

Московский государственный университет имени М.В. Ломоносова  
Факультет вычислительной математики и кибернетики  
Кафедра алгоритмических языков

на правах рукописи

Сутырин Павел Георгиевич

Алгоритмы обработки графовых описаний бесконтекстных языков

Специальность 05.13.11 — математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

**АВТОРЕФЕРАТ**  
диссертации на соискание учёной степени кандидата  
физико-математических наук

Москва — 2010

Работа выполнена на кафедре алгоритмических языков факультета вычислительной математики и кибернетики Московского государственного университета им. М.В. Ломоносова.

Научный руководитель: доктор физико-математических наук,  
профессор М.Г. Мальковский.

Официальные оппоненты: доктор физико-математических наук,  
профессор Р.И. Подловченко

доктор физико-математических наук,  
профессор Б.Ф. Мельников

Ведущая организация: Вычислительный центр  
им. А. А. Дородницына  
Российской академии наук

Защита диссертации состоится 28 мая 2010 года в 11.00 на заседании диссертационного совета Д 501.001.44 при Московском государственном университете имени М.В. Ломоносова по адресу: 119991, ГСП-1, Москва, Ленинские горы, МГУ, второй учебный корпус, факультет вычислительной математики и кибернетики, ауд. 685.

С диссертацией можно ознакомиться в библиотеке факультета вычислительной математики и кибернетики МГУ. С текстом автореферата можно ознакомиться на официальном сайте ВМК МГУ <http://cs.msu.ru> в разделе “Наука” — “Работа диссертационных советов” — “Д 501.001.44”.

Автореферат диссертации разослан “ ” апреля 2010 года.

Ученый секретарь  
диссертационного совета  
профессор

Н.П. Грифонов

# Общая характеристика работы

## Актуальность темы

Контекстно-свободные (КС), или бесконтекстные языки образуют важный в практическом отношении класс формальных языков. Они используются для описания синтаксиса современных языков программирования и языков разметки, допускают построение эффективных анализаторов. КС-языки могут быть заданы такими формальными описаниями, как КС-грамматики, магазинные автоматы и D-графы.

Для некоторых задач в классе КС-языков доказана алгоритмическая неразрешимость, в частности для задач проверки однозначности КС-грамматики, эквивалентности двух КС-грамматик, принадлежности языка КС-грамматики классу регулярных языков (проблема регулярности).

Поиск более узких подклассов КС-языков, допускающих решение этих задач, представляет не только теоретический, но и практический интерес. В частности, алгоритм проверки регулярности полезен для автоматизированной оптимизации анализаторов путем замены магазинных автоматов эквивалентными конечными автоматами.

Известно, что проблема регулярности разрешима для подкласса детерминированных КС-языков, допускаемых детерминированными магазинными автоматами. При этом большое значение имеет поиск для нее практически применимых алгоритмов.

Разрешимость проблемы эквивалентности также доказана для детерминированных магазинных автоматов. Однако, имеющиеся доказательства (G. Senizergues, 1997, В. Мейтус, 1992) являются чрезвычайно громоздкими и сложными для понимания, что заставляет искать более компактное и естественное решение.

Введенные в работах Л.И. Станевичене D-графы<sup>1</sup> представляют собой новый способ описания КС-языков. Их прообразами являются графы магазинных автоматов. D-графы — более удобный инструмент исследования многих свойств КС-языков, чем магазинные автоматы и КС-грамматики. Так, с помощью D-графов было построено более простое доказательство теоремы Хомского-Шютценберже о морфическом представлении КС-языка.

---

<sup>1</sup>Станевичене Л.И. К теории бесконтекстных языков. М.: МГУ им. М.В. Ломоносова. 2000. Деп. в ВИНТИ 29.05.2000. № 1546-В00.

## **Цель работы**

Цель диссертационной работы состоит в развитии графовых описаний формальных языков на основе D-графов, получении с их помощью более качественных доказательств свойств языков и алгоритмов обработки языковых описаний, практически применимых для разработки трансляторов и их генераторов, а также для автоматизации теоретических исследований.

## **Задачи работы**

1. Применить аппарат D-графов к исследованию регулярных языков, исследовать задачу о звездной высоте с использованием теории графов.
2. Найти максимально широкий подкласс детерминированных бесконтекстных языков, для которого на основе графовых описаний удастся построить более простые решения проблем регулярности и эквивалентности.
3. Для подкласса графовых описаний с разрешимой эквивалентностью построить полную в этом подклассе систему эквивалентных преобразований.

## **Методы исследования**

В работе используются методы теории формальных языков, теории графов, теории сложности, комбинаторного анализа.

## **Научная новизна и основные результаты**

Основные результаты диссертации являются новыми, получены автором самостоятельно и состоят в следующем:

1. Предложена графовая характеристика бесконтекстных языков, обобщающая диаграммы конечных автоматов и D-графы, позволяющая применять алгоритмы обработки графов к описаниям регулярных и бесконтекстных языков, получать более компактные доказательства их свойств.
2. Для графовых описаний, задающих регулярные языки, построен алгоритм преобразования их в эквивалентные регулярные выражения.

Данный алгоритм строит выражения меньшей сложности, чем известные алгоритмы. Выделен подкласс графовых описаний, для которых предложенный алгоритм дает решение задачи о звездной высоте.

3. Выделен подкласс графовых описаний с однозначной парностью, для которого разрешимы проблемы эквивалентности и регулярности, построены соответствующие алгоритмы.
4. Для подкласса графовых описаний с однозначной парностью на основе алгоритма проверки эквивалентности построена система эквивалентных преобразований и доказана ее полнота в этом классе.

Достоверность полученных результатов подтверждается теоретическим исследованием, формулировкой и доказательством соответствующих теорем.

## **Теоретическая и практическая значимость**

В основном работа носит теоретический характер. Теоретическая значимость работы состоит в развитии графовых описаний бесконтекстных языков, получении на их основе более ясных и простых доказательств свойств подкласса  $LL(1)$ -языков, чем существующие. Практическая значимость состоит в применимости предложенных алгоритмов для построения эффективных трансляторов и их генераторов, а также для автоматизации теоретических исследований.

## **Апробация работы**

Результаты научных исследований были представлены на Международной конференции “Mathematical Modeling and Computational Physics” (Дубна, 2009), двух Международных научно-практических интернет-конференциях “Современные направления теоретических и прикладных исследований’2009” и “Научные исследования и их практическое применение. Современное состояние и пути развития’2009”, научных конференциях МГУ “Ломоносовские чтения” и “Тихоновские чтения” (2009), а также на научно-исследовательских семинарах в МГУ имени М.В. Ломоносова и Вычислительном центре им. А.А. Дородницына РАН (2010).

## **Публикации**

По теме диссертации опубликовано 6 работ.

## Структура и объем диссертации

Работа состоит из введения, 6 глав, заключения и списка литературы, содержащего 52 наименования. Общий объем работы составляет 117 страниц.

## Краткое содержание диссертации

Во **введении** обосновывается актуальность работы, формулируется цель и задачи исследования.

В **главе 1** вводится необходимый понятийный и терминологический аппарат, дается определение графового описания, рассматриваются его свойства.

D-множеством называется подмножество  $\mathcal{P}$  декартова произведения  $A_1 \times A_2$  непустых непересекающихся алфавита  $A_1$  открывающих и алфавита  $A_2$  закрывающих скобочных символов, оно задает парность между скобками.

Пусть  $\Sigma$  — непустой алфавит терминальных символов, в который не входит символ  $\varepsilon$ . Графовым описанием (ГО) над алфавитом  $\Sigma$  называется семерка

$$G = (\Sigma, V, E, \mathcal{P}, \lambda, V_s, F),$$

где  $V$  — непустое множество вершин,  $E = \{(u, v) \mid u, v \in V\}$  — множество ориентированных дуг,  $\mathcal{P}$  — D-множество,  $\lambda : E \rightarrow (\Sigma \cup \{\varepsilon\}) \times (A_1 \cup A_2)$  — функция пометки дуги, значение которой есть пара из терминальной и скобочной пометок,  $V_s \in V$  — начальная вершина,  $F \subseteq V$  — множество заключительных вершин. Для каждого пути  $T$  в графе  $(V, E)$  определяются его терминальная пометка  $\omega(T)$  — цепочка в алфавите  $\Sigma$ , составленная из терминальных пометок соответствующих дуг, и скобочная пометка  $\pi(T)$  — цепочка в алфавите  $A_1 \cup A_2$ , аналогичным образом полученная из скобочных пометок дуг. Предложением графового описания  $G$  называется такой его путь из начальной вершины в одну из заключительных, скобочная пометка которого образует правильную скобочную систему в D-множестве  $\mathcal{P}$ . Пути в  $G$ , являющиеся фрагментами предложений, будем называть маршрутами. Множество всех предложений ГО  $G$  обозначим через  $Sentences(G)$ . Язык, задаваемый графовым описанием  $G$  есть множество всех цепочек в алфавите  $\Sigma$ , являющихся терминальными пометками предложений  $G$ , т.е.

$$L(G) = \{\omega(T) \mid T \in Sentences(G)\}.$$

Два графовых описания  $G_1$  и  $G_2$  называются эквивалентными, если  $L(G_1) = L(G_2)$ .

Таким образом, в графовом описании объединяется графовая структура состояний и переходов, подобная диаграмме конечного автомата и скобочная структура, определяемая парностью скобочных символов D-множества.

Отличие графового описания от D-графа состоит в выделении в явном виде пары  $(V, E)$  как основы графа и введении двойной пометки дуги. Кроме того, рассмотрение D-множеств с нейтральным элементом позволяет получить конечный автомат как простой частный случай графового описания.

На множестве маршрутов графового описания определяется отношение развития. Оно аналогично отношению развития, определенному для маршрутов D-графа и выражает возможность получения сложных маршрутов из более простых (т.е., их развития) посредством включения в них новых фрагментов.

Вводится понятие ядра графового описания. Ядро представляет собой конечное множество элементов графа, отвечающих за развитие всех более сложных предложений из конечного числа простых. Используется результат теории D-графов о том, что каждое достаточно сложное предложение может быть получено итеративным развитием элементов ядра (теорема о развитии предложений). Материалом для развития выступают элементарные предложения графа, называемые линиями, а также фрагменты предложений, называемые формантами.

Форманты бывают двух видов — форманты-циклы и форманты-гнезда.

Формант-цикл представляет собой циклический путь  $T$  в графе, скобочная пометка  $\pi(T)$  которого является правильной скобочной системой (иными, словами, является нейтральной). Это позволяет на основе произвольного числа итераций нейтрального цикла получать более сложные предложения из любых предложений, содержащих данный формант.

Формант-гнездо представляет собой тройку  $(T_1, T_2, T_3)$ , где  $T_2$  есть нейтральный путь, а  $T_1$  и  $T_3$  — циклические пути, скобочные пометки которых являются парными друг другу (поэтому и сами эти циклы называются парными). Это позволяет на основе произвольного числа согласованных (одновременных) итераций парных циклов получать более сложные предложения из любых предложений, содержащих данный формант.

Из теории D-графов заимствуется определение функции *reduction* для маршрутов, значение  $reduction(T)$  которой для маршрута  $T$  есть множество всех маршрутов, которые могут быть получены максимальным сокращением формантов в  $T$ .

Показывается (на основе соответствующего результата теории D-графов), что класс графовых описаний задает в точности класс бесконтекстных языков — теорема 1.1. Показывается также, что для графовых описаний сохраняется выводимость леммы Огдена о разрастании цепочек бесконтекстных языков как следствия из теоремы о развитии предложений.

**Определение.** Графовое описание  $G$  будем называть детерминированным, если для каждой вершины  $u \in V(G)$  множество дуг  $E' = \{(u, v) \in E(G)\}$  обладает следующим свойством: если  $E'$  содержит более одной дуги, то либо  $\forall e_1, e_2 \in E'$  выполняется  $\pi(e_1) \in A_1$ ,  $\pi(e_2) \in A_1$  и  $\omega(e_1) \neq \omega(e_2)$ , либо  $\forall e_1, e_2 \in E'$  выполняется  $\pi(e_1) \in A_2$ ,  $\pi(e_2) \in A_2$  и не существует  $a_1 \in A_1$  такого, что  $(a_1, \pi(e_1)) \in \mathcal{P}$  и  $(a_1, \pi(e_2)) \in \mathcal{P}$ .

**Теорема 1.2.** Класс языков, задаваемых детерминированными графовыми описаниями, совпадает с классом детерминированных бесконтекстных языков.

**Определение.** Графовым описанием с однозначной парностью будем называть ГО  $G$ , такое, что для любой его дуги  $e$ , такой что  $\pi(e) \in A_1$ , существует не более одной дуги  $e'$ , такой что  $\pi(e') \in A_2$  и  $(\pi(e), \pi(e')) \in \mathcal{P}$ .

**Теорема 1.3.** Класс языков, задаваемых графовыми описаниями с однозначной парностью, в точности совпадает с классом LL(1)-языков.

**Теорема 1.4.** Если ГО  $G$  не содержит формантов-гнезд, то язык  $L(G)$  является регулярным.

Памятью графового описания  $G$  будем называть пару  $\rho = (\text{end}(T), T'')$ , где  $T'' \in \text{reduction}(T)$  для некоторого префикса  $T$  предложения  $TT' \in \text{Sentences}(G)$ . Множество всех памятей ГО  $G$  будем обозначать как

$$\text{Mem}(G) = \{(\text{end}(T), T'') \mid \exists T' : TT' \in \text{Sentences}(G), T'' \in \text{reduction}(T)\}.$$

Память графового описания соответствует состоянию памяти эквивалентного магазинного автомата, обрабатывающего входную цепочку слева направо.

**Теорема 1.4** Множество  $\text{Mem}(G)$  всех памятей ГО  $G$  является регулярным.

В главе 2 рассматривается подкласс графовых описаний, задающих регулярные языки. Дается обзор существующих алгоритмов анализа конечных автоматов и построения эквивалентных регулярных выражений (С. Клини (1956), Д. Вуд (1987), Б. Ф. Мельников, А. А. Вахитова (1998)) Предлагается новый алгоритм GRAPHDESC-TO-REGEX преобразования графового описания, задающего регулярный язык, в регулярное выражение.

Этот алгоритм строит иерархический граф специального вида, соответствующий разбиению исходного графа в соответствии с рядом шабло-



нов, затем по иерархическому графу строит регулярное выражение, звездная высота которого соответствует глубине иерархического графа.

Шаблон *Section*, “секция” основан на анализе сильной связности графа. В каждой сильно связной компоненте (секции) графа выделяется дуга, удаление которой приводит к сокращению сильной связности секции. После удаления ее из секции алгоритм применяется рекурсивно к оставшимся частям секции.

Шаблон *Backsum*, “сумма обратных дуг” применяется к подграфу при наличии в нем дуг, ведущих из заключительных вершин в начальную, так, что в результирующем регулярном выражении для представления их совокупности (суммы) будет использована лишь одна операция звездочки.

Шаблон *Junction*, “точка сочленения” применяется к подграфу, имеющему точку сочленения, через которую проходят все его предложения. Разбиение графа в этой точке (и рекурсивное применение алгоритма к его частям) позволяет получить в результате более короткое выражение-конкатенацию.

**Теорема 2.1.** Алгоритм GRAPHDESC-TO-REGEXPR является корректным. Звездная высота получаемого регулярного выражения равна циклическое глубине иерархического графа.

**Теорема 2.2.** Алгоритм GRAPHDESC-TO-REGEXPR имеет временную сложность  $\Theta(|E|!)$ , где  $E$  есть множество дуг графового описания.

Производится сравнение предложенного алгоритма с существующими, показывающее уменьшение таких характеристик получаемого регулярного выражения, как звездная высота и длина. Так, для графового описания, состоящего из одной вершины и имеющего на ней циклы с пометками  $a_1, a_2, \dots, a_n$  все известные алгоритмы строят выражения вида  $a_1(a_2(\dots a_n^*)^* \dots)^*$  звездной высоты  $n$ , тогда как алгоритм GRAPHDESC-TO-REGEXPR строит выражение  $(a_1 + a_2 \dots + a_n)^*$  звездной высоты 1.

Рассмотрим также графовое описание  $G$  без циклов, имеющее точку сочленения, соединяющую два его фрагмента  $G_1$  и  $G_2$ , с числом предложений в них  $m_1$  и  $m_2$  соответственно. Число предложений ГО  $G$  есть  $m_1 m_2$ , и существующие алгоритмы строят регулярное выражение с числом слагаемых порядка  $m_1 m_2$ , тогда как алгоритм GRAPHDESC-TO-REGEXPR строит выражение с числом слагаемых порядка  $m_1 + m_2$ .

**Теорема 2.3.** В классе графовых описаний с попарно различными терминальными пометками на дугах алгоритм GRAPHDESC-TO-REGEXPR строит эквивалентное регулярное выражение минимально возможной звездной высоты, т.е. дает решение задачи о звездной высоте регулярного языка.

В **главе 3** рассматривается проблема регулярности для графовых описаний. R. Stearns (1967) и L. Valiant (1975) получили доказательства раз-

решимости проблемы регулярности для детерминированных бесконтекстных языков. Однако эти доказательства весьма громоздки, что заставляет искать более компактные доказательства. Подклассы детерминированных магазинных автоматов, задающих регулярные языки рассматривались А.А. Вылитком (1998). Решение проблемы регулярности для т.н. праволинейных детерминированных магазинных автоматов было предложено Л.И. Станевичене (2000).

В настоящей работе предлагается решение проблемы регулярности для подкласса LL(1)-языков, которое опирается на анализ памяти и формантов графовых описаний.

**Теорема 3.1.** Язык, задаваемый графовым описанием  $G$ , тогда и только тогда является регулярным, когда отношение эквивалентности на множестве памяти  $Mem(G)$  имеет конечный индекс.

**Теорема 3.2.** Для того, чтобы язык, задаваемый детерминированным графовым описанием  $G$  с однозначной парностью, являлся регулярным, необходимо и достаточно, чтобы в нем не было ни одного форманта-гнезда  $(T_1, T_2, T_3)$ , такого что одновременно  $\omega(T_1) \neq \Lambda$  и  $\omega(T_3) \neq \Lambda$ .

Есть основания полагать, что подобный подход может быть применен при решении проблемы регулярности в других классах языков.

Строится алгоритм REGULARITY проверки регулярности графового описания с однозначной парностью, доказывається его корректность, исследуется вычислительная сложность:

**Теорема 3.3.** Алгоритм REGULARITY является корректным.

**Теорема 3.4.** Алгоритм REGULARITY, применяемый к графовому описанию  $G$ , имеет временную сложность  $O(|\mathcal{P}|^{|E|})$ , где  $\mathcal{P}$  есть D-множество ГО  $G$ , а  $E$  — множество дуг  $G$ .

Описываются пути решения проблемы регулярности во всем классе детерминированных ГО. Так, в графовых описаниях этого класса наличие помеченных парных циклов не обязательно влечет нерегулярность соответствующего языка из-за возможности наличия в графе дополнительных циклов, позволяющих “разбалансировать” парность указанных циклов.

В **главе 4** рассматривается проблема эквивалентности для детерминированных графовых описаний. Дается обзор существующих решений. Доказательства разрешимости проблемы эквивалентности для детерминированных бесконтекстных языков были построены В. Мейтусом (1992) и G. Senizergues (1997), однако эти доказательства громоздки и содержат неясные места. Известны доказательства разрешимости проблемы эквивалентности для подклассов детерминированных бесконтекстных языков: Зубенко (1978), Романовский (1980), Непомнящая (1981), Мейтус (1989), Анисимов (1997), Станевичене (2000).

Понятие графа сопряжений широко используется при рассмотрении задач проверки эквивалентности для автоматов разных классов. В нем содержится информация, достаточная для отслеживания совместной работы автоматов над цепочками терминальных символов. В случае конечных автоматов граф сопряжения может быть преобразован в конечный автомат, что дает одно из доказательств замкнутости класса регулярных языков относительно операции пересечения.

Во всем классе бесконтекстных языков преобразование графа сопряжений в магазинный автомат невозможно в силу незамкнутости класса относительно операции пересечения, более того, известно (Станевичене, 1996), что сама задача построения графа сопряжений в классе детерминированных магазинных автоматов алгоритмически неразрешима.

**Теорема 4.2.** Задача построения графа сопряжения для детерминированных графовых описаний является алгоритмически неразрешимой.

Показывается разрешимость этой задачи в классе детерминированных ГО с однозначной парностью, предлагается алгоритм EQUIVALENCE ее решения и исследуются его свойства:

**Теорема 4.3.** Алгоритм EQUIVALENCE является корректным.

**Теорема 4.4.** Алгоритм EQUIVALENCE, применяемый к графовым описаниям  $G_1$  и  $G_2$  имеет временную сложность

$$O(\max(|\mathcal{P}_1| + |E_1|, |\mathcal{P}_2| + |E_2|)^{\max(|E_1|, |E_2|)}),$$

где  $\mathcal{P}_i$  есть D-множество графового описания  $G_i$ , а  $E_i$  — множество дуг  $G_i$ ,  $i = 1, 2$ .

Описываются пути расширения класса языков, для которого можно рассчитывать получить решение проблемы эквивалентности на основе графовых описаний. Так, ослабление требования однозначной парности циклов в формантах-гнездах приводит к возникновению более сложных вариантов “зацепления” циклов в рассматриваемых описаниях, однако есть основания полагать, что структура их будет обозрима и удастся отразить ее в графе сопряжений. Полное исследование этой проблемы выходит за рамки данной работы.

В **главе 5** решается задача построения полной системы эквивалентных преобразований для подкласса графовых описаний с однозначной парностью.

Преобразованием графовых описаний является любая их пара  $(G_1, G_2)$ . Преобразование  $(G_1, G_2)$  называется эквивалентным, если  $G_1 \sim G_2$ . Для множества (системы)  $T$  эквивалентных преобразований вводится понятие замыкания  $\hat{T}$ , определяемое следующим образом. Пара ГО  $(G'_1, G'_2)$  при-

надлежит  $\tilde{T}$  тогда и только тогда, когда существует последовательность

$$(G_1, G_2), (G_2, G_3), \dots, (G_{k-1}, G_k)$$

преобразований из  $T$ , для которой  $G'_1 = G_1$  и  $G'_2 = G_2$ . Система эквивалентных преобразований  $T$  называется полной в некотором классе графовых описаний, если любое эквивалентное преобразование графовых описаний этого подкласса принадлежит замыканию  $\tilde{T}$ .

Полная система эквивалентных преобразований строится на основе оригинального алгоритма EQUIVALENCE проверки эквивалентности через рассмотрение обратимых эквивалентных преобразований, состоящих в подстановке эквивалентных фрагментов в графовые описания.

Отметим, что сама идея построения полной системы с использованием алгоритма проверки эквивалентности была высказана (устно) Р.И. Подловченко в 2010 году.

Вводится понятие фрагмента графового описания, частным случаем которого является графовое описание, и рассматриваются те пары эквивалентных фрагментов, которые возникают в ходе работы алгоритма EQUIVALENCE. Каждая пара эквивалентных фрагментов соответствует системе эквивалентных преобразований, получаемых подстановкой одного фрагмента вместо другого.

В процессе анализа алгоритма EQUIVALENCE получены следующие классы эквивалентных фрагментов.

Класс *Linear*, “линейные” содержит все пары линейных фрагментов графовых описаний, имеющих бесцикловую структуру.

Класс *NeutralCycle*, “нейтральный цикл” содержит все пары фрагментов, содержащих формант-цикл, и получаемые друг из друга вынесением (или, в обратную сторону, включением) в нейтральный цикл эквивалентного линейного фрагмента, а также итерацией циклов, кратность которой определяется наименьшим общим кратным для соответствующих пометок.

Класс *PairCycles*, “парные циклы”, содержат все пары фрагментов, содержащих формант-гнездо, и получаемых друг из друга аналогичным вынесением и включением линейных маршрутов, а также аналогичной кратной итерацией.

На основе эквивалентных преобразований, соответствующим указанным классам фрагментов, строится система  $T_0$ .

**Теорема 5.1** Система  $T_0$  эквивалентных преобразований графовых описаний является полной в классе детерминированных графовых описаний с однозначной парностью.

В **главе 6** описываются области практического применения результатов работы.

Исследуются пути применения полученных алгоритмов в построении трансляторов и их генераторов. При автоматизированном построении оптимальных трансляторов по языковому описанию часто применяется метод разбиения языковых описаний на части и оптимизация соответствующих компонентов трансляторов по отдельности. В качестве одного из шагов оптимизации очередного описания бесконтекстного языка может быть применен алгоритм проверки регулярности соответствующего языка, с последующей генерацией для этого описания эквивалентного конечного автомата.

Другое актуальное направление — это собственно теоретические исследования. При построении качественных доказательств в ходе научного поиска требуется подтверждать гипотезы убедительными закономерностями или опровергать их точными контрпримерами. Однако помимо чисто творческой составляющей научного поиска, необходимо учитывать и техническую. Даже простейшие образцы объектов, обладающие интересующими исследователя свойствами, могут иметь высокую сложность (размер, количество элементов), и ручная работа с ними займет значительное время и будет подвержена ошибкам. В связи с этим актуальной представляется задача создания инструментария, многократно усиливающего возможности исследователя по работе с интересующими его объектами. Необходимо, чтобы вместо выписывания на бумаге или на доске однодвух характерных примеров, исследователь имел возможность сформировать лаконичный запрос к программе-генератору и получить десятки примеров с нужными ему свойствами. Необходимо, чтобы вместо ручной проверки новых свойств, точно еще не сформулированных, исследователь мог легко написать на подходящем языке программирования прототип проверяющей функции и в ходе испытаний на сгенерированных примерах оперативно её уточнять.

Третье направление, для которого существует потребность в инструментальной поддержке — это образовательные задачи. Для облегчения усвоения материала по многим разделам теории формальных языков необходима наглядная демонстрация. Для этих целей широко используются современные технические средства (компьютерный проектор, интерактивная доска, и т.д.). Однако подготовка соответствующих демонстрационных материалов, даже с использованием на первый взгляд подходящего программного обеспечения (такого как векторный редактор), может быть по-прежнему сопряжена со значительным объемом ручной работы, связанной с проверкой корректности материалов с точки зрения конкретной области теории, к которой относится создаваемый пример. Особенно это заметно в случаях, когда требуется создать не одну иллюстрацию, а целую их серию, например, для пошаговой демонстрации процесса синтаксиче-

ского анализа. Использование программного средства общего назначения, не учитывающего специфику задачи, может потребовать большого объема рутинной ручной работы и привести к незамеченным ошибкам.

В заключениях сформулированы основные результаты работы, представлена информация о публикациях и апробации.

## Публикации по теме диссертации

1. Сутырин П.Г. Алгоритм построения регулярного выражения по конечному автомату // Сборник тезисов лучших дипломных работ 2006 года. — М.: Издательский отдел Факультета ВМиК МГУ, 2006. — С. 94–95.
2. Сутырин П.Г., Мальковский М.Г. Иерархические конечные автоматы и звездная высота регулярных выражений // Программные системы и инструменты: Тематический сборник / Под ред. Королева Л.Н. — М.: Издательский отдел ф-та ВМиК МГУ, 2008. № 9. — С. 32–49.
3. Мальковский М.Г., Сутырин П.Г. О задачах преобразования описаний формальных языков // Сборник научных трудов международной научно-практической конференции "Современные направления теоретических и прикладных исследований'2009". — Одесса: Изд-во Черноморье, 2009. Том 2. — С. 15–18.
4. Сутырин П.Г. Об инструментальных средствах теории формальных языков // Сборник научных трудов по материалам международной научно-практической конференции "Научные исследования и их практическое применение. Современное состояние и пути развития '2009". — Одесса: Изд-во Черноморье, 2009. Том 16. — С. 7–9.
5. Вылиток А.А., Сутырин П.Г., Харченков С.Л., Юдочев Д.В. О сопряжениях графовых описаний формальных языков // Программные системы и инструменты. Тематический сборник / Под ред. Королева Л.Н. — М.: Издательский отдел Факультета ВМиК МГУ, 2009. № 10. — С. 91–99.
6. Сутырин П.Г. О множестве памяти D-графа // Вестн. Моск. Ун-та, Сер. 15, Вычисл. матем. и кибернетика, 2010. № 1. — С. 45–51.