О структуре неоднозначностей в формальных языках

Выполнила: Беликова Ю.А., МГТУ им. Н.Э. Баумана

Научный руководитель: Непейвода А.Н, МГТУ им. Н.Э. Баумана, ИПС им. А.К. Айламазяна РАН

Совместное Совещание по языку Рефал
МГТУ им. Н.Э. Баумана (кафедра ИУ9) и
ИПС им. А.К. Айламазяна РАН

Постановка задачи

- ✓ **На сегодняшний день не существует полной теории для оценки неоднозначности в рекурсивных образцах и расширенных регулярных выражениях.**

- ✓ **Нет эффективных инструментов для поиска описанных неоднозначностей.**

План

1. Неоднозначность в классических регулярных выражениях.
2. Неоднозначность в расширенных регулярных выражениях.
3. Неоднозначность в образцах.
4. Обзор современных алгоритмов поиска неоднозначностей.
5. Обзор предлагаемого подхода.
6. Результаты.

Классические регулярные выражения

Классические регулярные выражения – это множество выражений над некоторым алфавитом, замкнутое относительно операций итерации Клини, конкатенации и объединения.

Лемма о накачке

Лемма 1. *Если язык L является регулярным, то существует число $n \geq 1$ такое что для любого слова ω из языка L, где $|\omega| \geq n$ существует разбиение $\omega = xyz$, $y \neq \epsilon$, $|xy| \leq n$ и $\forall k \geq 0\ xy^k z \in L$.*

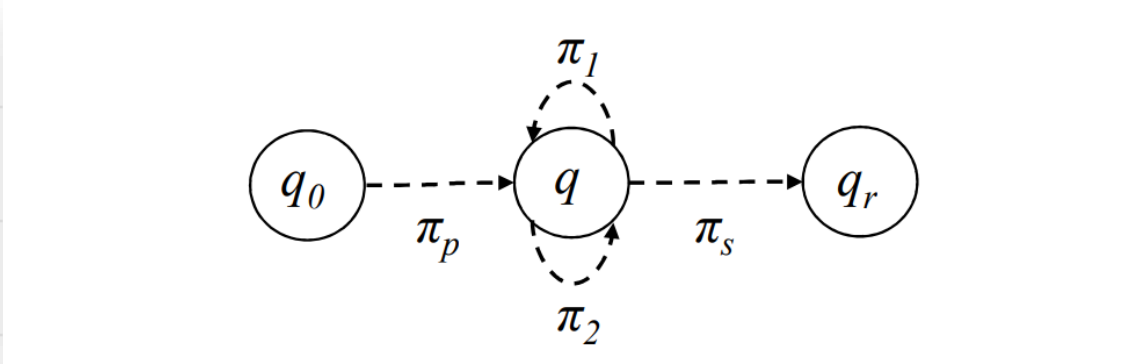
Неоднозначность в классических регулярных выражениях

Подход, предложенный авторами инструмена RegexScalpel, - анализ подвыражений следующих типов:

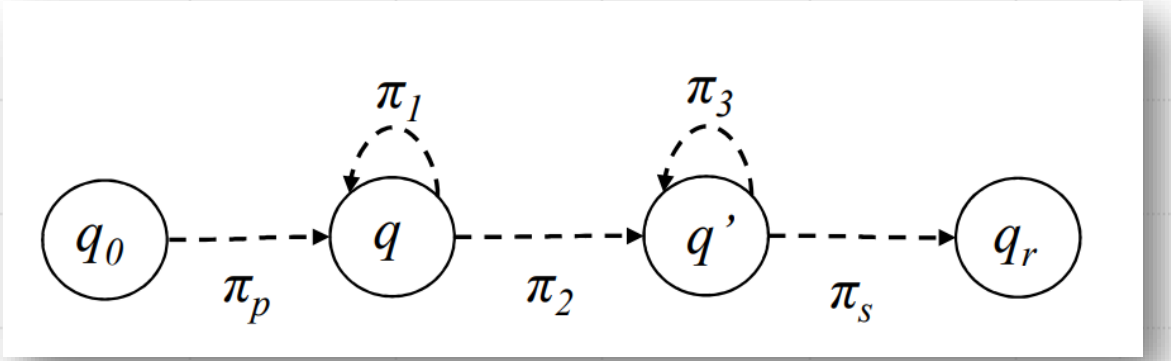
1. Вложенные квантификаторы (NQ)
2. Перекрывающаяся альтернатива под квантификатором (QOD)
3. Перекрывающаяся конкатенация под квантификатором (QOA)

Неоднозначности в моделях НКА

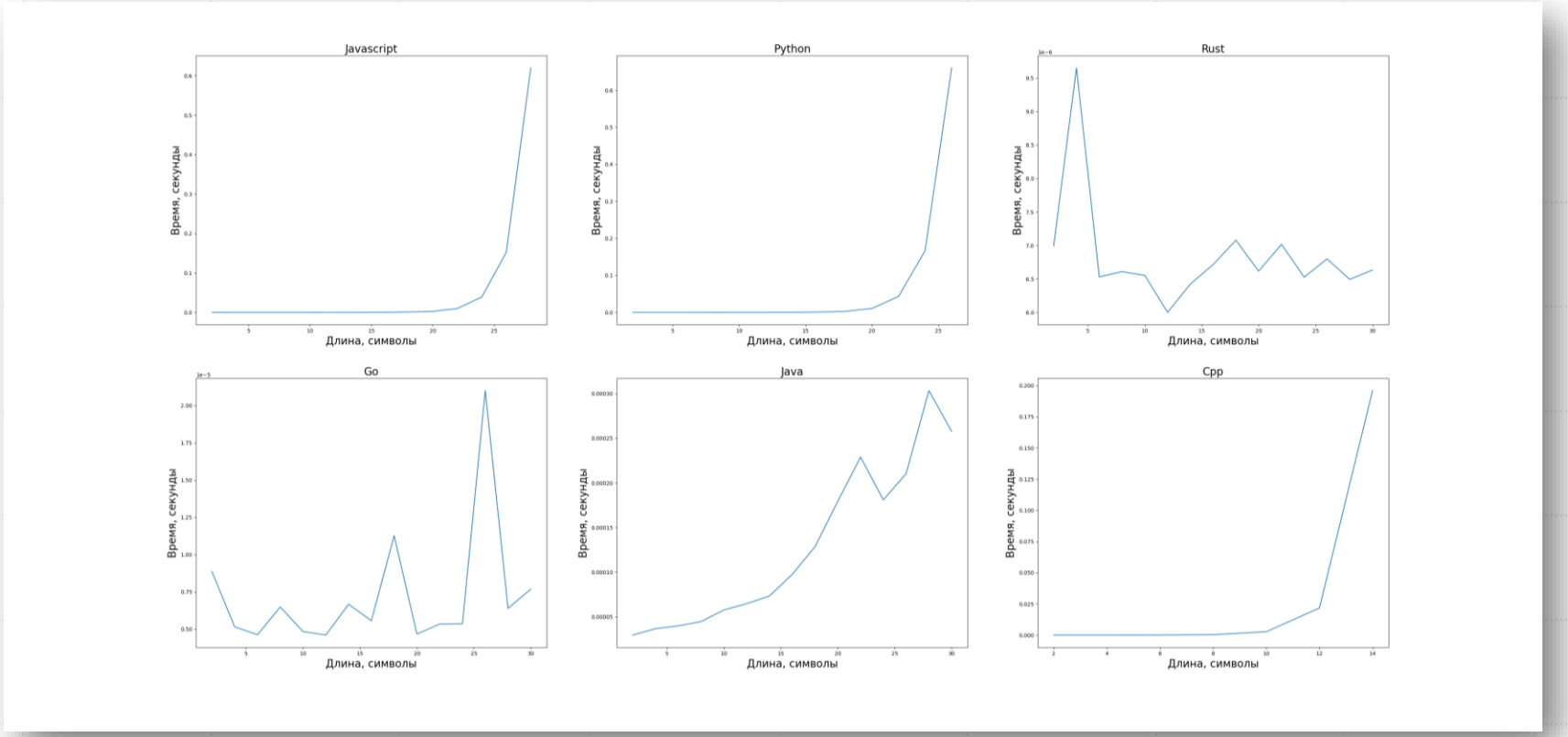
1. Экспоненциальная неоднозначность



2. Полиномиальная неоднозначность



Катастрофический возврат



Регулярное выражение: **(\d+|(\d*,\d+))+**

Атакующая строка: **0...0,**

Неоднозначность в расширенных регулярных выражениях

- ✓ Расширенные регулярные выражения допускают нетривиальную структуру неоднозначности, не исчерпывающуюся квазиполиномиальными случаями.
- ✓ **Пример.** Число разборов строки длины n по регулярному выражению (a*)\1* в точности равно числу делителей n.

Лемма о накачке нециклических расширенных регулярных выражений

Лемма 2. *Пусть* α *– расширенное регулярное выражение. Тогда существует константа* $N > 0$*, что если* $\omega \in L(\alpha)$ *и* $|\omega| > N$*, тогда существует такое разбиение* $\omega = x_0 y x_1 y \ldots y x_m$*, что для некоторого* $m \geq 1$ *выполняется:*

1. $|x_0 y| < N$,
2. $|y| \geq 1$,
3. $x_0 y^j x_1 y^j \ldots y^j x_m \in L(\alpha)$ *для* $\forall j > 0$.

Представление расширенных регулярных выражений как образцов

Пример Саломаа-Матееску. Регулярное выражение ((a|b|c)*)ab\2bca((a|b|c)*)abc\3 экспоненциально неоднозначно.

Действительно, образец XabXbcaYabcY неоднозначен, так как существует подстановки f и g: f(X) = ca, f(Y) = abcbc, g(X) = caabc, g(Y) = bc.

Глобальная неоднозначность в образцах

Определение 2. *Образец* $P(x_1, \ldots, x_n)$ *называется (глобально) неоднозначным, если существует хотя бы одно слово* w *и различные подстановки* σ_1, σ_2 *такие, что* $w = \sigma_1(P) = \sigma_2(P)$.

Локальная неоднозначность в образцах

Определение 3. *Скажем, что образец* P *локально бесконечно неоднозначен, если существует его разбиение* P_1P_2 *такое, что для любого* $k \in$ *существуют* $w, u_1, \ldots, u_k, v_1, \ldots, v_k (\forall i,j (w = u_i v_i,\ u_i \neq u_j$ *и* $u_i(P_1), v_i \in L_{Pref}(P_2)))$.

Признак неоднозначности Матееску

Лемма 3. *Если образец P содержит минимум две переменные, и при этом хотя бы одна переменная имеет единственное вхождение в P, то P бесконечно неоднозначен.*

Современные фазз-алгоритмы. ReScue

Основная идея.

Представление регулярного выражения как расширенного НКА (англ. e-NFA).

Стадии алгоритма:

1. Инициализация (англ. seeding)
2. Инкубация (англ. incubating)
3. Накачка (англ. pumping)

Input: $s, l(|s| \leq l)$
$m \leftarrow 0$
$i^* \leftarrow 0, j^* \leftarrow 0$
for each $i \in 1..|s|$ **do**
  for each $j \in i+1..|s|$ **do**
    $s' \leftarrow s(1:i-1) \cdot s(i:j)^2 \cdot s(j+1:|s|)$
    if $f_{incub}(s') > m$ **then**
      $m \leftarrow f_{incub}(s')$
      $i^* \leftarrow i, j^* \leftarrow j$
$k \leftarrow \left\lfloor \frac{l-|s|}{j^*-i^*+1} \right\rfloor$
return $s(1:i^*-1) \cdot s(i^*:j^*)^k \cdot s(j^*+1:|s|)$

Современные фазз-алгоритмы. Regulator

Основная идея.

Использование промежуточного представления регулярного выражения в виде байт-кода.

Стадии алгоритма:

1. Инкубация (англ. incubating)
2. Накачка (англ. pumping)

Input: $corpus\ C : \mathcal{P}(\Sigma^* \times \Pi)$
$T \leftarrow []$
for each $e \in BranchingEdges$ **do**
append $(maxRepresentative(e) \in C)$ to T
for each $(w, \pi) \in C$ **do**
if $w \notin T \vee Staleness[w] < RAND()$ **then**
append w to T
$R \leftarrow []$
for each $w \in T$ **do**
for each $i \in 0...NumChildren$ **do**
$w' \leftarrow Mutate(w)$
append (w, w') to R
return R

Структура неоднозначности в расширенных регулярных выражениях

Уравнение сопряжения

$$w_1 u = u w_2$$

- ✓ $w_1 = ts$
- ✓ $w_2 = st$
- ✓ $u = (ts)^n t$

Окрестность регулярного подвыражения

Определение 4. *Назовём левой n-k-окрестностью подвыражения r' (обозначаемой $Aff_{n,k}(r')$), входящего в выражение $r_1 r' r_2$, множество строк длины k, таких что:*

- *если $n = 0$, тогда эти строки входят в $L(r')$;*
- *если $n > 0$, тогда они входят в $L(r_1 r' r_2)$, причём их префикс длины n входит в язык суффиксов r_1, а суффикс длины $k - n$ входит в язык префиксов $r' r_2$.*

Предположение о пересечении окрестностей

Предложение 1. *Пусть выражение* $\mathtt{X}\Phi_1\mathtt{X}'\Phi_2$ *имеет бесконечную локальную неоднозначность в префиксе* $\mathtt{X}\Phi_1\mathtt{X}'$*, притом что выражения для* $\mathtt{X}$*,* $\mathtt{X}'$ *и* Φ_1 *однозначны. Тогда:*

- $\forall N, i(\mathit{Aff}_N(\mathtt{X}) \cap \mathit{Aff}_{i,N}(\Phi_1) \neq \varnothing)$;
- $\forall N(\mathit{Aff}_N(\mathtt{X}) \cap \mathit{Aff}_N(\mathtt{X}') \neq \varnothing)$.

Доказательство.

- Пусть существует такое m, что $\mathit{Aff}_m(\mathtt{X}) \cap \mathit{Aff}_{i,m}(\Phi_1) = \varnothing$. Тогда прочитав m символов строки, можно точно сказать, принадлежат ли они языку $\mathtt{X}$, или переходу через Φ_1.
- Аналогично, существование такого m, при котором m-окрестности $\mathtt{X}$ и $\mathtt{X}'$ не пересекаются, означает, что как минимум за m прочитанных символов можно точно сказать, к какому из языков относится фрагмент входной строки.

□

Пример.

Выражение ((ab)*|(ba)*)a\1 однозначно.

Действительно,

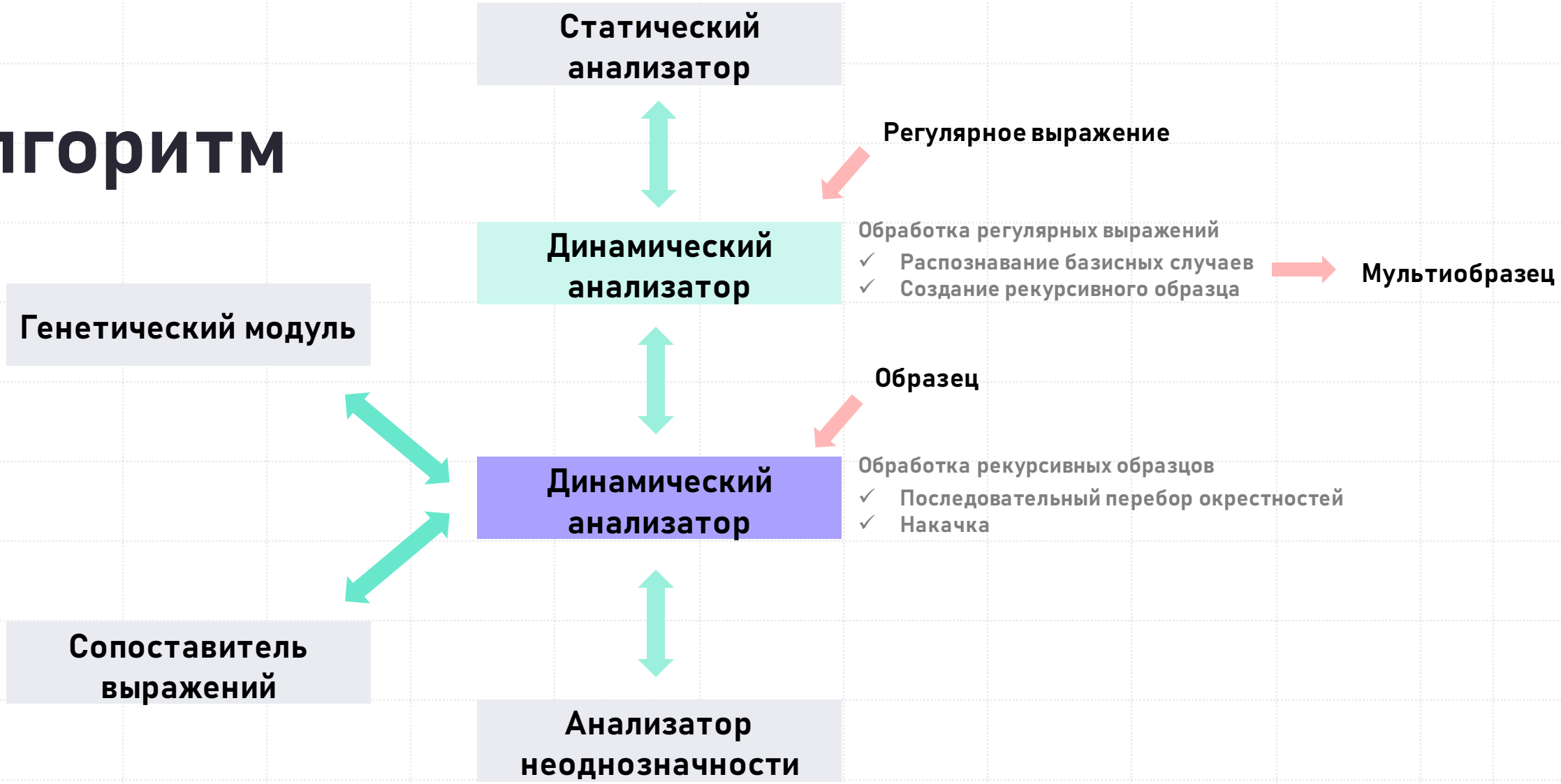
$\mathrm{Aff}_3((ab)^*|(ba)^*) = \{aba, bab\}$

$\mathrm{Aff}_{1,3}(a) = \{aab, baa\}$

Базисные случаи

Алгоритм	(...*...)	(...)*	\i*
–	–	–	–
Подстановка	–	–	+
Подстановка	–	+	–
Разрезание	+	–	–
Подстановка + раскрытие Клини	–	+	+
Накачка осей и диагонали	+	–	+
Разрезание + раскрытие Клини	+	+	–
Накачка осей и диагонали	+	+	+

Алгоритм


Статический анализатор
Регулярное выражение
Динамический анализатор
Обработка регулярных выражений
✓ Распознавание базисных случаев
✓ Создание рекурсивного образца
Мультиобразец
Генетический модуль
Образец
Динамический анализатор
Обработка рекурсивных образцов
✓ Последовательный перебор окрестностей
✓ Накачка
Сопоставитель выражений
Анализатор неоднозначности

Стратегии подбора значений

1. Генетический поиск
2. Перебор на основе перекрытий
3. Сохранение подстановок

Пример работы

```
Example: ((b|a)*)bd(b*)\1bbb\3
Found: polynomial
Pumping pattern:
[Y1]bd[Y2]bbb[Y0]m, [Y0] = (b)*, [Y1] = (b)*, [Y2] = (bbb)*
[Y0]bd[Y2]j, [Y0] = (b)*, [Y2] = (bbb)*
```

Результаты

1. **Предложен подход к определению неоднозначностей на основе перекрытий.**
2. **Реализован динамический анализатор, комбинирующий статический анализ неоднозначностей и динамический анализ по перекрытию.**
3. **Тестирование доказало эффективность предложенного метода для поиска описанных неоднозначностей.**

Выводы

- ✓ **Предложенный подход успешно комбинирует имеющуюся теоретическую базу для языков образцов и классических регулярных выражений.**
- ✓ **По сравнению с наивными современными инструментами анализа, предложенный алгоритм опирается на нетривиальную структуру расширенного выражения, а также рекурсивного образца, что обеспечивает значительное преобладание в эффективности в рамках заданного домена.**
- ✓ **Необходимо расширение функционала и оптимизация.**

Список литературы

1. Clarle Benjamin, Narendran Paliath. On Extended Regular Expressions // Language and Automata Theory and Applications / ed. by Dediu Adrian Horia, Ionescu Armand Mihai, Martín-Vide Carlos. — Berlin, Heidelberg : Springer Berlin Heidelberg. — 2009. — P. 279–289.

2. Multi-pattern languages / Kari Lila, Mateescu Alexandru, P˘aun Gheorghe, and Salomaa Arto // Theoretical Computer Science. — 1995. — Vol. 141, no. 1. — P. 253–268.

3. ReScue: Crafting Regular Expression DoS Attacks / Shen Yuju, Jiang Yanyan, Xu Chang, Yu Ping, Ma Xiaoxing, and Lu Jian // Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. — New York, NY, USA : Association for Computing Machinery. — 2018. — ASE '18. — P. 225–235.

4. Regulator: Dynamic Analysis to Detect ReDoS / McLaughlin Robert, Pagani Fabio, Spahn Noah, Kruegel Christopher, and Vigna Giovanni // 31st USENIX Security Symposium (USENIX Security 22). — Boston, MA : USENIX Association. — 2022. — Aug. — P. 4219–4235. 30.

5. RegexScalpel: Regular Expression Denial of Service (ReDoS) Defense by Localize-and-Fix / Li Yeting, Sun Yecheng, Xu Zhiwu, Cao Jialun, Li Yuekang, Li Rongchen, Chen Haiming, Cheung Shing-Chi, Liu Yang, and Xiao Yang // 31st USENIX Security Symposium (USENIX Security 22). — Boston, MA : USENIX Association. — 2022. — Aug. — P. 4183–4200.