EESTI NSV TEADUSTE AKADEEMIA TOIMETISED. 23. KÖIDE
BIOLOOGIA. 1974, NR. 1

ИЗВЕСТИЯ АКАДЕМИИ НАУК ЭСТОНСКОЙ ССР. ТОМ 23
БИОЛОГИЯ. 1974, № 1

https://doi.org/10.3176/biol.1974.1.05

УДК 519.95; 51:801

ВИКТОР АЛАДЬЕВ

τ_n-ГРАММАТИКИ И ПОРОЖДАЕМЫЕ ИМИ ЯЗЫКИ

Кроме обычных порождающих грамматик (грамматик Хомского), имеются и другие способы задания языков, понимаемых как множества слов; некоторые из таких способов представляют собой исчисления с правилами иного типа, чем у грамматик Хомского (например, категориальные грамматики), другие даже не являются исчислениями (окрестностные грамматики). Кроме того, многие грамматические схемы позволяют не только задавать слова, но и вводить на них некоторые структуры (в частности, НС-грамматики позволяют вводить системы составляющих, грамматики зависимостей — деревья подчинения) или, скажем, связывать с описываемыми множествами слов некоторые распределения вероятностей. В данной статье мы вводим новый тип порождающих грамматик (τ_n-грамматики) и изучаем некоторые свойства порождаемых ими языков $L(\tau_n)$.

1. Вводимый нами новый тип формальных грамматик (τ_n-грамматики) является естественной составной частью математической теории однородных структур (ОС) [1]. Их изучение тесно связано с поведением конфигураций в ОС и моделированием развивающихся биологических систем. Однородные структуры и ряд аспектов поведения конфигураций в ОС допускают естественные биологические интерпретации [1–5]. Более того, в последние годы бурно развивается класс игр, имитирующих процессы, происходящие в живой природе. Большинство игр этого класса моделируется ОС. В качестве примера можно привести игру Конуэя «Жизнь» (Г а р д н е р М., Математические досуги, М., 1972, с. 458—488), 1-моделируемую уже бинарной двумерной ОС с индексом соседства Мура. Грубо говоря, любая ОС есть вариант игры «Жизнь» со своими генетическими законами, т. е. локальной функцией перехода. Поэтому и все результаты по теории ОС могут быть использованы также в играх этого класса. Мы будем рассматривать пока только одномерные ОС, хотя те же вопросы возникают и в случае высших размерностей. Предполагается, что читатель знаком с основными определениями и понятиями теории ОС [1], которые здесь будут использоваться. Основные же понятия теории формальных языков соответствуют работам [6–7].

Приведем те определения из теории формальных языков, которые будем использовать наиболее часто.

О п р е д е л е н и е 1. Порождающий грамматикой (или просто г р а м м а т и к о й) называется упорядоченная четверка $G=(V, \Sigma, \Pi, \sigma)$, где

1) V — конечный алфавит,
2) $\Sigma \subseteq V$ — конечный алфавит,

3) Π — конечное множество упорядоченных пар вида (u, v), где $u \in (V - \Sigma)^* - \{\varepsilon\}$ и $v \in V^*$; A^* есть множество всех конечных слов в алфавите A и ε — пустое слово,
4) $\sigma \in V - \Sigma$.

Элементы множества $V - \Sigma$ называются вспомогательными, а элементы множества Σ — основными символами. Σ и $V - \Sigma$ называются соответственно основным и вспомогательным алфавитом. Элементы (u, v) множества Π называются правилами (или правилами подстановки) и записываются обычно в виде $u \to v$. Элемент σ называется начальным символом или аксиомой.

Определение 2. Пусть $G = (V, \Sigma, \Pi, \sigma)$ — порождающая грамматика. Тогда множество слов $L(G) = \{x \in \Sigma^* | \sigma \overset{\Pi}{\to} x\}$ называется языком, порождаемым грамматикой G.

Определение 3. Порождающая грамматика $G = (V, \Sigma, \Pi, \sigma)$ называется контекстно-свободной (КС-грамматикой), если каждое ее правило имеет вид $\xi \to v$, где $\xi \in V - \Sigma$ и $v \in V^*$. Язык $L \subseteq \Sigma^*$ называется КС-языком, если он порождается некоторой КС-грамматикой.

Определение 4. Порождающая грамматика $G = (V, \Sigma, \Pi, \sigma)$ называется грамматикой непосредственно составляющих (НС-грамматикой), если каждое ее правило имеет вид $u\xi v \to uyv$, где $\xi \in V - \Sigma$; $u, v \in (V - \Sigma)^*$ и $y \in V^* - \{\varepsilon\}$. Язык, порождаемый НС-грамматикой, называется НС-языком.

Перейдем теперь к определению τ_n-грамматик, предварительно обсудив их на содержательном уровне.

Менее формально τ_n-грамматики определяются следующим образом. Имеется одномерная бесконечная цепочка идентичных конечных автоматов, каждый из которых может иметь состояние из множества $S^p = \{0, 1, \ldots, p-1\}$. Каждый автомат в цепочке в момент T получает информацию о состояниях своих непосредственных соседей, которые определяются согласно индексу соседства (v). Шаблон соседства будем полагать связным и имеющим размер n [1]. Следует сразу же отметить, что рассматривая связный шаблон соседства, мы нисколько не сужаем класс ОС и представимых ими языков, так как несвязный шаблон соседства есть просто частный случай связного, у которого по крайней мере информация одного автомата не влияет на определение локальной функции перехода.

Состояние автомата в момент $T+1$ есть локальная функция состояний всех его соседей в момент T. Применяя ко всем автоматам цепочки одновременно эту функцию, мы определяем глобальную функцию перехода τ. Словом будем называть конечную конфигурацию состояний цепочки, т. е. такую конфигурацию, которая имеет только конечное число автоматов с состояниями, отличными от 0. Множество всех таких слов обозначим через $\bar{C}_p$. Очевидно, что $\bar{C}_p$ замкнуто относительно операции τ, так как имеется состояние покоя — 0, которое заключается в том, что если все соседи автомата в момент T имели состояние 0, то и в момент $T+1$ этот автомат перейдет в состояние 0. Таким образом, локальная функция перехода $L^v_{(n)}$ определена на всех упорядоченных наборах $\langle x_1, \ldots, x_n \rangle$ $(x_i \in S^p;\ i = \overline{1, n})$ так, что имеют место следующие соотношения:

$$\begin{cases} x_1 x_2 \ldots x_n \to x_1', \\ \underbrace{00 \ldots 0}_{n} \to 0, \end{cases} \qquad x_i, x_1' \in S^p. \tag{1}$$

Главное отличие τ-алгоритма [5] переработки слов от известных алгоритмов (за исключением L-систем [3]) состоит в том, что все подстановки (1) в ОС осуществляются одновременно. Это свойство плюс то, что все допустимые подстановки применяются сразу на каждом шаге, говорит о том, что мы имеем дело с порождающей грамматикой именно как алгоритмом порождения слов, а не исчислением, как имеет место в случае других хорошо известных грамматик. Именно это обстоятельство снимает ряд вопросов (например, проблему неоднозначности вывода), играющих важную роль в других грамматиках, представляющих собой исчисления. Это с особой силой относится к искусственным языкам программирования.

Таким образом, можно определить язык как множество всех слов, порождаемых из начального слова $c_0 \in \overline{C}_p$ посредством описанной выше процедуры. Второе отличие такого языка от известных формальных языков, следовательно, состоит в том, что в нем не будут различаться вспомогательные и основные символы. Вообще и в случае ОС (как порождающей грамматики) можно различать вспомогательные и основные символы, но в смысле, отличном от общепринятого. В этом случае порождающая мощность таких грамматик, вообще говоря, возрастает.

Приведем теперь формальные определения.

Определение 5. τ_n-грамматикой называется упорядоченная четверка $\tau_n = (n, S^p, \tau, c_0)$, где

1) n — размер связного шаблона соседства ОС,
2) S^p — конечное множество внутренних состояний автомата в ОС,
3) τ — глобальная функция перехода в ОС,
4) c_0 — начальное слово, или аксиома ($c_0 \in \overline{C}_p$).

Определение 6. Пусть имеется τ_n-грамматика. Тогда множество $L(\tau_n) = \{c \in \overline{C}_p \mid c_0 \xrightarrow{\tau} c\}$ называется ызком, порождаемым τ_n-грамматикой. Каждое слово языка $L(\tau_n)$ является выводимым из аксиомы c_0.

Таким образом, язык $L(\tau_n) \subseteq \overline{C}_p$, ибо τ_n-грамматика выделяет именно те слова из множества $\overline{C}_p$, которые являются выводимыми из $c_0 \in \overline{C}_p$. Аксиомой может быть любое слово $c_0 \in \overline{C}_p$. Из теоремы I.10 [1] следует, что для любого языка имеет место строгое включение $L(\tau_n) \subset \overline{C}_p$. Будем говорить, что язык L относится к типу $L(\tau_n)$, если существует хотя бы одна τ_n-грамматика ($n \geqslant 1$), порождающая данный язык. Случай τ_1-грамматик мы опускаем, так как они порождают только конечные $L(\tau_1)$-языки.

2. Перейдем теперь к изучению некоторых свойств языков $L(\tau_n)$.

Теорема 1. *Любое конечное подмножество слов из $\overline{C}_p$ является языком $L(\tau_n)$. Существуют регулярные языки, которые не являются языками $L(\tau_n)$.*

Доказательство. Докажем первую часть теоремы. Пусть мы имеем конечное произвольное множество слов $\Omega \subset \overline{C}_p : \sigma_1, \sigma_2, \ldots, \sigma_m$, для которого имеют место соотношения: $(\forall i)(\forall j)(\sigma_i \neq \sigma_j)$ и $n^* = \max\limits_i |\sigma_i|$ $(i = \overline{1, m})$, где $|a|$ обозначает длину слова a. Определим τ_n-грамматику, порождающую язык Ω, следующим образом. Полагаем $n = 2n^*$, а функцию τ определяем таким образом, что $c_i\tau = c_{i+1}$ $(c_i \in \Omega;$ $i = \overline{1, m-1})$. Это можно сделать, так как шаблон соседства ОС равен $2n^*$ и может содержать полную информацию о любом слове $v_i \in \Omega$ $(i = \overline{1, m})$. Поэтому согласно нижеследующей схеме (рис. 1) такой

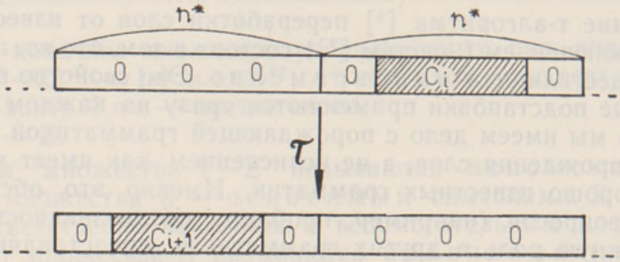

Рис. 1.

τ_n-грамматикой можно порождать не только любое конечное подмножество слов из $\overline{C}_p$, но даже сохранять произвольный порядок следования слов в порождаемой цепочке слов. Следует отметить, что существуют конечные множества слов, которые не являются даже OL-языками [4]. Этим первая часть теоремы доказана.

Рассмотрим теперь язык $L=\{10^m1 \mid m \geqslant 1\}$, где через a^m впредь будем обозначать сокращенную запись слова $\underbrace{a\ldots a}_{m}$. Попробуем построить праволинейную КС-грамматику, порождающую этот язык. В качестве такой грамматики берем грамматику $G=(V, \{0, 1\}, \Pi, \sigma)$, где

$$\Pi=\{\sigma\to 1\gamma,\ \gamma\to 0\mu,\ \mu\to 0\mu,\ \mu\to 1\}.$$

Очевидно, что КС-грамматика G порождает язык $L=\{10^m1 \mid m\geqslant 1\}$ и является праволинейной. А так как множество слов является регулярным тогда и только тогда, когда оно порождается некоторой праволинейной КС-грамматикой [7], то язык L является регулярным. С другой стороны, язык L не может быть языком $L(\tau_n)$ [1]. Что и завершает доказательство нашей теоремы.

Любое конечное множество слов $SV\subset\overline{C}_p$ согласно теореме 1 является $L(\tau_n)$-языком. Это же относится к любому (относительно n) множеству вида $\bigcup_i\{c_i\}\cup SV$, где $c_i\in\overline{C}_p\,(i=\overline{1, n})$. Однако множество $\lim\limits_{i\to\infty}\bigcup_i\{c_i\}\cup SV$ не обязательно является $L(\tau_n)$-языком. В качестве примера можно рассмотреть случай при $SV=\{1\}$ и $c_i=10^i1$ $(i=\overline{1, n})$.

Из следствия 2.1 [1] нетрудно получить следующую теорему.

Теорема 2. *В общем случае язык $L(\tau_n)$ является нерекурсивным множеством слов.*

С другой стороны известно [6–7], что любой КС-язык является рекурсивным множеством слов. Таким образом, существуют языки $L(\tau_n)$, не являющиеся КС-языками, и наоборот. Любой КС-язык является НС-языком, обратное, вообще говоря, неверно [7]. Например, язык $L=\{1^m2^m1^m \mid m\geqslant 1\}$ является НС-языком и языком $L(\tau_n)$, но не является КС-языком.

Среди языков $L(\tau_n)$ естественно вводится иерархия сложности в смысле множества представимых языков. Из свойств моделирования одной ОС другой [1] получаем, что для любого языка $L(\tau_n)$ существует τ_{n+1}-грамматика, порождающая язык $L(\tau_n)$. С другой стороны, не каждый язык $L(\tau_{n+1})$ может порождаться τ_n-грамматикой. Рассмотрим в качестве примера уже двухэлементное множество слов $A=\{10^{n-2}1,\ 10^{n-1}1\}$. Нетрудно убедиться, что множество A является языком $L(\tau_{n+1})$, но оно не может быть языком $L(\tau_n)$.

Существуют также бесконечные регулярные языки, которые порождаются τ_{n+1}-грамматиками и не порождаются никакой τ_n-грамматикой. В качестве примера можно рассмотреть язык $L=\{a^{nm}|m\geqslant 1; a\in S^p\setminus 0\}$. Множество слов L имеет вид $\{a^j|j\in B\}$, где B есть периодическое множество неотрицательных целых чисел. Поэтому L является регулярным множеством (теорема 2.I.2 [7]). Пусть теперь существует τ_n-грамматика, порождающая язык $L=\{a^{nm}|m\geqslant 1\}$. Но тогда для любого $m\geqslant 1$ должно выполняться соотношение $(E!t)\,(c_0\tau^t=a^{nm})$. Из самого вида языка L тогда следует, что функция τ должна удовлетворять соотношениям

$$\begin{cases} \underbrace{00\ldots 0}_{i}\underbrace{aa\ldots a}_{n-i}\to a, \\ \underbrace{aa\ldots a}_{k}\underbrace{00\ldots 0}_{n-k}\to 0, \qquad (0\leqslant d\leqslant n-1), \end{cases}$$

где $d=i-k>1$, ибо в противном случае для любой аксиомы $c_0\in\overline{C}_p$ язык был бы конечным. Более того, должно быть

$$\begin{cases} (\forall k'<k)\,(\underbrace{aa\ldots a}_{k'}\underbrace{00\ldots 0}_{n-k'}\to 0), \\ (\forall i'<i)\,(\underbrace{00\ldots 0}_{i'}\underbrace{aa\ldots a}_{n-i'}\to a), \qquad \underbrace{aa\ldots a}_{n}\to a. \end{cases}$$

Но тогда, если некоторое слово $c_0\tau^{t'}$ имеет вид a^{nm}, то слово $c_0\tau^{t'+1}$ имеет вид a^{nm+d}. А так как при условии $0\leqslant d\leqslant n-1$ имеет место $nm+d\not\equiv(\operatorname{mod} n)$, то слово $c_0\tau^{t'+1}\notin L$, что противоречит условию. С другой стороны, нетрудно задать τ_{n+1}-грамматику, порождающую этот язык. Вышесказанное резюмирует следующая

Теорема 3. $(\forall n\geqslant 2)\ (\#L(\tau_{n-1})\subset\#L(\tau_n))$, *где* $\#L(\tau_n)$ *обозначает множество всех языков* $L(\tau_n)$.

Таким образом, множества языков, порождаемых τ_n-грамматиками, упорядочиваются согласно величине шаблона соседства n и среди них существует бесконечно много классов сложности в указанном выше смысле. Так, например, язык $L=\{1^m2^m|m\geqslant 1\}$ не является языком $L(\tau_2)$ также как и OL-языком [4], но он является уже языком $L(\tau_n)$ для $n\geqslant 3$. Тогда как язык $L=\{1^{2^m}|m\geqslant 0\}$ является OL-языком [4], но не является языком $L(\tau_n)$. А так как τ_n-грамматика может порождать и нерекурсивные языки, то $L(\tau_n)$-язык имеет непустое пересечение с языками Клини, КС- и НС-языками. Можно задать $<\kappa,\ e>$ L-систему [4] при $G=S^p$, $E=\{0^{n-1}\}$, $g=c_0$ и $k+e=n$ так, что она будет порождать язык $L=L(\tau_n)$ для любой наперед заданной τ_n-грамматики. Обратное, как мы видели выше, неверно. Таким образом, $L(\tau_n)$-языки являются собственным подклассом класса языков, порождаемых $<\kappa,\ e>$ L-системами. На рис. 2 показаны пересечения множества $L(\tau_n)$-языков с множествами некоторых известных языков.

Рассмотрим теперь некоторые примеры языков, которые нам пригодятся в дальнейшем.

1. Язык $L=\{1^m23^m|m\geqslant 1\}\cup\{1^m3^m|m\geqslant 1\}$ порождается τ_3-грамматикой. В свою очередь языки $L_1=\{1^m23^m|m\geqslant 1\}$ и $L_2=\{1^m3^m|m\geqslant 1\}$ также являются языками $L(\tau_3)$, но не являются LAL-языками [8].

2. Языки $L_1=\{(1^2)^*23\}$ и $L_2=\{(12)^m|m\geqslant 1\}$ являются языками $L(\tau_3)$.

3. Языки $L_1=\{1^m2^m3^m|m\geqslant 1\}$ и $L_2=\{1^m2^m1^m|m\geqslant 1\}$ являются языками $L(\tau_5)$, но не являются КС-языками.

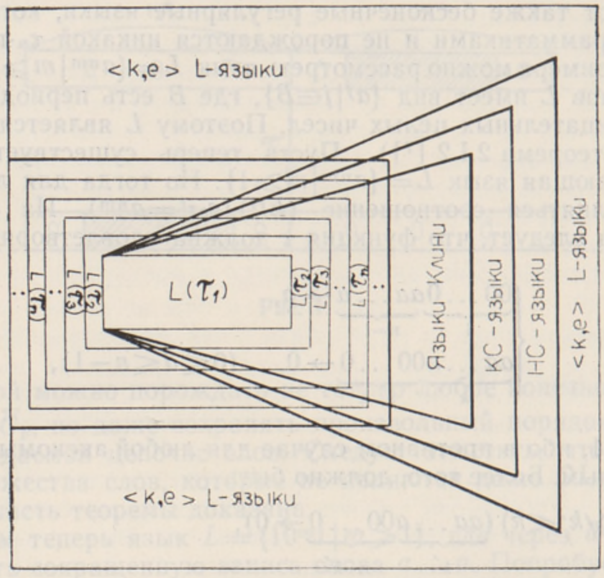

Рис. 2.

4. Определим один подкласс КС-языков — так наз. бесскобочные языки (БКС-языки) [10]. Пусть Σ_1 — некоторый алфавит, $\Sigma_2=\{m_1,\ldots,m_t\}$ и $\Sigma_1\cap\Sigma_2=\varnothing$. Пусть $G=(V,\ \Sigma_1\cup\Sigma_2,\ \Pi,\ \sigma)$, где $V=\Sigma_1\cup\Sigma_2\cup\{\sigma\}$ и

$$\Pi=\{\sigma\to m_i\sigma\sigma\,|\,1\leqslant i\leqslant t\}\cup\{\sigma\to a\,|\,a\in\Sigma_1\}.$$

Можно показать, что БКС-языки не могут быть $L(\tau_n)$-языками, хотя, на наш взгляд, они и могут интерпретироваться с биологической точки зрения как языки, описывающие некоторого рода самоусложняющиеся системы. Простейший пример БКС-языка может быть задан выводом вида

$$\Pi=\{\sigma\to m\sigma\sigma,\ \sigma\to a\}.$$

Порождающая грамматика является определенной и порождает слова языка следующего вида:

$$a$$

$$m\,|\,aa$$

$$m\,|\,\underbrace{ma^2}\underbrace{ma^2}$$

$$m\,|\,\underbrace{m^2a^2\ ma^2}\ \underbrace{m^2a^2\ ma^2}$$

$$m\,|\,\underbrace{m^3a^2\ ma^2\ m^2a^2\ ma^2}\ \underbrace{m^3a^2\ ma^2m^2a^2ma^2}.$$

....

....

Действительно, эта модель в какой-то мере демонстрирует самоусложняющуюся систему.

5. Пусть L есть бесконечный $L(\tau_n)$-язык. Определим язык $L^+=\{xx^R|x\in L\}$, где x^R есть обращение слова x. Для каждого слова из $\overline{C}_p$ $x_1\ldots x_m$ положим $(x_1\ldots x_m)^R=x_m\ldots x_1$. Для каждого множества $H\subseteq\overline{C}_p$ положим $H^R=\{\omega^R|\omega\in H\}$. Очевидно, $x\in\overline{C}_p\leftrightarrow x^R\in\overline{C}_p$. Язык L^+ не всегда является $L(\tau_n)$-языком. В качестве примера рассмотрим язык $L=\{10^m2|m\geqslant 1\}$. Он является $L(\tau_n)$-языком. Язык же $L^+=\{10^m220^m1|m\geqslant 1\}$ не является, очевидно, $L(\tau_n)$-языком.

6. Рассмотрим язык $L=\{a0^ma|m\geqslant 1;\ a\in S^p\diagdown 0\}$. Образуем языки xL и Lx при условиях, что $x\in\overline{C}_p$; $x=x_1\ldots x_d$ и слово $x_2\ldots x_d$ не содержит вхождений слов 0^2a0, $(0a)^2$ и $0a0^2$. Согласно сказанному выше, язык L является регулярным и не является $L(\tau_n)$-языком. Тогда как язык $Lx=\{a0^max|m\geqslant 1\}$ также является регулярным, ибо порождается праволинейной КС-грамматикой $G=(V, S^p, \Pi, \sigma)$ при

$$\Pi=\{\sigma\to a\gamma, \gamma\to 0\mu,\ \mu\to 0\mu,\ \mu\to ax\},$$

и к тому же он является уже $L(\tau_4)$-языком с аксиомой $c_0=a0ax$, где $a\in S^p\diagdown 0$. Аналогичные рассуждения имеют место и в случае xL-языка. Сам же язык L порождается OL-системой [4] при $G=\{a,\ 0\}$, $E=\varnothing$, $\delta=\{0a\to 00a\}$ и $g=a0a$.

7. Регулярный язык, порождаемый грамматикой $G=(\{S^p,\ \sigma\}, S^p,\ \Pi,\ \sigma)$ с $\Pi=\{\sigma\to x\sigma,\ \sigma\to y\}$ $(x,\ y\in\overline{C}_p)$, порождается и τ_n-грамматикой. Однако при замене правил вывода на нелинейные уже вида

$$\Pi=\{\sigma\to x\sigma y\sigma,\ \sigma\to z\}\quad (x,\ y,\ z\in\overline{C}_p)$$

НС-грамматика порождает язык, который не может быть $L(\tau_n)$-языком.

8. Рассмотрим множество L^0 всех конечных слов в алфавите $S^p\diagdown 0$ $(p\geqslant 2)$. КС-грамматика $G=(\{\sigma,\ S^p\diagdown 0\},\ S^p\diagdown 0,\ \Pi,\ \sigma)$ с правилами

$$\Pi=\{\sigma\to 1\sigma,\ \sigma\to 2\sigma,\ldots,\sigma\to(p-1)\sigma,\ \sigma\to 1,\ldots,\ \sigma\to(p-1)\}$$

порождает язык L^0. Для порождения L^0-языка можно использовать и $<0,\ 1>$ L-систему [4] следующего вида: $\Gamma=(G,\ E,\ \delta,\ g)$ при $G=S^p\diagdown 0, E=\{0\}, g=1$,

$$\begin{cases}\delta(a,b)=\{a\},\ \delta(a,0)=\{a+1\},\ \delta(\overline{a,0})=\{x,y\},\\ a,x,y\in S^p\diagdown 0\quad (a=1,p-2).\end{cases}$$

Возникает вопрос: *может ли быть L^0-язык $L(\tau_n)$-языком?*

9. Пусть преобразование ω таково, что $(\forall x\in\overline{C}_p)\,(\omega(x)=a^{|x|})$, где $a\in S^p\diagdown 0$ есть буква, имеющая максимальное число вхождений в слово x; если несколько букв a_i имеют одинаковое максимальное число вхождений в слово x, то $a=\max\limits_i(a_i)$. Можно показать, что $\omega(\overline{C}_p)$ является уже $L(\tau_2)$-языком.

10. Пусть $x=x_1\ldots x_m$ есть любое слово из $\overline{C}_p$. Тогда существует грамматика $\tau_n=(n, S^p\cup Q,\ \tau,\ c_0)$ $(c_0\in S^p\diagdown 0)$ такая, что порождает слово x не более чем за $l=[(m-n)/(n-1)]$ переходов функции τ при мощности вспомогательного алфавита Q не более чем в $[l(n-1)+n+1]\cdot$ $\cdot\, l/2$ букв ($l=0$, если $(m-n)/(n-1)\leqslant 0$).

11. Пусть имеется грамматика $\tau_n=(n, S^p, \tau, c_0)$. Множество слов $L_{\tau_n}^{-1}$ определяем как $L_{\tau_n}^{-1}=\{c_0\tau^{-i}\}$ $(i=0,\ 1,\ 2,\ldots)$, где $(c_0\tau^{-i})\tau=c_0\tau^{-i+1}$

$(i=1, 2, 3, \ldots)$. Возникает интересный вопрос: *всегда ли множество $L_{\tau_n}^{-1}$ является $L(\tau_m)$-языком?* Данный вопрос предполагает некоторые интерпретации с точки зрения обратимости протекания процессов в мире, определяемом однородной структурой. Можно ли показать, что множество $L_{\tau_n}^{-1}$ всегда является $L(\tau_m)$-языком, если функция τ в ОС не имеет НКФ?

Из определения языка $L(\tau_n)$ следует, что он не может быть пустым, так как сама аксиома является элементом языка. Однако $L(\tau_n)$ может быть конечным или бесконечным. Язык $L(\tau_n)$ может быть конечным тогда и только тогда, когда цепочка слов $c_0\tau^i$ $(i=0, 1, 2, \ldots)$ для τ_n-грамматики является периодической или содержит слово c' такое, что имеет место соотношение $(\forall i)$ $(c'\tau^i=c')$. Иначе язык $L(\tau_n)$ будет бесконечным. Но тогда из результатов § 2 гл. 2 [1] вытекает следующая

Теорема 4. *Свойства τ_n-грамматики порождать конечный или бесконечный язык $L(\tau_n)$ алгоритмически нераспознаваемы.*

Пустым словом в языке $L(\tau_n)$ будем называть нулевую конфигурацию ОС. Из результатов гл. 2 [1] также следует, что проблема существования пустого слова в $L(\tau_n)$-языке алгоритмически неразрешима.

Рассмотрим теперь некоторые соотношения между неконструируемостью в ОС и порождением ОС монотонных $L(\tau_n)$-языков (МН-языков). Будем говорить, что грамматика $\tau_n=(n, S^p, \tau, c_0)$ порождает МН-язык, если

$$(\forall t \geqslant 0)\ (|c_0\tau^t| \leqslant |c_0\tau^{t+1}|).$$

Очевидно, что это условие аналогично условию неубывания длины (УНД) у НС-грамматик. УНД означает следующее. Так, например, если продукции полутуэвской системы имеют вид

$$Pg_iQ \to P\overline{g}_iQ, \quad |g_i| \leqslant |\overline{g}_i| \quad (i=\overline{1, d}),$$

то будем говорить, что эта система удовлетворяет УНД. Можно показать, что класс языков, порождаемых НС-грамматиками, совпадает с классом языков, порождаемых полутуэвскими системами, удовлетворяющими УНД. Класс МН-языков, порождаемых τ_n-грамматиками, параллелен классу НС-языков, порождаемых грамматиками, удовлетворяющими УНД. Нетрудно заметить, что любой МН-язык (также как и НС-язык) является рекурсивным множеством слов.

Приведем теперь некоторые результаты. ОС может иметь НКФ (НКФ-1) и порождать только бесконечные МН-языки. В качестве примера можно рассмотреть одномерную ОС с функцией τ, определенной соотношениями

$$\begin{array}{lll} 00 \to 0, & 10 \to 1, & x, y \in S^p \setminus \{0, 1, 2\}, \\ 01 \to 1, & 02 \to 1, & x', y' \in S^p \setminus 0. \\ 0x \to x', & 20 \to 1, & \\ & y0 \to y', & \end{array}$$

Такая ОС имеет СТКФ уже следующего вида: $(010)\tau=(020)\tau=11$, и, значит, имеет НКФ и НКФ-1. С другой стороны, такая ОС порождает только МН-языки, так как выполняется условие

$$(\forall c_0 \in \overline{C}_p)\ (\forall t)\ (|c_0\tau^t| < |c_0\tau^{t+1}|).$$

ОС может также иметь НКФ-1 без НКФ и порождать только бесконечные МН-языки. Примером может служить ОС с функцией τ, заданной соотношениями

$$x_1 x_2 \ldots x_n \to \sum_{1}^{n} x_i \pmod p.$$

Из результатов работы [9] следует, что для того, чтобы ОС могла порождать только бесконечные МН-языки необходимо, чтобы она имела НКФ-1. Действительно, если в ОС отсутствуют НКФ-1, то она имеет графы состояний только типов: (а); (а), (с) или (с) [9]. Отсюда уже нетрудно заключить, что ОС не может порождать только бесконечные МН-языки. С другой стороны, пусть ОС имеет НКФ-1. Но тогда она может иметь графы состояний также и (а) типа [9]. Значит, условие существования НКФ-1 не является достаточным. Отсюда получаем следующую теорему.

Теорема 5. *Существование НКФ-1 в ОС необходимо, но не достаточно для того, чтобы ОС порождала только бесконечные МН-языки.*

Из теоремы 5 следует, что хотя в имеющихся в настоящее время классификациях примитивно рекурсивных множеств НС-языки (а также, пожалуй, и МН-языки) по сложности и оказываются на нижних ступенях иерархии, однако ОС, порождающие только бесконечные МН-языки, должны иметь НКФ-1.

В связи с вышесказанным возникает интересный вопрос: *можно ли доказать, что если в ОС отсутствуют НКФ-1, то она порождает только рекурсивные $L(\tau_n)$-языки?*

Последовательность слов из $\overline{C}_p\,\Omega: c_1, c_2, c_3, \ldots, c_i, \ldots$ будем называть формульной, если любое слово из Ω может быть описано одной (из конечного списка) формулой следующего вида:

$$c_i = c_{j_1}(i)\, c_{j_2}(i)\, c_{j_3}(i) \ldots c_{j_k}(i),$$

где $(\forall j_l)(\forall i)\,(c_i, c_{j_l}(i) \in \overline{C}_p)$ и $j_l \in \{1, 2, 3, \ldots\}$. Язык $L(\tau_n)$ будем называть формульным, если соответствующая τ_n-грамматика порождает формульную последовательность слов. Нетрудно заметить, что любой конечный $L(\tau_n)$-язык является формульным и любой формульный $L(\tau_n)$-язык является рекурсивным. Проиллюстрируем сказанное следующими примерами.

Грамматика $\tau_2 = (2, S^3, \tau, 12202)$, где функция τ определена соотношениями

$$\begin{array}{lll} 00 \to 0, & 10 \to a, & 20 \to 2, \\ 01 \to 2, & 11 \to 0, & 21 \to 2, \\ 02 \to 1, & 12 \to 0, & 22 \to 1, \quad a \in S^3, \end{array}$$

порождает формульный $L(\tau_2)$-язык вида $L = \{c_{2i-1}, c_{2i}\}$ при

$$\begin{cases} c_{2i-1} = 122\,(02)^i, \\ c_{2i} \;\;\;\, = 2012\,(12)^i \\ (i \;\;\;\, = 1, 2, 3, \ldots), \end{cases} \quad \text{или} \quad \begin{cases} c_1 = 12202,\ c_2 = 201212, \\ c_{2i-1} = c_{2i-3}\,(02), \\ c_{2i} = c_{2i-2}\,(12) \\ (i = 2, 3, 4, \ldots). \end{cases}$$

Заменяя в данном примере аксиому c_0 грамматики на $c_0 = 120202$, получаем опять формульный язык, описываемый уже формулами вида

$$\begin{cases} c_{2i-1} = (12)^i 0202, \\ c_{2i} = (20)^i 21212 \\ (i = 1, 2, 3 \ldots), \end{cases} \quad \text{или} \quad \begin{cases} c_1 = 120202,\ c_2 = 2021212, \\ c_{2i-1} = (12)\, c_{2i-3}, \\ c_{2i} = (20)\, c_{2i-2} \\ (i = 2, 3, 4, \ldots). \end{cases}$$

Будем называть $L(\tau_n)$-язык, описываемый m формулами, формульным языком глубины m. Понятие формульности представляет определенный интерес при исследовании синтаксической структуры слов, порождаемых τ_n-грамматикой. Вышеприведенные примеры давали нам формульные языки глубины 2. Грамматика $\tau_2=(2, S^5, \tau, 321)$ с функцией τ, определенной соотношениями

$$\begin{array}{llll} 00\to 0, & 13\to 1, & 21\to 2, & 04\to 0, \\ 10\to 1, & 01\to 2, & 43\to 1, & \\ 32\to 3, & 03\to 4, & 02\to 3, & \end{array}$$

а на остальных наборах $xy\to S^5$, дает пример формульного языка глубины $m=4$ с формулами вида

$$c_j=\alpha[\beta=j \pmod 4]\,(321)^{]j/4[},$$

где
$$\alpha[\beta]=\begin{cases}\Lambda, \text{ если } \beta=1,\\ 4, \text{ если } \beta=2,\\ 1, \text{ если } \beta=3,\\ 21, \text{ если } \beta=0.\end{cases}$$

Возникает вопрос: *разрешима ли алгоритмически проблема формульности произвольного $L(\tau_n)$-языка?*

Представляет интерес и обратная задача: по данным формулам построить τ_n-грамматику, порождающую $L(\tau_n)$-язык заданной формульности. Эта задача может иметь отрицательное решение уже для довольно простых типов формульности. Так, например, пусть нам нужно построить τ_n-грамматику, порождающую язык $L=\{c_i\,|\,i\geqslant 0\}$, где слова c_i для любого $i\geqslant n$ имеют вид $c_i=c_{i-2}c_{i-1}$. Положим, что $Q=\min\{c_{n-2}, c_{n-1}\}$. Теперь нетрудно показать, что $l(c_{n+k})\geqslant Q\cdot\Phi(k)$ $(k=0,1,2,\ldots)$, где $\Phi(k)$ есть последовательность чисел Фибоначчи. А так как индекс k у слова c_{n+k} растет линейно, а длина c_{n+k} не меньше величины $Q\cdot\Phi(k)$, то отсюда уже можно заключить, что не существует τ_n-грамматики, решающей поставленную выше задачу. Более того, если одна из формул $c_{i+k}=F_i(k)$ формульной последовательности $\{c_l\,|\,l\geqslant 0\}$ удовлетворяет условию

$$(\mathrm{E}i)\,(\forall k\geqslant 0)\,(l[F_i(k)]\geqslant\min(k^\alpha, \alpha^k)) \quad (\alpha>1),$$

то не существует τ_n-грамматики, порождающей $L(\tau_n)$-язык вида $\Omega=\{c_l\,|\,l\geqslant 0\}$. Для того, чтобы последовательность слов Ω порождалась некоторой τ_n-грамматикой, необходимо, чтобы выполнялись условия

$$(\forall\,\geqslant 0)\,(|\,l[c_{i+1}]-l[c_i])\,|\leqslant\beta=\text{const}) \quad (AVZ).$$

В противном случае последовательность Ω не может порождаться никакой τ_n-грамматикой. То же относится и к любому множеству слов $GS\subset\overline{C}_{p^*}$: необходимым условием быть множеству слов GS $L(\tau_n)$-языком является возможность представления GS в виде последовательности слов Ω, удовлетворяющей условию (AVZ).

Для оценки сложности вывода в τ_n-грамматиках естественно ввести функцию $T(\tau_n, c)$, содержательный смысл которой состоит в оценке числа переходов глобальной функции τ грамматики, необходимого для получения из аксиомы c_0 произвольного слова $c\in L(\tau_n)$. Для функции $T(\tau_n, c)$ нетрудно получить следующую оценку снизу:

$$\frac{c'_0\dot{-}c'}{2(n-1)}\leqslant T(\tau_n, c),$$

где $c' = l(c)$ и

$$x \dot{-} y = \begin{cases} 1, & \text{если } x \leqslant y, \\ x-y, & \text{если } x > y. \end{cases}$$

Верхнюю оценку в общем случае (так как язык $L(\tau_n)$ может быть и нерекурсивным) получить нельзя. Если же язык $L(\tau_n)$, например, является МН-языком, то оценки для функции $T(\tau_n, c)$ имеют следующий вид:

$$\frac{c'_0 \dot{-} c'}{2(n-1)} \leqslant T(\tau_n, c) \leqslant (1-1/p)\,(p^{c'} - p^{c'_0 - 1}).$$

Функция $T(\tau_n, c)$ характеризует по сути дела временную сложность τ_n-грамматик.

В общем случае грамматика Хомского является более широким понятием, чем грамматика, определенная в начале настоящей статьи. А именно, грамматика Хомского — это упорядоченная четверка $G = (V, \Sigma, \Pi, \sigma)$. Здесь V и Σ — непересекающиеся непустые конечные алфавиты; σ — некоторый элемент V; Π — конечное множество продукций вида $\varphi \to \Psi$, где φ и Ψ — различные слова в алфавите $V \cup \Sigma$ и $\to$ есть символ, не входящий в алфавит $V \cup \Sigma$. Поэтому выделим теперь те однородные структуры, которые эквивалентны грамматикам Хомского. При этом под эквивалентностью двух порождающих грамматик понимается совпадение порождаемых ими языков.

Пусть некоторая грамматика $\tau_n = (n, S^p, \tau, c_0)$ порождает язык $L(\tau_n)$. Попытаемся определить те условия, при которых грамматика Хомского G будет порождать в точности язык $L(\tau_n)$. Определяем грамматику $G = (V, \Sigma, \Pi, \sigma)$ следующим образом: $V = \{\sigma, \alpha, \gamma, \beta, \Theta, \delta, \mu, \varrho\}$, $\Sigma = S^p$,

$$\Pi: \left\{ \begin{array}{lcll}
\sigma & \longrightarrow & \alpha c_0 \gamma, & \\
\sigma & \longrightarrow & c_0 = x_i x_{i+1} \ldots x_{i+t}, & \\
\alpha x_i x_{i+1} \ldots x_{i+t} \gamma & \longrightarrow & x'_{i-n} \ldots x'_i \ldots x'_{,i+t}, & \Big\} \\
\alpha x_i x_{i+1} \ldots x_{i+t} \gamma & \longrightarrow & \alpha x'_{i-n} \ldots x'_i \ldots x'_{i+t} \gamma, & \Big\}\ t < n-1 \\
\alpha x_i x_{i+1} \ldots x_{i+n-2} & \longrightarrow & \Theta x'_{i-n} \beta x'_{i-n+1} \beta \ldots \beta x'_{i-1} \beta x_i \ldots x_{i+n-2}, & \\
\beta x_i x_{i+1} \ldots x_{i+n-1} & \longrightarrow & \beta x'_i \beta x_{i+1} \ldots x_{i+n-1}, & \\
\beta x_i x_{i+1} \ldots x_{i+n-2} \gamma & \longrightarrow & \beta x'_i \beta x'_{i+1} \beta \ldots \beta x'_{i+n-2} \delta, & \\
\beta x_i \delta & \longrightarrow & \mu x_i \gamma, & \\
\beta x_i \delta & \longrightarrow & \varrho x_i, & \\
\beta x_i \varrho & \longrightarrow & \varrho x_i, & \\
\Theta x_i \varrho & \longrightarrow & x_i, & \\
\beta x_i \mu & \longrightarrow & \mu x_i, & \\
\Theta x_i \mu & \longrightarrow & \alpha x_i. &
\end{array} \right.$$

Исследуя теперь деревья выводов в определенной таким образом грамматике Хомского, можно получить, что такая грамматика может быть эквивалентна (при условии детализации продукций в Π согласно функции перехода τ в ОС) грамматике $\tau_n = (n, S^p, \tau, c_0)$ при условии, что τ_n-грамматика порождает язык, обладающий определенным свойством. Прежде всего введем следующее определение. Язык $L(\tau_n)$ будем называть m-укорачиваемым слева {справа}, если для любого t слово $c_0\tau^{t+1}$ по сравнению со словом $c_0\tau^t$ может изменяться по длине слева {справа} в интервалах $+(n-1) : -m\{0 : -m\}$ букв. Используя вышесказанное, можно показать, что имеет место следующее

Предложение. *Если $L(\tau_n)$-язык является 0-укорачиваемым слева и $(n-1)$-укорачиваемым справа, то существует грамматика Хомского G, такая, что $L(G) = L(\tau_n)$.*

Доказательство этого предложения оставляем читателю в качестве упражнения.

Введем теперь в рассмотрение понятие изотонной структурной (ИС) грамматики [11]. И С - г р а м м а т и к а есть пятерка $G=(N, T, \#, \Pi, \sigma)$, *где*

1) N — непустой вспомогательный алфавит,
2) T — непустой основной алфавит ($T \cap N = \varnothing$),
3) $\#$ — пустой символ,
4) Π — конечное множество изотонных продукций,
5) σ — начальный символ или аксиома.

Изотонная продукция представляет собой отображение конфигурации блока ОС в новую конфигурацию того же блока, причем символы конфигурации принадлежат алфавиту $T \cup N \cup \{\#\}$. В одномерном случае на изотонные продукции $\alpha \to \beta$ накладываются ограничения $\|\alpha\|=\|\beta\|$. Для того же, чтобы ИС-грамматика не порождала только тривиальных языков, введен пустой символ $\#$. Так, например, начальная строка при этих условиях имеет следующий вид: $\#\ldots\#\#\sigma\#\#\ldots\#$. Пустой символ $\#$, таким образом, аналогичен пустому символу в теории автоматов. В этом случае язык $L(G)$ есть множество всех слов (окруженных символами $\#$) из основных символов, выводимых из начального символа σ. Зададим ИС-грамматику $G=(N, T, \#, \Pi, \sigma)$ следующим образом: $N=\{\sigma, \alpha, \beta, \gamma, \delta\}$, $T=S^p$ и продукции в виде

$$\Pi: \begin{cases} \sigma\underbrace{\#\#\ldots\#\#}_{m} \longrightarrow \alpha x_1x_2\ldots x_m, \\ \#\alpha \longrightarrow \#\#, \\ \alpha x_1x_2\ldots x_m\underbrace{\#\ldots\#}_{n-1} \longrightarrow \alpha x'_{n-2}\ldots x'_0x'_1x'_2\ldots x'_m \quad (m\leqslant n-1), \\ \underbrace{\#\ldots\#}_{n}\alpha x_1x_2\ldots x_n \longrightarrow \beta x'_{n-2}\ldots x'_0x'_1\delta x_2\ldots x_n, \\ \delta x_ix_{i+1}\ldots x_{i+n-1} \longrightarrow x'_i\delta x_{i+1}\ldots x_{i+n-1}, \\ \delta x_ix_{i+1}\ldots x_{i+n-2}\# \longrightarrow x'_ix'_{i+1}\ldots x'_{i+n-2}\delta\#, \\ \delta\# \longrightarrow \gamma\#, \\ x_i\gamma \longrightarrow \gamma x_i, \\ \beta\gamma \longrightarrow \#\alpha, \end{cases}$$

где слово $x_1x_2\ldots x_m=c_0$ является аксиомой в грамматике $\tau_n=(n, S^p, \tau, c_0)$.

Исследуя деревья выводов в заданной таким образом ИС-грамматике, можно показать, что такая грамматика (при условии детализации продукций в Π согласно функции перехода τ в ОС) будет эквивалентна грамматике $\tau_n=(n, S^p, \tau, c_0)$. Таким образом, для любой τ_n-грамматики существует эквивалентная ей ИС-грамматика. Возможной модификацией в определении вывода в ИС-грамматике есть параллельное применение продукций, т. е. все вхождения левой части какой-либо продукции из Π одновременно заменяются на правую часть продукции. В общем случае можно это потребовать и для всех вхождений всех левых частей продукций из Π. Определенные таким образом грамматики называются параллельными [12]. Можно показать, что τ_n-грамматики являются собственным подклассом класса всех параллельных грамматик.

3. Говорят, что множество замкнуто относительно некоторой операции, если результат применения этой операции к любому элементу множества (если операция унарна) или к любой паре элементов (если операция бинарна) и т. п. содержится в этом множестве. Рассмотрим с этой точки зрения некоторые операции над языками $L(\tau_n)$.

Теорема 6. *Объединение двух языков $L(\tau_n)$ может быть, но не обязательно является, снова $L(\tau_n)$-языком.*

Доказательство. Пример 1 настоящей статьи показывает, что объединение двух языков $L(\tau_n)$ может быть снова $L(\tau_n)$-языком. Рассмотрим теперь языки $L_1=\{1^m, 2^m \mid m\geqslant 1\}$ и $L_2=\{2^m0^m1 \mid m\geqslant 1\}$. Очевидно, что оба они являются уже $L(\tau_3)$-языками. Является ли $L(\tau_n)$-языком язык $L=L_1\cup L_2=\{1^m \mid m\geqslant 1\}\cup\{2^m0^m1 \mid m\geqslant 1\}\cup\{2^m \mid m\geqslant 1\}$? Предположим, что он является $L(\tau_n)$-языком. Значит, существует грамматика $\tau_n=(n,\ S^p,\ \tau,\ c_0)$, которая порождает этот язык. Тогда для $(\forall\, m\geqslant 1)\,(E!t)\,(c_0\tau^t=2^m0^m1)$. Пусть $m\gg n$ и $c_0\tau^t=2^m0^m1$, т. е. $c_0\tau^t=\bar{0}\underbrace{22\ldots2}_{m}\underbrace{00\ldots01}_{m}\bar{0}$. С другой стороны, $(E!t')\,(c_0\tau^{t'}=1)$ и тогда $c_0\tau^{t'+1}=c_0'\in L$, причем $c_0'\neq 1$. Но тогда, применяя к слову $c_0\tau^t$ преобразование τ, получаем

$$\begin{array}{c}\underbrace{\bar{0}\overbrace{22\ldots2}^{m}}_{n}\overbrace{00\ldots0}^{m-n}\underbrace{\overbrace{0\ldots01}^{n}}\bar{0}\\ \downarrow\qquad\qquad\qquad\downarrow\tau\\ \bar{0}\quad c_0{}^{*}00\ldots0\ c'_0\ \bar{0}.\end{array}$$

Отсюда следует, что $c_0{}^{*}=0$, ибо в противном случае слово $c_0\tau^{t+1}=\bar{0}c^0\bar{0}=$ $=\bar{0}c_0{}^{*}0^{m-n}c'_0\bar{0}\notin L$. А так как слово $2^m\in L\,(m\geqslant 1)$, то в некоторый момент t'' имеем $c_0\tau^{t''}=2^m$ и дальнейшее применение τ-преобразования дает пустое слово $c_0\tau^{t''+1}=0$, т. е. язык L является конечным, что противоречит условию. Полученное противоречие говорит об отсутствии τ_n-грамматики, порождающей язык L. Более того, вероятно, можно показать, что уже объединение конечного множества $\cup\subset\overline{C}_p$ и $L(\tau_n)$-языка не всегда дает снова $L(\tau_n)$-язык. Теорема доказана.

Теорема 7. *Пересечение двух языков $L(\tau_n)$ может быть, но не обязательно является, снова языком $L(\tau_n)$.*

Доказательство. Пример 1 настоящей статьи показывает, что пересечение двух языков $L(\tau_n)$ может быть снова $L(\tau_n)$-языком. Действительно, пересечение языков $L\cap L_2$ есть снова $L(\tau_3)$-язык. Рассмотрим теперь языки L_1 и L_2, порожденные соответственно грамматиками $\tau_2=(2, S^2, \tau, 1)$ и $\tau_3=(3, S^2, \tau', 111)$, где

$$\tau: xy\to x+y \pmod 2,\qquad \tau': \begin{array}{l} xyz\to x+y+z \pmod 2.\\ 111\to\quad 0\end{array}$$

Можно убедиться, что пересечение $L(\tau_n)$-языков $L_1\cap L_2$ есть множество $L=\{1\,0^{2^m+\sum\limits_1^m 2^{m-i}}1 \mid m\geqslant 1\}$. Но из вышесказанного легко заключаем, что множество L не является $L(\tau_n)$-языком. Этим теорема доказана.

Введем операцию квази-пересечения языков следующим образом: квази-пересечение языка L_1 с языком $L_2\,(L_1\wedge L_2)$ есть язык $L_1'\subseteq L_1$, состоящий из всех тех слов $\alpha\in L_1$, для которых существуют буквенно-эквивалентные слова $\alpha^*\in L_2$. Будем говорить, что слова α и α^* буквенно-эквивалентны, если одно из них получается из другого только перестановкой входящих в него букв. Нетрудно убедиться, что для любых двух языков L_1 и L_2 имеют место следующие соотношения:

$$(L_1\cap L_2)\subseteq(L_1\wedge L_2),\quad L_1\wedge L_2\equiv L_1\wedge L_2^R.$$

Используя же пример теоремы 7, нетрудно убедиться, что квази-пересечение двух $L(\tau_n)$-языков не всегда является снова $L(\tau_n)$-языком.

Теорема 8. *Результат обращения языка $L(\tau_n)$ есть снова язык типа $L(\tau_n)$.*

Доказательство этой теоремы оставляем читателю в качестве упражнения.

Показать, что для этого достаточно в первоначальной локальной функции $\overset{v}{L}_{(n)}$, определяющей преобразование τ, каждое соотношение вида $x_1x_2\ldots x_n \to x_1'$ заменить соотношением $x_1x_2\ldots x_n \to \overset{v}{L}_{(n)}(y_1, y_2, \ldots, y_n)$, где $y_1y_2\ldots y_n = (x_1x_2\ldots x_n)^R$ и $x_i, y_i, x_1' \in S^p (i=\overline{1, n})$. Из этого факта, например, вытекает следующий результат. Пусть функция $\overset{v}{L}_{(n)}$, определяющая преобразование τ в грамматике $\tau_n = (n, S^p, \tau, c_0)$, такова, что

$$\overset{v}{L}_{(n)}(x_1, x_2, \ldots, x_n) = \overset{v}{L}_{(n)}(y_1, y_2, \ldots, y_n)$$

при условии, что $x_1x_2\ldots x_n = (y_1y_2\ldots y_n)^R$. Такую функцию $\overset{v}{L}_{(n)}$ естественно назвать симметричной. Тогда обращение любого языка $L(\tau_n)$ есть снова $L(\tau_n)$-язык, порождаемый «почти той же» грамматикой $\tau_n = (n, S^p, \tau, \overline{c_0})$].

Из теоремы 8 следует, что множество L является $L(\tau_n)$-языком тогда и только тогда, когда таковым является и множество L^R.

Теорема 9. *Произведение двух языков $L(\tau_n)$ и итерация языка $L(\tau_n)$ могут быть, но не обязательно являются снова $L(\tau_n)$-языком.*

Доказательство. Язык $L=\{1^m | m \geqslant 1\}$ является уже языком $L(\tau_2)$. Языки $L^2=LL$ и L^* (итерация) также являются языками $L(\tau_2)$. Рассмотрим теперь язык $L=\{1, 211\} \cup \{2^{2m}0^{m+1}11 | m\geqslant 1\}$. Можно показать, что язык L является $L(\tau_4)$-языком. Тогда произведение $LL=L^2$ есть язык $L^2=\alpha=\{11, 1211, 2111, 211211\} \cup \{12^{2m}0^{m+1}11, 2112^{2m}0^{m+1}11 | m\geqslant 1\} \cup \{2^{2p}0^{p+1}112^{2m}0^{m+1}11 | p, m\geqslant 1\} \cup \{2^{2p}0^{p+1}11211 | p\geqslant 1\}$. Допустим, что язык L^2 является $L(\tau_n)$-языком, порождаемым некоторой грамматикой $\tau_n=(n, S^p, \tau, c_0)$. В этом случае должно иметь место соотношение

$$(\forall p\geqslant 1)(\forall m\geqslant 1)(E!t)(c_0\tau^t = 2^{2p}0^{p+1}112^{2m}0^{m+1}11).$$

Пусть $n \ll p, m$. Тогда слово $c_0\tau^{t+1}$ принимает следующий вид:

$$\overline{0}c_0'\,0^{p-n+2}c_0''\,0^{m-n+2}c_0'''\overline{0},$$

причем $(2^{2p})\tau=c_0'$, $(112^{2m})\tau=c_0''$ и $(11)\tau=c_0'''$. Однако так как слово $c_0\tau^{t+1}$ должно принадлежать языку L^2, то имеет смысл ограничиться рассмотрением только следующих возможностей. Пусть $c_0'''=0$, но тогда язык L^2 будет конечным (что противоречит условию), ибо слово $11\in L^2$, и значит для него имеет место $(E!t')(c_0\tau^{t'}=11)$. А это значит, что $c_0'''\in\alpha\setminus\{11\}$ или $c_0'''=11211$. В первом случае тогда имеем $c_0'=c_0''=0$. Откуда следует, что существует бесконечно много слов в цепочке $<c_0>$, которые порождают одно и то же слово $c_0'''\in\alpha\setminus\{11\}$ на следующем шаге работы ОС, чего быть не может. Значит, под сомнением остался только случай $c_0'''=11211$. Тогда либо $c_0'=0$, либо $c_0''=0$. И в обоих случаях получаем соотношения для глобальной функции перехода

$$\underbrace{22\ldots2}_{i}\,\underbrace{00\ldots0}_{j} \to 0 \qquad (i+j=n;\ i=\overline{0, n}),$$

а это свидетельствует о том, что подслова 2^{2m} в словах из языка L^2 могут иметь только конечную длину, что противоречит условию. Таким образом, язык L^2 не может быть $L(\tau_n)$-языком. А так как итерация (с точностью до пустого слова) языка L определяется как

$$L^* = L \cup L^2 \cup L^3 \cup \ldots \cup L^d \cup \ldots,$$

то теперь нетрудно завершить доказательство самой теоремы.

Из теорем 6—9 следует, что множество $L(\tau_n)$-языков незамкнуто относительно операций объединения, произведения, итерации и пересечения и замкнуто относительно операции обращения. Из вышеизложенного следует, что класс $L(\tau_n)$-языков является собственным подклассом класса языков, порождаемых $<\kappa, e>$ L-системами. А такие языки незамкнуты (уже OL-языки) относительно всех основных операций [4]. Таким образом, сужение $<\kappa, e>$ L-языков до $L(\tau_n)$-языков минимально расширяет класс операций, относительно которого эти языки замкнуты.

Под дополнением языка $L(\tau_n)$ мы понимаем дополнение его относительно множества $\overline{C}_p$. Нам не удалось найти ни одного примера $L(\tau_n)$-языка, дополнение которого было бы языком $L(\tau_n)$. Более того, мы предполагаем, что имеет место следующий результат: *дополнение любого $L(\tau_n)$-языка не является $L(\tau_n)$-языком,* однако доказать его мы пока не смогли. Выше мы показали, что семейство всех $L(\tau_n)$-языков замкнуто только относительно обращения и поэтому не может принадлежать AFL. Таким образом, мы имеем пример семейства языков, которое очень интересно для моделирования развивающихся биологических систем, но к которому теория AFL не применима. Это является еще одним доводом в пользу изучения введенного нами семейства формальных языков.

Операции левого и правого деления языка L на слово a определяются и обозначаются соответственно следующим образом:

$$a \setminus L = \{\Psi \mid a\Psi \in L\}, \quad L/a = \{\Psi \mid \Psi a \in L\}.$$

Левое (правое) деление $L(\tau_n)$-языка на слово не всегда дает снова $L(\tau_n)$-язык. В качестве контрпримера можно рассмотреть язык $L = \{10^m 123 \mid m \geqslant 1\}$, который является уже $L(\tau_3)$-языком. Тогда как множество слов $L/a = \{10^m 1 \mid m \geqslant 1\}$ $(a = 23)$, как следует из вышесказанного, не является $L(\tau_n)$-языком. Аналогично рассматривается и случай левого деления языка на слово.

Рассмотрим еще ряд операций над $L(\tau_n)$-языками. Пусть множество $GS \subset \overline{C}_p$ является $L(\tau_n)$-языком и τ' — произвольная глобальная функция перехода в алфавите S^p. Определим множество $\tau'(GS) = \{x \mid x = \tau'(x'),\ x' \in GS\}$. Всегда ли множество $\tau'(GS)$ будет снова $L(\tau_n)$-языком? Рассмотрим грамматику $\tau_3 = (3, S^4, \tau, 102)$ с функцией перехода τ, определенной соотношениями

$020 \to 3,$	$001 \to 0,$	$332 \to 3,$	$333 \to 3,$
$102 \to 3,$	$200 \to 2,$	$133 \to 3,$	$000 \to 0.$
$010 \to 1,$	$013 \to 1,$	$320 \to 3,$	

На остальных наборах $<x, y, z>$ $(x, y, z \in S^4)$ функция τ определяется произвольно. Заданная таким образом τ_3-грамматика порождает язык $GS = \{13^m 2 \mid m \geqslant 2\} \cup \{102\}$. Определим теперь τ'-преобразование следующими соотношениями:

$000 \to 0,$	$013 \to 1,$	$020 \to 1,$
$001 \to 0,$	$333 \to 0,$	$200 \to 0,$
$010 \to 1,$	$102 \to 0,$	$133 \to 0.$
$332 \to 0,$	$320 \to 1,$	

На остальных наборах $<x, y, z>$ $(x, y, z\in S^4)$ функция τ' определяется произвольно. Тогда нетрудно убедиться, что $\tau'(GS)=\{10^m1\,|\,m\geqslant 1\}$, а это множество, как мы уже видели, не может быть $L(\tau_n)$-языком. Этим доказана следующая

Теорема 10. *Пусть L является $L(\tau_n)$-языком и τ'-произвольной глобальной функцией в том же алфавите, что и L. Тогда множество $\tau'(L)$ не обязательно является снова $L(\tau_n)$-языком.*

Пусть бесконечное множество $L\subset\overline{C}_p$ есть $L(\tau_n)$-язык. Тогда по теореме 8 $L(\tau_n)$-языком будет и множество L^R. Однако множество $L\cup L^R$ не всегда является $L(\tau_n)$-языком. Рассмотрим следующий пример. Нетрудно убедиться, что языки $L=\{10^m2\,|\,m\geqslant 1\}$ и $L^R=\{20^m1\,|\,m\geqslant 1\}$ являются $L(\tau_n)$-языками. Тогда как множество $L'=L\cup L^R$ не является $L(\tau_n)$-языком. Действительно, пусть L' есть $L(\tau_n)$-язык. Но тогда существуют такие $c_0\in L'$ и t, что $c_0\tau^t=20^m1\,(m\gg n)$. В этом случае могут быть следующие возможности (исходя из того, что $c_0\tau^{t+1}\in L'$):

1) $c_0\tau^{t+1}=20^{m'}1$ $\quad(m'<m)$,
2) $c_0\tau^{t+1}=20^{m'}1$ $\quad(m'>m)$,
3) $c_0\tau^{t+1}=10^q2$.

Первый случай невозможен, так как в L' будут отсутствовать слова 20^m1 произвольной длины. Случай 2 невозможен, так как исключает множество слов вида 10^m2 начиная с некоторого $m\geqslant m^*$. В случае же 3 мы получаем цикл, т. е. язык L' является тогда конечным, что противоречит условию. Аналогично можно показать, что и множество $L''=LL^R$ не является языком $L(\tau_n)$. Таким образом, если бесконечное множество $L\subset\overline{C}_p$ является $L(\tau_n)$-языком, то $L\cup L^R$ и LL^R не обязательно являются тоже $L(\tau_n)$-языками.

Теперь мы рассмотрим некоторые преобразования, которые сохраняют свойство быть $L(\tau_n)$-языком. Сопоставим каждому символу $a\in S^p$ алфавит Σ_a и множество $\mu(a)\subseteq\Sigma_a{}^*$. Пусть $\mu(0)=0$ и $\mu(x_1\ldots x_r)=$ $=\mu(x_1)\ldots\mu(x_r)$ для каждого слова $x_1\ldots x_r\in\overline{C}_p$. Тогда отображение μ множества $\overline{C}_p$ во множество $2^{(\bigcup_a\Sigma_a)^*}$ будем называть **подстановкой**. Если же для каждого $a\in S^p$ множество $\mu(a)$ состоит из единого слова $w_a\in\Sigma_a{}^*$, то μ рассматривается как отображение множества $\overline{C}_p$ во множество $(\bigcup_a\Sigma_a)^*$ и называется **гомоморфизмом**. Это определение согласуется с обычным алгебраическим понятием гомоморфизма одной полугруппы в другую. При сделанных определениях имеет место следующая

Теорема 11. *Если L является $L(\tau_n)$-языком в алфавите S^p и μ-гомоморфизм, такой, что $\mu(a)$ есть слово вида $y_1{}^ay_2{}^a\ldots y_d{}^aq^a$ $(y_i{}^a\in$ $\in\{S^p\cup S^{p'}\}\setminus 0\,(i=\overline{1,d}))$; $((\forall a)\,(q^a\in Q))$, $(S^p\cup S^{p'})\cap Q=\varnothing)$ для каждого $a\in S^p\setminus 0$ и $\mu(0)=0^{d+1}$, то $\mu(L)$ тоже является $L(\tau_n)$-языком.*

Чтобы не загромождать изложение, дадим только схему доказательства. Так как L является $L(\tau_n)$-языком, то существует грамматика $\tau_n=(n, S^p, \tau, c_0)$, порождающая этот язык,

$$\tau: c_0\to c_1\to c_2\to\ldots\to c_i\to c_{i+1}\to\ldots$$

И для каждого $i\geqslant 0$ функция τ порождает (из аксиомы c_0) новое слово языка L

$$c_i=\quad\overline{0}x_1x_2\ldots x_m\overline{0},$$

$$c_i\tau=c_{i+1}=\quad\overline{0}x'_{-(n-2)}\ldots x'_0x'_1x'_2\ldots x'_m\overline{0}.$$

Гомоморфизм μ переводит слова c_i и c_{i+1} соответственно в слова вида $\mu(c_i)$ и $\mu(c_{i+1})$. Определим теперь функцию τ_μ в грамматике $\tau_{n'} = (n', S^p \cup S^{p'} \cup Q, \tau_\mu, \mu(c_0))$ при $n' = (d+1)n$ так, чтобы выполнялись соотношения $(\forall i \geqslant 0)(\mu(c_i)\tau_\mu = \mu(c_{i+1}))$. Исходя из условий теоремы, это можно сделать. Таким образом, язык $\mu(L)$ является $L(\tau_n)$-языком.

Однако далеко не каждый гомоморфизм сохраняет свойство языка L быть $L(\tau_n)$-языком. Так, нетрудно убедиться, что язык $L = \{14^m 24^m 3 \mid m \geqslant 0\}$ является уже $L(\tau_3)$-языком. Определим μ-гомоморфизм, например, следующим образом:

$$\mu(0) = 0,\ \mu(1) = 11,\ \mu(4) = 0,\ \mu(2) = 1,\ \mu(3) = 11.$$

Тогда $\mu(L) = \{110^m 10^m 11 \mid m \geqslant 0\}$; но множество слов $\mu(L)$ не может быть $L(\tau_n)$-языком.

Нетрудно показать, что если L есть $L(\tau_n)$-язык в алфавите S^p и μ-гомоморфизм имеет вид

$$\mu(0) = 0,\quad (\forall a \neq 0)(\mu(a) = a^*) \quad (a^* \neq 0) \quad \text{и}$$
$$(\forall a, a')((a, a' \neq 0) \,\&\, (a \neq a') \rightarrow \mu(a) \neq \mu(a')),$$

то $\mu(L)$ тоже является $L(\tau_n)$-языком. Такой тип гомоморфизма мы назовем μ-перетасовкой. Отсюда вытекает

Теорема 12. *Если L есть $L(\tau_n)$-язык и μ-перетасовка, то $\mu(L)$ тоже является $L(\tau_n)$-языком.*

Пусть преобразование μ_e^a таково, что $(\forall x \in L)(\mu_e^a(x) = x_{a/e})$, где $x_{a/e}$ есть слово, полученное из слова x заменой в нем всех вхождений буквы $a \in S^p$ на букву $e \notin S^p$. Тогда имеет место следующая

Теорема 13. *Если L есть $L(\tau_n)$-язык, то таковым является и язык $\mu_e^a(L)$.*

Если же не накладывать условия $e \notin S^p$ в определении преобразования μ_e^a, то теорема 13 в общем случае не будет иметь места. В качестве контрпримера можно рассмотреть $L(\tau_3)$-язык $L = \{10^m 2 \mid m \geqslant 1\}$ и μ_1^2. Тогда $\mu_1^2(L) = \{10^m 1 \mid m \geqslant 1\}$, но этот язык уже не является $L(\tau_n)$-языком.

Будем называть грамматику $\tau_n = (n, S^p, \tau, c_0)$ неукорачивающей, если $(\forall i \geqslant 0)(|c_0\tau^i| \leqslant |c_0\tau^{i+1}|)$. Приведем без доказательства следующие две теоремы.

Теорема 14. *Если язык L над алфавитом S^p порождается неукорачивающей τ_n-грамматикой и x — произвольное слово вида $x_1 \ldots x_d q$ $(x_i \in S^p \cup Q\ (i = \overline{1, d})$; $q \in Q$; $S^p \cap Q = \varnothing)$, то языки xL и Lx тоже являются языками $L(\tau_n)$.*

Пусть теперь L_1 есть $L(\tau_n)$-язык в алфавите S^p и L_2 есть $L(\tau_m)$-язык в алфавите $S^{p'}(S^p \cap S^{p'} = 0)$. Причем, если $L_1 = \{c_0^1 \tau_n^i \mid i \geqslant 0\}$ и $L_2 = \{c_0^2 \tau_m^i \mid i \geqslant 0\}$, то определяем язык $L_1 * L_2$ как $\{c'_i \mid i \geqslant 0\}$, где

$$c_i^1 = c_0^1 \tau_n^i = \bar{0} x_1 x_2 \ldots x_l \bar{0},$$

$$c_i^2 = c_0^2 \tau_m^i = \bar{0} y_1 y_2 \ldots y_p \bar{0},$$

$$c_i' = c_i^1 * c_i^2 = \bar{0} y_1 \ldots y_{p-l} x_1 \ldots y_{p-1} x_{l-1} y_p x_l \bar{0}.$$

Язык же $L_1 \times L_2$ определяем как $\{c'_i = c_i^1 c_i^2 \mid i \geqslant 0\}$.

Теорема 15. *Если языки L_1 и L_2 порождаются неукорачивающими τ_n- и τ_m-грамматиками, соответственно, то языки $L_1{}^*L_2$ и $L_1 \times L_2$ тоже являются $L(\tau_n)$-языками.*

Однако теорема 15 нарушается уже для случая языка $L_1 \times L_2$, если положить, например, $S^p = S^{p'}$ и $L_2 = L_1{}^R$. В качестве примера можно рассмотреть $L(\tau_3)$-язык $L_1 = \{10^m 2 \mid m \geqslant 1\}$. Из вышесказанного следует, что язык $L_1 \times L_1{}^R = \{10^m 220^m 1 \mid m \geqslant 1\}$ не является $L(\tau_n)$-языком.

Рассмотрим теперь конечное преобразование $L(\tau_n)$-языков. Конечным преобразователем (КП) называется упорядоченная шестерка $S = (K, \Sigma, \Delta, \delta, \lambda, q_1)$, где

1) K — непустое конечное множество (множество состояний),
2) Σ — алфавит (алфавит входных символов),
3) Δ — алфавит (алфавит выходных символов),
4) δ — отображение множества $K \times \Sigma$ в множество K,
5) λ — отображение множества $K \times \Sigma$ в множество Δ^*,
6) q_1 — выделенный элемент множества K (начальное состояние).

Если λ — отображение множества $K \times \Sigma$ в множество Δ, то КП S называется синхронным.

Отображение, осуществляемое КП, может быть определено следующим образом. Пусть $S = (K, \Sigma, \Delta, \delta, \lambda, q_1)$ есть КП. Тогда отображение $S(x) = \lambda(q_1, x)$, определенное для каждого $x \in \Sigma^*$, называется конечным преобразованием.

Теорема 16. *Не каждое конечное преобразование сохраняет свойство множества быть $L(\tau_n)$-языком.*

В качестве доказательства рассмотрим следующий пример. Язык $L = \{12^m 3 \mid m \geqslant 1\}$ является уже $L(\tau_3)$-языком. Введем конечное преобразование S, определяемое синхронным КП при следующих условиях:

$$
\begin{array}{lll}
K = \{q_1, q_2\}, & \Sigma = \{1, 2, 3\}, & \Delta = \{0, 4\}, \\
\delta(q_1, 1) = q_2, & & \lambda(q_1, 1) = 4, \\
\delta(q_2, 2) = q_2, & & \lambda(q_2, 2) = 0, \\
\delta(q_2, 3) = q_1, & & \lambda(q_2, 3) = 4.
\end{array}
$$

Нетрудно убедиться, что множество $S(L) = \{40^m 4 \mid m \geqslant 1\}$ не является $L(\tau_n)$-языком. С другой стороны, конечное преобразование, реализующее уже μ-перетасовку, сохраняет $L(\tau_n)$-язык.

4. В заключение настоящей работы рассмотрим некоторые алгоритмические проблемы для τ_n-грамматик. Сводка результатов, относящихся к проблемам разрешения для τ_n-грамматик, представлена в таблице, где указано, для каких проблем доказана их алгоритмическая разрешимость или неразрешимость.

Ответ на первую проблему следует непосредственно из определения языка $L(\tau_n)$. На проблему 9 дает ответ теорема 2. Ответ на проблему 3 дает теорема 1.10 из книги [1]. Теорема 4 решает проблему 2. Обратимся теперь к проблеме 7. Рассмотрим две грамматики $\tau_n = (n, S^p, \tau, c_0)$ и $\tau'_n = (n, S^p, \tau, c'_0)$. Нетрудно убедиться, что условие $L(\tau_n) \subset L(\tau'_n)$ эквивалентно условию $c_0 \in \langle c_0' \rangle$ $(c_0, c'_0 \in \overline{C}_p)$, т. е. проблеме 9. Действительно, пусть $c_0 \in \langle c'_0 \rangle$. Тогда $L(\tau_n) \subset L(\tau'_n)$. Пусть теперь $L = (\tau_n) \subset \subset L(\tau'_n)$. Но тогда соотношение $c_0 \notin \langle c_0' \rangle$ противоречит условию. Значит, $c_0 \in \langle c'_0 \rangle$. Отсюда нетрудно заключить, что и проблема 7 алгоритмически неразрешима. Относительно проблемы 6 можно сделать следующее замечание. Она неразрешима, если ОС, не имеющие НКФ, могут иметь нерекурсивные последовательности конфигураций $\langle c_0 \rangle$. Рассмотрим опять две грамматики $\tau_n = (n, S^p, \tau, c_0)$ и $\tau'_n = (n, S^p, \tau, c'_0)$.

№	Проблема	Результат
1.	Пуст ли $L(\tau_n)$-язык?	Нет
2.	Бесконечен ли язык, порождаемый τ_n-грамматикой?	Н
3.	Совпадает ли язык, порождаемый τ_n-грамматикой, с множеством $\overline{C}_p$?	Нет
4.	Порождают ли две τ_n-грамматики один и тот же язык?	?
5.	Является ли объединение, произведение, итерация, пересечение и дополнение $L(\tau_n)$-языков языком того же типа?	?
6.	Пусто ли пересечение языков, порождаемых τ_n-грамматиками?	?
7.	Является ли один $L(\tau_n)$-язык строгим включением другого языка того же типа?	Н
8.	Сохраняет ли конечное преобразование свойство множества быть $L(\tau_n)$-языком?	?
9.	Для любых слов c_0, c' (c' может быть пустым) выводимо ли c' из c_0 для данной τ_n-грамматики?	Н
10.	Порождает ли τ_n-грамматика регулярный, КС- или НС-язык?	?
11.	Является ли произвольное множество $U \subset \overline{C}_p L(\tau_n)$-языком?	?
12.	Существует ли гомоморфизм g такой, что $g(L(\tau_n)) = \overline{C}_p$?	Нет
13.	Существует ли гомоморфизм g^{-1} такой, что $g^{-1}(\overline{C}_p) = L(\tau_n)$?	?
14.	Является ли язык $\tau'(L(\tau_n))$ снова языком того же типа, где τ' — преобразование задано в алфавите $S^p \cup \{p-1+i\}$ $(i \geqslant 0)$?	?
15.	Является ли $L(\tau_n)$-язык формульным?	?

Обозначения: Н — проблема неразрешима (т. е. не существует алгоритма, дающего ответ на данный вопрос для любых τ_n-грамматик); Нет — ответ на вопрос будет «нет» для любых τ_n-грамматик; ? — неизвестно, существует ли разрешающий алгоритм для данной проблемы.

Будем говорить, что τ-преобразование обладает НКФ, если $(\exists c_0)$ $(\exists c'_0)(c_0 \neq c'_0 \rightarrow \tau(c_0) = \tau(c'_0))$. Пусть τ-преобразование не обладает НКФ. Но тогда, как нетрудно показать, для наших двух грамматик проблема пересечения порождаемых ими языков эквивалентна проблеме 7, которая алгоритмически неразрешима. Однако отсюда еще не следует ответ на проблему 6, так как мы исключили из рассмотрения большой класс τ_n-грамматик. Более того, если в классе τ_n-грамматик с τ-преобразованиями, не обладающими НКФ, проблема 6 неразрешима, то из этого следует, что и проблема рекурсивности $L(\tau_n)$-языков для таких τ_n-грамматик тоже неразрешима, и наоборот. Таким образом, если мы установим, что τ_n-грамматика, не имеющая НКФ, может порождать нерекурсивные языки, то проблема 6 алгоритмически неразрешима. В противном же случае вопрос остается открытым и требует для решения другого подхода. Доказательство проблемы 12 легко следует из определения гомоморфизма, теоремы 1.10 [1] и того факта, что $g(L(\tau_n)) = \{x' \mid x' = g(x),\ x \in L(\tau_n)\}$. Тогда как уже не существует алгоритма, позволяющего по произвольной КС-грамматике G узнать, существует ли гомоморфизм g такой, что $g(L(G)) = \{a, в\}^*$ [7]. В проблеме 4 мы не можем использовать метод решения проблемы 7, так как, в принципе, две грамматики $\tau_n = (n, S^p, \tau, c_0)$ и $\tau'_m = (m, S^p, \tau', c'_0)$ $(c_0 \neq c'_0)$ могут порождать один и тот же язык. Достаточно рассмотреть случай конечных $L(\tau_n)$-языков.

Представляет интерес рассмотрение обобщенных τ_n-грамматик и порождаемых ими языков. Обобщенную τ_n-грамматику можно определить следующим образом. Пусть $T^p_m = \{\tau^1_{n_1}, \tau^2_{n_2}, \ldots, \tau^m_{n_m}\}$ есть упорядоченное множество глобальных функций перехода в алфавите S^p с шаблонами соседства размера $n_i (i = \overline{1, m})$. Функцию $\Gamma = \Gamma(n)$, определенную на множестве N со значениями во множестве $\{1, 2, \ldots, m\}$, будем называть функцией управления **выводом**. Тогда T_m^{Γ}-грамматика есть упорядоченная тройка (T_m^p, Γ, c_0), где

1) T_m^p — множество глобальных функций перехода,
2) Γ — функция управления выводом,
3) c_0 — аксиома.

Языком $L(T_m^{\Gamma})$ называется множество $L = \{c_0, \ c_i = c_{i-1}\tau_{n_{\Gamma(i)}}^{\Gamma(i)} \mid i \geqslant 1\}$, где $(\forall i \geqslant 1)\,(c_i \in \overline{C}_p \,\&\, \tau_{n_{\Gamma(i)}}^{\Gamma(i)} \in T_m^p)$. Такого рода обобщенные τ_n-грамматики соответствуют полигенным однородным структурам [1].

В настоящей работе мы рассмотрели наиболее общие свойства $L(\tau_n)$-языков. В дальнейшем весьма целесообразно рассмотрение целого ряда более или менее частных примеров таких языков с целью получения более тонких свойств, так как продолжение изучения $L(\tau_n)$-языков на общем уровне представляется нам весьма затруднительным.

Из результатов нашей статьи и целого ряда других соображений мы пришли к выводу, что однородные структуры удобны для биологического моделирования на структурном уровне, тогда как, например, $<\kappa, e>$ L-системы удобнее для формального описания развивающихся биологических систем, хотя здесь и возникает целый ряд затруднений при возрастании размерности системы, которых, возможно, удастся избежать именно использованием n-мерных τ_m-грамматик. Следует отметить, что мы здесь не понимаем под структурным уровнем разного рода технические модели и реализации, использующие принцип однородности составляющих элементов (и даже их соединений в системе). Такого рода модели интересны с технической точки зрения (итеративные сети, однородные вычислительные системы, реализация на них булевых функций, проблемы различного рода минимизаций и т. п.), однако, на наш взгляд, именно формальный подход к изучению однородных структур (восходящий к работам Э. Мура, Дж. Майхилла и Х. Ямады) позволит выяснить фундаментальные свойства таких структур. Все это требует более детального исследования. В дальнейшем мы предполагаем изучить более специфические свойства $L(\tau_n)$-языков и их грамматик.

В заключении автор выражает благодарность всем участникам семинара «Теория формальных грамматик и программирования» за обсуждение настоящей работы.

ЛИТЕРАТУРА

1. Аладьев В. К., Теории однородных структур, Таллин, 1972.
2. Нейман Дж. фон, Теория самовоспроизводящихся автоматов, М., 1971.
3. Lindenmayer A., Cellular automata, formal languages and development systems, IV Intern. Congr. Logic, Method. and Phil. Sci., Bucharest, 1971.
4. Lindenmayer A., Rozenberg G., Developmental systems and languages, IV Annual ACM Symp. Theory Comput., Denver, 1972.
5. Аладьев В., Некоторые алгоритмические вопросы математической биологии, Изв. АН ЭССР, Биол., 22, № 1 (1973).
6. Гросс М., Лантен А., Теория формальных грамматик, М., 1971.
7. Гинзбург С., Математическая теория контекстно-свободных языков, М., 1970.
8. Joshi A., Kosaraju S., Yamada H., Inf. and Contr., 21, No. 2 (1972).
9. Aladyev V., Some questions concerning the nonconstructibility and computability in homogeneous structures, ENSV TA Toim., Füüs. Mat., 22, No. 2 (1973)

10. Oettinger A., Proc. Symp. Appl. Math., 12 (1961).
11. Mercer A., An array grammar programming system, Techn. Rept TR-180, Univ. Maryland, 1972.
12. Rosenfeld A., Array automata and array grammars, Techn. Rept TR-193, Univ. Maryland, 1972.

Институт экспериментальной биологии
Академии наук Эстонской ССР

Поступила в редакцию
27/XII 1972

VIKTOR ALADJEV

τ_n-GRAMMATIKAD JA NENDE POOLT GENEREERITUD KEELED

Resümee

Uuritakse uut tüüpi, nn. τ_n-grammatikaid ja nende poolt genereeritavaid keeli. τ_n-grammatika reeglid erinevad oluliselt tuntud grammatikate reeglitest. Selliste grammatikate uurimine pakub huvi homogeensete struktuuride matemaatilise teooria arendamise seisukohalt.

Eesti NSV Teaduste Akadeemia
Eksperimentaalbioloogia Instituut

Toimetusse saabunud
27. XII 1972

VICTOR ALADYEV

τ_n-GRAMMARS AND LANGUAGES GENERATED BY THEM

Summary

In this paper a new style of formal grammar called τ_n-grammars has been studied. The rules in τ_n-grammars have a character essentially different from the "rewrite rule" in known grammars. Such a study of τ_n-grammars is of interest for following the mathematical theory of homogeneous structures.

Academy of Sciences of the Estonian SSR,
Institute of Experimental Biology

Received
Dec. 27, 1972