Доклады Академии наук СССР
1974. Том 217, № 5

УДК 519.95 *МАТЕМАТИКА*

А. Н. МАСЛОВ

ИЕРАРХИЯ ИНДЕКСНЫХ ЯЗЫКОВ ПРОИЗВОЛЬНОГО УРОВНЯ

(Представлено академиком А. Н. Колмогоровым 30 I 1974)

Строится последовательность классов языков, являющихся расширением класса контекстно-свободных (к.с.) языков ([1,2]). Второй член последовательности — класс индексных языков ([3]). Класс конечноавтоматных языков является нулевым членом. Ряд фрагментов языков программирования, не являющихся контекстно-свободными, принадлежит второму и третьему классу из этой последовательности. Класс с номером $i+1$ может быть задан четырьмя способами: 1) порождающей грамматикой с индексами уровня не выше i; 2) распознающим недетерминированным автоматом с памятью уровня $i+1$, состоящей из магазинного списка памятей уровня i; 3) как наименьшее решение специальной системы уравнений с некоммутативными переменными; 4) макрограмматикой глубины i.

Некоторые свойства введенных классов легко получаются при рассмотрении одного способа задания и не столь очевидны с точки зрения других заданий.

1. Введем формальную операцию возведения в степень для букв A и B алфавита V. Множество всех изображений A^B обозначим через V^V. Доопределим операцию возведения в степень до операции над языками с помощью равенств:

$$(L_1\cup L_2)^L=L_1^L\cup L_2^L,\quad (L_1L_2)^L=L_1^LL_2^L,\quad A^{(L_1\cup L_2)}=A^{L_1}\cup A^{L_2},\quad A^{BL}=(A^B)^L,$$

где A и B — символы из алфавита V, а L, L_1 и L_2 — языки в алфавите V. Указанные равенства позволяют по интерпретации для обозначений A^B (т. е. по отображению $V^V\to V$) определить язык $L_1^{L_2}$ для языков L_1 и L_2 в том же алфавите V. Тем самым, в частности, определена операция возведения в степень для слов.

В изображении A^B букву B будем называть индексом буквы A. Теперь можно определить класс индексных грамматик, используя несколько более удобные обозначения, чем в ([3]).

Определение. Индексной грамматикой называется пятерка $G=(\Sigma, V, P, I, S_0)$, где $\Sigma=\{\sigma_1,\ldots,\sigma_d\}$ — терминальный алфавит, $V=\{S_0, S_1,\ldots, S_m\}$ — нетерминальный алфавит, S_0 — начальный символ, P — конечное множество продукций вида $S_i\to\omega$, $\omega\in(V^V\cup V\cup\Sigma)^*$ и I: $V^V\to V$ — интерпретация операции возведения в степень.

Слово y выводимо из слова x применением продукции $p\in P$ (обозначается $x\overset{p}{\vdash}y$), если $x=aS_i{}^zb$, $y=a\omega^zb$, p: $S_i\to\omega$, слова a, b и z в произвольном алфавите. Слово y выводимо из слова x, обозначается $x\vdash y$, если существует последовательность слов $x=y_0, y_1,\ldots, y_n=y$ такая, что либо $y_i\overset{p}{\vdash}y_{i+1}$ для $p\in P$, либо y_{i+1} получается из y_i заменой некоторого вхождения A^B на $I(A^B)$. Каждая грамматика G определяет язык $L(G)=\{x\in\Sigma^*|S_0\vdash x\}$ выводимых в ней слов.

Определение выводимости в индексной грамматике допускает естественное расширение. Пусть $V(1)=V^*$, $V(2)=(V^{V(1)})^*$, $V(k)=\{A^x|A\in V, x\in V(k-1)\}^*$ и $V(\infty)=\bigcup\limits_k V(k)$. Например, $A^{BC^{DC}B^F}\in V(3)$ и $A_1{}^{A_2\ldots A_k}\in V(k)$.

Слово y выводимо из слова x применением правила $p \in P$ на уровне $k > 1$ (обозначается $x \overset{p,k}{\vdash} y$), если существуют слова a_i, b_i, z_i и t_i такие, что $x = a_0 S_{i_0}{}^{z_0} b_0$, $y = a_0 S_{i_0}^{t_0} b_0$, $z_j = a_{j+1} S_{i_{j+1}}^{z_{j+1}} b_{j+1}$, $t_j = a_{j+1} S_{i_{j+1}}^{t_{j+1}} b_{j+1}$ при $1 < j < k-1$, $z_{k-1} = a_k S_{i_k}^{z_k} b_k$, $t_{k-1} = a_k \omega^{z_k} b_k$, $p : S_{i_k} \to \omega$.

Будем отождествлять выводимость $\overset{p}{\vdash}$ и выводимость $\overset{p\,1}{\vdash}$. Неформально, выводимость на уровне k обозначает применение продукции к нетерминалу, являющемуся индексом k-го уровня. Слово y выводится из слова x на уровне k (обозначается $\overset{k}{\vdash}$), если существует последовательность слов $x = y_0$, $y_1, \ldots, y_n = y$, $y_i \in V(k)$, такая, что либо $y_i \overset{p,k'}{\vdash} y_{i+1}$ для $p \in P$ и $k' \leqslant k$, либо y_{i+1} получается из y_i заменой обозначения A^B на $I(A^B)$ на максимальном уровне, т. е. B не имеет индекса. Ясно, что $\overset{2}{\vdash}$ и $\vdash$ совпадают.

Язык $L_k(G) = \{x \in \Sigma^* \mid S_0 \overset{k}{\vdash} x\}$, образованный из слов, выводимых на уровне k, назовем индексным языком уровня $k+1$, а язык $L_\infty(G) = \{x \in \Sigma^* \mid \exists k (S_0 \overset{k}{\vdash} x)\}$ — индексным языком уровня ∞.

Замечания. Допустив в качестве интерпретации отображения I: $V^V \to (V(1) \cup \Sigma)^*$ мы не изменим классы языков $L_k(G)$. Также не изменяет эти классы доопределение тупиковых изображений, включающих терминалы, с помощью равенств $\sigma^z = \sigma$.

Отметим следующие результаты. Класс языков L_∞ совпадает с классом рекурсивно-перечислимых языков. Для индексных грамматик уровня k, $k < \infty$, разрешима проблема пустоты порождаемого языка.

2. Класс контекстно-свободных языков совпадает с классом языков, допускаемых недетерминированным магазинным автоматом [1, 2]. Аналогичную роль для индексных языков уровня $k+2$ играют магазинные автоматы уровня $k+2$.

Магазинный автомат (с одним состоянием) можно понимать как левый вывод в к.с. грамматике, продукции которой приведены к виду $A \to BC$ или $A \to \sigma$. В этом случае нетерминальная часть промежуточного слова при левом выводе можно рассматривать как содержимое магазина (точнее, магазинного списка). Продукции индексной грамматики можно привести к виду $A \to B^C$, $A \to BC$ или $A \to \sigma$. Промежуточное слово при левом выводе имеет вид $\sigma_{j_1} \ldots \sigma_{j_n} A_{i_1}{}^{z_1} \ldots A_{i_m}{}^{z_m}$, $z_i \in V(\infty)$. Нетерминальная часть должна естественным образом располагаться в памяти допускающего автомата.

Память уровня 1 — это магазин. Автомат может совершать с магазином следующие действия: 1) прочитать очередной символ на входной ленте и символ в вершине магазина, а затем стереть последний; 2) прочитать и заменить на два символа верхний символ в магазине.

Память уровня k — это магазинный список пар: активный символ — память уровня $k-1$. В частности, память уровня 2 — это магазинный список пар: активный символ — магазинный список символов. Автомат работает с вершинной парой, а при ее опустошении переходит к следующей паре. Автомат может совершать (недетерминированно) следующие действия, возможность выполнения которых зависит от читаемых символов (символ на входной ленте не читается, если не указано обратное): 1) если память уровня $k-1$ в вершинной паре пуста, то прочитать очередной символ на входной ленте и активный символ, а затем стереть вершинную пару; 2) прочитать активный символ и заменить вершинную пару на две, отличающихся от первоначальной лишь активным символом; 3) прочитать активный символ и изменить его, добавив при этом вершинный символ в память уровня $k-1$ (т. е. в вершину магазина уровня $k-1$ добавить пару «некоторый символ — пустая память уровня $k-2$»; 4) прочитать активный символ (вершинной пары) и активный символ ее памяти уровня $k-1$, если при этом память уровня $k-2$ (в вершине памяти уровня $k-1$) пуста, то стереть вершинную пару в памяти уровня $k-1$ и изменить прочитанный активный символ; 5) совершить одно из действий 2—5 с памятью уровня $k-1$, находящейся в вершинной паре.

В начальный момент в магазине хранится единственная пара «начальный символ — пустая память уровня $k-1$». Если, прочитав слово, автомат опустошил память, слово считается допустимым. Ясно, что левый вывод в грамматике уровня k и допустимость соответствующим автоматом (с одним состоянием), использующим память уровня $k+1$, тождественны. Мы не будем рассматривать подкласс детерминированных автоматов и заниматься обсуждением удобств описанной памяти при программной реализации.

Мы не расширим класс допустимых языков, если предположим наличие нескольких (но конечного числа) состояний у автомата.

Из этого следует, что семейство языков, допускаемое автоматами с памятью уровня k, является ([8]) полным абстрактным семейством языков (а.с.я.), которое порождено одним языком с помощью операций, участвующих в определении полного а.с.я. Отметим, что гнездный стэк ([4]) получается из автомата с памятью уровня 2 отождествлением совпадающих начал у составляющих магазинов.

3. Язык L_∞ может быть определен как наименьшее решение системы уравнений $S=\{A=\sum\limits_l \omega_l \mid (A\to\omega_l)\in P$ или $\omega_l=C^{\bar{B}}$, $I(A^B)=C\}$, где $\bar{B}B\cap\Sigma^*=\varepsilon$ (ε — пустое слово), с некоммутативными переменными ([7]). Символ $\bar{B}$ является левым обратным для B. Чтобы выделить системы уравнений, задающие подкласс L_k, достаточно заменить алфавит V на алфавит $\bigcup\limits_{i=0}^{k} V_i$, где $V_i=\{A_i \mid A\in V\}$, а уравнения S — на уравнения $S'=\{A_i=\sum\limits_l \omega_{l,i}' \mid 0\leqslant i<k$, $(A=\sum\limits_i \omega_l)\in S$; $\omega_{l,i}'$ получено из ω_l заменой всех нетерминалов B на j-ом уровне на B_{i+j} (см. также ([6])).

4. Другим способом задания индексных знаков уровня k являются макрограмматики. Поставим в соответствие каждому терминальному символу σ функциональный (от одной переменной) терминальный символ $\sigma(\ )$, а каждой цепочке $\sigma_{i_1}\sigma_{i_2}\ldots\sigma_{i_m}$ — формулу $\sigma_{i_1}(\sigma_{i_2}(\ldots(\sigma_{i_m}(\varepsilon))\ldots))$. Множества формул, соответствующих цепочкам описанных выше языков, порождаются макрограмматиками. Вариант макрограмматик, порождающих класс с номером 2 (индексные языки), изучен в ([5]).

Будем использовать $\mathbf{X}$ в качестве сокращения для $X_1,\ldots,X_n$. Пусть имеется набор нетерминальных функциональных символов F_i, которым могут быть приданы параметры $F_i[\mathbf{p}]$, переменные $F_i(\mathbf{x})$ или и параметры и переменные одновременно $F_i[\mathbf{p}](\mathbf{x})$; размерность векторов $\mathbf{p}$ и $\mathbf{x}$ однозначно определяется символом F_i. Символы p_i и x_i являются элементарными формулами. Остальные формулы получаются из элементарных, формулы ε (пустое слово) и терминальных и нетерминальных функциональных символов — подстановкой формул вместо параметров и переменных. Макрограмматика задается начальным символом F_0 и набором правил вида $F[\mathbf{p}](\mathbf{x})\to\Phi\{\mathbf{p}\}(\Psi_1\{\mathbf{p}\}(\mathbf{x}),\ldots,\Psi_m\{\mathbf{p}\}(\mathbf{x}))$ (Φ и Ψ_i — формулы, а фигурные скобки обозначают, что элементарными подформулами в Φ и Ψ_i являются только p_j, причем p_j находятся на глубине $\leqslant 1$ относительно вложенности квадратных скобок), а также правил аналогичного вида, в которых параметры и (или) переменные отсутствуют, в частности, правил вида $F(\mathbf{x})\to x_i$.

Если некоторая формула имеет подформулу $F[\chi](\theta)$, то, при наличии правила $F[\mathbf{p}](\mathbf{x})\to\Phi\{\mathbf{p}\}(\mathbf{\Psi}\{\mathbf{p}\}(\mathbf{x}))$, эта подформула может быть заменена на $\Phi\{\chi\}(\mathbf{\Psi}\{\chi\}(\theta))$, причем: 1) фигурные скобки обозначают, что в Φ и Ψ_i вместо p_j всюду подставлены формулы χ_j; 2) замену можно осуществлять только в том случае, когда в результате замены получается формула, т. е. размерности векторов переменных и параметров соответствуют функциональным символам, которым они приданы; 3) правило, для которого $\Phi\{\mathbf{p}\}=p_i$, можно применять только в том случае, когда $\chi_i=H$ или $\chi_i=H(\Gamma)$, в последнем случае осуществляется приведение $H(\Gamma)$ к $H[\Gamma]$.

Множество цепочек, соответствующих терминальным формулам, выводимым из F_0 с помощью описанных замен и аналогичных замен при отсутствии параметров и (или) переменных, называется порождаемым языком.

Макрограмматики общего вида порождают класс языков L_∞. Макрограмматики, не использующие параметры, порождают класс контекстно-свободных языков. Фиксируем число k. Если в каждом выводе макрограмматики глубина вложенности квадратных скобок или, эквивалентно, глубина приведений, не превышает k, то макрограмматика имеет глубину k. Можно показать, что макрограмматика глубины k порождает класс языков с номером k.

Подробное изложение результатов будет опубликовано в журнале «Проблемы передачи информации».

Автор признателен А. А. Мучнику за поддержку в работе и Э. Д. Стоцкому, В. М. Ломковской и А. Н. Колодию за полезное обсуждение.

Поступило
9 I 1974

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

[1] *С. Гинзбург,* Математическая теория контекстно-свободных языков, М., 1970. [2] *А. В. Гладкий,* Формальные грамматики и языки, «Наука», 1973. [3] *A. V. Aho,* J. Assoc. Comp. Machinery, v. 15, № 4, 647 (1968). [4] *A. V. Aho,* J. Assoc. Comp. Machinery, v. 16, № 3, 383 (1969). [5] *M. J. Fischer,* Report No. NSF-22, The Computational Laboratory of Harvard University, 1968. [6] *O. Mayer,* Lect. Notes Comp. Sci., v. 2, 166 (1973). [7] *A. Blikle,* Information and Control, v. 21, № 2, 134 (1972). [8] *S. Ginsburg, S. A. Greibach,* Mem. Am. Math. Soc., № 87, 1 (1969).