Алгоритъм на Божозовски за строене на хубав* автомат

Иво Стратев

9 юни 2020 г.

1 Да си припомним някои неща

Със Σ означавахме крайната азбука, над която работим. Това беше просто крайно и непразно множество. Множеството на всички думи над азбуката Σ бележихме със Σ^*. Език над азбуката Σ за нас беше всяко подмножество на Σ^*. Значи множеството на всички езици над азбуката Σ е степенното множество на Σ^*, тоест $\mathcal{P}(\Sigma^*)$. Празният език е $\emptyset$, независимо от азбуката. Езикът на празната дума е $\{\varepsilon\}$, независимо от азбуката. Ако $u \in \Sigma$, то $\{u\}$ е езикът на думата с единствена буква u и $\{u\}$ е език над Σ. Ако A и B са езици над Σ, то конкатенацията на A и B беше $A \cdot B = \{\alpha.\beta \mid \alpha \in A \ \&\ \beta \in B\}$ и също е език над същата азбука Σ. Ако A и B са езици над Σ, то $A \cup B$ също е език над Σ.

Нека L е език над Σ. Тогава $L^0 = \{\varepsilon\}$ и ако $n \in \mathbb{N}$, то $L^{n+1} = L \cdot L^n$ и $L^* = \bigcup\limits_{k \in \mathbb{N}} L^k$ и $L^+ = L \cdot L^* = \bigcup\limits_{k \in \mathbb{N}} L^{k+1}$. Очевидно L^* и L^+ са езици над Σ.

1.1 Примери (Сметки)

Имаме $\{a\}^* = \bigcup\limits_{n \in \mathbb{N}} \{a\}^n = \bigcup\limits_{n \in \mathbb{N}} \{a^n\} = \{a^n \mid n \in \mathbb{N}\}$ и $\{a\}^+ = \{a\} \cdot \{a\}^*$.
Значи $\{a\}^* = \{a^0\} \cup \{a^n \mid n \in \mathbb{N}_+\} = \{\varepsilon\} \cup \{a\}^+ = \{\varepsilon\} \cup \{a\} \cdot \{a\}^*$.

Да направим една по-голяма сметка:

$$
\begin{aligned}
& L \\
&= \{a,b\}^* \cdot \{c\}^* = (\{\varepsilon\} \cup \{a,b\}^+) \cdot \{c\}^* \\
&= (\{\varepsilon\} \cup (\{a,b\} \cdot \{a,b\}^*) \cdot \{c\}^* \\
&= (\{\varepsilon\} \cup (\{a\} \cup \{b\}) \cdot \{a,b\}^*) \cdot \{c\}^* \\
&= (\{\varepsilon\} \cup \{a\} \cdot \{a,b\}^* \cup \{b\} \cdot \{a,b\}^*) \cdot \{c\}^* \\
&= \{\varepsilon\} \cdot \{c\}^* \cup \{a\} \cdot \{a,b\}^* \cdot \{c\}^* \cup \{b\} \cdot \{a,b\}^* \cdot \{c\}^* \\
&= \{c\}^* \cup \{a\} \cdot L \cup \{b\} \cdot L \\
&= \{\varepsilon\} \cup \{c\} \cdot \{c\}^* \cup \{a\} \cdot L \cup \{b\} \cdot L
\end{aligned}
$$

Следователно L може да бъде представен като обединение на езици започващи с конкретна буква или на празната дума. Нещо, което ще ни бъде полезно за алгоритъма, който ще разгледаме.

2 Операцията, която ще ни трябва

Нека Σ е крайна азбука.
Дефинириме операция $remove_\Sigma : \Sigma \times \mathcal{P}(\Sigma^*) \to \mathcal{P}(\Sigma^*)$ по следния начин $remove_\Sigma(x, L) = \{\alpha \in \Sigma^* \mid x.\alpha \in L\}$. Очевидно тя е коректно дефинирана операция, тоест пресмятайки я върху конкретна буква x от Σ и конкретен език L над Σ получаваме друг език над Σ $\{\alpha \in \Sigma^* \mid x.\alpha \in L\}$, защото отделяме от Σ^*. Идеята зад дефинираната операция е все едно да махнем буквата x от думите започващи с x от L и като резултат да получаваме остатъците от думите след махането. Като под махан разбираме прочитане на буквата.

2.1 Примерни сметки:

Очевидно $remove_\Sigma(x, \emptyset) = \emptyset$ понеже в празният език няма думи.
Също $remove_\Sigma(x, \{\varepsilon\}) = \{\alpha \in \Sigma^* \mid x.\alpha \in \{\varepsilon\}\} = \emptyset$ понеже $\{\varepsilon\}$ има единствена дума, която няма букви, тоест няма какво да махнем от нея.
Ако $x \in \Sigma$, то $remove_\Sigma(x, \{x\}) = \{\alpha \in \Sigma^* \mid x.\alpha \in \{x.\varepsilon\}\} = \{\varepsilon\}$, понеже $\{x\} = \{x.\varepsilon\}$.
Ако x и y са различни букви от Σ, то
$remove_\Sigma(x, \{y\}) = \{\alpha \in \Sigma^* \mid x.\alpha \in \{y\}\} = \emptyset$.

За момента фиксираме азбуката $\Sigma = \{a, b, c\}$ и за това вместо $remove_{\{a,b,c\}}$ ще пишем просто rem. Нека направим няколко сметки:

$$rem(a, \{aa\}) = \{a\} \quad rem(a, \{ba\}) = \emptyset$$
$$rem(a, \{aa, ba\}) = rem(a, \{aa\} \cup \{ba\}) = rem(a, \{aa\}) \cup rem(a, \{ba\}) = \{a\} \cup \emptyset = \{a\}$$
$$rem(a, \{a\}^+) = rem(a, \{a\} \cdot \{a\}^*) = \{a\}^*$$
$$rem(a, \{a\}^*) = rem(a, \{\varepsilon\} \cup \{a\} \cdot \{a\}^*) = \{a\}^*$$
$$rem(a, \{aa\}^+) = rem(a, \{aa\} \cdot \{aa\}^*) = \{a\} \cdot \{aa\}^*$$

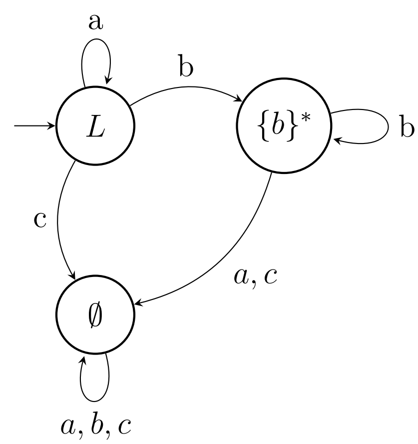
Нека $L = \{a\}^* \cdot \{b\}^+ = (\{\varepsilon\} \cup \{a\} \cdot \{a\}^*) \cdot \{b\}^+ = \{b\}^+ \cup \{a\} \cdot \{a\}^* \cdot \{b\}^+ = \{b\} \cdot \{b\}^* \cup \{a\} \cdot L$. Тогава

$$\begin{aligned}
rem(a, L) &= rem(a, \{b\} \cdot \{b\}^* \cup \{a\} \cdot L) = L \\
rem(b, L) &= rem(b, \{b\} \cdot \{b\}^* \cup \{a\} \cdot L) = \{b\}^* \\
rem(c, L) &= rem(c, \{b\} \cdot \{b\}^* \cup \{a\} \cdot L) = \emptyset \\
rem(a, \{b\}^*) &= \emptyset \\
rem(b, \{b\}^*) &= \{b\}^* \\
rem(a, \{b\}^*) &= \emptyset \\
rem(a, \emptyset) &= rem(b, \emptyset) = rem(c, \emptyset) = \emptyset
\end{aligned}$$

Значи

$$\begin{aligned}
rem(a, L) &= L \\
rem(b, L) &= \{b\}^* \\
rem(c, L) &= \emptyset \\
rem(a, \{b\}^*) &= \emptyset \\
rem(b, \{b\}^*) &= \{b\}^* \\
rem(c, \{b\}^*) &= \emptyset \\
rem(a, \emptyset) &= \emptyset \\
rem(b, \emptyset) &= \emptyset \\
rem(c, \emptyset) &= \emptyset
\end{aligned}$$

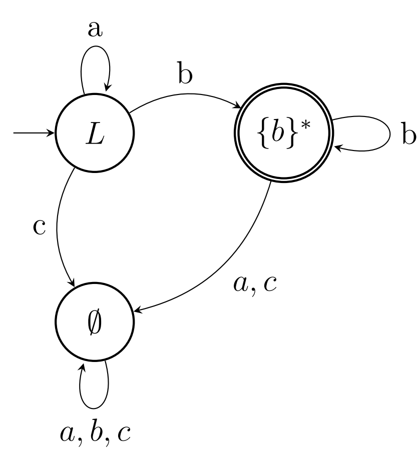
Ако "навържем"сметките от преди малко получаваме следния автомат:



Така констуирания автомат няма нито едно финално състояние, тоест той разпознава празният език. На нас ни се иска, той да разпознава езикът $L = \{a\}^* \cdot \{b\}^+$. За целта финални са всички състояния, които съдържат празната дума ε. Имаме:

$$\begin{aligned} L = \{b\} \cdot \{b\}^* \cup \{a\} \cdot L & \quad \text{не съдържа празната дума} \\ \{b\}^* = \{\varepsilon\} \cup \{b\}^+ & \quad \text{съдържа празната дума} \\ \emptyset & \quad \text{не съдържа празната дума} \end{aligned}$$

Така само $\{b\}^*$ става финално състояние и получаваме следния автомат



Лесно се съобразява, че този автомат разпознава $L = \{a\}^* \cdot \{b\}^+$.

3 Алгоритъм на Божозовски

Това, което може да забележим е, че постройхме автомат пресмятайки операцията *rem* върху различни езици, започвайки от езика, за който искаме да конструираме автомат, докато получаваме нови езици с всяка буква на азбуката. Състоянията бяха различните езици, началното, езика за който конструираме автомат, а финални са състоянията, които съдържат празната дума. Така готови сме да представим чрез псевдо код алгоритъма на Божозовкси.

```
function bojoAlgorithm(language) {
    const Sigma = alphabet(language);
    let languages = [language];
    let notVisited = [language];
    let transitions = [];
    let finals = [];
    while(!notVisited.isEmpty()) {
        const current = notVisited.pop();
        for(let letter in Sigma) {
            const result = rem(letter, current);
            transitions.insert([current, letter, result]);
            if(!languages.isMember(result)) {
                notVisited.push(result);
                languages.push(result);
            }
        }
        if(includesEmptyWord(current)) {
            finals.insert(current);
        }
    }
    return [Sigma, languages, transitions, language, finals];
}
```

3.1 Няколко практически съвета

- Преди да тръгнем да смятаме най-добре е да представим езика, от който махаме букви като **обединие** на празния език или езици **започващи с конкретна буква**.

- След като се сметне $rem(x, M)$ трябва **много внимателно** да се провери, дали е **нов език**.

- Резултата от сметката $rem(x, M)$ е добре да **подчертаем** ако е **нов език** (такъв какъвто не сме срещали досега) и е добре да **заградим**, ако видим, че **съдържа празната дума**.

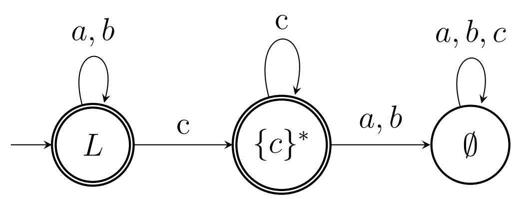
- Добре е поватрарящи се езици, участващи като изрази ако не са крайни и не са кратки за писане да означаваме по някакъв начин с цел съкращаване.

4 Пример 1

Нека постройм по разгледания алгоритъм автомат за езикът $L = \{a, b\}^* \cdot \{c\}^*$. В началото видяхме, че $L = \{\varepsilon\} \cup \{c\} \cdot \{c\}^* \cup \{a\} \cdot L \cup \{b\} \cdot L$. (направихме една по-голяма сметка) Понеже L съдържа празната дума и за нас е нов език, то следвайки съветите $\boxed{\underline{L}}$.
Започваме да смятаме:

$$\begin{aligned}
rem(a,\ L) &= rem(a,\ \{\varepsilon\} \cup \{c\} \cdot \{c\}^* \cup \{a\} \cdot L \cup \{b\} \cdot L) = L \\
rem(b,\ L) &= rem(b,\ \{\varepsilon\} \cup \{c\} \cdot \{c\}^* \cup \{a\} \cdot L \cup \{b\} \cdot L) = L \\
rem(c,\ L) &= rem(c,\ \{\varepsilon\} \cup \{c\} \cdot \{c\}^* \cup \{a\} \cdot L \cup \{b\} \cdot L) = \boxed{\underline{\{c\}^*}} \\
& \qquad rem(a,\ \{c\}^*) = \underline{\emptyset} \\
& \qquad rem(b,\ \{c\}^*) = \emptyset \\
& \qquad rem(c,\ \{c\}^*) = \{c\}^* \\
& \qquad rem(a,\ \emptyset) = \emptyset \\
& \qquad rem(b,\ \emptyset) = \emptyset \\
& \qquad rem(c,\ \emptyset) = \emptyset
\end{aligned}$$

Така получаваме следния автомат



5 Пример 2

Нека постройм по разгледания алгоритъм автомат за един познат за нас език.

$$L = \{\omega \in \{a,b\}^* \mid \mathcal{N}_a(\omega) \equiv 0 \pmod 3\}$$

Очевидно сменяме и азбуката вече $\Sigma = \{a,b\}$, но продължаваме да пишем просто rem. Нека изразим L в **удобен за смятане вид**, тоест като **обединие на езици започващи с фиксирана буква** или празната дума.

$$L = \{\varepsilon\} \cup \{\omega \in \{a,b\}^* \mid \mathcal{N}_a(\omega) \equiv 0 \pmod 3 \;\&\; |\omega| > 0\}$$
$$= \{\varepsilon\} \cup \{a\} \cdot \{\omega \in \{a,b\}^* \mid \mathcal{N}_a(\omega) \equiv 2 \pmod 3\} \cup \{b\} \cdot \{\omega \in \{a,b\}^* \mid \mathcal{N}_a(\omega) \equiv 0 \pmod 3\}$$

Да означим $\{\omega \in \{a,b\}^* \mid \mathcal{N}_a(\omega) \equiv 2 \pmod 3\}$ с L_2. Тогава $L = \{\varepsilon\} \cup \{a\} \cdot L_2 \cup \{b\} \cdot L$. Следвайки съветите $\boxed{\underline{L}}$.

Пресмятаме

$$rem(a,\ L) = rem(a,\ \{\varepsilon\} \cup \{a\} \cdot L_2 \cup \{b\} \cdot L) = \underline{L_2} = \{\omega \in \{a,b\}^* \mid \mathcal{N}_a(\omega) \equiv 2 \pmod 3\}$$
$$rem(b,\ L) = rem(b,\ \{\varepsilon\} \cup \{a\} \cdot L_2 \cup \{b\} \cdot L) = L$$

Изразяваме L_2 като обединение:

$$L_2 = \{\omega \in \{a,b\}^* \mid \mathcal{N}_a(\omega) \equiv 2 \pmod 3\}$$
$$= \{a\} \cdot \{\omega \in \{a,b\}^* \mid \mathcal{N}_a(\omega) \equiv 1 \pmod 3\} \cup \{b\} \cdot \{\omega \in \{a,b\}^* \mid \mathcal{N}_a(\omega) \equiv 2 \pmod 3\}$$

Да означим $\{\omega \in \{a,b\}^* \mid \mathcal{N}_a(\omega) \equiv 1 \pmod 3\}$ с L_1. Тогава $L_2 = \{a\} \cdot L_1 \cup \{b\} \cdot L_2$.

Правим още пресмятания

$$rem(a,\ L_2) = rem(a,\ \{a\} \cdot L_1 \cup \{b\} \cdot L_2) = \underline{L_1} = \{\omega \in \{a,b\}^* \ \mid\ \mathcal{N}_a(\omega) \equiv 1 \pmod 3\}$$
$$rem(b,\ L_2) = rem(b,\ \{a\} \cdot L_1 \cup \{b\} \cdot L_2) = L_2$$

Изразяваме L_1 като обединение:

$$\begin{aligned} L_1 &= \{\omega \in \{a,b\}^* \ \mid\ \mathcal{N}_a(\omega) \equiv 1 \pmod 3\} \\ &= \{a\} \cdot \{\omega \in \{a,b\}^* \ \mid\ \mathcal{N}_a(\omega) \equiv 0 \pmod 3\} \cup \{b\} \cdot \{\omega \in \{a,b\}^* \ \mid\ \mathcal{N}_a(\omega) \equiv 1 \pmod 3\} \\ &= \{a\} \cdot L \cup \{b\} \cdot L_1 \end{aligned}$$

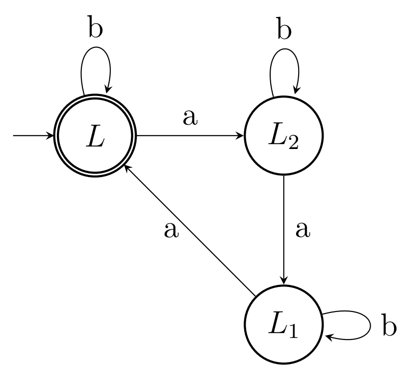
Правим още две пресмятания:

$$rem(a,\ L_1) = rem(a,\ \{a\} \cdot L \cup \{b\} \cdot L_1) = L$$
$$rem(b,\ L_1) = rem(a,\ \{a\} \cdot L \cup \{b\} \cdot L_1) = L_1$$

Така всички сметки са:

$$\begin{aligned} rem(a, L) &= L_2 \\ rem(b, L) &= L \\ rem(a, L_2) &= L_1 \\ rem(b, L_2) &= L_2 \\ rem(a, L_1) &= L \\ rem(b, L_1) &= L_1 \end{aligned}$$

В хода на пресмятанията единствено сме подчертали L_2 и L_1 и в началото оградихме L. С други думи само L съдържа празната дума, значи то ще единственото финално състояние. Така автомата разпонзаващ $L = \{\omega \in \{a,b\}^* \ \mid\ \mathcal{N}_a(\omega) \equiv 0 \pmod 3\}$ е



6 Задачи за упражнение

Постройте автомат по алгоритъма на Божозовски за езиците:

1. $\{a\} \cdot \{a, b\}^* \cdot \{b\}$
2. $(\{a, b\}^+ \cdot \{a\})^*$
3. $\{\omega \in \{a, b\}^* \mid \mathcal{N}_a(\omega) \equiv 0 \pmod 2 \;\&\; \mathcal{N}_b(\omega) \equiv 1 \pmod 2\}$