Софийски университет "Св. Климент Охридски"
Факултет по математика и информатика
Катедра Математическа логика и приложенията ѝ

Директна конструкция на бимашина по контекстно правило за заместване

Дипломна работа на Иван Петров Пейков
ф.н. М-21185

Ръководител катедра:
проф. дмн. Иван Сосков

Научен ръководител:
ст.н.с. II ст. д-р Стоян Михов

— София, 2006г. —

Съдържание

1 Увод **3**

2 Контексти и правила за заместване **4**

3 Крайни автомати **8**

4 Детерминистични крайни автомати **12**

5 Бимашини **15**

6 Директна конструкция **18**
6.1 Левият автомат 19
6.2 Десният автомат 19
6.3 Изходната функция 19
6.4 Бимашина за контекстно правило 23

7 Алгоритми **27**
7.1 Конкатенация отляво 27
7.2 Превод на бимашина, работеща върху изхода си 28
7.3 Конструкция на бимашина по контекстно правило 29

8 Сложност **31**

1. Увод

Контекстните правила за заместване са добре познат формализъм, широко използван в много области на компютърната лингвистика. За пръв път намират приложение в трудовете на Чомски ([1]) и както се оказва по-късно са достатъчно изразителни, за да моделират успешно множество езикови феномени.

През 1972г. Джонсън ([2]) забелязва, че с ограничението да работят единствено върху входа си, контекстните правила за заместване стават еквивалентни по изразителност на регулярните релации. Този чисто теоретичен резултат е потвърден по-късно от Каплан и Кей ([3]), които показват практическата приложимост на контекстните правила за заместване, реализирайки ги с крайни преобразуватели. Това дава тласък на множество разработки в областта и остава едно от съществените постижения в съвременната компютърна лингвистика.

Бимашините, въведени от Шутценберже ([4]), са детерминистични абстрактни машини, еквивалентни по изразителност на регулярните функции, които по една или друга причина остават сравнително неизследвани до наши дни. Специфично за тези машини е, че обработват подаден входен текст едновременно отляво надясно и отдясно наляво. На всяка позиция се извежда изход, зависещ както от прочетеното вляво от текущата позиция, така и от прочетеното вдясно от нея. Бимашините работят линейно спрямо дължината на входния текст, което ги прави особено ефективни и приложими на практика.

Целта на настоящата работа е да представи конструкция, която по дадено контекстно правило за заместване строи директно бимашина, реализираща съответната му регулярната функция.

Известни са конструкции, строящи краен преобразувател за контекстно правило ([3]) и бимашина от краен преобразувател ([5]). През 2004г. Скут ([6]) и др. показват директна конструкция на бимашина по ограничено контекстно правило. Доколкото ни е известно, настоящата работа показва първата до момента директна конструкция на бимашина, която не налага никакви ограничения върху контекстното правило.

Дипломната работа е структурирана както следва. В Глава 2 въвеждаме контекстните правила за заместване и формализираме задачата. В Глави 3 и 4 представяме някои означения и свойства, свързани с крайните автомати. В Глава 5 дефинираме бимашина и изследваме две от възможните ѝ операционни семантики. В Глава 6 описваме директната конструкция на бимашина по контекстно правило и показваме нейната коректност. В оставащите Глави 7 и 8 излагаме някои от използваните алгоритми и изследваме сложността на конструкцията.

Ще означаваме конкатенацията на две думи α и β с $\alpha \cdot \beta$ или директно с $\alpha\beta$, когато смисълът е ясен от контекста. Ще означаваме факта, че α е поддума на β, пишейки $\alpha \subset \beta$ или $\alpha \subseteq \beta$, в случай че се допуска равенство. Всички останали означения ще въвеждаме при нужда или ще смятаме за достатъчно широко приети.

2. Контексти и правила за заместване

Нека фиксираме крайна азбука Σ. Правила от вида

$$E \to \beta / L_R \tag{1}$$

където $L, E, R \in \mathcal{RE}(\Sigma)$ са регулярни изрази над Σ, а $\beta \in \Sigma^*$, наричаме контекстни правила за заместване. Приложение на правилото върху дума $\alpha \in \Sigma$ следва да се интерпретира като едновременно заместване с β на поддумите на α, които принадлежат на езика $\mathcal{L}(E)$ на регулярния израз E и се намират между поддума от $\mathcal{L}(L)$ и поддума от $\mathcal{L}(R)$:

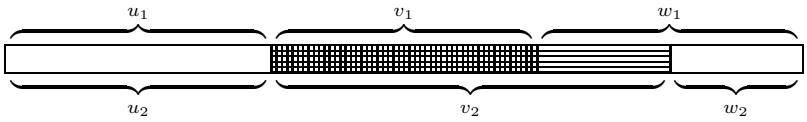
$$\alpha_1\alpha_2\ldots\alpha_{i-1}\underbrace{\alpha_i\ldots\alpha_{j-1}}_{\in\mathcal{L}(L)}\overbrace{\underbrace{\alpha_j\ldots\alpha_{k-1}}_{\in\mathcal{L}(E)}}^{\beta}\underbrace{\alpha_k\ldots\alpha_{l-1}}_{\in\mathcal{L}(R)}\alpha_l\ldots\alpha_{n-1}\alpha_n$$

Дефиниция 2.1. *Нека $t \in \Sigma^*$, а $L, E, R \in \mathcal{RE}(\Sigma)$. Всяка тройка $\langle u, v, w\rangle$, такава, че $u \in \mathcal{L}(\Sigma^* L)$, $v \in \mathcal{L}(E)$, $w \in \mathcal{L}(R\Sigma^*)$ и $uvw = t$, наричаме контекст за заместване в t с префикс от L, фокус от E и суфикс от R. Думите u, v и w наричаме съответно префикс, фокус и суфикс на контекста. Означаваме с $\mathcal{C}(t; L, E, R)$*

$$\mathcal{C}(t; L, E, R) = \{\langle u, v, w\rangle | u \in \mathcal{L}(\Sigma^* L)\ \&\ v \in \mathcal{L}(E)\ \&\ w \in \mathcal{L}(R\Sigma^*)\ \&\ t = uvw\}$$

множеството от всички такива контексти. Когато t, L, E и R се подразбират, ще пишем само $\mathcal{C}$.

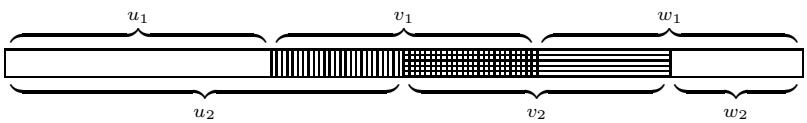
От дотук казаното става ясно, че при прилагането на правила за заместване могат да възникнат определени многозначности. Например, ако два контекста имат един и същи префикс, но различни фокуси (и съответно суфикси), заместването не е еднозначно определено:



Такъв тип многозначности ще разрешаваме, избирайки контекста с най-дълъг фокус.

Пример 2.1. *Да разгледаме правилото $a^+ \to A$ / b_a и думата $\alpha =$ baaaab. Поради възникналата многозначност, резултатът от приложението на правилото върху α може да е bAaaab, bAaab или bAab (замествайки съответно в контекст $\langle b, a, aaab\rangle$, $\langle b, aa, aab\rangle$ или $\langle b, aaa, ab\rangle$). Уславяйки се да избираме винаги контекста с най-дълъг фокус, разрешаваме тази многозначност и дефинираме резултата от приложението на правилото върху α да е точно bAab.*

Дефиниция 2.2. *Казваме, че два контекста $\langle u_i, v_i, w_i\rangle \in \mathcal{C}$ за $i = 1, 2$ се застъпват, и пишем $\langle u_1, v_1, w_1\rangle \prec \langle u_2, v_2, w_2\rangle$, ако $u_1 \subset u_2 \subset u_1 v_1$.*



При застъпващи се контексти заместването по контекстно правило трябва да се извърши в точно единия от двата контекста. Ще разрешаваме такива многозначности, избирайки винаги най-левия възможен контекст и игнорирайки тези, които се застъпват с него.

Такава стратегия за разрешаване на многозначностите се нарича заместване на "най-ляво, най-дълго срещане".

Пример 2.2. *Да разгледаме правилото* $xy|yz \rightarrow \epsilon \ / \ x_z$. *Приложението му върху дума като* $xyzzxxyzz$ *поражда контекстите* $\langle x, yz, zxxyzz\rangle$, $\langle xyzzx, xy, zz\rangle$ *и* $\langle xyzzxx, yz, z\rangle$, *вторите два от които се застъпват. Стратегията "най-ляво, най-дълго срещане"определя първите два контекста като валидни за заместване, а резултатът от заместването е точно* $xzxzz$.

Съществуват и други стратегии за заместване. Ще се концентрираме върху реализирането на гореописаната стратегия, защото смятаме, че тя е най-естествената и най-често приложима в практиката.

Нека сега дефинираме формално множеството от контекстите, които ще смятаме *валидни* за заместване след прилагането на гореописаната стратегия за разрешаване на многозначностите.

Дефиниция 2.3. *Дефинираме следните оператори над множества от контексти:*

$$\begin{array}{lcl} \texttt{OVER}(C, C') & = & \{\langle u, v, w\rangle_{\in C} | (\exists \langle u', v', w'\rangle_{\in C'})(\langle u', v', w'\rangle \prec \langle u, v, w\rangle)\} \\ \texttt{LEFT}(C) & = & \{\langle u, v, w\rangle_{\in C} | \neg(\exists \langle u', v', w'\rangle_{\in C})(u' \subset u)\} \\ \texttt{LONG}(C) & = & \{\langle u, v, w\rangle_{\in C} | \neg(\exists \langle u\ , v', w'\rangle_{\in C})(v \subset v')\} \end{array}$$

Първият оператор `OVER` определя всички контексти от дадено множество, които се застъпват от произволен контекст, принадлежащ на друго множество. Вторият оператор `LEFT` определя *най-левите* контексти в дадено множество.

Имайки горните дефиниции можем да пристъпим към формалната дефиниция на валидни контексти. Строим редица $\{\mathcal{C}_i\}_{i=0}^{\infty}$, където $\mathcal{C}_i \subseteq \mathcal{C}$ за всяко $i \geq 0$ чрез следната индуктивна дефиниция:

- $\mathcal{C}_0 = \emptyset$
- $\mathcal{C}_{i+1} = C_i \cup \texttt{LEFT}(\mathcal{C} - \mathcal{C}_i - \texttt{OVER}(\mathcal{C}, \mathcal{C}_i))$

Вижда се, че това е монотонно растяща редица от множества, следователно има точна горна граница. Означаваме $\mathcal{C}_v = \cup_{i=0}^{\infty}\mathcal{C}_i$ и наричаме $\mathcal{C}_v$ множество на валидните контексти. Означаваме $\mathcal{C}_l = \texttt{LONG}(\mathcal{C}_v)$ и наричаме $\mathcal{C}_l$ множество на най-дългите валидни контексти.

Твърдение 2.1. $\mathcal{C}_v \subseteq \mathcal{C}$

Твърдение 2.2. $\mathcal{C}_v$ *съдържа само незастъпващи се контексти*

Доказателство. Да допуснем, че $\langle u_i, v_i, w_i\rangle \in \mathcal{C}_v$ за $i = 1, 2$ и $\langle u_1, v_1, w_1\rangle \prec \langle u_2, v_2, w_2\rangle$. Тогава $u_1 \subset u_2 \subset u_1 v_1$. Нека k_i са най-малките индекси, за които съответно $\langle u_i, v_i, w_i\rangle \in \mathcal{C}_{k_i}$ (такива непременно има, защото сме приели, че $\langle u_i, v_i, w_i\rangle \in \mathcal{C}_v = \cup_{k=0}^{\infty}\mathcal{C}_k$).

Нека приемем, че $k_1 \geq k_2$. Тогава

$$\langle u_2, v_2, w_2\rangle \in \mathcal{C}_{k_2} = \mathcal{C}_{k_2-1} \cup \texttt{LEFT}(\mathcal{C} - \mathcal{C}_{k_2-1} - \texttt{OVER}(\mathcal{C}, \mathcal{C}_{k_2-1}))$$

Да означим с $R = \mathcal{C} - \mathcal{C}_{k_2-1} - \texttt{OVER}(\mathcal{C}, \mathcal{C}_{k_2-1})$. Тъй като $\langle u_2, v_2, w_2\rangle \notin \mathcal{C}_{k_2-1}$, значи $\langle u_2, v_2, w_2\rangle \in \texttt{LEFT}(R)$. От друга страна, тъй като $k_1 \geq k_2$, то $\langle u_1, v_1, w_1\rangle \in R$. Това ни води до противоречие с дефиницията на `LEFT`, и следователно $k_1 < k_2$.

След като установихме, че $k_1 < k_2$, можем да заключим, че $\langle u_2, v_2, w_2\rangle \notin \mathcal{C}_k$ за $k \leq k_1$. При $k \geq k_1$, $\langle u_1, v_1, w_1\rangle \in \mathcal{C}_k$ и значи $\langle u_2, v_2, w_2\rangle \in \texttt{OVER}(\mathcal{C}, \mathcal{C}_k)$, т.е. $\langle u_2, v_2, w_2\rangle \notin \mathcal{C}_{k+1}$. Така получихме противоречие с факта, че $\langle u_2, v_2, w_2\rangle \in \mathcal{C}_{k_2}$ $(k_2 > k_1)$, и следователно в $\mathcal{C}_v$ няма застъпващи се контексти. □

Твърдение 2.3. *Нека $\langle u, v, w\rangle \in \mathcal{C}$ е невалиден контекст ($\langle u, v, w\rangle \notin \mathcal{C}_v$). Тогава има валиден контекст ($\langle u_0, v_0, w_0\rangle \in \mathcal{C}_v$), такъв, че $\langle u_0, v_0, w_0\rangle \prec \langle u, v, w\rangle$*

Доказателство. Да разгледаме редицата $\{\mathcal{O}_i\}_{i=0}^{\infty}$, където $\mathcal{O}_i = \texttt{OVER}(\mathcal{C}, \mathcal{C}_i)$. Лесно се проверява, че това е монотонно растяща редица, чиято точна горна граница е именно $\texttt{OVER}(\mathcal{C}, \mathcal{C}_v)$.

Ще проверим, че $\mathcal{C}_v \cup \texttt{OVER}(\mathcal{C}, \mathcal{C}_v) = \mathcal{C}$. И наистина, да допуснем, че съществува контекст $\langle u, v, w\rangle \in \mathcal{C}$, такъв, че $\langle u, v, w\rangle \notin \mathcal{C}_v$ и $\langle u, v, w\rangle \notin \texttt{OVER}(\mathcal{C}, \mathcal{C}_v)$, и да го изберем с минимален суфикс. Тъй като $\mathcal{C}$ е крайно множество, има индекс k, за който $\mathcal{C}_k = \mathcal{C}_v$. Понеже $\langle u, v, w\rangle \notin \texttt{OVER}(\mathcal{C}, \mathcal{C}_k)$, следва, че $\mathcal{C} - \mathcal{C}_k - \texttt{OVER}(\mathcal{C}, \mathcal{C}_k) \neq \emptyset$, и следователно $\mathcal{C}_v \subset \mathcal{C}_{k+1}$ не е точна горна граница на редицата $\{\mathcal{C}_i\}$. Това е противоречие, и следователно $\mathcal{C}_v \cup \texttt{OVER}(\mathcal{C}, \mathcal{C}_v) = \mathcal{C}$.

Сега да се върнем на твърдението. Нека $\langle u, v, w\rangle \notin \mathcal{C}_v$. Тогава (според току що направените разсъждения) $\langle u, v, w\rangle \in \texttt{OVER}(\mathcal{C}, \mathcal{C}_v)$, и следователно има индекс p ($\texttt{OVER}(\mathcal{C}, \mathcal{C}_v)$ е точна горна граница на $\{\mathcal{O}_i\}$), за който $\langle u, v, w\rangle \in \mathcal{O}_p$. Тогава по дефиницията на $\{\mathcal{O}_i\}$ следва, че има $\langle u_0, v_0, w_0\rangle \in \mathcal{C}_p$, за който $\langle u_0, v_0, w_0\rangle \prec \langle u, v, w\rangle$. □

Твърдение 2.4. *Нека $\langle u, v_i, w_i\rangle \in \mathcal{C}$ са контексти ($i = 1, 2$). Тогава $\langle u, v_1, w_1\rangle$ е валиден точно тогава, когато и $\langle u, v_2, w_2\rangle$ е валиден.*

Доказателство. Нека $\langle u, v_1, w_1\rangle$ е валиден контекст и да допуснем, че $\langle u, v_2, w_2\rangle$ е невалиден. Тогава съществува $k \geq 0$, за което $\langle u, v_2, w_2\rangle \in \texttt{OVER}(\mathcal{C}, \mathcal{C}_k)$. Но това означава, че има $\langle u_0, v_0, w_0\rangle \in \mathcal{C}_k$, за който $u_0 \subset u \subset u_0 v_0$. Следователно $\langle u_0, v_0, w_0\rangle \prec \langle u, v_1, w_1\rangle$. Но според Твърдение 2.2 в $\mathcal{C}_v$ няма застъпващи се контексти, което е противоречие с допускането ни. □

Следва дефиниция на резултат от заместването по контекстно правило в даден входен текст.

Дефиниция 2.4. *Нека* $\alpha \in \Sigma^*$ *и* $E \to \beta \ / \ L_R$ *е контекстно правило за заместване. Резултат от приложението на* $E \to \beta \ / \ L_R$ *върху* α *наричаме думата*

$$u_1 \cdot \beta \cdot (u_1 v_1)^{-1} u_2 \cdot \beta \cdot (u_2 v_2)^{-1} u_3 \cdot \beta \cdot \ldots (u_{k-1} v_{k-1})^{-1} u_k \cdot \beta \cdot w_k$$

където $\mathcal{C}_l(\alpha; L, E, R) = \{\langle u_i, v_i, w_i \rangle | i = 1, \ldots, k\}$ *и* $i < j \to u_i \subset u_j$.

Тъй като Дефиниция 2.4 би усложнила прекалено много разглежданията ни, ще въведем еквивалентната на нея

Дефиниция 2.5. *Нека* $\alpha = a_1 a_2 \ldots a_n \in \Sigma^*$ *и* $E \to \beta \ / \ L_R$ *е контекстно правило за заместване. Резултат от приложението на* $E \to \beta \ / \ L_R$ *върху* α *наричаме думата* $\omega_1 \cdot \omega_2 \ldots \cdot \omega_n \cdot \omega_{n+1}$, *където за* $i = 1, 2, \ldots, n$

$$\omega_i = \begin{cases} \epsilon & \text{ако съществува } \langle u, v, w \rangle \in \mathcal{C}_v(\alpha; L, E, R),\ |u| < i - 1 < |uv| \\ \beta & \text{ако съществува } \langle u, v, w \rangle \in \mathcal{C}_v(\alpha; L, E, R),\ |u| = i - 1 < |uv| \\ \beta a_i & \text{ако съществува } \langle u, v, w \rangle \in \mathcal{C}_v(\alpha; L, E, R),\ |u| = i - 1 = |uv| \\ & \text{и не съществува } \langle u, v', w' \rangle \in \mathcal{C}_v(\alpha; L, E, R), \text{ такъв, че } v' \neq \epsilon \\ a_i & \text{в противен случай} \end{cases}$$

а $\omega_{n+1} \in \Sigma^*$ *е дефинирана както следва*

$$\omega_{n+1} = \begin{cases} \beta & \text{ако } \langle \alpha, \epsilon, \epsilon \rangle \in \mathcal{C}_v(\alpha; L, E, R) \\ \epsilon & \text{в противен случай} \end{cases}$$

За да опростим някои от доказателствата, ще направим преработка на Дефиниция 2.5 и ще изведем

Дефиниция 2.6. *Нека* $\alpha = a_1 a_2 \ldots a_n \in \Sigma^*$ *и* $E \to \beta \ / \ L_R$ *е контекстно правило за заместване. Резултат от приложението на* $E \to \beta \ / \ L_R$ *върху* α *наричаме думата* $(\omega_1 \pi_1) \cdot (\omega_2 \pi_2) \ldots \cdot (\omega_n \pi_n)$, *където* ω_i *са дефинирани както в Дефиниция 2.5, а*

$$\pi_i = \begin{cases} \beta & \text{ако } i = n \text{ и } \langle \alpha, \epsilon, \epsilon \rangle \in \mathcal{C}_v(\alpha; L, E, R) \\ \epsilon & \text{в противен случай} \end{cases}$$

Доказателството, че Дефиниции 2.4, 2.5 и 2.6 са еквивалентни, е тривиално и прекалено техническо, за да представлява интерес в настоящата работа. Оттук нататък ще използваме в разглежданията си само Дефиниция 2.6.

3. Крайни автомати

Дефиниция 3.1. *Краен автомат (КА) наричаме петорка от вида*

$$\mathcal{A} = \langle \Sigma, Q, S, F, \Delta \rangle$$

където Σ е крайна азбука, Q е множество от състояния, $S \subseteq Q$ е множество от начални състояния, $F \subseteq Q$ е множество от финални състояния, а $\Delta \subseteq Q \times \Sigma \times Q$ е релация на прехода. Разширяваме индуктивно Δ до Δ^*

- $\langle q, \epsilon, q \rangle \in \Delta^*$ *за всяко* $q \in Q$
- $\langle q_1, \alpha a, q_2 \rangle \in \Delta^*$ *ако има* $q \in Q$, *такова, че* $\langle q_1, \alpha, q \rangle \in \Delta^*$ *и* $\langle q, a, q_2 \rangle \in \Delta$

Дефиниция 3.2. *Нека $\Delta \subseteq Q \times \Sigma \times Q$, $T \subseteq Q$ и $a \in \Sigma$. За удобство въвеждаме следната нотация:*

$$T \xrightarrow{\Delta, a} \{q | (\exists q'_{\in T})(\langle q', a, q \rangle \in \Delta)\}$$

Дефиниция 3.3. *Нека $\mathcal{A}$ е КА и $\alpha \in \Sigma^*$ е дума. Казваме, че редицата $q_0, q_1, \ldots q_{|\alpha|}$ е изпълнение на $\mathcal{A}$ върху α, ако $q_i \in Q$ за $i = 0 \ldots |\alpha|$, $q_0 \in S$ и $\langle q_{i-1}, \alpha_i, q_i \rangle \in \Delta$ за $i = 1 \ldots |\alpha|$. Едно такова изпълнение наричаме успешно, ако $q_{|\alpha|} \in F$ е финално. В случаите, когато изложението би било по-ясно, ще използваме алтернативна нотация за изпълнение, а именно $\sigma : [0, |\alpha|] \to Q$. Казваме, че $\mathcal{A}$ разпознава α, ако съществува успешно изпълнение на $\mathcal{A}$ върху α.*

Дефиниция 3.4 (Език на КА)**.** *Нека $\mathcal{A}$ е КА. Език на $\mathcal{A}$ наричаме множеството $\mathcal{L}(\mathcal{A}) = \{\alpha | \mathcal{A}$ разпознава $\alpha\}$.*

Дефиниция 3.5 (Еквивалентност на КА)**.** *Нека $\mathcal{A}_{1,2}$ са КА. Казваме, че $\mathcal{A}_1$ е еквивалентен на $\mathcal{A}_2$ (и пишем $\mathcal{A}_1 \equiv \mathcal{A}_2$), ако $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$.*

Дефиниция 3.6 (Нормална форма)**.** *Нека $\mathcal{A} = \langle \Sigma, Q, S, F, \Delta \rangle$ е КА. Казваме, че $\mathcal{A}$ се намира в нормална форма, ако за всеки преход $\langle q_1, a, q_2 \rangle \in \Delta$, $q_1 \notin F$ и $q_2 \notin S$. Неформално погледнато, нормалността на $\mathcal{A}$ се изразява в това, че не съдържа преходи, излизащи от финални или влизащи в начални състояния.*

Твърдение 3.1. *За всеки КА $\mathcal{A}$, съществува еквивалентен $\mathcal{A}^N$, който е в нормална форма.*

Доказателство. Нека $\mathcal{A} = \langle \Sigma, Q, S, F, \Delta \rangle$ е КА. Ще построим $\mathcal{A}^N$ от $\mathcal{A}$, премахвайки преходите към начални и от финални състояния. Да разгледаме $\mathcal{A}^N = \langle \Sigma, Q^N, S^N, F^N, \Delta^N \rangle$, където

$$Q^N = Q \times \{1\} \cup S \times \{2\} \cup F \times \{3\}$$

Останалите компоненти на $\mathcal{A}^N$ са $S^N = S \times \{2\}$, $F^N = F \times \{3\}$

$$\begin{aligned} \Delta^N \quad = \quad & \{\langle \langle q_1, 1 \rangle, a, \langle q_2, 1 \rangle \rangle | \langle q_1, a, q_2 \rangle \in \Delta\} \cup \\ & \{\langle \langle q_1, 2 \rangle, a, \langle q_2, 1 \rangle \rangle | \langle q_1, a, q_2 \rangle \in \Delta \ \& \ q_1 \in S\} \cup \\ & \{\langle \langle q_1, 1 \rangle, a, \langle q_2, 3 \rangle \rangle | \langle q_1, a, q_2 \rangle \in \Delta \ \& \ q_2 \in F\} \end{aligned}$$

Така построеният $\mathcal{A}^N$ е очевидно в нормална форма. Остава да покажем, че е еквивалентен на $\mathcal{A}$.

Нека $\alpha \in \mathcal{L}(\mathcal{A})$ и $q_0, q_1, \ldots q_{|\alpha|-1}, q_{|\alpha|}$ е успешно изпълнение на $\mathcal{A}$ върху α. Да проверим, че $\langle q_0, 2\rangle, \langle q_1, 1\rangle, \langle q_2, 1\rangle, \ldots, \langle q_{|\alpha|-1}, 1\rangle, \langle q_{|\alpha|}, 3\rangle$ е успешно изпълнение на $\mathcal{A}^N$ върху α. Най-напред $\langle q_0, 2\rangle \in S^N$, защото $q_0 \in S$ и $\langle q_{|\alpha|}, 3\rangle \in F^N$, защото $q_{|\alpha|} \in F$. От построението на Δ^N е ясно, че $\langle\langle q_{i-1}, 1\rangle, \alpha_i, \langle q_i, 1\rangle\rangle \in \Delta^N$ при $i = 2 \ldots |\alpha| - 1$, защото $\langle q_{i-1}, \alpha_i, q_i\rangle \in \Delta$ при $i = 2 \ldots |\alpha| - 1$. Тъй като $\langle q_0, \alpha_0, q_1\rangle \in \Delta$ и $q_0 \in S$, следователно по дефиницията на Δ^N е вярно, че $\langle\langle q_0, 2\rangle, \alpha_0, \langle q_1, 1\rangle\rangle \in \Delta^N$. Аналогично $\langle\langle q_{|\alpha|-1}, 1\rangle, \alpha_{|\alpha|}, \langle q_{|\alpha|}, 3\rangle\rangle \in \Delta^N$. Така показахме, че съществува успешно изпълнение на $\mathcal{A}^N$ върху α, т.е. $\alpha \in \mathcal{L}(\mathcal{A}^N)$.

В обратната посока, доказателството протича аналогично, с което показахме, че $\mathcal{A} \equiv \mathcal{A}^N$. □

Теорема 3.1 (Клини). *За всеки регулярен израз $\mathcal{E}$, съществува КА $\mathcal{A}$, за който $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{E})$.*

Автоматът от Теоремата на Клини не е единствен и има различни конструкции, които директно го строят от $\mathcal{E}$ (например [7]). Тъй като конкретната конструкция няма значение за нашите цели, винаги, когато се налага да построим КА по регулярен израз $\mathcal{E}$, ще пишем $\mathcal{A}(\mathcal{E})$ и ще имаме предвид КА в нормална форма, такъв, че $\mathcal{L}(\mathcal{A}(\mathcal{E})) = \mathcal{L}(\mathcal{E})$.

Дефиниция 3.7 (Огледален КА). *Нека $\mathcal{A} = \langle \Sigma, Q, S, F, \Delta\rangle$ е КА. Автоматът $\widetilde{\mathcal{A}} = \langle \Sigma, Q, F, S, \widetilde{\Delta}\rangle$, където $\widetilde{\Delta} = \{\langle q_2, a, q_1\rangle | \langle q_1, a, q_2\rangle \in \Delta\}$ наричаме огледален на $\mathcal{A}$.*

Твърдение 3.2. *Нека $\mathcal{A} = \langle \Sigma, Q, S, F, \Delta\rangle$ е КА и $\widetilde{\mathcal{A}}$ е неговият огледален. Тогава за всяко $\alpha \in \Sigma^*$, $q_0, q_1, \ldots q_{|\alpha|}$ е успешно изпълнение на $\mathcal{A}$ точно тогава, когато $q_{|\alpha|}, q_{|\alpha|-1}, \ldots q_0$ е успешно изпълнение на $\widetilde{\mathcal{A}}$.*

Доказателство. Очевидно от дефиницията на огледален КА. □

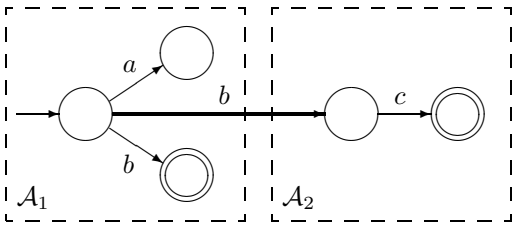
Добре известно е, че класът на регулярните езици е затворен относно конкатенация. Иначе казано, имайки езиците $L_{1,2}$, разпознаваеми с крайни автомати $\mathcal{A}_{1,2}$, съществува краен автомат $\mathcal{A}$, разпознаващ $L_1 \cdot L_2$.

Ще разгледаме две конструкции, които строят конкатенацията на два автомата и имат особеното свойство, че всяко тяхно изпълнение съдържа като подизпълнение съответно изпълнение на първия или втория автомат.

Дефиниция 3.8 (Конкатенация отляво). *Нека $\mathcal{A}_i = \langle \Sigma, Q_i, S_i, F_i, \Delta_i\rangle$, за $i = 1, 2$ са КА и $Q_1 \cap Q_2 = \emptyset$. Казваме, че $\mathcal{A} = \langle \Sigma, Q, S, F, \Delta\rangle$ е получен при конкатенация* **отляво** *на $\mathcal{A}_1$ с $\mathcal{A}_2$ и пишем $\mathcal{A} = \mathcal{A}_1 \cdot_l \mathcal{A}_2$, ако:*

- $Q = Q_1 \cup Q_2$
- $S = \begin{cases} S_1 \cup S_2 & \text{ако } S_1 \cap F_1 \neq \emptyset \\ S_1 & \text{ако } S_1 \cap F_1 = \emptyset \end{cases}$
- $F = F_2$
- $\Delta = \Delta_1 \cup \Delta_2 \cup \{\langle q_1, a, q_2\rangle | (\exists q_{\in F_1})(\langle q_1, a, q\rangle \in \Delta_1)\ \&\ q_2 \in S_2\}$

За да илюстрираме операцията *конкатенация отляво*, нека разгледаме примера от Фиг. 1. Прави впечатление, че всяко успешно изпълнение на така получения автомат $\mathcal{A}_1 \cdot_l \mathcal{A}_2$ съдържа успешно изпълнение на $\mathcal{A}_2$.



Фиг. 1. Конкатенация отляво

Твърдение 3.3. *Нека $\mathcal{A}_{1,2}$ са КА и $\mathcal{A} = \mathcal{A}_1 \cdot_l \mathcal{A}_2$. Тогава всяко успешно изпълнение на $\mathcal{A}$ върху $\overline{a_1 a_2 \dots a_n} \in \Sigma^*$ е от вида*

$$q_0, \dots, q_{p-1}, q_p, \dots, q_n$$

където за някое $p \in [0, n]$, $q_p, \dots, q_n$ е успешно изпълнение на $\mathcal{A}_2$ върху $\overline{a_{p+1} \dots a_n}$ и съществува състояние $q \in F_1$, такова, че $q_0, \dots, q_{p-1}, q$ е успешно изпълнение на $\mathcal{A}_1$ върху $\overline{a_1 \dots a_p}$.

Доказателство. Нека $q_0, \dots, q_n$ е произволно успешно изпълнение на $\mathcal{A}$ върху $\overline{a_1 a_2 \dots a_n}$. След като е успешно, q_n без съмнение е финално състояние, т.е. $q_n \in F = F_2$ (по дефиницията на конкатенация отляво). Нека $p \in [0, n]$ е най-малкият индекс, за който $q_p \in Q_2$ (сигурни сме, че такова p съществува, защото $q_n \in Q_2$). Сега забелязваме, че за произволно $i \in [p, n]$, $q_i \in Q_2$. Това е така, защото ако допуснем противното и изберем най-малкото $i > p$, за което $q_i \in Q_1$ би излязло, че $\langle q_{i-1}, a_i, q_i \rangle \in \Delta$, където $q_{i-1} \in Q_2$, а $q_i \in Q_1$. Както се вижда от конструкцията на $\mathcal{A}$, това е невъзможно.

Следователно $q_0, \dots, q_{p-1} \in Q_1$, а $q_p, \dots, q_n \in Q_2$. Сега ако $p > 0$, $\langle q_{p-1}, a_p, q_p \rangle \in \Delta$ и от дефиницията на Δ е ясно, че $q_p \in S_2$. В противен случай, ако $p = 0$, $q_p \in S_2$, защото $q_0, \dots, q_n$ е изпълнение на $\mathcal{A}$. Поради същата причина се вижда, че $\langle q_{i-1}, a_i, q_i \rangle \in \Delta_2$ (за $i \in [p+1, n]$), и следователно $q_p, \dots, q_n$ е успешно изпълнение на $\mathcal{A}_2$ върху $\overline{a_{p+1} \dots a_n}$.

Ако $p = 0$, то $\overline{a_1 \dots a_p} = \epsilon$. От дефиницията на $\mathcal{A}$ и от факта, че $q_0 \in S \cap S_2 \neq \emptyset$, следва, че $S_1 \cap F_1 \neq \emptyset$. Избираме произволно състояние $q \in S_1 \cap F_1$ и го обявяваме за успешно изпълнение на $\mathcal{A}_1$ върху ϵ. Нека сега $p > 0$. Тогава $q_0 \in Q_1$, и следователно $q_0 \in S_1$. Сега, понеже $\langle q_{i-1}, a_i, q_i \rangle \in \Delta$ за $i \in (0, p)$ и $q_i \in Q_1$ за $i \in [0, p)$, веднага се вижда, че $\langle q_{i-1}, a_i, q_i \rangle \in \Delta_1$ за $i \in (0, p)$, и следователно $q_0, \dots, q_{p-1}$ е изпълнение на $\mathcal{A}_1$ върху $\overline{a_1 \dots a_{p-1}}$. Сега по дефиницията на $\mathcal{A}$ следва, че съществува $q \in F_1$, такова, че $\langle q_{p-1}, a_p, q \rangle \in \Delta_1$, и следователно $q_0, \dots, q_{p-1}, q$ е успешно изпълнение на $\mathcal{A}_1$ върху $\overline{a_1 \dots a_p}$. □

Твърдение 3.4. *Нека $\mathcal{A}_{1,2}$ са КА. Тогава $\mathcal{L}(\mathcal{A}_1 \cdot_l \mathcal{A}_2) = \mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2)$.*

Доказателство. Нека $\mathcal{A}_{1,2} = \langle \Sigma, Q_{1,2}, S_{1,2}, F_{1,2}, \Delta_{1,2} \rangle$ и конкатенацията отляво на $\mathcal{A}_1$ с $\mathcal{A}_2$ е $\mathcal{A} = \mathcal{A}_1 \cdot_l \mathcal{A}_2 = \langle \Sigma, Q, S, F, \Delta \rangle$.

Нека $\alpha \in \mathcal{L}(\mathcal{A})$. Това означава, че съществува успешно изпълнение $q_0, \ldots, q_n$ на $\mathcal{A}$ върху α. Според Твърдение 3.3 съществуват успешни изпълнения на $\mathcal{A}_1$ върху $\overline{a_1 \ldots a_p}$ и на $\mathcal{A}_2$ върху $\overline{a_{p+1} \ldots a_n}$. Това показва съответно, че $\overline{a_1 \ldots a_p} \in \mathcal{L}(\mathcal{A}_1)$ и $\overline{a_{p+1} \ldots a_n} \in \mathcal{L}(\mathcal{A}_2)$, и следователно $\alpha = \overline{a_1 \ldots a_p a_{p+1} \ldots a_n} \in \mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2)$.

Обратно, нека $\alpha \in \mathcal{L}(\mathcal{A}_1)$, а $\beta \in \mathcal{L}(\mathcal{A}_2)$. Тогава има успешни изпълнения $q_{i_0}, \ldots, q_{i_n}$ на $\mathcal{A}_1$ върху α и $q_{j_0}, \ldots, q_{j_m}$ на $\mathcal{A}_2$ върху β. Очевидно $q_{i_0}, \ldots, q_{i_{n-1}}, q_{j_0}, \ldots, q_{j_m}$ ще е успешно изпълнение на $\mathcal{A}$ върху $\alpha\beta$, и следователно $\alpha\beta \in \mathcal{L}(\mathcal{A})$. С това показахме, че $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2)$. □

Аналогично на операцията конкатенация отляво дефинираме и дуалната ѝ — конкатенация отдясно. Пропускаме доказателствата на твърденията, тъй като те не се различават от вече изложените.

Дефиниция 3.9 (Конкатенация отдясно)**.** *Нека $\mathcal{A}_i = \langle \Sigma, Q_i, S_i, F_i, \Delta_i \rangle$, за $i = 1, 2$ са КА и $Q_1 \cap Q_2 = \emptyset$. Казваме, че $\mathcal{A} = \langle \Sigma, Q, S, F, \Delta \rangle$ е получен при конкатенация* **отдясно** *на $\mathcal{A}_1$ с $\mathcal{A}_2$, и пишем $\mathcal{A} = \mathcal{A}_1 \cdot_r \mathcal{A}_2$, ако:*

- $Q = Q_1 \cup Q_2$
- $S = S_1$
- $F = \begin{cases} F_1 \cup F_2 & \text{ако } S_2 \cap F_2 \neq \emptyset \\ F_2 & \text{ако } S_2 \cap F_2 = \emptyset \end{cases}$
- $\Delta = \Delta_1 \cup \Delta_2 \cup \{\langle q_1, a, q_2 \rangle | (\exists q_{\in S_2})(\langle q, a, q_2 \rangle \in \Delta_2) \ \& \ q_1 \in F_1\}$

Твърдение 3.5. *Нека $\mathcal{A}_{1,2}$ са КА и $\mathcal{A} = \mathcal{A}_1 \cdot_r \mathcal{A}_2$. Тогава всяко успешно изпълнение на $\mathcal{A}$ върху $\overline{a_1 a_2 \ldots a_n} \in \Sigma^*$ е от вида*

$$q_0, \ldots, q_p, q_{p+1}, \ldots, q_n$$

където за някое $p \in [0, n]$, $q_0, \ldots, q_p$ е успешно изпълнение на $\mathcal{A}_1$ върху $\overline{a_1 \ldots a_p}$ и съществува състояние $q \in S_2$, такова, че $q, q_{p+1}, \ldots, q_n$ е успешно изпълнение на $\mathcal{A}_2$ върху $\overline{a_{p+1} \ldots a_n}$.

Твърдение 3.6. *Нека $\mathcal{A}_{1,2}$ са КА. Тогава $\mathcal{L}(\mathcal{A}_1 \cdot_r \mathcal{A}_2) = \mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2)$.*

4. Детерминистични крайни автомати

Дефиниция 4.1. *Наричаме един краен автомат $\mathcal{A} = \langle \Sigma, Q, S, F, \Delta \rangle$ детерминистичен (ДКА), ако за всяко $q_1 \in Q$ и $a \in \Sigma$ съществува най-много едно $q_2 \in Q$, такова, че $\langle q_1, a, q_2 \rangle \in \Delta$, и освен това $|S| = 1$. С други думи, детерминистичен е краен автомат с едно начално състояние и функционална релация на прехода. Именно тези свойства ни дават правото понякога да записваме детерминистичните крайни автомати като $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$, където $q_0 \in Q$ е начално състояние, а $\delta : Q \times \Sigma \to Q$ е функция на прехода.*

Конструкция 4.1 (Детерминизация)**.** *Нека $\mathcal{A} = \langle \Sigma, Q, S, F, \Delta \rangle$ е КА. От $\mathcal{A}$ ще построим $\mathcal{A}^D = \langle \Sigma, Q^D, S^D, F^D, \Delta^D \rangle$ като за целта първо строим паралелно редиците $\{Q_i\}_{i=0}^{\infty}$ и $\{\Delta_i\}_{i=0}^{\infty}$ (където $Q_i \subseteq 2^Q$, а $\Delta_i \subseteq 2^Q \times \Sigma \times 2^Q$) по следната индуктивна схема:*

- $Q_0 = \{S\}$, $\Delta_0 = \emptyset$
- $Q_{i+1} = Q_i \cup \{T | (\exists T'_{\in Q_i})(\exists a_{\in \Sigma})(T' \xrightarrow{\Delta, a} T)\}$
- $\Delta_{i+1} = \Delta_i \cup \{\langle T', a, T \rangle | T' \in Q_i \ \& \ T' \xrightarrow{\Delta, a} T\}$

Веднага се вижда, че и двете редици са монотонно растящи, и следователно имат точни горни граници. Определяме

- $Q^D = \cup Q_i$
- $S^D = \{S\}$
- $F^D = \{T | T \in Q^D \ \& \ T \cap F \neq \emptyset\}$
- $\Delta^D = \cup \Delta_i$

Ще казваме, че $\mathcal{A}^D$ е получен от $\mathcal{A}$ чрез детерминизация.

Лема 4.1. *Нека $\mathcal{A}$ е КА и $\mathcal{A}^D$ е получен от него чрез детерминизация (Конструкция 4.1). Тогава $\mathcal{A}^D$ е детерминистичен.*

Доказателство. Ще проверим, че релацията Δ^D е функционална. Да допуснем, че има $T', T_1, T_2 \subseteq Q^D$ и $a \in \Sigma$, такива, че $\langle T', a, T_1 \rangle \in \Delta^D$, $\langle T', a, T_2 \rangle \in \Delta^D$ и $T_1 \neq T_2$. Тъй като Δ^D е точна горна граница на $\{\Delta_i\}$, съществува $n \geq 0$, такова, че $\langle T', a, T_1 \rangle \in \Delta_n$ и $\langle T', a, T_2 \rangle \in \Delta_n$, и следователно (от дефиницията на редицата) е вярно, че $T' \xrightarrow{\Delta, a} T_1$ и $T' \xrightarrow{\Delta, a} T_2$. Иначе казано $T_1 = \{q | (\exists q'_{\in T'})(\langle q', a, q \rangle \in \Delta)\} = T_2$. С което получихме противоречие и доказахме, че Δ^D е функционална релация.

Без съмнение $|S^D| = |\{S\}| = 1$, с което успешно показахме, че $\mathcal{A}^D$ е детерминистичен.

Ще означаваме началното състояние на $\mathcal{A}^D$ с $q_0^D = Q^D$, а функцията на прехода с $\delta^D : Q^D \times \Sigma \longrightarrow Q^D$ (определена от графиката си Δ^D). □

Лема 4.2. *Нека $\mathcal{A}$ е КА, $\alpha \in \Sigma^*$ и $q_0, q_1, \ldots, q_{|\alpha|}$ е изпълнение на $\mathcal{A}$ върху α. Нека още $\mathcal{A}^D$ е получен от $\mathcal{A}$ чрез детерминизация. Тогава съществува изпълнение на $\mathcal{A}^D$: $T_0, T_1, \ldots, T_{|\alpha|}$, такова, че $q_i \in T_i$ за всяко $i = 0, 1, \ldots, |\alpha|$.*

Доказателство. Ще докажем исканото с индукция по дължината на α.

– $|\alpha| = 0$. След като $\alpha = \epsilon$, изпълнението на $\mathcal{A}$ съдържа единствено състояние $q_0 \in S$. Търсеното изпълнение на $\mathcal{A}^D$ също има единствено състояние $T_0 = q_0^D = S$. Очевидно $q_0 \in T_0$.

– $|\alpha| > 0$. Нека $\alpha = \alpha' a$. По индукционно предположение твърдението на лемата е вярно за α', и следователно съществува изпълнение $T_0, T_1, \ldots, T_{|\alpha'|}$ на $\mathcal{A}^D$ върху α', за което $q_i \in T_i$ при всяко $i = 0, \ldots, |\alpha'|$. Тъй като $|\alpha'| = |\alpha| - 1$ и $q_0, q_1, \ldots, q_{|\alpha|-1}, q_{|\alpha|}$ е изпълнение на $\mathcal{A}$ върху $\alpha' a$, значи $\langle q_{|\alpha'|}, a, q_\alpha \rangle \in \Delta$. Следователно, ако $T_{|\alpha'|} \xrightarrow{\Delta, a} T$, то $q_{|\alpha|} \in T$. Нека сега разгледаме в конструкцията на $\mathcal{A}^D$ най-малкото $i \geq 0$, за което $T_{|\alpha'|} \in Q_i$ (такова задължително има, защото $T_{|\alpha'|} \in \cup Q_i$). Тогава на следващата стъпка $Q_{i+1} = Q_i \cup \{T | (\exists T'_{\in Q_i})(\exists a_{\in \Sigma})(T' \xrightarrow{\Delta, a} T)\}$, и понеже $T_{|\alpha'|} \in Q_i$ и $T_{|\alpha'|} \xrightarrow{\Delta, a} T$, следва, че $T \in Q_{i+1}$. Аналогично се вижда, че $\langle T_{|\alpha'|}, a, T \rangle \in \Delta_{i+1}$. Следователно $\delta^D(T_{|\alpha'|}, a) = T$ и $T_0, T_1, \ldots, T_{|\alpha'|}, T$ е изпълнение на $\mathcal{A}^D$ върху α удовлетворяващо искането свойство.

□

Лема 4.3. *Нека $\mathcal{A}$ е КА и $\mathcal{A}^D$ е получен от него чрез детерминизация. Нека още $\alpha \in \Sigma^*$, $T_0, T_1, \ldots, T_{|\alpha|}$ е изпълнение на $\mathcal{A}^D$ и $q \in T_{|\alpha|}$. Тогава съществува изпълнение на $\mathcal{A}$ — $q_0, q_1, \ldots, q_{|\alpha|} = q$, такова, че $q_i \in T_i$ за всяко $i = 0, 1, \ldots, |\alpha|$.*

Доказателство. Отново провеждаме доказателството по индукция относно дължината на α.

– $|\alpha| = 0$. Тъй като $\alpha = \epsilon$, изпълнението на $\mathcal{A}^D$ е редица от единствено състояние $T_0 = q_0^D = S$. Тъй като $q \in T_0 = S$, редицата от единственото състояние q е изпълнение на $\mathcal{A}$, отговарящо на изискванията.

– $|\alpha| > 0$. Нека $\alpha = \alpha' a$. След като $T_0, T_1, \ldots, T_{|\alpha'|}, T_{|\alpha|}$ е изпълнение на $\mathcal{A}^D$, значи $\langle T_{|\alpha'|}, a, T_{|\alpha|} \rangle \in \Delta^D$, и следователно $\langle T_{|\alpha'|}, a, T_{|\alpha|} \rangle \in \Delta_n$ за някое $n \geq 0$ (Δ^D е точна горна граница на редицата $\{\Delta_i\}_{i=0}^{\infty}$). Нека $m < n$ е най-големият индекс, такъв, че $\langle T_{|\alpha'|}, a, T_{|\alpha|} \rangle \notin \Delta_m$. Тогава на стъпка $m + 1$ от конструкцията тази тройка е била добавена към Δ_{m+1}, защото $T_{|\alpha'|} \xrightarrow{\Delta, a} T_{|\alpha|}$. Сега понеже $q \in T_{|\alpha|}$, значи има $q' \in T_{|\alpha'|}$, такова, че $\langle q', a, q \rangle \in \Delta$. По индукционно предположение твърдението е вярно за $|\alpha'|$, и следователно съществува изпълнение $q_0, q_1, \ldots, q'$ на $\mathcal{A}$ върху α', изпълняващо изискванията. Добавяме q към него и получаваме изпълнението на $\mathcal{A}$ върху α, което търсехме.

□

Твърдение 4.1. *Нека $\mathcal{A}$ е КА и $\mathcal{A}^D$ е получен от него чрез детерминизация. Тогава $\mathcal{A}$ и $\mathcal{A}^D$ са еквивалентни.*

Доказателство. Нека $\alpha \in \mathcal{L}(\mathcal{A})$. Тогава съществува успешно изпълнение $q_0, q_1, \ldots, q_{|\alpha|} \in F$ на $\mathcal{A}$ върху α. Според Лема 4.2 съществува изпълнение $T_0, T_1, \ldots, T_{|\alpha|}$ на $\mathcal{A}^D$, такова, че $q_i \in T_i$ за $i \in [0, |\alpha|]$. Но след като $q_{|\alpha|} \in$

$T_{|\alpha|}$, значи $T_{|\alpha|} \cap F \neq \emptyset$, и следователно $T_{|\alpha|} \in F^D$, т.е. $\mathcal{A}^D$ има успешно изпълнение върху α или иначе казано $\alpha \in \mathcal{L}(\mathcal{A}^D)$.

Обратно, нека $\alpha \in \mathcal{L}(\mathcal{A}^D)$. Това означава, че има успешно изпълнение $T_0, T_1, \ldots, T_{|\alpha|}$ на $\mathcal{A}^D$ върху α. След като $T_{|\alpha|} \in F^D$, непременно е вярно, че $T_{|\alpha|} \cap F \neq \emptyset$, т.е. съществува $q \in T_{|\alpha|}$, което е финално състояние на $\mathcal{A}$. Използвайки Лема 4.3 заключаваме, че има изпълнение $q_0, q_1, \ldots, q$ на $\mathcal{A}$ върху α, което се оказва успешно ($q \in F$), откъдето веднага следва, че $\alpha \in \mathcal{L}(\mathcal{A})$. □

Следствие 4.1. *За всеки краен автомат $\mathcal{A}$ съществува детерминистичен краен автомат $\mathcal{A}^D$, такъв, че $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}^D)$.*

5. Бимашини

Дефиниция 5.1. *Бимашина наричаме тройка от вида*

$$\mathcal{B} = \langle \mathcal{A}_L, \mathcal{A}_R, \psi \rangle$$

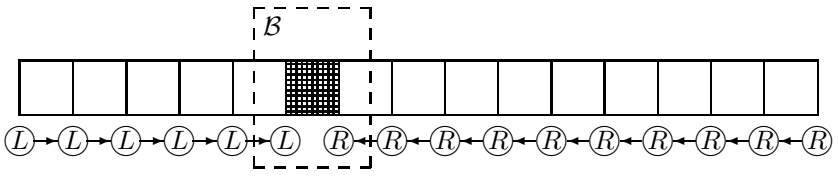
където $\mathcal{A}_{L,R} = \langle \Sigma, Q_{L,R}, q_{L,R}, \delta_{L,R} \rangle$ *са детерминистични крайни автомати без финални състояния (наричаме ги съответно ляв и десен), а* $\psi : Q_L \times \Sigma \times Q_R \to \Sigma^*$ *е изходна функция.*

Бимашините са абстрактни машини, които работят върху входна дума (в двете посоки) и извеждат изходна дума въз основа на състоянията, през които преминават автоматите им, и вече прочетения вход. Тяхната операционна семантика се задава от транзитивното затваряне на ψ, $\psi^* : Q_L \times \Sigma^* \times Q_R \to \Sigma^*$, което дефинираме както следва:

- $\psi^*(q_1, \epsilon, q_2) = \epsilon$
- $\psi^*(q_1, a\alpha, q_2) = \psi(q_1, a, \delta_R^*(q_2, \widetilde{\alpha})) \cdot \psi^*(\delta_L(q_1, a), \alpha, q_2)$

Заради функционалната природа на бимашините, често ще използваме означението $\mathcal{B} : \Sigma^* \to \Sigma^*$, което за всяко $\alpha \in \Sigma^*$ дефинираме като $\mathcal{B}(\alpha) = \psi^*(q_L, \alpha, q_R)$. Ще казваме, че за произволна дума $\alpha \in \Sigma^*$ над азбуката на бимашината, $\mathcal{B}(\alpha)$ е резултатът от изпълнението на $\mathcal{B}$ върху α.

Зад дългата дефиниция на резултат от изпълнение на бимашина се крие интуитивно проста стратегия. Можем да си мислим, че бимашината чете входната дума и за всеки символ от нея извежда дума над азбуката си. Резултатът от изпълнението е конкатенацията на всички така изведени думи. На всяка стъпка изходната функция взима решение какво да изведе в зависимост от текущия символ и двете състояния, до които биха стигнали левият и десният автомат, четейки входа съответно отляво надясно и отдясно наляво непосредствено преди да прочетат текущия символ.



Фиг. 2. Изпълнение на бимашина

Става ясно, че така въведените бимашини работят единствено върху входната си дума. Сега ще въведем модифицирана версия на тази стратегия, която ще се отличава единствено по операционната семантика на бимашината.

Дефиниция 5.2. *(Лява) Бимашина, работеща върху изхода си наричаме*

$$\mathcal{B} = \langle \mathcal{A}_L, \mathcal{A}_R, \psi \rangle$$

където $\mathcal{A}_{L,R}$ отново са ляв и десен ДКА, ψ е изходна функция, а операционната семантика $\mathcal{B}$ на бимашината е дефинирана като $\mathcal{B}(\alpha) = \psi^{**}(q_L, \alpha, q_R)$, където

- $\psi^{**}(q_1, \epsilon, q_2) = \epsilon$
- $\psi^{**}(q_1, a\alpha, q_2) = \psi(q_1, a, \delta_R^*(q_2, \widetilde{\alpha})) \cdot \psi^{**}(\delta_L^*(q_1, \psi(q_1, a, \delta_R^*(q_2, \widetilde{\alpha}))), \alpha, q_2)$

Специфично за бимашините, работещи върху изхода си, е, че левият им автомат не чете входната дума, а изхода, определен от изходната функция. Съвсем симетрично можем да дефинираме и бимашина, работеща върху изхода си отдясно наляво.

Сега ще покажем, че всяка бимашина, работеща върху изхода си, може да се симулира чрез еквивалентна класическа бимашина (работеща върху входа си).

Конструкция 5.1. *Нека $\mathcal{B}$ е бимашина, работеща върху изхода си. Дефинираме КА*

$$\mathcal{A}_L^N = \langle \Sigma, Q_L \times Q_R, \{q_L\} \times Q_R, \Delta_L \rangle$$

където релацията на прехода е определена както следва

$$\langle \langle p_1, q_1 \rangle, a, \langle p_2, q_2 \rangle \rangle \in \Delta_L \iff \delta_R(q_2, a) = q_1 \ \& \ \delta_L^*(p_1, \psi(p_1, a, q_2)) = p_2$$

Сега строим левия ДКА $\mathcal{A}'_L$ чрез детерминизация на $\mathcal{A}_L^N$ (Конструкция 4.1). За десен автомат на търсената класическа бимашина взимаме $\mathcal{A}'_R = \mathcal{A}_R$ и определяме изходната функция $\psi' : Q'_L \times \Sigma \times Q'_R$ както следва

$$\psi'(L, a, r) = \beta \iff (\exists \langle p, q \rangle \in L)(\delta_R(r, a) = q \ \& \ \psi(p, a, r) = \beta)$$

Така завършихме дефиницията на $\mathcal{B}' = \langle \mathcal{A}'_L, \mathcal{A}'_R, \psi' \rangle$.

За да сме сигурни, че $\mathcal{B}'$ е класическа бимашина, еквивалентна на $\mathcal{B}$, трябва да проверим, че тя е коректно дефинирана и че за всяко $\alpha \in \Sigma^*$, $\mathcal{B}'(\alpha) = \mathcal{B}(\alpha)$. За коректността единственото, което трябва да покажем, е, че ψ' действително е функция.

Твърдение 5.1. *Нека $L \in Q'_L$ е състояние на $\mathcal{A}'_L$ и $\langle p_k, q_k \rangle \in L (k = 1, 2)$. Тогава $q_1 = q_2 \to p_1 = p_2$*

Доказателство. Нека $Q'_L = \cup Q_i$ е точната горна граница на редицата $\{Q_i\}_{i=0}^{\infty}$ от Конструкция 4.1. Ще проведем доказателството с пълна индукция по най-малкия индекс i, за който $L \in Q_i$ (такъв задължително има, защото Q'_L е точна горна граница на редицата).

$i = 0$ Очевидно $p_1 = p_2 = q_L$, защото $L = \{q_L\} \times Q_R$.

$i + 1$ Нека $i + 1$ е най-малкият индекс, за който $L \in Q_{i+1}$. Тогава (от Конструкция 4.1) съществува $L' \in Q_i$, такова, че $\delta'_L(L', a) = L$ за някое $a \in \Sigma$. Нека $\langle p_k, q_k \rangle \in L$. Следователно съществуват $\langle p'_k, q'_k \rangle \in L'$, такива, че $\langle \langle p'_k, q'_k \rangle, a, \langle p_k, q_k \rangle \rangle \in \Delta_L$ (за $k = 1, 2$). Сега ако приемем, че $q_1 = q_2$, то $q'_1 = \delta_R(q_1, a) = \delta_R(q_2, a) = q'_2$. Нека $i' < i + 1$ е най-малкият индекс, за който $L' \in Q_{i'}$. По индукционното предположение за i' следва, че $p'_1 = p'_2$. Най-накрая $p_1 = \delta_L^*(p'_1, \psi(p'_1, a, q_1)) = \delta_L^*(p'_2, \psi(p'_2, a, q_2)) = p_2$, което и искахме да покажем.

□

Следствие 5.1. *ψ' е коректно дефинирана функция*

Доказателство. Да допуснем, че $\psi'(L, a, r) = \beta_{1,2}$. От дефиницията на ψ' би следвало, че съществуват $\langle p_i, q_i\rangle \in L$ за $i = 1, 2$, такива, че $\delta_R(r, a) = q_i$, и освен това $\psi(p_i, a, r) = \beta_i$. Но тогава $q_1 = \delta_R(r, a) = q_2$ и от Твърдение 5.1 следва, че $p_1 = p_2$. Това ни води до извода, че $\beta_1 = \psi(p_1, a, r) = \psi(p_2, a, r) = \beta_2$. С което показахме, че ψ' е коректно дефинирана функция. □

Остава да докажем, че така построената класическа бимашина е еквивалентна на бимашината, работеща върху изхода си, от която тръгнахме.

Твърдение 5.2. *Нека $t = \alpha\beta \in \Sigma^*$, $\delta'^*_L(q'_L, \alpha) = L$. Тогава*

$$\langle \delta^*_L(q_L, \psi^{**}(q_L, \alpha, \delta^*_R(q_R, \widetilde{\beta}))), \delta^*_R(q_R, \widetilde{\beta})\rangle \in L$$

Доказателство. Ще проведем доказателството с индукция по α.

$\alpha = \epsilon$ Когато $\alpha = \epsilon$, по дефиницията на ψ^{**} имаме, че $\psi^{**}(q_L, \alpha, \delta^*_R(q_R, \widetilde{\beta})) = \epsilon$, и следователно $\delta^*_L(q_L, \psi^{**}(q_L, \alpha, \delta^*_R(q_R, \widetilde{\beta}))) = q_L$. Тъй като $L = q'_L = \{q_L\} \times Q_R$, очевидно $\langle q_L, \delta^*_R(q_R, \widetilde{\beta})\rangle \in L$, с което показахме исканото.

$\alpha = \alpha_1 a$ Нека твърдението е вярно за α_1 и да го проверим за $\alpha = \alpha_1 a$. Нека още $\delta^*_L(q_L, \psi^{**}(q_L, \alpha_1, \delta^*_R(q_R, \widetilde{\beta}))) = p_1$, $\delta^*_R(q_R, \widetilde{\beta}) = r$, а $\delta^*_R(r, a) = r_1$. По индукционно предположение можем да заключим, че $\langle p_1, r_1\rangle \in L'$. От друга страна $L = \delta'_L(L', a)$, и понеже $\delta_R(r, a) = r_1$, можем да твърдим, че $\langle\langle p_1, r_1\rangle, a, \langle \delta^*_L(p_1, \psi(p_1, a, r)), r\rangle\rangle \in \Delta_L$. Следователно (от Конструкция 4.1) имаме, че $\langle \delta^*_L(p_1, \psi(p_1, a, r)), r\rangle \in L$. Но от дефиницията на ψ^{**}, $\delta^*_L(p_1, \psi(p_1, a, r)) = \delta^*_L(q_L, \psi^{**}(q_L, \alpha, \delta^*_R(q_R, \widetilde{\beta})))$, с което твърдението е доказано.

□

Следствие 5.2. *За произволно $\alpha \in \Sigma^*$, $\mathcal{B}(\alpha) = \mathcal{B}'(\alpha)$*

Доказателство. Да започнем със случая, когато $\alpha = \epsilon$. От дефинициите на ψ и ψ' следва, че $\mathcal{B}(\alpha) = \psi^{**}(q_L, \epsilon, q_R) = \epsilon = \psi'^*(q'_L, \alpha, q'_R) = \mathcal{B}'(\alpha)$.

Когато $\alpha = \overline{a_1 a_2 \dots a_n}$, ще проверим, че за $i = 1, 2, \dots, n$ и $r = \delta^*_R(q_R, \overline{a_n a_{n-1} \dots a_{i+1}})$

$$\psi(\delta^*_L(q_L, \psi^{**}(q_L, \overline{a_1 a_2 \dots a_{i-1}}, \delta_R(r, a_i))), a_i, r) = \psi'(\delta'^*_L(q'_L, \overline{a_1 a_2 \dots a_{i-1}}), a_i, r)$$

Но това е така, защото според доказаното вече Твърдение 5.2 и дефиницията на ψ':

$$\langle \delta^*_L(q_L, \psi^{**}(q_L, \overline{a_1 a_2 \dots a_{i-1}}, \delta_R(r, a_i))), \delta_R(r, a_i)\rangle \in \delta'^*_L(q'_L, \overline{a_1 a_2 \dots a_{i-1}})$$

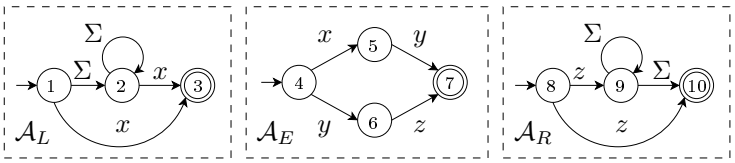
Така проверихме, че за всяка позиция от входната дума $\mathcal{B}$ и $\mathcal{B}'$ извеждат еднакви резултати. Оттук следва, че $\mathcal{B}(\alpha) = \mathcal{B}'(\alpha)$. □

6. Директна конструкция

Нека $E \to \beta \ / \ L_R$ е контекстно правило за заместване над крайна азбука Σ. Строим крайните автомати $\mathcal{A}_L$, $\mathcal{A}_E$ и $\mathcal{A}_R$, такива, че

$$\begin{array}{rcccll} \mathcal{A}_L & = & \mathcal{A}(\Sigma^* L) & = & \langle \Sigma, Q_L, S_L, F_L, \Delta_L \rangle & , \quad \mathcal{L}(\mathcal{A}_L) = \mathcal{L}(\Sigma^* L) \\ \mathcal{A}_E & = & \mathcal{A}(E) & = & \langle \Sigma, Q_E, S_E, F_E, \Delta_E \rangle & , \quad \mathcal{L}(\mathcal{A}_E) = \mathcal{L}(E) \\ \mathcal{A}_R & = & \mathcal{A}(R\Sigma^*) & = & \langle \Sigma, Q_R, S_R, F_R, \Delta_R \rangle & , \quad \mathcal{L}(\mathcal{A}_R) = \mathcal{L}(R\Sigma^*) \end{array}$$

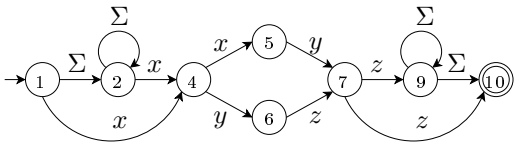
Макар да не сме указали конкретната конструкция на краен автомат по регулярен израз, ще имаме неявното изискване $\mathcal{A}_L$, $\mathcal{A}_E$ и $\mathcal{A}_R$ да са в нормална форма.



Фиг. 3. Крайни автомати в нормална форма, построени съответно за регулярните изрази $\Sigma^* x$, $xy|yz$ и $z\Sigma^*$

След като сме построили $\mathcal{A}_L$, $\mathcal{A}_E$ и $\mathcal{A}_R$, ги конкатенираме, за да получим

$$\mathcal{A} = \langle \Sigma, Q, S, F, \Delta \rangle = \mathcal{A}_L \cdot_l \mathcal{A}_L \cdot_r \mathcal{A}_R$$



Фиг. 4. Краен автомат, получен при конкатенацията $\mathcal{A} = \mathcal{A}_L \cdot_l \mathcal{A}_E \cdot_r \mathcal{A}_R$, построен за контекстното правило $xy|yz \to \epsilon \ / \ x_z$ от Пример 2.2

От свойствата на нормална форма на КА и конкатенация отляво/отдясно можем да запишем следните

Свойство 6.1. *Всяко успешно изпълнение на $\mathcal{A}$ върху $t \in \Sigma^*$ е от вида*

$$q_{l_0}, q_{l_1}, \ldots, q_{l_{|u|-1}}, q_{e_0}, q_{e_1}, \ldots, q_{e_{|v|}}, q_{r_1}, q_{r_2}, \ldots, q_{r_{|w|}}$$

където $\langle u, v, w \rangle \in \mathcal{C}(t; L, E, R)$, $q_{e_0} \in S_E$, $q_{e_{|v|}} \in F_E$, $q_{e_i} \in Q_E - S_E - F_E$ за всяко $i \in (0, |v|)$, $q_{l_i} \in Q_L$ за всяко $i \in [0, |u|)$, $q_{r_i} \in Q_R - S_R - F_R$ за всяко $i \in (|uv|, |uvw|)$, а $q_{r_{|uvw|}} \in F_R$. Такова изпълнение на $\mathcal{A}$ ще наричаме изпълнение за контекста $\langle u, v, w \rangle$. Ще бележим с $\mathcal{X}(t; L, E, R) = \{\sigma | \sigma$ е изпълнение за някой $\langle u, v, w \rangle \in \mathcal{C}(t; L, E, R)\}$

Свойство 6.2. *За всеки контекст* $\langle u, v, w\rangle \in \mathcal{C}(t; L, E, R)$ *има изпълнение* $\sigma \in \mathcal{X}(t; L, E, R)$ *за* $\langle u, v, w\rangle$.

Ще означаваме с $\mathcal{X}_v(t; L, E, R) = \{\sigma | \sigma$ е изпълнение за някой $\langle u, v, w\rangle \in \mathcal{C}_v(t; L, E, R)\}$

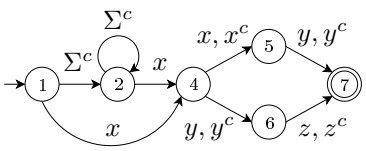
За да построим търсената класическа бимашина, най-напред ще построим бимашина $\mathcal{B} = \langle \mathcal{A}_L, \mathcal{A}_R, \psi\rangle$, работеща върху изхода си. Тя ще не послужи за разрешаване на контекстните многозначности.

6.1. Левият автомат

Нека разширим входната азбука Σ до $\Sigma^c = \Sigma \cup \{a^c | a \in \Sigma\}$. Неформално погледнато, Σ^c разширява Σ, копирайки всеки от символите в Σ. За да построим левия автомат, най-напред дефинираме

$$\mathcal{A}_1^N = \langle \Sigma^c, Q, S, F, \Delta_1^N\rangle$$

където $\Delta_1^N = \Delta \cup \{\langle q_1, a^c, q_2\rangle | \langle q_1, a, q_2\rangle \in \Delta \ \&\ q_2 \notin S_E\} - Q \times \Sigma^c \times Q_R$.



Фиг. 5. Недетерминираната версия $\mathcal{A}_1^N$ на левия автомат $\mathcal{A}_1$, построен за контекстното правило $xy|yz \to \epsilon \ /\ x_z$

Интуитивно, $\mathcal{A}_1^N$ се държи като $\mathcal{A}_L \cdot_l \mathcal{A}_E$ с единствената разлика, че при срещане на клонираните символи не позволява преходи към начални състояния на $\mathcal{A}_E$, като по този начин предотвратява изпълнения за контексти, чийто фокус тепърва предстои да бъде обработен. От друга страна изходната функция на бимашината ще има грижата да извежда клонираните символи само докато обработва фокуси на валидни контексти.

Детерминираме $\mathcal{A}_1^N$ (Конструкция 4.1) и получаваме левия автомат $\mathcal{A}_1$ на търсената бимашина.

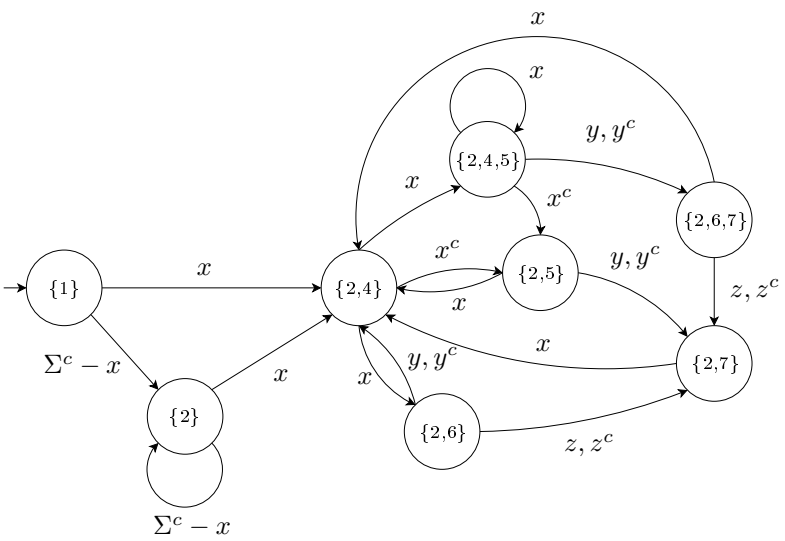
6.2. Десният автомат

Най-напред "обръщаме"$\mathcal{A}$, за да получим огледалния му $\mathcal{A}_2^N = \widetilde{\mathcal{A}}$. След това получаваме десния автомат на бимашината $\mathcal{A}_2$ като детерминираме $\mathcal{A}_2^N$ (Конструкция 4.1).

6.3. Изходната функция

Изходната функция ψ на бимашината $\mathcal{B}$ дефинираме както следва

$$\psi(L, a, R) = \begin{cases} a^c & \text{ако } \delta_1(L, a) \cap R \cap (Q_E - S_E - F_E) \neq \emptyset \\ a & \text{ако } \delta_1(L, a) \cap R \cap (Q_E - S_E - F_E) = \emptyset \end{cases}$$



Фиг. 6. Детерминистичният $\mathcal{A}_1$ получен за $\mathcal{A}_1^N$ от Фиг. 5. С цел четимост на примера са пропуснати преходите (където не са указани) от всяко състояние със всеки символ от Σ^c към $\{2\}$

С това завършихме дефиницията на $\mathcal{B} = \langle \mathcal{A}_L, \mathcal{A}_R, \psi \rangle$ - бимашина, работеща върху изхода си. Нека преди да завършим конструкцията на търсената класическа бимашина да покажем някои интересни свойства на $\mathcal{B}$.

Лема 6.1. *Нека $t = \alpha\beta$, $\delta_2^*(q_2, \widetilde{\beta}) = R$ и $\delta_1^*(q_1, \psi^{**}(q_1, \alpha, R)) = L$. Нека още σ е изпълнение на $\mathcal{A}$ върху α, такова, че $\sigma(|\alpha|) \in Q_L$. Тогава $\sigma(|\alpha|) \in L$.*

Доказателство. Ще покажем исканото с индукция по дължината на α.

$\alpha = \epsilon$ Очевидно $L = \delta_1^*(q_1, \psi^{**}(q_1, \epsilon, R)) = \delta_1^*(q_1, \epsilon) = q_1 = S$. Откъдето $\sigma(|\alpha|) = \sigma(0) \in S = L$.

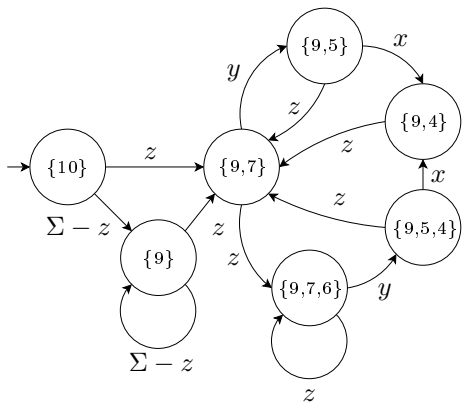
$\alpha = \alpha' a$ След като $\sigma(|\alpha|) \in Q_L$, заради Свойство 6.1 можем да заключим, че и $\sigma(|\alpha'|) \in Q_L$. Нека означим $R' = \delta_2(R, a)$ и $L' = \delta_1^*(q_1, \psi^{**}(q_1, \alpha', R))$. Тогава по индукционното предположение ще следва, че $\sigma(|\alpha'|) \in L'$. Тъй като σ е преди всичко изпълнение на $\mathcal{A}$, значи $\langle \sigma(|\alpha'|), a, \sigma(|\alpha|) \rangle \in \Delta$. Остава да забележим, че $L = \delta_1^*(L', \psi(L', a, R))$, и освен това $\sigma(|\alpha|) \notin S_E \cup Q_R$ и от дефиницията на δ_1 можем да заключим, че $\sigma(|\alpha|) \in L$.

□

Лема 6.2. *Нека $t = \alpha\beta$ и $\delta_2^*(q_2, \widetilde{\beta}) = R$, а $\delta_1^*(q_1, \psi^{**}(q_1, \alpha, R)) = L$. Тогава, ако $q \in L$, то съществува σ, изпълнение на $\mathcal{A}$ върху α, такова, че $\sigma(|\alpha|) = q$.*

Доказателство. Ще проведем доказателството с индукция по α:

$\alpha = \epsilon$ Тогава $L = q_1 = S$. Сега ако $q \in L$, разглеждаме изпълнение σ от единствено състояние q. Очевидно σ е изпълнение на $\mathcal{A}$ и $\sigma(|\alpha|) = q$.



Фиг. 7. Десният автомат $\mathcal{A}_R$ за правилото $xy|yz \to \epsilon \;/\; x_y$. За яснота са пропуснати преходите към $\{9\}$

		L_0	L_1	L_2	L_3	L_4	L_5	L_6	L_7
		$\{1\}$	$\{2\}$	$\{2,4\}$	$\{2,4,5\}$	$\{2,5\}$	$\{2,6\}$	$\{2,6,7\}$	$\{2,7\}$
R_0	$\{10\}$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R_1	$\{9\}$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R_2	$\{9,7\}$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R_3	$\{9,5\}$	$\cdot/\cdot$	$\cdot/\cdot$	x/x^c	x/x^c	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R_4	$\{9,7,6\}$	$\cdot/\cdot$	$\cdot/\cdot$	y/y^c	y/y^c	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R_5	$\{9,4\}$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R_6	$\{9,5,4\}$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$

Фиг. 8. Изходната функция на $\mathcal{B}$ за правилото $xy|yz \to \epsilon \;/\; x_z$. Отбелязани са само нетривиалните заместания

$\alpha = \alpha' a$ Нека $\delta_2(R, a) = R'$ и $\delta_1^*(q_1, \psi^{**}(q_1, \alpha', R')) = L'$. Сега $L = \delta_1^*(L', \psi(L', a, R))$. Да обърнем внимание, че $\psi(L', a, R) \in \{a, a^c\}$, и следователно $L = \delta_1(L', \psi(L', a, R))$.

Нека сега $q \in L$. Тогава има $q' \in L'$, такова, че $\langle q', \psi(L', a, R), q\rangle \in \Delta_1^N$, което, заради построението на Δ_1^N, означава, че $\langle q', a, q\rangle \in \Delta$. Но по индукционното предположение следва, че съществува σ', изпълнение на $\mathcal{A}$ върху α', такова, че $\sigma'(|\alpha'|) = q'$. Дефинираме

$$\sigma(i) = \begin{cases} \sigma'(i) & i \le |\alpha'| \\ q & i = |\alpha| \end{cases}$$

Лесно се вижда, че σ' е точно търсеното изпълнение на $\mathcal{A}$ върху α.

□

Твърдение 6.1. *Нека* $t = \alpha\beta \in \Sigma^*$, $\delta_2^*(q_2, \widetilde{\beta}) = R$, *а* $\delta_1^*(q_1, \psi^{**}(q_1, \alpha, R)) = L$. *Тогава за всяко* $q_{\in Q_E}$, $q \in L \cap R$ *точно тогава, когато* $(\exists \sigma_{\in \mathcal{X}_v})(\sigma(|\alpha|) = q)$.

Доказателство. Преди да започнем доказателството ще проверим, че твърдението може да се сведе до тези $q \in Q_E$, за които съществува изпълнение $\sigma \in \mathcal{X}$, такова, че $\sigma(|\alpha|) = q$. И наистина, ако $q \in L \cap R$, то $q \in L$, и тогава по Лема 6.2 ще съществува изпълнение σ_L на $\mathcal{A}$ върху α, такова, че $\sigma_L(|\alpha|) = q$. От друга страна $q \in R$, и следователно по Лема 4.3 има изпълнение σ_R на $\widetilde{\mathcal{A}}$ върху $\widetilde{\beta}$, такова, че $\sigma_R(|\beta|) = q$. Тогава σ, дефинирано както следва:

$$\sigma(i) = \begin{cases} \sigma_L(i) & i \leq |\alpha| \\ \sigma_R(|t| - i) & i > |\alpha| \end{cases}$$

е успешно изпълнение на $\mathcal{A}$, и следователно $\sigma \in \mathcal{X}$.

Проверката в обратната посока е очевидна, защото $\mathcal{X}_v \subseteq \mathcal{X}$.

Нека сега дефинираме множеството

$$A = \{\langle \alpha, q \rangle | \alpha \subseteq t \ \& \ q \in Q_E \ \& \ (\exists \sigma_{\in \mathcal{X}})(\sigma(|\alpha|) = q)\}$$

и релация "$\prec$"в A, за която

$$\langle \alpha_1, q_1 \rangle \prec \langle \alpha_2, q_2 \rangle \iff \alpha_1 \subset \alpha_2 \vee \alpha_1 = \alpha_2 \ \& \ q_1 \in (Q_E - S_E) \ \& \ q_2 \in S_E$$

Това е частична наредба в A, с помощта на която A става фундирано множество.

Ще докажем твърдението с *индукция по структурата на* A. Да фиксираме един елемент $\langle \alpha, q \rangle \in A$ и да допуснем, че за всеки предхождащ го елемент $\langle \alpha', q' \rangle \in A$ ($\langle \alpha', q' \rangle \prec \langle \alpha, q \rangle$) твърдението е вярно. Ще покажем, че продължава да е вярно и за $\langle \alpha, q \rangle$.

$\alpha = \epsilon$ $\Rightarrow$ Нека $q \in L \cap R$. Но $\alpha = \epsilon$, значи $q = \sigma(|\alpha|) = \sigma(0) \in S \cap Q_E$, и следователно $q \in S_E$. Това означава, че има контекст $\langle \epsilon, v, w \rangle \in \mathcal{C}$ и σ е изпълнение за него. Но $\langle \epsilon, v, w \rangle \in \mathcal{C}_1$ (от построението на $\mathcal{C}_v$), и следователно $\langle \epsilon, v, w \rangle \in \mathcal{C}_v$, а $\sigma \in \mathcal{X}_v$.

$\Leftarrow$ Нека $\sigma \in \mathcal{X}_v$. Тогава $q = \sigma(|\alpha|) = \sigma(0) \in S = q_1 = L$. От друга страна $\widetilde{\sigma}$ е изпълнение на $\widetilde{\mathcal{A}}$ и по Лема 4.2 $q \in \delta_2^*(q_2, \widetilde{\beta}) = R$, което ни води до $q \in L \cap R$.

$\alpha = \alpha' a$ Нека означим $R' = \delta_2(R, a)$, $L' = \delta_1^*(q_1, \psi^{**}(q_1, \alpha', R'))$.

$\Rightarrow$ Нека $q \in L \cap R$. Има два варианта за q:

$\mathbf{q \in Q_E - S_E}$ Тогава (от Конструкция 4.1) съществува $q' \in L'$, такова, че $\langle q', a, q \rangle \in \Delta$, и следователно $q' \in \delta_2(R, a) = R'$. От Свойство 6.1 можем да твърдим, че $q' \in Q_E$, и следователно $q' \in L' \cap R' \cap Q_E$. Индукционното предположение за $\langle \alpha', q' \rangle$ ни дава, че има $\sigma' \in \mathcal{X}_v$ и $\sigma'(|\alpha'|) = q'$. Тъй като $q \in R$, съществува изпълнение σ_R на $\widetilde{\mathcal{A}}$ върху $\widetilde{\beta}$, за което $\sigma_R(0) \in F$ и $\sigma(|\widetilde{\beta}|) = q$ (Лема 4.3). Сега можем да построим σ – успешно изпълнение на $\mathcal{A}$, дефинирано както следва

$$\sigma(i) = \begin{cases} \sigma'(i) & \text{ако } i < |\alpha| \\ \sigma_R(|t| - i) & \text{ако } i \geq |\alpha| \end{cases}$$

Веднага се вижда, че ако σ' е било изпълнение за $\langle u_0, v_0, w_0\rangle \in \mathcal{C}_v$, то σ е изпълнение за $\langle u_0, v, w\rangle \in \mathcal{C}$. Сега от Твърдение 2.4 следва, че $\langle u_0, v, w\rangle \in \mathcal{C}_v$ също е валиден контекст, т.е. $\sigma \in \mathcal{X}_v$.

$\mathbf{q \in S_E}$ Първото, което веднага забелязваме, е, че $\psi(L', a, R) \notin (\Sigma^c - \Sigma)$ (това е така, защото в противен случай щяхме да имаме $L \cap S_E = \emptyset$). Директно следствие от дефиницията на ψ е, че $L \cap R \cap (Q_E - S_E - F_E) = \emptyset$. Нека сега $\sigma \in \mathcal{X}$ е такова, че $\sigma(|\alpha|) = q$, и това е изпълнение за контекста $\langle u, v, w\rangle \in \mathcal{C}$. Ще покажем, че този контекст е валиден ($\langle u, v, w\rangle \in \mathcal{C}_v$). И наистина, да допуснем, че $\langle u, v, w\rangle \notin \mathcal{C}_v$. Тогава непременно има $\langle u', v', w'\rangle \in \mathcal{C}_v$, такъв, че $u' \subset u \subset u'v'$ (Твърдение 2.3), и значи има изпълнение $\sigma' \in \mathcal{X}_v$ за $\langle u', v', w'\rangle$, такова, че $\sigma'(|\alpha|) \in Q_E - S_E - F_E$ (Свойство 6.1). Тогава $\langle \alpha, \sigma'(|\alpha|)\rangle \prec \langle \alpha, q\rangle$ и по индукционното предположение $\sigma'(|\alpha|) \in L \cap R$, откъдето стигаме до $L \cap R \cap (Q_E - S_E - F_E) \neq \emptyset$, което е противоречие с допускането ни, че $\langle u, v, w\rangle \notin \mathcal{C}_v$. Следователно $\sigma \in \mathcal{X}_v$.

$\Leftarrow$ Нека $\sigma \in \mathcal{X}_v$ и $\sigma(|\alpha|) = q$. Тъй като σ е успешно изпълнение на $\mathcal{A}$ върху $\alpha\beta$, значи има изпълнение σ_R на $\widetilde{\mathcal{A}}$ върху $\widetilde{\beta}$ и $\sigma_R(|\beta|) = q$. Това означава, че $q \in R$ (Лема 4.2). Остава да покажем, че $q \in L$. Отново разглеждаме два случая:

$\mathbf{q \in Q_E - S_E}$ От Свойство 6.1 следва, че $\sigma(|\alpha'|) \in Q_E$, и по индукционното предположение можем да твърдим, че $\sigma(|\alpha'|) \in L' \cap R'$. Тъй като $q \notin S_E \cup Q_R$, каквато и да е стойността на $\psi(L', a, R)$, ще имаме $q \in \delta_1(L', \psi(L', a, R)) = L$.

$\mathbf{q \in S_E}$ Заради вида на изпълненията на $\mathcal{A}$ (Свойство 6.1), имаме $\sigma(|\alpha'|) \in Q_L$. Тогава по Лема 6.1, $\sigma(|\alpha'|) \in L'$. Да допуснем, че $q \notin L$. Това би било възможно, единствено ако $\psi(L', a, R) = a^c$, което от своя страна може да се случи, единствено ако $\delta_1(L', a) \cap R \cap (Q_E - S_E - F_E) \neq \emptyset$. Нека q_0 е свидетел за това ($q_0 \in \delta_1(L', a) \cap R \cap (Q_E - S_E - F_E)$). Тогава $q_0 \in L \cap R$ и $q_0 \in (Q_E - S_E - F_E)$. Тъй като $\langle \alpha, q_0\rangle \prec \langle \alpha, q\rangle$, можем да приложим индукционното предположение за $\langle \alpha, q_0\rangle$ и да твърдим, че има $\sigma_0 \in \mathcal{X}_v$, такова, че $\sigma_0(|\alpha|) = q_0$. От Свойство 6.1 и от факта, че $q_0 \in (Q_E - S_E - F_E)$, излиза, че има $\langle u_0, v_0, w_0\rangle \in \mathcal{C}_v$ и $u_0 \subset \alpha \subset u_0 v_0$. Но ако σ е изпълнение за $\langle u, v, w\rangle \in \mathcal{C}_v$, веднага се вижда, че $\alpha = u$, и следователно $\langle u, v, w\rangle \notin \mathcal{C}_v$ (има валиден контекст, който го застъпва и от Твърдение 2.2). Това е противоречие с допускането ни, че $q \notin L$. Така показахме, че $q \in L \cap R$.

□

6.4. Бимашина за контекстно правило

Най-накрая, строим $\mathcal{B}^I$ - класическа бимашина, работеща върху входа си, еквивалентна на $\mathcal{B}$. Въз основа на $\mathcal{B}^I = \langle \mathcal{A}_1^I, \mathcal{A}_2^I, \psi^I\rangle$ строим бимашина

за контекстното правило $E \to \beta \ / \ L_R$

$$\mathcal{B}' = \langle \mathcal{A}'_1, \mathcal{A}'_2, \psi' \rangle$$

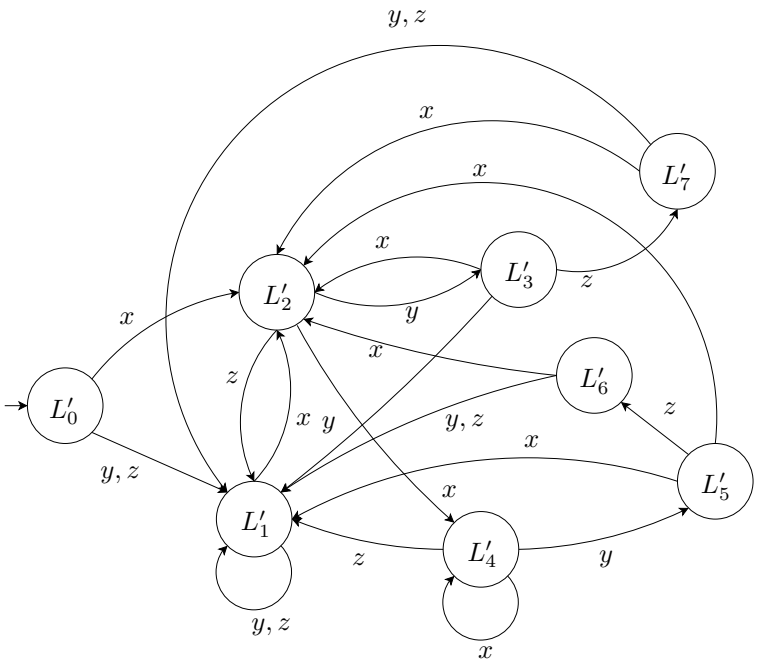
където $\mathcal{A}'_1 = \mathcal{A}^I_1$, $\mathcal{A}'_2 = \mathcal{A}^I_2$, и единствената разлика с $\mathcal{B}^I$, изходната функция ψ', е дефинирана както следва

$$\psi'(L, a, R) = \psi'_1(L, a, R) \cdot \psi'_2(L, a, R)$$

където

$$\psi'_1(L, a, R) = \begin{cases} \epsilon & \text{ако } \langle L', \delta'_2(R, a) \rangle \in L \ \& \ L' \cap \delta'_2(R, a) \cap (Q_E - S_E - F_E) \neq \emptyset \\ \beta & \text{ако } \langle L', \delta'_2(R, a) \rangle \in L \ \& \ L' \cap \delta'_2(R, a) \cap (S_E - F_E) \neq \emptyset \\ \beta a & \text{ако } \langle L', \delta'_2(R, a) \rangle \in L \ \& \ L' \cap \delta'_2(R, a) \cap (S_E \cap F_E) \neq \emptyset \\ a & \text{в противен случай} \end{cases}$$

$$\psi'_2(L, a, R) = \begin{cases} \beta & \text{ако } \langle L', R \rangle \in \delta'_1(L, a) \ \& \ L' \cap R \cap S_E \neq \emptyset \ \& \ R = q'_2 \\ \epsilon & \text{в противен случай} \end{cases}$$



Фиг. 9. Левият автомат на $\mathcal{B}'$ за правилото $xy|yz \to \beta \ / \ x_z$. За яснота са пропуснати преходите към L'_0

Лема 6.3. *Нека $t = \alpha\beta \in \Sigma^*$, $\delta'^*_1(q'_1, \alpha) = L$, $\delta'^*_2(q'_2, \widetilde{\beta}) = R'$. Тогава $\langle L', R' \rangle \in L$ и $q \in L' \cap R' \cap Q_E$, точно тогава, когато съществува $\langle u, v, w \rangle \in \mathcal{C}_v(t; L, E, R)$ и изпълнение за него $\sigma \in \mathcal{X}_v(t; L, E, R)$, такова, че $\sigma(|\alpha|) = q$.*

Доказателство. Нека $\langle L', R' \rangle \in L$ и $q \in L' \cap R' \cap Q_E$. Тогава от Твърдения 5.1 и 5.2 веднага следва, че $R' = \delta_2^*(q_2, \widetilde{\beta})$ и $L' = \delta_1^*(q_1, \psi^{**}(q_1, \alpha, R'))$. Сега тъй като $q \in L' \cap R' \cap Q_E$, от Твърдение 6.1 можем да заключим, че съществува валиден контекст $\langle u, v, w \rangle \in \mathcal{C}_v$ и изпълнение за него $\sigma \in \mathcal{X}_v$, такива, че $\sigma(|\alpha|) = q$.

В обратната посока доказателството е аналогично. □

Сега вече сме готови да покажем, че така конструираната бимашина наистина работи както очакваме, а именно – извършва заместване по контекстното правило $E \to \beta \ / \ L_R$.

Твърдение 6.2. *Нека $E \to \beta \ / \ L_R$ е контекстно правило за заместване и $\mathcal{B}'$ е бимашината, построена за него. Нека още $\alpha \in \Sigma^*$ е произволна дума над азбуката Σ. Тогава $\mathcal{B}'(\alpha)$ е точно резултатът от прилагането на контекстното правило върху α.*

Доказателство. Ако $\alpha = \epsilon$, по Дефиниция 2.6, резултатът от прилагането на $E \to \beta \ / \ L_R$ е точно $\epsilon = \mathcal{B}'(\alpha)$.

Нека $\alpha = \overline{a_1 a_2 \dots a_n}$, където $a_i \in \Sigma$ за $1 \leq i \leq n$. Нека $(\omega_1 \pi_1) \cdot (\omega_2 \pi_2) \dots \cdot (\omega_n \pi_n)$ е резултатът от прилагането на правилото, определен от Дефиниция 2.6. За произволно $i \in [1, n]$ ще покажем, че ако $\delta_1'^*(q_1', \overline{a_1 a_2 \dots a_{i-1}}) = L$, а $\delta_2'^*(q_2', \overline{a_n a_{n-1} \dots a_{i+1}}) = R$, то $\psi_1'(L, a_i, R) = \omega_i$ и $\psi_2'(L, a_i, R) = \pi_i$.

Според дефиницията на ψ_1' трябва да разгледаме четири случая:

I случай Нека е вярно $\langle L', \delta_2'(R, a) \rangle \in L$ & $L' \cap \delta_2'(R, a) \cap (Q_E - S_E - F_E) \neq \emptyset$. Тогава според Лема 6.3 има валиден контекст $\langle u, v, w \rangle \in \mathcal{C}_v$ и изпълнение за него $\sigma \in \mathcal{X}_v$, такова, че $\sigma(i-1) \in (Q_E - S_E - F_E)$. Заради вида на изпълненията на $\mathcal{A}$ (Свойство 6.1) следва, че $|u| < i - 1 < |uv|$. Това според Дефиниция 2.6 означава, че $\omega_i = \epsilon$.

II случай Нека е вярно $\langle L', \delta_2'(R, a) \rangle \in L$ & $L' \cap \delta_2'(R, a) \cap (S_E - F_E) \neq \emptyset$. Тогава според Лема 6.3 има валиден контекст $\langle u, v, w \rangle \in \mathcal{C}_v$ и изпълнение за него $\sigma \in \mathcal{X}_v$, такова, че $\sigma(i-1) \in (S_E - F_E)$. Заради вида на изпълненията на $\mathcal{A}$ (Свойство 6.1) следва, че $|u| = i - 1 < |uv|$. Това според Дефиниция 2.6 означава, че $\omega_i = \beta$.

III случай Нека е вярно $\langle L', \delta_2'(R, a) \rangle \in L$ & $L' \cap \delta_2'(R, a) \cap (S_E \cap F_E) \neq \emptyset$. Тогава според Лема 6.3 има валиден контекст $\langle u, v, w \rangle \in \mathcal{C}_v$ и изпълнение за него $\sigma \in \mathcal{X}_v$, такова, че $\sigma(i-1) \in (S_E \cap F_E)$. Заради вида на изпълненията на $\mathcal{A}$ (Свойство 6.1) следва, че $|u| = i - 1$. Тъй като не сме попаднали във II случай следва, че не съществува $\langle u, v, w \rangle \in \mathcal{C}_v$, такъв, че $|v| \neq \epsilon$, и следователно (от Дефиниция 2.6) $\omega_i = \beta a_i$.

IV случай Нека не сме попаднали в нито един от предните три случая. Сега да допуснем, че съществува валиден контекст $\langle u, v, w \rangle \in \mathcal{C}_v$, такъв, че $|u| \leq i - 1 < |uv|$ или $|u| = i - 1 = |uv|$. Тогава има изпълнение за него $\sigma \in \mathcal{X}_v$, такова, че $\sigma(i-1) \in Q_E - F_E$ или $\sigma(|i-1|) \in S_E \cap F_E$. Тогава (според Лема 6.3) $\langle L', \delta_2'(R, a) \rangle \in L$ и $\sigma(i-1) \in L' \cap \delta_2'(R, a) \cap Q_E - F_E$ или $\sigma(i-1) \in L' \cap \delta_2'(R, a) \cap S_E \cap F_E$, което е противоречие, защото не сме попаднали в нито един от първите три случая. Така заключаваме, че $\omega_i = a_i$.

	L'_0	L'_1	L'_2	L'_3	L'_4	L'_5	L'_6	L'_7
R'_0	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R'_1	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R'_2	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	z/ϵ	y/ϵ	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R'_3	$\cdot/\cdot$	$\cdot/\cdot$	x/β	$\cdot/\cdot$	x/β	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R'_4	$\cdot/\cdot$	$\cdot/\cdot$	y/β	z/ϵ	y/ϵ	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R'_5	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
R'_6	$\cdot/\cdot$	$\cdot/\cdot$	x/β	$\cdot/\cdot$	x/β	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$

Фиг. 10. Изходната функция на $\mathcal{B}'$ за правилото $xy|yz \to \beta \ / \ x_z$

Така показахме, че $\psi'_1(L, a_i, R) = \omega_i$ за $i = 1, 2, \ldots, n$. Остава да покажем, че $\psi'_2(L, a_i, R) = \pi_i$ за всяко $i = 1, 2, \ldots, n$.

Нека $\psi'_2(L, a_i, R) = \beta$. Тогава $\langle L', R\rangle \in \delta'_1(L, a)$ и $L' \cap R \cap S_E \neq \emptyset$. Според Лема 6.3 ще има $\langle u, v, w\rangle \in \mathcal{C}_v$ и изпълнение за него $\sigma \in \mathcal{X}_v$, такива, че $\sigma(i) \in S_E$. От друга страна $R = q'_2$ и заради нормалната форма на $\mathcal{A}$ следва, че $\alpha = t$, и следователно $\langle u, v, w\rangle = \langle \alpha, \epsilon, \epsilon\rangle \in \mathcal{C}_v$. От това, разбира се следва, че $\pi_i = \beta$.

Аналогично показваме, че ако $\pi_i = \beta$, то $\psi'_2(L, a_i, R) = \beta$.

С това завършихме доказателството, че $\psi'(L, a_i, R) = \omega_i\pi_i$, и следователно $\mathcal{B}'(\alpha)$ е точно резултатът от приложението на $E \to \beta \ / \ L_R$ върху α. □

Пример 6.1. *Да изпълним бимашината, построена за* $xy|yz \to B \ / \ x_z$ *върху текста* $xyzzxxyzz$ *от Пример 2.2. Изпълнението на* $\mathcal{A}'_1$ *ще бъде редицата* $L'_0, L'_2, L'_3, L'_7, L'_1, L'_2, L'_4, L'_5, L'_6, L'_1$. *Десният автомат* $\mathcal{A}'_2$ *ще премине през* $R'_0, R'_2, R'_4, R'_6, R'_5, R'_1, R'_2, R'_4, R'_6, R'_5$, *четейки текста отдясно наляво. Според Фиг. 10, изходът на* $\mathcal{B}'$ *е точно* $xBzxBzz$.

7. Алгоритми

В тази глава ще представим някои от по-интересните алгоритми, необходими за конструкцията на бимашина по контекстно правило.

Алгоритмите са изложени без доказателство за коректност, тъй като следват без изменение съответните конструкции, показани по-рано.

7.1. Конкатенация отляво

Алгоритъмът `lconcat` за конкатенация "отляво"съответства на конструкцията от Дефиниция 3.8.

```
subroutine lconcat(Automaton A, Automaton B)
   let C = new Automaton

   foreach state in A.states
      let cloned_state = C.add_state(state)
      if state.is_start then
         cloned_state.set_start(true)
      end if
   end foreach

   foreach state in B.states
      let cloned_state = C.add_state(state)
      if state.is_start and A.accepts("") then
         cloned_state.set_start(true)
      end if
      if state.is_final then
         cloned_state.set_final(true)
      end if
   end foreach

   foreach trans in A.transitions
      C.add_transition(trans.source, trans.char, trans.target)
      if trans.target.is_final then
         foreach state in B.start_states
            C.add_transition(trans.source, trans.char, state)
         end foreach
      end if
   end foreach
   foreach trans in B.transitions
      C.add_transition(trans.source, trans.char, trans.target)
   end foreach

   return C
end subroutine
```

7.2. Превод на бимашина, работеща върху изхода си

Следващият алгоритъм `output_to_input` съответства на Конструкция 5.1 и по зададена бимашина `B` , работеща върху изхода си, строи еквивалентната класическа бимашина, работеща върху входа си.

```
subroutine output_to_input(Bimachine B)
   let psi = B.output_function
   let AL  = B.left_automaton
   let AR  = B.right_automaton

   let AN = new Automaton
   let states = new Table
   foreach state1 in AL.states
      foreach state2 in AR.states
         states[state1][state2] = AN.add_state()
      end foreach
   end foreach
   foreach p1 in AL.states
      foreach q1 in AR.states
         foreach p2 in AL.states
            foreach q2 in AR.states
               foreach a in alphabet
                  if AR.trans_function(q2, a) = q1 and
                     AL.trans_function(p1, psi(p1, a, q2)) = p2 then
                     AN.add_transition(states[p1][q1], a, states[p2][q2])
                  end if
               end foreach
            end foreach
         end foreach
      end foreach
   end foreach

   let A1 = AN.build_deterministic()

   let new_psi = new Function
   foreach L in A1.states
      foreach a in alphabet
         foreach r in AR.states
            foreach p in AL.states
               foreach q in AR.states
                  if states[p][q] in L and
                     AR.trans_function(r, a) = q then
                     new_psi.define(L, a, r, psi(p, a, r))
                  end if
               end foreach
            end foreach
```

```
         end foreach
      end foreach
   end foreach

   return new Bimachine(A1, AR, new_psi)
end subroutine
```

7.3. Конструкция на бимашина по контекстно правило

Алгоритъмът `construct_bimachine` директно съответства на конструкцията от Глава 6 и по зададено контекстно правило за заместване строи бимашина, която го реализира.

```
subroutine construct_bimachine(L, E, R, word)
   let AL = new Automaton(".*"+L)
   let AE = new Automaton(E)
   let AR = new Automaton(R+".*")

   let A = rconcat(lconcat(AL, AE), AR)

   let A2 = A.reverse().build_deterministic()
   let AN = A.clone()

   foreach tr in AN.trans
      if tr.target in AR.states then
         AN.remove_transition(tr)
      else
         if tr.target not in AE.start_states then
            AN.add_transition(tr.source, tr.char + alphabet.size, tr.target)
         end if
      end if
   end foreach

   let A1 = AN.build_deterministic()

   let psi = new Function
   let inter = AE.states - AE.start_states - AE.final_states
   foreach L in A1.states
      foreach R in A2.states
         foreach a in alphabet
            if A1.trans_function(L, a).intersect(R).intersect(inter) then
               psi.define(L, a, R, a + alphabet.size)
            else
               psi.define(L, a, R, a)
            end if
```

```
      end foreach
   end foreach
end foreach

let B = output_to_input(new Bimachine(A1, A2, psi))

let final_psi = new Function
let intermediate  = AE.states - AE.start_states - AE.final_states
let startnonfinal = AE.start_states - AE.final_states
let startandfinal = AE.start_states.intersect(AE.final_states)
foreach L in B.left_automaton.states
   foreach R in B.right_automaton.states
      foreach a in alphabet
         let result = a
         foreach (p,q) in L
            continue unless q = B.right_automaton.trans_function(R, a)
            let common = p.intersect(q)
            if common.intersect(intermediate).length > 0 then
               result = ""
               break
            else if common.intersect(startnonfinal).length > 0 then
               result = word
               break
            else if common.intersect(startandfinal).length > 0 then
               result = word + a
               break
            end if
         end foreach
         if R = B.right_automaton.start_state then
            foreach (p,q) in B.left_automaton.trans_function(L, a)
               continue unless q = R
               if p.intersect(q).intersect(AE.start_states).lenght > 0 then
                  result = result + word
               end if
            end foreach
         end if
         final_psi.define(L, a, R, result)
      end foreach
   end foreach
end foreach

B.output_function = final_psi

return B
end subroutine
```

8. Сложност

В тази глава ще дадем горна граница за размера на така конструираната бимашина.

Твърдение 8.1. *Нека $\mathcal{B} = \langle \mathcal{A}_L, \mathcal{A}_R, \psi \rangle$ е бимашина, работеща върху изхода си, а $\mathcal{B}' = \langle \mathcal{A}'_L, \mathcal{A}'_R, \psi' \rangle$ е еквивалентна на нея класическа бимашина, построена с Конструкция 5.1. Тогава*

а) $|Q'_R| = |Q_R|$

б) $|Q'_L| \leq (|Q_L| + 1)^{|Q_R|}$

Доказателство. а) От Конструкция 5.1 веднага се вижда, че $\mathcal{A}_R = \mathcal{A}'_R$, с което $|Q'_R| = |Q_R|$ е тривиално вярно.

б) Поради Твърдение 5.1 можем да разглеждаме състоянията в Q'_L като частични функции от Q_R в Q_L. Това означава, че $|Q'_L|$ не надминава броя на частичните функции, дефинирани от Q_R в Q_L, и следователно $|Q'_L| \leq (|Q_L| + 1)^{|Q_R|}$.

□

Сега да преминем към изчисляването на горна граница за сложността на бимашината, построена в Глава 6. Иначе казано, ще оценим броя на състоянията на двата автомата въз основа на контекстното правило.

Нека фиксираме едно контекстно правило за заместване $E \to \beta \;/\; L_R$ и означим $l = |L|$, $e = |E|$ и $r = |R|$. Тогава броят на състоянията на $\mathcal{A}_L$, $\mathcal{A}_E$ и $\mathcal{A}_R$ ще е съответно $\mathcal{O}(l)$, $\mathcal{O}(e)$ и $\mathcal{O}(r)$. Оттук веднага следва, че състоянията на $\mathcal{A}$ са $\mathcal{O}(l + e + r)$.

Това ни дава горна граница за състоянията на $\mathcal{A}_2$, а именно $\mathcal{O}(2^{l+e+r})$.

Тъй като $\mathcal{A}_1^N$ се получава от $\mathcal{A}$ чрез модификация на релацията на прехода, броят на състоянията му остава същият ($\mathcal{O}(l + e + r)$). След детерминизация до $\mathcal{A}_1$ тази сложност нараства до $\mathcal{O}(2^{l+e+r})$. Най-накрая, според Твърдение 8.1, Конструкция 5.1 строи $\mathcal{A}'_1$ с $\mathcal{O}(2^{(l+e+r)2^{l+e+r}})$ състояния.

Така получихме и крайната оценка за броя на състоянията на левия и десния автомат на бимашината — съответно $\mathcal{O}(2^{n2^n})$ и $\mathcal{O}(2^n)$, където $n = l + e + r$ е общата дължина на регулярните изрази в контекстното правило за заместване.

Литература

[1] Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of English.* New York: Harper and Row.

[2] C. Douglas Johnson. 1972. *Formal Aspects of Phonological Description.* Mouton, The Hague.

[3] Kaplan, Ronald M. and Martin Kay. 1994. *Regular models of phonological rule systems.* Computational Linguistics, 20(3):331–378.

[4] Schutzenberger, Marcel Paul. 1961. *A remark on finite transducers.* Information and Control, 4:185–187.

[5] E. Roche and Y. Schabes. 1996. *Introduction to finite-state devices in natural language processing.* Technical report, Mitsubishi Electric Research Laboratories, TR-96-13.

[6] Wojciech Skut, Stefan Ulrich, Kathrine Hammervold. 2004. *A Bimachine Compiler for Ranked Tagging Rules.* CoRR cs.CL/0407046.

[7] K. Thompson. 1968. *Regular expression search algorithm.* Communications of the ACM, 11(6):419– 422