УДК 004.67
DOI: 10.25140/2411-5363-2020-2(20)-167-174

Юлия Бабич, Николай Бабич, Елена Павлышко, Виктория Наконечная

ИССЛЕДОВАНИЕ ДЕТЕРМИНИРОВАННЫХ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ СТРУКТУРЫ ДАННЫХ XML-ТИПА

Актуальность темы исследования. *В данной статье проведён глубокий анализ большого набора данных с помощью поисковых систем и хостинговых платформ. Использованы четыре стратегии сбора данных: анализ поисковой системы Google, сканирование адресного пути, анализ веб-сайтов, поиск потенциальных данных для получения большего количества файлов-схем из сети Интернет. Получил дальнейшее практическое исследование набор данных для изучения детерминированных регулярных выражений.*

Постановка проблемы. *Современные языки описания структур данных XML-типа требуют применение детерминированных регулярных выражений, позволяющих считывать строки посимвольно. Поэтому исследование данных выражений позволит ускорить процесс обработки данных и получить более точный результат.*

Анализ последних исследований и публикаций. *Проведённый анализ современных литературных источников и публикаций на данную тематику показал, что большинство из них используют небольшие объёмы данных, что является недостаточным для проведения эффективного анализа.*

Выделение неисследованных частей общей проблемы. *Для эффективного анализ данных из сети Интернет был использован большой набор данных и четыре стратегии его сбора и анализа.*

Целью *написания данной статьи является исследование детерминированных регулярных выражений, которые всё чаще применяются в структурах данных XML- типа.*

Изложение основного материала. *Разработка четырёх стратегий сбора данных в сети Интернет дала возможность получить больше XML-схем, что в 35 раз больше, чем в ранее проведённых исследованиях. Применение детерминированных регулярных выражений в целом и их подклассов для анализа больших наборов данных.*

Выводы в соответствии со статьей*. Впервые применены детерминированные регулярные выражения с использованием структуры данных XML- типа. Получен большой объём данных – 276371 файлов с помощью четырёх стратегий их сбора.*

Ключевые слова: *набор данных; регулярные выражения; детерминированные регулярные выражения; XML-тип.*
Рис.: 4. Табл.: 3. Библ.: 10.

Актуальность темы исследования. Анализ больших наборов данных в современном мире требует применения новых стратегий их сбора. Использование регулярных выражений (РВ), которые являются фундаментальными в языках программирования, поисковых системах, утилитах обработки текста, запросах в базах данных и т. д. [1], оказалось недостаточным для больших объёмов данных, поэтому проведено исследование детерминированных регулярных выражений (ДРВ).

Постановка проблемы. В настоящее время язык разметки XML широко использует ДРВ для обмена данными в сети. Структуры данных XML определяются схемами, которые обеспечивают множество удобств и преимуществ для различных процессов, таких как обработка и автоматическая интеграция данных, статический анализ трансформаций и др. Среди популярных языков описания структуры XML-документа выделяют DTD и XSD, которые рекомендованы Консорциумом World Wide Web (W3C) [1]. Одним из основных требований к моделям контента с использованием данных языков разметки является применение ДРВ, которые могут считывать входную строку посимвольно слева направо, не возвращаясь назад. Одним из непосредственных преимуществ использования таких выражений является эффективный синтаксический анализ.

Анализ последних исследований и публикаций. Проводимые ранее исследования использовали порядка 100 XSD/DTD схем [2], что является недостаточным для проведения анализа больших наборов данных. В нашем исследовании все данные получены из репозитория РВ RegExLib, языка Relax NG, XSD и DTD [3] – это основной репозиторий РВ, доступный в Интернете, который содержит несколько видов выражений для соответствия URI, коду разметки, фрагментам кода Java, SQL-запросам, спаму и т. д. Другое направление исследований в данной сфере применяет популярную схему языка XML – это язык Regular Language для XML следующего поколения. Спецификация W3C требует, чтобы содержимое модели DTD и XSD было ДРВ, в то время как в языке Relax NG и RegExLib нет ограничений детерминизма. Поскольку и XSD, и RegExLib поддерживают такой вид оператора, как «счётчик», они и будут использованы в качестве примеров для изучения практического применения ДРВ в дальнейших исследованиях.

Выделение неисследованных частей общей проблемы. Для сбора данных были использованы четыре стратегии:

- анализ поисковой системы Google. Для поиска URL-адресов с помощью поисковой системы необходимы текстовые файлы XML-схем. Использовались API поисковой системы Google для получения ресурсов, файловых инструкций и сайтов с URL-адресами DTD, XSD и Relax NG. Во время эксперимента сохранялись URL-адреса XSD, DTD, Relax NG в соответствующих документах и одновременно удалялись повторяющиеся URL-адреса. При загрузке файла схемы была создана иерархия папок в соответствии с URL-каталогом, и каждая из папок сохранена локально. Преимущество данной стратегии заключается в том, что, когда файл схемы XML ошибочен или неоднозначен, то можно отследить его URL в сети Интернет в соответствии с сохраненным каталогом и проверить его. Этот принцип также используется в следующих трех стратегиях;

- сканирование адресного пути. Например, задан начальный URL-адрес: http://52north.org/schema/users/1.0/users.xsd, будет выполнен обход 52north.org, 52north.org/schema, 52north.org/schema/users и 52north.org/schema/users/1.0. Данная стратегия определения пути очень эффективна в поиске изолированных ресурсов или источников, для которых невозможно найти ссылку при регулярном обходе;

- анализ веб-сайтов. В качестве примера рассмотрим источник http://repo1.maven.org (Maven2), в котором некоторые URL-адреса скрыты в JAR/ZIP-файлах и поэтому не могут быть найдены через поисковые системы. Данные файлы загружаются как архивы JAR и ZIP, а затем извлекаются локально. Далее производится фильтрация файлов других типов, при этом сохраняются только XSD, DTD и Relax NG;

- поиск потенциальных данных для получения большего количества файлов-схем из сети Интернет. У каждого сайта могут быть разные файлы схем, например, Relax NG имеет и XSD, и DTD схемы, поэтому был проведён перекрёстный поиск. Этот метод доказал свою эффективность в нашем исследовании, поскольку предоставляет доступ к большому количеству URL-адресов и файлам схем.

Целью написания данной статьи является исследование ДРВ, которые всё чаще применяются в больших наборах данных и структурах данных XML-типа.

Изложение основного материала. В последние годы Консорциум всемирной паутины W3C требует, чтобы содержимое языков схем было в виде ДРВ. Проведя эксперимент с использование четырёх стратегий, было получено 276371 файлов данных, из которых 124326 – DTD, 134816 –XSD, 13946 – Relax NG и 3950 – выражения RegExLib, что в 35 раз больше, чем в ранее проведенных исследованиях. Такой объём данных для изучения ДРВ имеет большое практическое значение, поскольку отображает реальную картину использования крупномасштабных случайных схем. В результате исследования, обнаружено, что более 98 % РВ в Relax NG и более 56 % в RegExLib являются детерминированными. Эти цифры указывают на то, что именно ДРВ имеют практическое применение.

Поскольку на практике большинство моделей контента, используемых в DTD и XSD, состоят из ограниченных подклассов ДРВ, поэтому исследования сосредоточены на изучении именно этих подклассов и их применении [3].

Чтобы определить, является ли стандартное РВ детерминированным, был применён временной алгоритм $O(\left|\sum E\right|\left|E\right|)$, где $\left|\sum E\right|$ – множество различных символов в E [1,3]. Для РВ со счётчиком временной алгоритм – это выражение $O(\left|\sum E\right|\left|E\right|)$. Chen и Lu [2] исследовали алгоритмы, проверяли детерминизм входного стандарта РВ и выполняли проверку, если выражение не являлось детерминированным, то использовали временной алгоритм $O(\left|\sum E\right|\left|E\right|)$ со счётчиком [3]. Учёный Peng и др. [4] предложили временной алгоритм $O(\left|\sum E\right|\left|E\right|)$ для проверки ДРВ с чередованием. Groz и Maneth [4] впервые предложили $O(\left|E\right|)$ временной алгоритм для проверки стандартных ДРВ и ДРВ со счётчиком.

Пусть $\sum$ алфавит символов (элементов). Множество всех конечных слов в $\sum$ обозначим через $\sum^*$. Пустое слово обозначается через ε. Стандарт РВ в $\sum$ определяется как: ϕ, ε или $\alpha \in \sum$ – РВ, объединение $E_1|E_2$ конкатенация E_1E_2 или E_1^* является РВ для РВ E_1 и E_2. N обозначим множество $\{0,1,2,...\}$. Через РВ со счётчиком и чередованием отличается от стандартных РВ использованием численного оператора итерации $E^{[m,n]}$ и оператора чередования $E_1 \& E_2$. Оценочные m и n удовлетворяют следующим условиям: $m \in N$, $n \in N \setminus \{0\} \cup \{\infty\}$ и $m \le n$. Для обозначения множества строк, полученных из s_1 и s_2 всеми возможными способами, используем $s_1 \& s_2$. Для $s_1, s_2 \in \sum^*$ и $a, b \in \sum$, $s_1 \& \varepsilon = \varepsilon \& s_1 = \{s\}$ *and* $as_1 \& bs_2 = \{a(s_1 \& bs_2)\} \cup \{b(as_1 \& s_2)\}$.

В РВ каждый символ индексируется, для того, чтобы он встречался только один раз. Например, $(a_1 + b_1)^* a_2$ является маркировкой выражения $(a+b)^* a$. Маркировка E обозначается $\overline{E}$. Эти же обозначения будут использоваться для удаления индексов указанных символов: $\overline{E} = E$. Из контекста будет ясно, добавляются или уменьшаются индексы [5]. Выражение E является детерминированным, тогда и только тогда, когда для всех слов $uxv, uyw \in L(\overline{E})$, где $|x| = |y| = 1|$, если $x \ne y$, то $\overline{x} \ne \overline{y}$.

Детерминизм требует, чтобы позиция сопоставления была уникальной при сопоставлении предложений с РВ. Например, $a(a)^*$ является детерминированным, тогда как $(a)^*a$ – нет, хотя указанные символы эквивалентны. Для $a_2 \in L((a)^* a)$ и $a_1 a_2 \in L((a)^* a)$ предположим $u = \varepsilon$, $x = a$, $y = a$, $v = \varepsilon$, $w = a$, следовательно, получаем $x = y = a$, но $\overline{x}(a_2) \ne \overline{y}(a_1)$, поэтому $(a)^* a$ не является ДРВ.

Далее следуют некоторые подклассы ДРВ, которые часто используются на практике:

- SORE (однократное РВ). Пусть $\sum$ – алфавит. SORE является стандартным РВ над $\sum$, в котором каждый конечный символ встречается не более одного раза. Например, $(a^* b^{[0,2]})^+$ является однократным РВ, а выражение $(a^* b^{[0,2]} a^*)^+$ – нет, хотя указанные символы эквивалентны.

- Simplified CHARE является однократным РВ над $\sum$ формы $f_1 ... f_n$, где $n \ge 1$. Фактор f_i является выражением вида $(a_1 + ... + a_m)$, $(a_1 + ... + a_m)^?$, $(a_1 + ... + a_m)^*$, $(a_1 + ... + a_m)^+$, где $m \ge 1$ и $a_i \in \sum$.

- eSimplified CHARE является SORE над $\sum$ вида $f_1 ... f_n$, где $n \ge 1$. Фактор f_i – это выражение $(b_1 + ... + b_m)$, $(b_1 + ... + b_m)^?$, $(b_1 + ... + b_m)^*$, $(b_1 + ... + b_m)^+$, где $m \ge 1$ и b_i является a_i или a_i^+, где $a_i \in \sum$.

Модель контента XSD поддерживает ограниченную форму чередования, в то время как Relax NG – неограниченную. Для анализа детерминизма Relax NG нужны инструменты, которые могут определять детерминизм неограниченного чередования. В конкретном случае изучен детерминизм РВ, генерируемый схемами REs из RegExLib, результаты представлены на рис. 1.

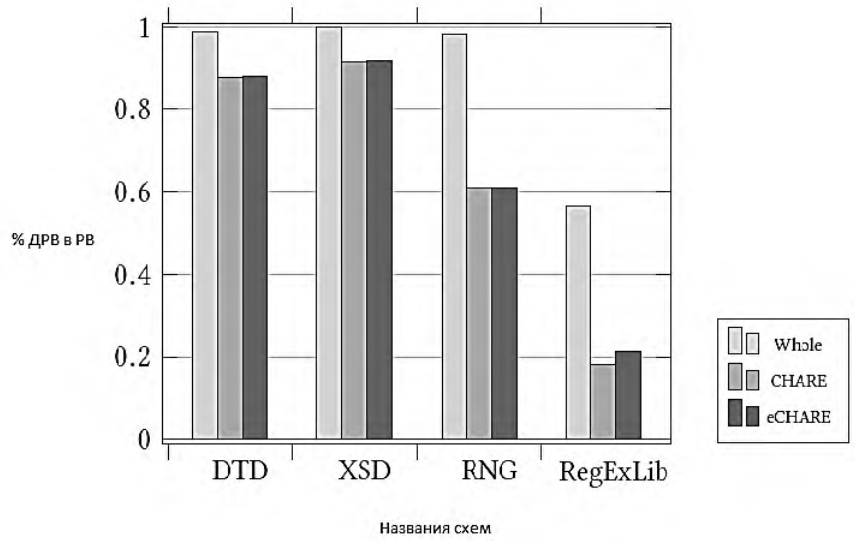

Рис. 1. Процент детерминированных регулярных выражений

Таким образом, исходя из полученных результатов, можно сказать, что ДРВ очень часто используются на практике.

На практике многие DTD и XSD содержат подклассы ДРВ, поэтому проведено их исследование. Существующие подклассы ДРВ определены на стандартных РВ, то есть SORE [4], Simplified CHARE [3], eSimplified CHARE [4]. Последние два являются подклассами SORE. Также включены три новых подкласса: SOREwCorI, SOREwC, SOREwI.

Исследования показывают, что используемые подклассы ДРВ имеют следующие слабые стороны:

- отсутствуют эксперименты по крупномасштабным реальным данным;
- нет подкласса ДРВ, который принадлежит неSORE, хотя в экспериментах с SORE данные подклассы занимают высокий процент;
- существующие подклассы ДРВ, определенны на стандартных РВ, без учёта и/или чередования.

Из этого следует, что текущие исследования по подклассам ДРВ всё ещё находятся на начальной стадии и необходимо дальнейшие их изучение.

Для этого используем алфавит $\sum$ и высоту звезды РВ над ним [4] – $h(E)$, она представляет собой неотрицательное целое число, определенное рекурсивно следующим образом:

1. $h(E)=0, if\ E\neq\varnothing\ or\ a\ for\ a\in\sum$.

2. $h(E)=\max\{h(E_1),h(E_2)\}, if\ E=(E_1+E_2)\ or\ E=(E_1+E_2) or\ E=(E_1E_2)\ where\ E_1,E_2$ является РВ в $\sum$.

3. $h(E)=h(E_1)+1, if\ E=(E_1)^*\ and\ E_1$ является РВ в $\sum$.

Результаты представлены в таблице 1.

Таблица 1

Высота звезды в DTD, XSD и RING

Высота звезды	DTDs (%)	XSDs (%)	RNGs (%)
0	26,56	65,27	66,42
1	71,72	34,11	32,86
2	1,69	0,57	0,72
3	0,03	0,05	0

Глубина вложения РВ в алфавит $\sum$, обозначена $ND(E)$ и представляет собой неотрицательное целое число, рекурсивно определяемое следующим образом:

- $ND(E) = 0$ *if* $E \neq \varnothing$ *or a for* $a \in \sum$;
- $ND(E) = \max\{ND(E_1), ND(E_2)\}$, *if* $E = (E_1 + E_2)$, $E = (E_1 \& E_2)$ *or* $E = (E_1 E_2)$, *where* E_1, E_2 *are PB are* $\sum$
- $ND(E) = ND(E_1) + 1$, *if* $E = (E_1)^*$, $E = (E_1)^?$ *or* $E = (E_1)^{[m,n]}$ *for* E_1 *is a PB over* $\sum$.

Результаты представлены в таблице 2.

Таблица 2

Глубина вложения в DTD, XSD и RNG

Глубина вложения	DTDs (%)	XSDs (%)	RNGs (%)
0	94,58	99,24	91,31
1	4,60	0,73	8,45
2	0,58	0,02	0,09
3	0,24	0,01	0,15

Определим плотность схемы, как число элементов, встречающихся в правой части его правил, разделенных на количество элементов.

$$d = \frac{1}{N}\sum_{i=1}^{N}|A_i|,$$

где N – общее количество определений элементов, встречающихся в схеме, A_i – строка в правой части правила, а $|A_i|$ обозначает размер A_i. Файлы XML-схемы с большим значением плотности имеют более высокую сложность. Эксперимент показал, что средняя плотность Relax NG, XSD и DTD составляет 1,8689, 1,3476 и 10002 соответственно. На рис. 2, показана плотность трех видов XML-схем.

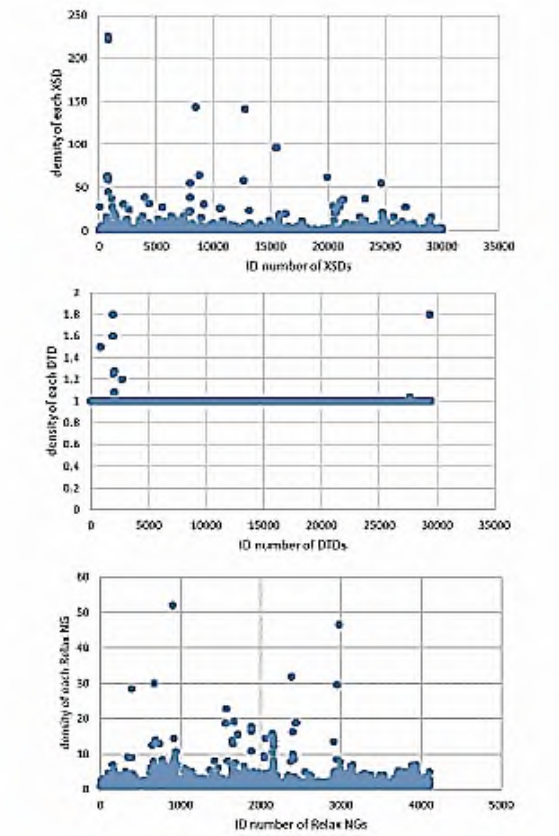

Рис. 2. Плотность схем XSD, DTD и Relax NG

Эксперимент показал, что XSD является наиболее часто используемой схемой для определения XML (рис. 3).

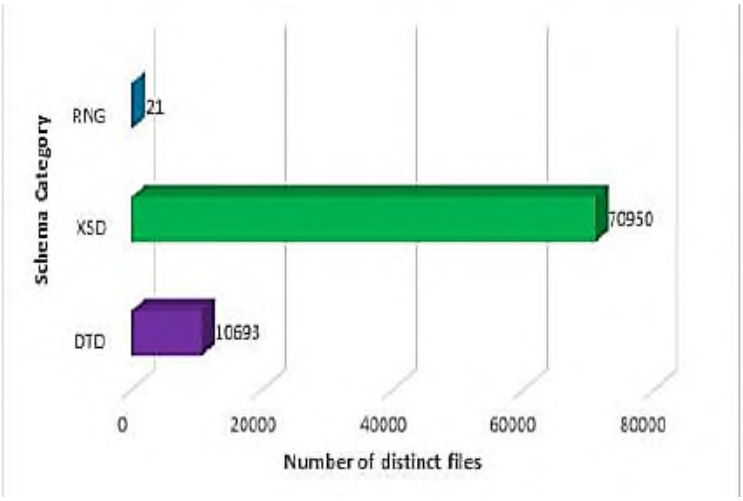

Рис. 3. Частота использования схем при извлечении файлов из сети Интернет

После построения схемы сети было выбрано 42 файла, которые имеют явно более высокие значения SchemaRank, чем другие схемы, и сосредоточились на 39 880 РВ, проанализированных из этих файлов схемы. Все эти выражения являются ДРВ, что подтверждает их частоту применения. Проанализировано, к каким подклассам принадлежат ДРВ, изучено три основных – SORE, Simplified CHARE и eSimplified CHARE, представленных на рис. 4.

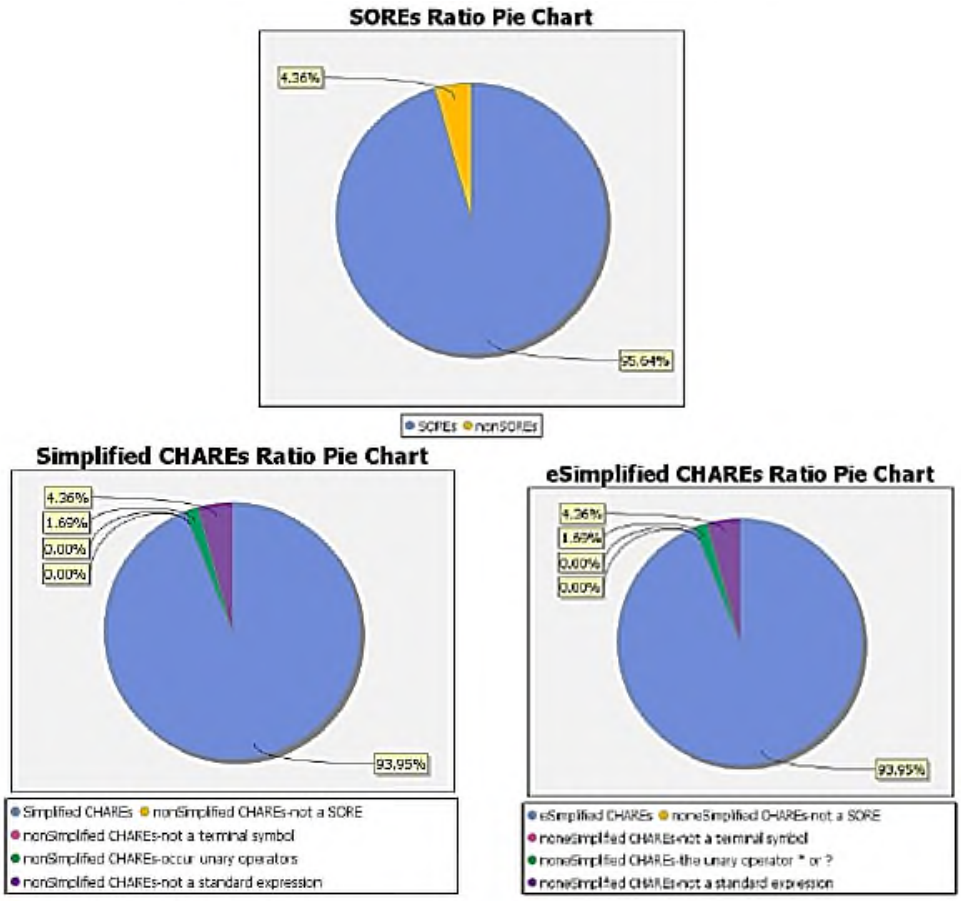

Рис. 4. Процент подклассов ДРВ на подмножестве XSD

Результаты показывают, что для применения на практике необходимо определить новый подкласс.

Иногда встречаются случаи, когда XML-схема была неправильно сформирована. Согласно [6], только 24,8 % XML в Интернете содержат ссылку на DTD или XSD, из которых только одна треть является действительной. SchemaRank в таких случаях помогает, если используется элемент или атрибут, который не определен в схеме. Для этого необходимо нормализировать ДРВ в наборе данных [7,8], заменив символы в порядке слева направо символами $a_1, a_2, a_3 ... a_n$. В результате выражения с одинаковыми или похожими структурами могут быть объединены для получения более компактного набора данных [9, 10], как показано в таблице 3. Этот нормализованный набор данных ДРВ будет полезен в приложениях и сам по себе имеет ценность.

Таблица 3

Количество ДРВ

Тип	Исходный набор ДРВ	Нормализированный набор ДРВ
DTD	87,176	3,767
XSD	266,100	14,771
RNG	353,926	2,791
RegExLib	2,234	724
Всего	709,436	20,339

Выводы в соответствии со статьей. В статье впервые предложено четыре стратегии сбора данных в сети Интернет, что дало возможность получить больше XML-схем для более точных результатов.

Проведено исследование применения ДРВ, основываясь на большом наборе реальных данных. Результаты экспериментов показали, что ДРВ намного чаще применяются на практике, чем РВ, и дальнейшее исследование их подклассов обеспечит более точный результат применения схем данных в сети Интернет.

Список использованных источников

1. Yeting Li, Xiaolan Zhang, Feifei Peng, Haiming Chen. Practical Study of Subclasses of Regular Expressions in DTD and XML Schema. *Springer International Publishing*, Cham, 2016.

2. Regex Advice. RegExLib. URL: http://www.regexlib.com.

3. Фридл Дж. Регулярные выражения : учебное пособие. 3-е изд. Санкт-Петербург : Символ-Плюс, 2008. 608 с.

4. The regular expressions in practice. URL: https://www.regular.com.

5. Косенко Ю. І., Рослякова С. В., Носов П. С. Система ідентифікації функціональної ентропії суб'єкта критичної інфраструктури. *Современные направления теоретических и прикладных исследований* : сборник научных трудов по материалам Международной научно-практической конференции. Одесса, 2013. Вип. 2. С. 50–54.

6. Pogorilyy S., Shkulipa I. A. Conception for Creating a System of Parametric Design of Parallel Algorithms and their Software Implementations. *Cybernetics and System Analysis*. 2009. № 6. P. 952–958.

7. Grijzenhout S., Marx M. The quality of the XML web. *Web Semantics: Science, Services and Agents on the World Wide Web*. 2013. № 19. P. 59–68.

8. Системні дослідження та інформаційні технології. URL: http://journal.iasa.kpi.ua.

9. Jsoup: Java HTML Parser. URL: https://jsoup.org/apidocs/overview-summary.html.

10. Morton M. The process of using regular expressions. *Cybernetics and System Analysis*. 2017. № 2. P. 42–49.

References

1. Yeting, Li, Xiaolan, Zhang, Feifei, Peng, & Haiming, Chen. (2016). *Practical Study of Subclasses of Regular Expressions in DTD and XML Schema*. Springer International Publishing, Cham.

2. Regex Advice. (2001). RegExLib. Retrieved from http://www.regexlib.com.

3. Friedl, J. (2008). *Reguliarnie vyrageniia [Regular Expressions]*. (3rd ed.). St. Petersburg: SymbolPlus [in Russian].

4. Regex Advice. (2001). *The regular expressions in practice*. Retrieved from http://www.regular.com.

5. Kosenko, Yu. I. (2013). Systema identifikatsii funktsionalnoi entropii subekta kriticheskoi infrastruktury [System of identification of functional entropy of the subject of critical infrastructure]. *Sovremennye napravleniia teoreticheskikh i prikladnykh issledovanii: sbornik nauchnykh trudov po materialam Mezhdunarodnoi nauchno-prakticheskoi konferentsii – Modern areas of theoretical and applied*

research: Collection of scientific papers on the materials of the international scientific-practical conference. Odessa [in Ukrainian].

6. Pogorilyi, S., Shkulipa, I. (2009). A Conception for Creating a System of Parametric Design of Parallel Algorithms and their Software Implementations. *Cybernetics and System Analysis,* 54, 952–958.

7. Grijzenhout Steven, Marx Maarten. (2013). The quality of the XML web. *Web Semantics: Science, Services and Agents on the World Wide Web,* 19, 59–68.

8. System research and information technology. Retrieved from http://journal.iasa.kpi.ua.

9. Jsoup: Java HTML Parser. Retrieved from https://jsoup.org/apidocs/overview-summary.html.

10. Morton, M. (2018). *The process of using regular expressions.* New York: NYED Isis 18, 61–68.

UDC 004.67

Yuliia Babych, Mykola Babych, Olena Pavlyshko, Victoriia Nakonechna

RESEARCHING OF DETERMINED REGULAR EXPRESSIONS USING THE XML TYPE DATA STRUCTURE

Urgency of the research*. This article provides an in-depth analysis of a large data set using search engines and hosting platforms. Four data collection strategies analysis of the Google search engine, scanning of the address path, analysis of websites, searching for potential data were used to obtain more diagram files from the Internet. A data set for studying deterministic regular expressions received further practical research.*

Target setting*. Modern XML-type data structure description languages require deterministic regular expressions to read lines character by character. Therefore, the study of these expressions will speed up the data processing and get a more accurate result.*

Actual scientific researches and issues analysis*. The analysis of modern literary sources and publications on this topic showed that most of them use small amounts of data, which is insufficient to conduct an effective analysis.*

Uninvestigated parts of general matters defining. *For effective analysis of data from the Internet, a large data set and four strategies for its collection and analysis were used.*

The research objective *of this article is to study deterministic regular expressions, which are increasingly used in XML-type data structures.*

The statement of basic materials*. The development of four strategies for collecting data on the Internet made it possible to get more XML-schemes, which is 35 times more than in previous studies. The use of deterministic regular expressions in general and their subclasses for the analysis of large data sets.*

Conclusions*. For the first time, deterministic regular expressions are applied using an XML-type data structure. A large amount of data was obtained - 276371 files using four strategies for their collection.*

Keywords*: data set, regular expressions, deterministic regular expressions, XML type.*

Fig.: 4. Table: 3. References: 10.

Бабич Юлия Игоревна – кандидат технических наук, доцент кафедры информационных технологий проектирования в машиностроении, Одесский национальный политехнический университет (просп. Шевченко, 1, г. Одесса, 65000, Украина).
Yuliia Babych – PhD in Technical Science, Associate Professor of Information Technology, Design in Mechanical Engineering department, Odessa National Polytechnic University (1 Shevchenka Str., 65000 Odessa, Ukraine).
E-mail: juliakosenko1987@gmail.com
ORCID: https://orcid.org/0000-0001-9966-2810

Бабич Николай Иванович – кандидат технических наук, доцент кафедры информационных систем, Одесский национальный политехнический университет (пр. Шевченко, 1, г. Одесса, 65000, Украина).
Mykola Babych – PhD in Technical Science, Associate Professor, Department of Information Systems, Odessa National Polytechnic University (1 Shevchenka Str., 65000 Odessa, Ukraine).
E-mail: babich.tiger@gmail.com
ORCID: http://orcid.org/0000-0002-3946-9880

Павлышко Елена Георгиевна – старший преподаватель кафедры информационных технологий проектирования в машиностроении, Одесский национальный политехнический университет (пр. Шевченко, 1, г. Одесса, 65000, Украина).
Olena Pavlyshko – Senior lecturer at Department of Information Technology, Design in Mechanical Engineering, Odessa National Polytechnic University (1 Shevchenka Str., 65000 Odessa, Ukraine).
E-mail: pavlyshko.o.g@opu.ua.

Наконечная Виктория Ивановна – преподаватель кафедры экономики, управления и администрирования, Херсонский политехнический колледж Одесского национального политехнического университета (ул. Небесной сотни, 23, г. Херсон, 73013, Украина).
Victoriia Nakonechna – Instructor at Department of Management, Economics and Administration Kherson Polytechnic College of Odessa National Polytechnic University (23 Nebesna Sotnya Str., 73013 Kherson, Ukraine).
E-mail: vgrabar2009@meta.ua

Бабич Ю., Бабич Н., Павлышко Е., Наконечная В. Исследование детерминированных регулярных выражений с использованием структуры данных XML-типа. *Технічні науки та технології.* 2020. № 2 (20). С. 167-174.