma \in \mathcal{L}(A') \implies \overline{\mathcal{L}(A)} \subseteq \mathcal{L}(A') \implies \mathcal{L}(A') = \overline{\mathcal{L}(A)} \quad \square$$

5 За всеки НКА съществува еквивалентен на него НКА със само едно финално състояние

Нека $N = \langle Q, \Sigma, \Delta, s, F \rangle$ и нека $f \notin Q$. Нека $Q' = Q \setminus F \cup \{f\}$ и нека $replace : Q \rightarrow Q'$

$$replace(q) = \begin{cases} q & , q \notin F \\ f & , q \in F \end{cases}. \text{ Тогава}$$

$S = \langle (Q', \Sigma, \Delta', replace(s), \{f\}) \rangle$, където

$$\Delta' = \{((replace(q), a), \{replace(p) \mid p \in \Delta(q, a)\}) \mid q \in Q, a \in \Sigma\}.$$

Идеята на тази конструкция е да слеем всички финални състояния в едно. Очевидно функцията $replace$ прави точно това, ако ѝ бъде подадено състояние на автомата N и то е финално за N то бива изобразено в единственото финално за S , в противен случай $replace$ действа като id_Q . Също така очевидна е замената на всяко финално състояние от N с f във функцията на преходите на построения автомат S .

Сега формално ще докажем, че $\mathcal{L}(S) = \mathcal{L}(N) = \{\omega \in \Sigma^* \mid \Delta^*(s, \omega) \cap F \neq \emptyset\}$.

Нека вземем произволна дума $\omega \in \mathcal{L}(S) \implies$

$$\exists a_1, \dots, a_{|\omega|} \in \Sigma, q_0, q_1, \dots, q_{|\omega|} \in Q' : \omega = \prod_{i=1}^{|\omega|} a_i \wedge q_0 = replace(s)$$

$\wedge q_{|\omega|} = f \wedge \forall i \in \{0, \dots, |\omega| - 1\} q_{i+1} \in \Delta'(q_i, a_{i+1})$. Ще покажем, че имаме изчисление по автомата N , което завършва във финално за N състояние.

Ако $s \in F \implies replace(s) = f$. Тогава нека $p_0 = s$, в противен случай $p_0 = s = q_0$. След което за всяко $k \in \{1, \dots, |\omega|\}$ ако $q_k \neq f$ полагаме $p_k = q_k$, ако $q_k = f$ последователно пробваме с $p_k \in \Delta(p_{k-1}, a_k) \cap F$ от дефиницията на Δ' е ясно, че ще имаме такова изчисление, за което $\forall i \in \{0, \dots, |\omega| - 1\} p_{i+1} \in \Delta(p_i, a_{i+1})$, както и $p_{|\omega|} \in F \implies \Delta^*(s, \omega) \cap F \neq \emptyset \implies \omega \in \mathcal{L}(N) \implies \forall \alpha \in \mathcal{L}(S) \alpha \in \mathcal{L}(N) \implies \mathcal{L}(S) \subseteq \mathcal{L}(N)$.

Сега нека по произволен начин изберем думата $\beta \in \mathcal{L}(N) \implies$

$$\exists b_1, \dots, b_{|\beta|} \in \Sigma, t_0, t_1, \dots, t_{|\beta|} \in Q : \beta = \prod_{i=1}^{|\beta|} b_i \wedge t_0 = s$$

$\wedge t_{|\beta|} \in F \wedge \forall i \in \{0, \dots, |\beta| - 1\} t_{i+1} \in \Delta(t_i, b_{i+1})$. Ще покажем, че имаме изчисление по автомата S , за което е изпълнено, че

$$\exists r_0, r_1, \dots, r_{|\beta|} \in Q' : \forall i \in \{0, \dots, |\beta| - 1\} r_{i+1} \in \Delta'(r_i, b_{i+1}) \wedge r_{|\beta|} = f.$$

Полагаме $\forall i \in \{0, \dots, |\beta|\} r_i = replace(t_i)$. Очевидно $r_{|\beta|} = f$, както и

$\forall j \in \{0, \dots, |\beta| - 1\} r_{j+1} \in \Delta'(r_j, b_{j+1})$ също имаме, че

$$r_0 = replace(t_0) = replace(s) \implies \Delta'^*(r_0, \beta) = r_{|\beta|} = f \implies \beta \in \mathcal{L}(S) \implies$$

$$\forall \gamma \in \mathcal{L}(N) \gamma \in \mathcal{L}(S) \implies \mathcal{L}(N) \subseteq \mathcal{L}(S) \implies \mathcal{L}(S) = \mathcal{L}(N) \quad \square$$

6 За всеки език L разпознаван от краен автомат съществува НКА, който разпознава L^{rev}

Нека БОО $N = \langle Q, \Sigma, \Delta, s, F \rangle$ е НКА. Ако ни е даден ДКА то можем да използваме естествения изоморфизъм между Q и $\{\{q\} \mid q \in Q\}$, който на състояние $q \in Q$ съпоставя неговия синглетон, тоест $q \mapsto \{q\}$. От горното твърдение е ясно и че БОО можем да поискаме $|F| = 1$, тоест $F = \{f\}$. Сега ще построим автомат R , такъв че $\mathcal{L}(R) = \mathcal{L}(N)^{rev}$. Нека $R = \langle Q, \Sigma, \nabla, f, \{s\} \rangle$, където $\nabla = \{((q, a), \{p \in Q \mid q \in \Delta(p, a)\}) \mid q \in Q, a \in \Sigma\}$.

Идеята ни този път се базира на факта, че началното състояние е единствено, тогава ако и крайното е единствено то за дадена дума ω от $\mathcal{L}(N)$ можем да разглеждаме изчислението като път в граф и гледайки го на обратно получаваме, че $\omega^{rev} \in \mathcal{L}(N)^{rev}$. Тоест обръщайки стрелките в графа на автомата N и разменяйки ролята на s и f получаваме автомат за $\mathcal{L}(N)^{rev}$, той обаче в повечето случаи ще бъде недетерминиран, защото е възможно различни състояния с една и съща буква да са отивали в едно и също, тогава в новия автомат от това състояние с една и съща буква ще има преход към различни състояния.

Нека вземем произволна дума $\omega \in \mathcal{L}(R) \iff$

$$\exists a_1, \dots, a_{|\omega|} \in \Sigma, q_0, q_1, \dots, q_{|\omega|} \in Q : \omega = \prod_{i=1}^{|\omega|} a_i \wedge q_0 = f$$

$$\wedge q_{|\omega|} = s \wedge \forall i \in \{0, \dots, |\omega| - 1\} q_{i+1} \in \nabla(q_i, a_{i+1})$$

$$\iff \forall i \in \{1, \dots, |\omega|\} q_{i-1} \in \Delta(q_i, a_i)$$

$$\iff f = q_0 \in \Delta^*(q_{|\omega|}, a_{|\omega|} \dots a_1) = \Delta^*(s, \omega^{rev})$$

$$\iff \omega^{rev} \in \mathcal{L}(N) \iff (\omega^{rev})^{rev} = \omega \in \mathcal{L}(N)^{rev}$$

$$\iff \forall \alpha \in \mathcal{L}(R) \iff \alpha \in \mathcal{L}(N)^{rev} \iff \mathcal{L}(R) = \mathcal{L}(N)^{rev} \quad \square$$

7 За всеки НКА съществува еквивалентен на него ДКА

Нека $N = \langle Q, \Sigma, \Delta, s, F \rangle$ е НКА и нека

$$D = \langle 2^Q, \Sigma, \delta, \{s\}, \{P \subseteq Q \mid P \cup F \neq \emptyset\} \rangle, \text{ където}$$

$$\delta = \{((P, a), \bigcup_{p \in P} \Delta(p, a)) \mid P \subseteq Q, a \in \Sigma\}.$$

Тогава $\mathcal{L}(D) = \mathcal{L}(N)$.

От дефиницията на $\delta : 2^Q \times \Sigma \rightarrow 2^Q$ е ясно че тя е тотална функция следователно D е тотален ДКА.

8 Еквивалентност на класа от автоматни езици с класа на регулярни езици

8.1 Теорема на Клини (За всеки автоматен език съществува регулярен израз, който го описва)

Нека $A = \langle Q, \Sigma, \Delta, s, F \rangle$ е НКА. Тогава съществува регулярен израз r , такъв че $\mathcal{L}(r) = \mathcal{L}(A)$.

Доказателство:

Нека фиксираме наредба на състоянията (те са крайно множество), тоест нека $Q = \{q_1, \dots, q_{|Q|}\}$. Нека също фиксираме наредба на азбуката (отново крайно множество), тоест нека $\Sigma = \{a_1, \dots, a_{|\Sigma|}\}$. Нека

$$\begin{aligned} \forall i, j \in \{1, \dots, |Q|\}, k \in \{0, 1, \dots, |Q|\} \quad L(i, j, k) = \\ = \mathcal{L}(\langle \{q_i, q_j\} \cup \{q_1, \dots, q_k\}, \Sigma, \{((q_i, a), \Delta(q_i, a) \cap \{q_1, \dots, q_k\}) \mid a \in \Sigma\} \cup \\ \{((q_c, a), \Delta(q_c, a) \cap (\{q_1, \dots, q_k\} \cup \{q_j\}) \mid a \in \Sigma, c \in \{1, \dots, k\}), q_i, \{q_j\} \rangle). \end{aligned}$$

Тогава $\mathcal{L}(A) = \{\omega \in \Sigma^* \mid \Delta^*(q_1, \omega) \cap F \neq \emptyset\} = \bigcup_{q_i \in F} L(1, i, |Q|)$. Ще докажем по

индукция, че

$\forall i, j \in \{1, \dots, |Q|\}, k \in \{0, 1, \dots, |Q|\} \quad L(i, j, k)$ е регулярен, тогава $\mathcal{L}(A)$ също ще бъде регулярен, защото е обединение на краен брой регулярни (точно $|F|$ на брой).

База ще докажем, че $\forall i, j \in \{1, \dots, |Q|\} \quad L(i, j, 0)$ е регулярен.

Имаме два случая $i = j$ и $i \neq j$. Ако $i = j$ то $L(i, j, 0) = L(i, i, 0) = \{\varepsilon\} \cup \{a \in \Sigma \mid q_i \in \Delta(q_i, a)\}$. В противен случай $i \neq j$ и тогава $L(i, j, 0) = \{a \in \Sigma \mid q_j \in \Delta(q_i, a)\}$. И в двата случая езиците, които получаваме са крайни и следователно са регулярни.

Правим индукционна хипотеза, че

$\exists p \in \mathbb{N} : \forall i, j \in \{1, \dots, |Q|\}, k \in \mathbb{N} : k \leq p \quad L(i, j, k)$ е регулярен.

Съобразяваме, че $\forall i, j \in \{1, \dots, |Q|\} \quad L(i, j, p+1) = L(i, j, p) \cup L(i, p+1, p) \cdot L(p+1, p+1, p)^* \cdot L(p+1, j, p)$. Използваме индукционното предположение според, което езиците $L(i, j, p)$, $L(i, p+1, p)$, $L(p+1, p+1, p)$, $L(p+1, j, p)$ са регулярни. Тогава от съображението ни $L(i, j, p+1)$ се представя само чрез операции, относно които регулярните езици са затворени приложени върху краен брой регулярни езици, следователно $L(i, j, p+1)$ също е регулярен. Което значи, че $\forall i, j \in \{1, \dots, |Q|\}, k \in \{0, 1, \dots, |Q|\} \quad L(i, j, k)$ е регулярен. Това означава и че $\mathcal{L}(A)$ е регулярен. След като е регулярен съществува регулярен израз, който го описва. Ще построим един такъв.

Видяхме, че $\forall i, j \in \{1, \dots, |Q|\} \quad L(i, j, 0)$ представляват или само крайно множество от букви или крайно множество от букви и празната дума. Тогава ще дефинираме функция, която приема подмножество на Σ и връща регулярен израз за него. Нека

$\forall i \in \{1, \dots, |\Sigma|\} \quad \text{index}(a_i) = i$. Тоест index е естествената биекция между първите $|\Sigma|$ естествени числа и буквите от Σ , спрямо наредбата, която фиксирахме. Нека $\forall S \in 2^\Sigma \setminus \{\emptyset\} \quad \text{MinLetter}(S) = \text{index}^{-1}(\min\{\text{index}(l) \mid l \in S\})$. Ясно е, че ако S е непразно подмножество на Σ , MinLetter връща буквата с най-

малък номер във фиксираната наредба. Нека

$$\forall S \in 2^{\Sigma} \text{CharsRegExp}(S) =$$

$$= \begin{cases} \emptyset & , S = \emptyset \\ a & , S = \{a\} \\ \text{CharsRegExp}(\text{MinLetter}(S)) + \text{CharsRegExp}(S \setminus \{\text{MinLetter}(S)\}) & , |S| > 1 \end{cases}$$

Това е ясно, че ако $|S| < 2$ $\mathcal{L}(\text{CharsRegExp}(S)) = S$. Ако $|S| > 1$ тогава $\text{CharsRegExp}(S) = \text{CharsRegExp}(\text{MinLetter}(S)) + \text{CharsRegExp}(S \setminus \{\text{MinLetter}(S)\})$ очевидно CharsRegExp е терминираща рекурсия и връща коректен резултат. Терминираща е, защото връща извикване към себе си със синглетон $\text{CharsRegExp}(\text{MinLetter}(S))$, конкатенира го със символа за обединение и след това отново прави извикване към себе си, но с аргумент, множество с мощност единица по-малко, следователно на всяка стъпка мощността намалява с единица, което значи, че тя ще стигне до 1 тоест наистина рекурсията терминира. На всяка стъпка се освен без последната, на която рекурсията терминира резултата от две извиквания се конкатенира със знака за обединение, което значи, че накрая всички букви от S са разделени със знак за обединение, което означава, че целият израз е коректен и описва точно множеството S , тоест $S = \mathcal{L}(\text{CharsRegExp}(S))$.

Сега ще дефинираме рекурсивна функция, която да ни върне регулярен израз, който описва

$$\forall i, j \in \{1, \dots, |Q|\}, k \in \{0, 1, \dots, |Q|\} L(i, j, k). \text{ Нека}$$

$$\forall i, j \in \{1, \dots, |Q|\}, k \in \{0, 1, \dots, |Q|\} \sigma(i, j, k) =$$

$$= \begin{cases} \varepsilon + \text{CharsRegExp}(L(i, i, 0) \setminus \{\varepsilon\}) & , k = 0 \wedge i = j \\ \text{CharsRegExp}(L(i, j, 0)) & , k = 0 \wedge i \neq j \\ (\sigma(i, j, k-1)) + (\sigma(i, k, k-1)).(\sigma(k, k, k-1))^* . (\sigma(k, j, k-1)) & , k \neq 0 \end{cases}$$

Очевидно σ е терминираща рекурсия и коректна, тоест

$$\forall i, j \in \{1, \dots, |Q|\}, k \in \{0, 1, \dots, |Q|\} L(i, j, k) = \mathcal{L}(\sigma(i, j, k)). \sigma \text{ имплементира рекурсия базирана на индукцията, с която показахме, че всеки от езиците } L(i, j, k) \text{ е регулярен.}$$

Остава ни да дефинираме рекурсивна функция, която да ни върне регулярен израз описващ $\mathcal{L}(A)$. Нека

$$\forall S \in 2^{\{1, \dots, |Q|\}} \text{SumRegExp}(S) =$$

$$= \begin{cases} \emptyset & , S = \emptyset \\ \sigma(1, j, |Q|) & , S = \{j\} \\ \text{SumRegExp}(\min(S)) + \text{SumRegExp}(S \setminus \{\min(S)\}) & , |S| > 1 \end{cases}$$

Базирайки се на наблюденията ни относно CharsRegExp SumRegExp е терминираща рекурсия и коректно връща регулярен израз описващ езика

$$\mathcal{L}(\text{SumRegExp}(S)) = \bigcup_{s \in S} L(1, s, |Q|). \text{ Тогава}$$

$$\mathcal{L}(A) = \mathcal{L}(\text{SumRegExp}(\{j \in \{1, \dots, |Q|\} \mid q_j \in F\})).$$

8.2 За всеки регулярен израз съществува краен автомат, който описва същия език

8.2.1 Изконни езици

8.2.2 Затвореност относно конкатенация

8.2.3 Затвореност относно обединение

8.2.4 Затвореност относно операцията звезда на Клини

8.2.5 Заключение

9 Лема за покачването (за регулярни езици)

9.1 Следствия

9.1.1 Проверка дали един регулярен език е празен

9.1.2 Проверка дали два регулярни езика съвпадат

9.1.3 Проверка дали един регулярен език е краен или безкраен

10 Теорема на Майхил-Нероуд

11 Алгоритъм за минимализация на автомат