

Описание формальных языков L-графами

А.А. Вылиток

Аннотация—Теория формальных языков, активно развивающаяся уже седьмой десяток лет, содержит задачи, для которых до сих пор решения либо не найдены, либо они не пригодны для построения на их основе практических методов. Трудности обусловлены сложностью анализа протоколов вычислений и выводов таких классических описаний, как машины Тьюринга, магазинные автоматы и др. В данной работе рассматривается способ представления формальных языков в виде L-графов, дающих более удобную для анализа понятийную среду, позволяющую привлекать эффективные методы теории графов к решению задач теории формальных языков, в частности, к оптимизации формальных описаний с помощью приёмов, известных для конечных автоматов (последние можно считать наиболее простыми представителями L-графов). L-графы без ограничений характеризуют класс рекурсивно-перечислимых языков, то есть языков типа 0 по Хомскому. Другими словами, любой рекурсивно-перечислимый язык может быть задан L-графом, и наоборот, язык, задаваемый L-графом, является рекурсивно-перечислимым. Описываются также подклассы L-графов, характеризующие бесконечные (контекстно-свободные) и регулярные языки.

Ключевые слова—формальный язык, способы описания формального языка, теория формальных языков, L-граф, обобщение диаграммы конечного автомата

1. Введение

Среди описаний формальных языков графами наиболее известны диаграммы состояний для конечных автоматов, — они часто используются в учебниках и монографиях (например, [1]-[2]) как иллюстрации автоматов. На рисунке 1 приведен пример такой диаграммы для автомата $\mathcal{A} = \langle \{1, 2, 3\}, \{a, b\}, \delta, \{1\}, \{3\} \rangle$, где $\delta = \{(1, a, 1), (1, b, 2), (2, a, 2), (2, b, 3), (3, a, 3)\}$ — функция переходов в виде множества. Представленная диаграмма — это ориентированный размеченный граф, вершины которого помечены состояниями из множества $\{1, 2, 3\}$, дуги помечены символами из алфавита входных символов $\{a, b\}$ и соответствуют переходам между состояниями, определяемым множеством переходов δ . Для наглядности вершины с начальным состоянием помечаются короткой входящей стрелкой, а вершины с заключительным состоянием — короткой исходящей стрелкой.

Заметим, что можно рассматривать граф $\mathcal{A}$ на рисунке 1 как способ задания формального языка в алфавите $\{a, b\}$ без исходной автоматной интерпретации. Действительно, рассмотрим несколько путей из начальной вершины в заключительную:

$\rightarrow 1 \xrightarrow{b} 2 \xrightarrow{b} 3 \rightarrow$, $\rightarrow 1 \xrightarrow{a} 1 \xrightarrow{b} 2 \xrightarrow{b} 3 \rightarrow$, $\rightarrow 1 \xrightarrow{b} 2 \xrightarrow{a} 2 \xrightarrow{b} 3 \rightarrow$,
 $\rightarrow 1 \xrightarrow{b} 2 \xrightarrow{b} 3 \xrightarrow{a} 3 \rightarrow$ с пометками путей bb, abb, bab, bba .
 Эти цепочки принадлежат задаваемому графом $\mathcal{A}$ языку

$L(\mathcal{A})$, состоящему из пометок всех успешных путей, то есть путей из начальной вершины в заключительную.

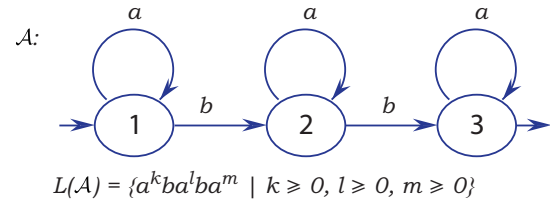


Рис. 1. Граф $\mathcal{A}$, представляющий конечный автомат

Нетрудно видеть, что $L(\mathcal{A}) = \{a^k b a^l b a^m \mid k, l, m \geq 0\}$.

Теперь разрешим на каждой дуге графа располагать помимо основной символьной пометки пару вспомогательных скобочных пометок (граф $\mathcal{B}$ на рис. 2). Для удобства различения первой и второй скобочных пометок на рисунке первая изображается как прямоугольная скобка, а вторая как угловая. Каждая из скобочных пометок может отсутствовать. Например, на дуге $2 \xrightarrow{b} 2$ есть только первая скобочная пометка, на дуге $3 \xrightarrow{c} 3$ только вторая, а на дуге $1 \rightarrow 2$ ни одной. Заметим, что на дуге $1 \rightarrow 2$ отсутствует и основная символьная пометка. Такие дуги являются аналогами переходов без чтения входного символа в автоматах (ε -переходов).

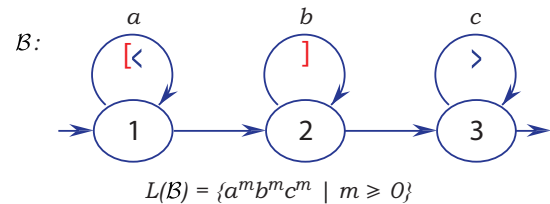


Рис. 2. Граф $\mathcal{B}$ со вспомогательными скобочными пометками

Для определения языка, задаваемого графом с дополнительными скобочными пометками, потребуем, чтобы в успешном пути цепочка из первых скобочных пометок этого пути (назовем ее первым следом пути) являлась бы правильной скобочной последовательностью, и, независимо от первого следа, цепочка из вторых скобочных пометок (второй след) тоже являлась бы правильной. Понятие правильной скобочной последовательности можно пояснить так: если в корректном арифметическом выражении удалить все символы кроме скобок, то результатом будет правильная скобочная последовательность (возможно пустая). Например, выражению $(a \times (b+c)) \times (d-e)$ соответствует скобочная система $((()))()$. Рассмотрим путь $\rightarrow 1 \xrightarrow{a} 1 \xrightarrow{b} 2 \xrightarrow{b} 2 \xrightarrow{c} 3 \rightarrow$. Его первый след $[[]]$ является правильной последовательностью, и второй след $\langle \rangle$ — тоже правильный. Поэтому рассматриваемый путь успешный,

а его основная пометка abc входит в задаваемый графом $\mathcal{B}$ язык. Можно заметить, что для баланса скобок в следах успешных путей графа $\mathcal{B}$ его основная пометка должна содержать одинаковое количество символов a , b и c . Следовательно, $L(\mathcal{B}) = \{a^n b^n c^n \mid n \geq 0\}$.

Один и тот же язык можно описать различными графами. Так, граф $\mathcal{C}$ на рисунке 3 эквивалентен графу $\mathcal{B}$, поскольку $L(\mathcal{B}) = L(\mathcal{C})$, то есть их языки равны.

Представление формальных языков L-графа позволяет обобщать некоторые алгоритмы обработки конечных автоматов на более общие случаи, упрощать доказательства известных фактов и обнаруживать новые, привлекая развитый математический аппарат теории графов.

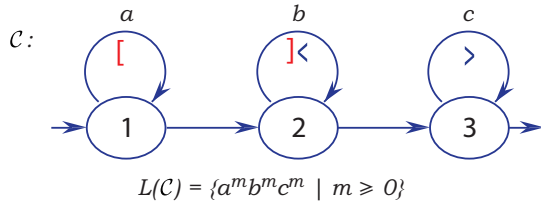


Рис. 3. Эквивалентный граф $\mathcal{C}$: $L(\mathcal{C}) = L(\mathcal{B})$

II. L-ГРАФ

Рассматриваемое здесь графовое представление формальных языков обобщает на случай рекурсивно-перечислимых языков подход к исследованию бесконтактных (контекстно-свободных) языков, описанный в [3]–[6].

Префикс L в названии L-граф происходит от латинского слова Lingua — язык. Можно интерпретировать название и по-английски как Language graph, то есть граф, задающий язык. Первое сообщения об L-графах появилось в [7].

A. Скобочные множества и последовательности

Приведем точные определения некоторых связанных со скобочными последовательностями понятий.

Определение 1: Пусть P_{left} и P_{right} — два непересекающихся алфавита и пусть существует биекция $\phi : P_{left} \rightarrow P_{right}$. Назовем тройку $\langle P_{left}, P_{right}, \phi \rangle$ *скобочным множеством*.

Назовем элементы множества P_{left} *открывающими* скобками, а элементы множества P_{right} — *закрывающими* скобками.

Пусть P — скобочное множество, $P = \langle P_{left}, P_{right}, \phi \rangle$. Введем следующие понятия и обозначения, связанные с P :

$P_{\langle}$ — множество открывающих скобок, $P_{\langle} = P_{left}$;
 $P_{\rangle}$ — множество закрывающих скобок, $P_{\rangle} = P_{right}$;
 $P_{\langle \rangle}$ — множество открывающих и закрывающих скобок, $P_{\langle \rangle} = P_{\langle} \cup P_{\rangle}$;
 P_{pair} — это множество пар скобок, $P_{pair} = \{ (a, b) \mid a \in P_{left}, b \in P_{right}, b = \phi(a) \}$.

Ниже, там где это не вызовет недоразумений, под скобочным множеством будем подразумевать множество скобок $P_{\langle \rangle}$.

Для обозначения пустой цепочки используется ε .

Определение 2: Язык, порождаемый бесконтекстной грамматикой с начальным символом S и правилами вида

$$S \rightarrow a_i S b_i S \mid \varepsilon, \quad (1)$$

где $(a_i, b_i) \in P_{pair}$, $i = 1, \dots, |P_{\langle \rangle}|$, назовем *языком Дика над скобочным множеством P* и обозначим его $D(P)$.

Цепочки в алфавите $P_{\langle \rangle}$ будем называть *скобочными последовательностями*. Скобочные последовательности, принадлежащие языку Дика, будем называть *правильными*.

Из определений следует, что если $\alpha \in D(P)$, $\beta \in D(P)$, $(a, b) \in P_{pair}$, то $\alpha\beta \in D(P)$, $a\alpha b \in D(P)$, $a\alpha b\beta \in D(P)$. Пусть $\gamma = \beta'\beta\beta''$, $\alpha \in D(P)$. Тогда если $\beta'\alpha\beta'' \in D(P)$, то $\gamma \in D(P)$.

Скобки $a \in P_{\langle}$ и $b \in P_{\rangle}$ будем называть *парными в последовательности γ* , если $(a, b) \in P_{pair}$, $\gamma = \alpha a \gamma_0 b \beta$, $\gamma_0 \in D(P)$, $\alpha, \beta \in P_{\langle \rangle}^*$.

Заметим, что в правильной скобочной последовательности скобки a_i и b_i , полученные при выводе применением правила $S \rightarrow a_i S b_i S$, являются парными в этой последовательности.

В правильной скобочной последовательности каждая скобка имеет единственную парную к ней в этой последовательности.

Пример 1: Пусть $P = \langle \{ [, [_2 \}, \{] ,] _2 \}, \phi \rangle$ — скобочное множество, $\phi([_1]) =]_1$, $\phi([_2]) =]_2$.

Рассмотрим цепочку из квадратных скобок $\pi = [_1 [_1 [_2 [_2] _1 [_1 [_1] _1$, $\pi \in D(P)$. Первая скобка в ней парна восьмой, вторая парна пятой, третья — четвертой, шестая скобка парна седьмой, вторая и седьмая скобки парными не являются.

Множество пар: $P_{pair} = \{ ([_1,] _1), ([_2,] _2) \}$.

Скобочное множество: $P = \langle \{ [_1, [_2 \}, \{] _1,] _2 \}, \phi \rangle$, $\phi([_1]) =]_1$, $\phi([_2]) =]_2$.

B. Определение L-графа

Определение 3: L-граф задается шестеркой

$$G = \langle V, \Sigma, P, E, I, F \rangle, \quad (2)$$

где:

V — конечное множество вершин;

Σ — алфавит языка;

P — скобочное множество;

$I \subseteq V$ — множество начальных вершин;

$F \subseteq V$ — множество заключительных вершин;

$E \subseteq V \times (\Sigma \cup \{\varepsilon\}) \times (P \cup \{\varepsilon\}) \times (P \cup \{\varepsilon\}) \times V$ — множество дуг.

Пусть $\pi \in E$, $\pi = \langle q, a, d_1, d_2, p \rangle$.

Введем вспомогательные понятия и обозначения, связанные с дугой π :

$beg(\pi)$ — начальная вершина дуги, $beg(\pi) = q$;

$end(\pi)$ — заключительная вершина дуги, $end(\pi) = p$;

$\omega(\pi)$ — пометка дуги, $\omega(\pi) = a$;

$\iota_1(\pi)$ — первый(скобочный) след, $\iota_1(\pi) = d_1$;

$\iota_2(\pi)$ — второй(скобочный) след, $\iota_2(\pi) = d_2$.

Пусть $T = p_0 \pi_1 p_1 \dots \pi_k p_k$, $\pi_i \in E$, $i = 1, \dots, k$, $p_i \in V$, $i = 0, \dots, k$, $beg(\pi_i) = p_{i-1}$, $end(\pi_i) = p_i$.

Для $i = 0$ путь T назовем пустым. Он представляется одной вершиной. Непустой путь T однозначно определяется своими дугами и его можно обозначить

$T = \pi_1 \pi_2 \dots \pi_k$. Далее путь T будем определять либо вершиной: $T = p_0$ для пустого пути; либо последовательностью дуг: $T = \pi_1 \dots \pi_k$ для непустого пути.

Длина пути — это количество дуг в пути, обозначается $|T|$.

Начальная вершина пути (обозначение $beg(T)$) — вершина, из которой исходит первая дуга пути, $beg(T) = p_0$.

Конечная вершина пути (обозначение $end(T)$) — вершина, в которую входит последняя дуга пути, $end(T) = p_k$.

Рекурсивно определим понятия пометки пути $\omega(T)$ и i -го следа пути $\iota_i(T)$, $i = 1, 2$.

Для $T = p$ положим $\omega(T) = \varepsilon$, $\iota_i(T) = \varepsilon$.

Для $T = e$, $e \in E$, $e = (q, x, \beta_1, \beta_2, p)$ положим $\omega(T) = x$, $\iota_i(T) = \beta_i$, $i = 1, 2$.

Для $T = T_0 e$, $e \in E$, $e = (q, x, \beta_1, \beta_2, p)$ положим $\omega(T) = \omega(T_0)x$, $\iota_i(T) = \iota_i(T_0)\beta_i$, $i = 1, 2$.

Путь T называется *успешным*, если $beg(T) \in I$, $end(T) \in F$, $\iota_i(T) \in D(P)$, $i = 1, 2$.

Определение 4: Языком, определяемым L-графом $G = \langle V, \Sigma, P, E, I, F \rangle$, назовем множество пометок его успешных путей и обозначим $L(G)$: $L(G) = \{x \in \Sigma^* \mid \exists T : x = \omega(T), beg(T) \in I, end(T) \in F, \iota_i(T) \in D(P), i = 1, 2.\}$

L-графы без дополнительных ограничений характеризуют класс рекурсивно-перечислимых языков (т.е. класс языков типа 0 по Хомскому) [7]. Как известно, автомат с конечным числом состояний и двумя магазинами (по-другому, машина с двумя стеками) по своим возможностям эквивалентен машине Тьюринга, также характеризующей рекурсивно-перечислимые языки [8]-[9]. L-граф можно интерпретировать как автомат с двумя магазинами: переход по дуге в следующую вершину означает смену состояния, открывающая скобочная пометка дуги при этом заносится в соответствующий магазин (один магазин для первого скобочного следа дуги, другой для второго), закрывающая скобочная пометка приводит к удалению с вершины магазина парной ей открывающей скобки.

В последующих публикациях будет приведено «грамматическое» обоснование: существует алгоритм построения по заданному L-графу эквивалентной грамматики типа 0 по Хомскому и обратный алгоритм, преобразующий грамматику типа 0 в эквивалентный L-граф.

Определение 5: Пусть $G = \langle V, \Sigma, P, E, I, F \rangle$ — L-граф. Будем называть его *бесконтекстным*, если для любого $\pi \in E$ $\iota_2(\pi) = \varepsilon$.

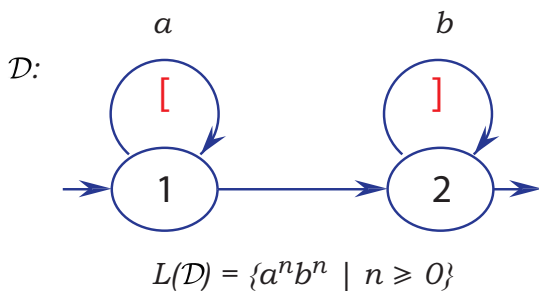


Рис. 4. Бесконтекстный L-граф $\mathcal{D}$

Пример 2: Бесконтекстный язык, заданный формулой $L(\mathcal{D}) = \{a^n b^n \mid n \geq 0\}$, может быть представлен графом, изображенным на рисунке 4.

Бесконтекстные L-графы характеризуют класс бесконтекстных (контекстно-свободных) языков. Бесконтекстный L-граф можно интерпретировать как автомат с одним магазином (для первого скобочного следа), поскольку второй скобочный след дуги всегда пуст и магазин для него не требуется. Магазиновые автоматы, как известно, характеризуют класс бесконтекстных языков [9].

Существует алгоритм построения по бесконтекстной грамматике эквивалентного бесконтекстного L-графа, и обратный алгоритм построения по бесконтекстному L-графу эквивалентной бесконтекстной грамматики. Эти алгоритмы будут представлены в других публикациях. Еще одно представление бесконтекстных языков с помощью графов описано в [10].

Определение 6: Граф назовем *регулярным*, если для любого $\pi \in E$ $\iota_1(\pi) = \varepsilon$, $\iota_2(\pi) = \varepsilon$.

Пример 3: Приведенный на рисунке 5 регулярный граф задает язык $L(\mathcal{R}) = \{a^n b^m \mid n \geq 0, m > 0\}$.

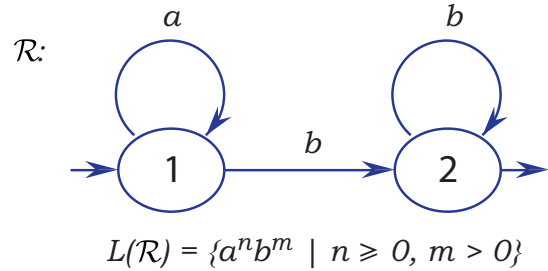


Рис. 5. Регулярный граф $\mathcal{R}$

С. Ядро бесконтекстного L-графа

Ядро — это, по сути, конечное подмножество множества всех путей бесконтекстного L-графа, исследование которого достаточно для распознавания многих свойств: пуст ли язык, конечен или бесконечен, линейный или нет и т.п. Рассматривается параметризованное понятие (w, d) -ядра, где параметры w и d обеспечивают определенные ограничения на повторяемость циклов в путях L-графа. Параметр w ограничивает количество повторений сбалансированных по скобкам циклов, представляя английское слово *width*, а параметр d ограничивает количество повторений согласованных пар циклов с вложением скобок вида $(a, b) \in P_{pair}$, представляя английское слово *depth*.

Пусть G — L-граф. Назовем *маршрутом* любой фрагмент успешного пути L-графа G .

Пусть T — непустой маршрут и $beg(T) = end(T)$. Тогда назовем маршрут T циклом.

Введем отображение

$$\mu : P_{\diamond}^* \rightarrow P_{\diamond}^*.$$

Пусть w — произвольная цепочка из $P_{\diamond}^*$, тогда определим отображение μ следующим образом: $\mu(w) = t$, где t есть результат применения к цепочке w следующих действий:

1) выполнить присваивание $t := w$;

2) пока существует такая пара скобок $(a, b) \in P_{pair}$, что $t = t_1abt_2$, выполнять присваивание $t := t_1t_2$, где t_1 и t_2 — скобочные последовательности.

Цикл T называется *нейтральным*, если $\mu(\iota_i(T)) = \varepsilon, i = 1, 2$.

Пусть маршрут $T = T_0\pi_1T_1T_2T_3\pi_3T_4$ таков, что π_1 и π_3 являются дугами и

$$\mu(\iota_i(T_2)) = \mu(\iota_i(T_1T_2T_3)) = \mu(\iota_i(\pi_1T_1T_2T_3\pi_3)) = \varepsilon, i = 1, 2.$$

Тогда назовем тройку $(\pi_1T_1, T_2, T_3\pi_3)$ *гнездом* маршрута T . Будем говорить, что участки $\pi_1T_1T_3\pi_3$ парны друг другу (т. е. образуют гнездо в маршруте T).

Пусть T — некоторый успешный путь бесконтекстного L-графа L , w и d — положительные целые числа. Пусть T удовлетворяет следующим условиям: если $T_1, \dots, T_m$ — такие нейтральные циклы, что цикл $T_1, \dots, T_m$ является маршрутом пути T , то $m \leq w$; если $T_{11}, \dots, T_{1m}, T_{21}, T_{31}, \dots, T_{3m}$ таковы, что $T_{2(i+1)} = T_{1i}T_{2i}T_{3i}, 1 \leq i \leq m$, есть гнездо маршрута T , образованное циклами T_{1i} и T_{3i} , то $m \leq d$. Тогда назовем $T(w, d)$ -каноническим бесконтекстного L-графа L .

Множество (w, d) -канонических бесконтекстного L-графа L назовем (w, d) -ядром бесконтекстного L-графа L и обозначим через $Core(L, w, d)$.

Пример 4: Для бесконтекстного графа $\mathcal{F}$ (рис. 6), задающего язык $L(\mathcal{F}) = \{a^nbc^na^m \mid n \geq 0, m \geq 0\}$,

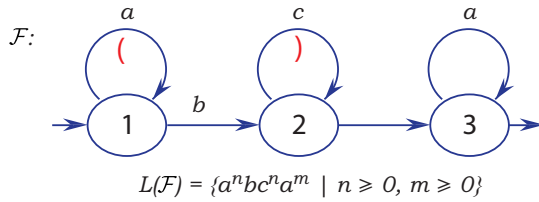


Рис. 6. Бесконтекстный L-граф $\mathcal{F}$

ядро при $w = 1, d = 1$ таково:

$$Core(L, 1, 1) : \begin{cases} \rightarrow 1 \xrightarrow{b} 2 \rightarrow 3 \rightarrow \\ \rightarrow 1 \xrightarrow{b} 2 \rightarrow 3 \xrightarrow{a} 3 \rightarrow \\ \rightarrow 1 \xrightarrow{a} 1 \xrightarrow{b} 2 \xrightarrow{c} 2 \rightarrow 3 \rightarrow \\ \rightarrow 1 \xrightarrow{a} 1 \xrightarrow{b} 2 \xrightarrow{c} 2 \rightarrow 3 \xrightarrow{a} 3 \rightarrow \end{cases}$$

Для бесконтекстных L-графов справедливо утверждение (аналогичное для D-графов из [5]) о развитии всех успешных маршрутов из элементов ядра. Действительно, все успешные маршруты для $\mathcal{F}$ можно получить, вставляя в исходные элементы ядра нужное количество раз нейтральный цикл $3 \xrightarrow{a} 3$ и согласованную пару циклов $1 \xrightarrow{a} 1$ и $2 \xrightarrow{c} 2$.

III. Заключение

Представление формальных языков с помощью L-графов дает возможность использовать эффективные алгоритмы теории графов (поиск кратчайших путей, выделение сильно связанных компонент и др.) для анализа структурных свойств языков, получать более простые и наглядные доказательства известных фактов и обнаруживать новые, строить практические алгоритмы обработки формальных языков (распознавание свойств, синтаксический анализ, трансляция) и обосновывать их корректность с помощью анализа путей в графе.

Список литературы

- [1] Серебряков В.А. Теория и реализация языков программирования. — М. : Наука, 2012. — 236 с.
- [2] Мельников Б.Ф. Недетерминированные конечные автоматы. — Тольятти : Изд-во ТГУ, 2009. — 160 с.
- [3] Станевичене Л. И. Об одном средстве исследования бесконтекстных языков // Кибернетика. — 1989. — № 4. — С. 135–136.
- [4] Вылиток А. А. О построении графа магазинного автомата // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. — 1996. — № 3. — С. 68–73.
- [5] Станевичене Л. И. О некоторых определениях класса КС-языков // Программирование. — 1999. — № 5. — С. 15–25.
- [6] Станевичене Л. И. К теории бесконтекстных языков. — М., 2000. — Деп. в ВИНТИ РАН 29.05.2000, №1546-B00.
- [7] Вылиток А. А., Сутырин П. Г. Характеризация формальных языков графами // Сборник тезисов научной конференции «Тихоновские чтения». — Москва, МГУ имени М. В. Ломоносова, факультет ВМК, 2010. — С. 82–83.
- [8] Минский М. Вычисления и автоматы. — М. : Мир, 1971. — 364 с. — Пер. с англ.: Minsky M. L. Computation: finite and infinite machines.
- [9] Хопкрофт Д. Э., Мотвани Р., Ульман Д. Д. Введение в теорию автоматов, языков и вычислений / Под ред. А. Б. Ставровский. — 2 изд. — М. : Издательский дом «Вильямс», 2008. — 528 с. — Пер. с англ.: Hopcroft J. E., Motwani R., Ullman J. D. Introduction to Automata Theory, Languages, and Computation. 2nd ed.
- [10] Generalova T.V. Some more on the modeling context-free languages by nondeterministic finite automata // International Journal of Open Information Technologies. — 2021. — Vol. 9, no. 7. — P. 1–4.

Describing Formal Languages Using L-Graphs

A.A. Vylitok

Abstract—Formal language theory, which has been actively developing for nearly seventy years, encompasses problems for which solutions either remain undiscovered or are unsuitable for constructing practical methods. The difficulties stem from the complexity of analyzing computation protocols and derivations in classical frameworks such as Turing machines, pushdown automata, and others. This work explores a method of representing formal languages through L-graphs, which provide a more convenient conceptual framework for analysis. This approach enables the application of efficient graph theory methods to solve problems in formal language theory, particularly the optimization of formal descriptions using techniques known for finite automata (which can be viewed as the simplest representatives of L-graphs). Unrestricted L-graphs characterize the class of recursively enumerable languages, i.e., Chomsky type-0 languages. In other words, any recursively enumerable language can be defined by an L-graph, and conversely, any language defined by an L-graph is recursively enumerable. Subclasses of L-graphs characterizing context-free and regular languages are also described.

Keywords—formal language, methods of describing a formal language, formal languages theory, L-graph, generalization of a finite automaton diagram

References

- [1] Serebryakov V.A. Theory and Implementation of Programming Languages. — Moscow: Nauka, 2012. — 236 p.
- [2] Melnikov B.F. Nondeterministic Finite Automata. — Tolyatti: TSU Publishing House, 2009. — 160 p.
- [3] Stanevichene L.I. On a Tool for Investigating Context-Free Languages // Cybernetics. — 1989. — No. 4. — P. 135–136.
- [4] Vylitok A.A. On Constructing a Pushdown Automaton Graph // Moscow University Bulletin. Series 15: Computational Mathematics and Cybernetics. — 1996. — No. 3. — P. 68–73.
- [5] Stanevichene L.I. On Some Definitions of the Class of Context-Free Languages // Programming and Computer Software. — 1999. — No. 5. — P. 15–25.
- [6] Stanevichene L.I. On the Theory of Context-Free Languages. — Moscow, 2000. — Deposited at VINITI RAS 29.05.2000, No. 1546-B00.
- [7] Vylitok A.A., Sutyryn P.G. Characterization of Formal Languages by Graphs // Abstracts of the Scientific Conference "Tikhonov Readings". — Moscow: Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, 2010. — P. 82–83.
- [8] Minsky M. Computation: Finite and Infinite Machines. — Moscow: Mir, 1971. — 364 p. — Translation from English.
- [9] Hopcroft J.E., Motwani R., Ullman J.D. Introduction to Automata Theory, Languages, and Computation. 2nd ed. / Ed. by A.B. Stavrovsky. — Moscow: Williams Publishing House, 2008. — 528 p. — Translation from English.
- [10] Generalova T.V. Some More on Modeling Context-Free Languages by Nondeterministic Finite Automata // International Journal of Open Information Technologies. — 2021. — Vol. 9, No. 7. — P. 1–4.