Viktor Alad'ev

Операции над языками, генерируемыми τ_n-грамматиками

Commentationes Mathematicae Universitatis Carolinae, Vol. 15 (1974), No. 2, 211--220

Persistent URL: http://dml.cz/dmlcz/105547

Commentationes Mathematicae Universitatis Carolinae

15,2 (1974)

ОПЕРАЦИИ НАД ЯЗЫКАМИ, ГЕНЕРИРУЕМЫМИ τ_m-ГРАММАТИКАМИ

Виктор АЛАДЬЕВ, Таллин

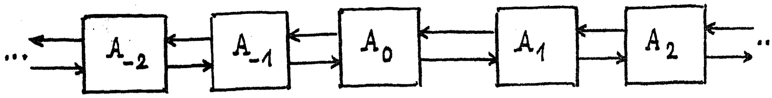
Резюме: В настоящей статье показывается, что языки, генерируемые τ_m-грамматиками, незамкнуты относительно основных операций за исключением операции обращения.

Ключевые слова: Однородная структура; слово, генерируемое однородной структурой; τ_m-грамматика; язык, генерируемый τ_m-грамматикой; L -системы.

AMS: 94A20, 68A30, 92A05 Ref. Ž. 8.764.11

Теория однородных структур представляет большой интерес для вычислительной техники, математической биологии, а также как отдельная ветвь абстрактных автоматов [1 - 3] . Впоследние годы наибольшее число работ (в том числе и настоящая) посвящается структурным и поведенческим аспектам теории. В работе [4] на основе одномерных однородных структур мы определили τ_m-грамматики и изучили ряд свойств языков, генерируемых такими грамматиками. τ_m-грамматики подобны грамматикам, введеным Линденмайером [5] (так называемые L -системы). Оба типа грамматик были впервые введены с целью изучения и описания формальных правил функционирования развивающихся биологических систем. Однако эти грамматики представляют определенный интерес и для теории формальных языков. Такие грамматики обладают тем свойством, что на каждом шаге в выводе перерабатывается каждый символ слова, с которым грамматика оперирует. Таким образом, τ_m-грамматики и L -системы подобны другим грамматикам с параллельными правилами замещения [6 - 8] , но параллелизм в первых выше, так как в τ_m-грамматиках и L-системах каждый символ слова заменяется на каждом шаге вывода. Более того, τ_m-грамматики на формальном уровне описывают функционирование однородных структур, на которых реализуются вычислительные и логические устройства, а сами структуры весьма удобны в качестве технической базы для организации параллельной обработки информации [2]. Известно [4], что класс всех языков, генерируемых τ_m-грамматиками, является собственным подклассом класса всех языков, порождаемых L -системами. Такое сужение связано с тем, что правила замещения в L -системах позволяют делать вставки любых подслов в любые места перерабатываемых системой слов, тогда как τ_m-грамматики допускают это только на концах слов. Более того, τ_m-грамматика является алгоритмом, тогда как L -системы есть, вообще говоря, исчисления. Известно [9], что языки, порождаемые L -системами, незамкнуты относительно всех основных операций: объединения, пересечения, дополнения, обращения, произведения, итерации, гомоморфизма и конечного преобразования. Возникает интересный вопрос сравнения степени влияния такого сужения τ_m-грамматик относительно L -систем на замкнутость языков, генерируемых τ_m-грамматиками, относительно основных операций. Для дальнейшего нам понадобится ряд понятий и определений.

Классическое понятие одномерной однородной структуры (ОС) состоит в следующем. Имеется связанное множество идентичных конечных автоматов Мура, помещенных в точки пространства Z^1 (рис).



Рисунок

Связь автоматов (не нарушая общности) определяется индексом соседства $\vartheta = \{0, 1, \ldots, m-1\}$ единичного A_i -автомата структуры. Индекс соседства одинаков для всех автоматов ОС и определяет шаблон соседства (набор соседних A_i -автомату автоматов) ШС $= \{i+0, i+1, \ldots, i+m-1\}$, т.е. соседними A_i-автомату служат A_j -автоматы $(j = i + k;\ k = \overline{0, m-1})$. Каждый A_i -автомат может находиться в любой дискретный момент времени $T \geq 0$ в состоянии из множества $S^p = \{0, 1, \ldots, p-1\}$. Работа любого A_i-автомата в ОС определяется локальной функцией перехода $L^{\vartheta}_{(m)}$, которая определяет состояние A_i -автомата в момент $T+1$ по состояниям всех его соседей в момент времени T . Функция $L^{\vartheta}_{(m)}$ полностью определяется соотношениями

$$\langle X_1 X_2 \ldots X_m \rangle \longrightarrow X'_1$$

$$(1) \qquad \langle 0\ 0\ \ldots\ 0 \rangle \longrightarrow 0$$

$$X_i,\ X'_1 \in S^p \qquad\qquad (i = \overline{1, m}) .$$

Отображение $\theta : Z^1 \longrightarrow S^p$ будем называть словом (конфигурацией) ОС. Слово, имеющее только конечное множество символов из $S^p \setminus \{0\}$ будем называть конечным и множество всех таких слов обозначать через $\overline{C}_p$. Мы будем рассматривать только слова из множества $\overline{C}_p$. Слово ОС в момент $T = 0$ будем обозначать через c_0 . Одновременное применение соотношений (1) определяет глобальную функцию τ , т.е. для данного множества S^{n}_{ϑ}, индекса соседства ϑ и слов $c_1, c_2 \in \overline{C}_n$ локальная функция $L^{\vartheta}_{(m)}$ определяет глобальное преобразование $\tau: c_1 \longrightarrow c_2$. Нетрудно убедиться, что для $(\forall c_1 \in \overline{C}_n)(c_1\tau \in \overline{C}_n)$. Таким образом, функционирование ОС состоит в переработке преобразованием τ конечных слов ОС. При сделанных предположениях ОС <u>генерирует</u> последовательность слов $\langle c_0\rangle = c_0, c_1, c_2, \dots, c_i, \dots$ $(c_i = c_0\, \tau^i;\ i = 1, 2, \dots)$. Тогда τ_m <u>-грамматикой</u> будем называть упорядоченную тройку $\tau_m = (\overline{C}_n, c_0, L^{\vartheta}_{(m)})$, где:

$\overline{C}_n$ – терминальный словарь,

$c_0 \in \overline{C}_n$ – начальное слово или аксиома,

$L^{\vartheta}_{(m)}$ – правила вывода.

А множество $\langle c_0\rangle$ генерируемое такой грамматикой, будем называть $L(\tau_m)$ <u>-языком.</u> От общеизвестных формальных грамматик τ_m-грамматики, также как и L-системы, отличаются двумя основными моментами:

а) отсутствием нетерминального словаря,

б) одновременным применением правил вывода (1).

Будем говорить, что множество слов $G \subset \overline{C}_n$ является языком $L(\tau_m)$, если существует τ_m-грамматика, генерирующая множество G. В работе [4] мы доказали, что множество $\overline{C}_n$ не может быть $L(\tau_m)$-языком.

Для $L(\tau_m)$-языков обычным образом определяются все основные операции. Только операция дополнения определяется относительно множества $\overline{C}_n$. Если мы обозначим через L^* класс языков, порождаемых L-системами, а через $L(\tau_m)$ класс языков, генерируемых τ_m-грамматиками, то свойства языков по отношению к основным операциям можно свести в следующую таблицу.

Таблица

п/п	Операции	Тип языка $L(\tau_m)$	Тип языка L^*
1	объединение	0	0
2	пересечение	0	0
3	дополнение	0	0
4	обращение	1	0
5	произведение	0	0
6	итерация	0	0
7	гомоморфизм	0	0
8	конечное преобразование	0	0

В этой таблице единица обозначает замкнутость относительно соответствующей операции класса языков определенного типа, нуль - незамкнутость. Из анализа приведенной таблицы, учитывая, что языки без нетерминальных символов оказывают чрезвычайно большое сопротивление замкнутости относительно основных операций над языками [10], можно сделать вывод, что при прочих равных условиях отсутствие возможности вставки подслов в любом месте генерируемого слова в τ_m-грамматиках исключает семейство $L(\tau_m)$-языков из семейства языков анти-AFL.

Результаты по незамкнутости относительно основных операций L^*-языков можно найти в работе [9]. В работе [4] мы доказали незамкнутость операций 1, 2, 5 - 8 и замкнутость операции 4 (таблица). Здесь мы решим вопрос и относительно операции дополнения.

__Теорема__. Класс $L(\tau_m)$-языков незамкнут относительно дополнения.

__Доказательство__. Выше уже говорилось, что в случае τ_m-

граммат..к естественно рассматривать дополнение $L(\tau_m)$-языка относительно множества $\overline{C}_n$. Исходя из формулировки теоремы достаточно найти пример языка $L(\tau_m)$, дополнение которого не является $L(\tau_m)$-языком. В качестве примера выбираем язык $L_1 = \{1^m \mid m \geq 1\}$. Нетрудно убедиться, что язык L_1 генерируется уже грамматикой $\tau_2 = (\overline{C}_2, 1, L^{\vartheta}_{(2)})$ с функцией $L^{\vartheta}_{(2)}$ определяемой следующими соотношениями

$$00 \longrightarrow 0 \qquad 10 \longrightarrow 1 \qquad 01 \longrightarrow 1 \qquad 11 \longrightarrow 1 \; .$$

Тогда дополнением $L(\tau_m)$-языка L_1 является множество $L_2 = \overline{C}_2 / L_1$. Но так как в определении τ_m-грамматик отсутствует нетерминальный словарь, то если L_2 есть $L(\tau_m)$-язык, то он обязательно должен генерироваться уже некоторой грамматикой $\tau_m = (\overline{C}_2, c_0, L^{\vartheta}_{(m)})$, где $c_0 \in \overline{C}_2$. Используем метод доказательства от противного. Предположим, что существует грамматика $\tau_m = (\overline{C}_2, c_0, L^{\vartheta}_{(m)})$ генерирующая язык L_2. Значит, τ_m-грамматика генерирует последовательность слов $\langle c_0 \rangle = c_0 \longrightarrow c_1 \longrightarrow c_2 \longrightarrow \ldots \longrightarrow c_i \longrightarrow \ldots$ такую, что $\bigcup_i \{c_i\} = L_2$. Исследуем теперь, как реагирует ОС, соответствующая такой τ_m-грамматике, на слова из множества L_1. Из определения L_2 следует, что $L_1 \cap L_2 = \emptyset$ и язык L_2 бесконечен.

Возьмем слово $\alpha = 1^m \in L_1$ $(m \geq n)$ такое, что $\alpha\tau \in L_1$. Очевидно, что такое слово должно существовать, ибо в противном случае либо $(\forall\alpha \in L_1)(\alpha\tau = c_0)$, либо $(\forall\alpha \in L_1)(\exists i \neq 0)(\alpha\tau = c_i)$ при $\ell(\alpha) \geq n$ и $c_0, c_i \in L_2$, где $\ell(\alpha)$ - длина слова α. Но в обоих случаях, как можно показать из определения языка L_2, существуют стираемые (а значит и неконструктируемые) конфигурации [1], что влечет за собой невозможность генерации τ_m-грамматикой языка L_2 а это противоречит условию. Как может себя вести слово $\alpha = 1^m$

$(m \geq n)$ под действием глобальной функции перехода τ ?
Прежде всего следует отметить, что соответствующая функции τ функция $L^{\vartheta}_{(n)}$ должна содержать соотношение $\langle \underbrace{11\ldots1}_{n} \rangle \longrightarrow 1$. Действительно, в противном случае мы получаем либо $\alpha\tau \notin L_1$ при $\ell(\alpha) \geq n$, что противоречит условию, либо $\alpha\tau = 0$. Но тогда слово $c_i = \alpha 0^n \alpha \in L_2$ будет таково, что $c_i\tau = 0$, а это невозможно из-за бесконечности языка L_2 и характера функционирования ОС.

Таким образом, функция $L^{\vartheta}_{(n)}$ должна содержать соотношения

$$
\begin{array}{lll}
 & \langle \underbrace{11\ldots1}_{k}\ \underbrace{00\ldots0}_{n-k} \rangle \longrightarrow 0 & \\
 & \langle \underbrace{11\ldots1}_{k'}\ \underbrace{00\ldots0}_{n-k'} \rangle \longrightarrow 0 & k' < k \\
(2) & \langle \underbrace{00\ldots0}_{u}\ \underbrace{11\ldots1}_{n-u} \rangle \longrightarrow 1 & u' < u \\
 & \langle \underbrace{00\ldots0}_{u'}\ \underbrace{11\ldots1}_{n-u'} \rangle \longrightarrow 1 & t = u - k\ .
\end{array}
$$

Согласно величине t могут быть следующие три возможности:

1) $t > 0, \quad \ell(\alpha) < \ell(\alpha\tau)$
2) $t = 0, \quad \ell(\alpha) = \ell(\alpha\tau)$
3) $t < 0, \quad \ell(\alpha) > \ell(\alpha\tau)$.

В первом случае в ОС существует конечное множество универсальных конфигураций (УКФ) [1]. Действительно, таким множеством являются уже конфигурации (слова):

$$c_0,\ 1^{m_1},\ 1^{m_2}, \ldots,\ 1^{m_{t-1}},\ 1^{k} \qquad (k = \overline{1, n-1})$$

$$m_1 = n; \quad m_i = m_1 + (i-1), \qquad (i = \overline{1, t})$$

так как при сделанных предположениях имеет место соотношение

$$\overline{C}_{p} = \{\bigcup_{i} \langle 1^{m_i} \rangle\} \cup \{\bigcup_{k} \langle 1^{k} \rangle\} \cup \langle C_0 \rangle\ .$$

Значит, множество УКФ состоит из $n + t$ слов. Существование же конечного множества УКФ в ОС противоречит теореме 1.10 [1].

Значит, для слов $\alpha = 1^m$ $(m \geq n)$ не могут существовать правила вывода (2) при условии $t > 0$. Не могут быть правилами вывода и соотношения (2) при условии $t = 0$. Действительно, в противном случае в ОС существуют слова $\alpha = 1^m$ $(m \geq n)$ такие, что $\alpha\tau = \alpha$. Но тогда для слов $c_i = 1^m 0^n 1^m \in L_2$ также имеет место соотношение $c_i\tau = c_i$, что противоречит бесконечности языка L_2. Остается случай при $t < 0$. В этом случае на каждом шаге после применения функции τ к слову $\alpha = 1^m$ $(m \geq n)$ происходит уменьшение его длины на t единицы. Следовательно, в результате такого применения функции τ к словам $\alpha = 1^m$ $(m \geq n)$ мы получим <u>ядро</u>, состоящее из слов $\beta = 1^k$ $(k = \overline{1, n-2})$, т.е. в ядро входят те слова, для которых нам не известен результат применения функции τ, так как мы опираемся только на правила вывода (2) при условии $t < 0$. Тогда для $n \geq 4$ ядро состоит из двух слов α_1 и α_2. При применении же к ним функции τ могут быть следующие возможности:

1) $\alpha_1\tau = \alpha_1$ или $\alpha_2\tau = \alpha_2$,

2) $\alpha_1\tau = \alpha_2\tau$,

3) $\alpha_1\tau = c_i$ или $\alpha_2\tau = c_i$ $(i \neq 0)$.

В первом случае существует слово $c_k = \alpha_1 0^n \alpha_1$ или $c_k = \alpha_2 0^n \alpha_2$ такое, что $c_k\tau = c_k$ и бесконечный язык L_2 не может быть генерируем τ_n-грамматикой. В остальных двух случаях в ОС будут существовать стираемые (а значит и неконструируемые) конфигурации [1] и τ_n-грамматика также не сможет генерировать язык L_2, что противоречит условию. Нам осталось рассмотреть лишь случай, когда $n = 3$ и ядро содержит только одно слово. В этом случае, учитывая все вышесказанное, функция $L^{\vartheta}_{(3)}$ будет определяться следующими соотношениями:

$$(3) \qquad \begin{array}{lll} 000 \longrightarrow 0 & 010 \longrightarrow X_1 & 100 \longrightarrow 0 \\ 001 \longrightarrow 0 & 011 \longrightarrow X_2 & 101 \longrightarrow X_3 \\ 110 \longrightarrow X_4 & 111 \longrightarrow 1\ , & X_i \in S^2 \quad (i = \overline{1,4}) \end{array}$$

ибо функция τ уменьшает длину любого слова $\alpha = 1^m$ $(m \geq 3)$, а это может происходить только при одном из следующих трех условий:

1) $\qquad \mu = 1 \qquad k = 2$,

2) $\qquad \mu = 0 \qquad k = 1$,

3) $\qquad \mu = 0 \qquad k = 2$.

Но из соотношений (3): $000 \longrightarrow 0$, $001 \longrightarrow 0$ и $100 \longrightarrow 0$ следует, что либо в ОС существуют конфигурации, если $010 \longrightarrow 0$, либо существует слово $\alpha = 1$ такое, что $\alpha\tau = \alpha$. Обе возможности говорят о том, что язык L_2 не может порождаться τ_3-грамматикой. А так как τ_m-грамматика, генерирующая язык L_2 , и соответствующая ей ОС должны перерабатывать и слова из множества L_1 , то из вышеприведенного анализа мы приходим к наличию противоречия с тем, что существует τ_m-грамматика, генерирующая язык L_2 . Таким образом, множество слов $L_2 = \overline{C}_2 \setminus L_1$ не может быть $L(\tau_m)$-языком. Теорема полностью доказана.

В заключение настоящей работы нам хотелось бы сформулировать следующую <u>гипотезу</u>: Дополнение $L(\tau_m)$-языка не может быть снова $L(\tau_m)$-языком. Многочисленные попытки опровергнуть эту гипотезу так же как и доказать ее не дали пока никакого результата, хотя результаты исследований поведенческих свойств однородных структур склоняют все же в сторону истинности гипотезы.

Литература

[1] В. АЛАДЬЕВ: К теории однородных структур, Таллин,1972.

[2] Э. ЕВРЕИНОВ, Ю. КОСАРЕВ: Однородные универсальные вычислительные системы высокой производительности, Новосибирск,1966.

[3] V. ALADYEV: Survey of research in the theory of homogeneous structures and their applications, Mathematical Biosciences 9 (to appear).

[4] В. АЛАДЬЕВ: τ_m-грамматики и порождаемые ими языки, Изв.АН ЭССР,23(1974).

[5] A. LINDENMAYER: Developmental systems without cellular interaction.Their languages and grammars, J. Theoret.Biol.30(1971),455-484.

[6] S. GREIBACH and J. HOPCROFT: Scattered context grammars, J.Comput.Syst.Sci.3(1969),323-347.

[7] B. NASH and R. COHEN: Parallel levelled grammars, Tenth IEEE Conf.Swit.Aut.Theory (1969),263-276.

[8] V. RAJLICH: Absolutely parallel grammars and two-way deterministic finite state transducers, Proc. Third ACM Symp.Theory Computing (1971),132-137.

[9] A. LINDENMAYER and G. ROZENBERG: Developmental systems and languages, Proc.Fourth ACM Symp.Theory Computing (1972),214-221.

[10] A. SALOMAA: On sentential forms of context-free grammars, Acta Informatica 2(1973),40-49.

Эстонский филиал ВГПТИ ЦСУ СССР

Таллин

Маакри 15

СССР

(Oblatum 29.1.1974)