Применение регулярных выражений для обработки текстовых данных

С. В. Козлов, А. В. Светлаков

***Аннотация* – В статье описывается применение регулярных выражений при решении задач синтаксического и лексического анализа. Приводится понятие регулярного выражения, кратко описывается его сущность. Авторами ставятся три основные задачи использования регулярных выражений в программных приложениях. Первая из них заключается в проверке текстовых сообщений на соответствие заданному шаблону поля ввода. Решение этой задачи позволяет верифицировать данные и систематизировать их в информационной системе в единообразной форме. Вторая задача состоит в анализе блоков текста при вводе в них данных. Решение этой задачи позволяет выявить фрагменты текста, введенные с ошибками, исследовать их и произвести соответствующую замену по заданным правилам. Третья задача определяет направление использования регулярных выражений при написании трансляторов в современных инструментальных средах. Ее решение открывает возможности разработки интерпретаторов и частотных словарей для лексического и синтаксического анализа текста. Для каждой из описываемых задач приводятся соответствующие примеры компьютерных программ. Авторы демонстрируют реализацию регулярных выражений в программном коде, написанном на языке программирования C#, собственных разработанных приложений анализа текстовых данных. Актуальность статьи связана с изучением методов синтаксического и лексического анализа информационных потоков в системах распознавания текстовых образов, которые эффективно применяются как инструменты искусственного интеллекта.**

***Ключевые слова* – регулярное выражение, лексический анализ, синтаксический анализ, паттерн, текст, символ, метасимвол, валидация.**

I. ВВЕДЕНИЕ

В настоящее время в связи с развитием IT-технологий многие методы теоретической математики находят свое применение в повседневной практике цифровизации систем жизнедеятельности человека [1, 2, 3]. Одним из таких инструментов выступают регулярные выражения. Разнообразное их использование становится все более широким с каждым днем, они эффективно внедряются как средства анализа текстовых потоков данных [4, 5]. При этом ввиду роста количества информационных систем, основным видом хранения сведений в которых является текст и взаимодействие осуществляется с помощью текстовых сообщений, автоматизированная обработка в них данных такого вида выходит на первый план. Регулярные выражения приобретают характер не только востребованного практикой функционального инструмента, но и уже в некоторых ситуациях незаменимого средства автоматического анализатора.

Так в области функционального программирования, инструменты которого широко применяются в искусственном интеллекте, регулярные выражения наиболее распространены как средство написания различных трансляторов [6, 7]. При этом, так как информационные потоки изначально представляют собой текстовые данные, они используются для всестороннего их анализа [8, 9]. В самом простом случае регулярные выражения позволяют классифицировать данные по типу [10]. Например, они дают возможность отнести числовые данные к целому или вещественному типу, определить их подтип. В случае текстовых сообщений можно построить разные классы символов. Такие, как класс букв, класс цифр, класс знаков препинания, класс символов специального назначения и другие знаки. Таким образом, при вводе данных в компьютерных [11] и мобильных [12] приложениях это открывает

Статья получена 12 июня 2022.

Козлов Сергей Валерьевич, Смоленский государственный университет, доцент кафедры прикладной математики и информатики, кандидат педагогических наук, доцент (email: svkozlov1981@yandex.ru)

Светлаков Алексей Владимирович, Смоленский государственный университет, студент физико-математического факультета (email: seferlian@mail.ru)

новые перспективы проверки корректности записей.

Отметим, что это не только отслеживание правильного формата параметров при заполнении полей ввода настольных программ и web-форм. В настоящее время регулярные выражения выступают инструментом верификации информации разного вида [13, 14], которая хранится как в базах данных информационной системы, с которой осуществляется работа, так и при реализации параллельных запросов в распределенные системы, связанные с ней [15, 16]. Это позволяет избежать множества ошибок, идентифицирующих записи пользователей в единых информационных средах, а, следовательно, увеличить эффективность работы с данными, оперируемыми в них.

Далее остановимся более подробно на задачах анализа данных, которые успешно можно решить в программных приложениях с использованием регулярных выражений.

II. ОСНОВНЫЕ ЗАДАЧИ ПРИМЕНЕНИЯ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ ПРИ ОБРАБОТКЕ ТЕКСТОВЫХ ДАННЫХ

На сегодняшний день регулярные выражения широко применяются в программировании как самостоятельный элемент при работе с символьным и строковым типом данных. Они, как правило, входят в базовый состав почти всех языков программирования. Это и не удивительно, поскольку с помощью регулярных выражений решают три основные задачи:

1) Проверка на соответствие нужному шаблону. Например, чтобы данные, вводимые пользователем, соответствовали требованиям.

2) Поиск, анализ и замена фрагментов текста.

3) Лексический и синтаксический анализ. Регулярные выражения упрощают написание трансляторов.

Первая задача активно используется в различных системах валидации [17]. Например, при регистрации пользователя на сетевых ресурсах.

Вторая задача представлена поиском фрагментов текста [18, 19], а, следовательно, поиском информации в тексте; статистическим анализом в автоматическом режиме, в частности, составление частотных словарей; заменой определенных фрагментов. Последнее может использоваться в html-тексте для замены тегов.

Третья задача более специфическая. Во-первых, регулярные выражения участвуют в системах проверки орфографии и синтаксиса естественного языка [20, 21]. Во-вторых, они могут использоваться и при проектировании трансляторов к языкам программирования [22, 23], особенно когда лексемы сложны с точки зрения их классического разбора.

Сами по себе регулярные выражения на профессиональном уровне изучаются в курсе математической лингвистики или в смежных с ней дисциплинах. Не будем останавливаться подробно на научном понятии регулярного выражения, опишем указанный объект менее формальным языком.

Регулярное выражение – это строка специального вида, состоящая из обычных символов и метасимволов. Обычные символы в регулярном выражении представляют сами себя. Метасимволы или символы-джокеры используются для представления других символов, группы символов, а также для записи выполнения специальных команд. Строка-образец в виде регулярного выражения обычно называется паттерном.

Классически выделяют следующие символы-джокеры: [,], \, /, ^, $, ., |, ?, *, +, (,), {, }. Для того, чтобы эти символы представляли сами себя, их необходимо экранировать метасимволом "\": например, записи \^ будет соответствовать символ ^. Существуют и другие метасимволы, а также метасимвольные последовательности, например, конструкция {3, 8} обозначает запись предыдущего подвыражения от 3 до 8 раз включительно.

Таким образом, совокупность регулярных выражений представляет собой некоторый метаязык, предназначенный для описания других языков.

В содержательных примерах будем использовать расширенные регулярные выражения, то есть регулярные выражения, включающие в себя бэкрефенсы и утверждения нулевой ширины. Такие регулярные выражения распознают подстроки в тексте, относящиеся не только к регулярным языкам, но и к контекстно-свободным языкам, и даже к контекстно-зависимым [24, 25]. Например, регулярное выражение *(?=(a(?-1)?b)c)a+(b(?-1)?c)* распознает строки вида $a^n b^n c^n$, а это контекстно-зависимый язык. Это можно проверить по лемме о накачке для контекстно-свободных языков.

Подробно рассмотрим применение регулярных выражений для решения каждой из описанных задач. При этом отметим, что в различных языках программирования использование регулярных выражений может отличаться, хотя синтаксис самих выражений примерно одинаков (используется либо POSIX-стандарт, либо PCRE). Так, например, JavaScript или PHP не требуют дополнительных манипуляций для работы с регулярными выражениями, а C# требует дополнительного

подключения специальной библиотеки System.Text.RegularExpressions.

III. СИСТЕМА ВАЛИДАЦИИ ПРИ РЕГИСТРАЦИИ

Часто при регистрации пользователей на сайте от них требуют соблюдения условий при вводе данных. Например, таковыми могут быть корректный адрес почты, пароль должен содержать не менее 6 символов и т. д. Это обычно решается с помощью регулярных выражений. На рисунке 1 показана реализованная на C# программа, которая проверяет корректность e-mail-а и пароля, требования к паролю написаны на рисунке.

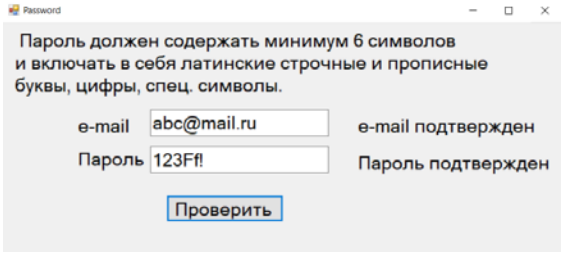

Рис. 1. Валидация формы

Используемое регулярное выражение для проверки e-mail-а представляет собой следующую запись: *^(([0-9A-Za-z]{1}[-0-9A-z\.]{1,}[0-9A-Za-z]{1})|([0-9А-Яа-я]{1}[-0-9А-я\.]{1,}[0-9А-Яа-я]{1}))@([-A-Za-z]{1,}\.){1,2}[-A-Za-z]{2,})$*. Разберем его более подробно:

^(- метасимвол привязки к началу строки

(

(- эта группа отвечает за логин латиницей

[0-9A-Za-z]{1} – 1-й символ только цифра или буква

[-0-9A-z\.]{1,} - в середине минимум один символ

[0-9A-Za-z]{1} - последний символ только цифра или буква

)

| - метасимвол альтернативного выбора

(- этот блок отвечает за логин кирилицей

[0-9А-Яа-я]{1} – 1-й символ только цифра или буква

[-0-9А-я\.]{1,} - в середине минимум один символ

[0-9А-Яа-я]{1} - последний символ только цифра или буква

)

)

@ - обязательное наличие собаки разделяющей логин от домена

(

[-0-9A-Za-z]{1,} - блок может состоять из дефисов, цифр и букв (не менее 1 символа)

\. - наличие точки в конце блока (экранированный символ)

){1,2} - допускается от 1 до 2 блоков по вышеукащанному паттерну

[-A-Za-z]{2,} - блок описывайющий домен вехнего уровня (ru, com, net) (не менее 2 символов)

)$ - метасимвол привязки к концу строки.

Примеры корректных адресов: 123456@i.ru, 123456@ru.name.ru, логин-1@i.ru, 1login@ru.name.ru. Примеры некорректных адресов: login_@i.ru, логинlogin@i.ru, @123456@i.ru, 123456@ru.name.ru.ua.

Для проверки пароля использовалось регулярное выражение вида: *^(?=.*[0-9])(?=.*[!@#$%^&*?+().,:;-])(?=.*[a-z])(?=.*[A-Z])[0-9a-zA-Z!@#$%^&*?+().,:;-]{6,}$*. Здесь используется предварительная проверка на соответствие нужным символам (утверждения нулевой ширины). Например, (?=.*[0-9]) проверяет в цепочке, есть ли хотя бы одна цифра в поле. А также цепочка должна содержать не менее 6 символов: [0-9a-zA-Z!@#$%^&*?+().,/:;-]{6,}.

Примеры некорректных паролей: 1Dd$2, 123123, 2d%dsdf43.

IV. ОБРАБОТКА ТЕКСТА НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Нередко пользователь допускает те или иные ошибки при вводе текста. Исправить некоторые из них помогает автозамена, которая работает в автоматическом режиме. Нередко регулярные выражения используются и здесь.

Программа для обработки текста и результаты ее работы представлена на рисунке 2.

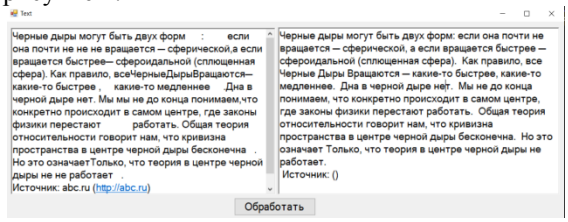

Рис. 2. Программа обработки текста

Используемые регулярные выражения можно увидеть из следующего программного кода на C#:

```
pattern0 = @" +";//удаляет множественные пробелы
buf = Regex.Replace(buf, pattern0, " ");
pattern1 = @" ([,\.!?;:])";//лишний пробел перед знаком
buf = Regex.Replace(buf, pattern1,"$1");
pattern1 = @"(?<! )(—)";//или отсутствие пробела перед тире
buf = Regex.Replace(buf, pattern1, " $1");
pattern2 = @"([,\.!?;:—])([^\.!?])";//отсутствие пробела после знака
buf = Regex.Replace(buf, pattern2, "$1 $2");
```

```
pattern3 = @"(?<!^)([А-ЯЁ])";//добавляет пробел перед большой буквой
buf = Regex.Replace(buf, pattern3, " $1");
pattern4 = @"([fh]t{1,2}ps?:\/\/)?[\w-]+\.\w{2,4}";//удаляет спам
buf = Regex.Replace(richTextBox1.Text, pattern4, "");
pattern5 = @"(\b\S+)(?:\s+\1\b)+";//удаляет дубликаты
buf = Regex.Replace(buf, pattern5, "$1",RegexOptions.IgnoreCase);
richTextBox2.Text = buf;
```

Паттерны 0-3 достаточно просты в своем использовании. Отметим лишь то, что здесь при замене текста, а именно применении метода Replace, используется группа регулярного выражения вне этого выражения. Она записывается в виде $1. Это необходимо, чтобы весь найденный шаблон заменить на его часть.

Паттерн 4 удаляет спам в виде ссылок на какие-либо ресурсы в интернете. Заметим, что удаление происходит вне зависимости, содержится ли протокол http или https в записи ссылки или нет.

Паттерн 5 находит и удаляет дублирующие друг друга слова. Для этого были использованы бэкреференс \1.

Отметим, что подобным образом регулярные выражения в данном виде можно использовать для автоматической модерации сообщений в чатах или форумах.

V. ДРУГИЕ ПРИЛОЖЕНИЯ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

Нередко в коде программы, предназначенной для создания частотного словаря, можно встретить регулярное выражение, похожее на *[„.!?(){}+*\-;:"\[\]\|]+*. Это регулярное выражение находит пробелы и все знаки в тексте или последовательность таких знаков. Если вместе с этим регулярным выражением использовать метод *Split*, то произойдет запись отдельных слов в массив, разделенных символами, указанными в регулярном выражении. Таким образом, регулярные выражения полезны и для написания программ для статистического анализа текста.

В статье [26] описан разработанный исполнитель. В рамках регулярных выражений особый интерес представляет цветовая лексема, записываемая в HEX-формате. Для проверки соответствия указанному шаблону в интерпретаторе данного исполнителя использовалось регулярное выражение *^\#[0-9A-F]{6}$*. Заметим, что оба указанные случая (частотный словарь и интерпретатор) по своей сути являются лексическим анализом текста.

Приведем похожий пример. Допустим, что в тексте надо найти спрятанные буквы английского алфавита и выделить их, заодно пометив все слово, в котором эта буква содержится, а также необходимо выделить информацию, заключенную в скобки (при этом сами скобки не трогая). Такая разработанная программа представлена на рисунке 3.

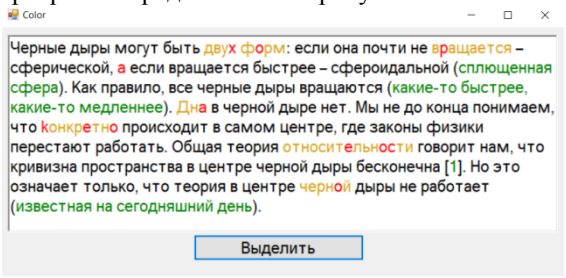

Рис. 3. Программа для выделения текста

Приведем часть кода программы, включающую в себя регулярные выражения, а также фрагмент, реализующий выделение текста в скобках.

```
pattern[0] = @"\[.*?\]|\(.*?\)";
pattern[1] = @"\b[^ ]*?[a-z]+?.*?\b";
pattern[2] = @"[a-z]+?";
matches[0] = Regex.Matches (richTextBox1.Text, pattern[0]);
 if (matches[0].Count!=0)
  {it = 0;
   ind = new int[matches[0].Count];
   for (int i = 0; i < matches[0].Count; i++)
     {ind[i] = richTextBox1.Text.IndexOf (Convert.ToString(matches[0][i]), it);
      it = ind[i] + Convert.ToString (matches[0][i]).Length;
      richTextBox1.Select(ind[i]+1, Convert.ToString (matches[0][i]).Length-2);
      richTextBox1.SelectionColor = Color.Green;
     }
  }
```

Регулярные выражения достаточно просты для разбора. Отметим лишь то, что в них используются ленивые квантификаторы [27] для того, чтобы поиск не захватывал лишние фрагменты.

Логика выделения текста в скобках такая: осуществляется поиск всех вхождений по данному регулярному выражению, затем находится индекс каждого вхождения, после чего выделяется текст, начиная с этого индекса с учетом длины найденного вхождения, и этот текст окрашивается зеленым цветом. В массив *ind* и записываются индексы каждого вхождения. *Matches* – это массив коллекций, где первый индекс отвечает за найденную коллекцию, а второй – за каждое найденное вхождение. Остальные цветовые выделения реализованы похожим образом.

Разобранный пример относится к синтаксическому анализу текста, и на основе этих же механизмов может осуществляться подсветка синтаксиса в коде программы, например, в системах проверки текста на заимствования.

Таким образом, регулярные выражения могут использоваться как вспомогательные инструменты при разборе текста, будь то естественный или искусственный язык.

VI. ЗАКЛЮЧЕНИЕ

Подводя общий вывод, отметим, что регулярные выражения стали неотъемлемым инструментом в программировании – это показывают вышеприведенные примеры – и распространены уже настолько широко, что им обучают даже за пределами математической лингвистики, где они изначально и появились. Регулярные выражения используются во многих современных информационных системах при регистрации, авторизации и аутентификации пользователей. Они позволяют унифицировать ввод данных, исключить определенные коллизии в записях баз данных информационных сред, что дает возможность оптимизировать работу в них.

При этом применение регулярных выражений для синтаксического и лексического анализа текстовых сообщений выступает инструментом не только их идентификации шаблону и статистического анализа. Их всестороннее внедрение в современные программные среды сделало более доступными средства интеллектуального анализа потоков информации. Они открывают перспективы создания таких информационных программных комплексов, которые совместно с основным своим функционалом, верифицируют потоки данных. Это устраняет на начальном этапе многочисленные неверные и неопределенные записи, дублирование их, что закладывает долгосрочную основу использования таких программных сред.

Кроме того, как инструмент интеллектуального анализа данных регулярные выражения в системах распознавания образов позволяют классифицировать информационные потоки, строить классы эквивалентности. В дальнейшем это становится средством обучения информационной системы действиям, связанным с той или иной выявленной ситуацией, формированием автоматизированной реакции на запросы пользователя. Таким образом, регулярные выражения становятся действенным и востребованным инструментом исследования потоков текстовой информации.

БИБЛИОГРАФИЯ

[1] Козлов С. В. Использование функциональных возможностей информационных систем в производственной сфере // ЭНЕРГЕТИКА, ИНФОРМАТИКА, ИННОВАЦИИ – 2017 (электроэнергетика, электротехника и теплоэнергетика, математическое моделирование и информационные технологии в производстве). Сборник трудов VII-ой Международной научно-технической конференции. – 2017. – В 3 т. Т 1. – С. 298-301.

[2] Андреев К. В., Быков А. А., Киселева О. М. Математическая модель предиктивного кодирования радиотехнических сигналов, основанная на алгоритме изменяющегося шага кодирования // Современные наукоемкие технологии. 2020. – № 11-2. – С. 261-267.

[3] Муха В. С. Математические модели многомерных данных // Доклады Белорусского государственного университета информатики и радиоэлектроники. – 2014. – № 2 (80). – С. 143-158.

[4] Втюрин М. В. Применение формальных грамматик для сокращения объема текстовой информации // Инновационное развитие: технический и технологический аспекты. Сборник статей международной научно-практической конференции. – 2019. – С. 22-25.

[5] Кагиров И. А., Леонтьева А. Б. Автоматический синтаксический анализ русских текстов на основе грамматики составляющих // Известия высших учебных заведений. Приборостроение. – 2008. – Т. 51. № 11. – С. 47-51.

[6] Волкова И. А., Вылиток А. А., Руденко Т. В. Формальные грамматики и языки. Элементы теории трансляции: учебное пособие для студентов II курса. – М., 2009 – 115 с.

[7] Компиляторы. Принципы, технологии, инструментарий / А. В. Ахо, М. С. Лам, Р. Сети, Д. Д. Ульман. – М., 2008. – 1184 с.

[8] Козлов С. В., Светлаков А. В. Теория формальных грамматик и ее применение // Системы компьютерной математики и их приложения. – 2021. – № 22. – С. 358-364.

[9] Янченко Е. В. Использование формальных грамматик в криптографии // Современные проблемы телекоммуникаций: материалы международной научно-технической конференции. – Новосибирск, 2021. – С. 155-158.

[10] Байдарманова Б. Н. Некоторые способы нахождения эквивалентных преобразований в контексте свободных грамматик // Theoretical & Applied Science. – 2013. – № 5 (1). – С. 5-11.

[11] Лебедева Е. А., Козлов С. В. Содержание и особенности разработки учебно-методического проекта по математике «Системы линейных уравнений» в среде программирования C# // Развитие научно-технического творчества детей и молодежи: сборник материалов III

Всероссийской научно-практической конференции с международным участием. – 2019. – С. 161-166.

[12] Синякова Н. Д., Козлов С. В. Применение web-сервисов в образовании // Прикладная математика и информатика: современные исследования в области естественных и технических наук. – Тольятти: Тольяттинский государственный университет. 2020. – С. 977-982.

[13] Фаворская М. Н. К вопросу об использовании формальных грамматик при распознавании объектов в сложных сценах // Решетневские чтения. – 2009. – Т. 2. – С. 540-541.

[14] Борисенкова А. В., Козлов С. В. Использование метода каскадов Хаара при распознавании образов на изображениях // Развитие научно-технического творчества детей и молодежи: Сборник материалов III Всероссийской научно-практической конференции с международным участием. – 2019. – С. 28-33.

[15] Мунерман В. И. Реализация параллельной обработки данных в облачных системах // Современные информационные технологии и ИТ-образование. – 2017. Т. 13. № 2. – С. 57-63.

[16] Макаров А. И., Миронов А. И., Мунерман В. И. Реализация параллелизма на уровне задач в системах высокой доступности // Системы высокой доступности. – 2018. – Т. 14. № 5. – С. 42-45.

[17] Короткова А. Ю. Регулярные выражения во главе шаблонов поиска и отбора // Информационные технологии в образовании: материалы X Всероссийской научно-практической конференции. – 2018. – С. 167-170.

[18] Пруцков А. В., Сусанина И. В. Практическое применение функционального программирования и регулярных выражений в библиометрическом анализе // International Journal of Open Information Technologies. – 2022. – Т. 10. № 5. – С. 63-68.

[19] Дубова А. А. Поиск данных с использованием регулярных выражений // В сборнике: Международная научно-техническая конференция молодых ученых БГТУ им. В.Г. Шухова. Посвящена 165-летию В.Г. Шухова. – Белгород, 2018. – С. 3881-3885.

[20] Шевелёва К. В., Авдеев Н. Н. Применимость регулярного выражения как математической модели орфографической ошибки // Актуальные проблемы прикладной математики, информатики и механики: сборник трудов Международной научной конференции. – 2019. – С. 335-340.

[21] Груздев Д. Ю., Макаренко А. С. Объектно-ориентированный язык программирования и регулярные выражения в практике письменного переводчика // Успехи гуманитарных наук. 2019. – № 8. – С. 146-153.

[22] Романюк Б.В. к вопросу о применении регулярных выражений // В сборнике: Проблемы информационной безопасности социально-экономических систем. VIII Всероссийская с международным участием научно-практическая конференция. – Симферополь, 2022. – С. 70-71.

[23] Козлов С. В., Светлаков А. В. О LL(1)-грамматиках, алгоритмах на них и методах их анализа в программировании // International Journal of Open Information Technologies. – 2022. Т. 10. № 3. – С. 30-38.

[24] Скрипов А. В. Описание контекстных условий формальных языков грамматики с контекстуальными аргументами // Вестник Уральского института экономики, управления и права. – 2013. № 1 (22). – С. 111-116.

[25] Мартыненко Б. К. Регулярные языки и КС-грамматики // Компьютерные инструменты в образовании. – 2012. № 1. – С. 14-20.

[26] Светлаков А. В., Бахман В. А., Бодю В. Ю. Визуализация технологического маршрута с помощью графического исполнителя // Потенциал инновационного развития Российской Федерации в новых геополитических условиях: сборник статей Национальной (Всероссийской) научно-практической конференции. – Уфа, 2021. – С. 40-45.

[27] Кулюкин К. С. Особенности задачи упрощения квантификаторов в регулярных выражениях // В книге: Конкурс научно-исследовательских работ студентов Волгоградского государственного технического университета. Тезисы докладов. Редколлегия: С.В. Кузьмин (отв. ред.) [и др.]. – 2020. – С. 171-172.

[28] Андрианов И. А., Григорьева А. Н. Модернизация индекса для поиска по регулярным выражениям // Системы управления и информационные технологии. – 2020. – № 2 (80). – С. 60-64.

Using regular expressions to process text data

S.V. Kozlov, A. V. Svetlakov

***Abstract* – This article describes the use of regular expressions in solving syntactic and lexical analysis problems. The concept of a regular expression is given, its essence is briefly described. The authors set three main tasks of using regular expressions in software applications. The first of these is to check text messages for compliance with a given input field template. The solution of this problem allows you to verify the data and systematize them in the information system in a uniform form. The second task is to analyze blocks of text when entering data into them. The solution to this problem allows you to identify text fragments entered with errors, examine them and make an appropriate replacement according to the specified rules. The third task determines the direction of using regular expressions when writing translators in modern instrumental environments. Its solution opens up the possibility of developing interpreters and frequency dictionaries for lexical and syntactic text analysis. For each of the described tasks, corresponding examples of computer programs are given. The authors demonstrate the implementation of regular expressions in program code written in the C# programming language, their own developed text data analysis applications. The relevance of the article is related to the study of methods for syntactic and lexical analysis of information flows in text pattern recognition systems, which are effectively used as artificial intelligence tools.**

***Keywords* – regular expression, lexical analysis, parsing, pattern, text, symbol, metacharacter, validation.**

REFERENCES

[1] Kozlov S. V. Ispol'zovanie funkcional'nyh vozmozhnostej informacionnyh sistem v proizvodstvennoj sfere // JeNERGETIKA, INFORMATIKA, INNOVACII – 2017 (jelektrojenergetika, jelektrotehnika i teplojenergetika, matematicheskoe modelirovanie i informacionnye tehnologii v proizvodstve). Sbornik trudov VII-oj Mezhdunarodnoj nauchno-tehnicheskoj konferencii. – 2017. – V 3 t. T 1. – S. 298-301.

[2] Andreev K. V., Bykov A. A., Kiseleva O. M. Matematicheskaja model' prediktivnogo kodirovanija radiotehnicheskih signalov, osnovannaja na algoritme izmenjajushhegosja shaga kodirovanija // Sovremennye naukoemkie tehnologii. 2020. – # 11-2. – S. 261-267.

[3] Muha V. S. Matematicheskie modeli mnogomernyh dannyh // Doklady Belorusskogo gosudarstvennogo universiteta informatiki i radiojelektroniki. – 2014. – # 2 (80). – S. 143-158.

[4] Vtjurin M. V. Primenenie formal'nyh grammatik dlja sokrashhenija ob"ema tekstovoj informacii // Innovacionnoe razvitie: tehnicheskij i tehnologicheskij aspekty. Sbornik statej mezhdunarodnoj nauchno-prakticheskoj konferencii. – 2019. – S. 22-25.

[5] Kagirov I. A., Leont'eva A. B. Avtomaticheskij sintaksicheskij analiz russkih tekstov na osnove grammatiki sostavljajushhih // Izvestija vysshih uchebnyh zavedenij. Priborostroenie. – 2008. – T. 51. # 11. – S. 47-51.

[6] Volkova I. A., Vylitok A. A., Rudenko T. V. Formal'nye grammatiki i jazyki. Jelementy teorii transljacii: uchebnoe posobie dlja studentov II kursa. – M., 2009 – 115 s.

[7] Kompiljatory. Principy, tehnologii, instrumentarij / A. V. Aho, M. S. Lam, R. Seti, D. D. Ul'man. – M., 2008. – 1184 s.

[8] Kozlov S. V., Svetlakov A. V. Teorija formal'nyh grammatik i ee primenenie // Sistemy komp'juternoj matematiki i ih prilozhenija. – 2021. – # 22. – S. 358-364.

[9] Janchenko E. V. Ispol'zovanie formal'nyh grammatik v kriptografii // Sovremennye problemy telekommunikacij: materialy mezhdunarodnoj nauchno-tehnicheskoj konferencii. – Novosibirsk, 2021. – S. 155-158.

[10] Bajdarmanova B. N. Nekotorye sposoby nahozhdenija jekvivalentnyh preobrazovanij v kontekste svobodnyh grammatik // Theoretical & Applied Science. – 2013. – # 5 (1). – S. 5-11.

[11] Lebedeva E. A., Kozlov S. V. Soderzhanie i osobennosti razrabotki uchebno-metodicheskogo proekta po matematike «Sistemy linejnyh uravnenij» v srede programmirovanija C# // Razvitie nauchno-tehnicheskogo tvorchestva detej i molodezhi: sbornik materialov III Vserossijskoj

nauchno-prakticheskoj konferencii s mezhdunarodnym uchastiem. – 2019. – S. 161-166.
[12] Sinjakova N. D., Kozlov S. V. Primenenie web-servisov v obrazovanii // Prikladnaja matematika i informatika: sovremennye issledovanija v oblasti estestvennyh i tehnicheskih nauk. – Tol'jatti: Tol'jattinskij gosudarstvennyj universitet. 2020. – S. 977-982.
[13] Favorskaja M. N. K voprosu ob ispol'zovanii formal'nyh grammatik pri raspoznavanii ob"ektov v slozhnyh scenah // Reshetnevskie chtenija. – 2009. – T. 2. – S. 540-541.
[14] Borisenkova A. V., Kozlov S. V. Ispol'zovanie metoda kaskadov Haara pri raspoznavanii obrazov na izobrazhenijah // Razvitie nauchno-tehnicheskogo tvorchestva detej i molodezhi: Sbornik materialov III Vserossijskoj nauchno-prakticheskoj konferencii s mezhdunarodnym uchastiem. – 2019. – S. 28-33.
[15] Munerman V. I. Realizacija parallel'noj obrabotki dannyh v oblachnyh sistemah // Sovremennye informacionnye tehnologii i IT-obrazovanie. – 2017. T. 13. # 2. – S. 57-63.
[16] Makarov A. I., Mironov A. I., Munerman V. I. Realizacija parallelizma na urovne zadach v sistemah vysokoj dostupnosti // Sistemy vysokoj dostupnosti. – 2018. – T. 14. # 5. – S. 42-45.
[17] Korotkova A. Ju. Reguljarnye vyrazhenija vo glave shablonov poiska i otbora // Informacionnye tehnologii v obrazovanii: materialy X Vserossijskoj nauchno-prakticheskoj konferencii. – 2018. – S. 167-170.
[18] Pruckov A. V., Susanina I. V. Prakticheskoe primenenie funkcional'nogo programmirovanija i reguljarnyh vyrazhenij v bibliometricheskom analize // International Journal of Open Information Technologies. – 2022. – T. 10. # 5. – S. 63-68.
[19] Dubova A. A. Poisk dannyh s ispol'zovaniem reguljarnyh vyrazhenij // V sbornike: Mezhdunarodnaja nauchno-tehnicheskaja konferencija molodyh uchenyh BGTU im. V.G. Shuhova. Posvjashhena 165-letiju V.G. Shuhova. – Belgorod, 2018. – S. 3881-3885.
[20] Sheveljova K. V., Avdeev N. N. Primenimost' reguljarnogo vyrazhenija kak matematicheskoj modeli orfograficheskoj oshibki // Aktual'nye problemy prikladnoj matematiki, informatiki i mehaniki: sbornik trudov Mezhdunarodnoj nauchnoj konferencii. – 2019. – S. 335-340.
[21] Gruzdev D. Ju., Makarenko A. S. Ob"ektno-orientirovannyj jazyk programmirovanija i reguljarnye vyrazhenija v praktike pis'mennogo perevodchika // Uspehi gumanitarnyh nauk. 2019. – # 8. – S. 146-153.
[22] Romanjuk B.V. k voprosu o primenenii reguljarnyh vyrazhenij // V sbornike: Problemy informacionnoj bezopasnosti social'no-jekonomicheskih sistem. VIII Vserossijskaja s mezhdunarodnym uchastiem nauchno-prakticheskaja konferencija. – Simferopol', 2022. – S. 70-71.
[23] Kozlov S. V., Svetlakov A. V. O LL(1)-grammatikah, algoritmah na nih i metodah ih analiza v programmirovanii // International Journal of Open Information Technologies. – 2022. T. 10. # 3. – S. 30-38.
[24] Skripov A. V. Opisanie kontekstnyh uslovij formal'nyh jazykov grammatiki s kontekstual'nymi argumentami // Vestnik Ural'skogo instituta jekonomiki, upravlenija i prava. – 2013. # 1 (22). – S. 111-116.
[25] Martynenko B. K. Reguljarnye jazyki i KS-grammatiki // Komp'juternye instrumenty v obrazovanii. – 2012. # 1. – S. 14-20.
[26] Svetlakov A. V., Bahman V. A., Bodju V. Ju. Vizualizacija tehnologicheskogo marshruta s pomoshh'ju graficheskogo ispolnitelja // Potencial innovacionnogo razvitija Rossijskoj Federacii v novyh geopoliticheskih uslovijah: sbornik statej Nacional'noj (Vserossijskoj) nauchno-prakticheskoj konferencii. – Ufa, 2021. – S. 40-45.
[27] Kuljukin K. S. Osobennosti zadachi uproshhenija kvantifikatorov v reguljarnyh vyrazhenijah // V knige: Konkurs nauchno-issledovatel'skih rabot studentov Volgogradskogo gosudarstvennogo tehnicheskogo universiteta. Tezisy dokladov. Redkollegija: S.V. Kuz'min (otv. red.) [i dr.]. – 2020. – S. 171-172.
[28] Andrianov I. A., Grigor'eva A. N. Modernizacija indeksa dlja poiska po reguljarnym vyrazhenijam // Sistemy upravlenija i informacionnye tehnologii. – 2020. – # 2 (80). – S. 60-64.