УДК 510.676, 519.7

СТРУКТУРА $(\mathbb{Q}, <)$ ЯВЛЯЕТСЯ АВТОМАТНОЙ

Васылькива О.В.
Кафедра информатики

В данной работе приводится способ кодирования дробей словами конечного алфавита, позволяющий с помощью конечного автомата по двум кодам дробей проверять, верно ли, что первая дробь больше второй.

It is proposed a codding of the rational numbers. For the codding, there exists a finite automaton such that for given two codes of rational numbers, the automaton checks whether or not the first number is more than the second one.

Ключевые слова: упорядоченное множество рациональных чисел; автоматные структуры.
Keywords: ordered set of the rational numbers; automatic structures.

Введение. Понятие автоматной структуры предложено в [1]. При изучении автоматности дробей относительно разных отношений, одним из первых возникает следующий вопрос: «Является ли множество дробей с отношением линейного порядка автоматным?»

Мы отвечаем на этот вопрос положительно.

1. Структура $(\mathbb{Q}, <)$ является автоматной. Пусть $\mathbb{Q}$ — это множество рациональных чисел. Чтобы доказать, что $(\mathbb{Q}, <)$ является автоматной, надо:

1) задать конечный алфавит Σ,

2) придумать отображение дробей (элементов из $\mathbb{Q}$) в слова алфавита Σ,

3) построить конечный автомат, который воспринимает те и только те слова, составленные из букв алфавита Σ, которые являются кодами дробей,

4) построить конечный автомат, который воспринимает в смысле [1] пару w_1, w_2 кодов дробей тогда и только тогда, когда первая дробь больше второй.

Алфавит у нас будет $\Sigma = \{\oplus, \ominus, 0, 1, ., u, d\}$. Очевидно, что любой конечный алфавит можно свести к алфавиту $\{0, 1\}$, но это повредило бы наглядности доказательства. Поэтому мы этого делать не будем.

Слово w будет начинаться с буквы $\oplus$ или $\ominus$ в зависимости от знака (считаем, что 0 является положительным числом). Затем будет идти двоичное представление целой части дроби, использующее буквы 0 и 1. Если у дроби есть дробная часть, то после целой части ставится точка, а если дробной части нет, то ничего больше не ставится. После точки идет последовательность из букв d и u, которая будет кодировать дробную часть дроби. Дробную часть дроби будем кодировать следующим образом.

Запишем дроби в следующем порядке:

$$\frac{1}{2}, \frac{1}{3}, \frac{2}{3}, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, \frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}, \frac{1}{7}, \ldots$$

Дробь $\frac{1}{2}$ будет иметь длину 0, дробь $\frac{1}{3}$ будет кодироваться словом d и иметь длину 1, дробь $\frac{2}{3}$ будет кодироваться словом u и иметь длину 1. Эти дроби являются базисными; буква d от слова **down** (меньше), а буква u от слова **up** (больше).

Далее на i-том шаге i-тая дробь получает в качестве кода следующую последовательность букв d и u (если рассматриваемая дробь равна дроби, которая уже была закодирована, то ее будем пропускать — это те дроби, которые можно сократить): сначала располагаем уже закодированные дроби в порядке возрастания и находим две соседние уже закодированные дроби, между которыми рассматриваемая дробь находится, затем копируем более длинный код дроби из кодов этих двух соседних дробей, а затем, если рассматриваемая дробь больше дроби, код которой скопирован, то для получения кода рассматриваемой дроби к скопированному коду добавляем в конце u, а если меньше, то в конце добавляем d.

Проиллюстрируем на примере, как это будет выглядеть при кодировании дроби $\frac{7}{9}$.

7/ 8	uuuuuu	4/ 9	duuu
6/ 7	uuuuu	3/ 7	duu
5/ 6	uuuu	2/ 5	du
4/ 5	*uuu*	3/ 8	dud
7/ 9	*uuu***d**	1/ 3	d
3/ 4	uu	2/ 7	ddu
5/ 7	uud	1/ 4	dd
2/ 3	u	2/ 9	dddu
5/ 8	udu	1/ 5	ddd
3/ 5	ud	1/ 6	dddd
4/ 7	udd	1/ 7	ddddd
5/ 9	uddd	1/ 8	dddddd
1/2		1/ 9	ddddddd

Лемма 1. *Любое слово из букв* $\{d, u\}$ *является кодом некоторой дроби.*

Доказательство. Доказательство индукцией по длине слова. Базис. Нулевой длине соответствует дробь $\frac{1}{2}$, код длины 1 имеют дроби $\frac{1}{3}$ (ее код есть d) и $\frac{2}{3}$ (ее код есть u). Покажем, что для слова w длины n найдется дробь, которая имеет код w. Рассмотрим слово w', которое имеет длину $n-1$ и которое является началом w. По индукционному предположению слово w' является кодом дроби q'. Пусть код дроби q' был построен на j-том шаге. По алгоритму построения, до j-го шага были построены более короткие коды дробей $q_b^{'}$ и $q_s^{'}$. Эти дроби являются соседними, если расположить все дроби, коды которых построены до j-го шага, в порядке возрастания. При этом $q_s^{'} < q' < q_b^{'}$. В силу плотности порядка дробей, на каком-то шаге появится дробь, лежащая между $q_s^{'}$ и q', и код этой дроби будет $w'd$, а также появится дробь, лежащая между q' и $q_b^{'}$, кодом которой будет $w'u$. ∎

Получаем, что множество дробей $\mathbb{Q}$ можно отобразить в слова алфавита $\Sigma = \{\oplus, \ominus, 0, 1, ., u, d\}$ и образ $\mathbb{Q}$ при этом отображении задать регулярным выражением.

$$\text{Дробь 0 имеет код } \alpha_0 = \oplus$$

$$\alpha_1 = (\oplus + \ominus)1(0+1)^* \text{ задает коды дробей без дробной части.}$$

$$\alpha_2 = (\oplus + \ominus).(u+d)^* \text{ задает коды дробей без целой части.}$$

$$\alpha_3 = (\oplus + \ominus)(1(0+1)^*).(u+d)^* \text{ задает коды дробей с целой и дробной частями.}$$

Тем самым получаем, что коды всех дробей задаются конечным автоматом, задающим регулярное выражение

$$\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3,$$

которое есть

$$\oplus + (\ (\lambda + (\oplus + \ominus)1(0+1)^*)(\lambda + (.(u+d)^*))\ )$$

Примеры кодов дробей:

0	$\oplus$
1	$\oplus 1$
-8	$\ominus 1000$
$-\frac{1}{2}$	$\ominus .$
$\frac{1}{3}$	$\oplus .d$
$-2\frac{3}{8}$	$\ominus 10.dud$

Приведем пример кодирования пары дробей. Например,

$$-6\frac{5}{13} = \ominus 110.dudu \ \text{ и } \quad -25\frac{1}{2} = \ominus 11001.$$

$-6\frac{5}{13}$	$=$	$\ominus$	1	1	0	.	d	u	d	u
$-25\frac{1}{2}$	$=$	$\ominus$	1	1	0	0	1	.	#	#

Заметим, что в конце слова, если оно заканчивается раньше, чем другое слово, записываются диезы. Код пары этих двух дробей будет подаваться автомату, задающему отношение порядка, в таком виде:

$\ominus$	$\ominus$	1	1	1	1	0	0	.	0	d	1	u	.	d	#	u	#

Конечный автомат, задающий отношение порядка (проверяющий, является ли первая дробь больше второй) будет действовать следующим образом. Если знаки разные, то все очевидно. Автоматность порядка на целых числах в бинарной записи известна, и остается случай, когда знак и целая часть у двух дробей равны. В этом случае идем по слову, пока буквы у кодов дробей совпадают. Как только встретим пару разных букв, то тот код, что имеет d, меньше, а тот, что имеет u, больше. Приведем пример автомата, который это делает.

	$\oplus$	$\ominus$	0	1	$.$	d	u	$\#$
a	$a^{\oplus}$	$a^{\ominus}$						
$a^{\oplus}$	$b_{\oplus}$	$true$						
$b_{\oplus}$			$b^0_{\oplus}$	$b^1_{\oplus}$	$c^{\cdot}_{\oplus}$			$false$
$b^0_{\oplus}$			$b_{\oplus}$	$false$	$true$			$true$
$b^1_{\oplus}$			$true$	$b_{\oplus}$	$true$			$false$
$c^{\cdot}_{\oplus}$			$false$	$false$	$c_{\oplus.}$			$false$
$c_{\oplus.}$						$c^d_{\oplus.}$	$c^u_{\oplus.}$	$false$
$c^d_{\oplus.}$						$c_{\oplus.}$	$false$	$true$
$c^u_{\oplus.}$						$true$	$c_{\oplus.}$	$false$
$a^{\ominus}$	$false$	$b_{\ominus}$						
$b_{\ominus}$			$b^0_{\ominus}$	$b^1_{\ominus}$	$c^{\cdot}_{\ominus}$			$true$
$b^0_{\ominus}$			$b_{\ominus}$	$true$	$false$			$false$
$b^1_{\ominus}$			$false$	$b_{\ominus}$	$false$			$false$
$c^{\cdot}_{\ominus}$			$true$	$true$	$c_{\ominus.}$			$true$
$c_{\ominus.}$						$c^d_{\ominus.}$	$c^u_{\ominus.}$	$true$
$c^d_{\ominus.}$						$c_{\ominus.}$	$true$	$false$
$c^u_{\ominus.}$						$false$	$c_{\ominus.}$	$true$

Список литературы

[1] Blumensath A., Graedel E. Automatic structures. // Proc. 15th IEEE Symp. on Logic in Computer Science, 2000, pp. 51–62.