Reverse на дума и на език. Детерминиране на КНА

Иво Стратев

9 юни 2020 г.

1 Reverse на дума и на език

Нека Σ е крайна азбука. Дефинираме операция $\mathtt{rev}_\Sigma : \Sigma^* \to \Sigma^*$ чрез рекурсия така

$$\begin{aligned} \mathtt{rev}_\Sigma(\varepsilon) &= \varepsilon \\ \mathtt{rev}_\Sigma(x.\alpha) &= \mathtt{rev}_\Sigma(\alpha).x \end{aligned}$$

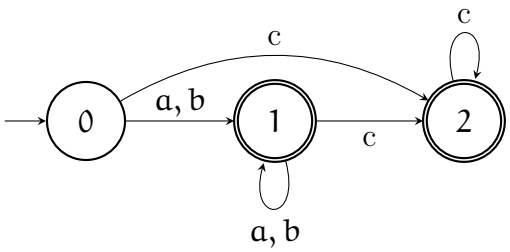
Това, което операцията $\mathtt{rev}_\Sigma$ прави е да обърне реда на буквите в една дума. Ето един пример

$$\begin{aligned} \mathtt{rev}(abc) &= \\ \mathtt{rev}(bc).a &= \\ (\mathtt{rev}(c).b).a &= \\ ((\mathtt{rev}(\varepsilon).c).b).a &= \\ (\varepsilon.c).b).a) &= \\ cba. & \end{aligned}$$

По естествен начин операцията $\mathtt{rev}_\Sigma$ може да бъде разширена от операция над думи до операция над езици $\mathtt{Reverse}_\Sigma : \mathcal{P}(\Sigma^*) \to \mathcal{P}(\Sigma^*)$. Дефинирана като образът на език при $\mathtt{rev}_\Sigma$. Тоест

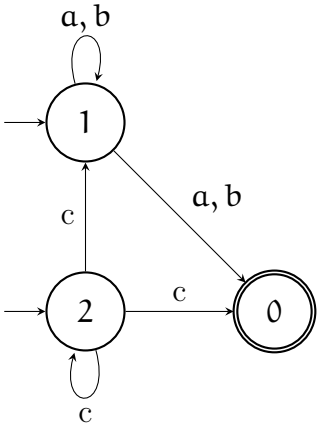
$$\mathtt{Reverse}_\Sigma(L) = \mathtt{rev}_\Sigma[L] = \{\mathtt{rev}_\Sigma(\alpha) \mid \alpha \in L\}$$

Да разгледаме езикът $L = \{c\}^+ \cup \{a, b\}^+ \cdot \{c\}^*$. Вземаме за очевидно, че $\mathtt{Reverse}_{\{a,b,c\}}(L) = \{c\}^+ \cup \{c\}^* \cdot \{a, b\}^+$. За L има КНА $\mathcal{N}$, който го разпознава.



Сега без доказателство ще дадем конструкция за получаването на автомат за езика $\mathsf{Reverse}_{\{a,b,c\}}(L)$ с две цели - да видим едно приложение на крайните недетерминирани автомати и с цел лесно получване на малък КНА, който да детерминираме. В него са думите от L прочетени на обратно.

Конструкцията е следната стройм КНА, който има същите състояния като $\mathcal{N}$. Ролите на начални и финални състояния са разменени и в релацията на преходите са разменени местата на състоянията в наредените тройки. Конкретно за разглежданият автомат $\mathcal{N}$ получаваме следният КНА.



В езика $\mathsf{Reverse}_{\{a,b,c\}}(L)$ са думите от L прочетени на обратно. За това искаме да започнем чететенето от някое финално състояние и да се движим на обратно по автомата $\mathcal{N}$, ако така по дадена дума успеем да достигнем до някое начално състояние тя ще е в езика $\mathsf{Reverse}_{\{a,b,c\}}(L)$.

2 Алгоритъм за детерминиране на КНА до тотален КДА

Нека $\mathcal{N} = \langle \Sigma, Q, S, \Delta, F \rangle$ е КНА. Нека $q \in Q$, $x \in \Sigma$ и нека означим с $\mathtt{States}_x(q)$ множеството $\{t \in Q \mid \langle q, x, t \rangle\}$. Множеството $\mathtt{States}_x(q)$ същност е множеството на всички състояния, в които може да се премине с буквата x от състояние q.

Ако фискираме буква $u \in \Sigma$, тогава $\mathtt{States}_u$ е функция от множеството на състоянията на автомата в степенното множество на множеството Q. Така можем да дефинираме функция $f : \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$, за която

$$(\forall T \in \mathcal{P}(Q))(\forall v \in \Sigma) \left(f(T, v) = \bigcup_{t \in T} \mathtt{States}_v(t) \right)$$

Ясно е, че $(\forall v \in \Sigma)(f(\emptyset, v) = \emptyset)$.

На базата на тези наблюдения се базира алгоритъма за детерминиране на КНА до тотален КДА. От теоретияна гледна точка горното наблюдение дава теоретична горна граница за броят на състоянията на автомата получен след детерминизация. Ако $n = |Q|$, то има тотален КДА еквиваленетен (разпознаващ същият език) на КНА $\mathcal{N}$ с най-много 2^n на брой състояния. Това е така, защото можем да получим еквивалентен КДА с множество от състояния $\mathcal{P}(Q)$, а както знаем от курса по ДС ако $n = |Q|$, то $|\mathcal{P}(Q)| = 2^n$. Тоест при детерминизация броят на състоянията може да нарастне експоненциално. На практика разбира се не винаги сме в най-лошият случай, по-често повечето подмножества на Q се оказват недостижими и могат да бъдат премахнати. Няма да се спираме на формална конструкция за получаването на тотален КДА по КНА, а с два примра ще покажем алгоритъма за детерминизация.

Стъпка по стъпка ще конструираме тотален КДА, като едновременно ще конструираме множеството от състоянита на автомата и функцията на преходите. Нека $n = |Q|$ и нека $Q = \{q_1, q_2, \ldots, q_n\}$. За описанието на всяка стъпка от алгоритъма ще изпозлваме таблица, която да ни казва от едно множество от състояния в кое множество от състояния трябва да се направи преход с дадена буква. Нека $T \in \mathcal{P}(Q)$, нека $z \in \Sigma$ и нека

$k = |T|$ и $T = \{t_1, t_2, \dots t_k\}$. В таблицата всеки ред отговаря на състояние от множеството T, а всеки стълб на състояние на автомата, който детерминираме. Непопълнена таблицата за множеството T и буквата z изглежда така.

z	q_1	q_2	$\dots$	q_n
t_1				
t_2				
$\vdots$				
t_k				

На позиция i, j се попълва чавка, ако в автомата $\mathcal{N}$ има преход от състояние t_i с буква z в състояние q_j, тоест $\langle t_i, z, q_j \rangle \in \Delta$.

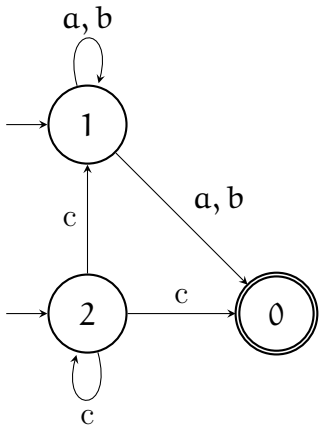
След като се попълнят редовете се тегли черта и се на новият ред в j-ят стълб се попълва чавка, ако за някое i на позиция i, j има чавка. Множеството от състояния, в което трябва да се отиде от множеството T с буква z е множеството на състоянията q_i, за $i \in \{1, 2, \dots, n\}$, за които на последният ред след чертата има чавка. Ясно е, че ако $k = 0$, то $T = \emptyset$ и таблицата няма редове. Следователно преходът, с коя да е буква от $\emptyset$ отново е в $\emptyset$.

Таблиците започват да се правят от множеството на началните състояния S на автомата $\mathcal{N}$ и се прави по една таблица за всяка буква от азбуката Σ. Таблиците се правят в опашков ред за всяко новополучено множество от състояния. В резултат ще сме конструирали множеството от състояния на детерминираният автомат и неговата функция на преходите. Начално състояние е множеството S, множеството на финалните състояния за онези подмнижества на Q, които съдържат поне едно финално състояние на автомата $\mathcal{N}$, тоест онези, които имат непразно сечение с F.

3 Пример за детерминизация

Разглеждаме автомата, който получихме за

$$\mathit{Reverse}_{\{a,b,c\}}(\{c\}^+ \cup \{a, b\}^+ \cdot \{c\}^*)$$



Формално имаме $\langle\{a, b, c\},\ \{0, 1, 2\},\ \{1, 2\},\ \Delta,\ \{0\}\rangle$, където релацията Δ описва преходите от диаграмата. Множеството на начални състояния е $\{1, 2\}$, азбуката е $\{a, b, c\}$, така че правим по една таблица за всяка буква.

a	0	1	2
1	✓	✓	
2			
	0	1	

b	0	1	2
1	✓	✓	
2			
	0	1	

c	0	1	2
1			
2	✓	✓	✓
	0	1	2

Така получихме следните преходи

$$\{1, 2\} \overset{a}{\longmapsto} \{0, 1\}$$
$$\{1, 2\} \overset{b}{\longmapsto} \{0, 1\}$$
$$\{1, 2\} \overset{c}{\longmapsto} \{0, 1, 2\}$$

Правим таблици за множеството $\{0, 1\}$.

a	0	1	2
0	✓	✓	
1			
	0	1	

b	0	1	2
0	✓	✓	
1			
	0	1	

c	0	1	2
0			
1			

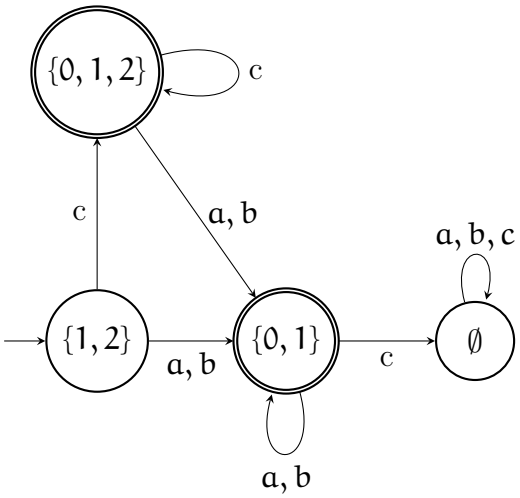
От множеството $\{0, 1\}$ и буквата c получихме преход към празното множество, което играе ролята на error състояние. Веднъж попаднали в него продължавайки да четем думата оставаме в него. Също така не получихме множество от състояния, което да не сме видяли до момента. Продължаваме с таблици за $\{0, 1, 2\}$.

a	0	1	2
0			
1	✓	✓	
2			
	0	1	

b	0	1	2
0			
1	✓	✓	
2			
	0	1	

c	0	1	2
0			
1			
2	✓	✓	✓
	0	1	2

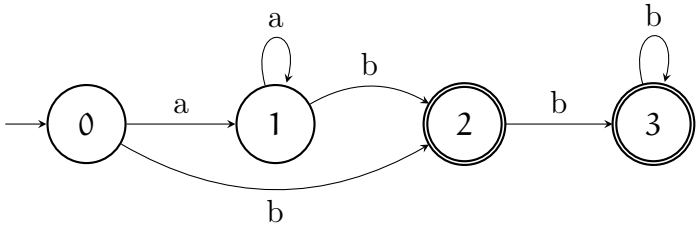
Като навържем резулатите от таблиците, които направихме получаваме тотален КДА.



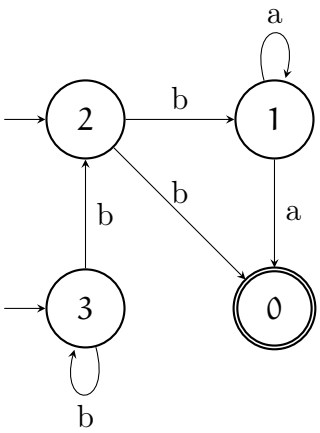
Началното състояние е $\{1, 2\}$. Финални са състоянията $\{0, 1\}$ и $\{0, 1, 2\}$. Общо автомата има 4 състояния. Автомата, който детерминирахме има 3 състояния и $2^3 = 8$. Така в този пример, излезе че значими (достижими) са само половината от подмножествата на $\{0, 1, 2\}$.

4 Пример 2

Нека $L = \{a\}^* \cdot \{b\}^+$. Нека $R = Reverse_{\{a,b\}}(L)$. Очевидно $R = \{b\}^+ \cdot \{a\}^*$. Един КНА за L е следният.



Като приложим конструкцията за автомат за R по автомат за L получаваме следният КНА.



Сега го детерминираме, като започваме да правим таблици по множеството $\{2, 3\}$.

a	0	1	2	3
2				
3				

b	0	1	2	3
2	✓	✓		
3			✓	✓
	0	1	2	3

a	0	1	2	3
0				
1	✓	✓		
2				
3				
	0	1		

b	0	1	2	3
0				
1				
2	✓	✓		
3			✓	✓
	0	1	2	3

a	0	1	2	3
0				
1	✓	✓		
	0	1		

b	0	1	2	3
0				
1				

Така получаваме следният тотален КДА, който разпознава езика R.