Алгоритм преобразования бесконтекстных выражений в эквивалентные L-графы

Му Цзинъюань

***Аннотация*—Рассматриваются бесконтекстные грамматики и их альтернативные представления — а именно, бесконтекстные выражения и бесконтекстные L-графы. Бесконтекстное выражение, по сути, является алгебраической формой представления бесконтекстных грамматик: оно не уступает известным методам в лаконичности, но при этом более подробно и наглядно показывает, как цепочки языка порождаются из подчиненных цепочек. Что касается бесконтекстного L-графа, то он представляет собой ориентированный граф, на дугах которого расположены пометки — символы из алфавита основных символов и дополнительные скобочные метки, влияющие на успешность прохождения пути из начальной вершины в заключительную (требуется сбалансированность скобок).**

В настоящей статье основное внимание уделяется алгоритму преобразования контекстно-свободных выражений в эквивалентные L-графы. Предлагаемый алгоритм сопровождается доказательством корректности и включает идею удаления избыточных вершин. Такой синтез алгебраического и графического подходов объединяет их ключевые преимущества: с одной стороны, бесконтекстные выражения обеспечивают компактность и ясность описания, а с другой — бесконтекстные L-графы наглядно визуализируют структуру языка. Такой комбинированный подход создает предпосылки для разработки более эффективных инструментов анализа и преобразования языковых описаний в компиляторах и системах обработки текста.

***Ключевые слова*—бесконтекстное выражение, обобщение регулярного выражения, контекстно-свободная грамматика, бесконтекстный L-граф.**

I. ВВЕДЕНИЕ

Понятие бесконтекстной (контекстно-свободной) грамматики занимает центральное место во многих разделах теоретической информатики и лежит в основе таких процессов, как компиляция и ассемблирование, а также находит применение в решении задач искусственного интеллекта и проектировании аппаратного обеспечения вычислительных систем.

Бесконтекстные грамматики достаточно выразительны, чтобы представлять синтаксис большинства языков программирования, и почти все современные языки программирования описываются бесконтекстными грамматиками. С другой стороны, эти грамматики достаточно просты, чтобы алгоритмы синтаксического анализа для них были эффективны, причём некоторые из таких алгоритмов имеют линейную сложность для распознавания принадлежности цепочки языку, порождаемому грамматике [1], [2], [3].

Статья получена 30 сентября 2025 г.

Му Цзинъюань, Университет МГУ – ППИ в Шэньчжэне (xirousang @gmail.com).

Альтернативой описанию языков с помощью бесконтекстных грамматик являются бесконтекстные выражения, введённые Л. И. Станевичене в [4], [5], и бесконтекстные L-графы, предложенные в [6]. Бесконтекстные выражения как алгебраическая форма представления более удобны для задания формального языка в программе в виде текстового описания. Более того, такая форма сохраняет лаконичность описательного метода и нагляднее демонстрирует закономерности генерации цепочек языка из подчинённых строк. L-графы предоставляют более интуитивно понятное графическое представление для бесконтекстных языков. Применение методов теории графов позволяет значительно повысить модульность обработки языковых описаний.

В работе предлагается алгоритм преобразования бесконтекстных выражений в соответствующие L-графы и доказывается их эквивалентность. Это позволяет обобщить известный подход для поиска строк по регулярным выражениям, реализованным с помощью конечных автоматов, на случай бесконтекстных языков.

II. ОПИСАНИЕ БЕСКОНТЕКСТНОГО ЯЗЫКА

A. Бесконтекстное выражение

Бесконтекстные выражения (по-другому: контекстно-свободные выражения, кратко — КС-выражения) характеризуют класс бесконтекстных (контекстно-свободных) языков. То есть любое бесконтекстное выражение задаёт некоторый бесконтекстный язык, и наоборот — любой бесконтекстный язык может быть задан бесконтекстным выражением.

В данном разделе на основе понятий, предложенных Л. И. Станевичене в [5], приводится модификация бесконтекстных выражений с добавлением конструкций, явно описывающих итерацию как в регулярных языках, классифицируются виды скобок в составе бесконтекстного выражения, и определяются их свойства и назначение.

Для определения КС-выражений часто используют операцию именования, имеющую два операнда: имя из некоторого конечного множества и выражение, которому это имя сопоставляется. Использование именованных подвыражений позволяет удобно размножать вхождения некоторых подцепочек в цепочку языка. Количество используемых в выражении имен может зависеть от конкретного языка. Операцию именования можно выразить с помощью скобок, окаймляющих операнд выражения скобками $<_\iota$ и $>_\iota$, где ι — операнд.

Определение 1: Пусть $Names$ — непустое конечное множество, содержащее операнд, $\mathcal{B}$ — множество именованного КС-выражения, состоящее из операнд ι и скобок,

$\mathcal{B} = \{<_\iota, >_\iota \mid \iota \in Names\}$. Пусть множества Σ, $\mathcal{B}$ и $\{+, \varepsilon, \emptyset, (,), [,]\}$ попарно не пересекаются, $\mathcal{A} = \Sigma \cup \mathcal{B} \cup \{+, \varepsilon, \emptyset, (,), [,]\}$. Определим рекурсивно КС-выражение над алфавитом Σ (это элементарные символы в алфавите $\mathcal{A}$):

1) $a \in \Sigma \cup \{\varepsilon, \emptyset\}$ — КС-выражение;
2) если α и β — КС-выражения, то:
   a) $\alpha + \beta$ — КС-выражение; подвыражения α и β данного выражения будем называть его *слагаемыми*, само выражение — *суммой*;
   b) $(\alpha)(\beta)$ — КС-выражение; подвыражения α и β выражения $(\alpha)(\beta)$ будем называть его *сомножителями*; заключать сомножитель в круглые скобки не обязательно, если он не является суммой;
3) если $\iota \in Names$, α, β и γ — КС-выражения, то $<_\iota(\alpha)\beta(\gamma)>_\iota$ есть КС-выражение, или *ι-гнездо*; имени ι сопоставляются подвыражения α и γ;
4) если α — КС-выражение, то $[\alpha]$ есть КС-выражение.

Следующие вспомогательные понятия помогают определить язык, задаваемый КС-выражением.

Определение 2: Назовем *фракцией* КС-выражения над алфавитом Σ элемент следующего рекурсивно определяемого множества КС-выражений:

1) $Fractions(a) = \{a\}$ для $a \in \Sigma \cup \{\varepsilon, \emptyset\}$;
2) если α, β и γ суть КС-выражения, то

$$Fractions((\beta)) = Fractions(\beta),$$
$$Fractions(\alpha + \beta) = Fractions(\alpha) \cup Fractions(\beta),$$
$$Fractions((\alpha)(\beta)) = Fractions(\alpha)Fractions(\beta),$$
$$Fractions(<_\iota(\alpha)\beta(\gamma)>_\iota) = \{<_\iota(\}Fractions(\alpha)\{)\}Fractions(\beta)\{(\}Fractions(\gamma)\{)>_\iota\},$$
$$Fractions([\beta]) = \{[\}Fractions(\beta)\{]\}. \quad (1)$$

Из определения фракции следует, что она представляет собой КС-выражение, не содержащее сумм и круглых скобок, порожденных операцией произведения. В состав фракций входят неразложимые слагаемые. Будем называть фракцией каждый элемент определяемого далее клана КС-выражения.

Пример 1: Рассмотрим КС-выражения $\zeta =$

$$<_1(a + b)[b](c)>_1(c + d)\big((a + b)(c + d)\big).$$

Разделим КС-выражение ζ на подвыражения по сомножителям

$$\overbrace{\underbrace{<_1(a + b)[b](c)>_1}_{\iota\text{-гнездо}} \quad \underbrace{(c + d)}_{\text{сомножитель}} \quad \underbrace{\big((a + b)(c + d)\big)}_{\text{сомножитель}}}^{\text{Произведение с 3 сомножителями}},$$

$Fractions(\zeta) = Fractions(<_1(a + b)[b](c)>_1)Fractions(c + d)Fractions((a + b)(c + d))$.

Вычислим фракции для каждого подвыражения:

1) $Fractions(<_1(a + b)[b](c)>_1) = \{<_1(a)[b](c)>_1, <_1(b)[b](c)>_1\}$;
2) $Fractions(c + d) = \{c, d\}$;
3) $Fractions((a + b)(c + d)) = \{ac, ad, bc, bd\}$.

Объединяем результаты $Fractions(\zeta) = \{<_1(a)[b](c)>_1, <_1(b)[b](c)>_1\}\{c, d\}\{ac, ad, bc, bd\}$. В итоговом множестве фракции содержится 16 элементов.

Введем Λ для обозначения пустой цепочки, тогда как ε в этой статье будет встречаться как символ алфавита, из которого составляются регулярные и бесконтекстные выражения.

Определение 3: Пусть α, β и γ — фракции. Для КС-выражения ζ, над алфавитом Σ определим его *пометку* следующим образом:

$$\omega(\zeta) = \begin{cases} \{a\}, \ \zeta = a \in \Sigma; \\ \{\Lambda\}, \ \zeta = \varepsilon \in \Sigma; \\ \emptyset, \ \zeta = \emptyset; \\ \omega(\alpha)\omega(\beta), \ \zeta = \alpha\beta; \\ \omega(\alpha)\omega(\beta)\omega(\gamma), \ \zeta = <_\iota(\alpha)\beta(\gamma)>_\iota; \\ \omega(\beta), \ \zeta = [\beta]. \end{cases} \quad (2)$$

Определение 4: *Клан, порождаемый КС-выражением ζ*, определим рекурсивно:

$$Clan(\zeta) = Trim(Fractions(\zeta)) \cup \{\beta_1 S \beta_3 \mid \alpha_1 S \alpha_3, \ \beta_1 S' \beta_3 \in Clan(\zeta), \ S = <_\iota(\gamma)\alpha_2(\gamma')>_\iota \text{ или } [\alpha_2], \ S' = <_\iota(\gamma)\beta_2(\gamma')>_\iota \text{ или } [\beta_2]\}. \quad (3)$$

Результат *приведения* некоторого КС-выражения ζ обозначим через $Trim(\zeta)$ (см. [5]). Цель приведения КС-выражения заключается в том, чтобы получить эквивалентное КС-выражение, которое имеет более простую структуру и не содержит ненужных элементов, таких как пустые скобки и ненужные гнезда.

Согласно определению клана, клан КС-выражения может не совпадать с множеством его приведённых фракций в случае наличия в КС-выражении нескольких подвыражений с одинаковыми именами. Это объясняется тем, что при формировании клана любое именованное подвыражение может быть заменено любым другим подвыражением с тем же именем. Данная замена реализует механизм разрастания языковых цепочек.

Определение 5: *Язык, задаваемый КС-выражением ζ*, есть подмножество множества Σ^*:

$$L(\zeta) = \bigcup_{\xi \in Clan(\zeta)} \omega(\xi). \quad (4)$$

Язык пуст, если пуст клан.

B. Классификация скобок в КС-выражениях

В КС-выражениях скобочная система определяется с помощью множества именованных и квадратных скобок, а также проекции $projection$ (см. [5]), возвращающей цепочку, которая состоит исключительно из скобок, принадлежащих заданному подмножеству множества именованных и квадратных скобок.

Определение 6: Пусть ζ — некоторое КС-выражение и для любого подмножества B алфавита скобок $\mathcal{B} \cup \{[,]\}$ цепочка $projection(\zeta, B)$ является *скобочной системой*.

В алфавите $\mathcal{A}$, помимо множества именованных КС-выражений $\mathcal{B}$ (угловых скобок), существуют ещё два типа скобок: круглые скобки $\{(,)\}$ и квадратные скобки $\{[,]\}$. Далее приведём правила порождения языка, задаваемые скобками в КС-выражениях.

1) угловые скобки — подцепочки с угловыми скобками, называемые *парными друг другу циклами, левым и правым соответственно*. Для КС-выражения $\zeta = <_\iota(\alpha)\gamma(\beta)>_\iota$, где α, β и γ — КС-выражения, то подцепочки $<_\iota(\alpha$ и $(\beta)>_\iota$ назовем его *парными циклами*;
2) квадратные скобки — подцепочки в квадратных скобках могут *повторяться любое число раз* (это особый вид скобок, добавленный к бесконтекстным выражениям, эти скобки являются аналогом операции звезды Клини в регулярных выражениях). Для КС-выражения $\zeta = \alpha[\gamma]\beta$, где α, β и γ — КС-выражения, то подцепочки $[\gamma]$ назовем его *нейтральными циклами*;
3) круглые скобки — подцепочки в круглых скобках обычно представляют собой либо форму суммы, либо КС-выражения, явно сопоставленные именованным скобкам (угловым скобкам).

C. Глубина, ширина и ядро КС-выражения

Цепочка w принадлежит языку $\mathcal{L}_P$ (см. [7]), если все скобки в ней сбалансированы.

Определение 7: Пусть ζ — КС-выражение и скобочная система P представляет собой проекцию $projection(\zeta,\ \mathcal{B})$. Назовем *$\iota$-глубиной* КС-выражения ζ число $depth_\iota(P)$, определяемое рекурсивно:

1) $depth_\iota(\Lambda) = 0$;
2) $depth_\iota(P) = depth_\iota(P') + 1,\ P = <_\iota P'>_\iota,\ P' \in \mathcal{L}_P$;
3) $depth_\iota(P) = max(depth_\iota(P_1), depth_\iota(P_2)),\ P = P_1P_2,\ P_1,\ P_2 \in \mathcal{L}_P$.

Определение 8: Назовем *глубиной* КС-выражения ζ число

$$depth(\zeta) = max\{depth_\iota(P) \mid P = projection(\zeta,\ \mathcal{B})\}. \tag{5}$$

Для КС-выражения ζ определим его проекцию K как $projection(\zeta,\ \mathcal{B}\cup\{[,\ ]\})$, которая возвращает цепочку, состоящую исключительно из элементов угловых скобок и квадратных скобок. Протяжение определяется как число делений для проекции K, при котором угловые скобки и квадратные скобки сохраняют сбалансированность после делений.

Определение 9: Назовем *протяжением* скобочной системы K число $parti(K)$, определяемое рекурсивно:

1) $parti(\Lambda) = 0$;
2) $parti(K) = 1,\ K = [K']$ или $K = <_\iota K'>_\iota,\ K' \in \mathcal{L}_P$;
3) $parti(K) = parti(K_1) + parti(K_2),\ K = K_1K_2,\ K_1,\ K_2 \in \mathcal{L}_P$.

Определение 10: Назовем *шириной* КС-выражения ζ число

$$\begin{gathered} width(\zeta) = max\{parti(K') \mid K = \\ projection(\zeta,\ \mathcal{B}\cup\{[,\ ]\}), \\ K = K_1K'K_2,\ \text{для некоторых фрагментов} \\ K_1 \text{ и } K_2,\ K' \in \mathcal{L}_P\}. \end{gathered} \tag{6}$$

Определение 11: Пусть $d > 0$. Назовем *d-ядром* КС-выражения ζ и обозначим через $Core(\zeta,\ d)$ подмножество всех элементов его клана, высота которых не превосходит d:

$$Core(\zeta,\ d) = \{\xi \in Clan(\zeta) \mid depth(\xi) \le d\}. \tag{7}$$

Пример 2: Рассмотрим КС-выражение $\zeta =$

$$x[a]y\big[[<_1(a)<_1(a)b(a)>_1(a)>_1u][b]<_2(c)d(a+c)>_2z\big]$$

Получаем следующие скобочные системы P:

1) $projection(\zeta,\ P_1) = <_1<_1>_1>_1$ с глубиной $depth_1(P_1) = 2$, где $P_1 = \{<_1,\ >_1\}$;
2) $projection(\zeta,\ P_2) = <_2>_2$ с глубиной $depth_2(P_2) = 1$, где $P_2 = \{<_2,\ >_2\}$.

Таким образом, глубина КС-выражения ζ равна $depth(\zeta) = 2$.

Рассмотрим теперь скобочные системы K: $projection(\zeta,\ \mathcal{B}\cup\{[,\ ]\}) = [\ ][[<_1<_1>_1>_1][\ ]<_2>_2]$ с протяжением $parti(K) = 2$ (то есть она может быть разделена на две цепочки: $[\ ]$ и $[[<_1<_1>_1>_1][\ ]<_2>_2]$). Чтобы получить ширину, необходимо найти все возможные протяжения и взять максимальное. В данном случае наблюдаем, что внутри цепочки $[[<_1<_1>_1>_1][\ ]<_2>_2]$ присутствует делимая на часть подцепочка $[<_1<_1>_1>_1][\ ]<_2>_2$ (обозначим её K'), для которой $parti(K') = 3$. Затем переходим к подцепочке $<_1<_1>_1>_1$ (обозначим её K'') , для которой $parti(K'') = 1$. Наконец, рассмотрим $<_1>_1$ (обозначим её K'''), для которой $parti(K''') = 1$. Следовательно, ширина КС-выражения ζ равна $width(\zeta) = 3$.

Заметим, что КС-выражения из $Core(\zeta,\ 1)$ сами по себе могут служить правилами вывода для порождения всего языка. Это возможно, поскольку случаи с глубиной $d > 1$ соответствуют повторению левосторонних и правосторонних циклов, уже заложенных в этих правилах.

D. L-граф

Понятие L-графа введено в [6]. L-граф представляет собой ориентированный граф, на дугах которого расположены пометки — символьные из алфавита основных символов и дополнительные скобочные, которые влияют на успешность прохождения пути из начальной вершины в заключительную (требуется баланс по скобкам).

Здесь мы можем представить язык, задаваемый L-графом, и предложить формат представления путей в L-графе.

Определение 12: Определим множество *успешных путей* L-графа G как

$$\begin{gathered} Sentences(G) = \{T \mid beg(T) \in I,\ end(T) \in \\ F,\ \mu(\iota_i(T)) = \varepsilon,\ i = 1, 2\}. \end{gathered} \tag{8}$$

Определение 13: *Язык, задаваемый L-графом G*, есть подмножество

$$L(G) = \bigcup_{T \in Sentences(G)} \omega(T) \tag{9}$$

множества Σ^*.

Формат последовательности, соответствующей отображению пути. Введем морфизм

$$\Phi : E^* \to \big(V \times (\Sigma \cup \{\varepsilon\}) \times (P \cup \{\varepsilon\})\big)^* \times V,$$

сопоставляющий путь T в L-графе G последовательность троек (начальная вершина, пометка дуги, скобочный

след) для всех дуг пути T в порядке их следования, дополненную в конце заключительной вершиной последней дуги пути T.

Пример 3: Рассмотрим L-граф $G =$

$\langle\{q_0,\ q_1,\ q_2\},\ \{a,\ b,\ c\},\ \{<_1,\ >_1\},\ E,\ \{q_0\},\ \{q_2\}\rangle$, где
$E = \{(q_0,\ a,\ <_1,\ \varepsilon,\ q_0),\ (q_0,\ b,\ \varepsilon,\ \varepsilon,\ q_1),$
$(q_1,\ a,\ >_1,\ \varepsilon,\ q_1),\ (q_1,\ c,\ \varepsilon,\ \varepsilon,\ q_2),\ (q_2,\ b,\ \varepsilon,\ \varepsilon,\ q_2)\}$,

изображенный на рисунке 1.

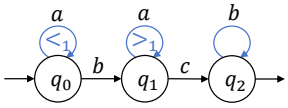

Рис. 1. Пример L-графа G

Согласно построенному L-графу G, мы можем записать его $Core(G,\ 1,\ 1) \cap Sentences(G)$ как

$$\left\{\begin{array}{c} \to q_0 \overset{b}{-} q_1 \overset{c}{-} q_2 \to, \\ \to q_0 \underset{<_1}{\overset{a}{-}} q_0 \overset{b}{-} q_1 \underset{>_1}{\overset{a}{-}} q_1 \overset{c}{-} q_2 \to, \\ \to q_0 \overset{b}{-} q_1 \overset{c}{-} q_2 \overset{b}{-} q_2 \to, \\ \to q_0 \underset{<_1}{\overset{a}{-}} q_0 \overset{b}{-} q_1 \underset{>_1}{\overset{a}{-}} q_1 \overset{c}{-} q_2 \overset{b}{-} q_2 \to \end{array}\right\}.$$

Здесь видим, что цикл $q_2 \overset{b}{-} q_2$ является нейтральным циклом, поскольку он сбалансирован по скобкам.

III. ПРЕОБРАЗОВАНИЕ БЕСКОНТЕКСТНОГО ВЫРАЖЕНИЯ В ЭКВИВАЛЕНТНЫЙ L-ГРАФ

A. Базовые и обобщенные конструкции КС-выражений и L-графов

Согласно определению КС-выражений, выделим пять их форм. Если КС-выражение ζ относится к одной из этих форм, то соответствующий L-граф G может быть построен непосредственно либо методом подграфов. Базовые конструкции (БК) и обобщённые конструкции (ОК) рассматриваются ниже следующим образом:

1) элементарный символ: для КС-выражения $\zeta = a$, где $a \in \Sigma \cup \{\varepsilon\}$, соответствующий L-граф $G = \langle\{q_0,\ q_1\},\ \{a\},\ \{\varepsilon\},\ \{(q_0,\ a,\ \varepsilon,\ \varepsilon,\ q_1)\},\ \{q_0\},\ \{q_1\}\rangle$ (рис. 2);

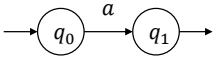

Рис. 2. БК-1: элементарный символ

для КС-выражения $\zeta = \alpha$, где $\alpha \in \Sigma^*$, соответствующий L-граф $G = \langle\{q_0, \ldots, q_n\},\ \Sigma_\alpha,\ \{\varepsilon\},\ E,\ \{q_0\},\ \{q_n\}\rangle$, где $E = E_\alpha \cup \{(q_0,\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_{\alpha_{start}}),\ (q_{\alpha_{end}},\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_n)\}$ (рис. 3);

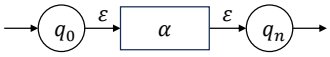

Рис. 3. ОК-1: элементарный символ

2) сумма: для КС-выражения $\zeta = a + b$, где $a,\ b \in \Sigma$, соответствующий L-граф $G = \langle\{q_0,\ q_1\},\ \{a,\ b\},\ \{\varepsilon\},\ E,\ \{q_0\},\ \{q_1\}\rangle$, где $E = \{(q_0,\ a,\ \varepsilon,\ \varepsilon,\ q_1),\ (q_0,\ b,\ \varepsilon,\ \varepsilon,\ q_1)\}$ (рис. 4);

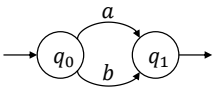

Рис. 4. БК-2: сумма

для КС-выражения $\zeta = \alpha + \beta$, где α, β — КС-выражения, соответствующий L-граф $G = \langle\{q_0, \ldots, q_n\},\ \Sigma_\alpha \cup \Sigma_\beta,\ \mathcal{B}_\alpha \cup \mathcal{B}_\beta,\ E,\ \{q_0\},\ \{q_n\}\rangle$, где $E = E_\alpha \cup E_\beta \cup \{(q_0,\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_{\alpha_{start}}),\ (q_0,\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_{\beta_{start}}),\ (q_{\alpha_{end}},\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_n),\ (q_{\beta_{end}},\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_n)\}$ (рис. 5);

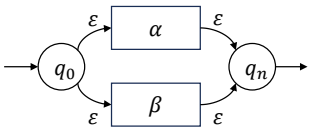

Рис. 5. ОК-2: сумма

3) произведение: для КС-выражения $\zeta = ab$, где $a,\ b \in \Sigma$, соответствующий L-граф $G = \langle\{q_0,\ q_1,\ q_2\},\ \{a,\ b\},\ \{\varepsilon\},\ E,\ \{q_0\},\ \{q_2\}\rangle$, где $E = \{(q_0,\ a,\ \varepsilon,\ \varepsilon,\ q_1),\ (q_1,\ b,\ \varepsilon,\ \varepsilon,\ q_2)\}$ (рис. 6);

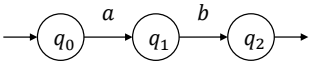

Рис. 6. БК-3: произведение

для КС-выражения $\zeta = \alpha\beta$, где α, β — КС-выражения, соответствующий L-граф $G = \langle\{q_0, \ldots, q_n\},\ \Sigma_\alpha \cup \Sigma_\beta,\ \mathcal{B}_\alpha \cup \mathcal{B}_\beta,\ E,\ \{q_0\},\ \{q_n\}\rangle$, где $E = E_\alpha \cup E_\beta \cup \{(q_{\alpha_{end}},\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_{\beta_{start}})\}$ (рис. 7);

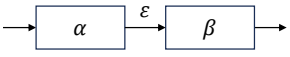

Рис. 7. ОК-3: произведение

4) ι-гнездо: для КС-выражения $\zeta = <_\iota(a)b(c)>_\iota$, где $a,\ b,\ c \in \Sigma$, соответствующий L-граф $G = \langle\{q_0,\ q_1\},\ \{a,\ b,\ c\},\ \{<_\iota,\ >_\iota\},\ E,\ \{q_0\},\ \{q_1\}\rangle$, где $E = \{(q_0,\ a,\ <_\iota,\ \varepsilon,\ q_0),\ (q_0,\ b,\ \varepsilon,\ \varepsilon,\ q_1),\ (q_1,\ c,\ >_\iota,\ \varepsilon,\ q_1)\}$ (рис. 8);

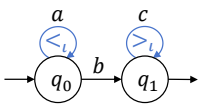

Рис. 8. БК-4: ι-гнездо

для КС-выражения $\zeta = <_\iota(\alpha)\beta(\gamma)>_\iota$, где α, β, γ — КС-выражения, соответствующий L-граф $G = \langle\{q_0, \ldots, q_n\},\ \Sigma_\alpha \cup \Sigma_\beta \cup \Sigma_\gamma,\ \mathcal{B}_\alpha \cup \mathcal{B}_\beta \cup \mathcal{B}_\gamma \cup \{<_\iota,\ >_\iota\},\ E,\ \{q_0\},\ \{q_n\}\rangle$, где $E = E_\alpha \cup E_\beta \cup E_\gamma \cup \{(q_0,\ \varepsilon,\ <_\iota,\ \varepsilon,\ q_{\alpha_{start}}),\ (q_{\alpha_{end}},\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_0),\ (q_0,\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_{\beta_{start}}),\ (q_{\beta_{end}},\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_n),\ (q_n,\ \varepsilon,\ >_\iota,\ \varepsilon,\ q_{\gamma_{start}}),\ (q_{\gamma_{end}},\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_n)\}$ (рис. 9);

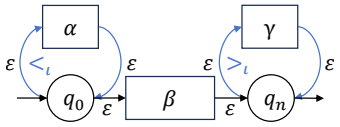

Рис. 9. ОК-4: ι-гнездо

5) нейтральный цикл: для КС-выражения $\zeta = [b]$, где $b \in \Sigma$, соответствующий L-граф $G = \langle\{q_0\},\ \{b\},\ \{\varepsilon\},\ \{(q_0,\ b,\ \varepsilon,\ \varepsilon,\ q_0)\},\ \{q_0\},\ \{q_0\}\rangle$ (рис. 10);

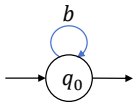

Рис. 10. БК-5: нейтральный цикл

для КС-выражения $\zeta = [\beta]$, где β — КС-выражение, соответствующий L-граф $G = \langle\{q_0, \ldots\},\ \Sigma_\beta,\ \{\varepsilon\},\ E_\beta \cup \{(q_0,\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_{\beta_{start}}),\ (q_{\beta_{end}},\ \varepsilon,\ \varepsilon,\ \varepsilon,\ q_0)\},\ \{q_0\},\ \{q_0\}\rangle$ (рис. 11).

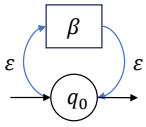

Рис. 11. ОК-5: нейтральный цикл

B. Устранение избыточных вершин из L-графа

При взаимном соединении построенных обобщенных конструкций могут возникать некоторые избыточные конструкции. Мы можем исключить избыточные вершины и получить эквивалентный L-граф, описывающий тот же язык.

Анализируя конструкции, в частности конструкции суммы и произведения, можно заметить, что они соответствуют параллельному и последовательному соединению элементарных символов соответственно.

Если КС-выражение начинается и/или заканчивается элементарным символом (т. е. $\alpha = a\alpha'b$, $a,\ b \in \Sigma$), соответствующие начальную и/или конечную вершины L-графа можно объединить, исключив избыточные ε-дуги. При этом отдельные вершины для параллельного соединения не требуются (рис. 12). Аналогичное устранение применимо и к последовательному соединению (рис. 13).

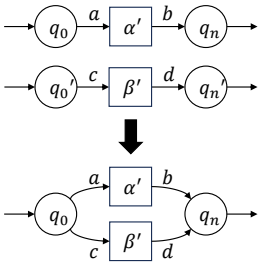

Рис. 12. Упрощение конструкции суммы

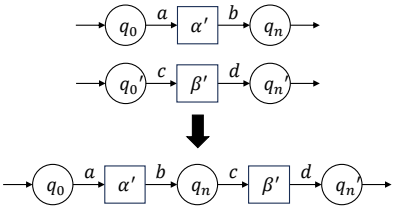

Рис. 13. Упрощение конструкции произведения

Конструкции ι-гнездо и нейтральный цикл требуют, чтобы их скобки были сбалансированы. ι-гнездо представляет собой два парных цикла (левый и правый) с заключенной между ними центральной частью. При этом как парная структура циклов, так и центральная часть должны быть сбалансированы по скобкам. В отличие от этого, нейтральный цикл является просто циклом, имеющим сбалансированные скобки.

По аналогии с устранением вершин при последовательном соединении элементарных символов, можно выполнить аналогичное устранение вершин для последовательно соединенных ι-гнезд (рис. 14).

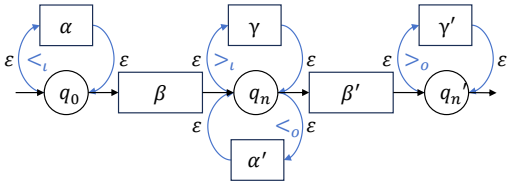

Рис. 14. Упрощение конструкции произведения ι-гнезда

Более детальные аспекты оптимизации L-графов, в частности сопряжение L-графов, будут исследованы в наших последующих работах.

C. Алгоритм преобразования КС-выражения в эквивалентный L-граф

Алгоритм 1: Для заданного КС-выражения ζ необходимо построить алгоритм $\mathcal{A}_{E\to L}(\zeta)$ для преобразования КС-выражения ζ в соответствующий L-граф G:

$$G := \mathcal{A}_{E\to L}(\zeta).$$

Вход: КС-выражение $\zeta \neq \emptyset$.
Выход: соответствующий эквивалентный L-граф G.
Шаг 1. Разложение на обобщённые конструкции.

Заданное КС-выражение ζ разделим по сомножителям:

$\zeta = \xi_1 \cdot \ldots \cdot \xi_i \cdot \ldots \cdot \xi_n,\ \xi_i$ — сомножитель $\zeta,\ 1 \le i \le n.$

Относим каждый сомножитель ξ_i к одному из пяти типов обобщённых конструкций.

Шаг 2. Построение подграфов для ОК-типов.

Строим рекурсивно для каждого ξ_i подграф согласно его типу:

1) ОК-1 (элементарный символ): вводим 2 новые вершины и добавляем 2 ε-дуги для соединения вершин;
2) ОК-2 (сумма): вводим 2 новые вершины и добавляем 4 ε-дуги для для параллельного соединения подграфов;
3) ОК-3 (произведение): добавляем 1 ε-дугу для последовательного соединения подграфов;
4) ОК-4 (ι-гнездо): вводим 2 новые вершины и добавляем 2 скобочные дуги (парные скобки), 2 ε-дуги для связи с центральной частью и 2 ε-дуги для возврата к вершинам;
5) ОК-5 (нейтральный цикл): добавляем 2 ε-дуги для зацикливания;

Шаг 3. Достижение базовой конструкции.

Завершаем рекурсию при достижении базовой конструкции, которую преобразуем в базовый L-граф напрямую.

Шаг 4. Объединяем все подграфы базовых конструкций.

Отмечаем, что при выполнении последовательного и параллельного соединений допускается устранение избыточных вершин, как описано в разделе III-B.

D. Эквивалентность КС-выражений и L-графов

Эквивалентность L-графа и КС-выражения заключается в том, что данное КС-выражение ζ и соответствующий L-граф G, полученный с помощью алгоритма преобразования в разделе III-C, задают один и тот же язык:

$$L(\mathcal{A}_{E\to L}(\zeta)) = L(\zeta). \tag{10}$$

Теорема 1: Язык задается КС-выражением тогда и только тогда, когда он определяется некоторым L-графом.

Следующие определения и лемма помогают нам доказать эквивалентность КС-выражений и L-графов.

Множество $V(\zeta)$ и $E(\zeta)$ представляют собой, соответственно множество вершин и множество дуг, полученные при преобразовании КС-выражения ζ в соответствующий L-граф G согласно алгоритму, описанному в разделе III-C.

Для $s \ge 0$ и $P,\ Q \in V(\zeta)$ введем обозначение $Path(P,\ Q,\ s)$ для множества всех путей в L-графе G между вершинами P и Q длины s, где длина пути определяется количеством дуг в нём. Пусть $Path(P,\ Q) = \cup_{s\ge 0} Path(P,\ Q,\ s)$.

Определение 14: Назовем *путем* КС-выражения ζ элемент следующего рекурсивно определяемого множества $PathE(\zeta)$:

$$\begin{gathered} PathE(\zeta) = Clan(\zeta) \cup \{x(<_\iota(u))^k y((v)>_\iota)^l z \mid \\ x<_\iota(u)y(v)>_\iota z \in PathsE(\zeta),\ k,\ l \ge 0\}. \end{gathered} \tag{11}$$

Для $s \ge 0$ и $a_1,\ a_2 \in \mathcal{A}(\zeta)$ введем обозначение $StrE(a_1,\ a_2,\ s)$ для множества всех цепочек длины s, порождённых участками путей КС-выражения ζ, начинающихся с a_1 и заканчивающихся a_2. Пусть $StrE(a_1,\ a_2) = \cup_{s\ge 0} StrE(a_1,\ a_2,\ s)$.

Введем морфизм

$$\psi : E^* \to (\Sigma^* \times P)^*,$$

который преобразует путь, удаляя все вершины и ε-дуги, объединяя пометки дуг с соответствующими скобочными следами (при этом для нейтральных циклов полученная комбинация заключается в квадратные скобки).

Лемма 1: Для любых $P,\ Q \in V(\zeta)$, если $Path(P,\ Q,\ s) \ne \emptyset$, то существует такая биекция

$$\varphi : Path(P,\ Q) \to StrE(a_1,\ a_2).$$

что для каждого $T \in Path(P,\ Q)$ выполняется равенство

$$\psi(T) = \varphi(T). \tag{12}$$

Доказательство. Рассмотрим случай $s = 0$. Тогда $Path(P,\ Q) = \{\Lambda\},\ StrE(a_1,\ a_2) = \{\Lambda\}$. Следовательно, в данном случае лемма верна.

Пусть $s > 0$. Из базовых конструкций и обобщённых конструкций, используя представление формата последовательности пути для $Path(P, Q)$, получаем:

1) БК-1: $Path(q_0,\ q_1) = \{\to q_0 \overset{a}{-} q_1 \to\}$;
   ОК-1: $a \in \Sigma_\alpha$, $Path(P,\ Q) =$
$$\left\{\begin{array}{c} \to q_0 \overset{\varepsilon}{-} q_n \to, \\ \ldots \\ P \ldots \overset{a}{-} \ldots Q, \\ \ldots \\ \to q_0 \overset{\varepsilon}{-} q_{\alpha_{start}} \ldots q_{\alpha_{end}} \overset{\varepsilon}{-} q_n \to \end{array}\right\};$$
2) БК-2: $Path(q_0,\ q_1) =$
   $\{\to q_0 \overset{a}{-} q_1 \to,\ \to q_0 \overset{b}{-} q_1 \to\}$;
   ОК-2: $a \in \Sigma_\alpha,\ b \in \Sigma_\beta$, $Path(P,\ Q) =$
$$\left\{\begin{array}{c} \to q_0 \overset{\varepsilon}{-} q_n \to, \\ \ldots \\ P \ldots \overset{a}{-} \ldots Q, \\ P \ldots \overset{b}{-} \ldots Q, \\ \ldots \\ \to q_0 \overset{\varepsilon}{-} q_{\alpha_{start}} \ldots q_{\alpha_{end}} \overset{\varepsilon}{-} q_n \to, \\ \to q_0 \overset{\varepsilon}{-} q_{\beta_{start}} \ldots q_{\beta_{end}} \overset{\varepsilon}{-} q_n \to \end{array}\right\};$$
3) БК-3: $Path(q_0,\ q_2) =$
   $\{\to q_0 \overset{a}{-} q_1,\ q_1 \overset{b}{-} q_2 \to,\ \to q_0 \overset{a}{-} q_1 \overset{b}{-} q_2 \to\}$;
   ОК-3: $a \in \Sigma_\alpha,\ b \in \Sigma_\beta$, $Path(P,\ Q) =$

$$\left\{\begin{array}{c} \to q_0 \overset{\varepsilon}{-} q_n \to, \\ \dots \\ P \dots \overset{a}{-} \dots \overset{b}{-} \dots Q, \\ \dots \\ \to q_0 \overset{\varepsilon}{-} q_{\alpha_{start}} \dots q_{\alpha_{end}} \overset{\varepsilon}{-} \\ q_{\beta_{start}} \dots q_{\beta_{end}} \overset{\varepsilon}{-} q_n \to \end{array}\right\};$$

4) БК-4: $Path(q_0,\ q_2) = \{\to q_0 \underset{<_\iota}{\overset{a}{-}} q_0,\ \to q_0 \overset{b}{-} q_1 \to, q_1 \underset{>_\iota}{\overset{c}{-}} q_1 \to,\ \to q_0 \underset{<_\iota}{\overset{a}{-}} q_0 \overset{b}{-} q_1,\ q_0 \overset{b}{-} q_1 \underset{>_\iota}{\overset{c}{-}} q_1 \to,\ \to q_0 \underset{<_\iota}{\overset{a}{-}} q_0 \overset{b}{-} q_1 \underset{>_\iota}{\overset{c}{-}} q_1 \to,\ \cdots\}$;
ОК-4: $a \in \Sigma_\alpha,\ b \in \Sigma_\beta,\ c \in \Sigma_\gamma,\ Path(P,\ Q) =$

$$\left\{\begin{array}{c} \to q_0 \overset{\varepsilon}{-} q_n \to, \\ \dots \\ \to q_0 \underset{<_\iota}{\overset{\varepsilon}{-}} \dots \overset{a}{-} \dots \overset{\varepsilon}{-} q_0, \\ \dots \\ P \dots \overset{b}{-} \dots Q, \\ \dots \\ q_n \underset{>_\iota}{\overset{\varepsilon}{-}} \dots \overset{c}{-} \dots \overset{\varepsilon}{-} q_n \to, \\ \dots \\ \to q_0 \underset{<_\iota}{\overset{\varepsilon}{-}} q_{\alpha_{start}} \dots q_{\alpha_{end}} \overset{\varepsilon}{-} q_0 \\ \overset{\varepsilon}{-} q_{\beta_{start}} \dots q_{\beta_{end}} \overset{\varepsilon}{-} \\ q_n \underset{>_\iota}{\overset{\varepsilon}{-}} q_{\gamma_{start}} \dots q_{\gamma_{end}} \overset{\varepsilon}{-} q_n \to \end{array}\right\};$$

5) БК-5: $Path(q_0,\ q_0) = \{\to q_0 \overset{b}{-} q_0 \to,\ \to q_0 \overset{b}{-} q_0 \overset{b}{-} q_0 \to,\ \cdots\}$;
ОК-5: $b \in \Sigma_\beta,\ Path(P,\ Q) =$

$$\left\{\begin{array}{c} \to q_0 \overset{\varepsilon}{-} q_0 \to, \\ \dots \\ P \dots \overset{b}{-} \dots Q, \\ \dots \\ \to q_0 \overset{\varepsilon}{-} q_{\beta_{start}} \dots q_{\beta_{end}} \overset{\varepsilon}{-} q_0 \to \end{array}\right\};$$

Применяя морфизм ψ к пяти типам путей, получаем:
1) БК-1: $\{a\}$; ОК-1: $\{\ \cdots a \cdots\ \}$;
2) БК-2: $\{a,\ b\}$; ОК-2; $\{\ \cdots a \cdots,\ \cdots b \cdots\ \}$;
3) БК-3: $\{a,\ b,\ ab\}$;
ОК-3: $\{\ \cdots a \cdots,\ \cdots b \cdots,\ \cdots a \cdots b \cdots\ \}$;
4) БК-4: $\{<_\iota(a),\ b,\ (c)>_\iota,\ <_\iota(a)b,\ b(c)>_\iota,\ <_\iota(a)b(c)>_\iota,\ \dots\}$;
ОК-4: $\{<_\iota(\cdots a \cdots),\ \cdots b \cdots,\ (\cdots c \cdots)>_\iota,\ <_\iota(\cdots a \cdots)\cdots b \cdots(\cdots c \cdots)>_\iota,\ \dots\}$;
5) БК-5: $\{[b],\ [b][b],\ \dots\}$;
ОК-5: $\{\ [\cdots b \cdots],\ [\cdots b \cdots][\cdots b \cdots],\ \dots\}$.

Анализ показывает, что построенные множества пяти типов путей находятся в точном соответствии с $StrE(a_1, a_2)$. Следовательно, определенное отображение φ образует требуемую биекцию. □

Следствие 1: $L(\mathcal{A}_{E\to L}(\zeta)) = L(\zeta)$.

Доказательство. По лемме существует такая биекция

$$\varphi : Sentences(\mathcal{A}_{E\to L}(\zeta)) \to Clan(\zeta),$$

что для каждого успешного пути $T \in Sentences(\mathcal{A}_{E\to L}(\zeta))$ выполняется равенство

$$\psi(T) = \varphi(T).$$

Следовательно,

$$\omega(T) = projection(\varphi(T),\ \Sigma).$$

Значит,

$$L(\mathcal{A}_{E\to L}(\zeta)) = L(\zeta).\ \square$$

Пример 4: Мы построим соответствующий L-граф, используя $Core(\zeta,\ 1)$ из КС-выражения, приведённого в примере 2.

Рассмотрим КС-выражение $\zeta =$

$$x[a]y\big[[<_1(a)b(a)>_1 u][b]<_2(c)d(a+c)>_2 z\big].$$

Шаг 1. Заданное КС-выражение разделим по сомножителям:

$$\underbrace{x}_{\text{БК-1}} \cdot \underbrace{[a]}_{\text{БК-5}} \cdot \underbrace{y}_{\text{БК-1}} \cdot \underbrace{\big[[<_1(a)b(a)>_1 u][b]<_2(c)d(a+c)>_2 z\big]}_{\text{ОК-5}}$$

Шаг 2. Построение подграфов для ОК-типов (рис. 15).

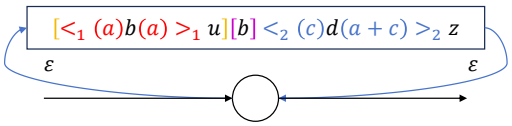

Рис. 15. Построение подграфа для ОК-5

Шаг 3. Повторяем шаги 1 и 2 до тех пор, пока не перестанут появляться ОК-типы.

Шаг 3.1 Заданное КС-выражение разделим по сомножителям:

$$\underbrace{[<_1(a)b(a)>_1 u]}_{\text{ОК-5}} \cdot \underbrace{[b]}_{\text{БК-5}} \cdot \underbrace{<_2(c)d(a+c)>_2}_{\text{ОК-4}} \cdot \underbrace{z}_{\text{БК-1}}$$

Шаг 3.2. Построение подграфов для ОК-типов (рис. 16 и рис. 17).

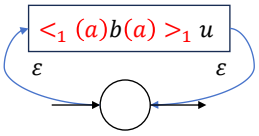

Рис. 16. Построение подграфа для ОК-5

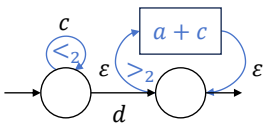

Рис. 17. Построение подграфа для ОК-4

Шаг 3.3. Повторяем шаги 1 и 2 до тех пор, пока не перестанут появляться ОК-типы.

Шаг 3.3.1 Заданное КС-выражение разделим по сомножителям:

$$\underbrace{<_1(a)b(a)>_1}_{\text{БК-4}} \cdot \underbrace{u}_{\text{БК-1}} \quad \underbrace{a+c}_{\text{БК-2}}$$

Шаг 3.3.2. Построение подграфов для ОК-типов.

Шаг 4. Объединяем все подграфы базовых конструкций (рис. 18).

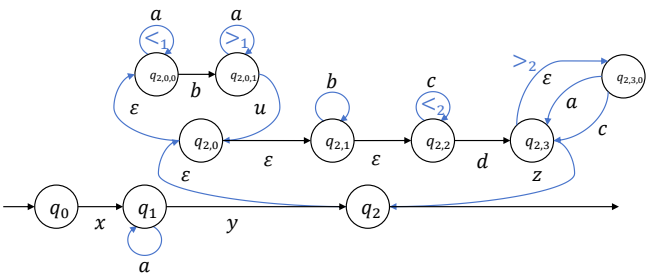

Рис. 18. Полученный эквивалентный L-граф

IV. ЗАКЛЮЧЕНИЕ

Уточнена классификация видов скобок в бесконтекстных выражениях. Разработан алгоритм преобразования бесконтекстных выражений в эквивалентные L-графы, и доказана корректность их эквивалентности. Предложена идея устранения избыточных вершин из L-графа.

БЛАГОДАРНОСТИ

Результаты настоящего исследования были получены при финансовой поддержке Правительства г. Шэньчжэня и Университета МГУ-ППИ в Шэньчжэне.

Список литературы

[1] 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.

[2] Волкова И. А., Вылиток А. А., Руденко Т. В. Формальные грамматики и языки. Элементы теории трансляции. — 3 изд. — М. : Издательский отдел факультета ВМиК МГУ им. М. В. Ломоносова, 2009. — 115 с.

[3] Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции / Под ред. В. М. Курочкин. — М. : Мир, 1978. — Т. 1. — Пер. с англ.: Aho A., Ullman J. The Theory of Parsing, Translation and Compiling. Vol. 1.

[4] Станевичене Л. И. К теории бесконтекстных языков. — М., 2000. — Деп. в ВИНИТИ РАН 29.05.2000, №1546-В00.

[5] Гомозов А. Л., Станевичене Л. И. Об одном обобщении регулярных выражений // Программирование. — 2000. — № 5. — С. 31–43. — Библиогр.: с.43.

[6] 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.

[7] Станевичене Л. И. О некоторых определениях класса КС-языков видеоданных // Программирование. — 1999. — № 5. — С. 15–25. — УДК 519.682.1.

МУ Цзинъюань,
аспирант Университета МГУ – ППИ в Шэньчжэне
(http://szmsubit.ru/),
email: xirousang@gmail.com.

Algorithm for transforming context-free expressions into equivalent L-graphs

Mu Jingyuan

***Abstract*—This work examines context-free grammars and their alternative representations—context-free expressions and context-free L-graphs. A context-free expression serves as an algebraic form for representing context-free grammars, matching known methods in conciseness while simultaneously providing a more detailed and visual demonstration of how language strings are generated from subordinate strings. A context-free L-graph is a directed graph whose arcs are labeled with symbols from a primary alphabet and additional bracket markers that affect the successful traversal of a path from the initial vertex to the final vertex (bracket balance is required).**

This article focuses on the algorithm for converting context-free expressions into equivalent L-graphs. The proposed algorithm is accompanied by a correctness proof and incorporates the idea of removing redundant vertices. Such a synthesis of algebraic and graphical approaches combines their key advantages: on one hand, context-free expressions ensure compact and clear descriptions, while on the other hand, context-free L-graphs provide intuitive visualization of the language structure. This combined approach opens prospects for developing more effective tools for analyzing and transforming language models in compilers and text processing systems.

***Keywords*—context-free expression, extension of regular expressions, context-free grammar, context-free L-graph.**

References

[1] Hopcroft J. E., Motwani R., Ullman J. D. Introduction to Automata Theory, Languages, and Computation / Ed. by A. B. Stavrovsky. — 2nd ed. — Moscow : Williams Publishing House, 2008. — 528 p. — Translation from English: Hopcroft J. E., Motwani R., Ullman J. D. Introduction to Automata Theory, Languages, and Computation. 2nd ed.

[2] Volkova I. A., Vylitok A. A., Rudenko T. V. Formal Grammars and Languages. Elements of Translation Theory. — 3rd ed. — Moscow : Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics Publishing Department, 2009. — 115 p.

[3] Aho A., Ullman J. The Theory of Parsing, Translation and Compiling / Ed. by V. M. Kurochkin. — Moscow : Mir Publishers, 1978. — Vol. 1. — Translated from English: Aho A., Ullman J. The Theory of Parsing, Translation and Compiling. Vol. 1.

[4] Stanevichene L. I. On the Theory of Context-Free Languages. — Moscow, 2000. — Deposited in VINITI RAS 29.05.2000, No. 1546-B00.

[5] Gomozov A. L., Stanevichene L. I. On One Generalization of Regular Expressions // Programming and Computer Software. — 2000. — No. 5. — P. 31–43. — Bibliogr.: p.43.

[6] Vylitok A. A., Sutyrin 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.

[7] Stanevichene L. I. On Some Definitions of the Class of Context-Free Languages // Programming and Computer Software. — 1999. — No. 5. — P. 15–25. — UDC 519.682.1.

MU Jingyuan,
Post-graduate student of Shenzhen MSU – BIT University, China (http://szmsubit.ru/),
email: xirousang@gmail.com.