Езици и автомати

Иво Стратев

7 февруари 2018 г.

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

Нека $A =< Q,\ \Sigma,\ \delta,\ s,\ F >$ е нетотален ДКА и нека $q_e \notin Q$. Тогава
$A' =< Q \cup \{q_e\},\ \Sigma,\ \delta',\ s,\ F >$ 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 дефиницята 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 функия 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

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

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

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

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

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

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

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

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

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

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

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

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

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

Следствие: БОО можем да разглеждаме само тотални ДКА.

2 Затвореност на езиците разпознавани от ДКА относно операцията обединие

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

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

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 излисления завърши във финално състояние.

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

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

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 разпознавни от ДКА относно операцията сечение

Нека Σ е азбука и $A_1 =< Q_1,\ \Sigma,\ \delta_1,\ s_1,\ F_1 >$ и $A_2 =< Q_2,\ \Sigma,\ \delta_2,\ s_2,\ F_2 >$ са тотални ДКА. Тогава нека $A =< Q_1 \times Q_2,\ \Sigma,\ \delta,\ (s_1,\ s_2),\ F_1 \times F_2 >$.
$\delta = \{(((q_1,\ q_2),\ a),\ (\delta_1(q_1,\ a),\ \delta_2(q_2,\ a))) \mid q_1 \in Q_1,\ q_2 \in Q_2,\ a \in \Sigma\}$

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

Нека $A =< Q,\ \Sigma,\ \delta,\ s,\ F >$ е тотален ДКА, знаем че БОО можем да го изискаме, 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 $A' =< Q,\ \Sigma,\ \delta,\ s,\ Q \backslash F >$. 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
$\overline{\mathcal{L}(A)} = \{\omega \in \Sigma^* \mid \delta^*(s,\ \omega) \notin F\} = \{\omega \in \Sigma^* \mid \delta^*(s,\ \omega) \in Q \backslash F\}$

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

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

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

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

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

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

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

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

7 За всеки НКА съществува еквивалентен на него 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)

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