Регулярные выражения с повторными переменными. Обзор формализмов и алгоритмы сопоставления.

Выполнила: Исмагилова Д.Н., МГТУ им. Баумана
Научный руководитель: Непейвода А.Н., МГТУ им. Баумана,
ИПС им. А.К. Айламазяна РАН

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

Проблематика

Расширенные регулярные выражения мало изучены, а потому не имеет достаточной научной базы для быстрого сопоставления с ними.

Кроме того не существует эффективных инструментов для работы с расширенными регулярными выражениями.

План доклада

- Обзор формализмов расширенных регулярных выражений и их свойства
- Эффективный разбор академических регулярных выражений на примере Re2
- Расширение существующего механизма на расширенные регулярные выражения
- Обращение классических регулярных выражений
- Ввод формализма, удобного для преобразований регулярных выражений
- Обращение расширенных регулярных выражений
- Результаты

Регулярные выражения с обратными ссылками

$(\mathtt{a} \mid \mathtt{b})^*\mathtt{c}$ - классическое регулярное выражение
$(\mathtt{a}^*)\mathtt{b}\backslash\mathtt{1}\mathtt{b}\backslash\mathtt{1}$ - регулярное выражение с обратными ссылками, соответствующее языку $\{\mathtt{a}^\mathtt{n}\mathtt{b}\mathtt{a}^\mathtt{n}\mathtt{b}\mathtt{a}^\mathtt{n}\}$, не являющимся даже контекстно-свободным.

Алгебра Клини

Алгебра Клини — это полукольцо $\langle \mathcal{A}, +, \cdot, \emptyset, 1 \rangle$, содержащее дополнительную операцию *, идемпотентное по $+$, и удовлетворяющее следующим аксиомам:

- $\forall a \in \mathcal{A}(1 + a \cdot a^* = a^* \;\&\; 1 + a^* \cdot a = a^*)$ (закон раскрытия итерации слева и справа);
- $\forall a, x \in \mathcal{A}((a{\cdot}x + x = x \Rightarrow a^* {\cdot} x + x = x) \;\&\; (x{\cdot}a + x = x \Rightarrow x{\cdot}a^* + x = x))$ (левая и правая лемма Ардена).

$a(ba)^* = (ab)^* a$ - sliding

$a^*(ba^*)^* = (a + b)^*$ - denesting

Семантики

Если расширенное регулярное выражение находится в **ε-семантике**, то неинициализированные при разборе ссылки заменяются на ε.

При **$\emptyset$-семантике** слово, сопоставляещееся по пути разбора с неинициализированными ссылками, считается не соответствующим регулярному выражению.

$\mathtt{a} \mid \mathtt{b}\backslash\mathtt{1}$ — синтаксически некорректно

$(\mathtt{a} \mid \mathtt{b}\backslash\mathtt{1})$ сопоставляется только с $\mathtt{a}$.

PCRE2

Скобочные группы могут быть именованными или безымянными.

- в регулярном выражении не может быть ссылки на несуществующую скобочную группу;
- скобочная группа не может входить в регулярное выражение позже, чем ссылка на её номер, за исключением групп 1–7;
- $\emptyset$-семантика.

Выражение `(ab)c\1` *распознаёт язык* {`abcab`}, *тогда как выражение* `a(bc)\1` *распознаёт язык* {`abcbc`}.

Формализм Кампенау-Саломаа-Ю

Все скобочные группы нумеруются автоматически по первому вхождению открывающей скобки.

- Каждое появление обратной ссылки должно быть предварено соответствующей закрытой скобочной группой.
- ε-семантика.

Выражение $(\mathtt{a}^* \mid \mathtt{b}^*)\backslash 1$ *распознаёт язык* $\{\mathtt{a}^{2n}\} \cup \{\mathtt{b}^{2m}\}$, *тогда как выражение* $((\mathtt{a}*)\backslash 1 \mid (\mathtt{b}*)\backslash 1)$ *некорректно в текущем формализме.*

Формализм Шмидта

Reference word (ref-word) над алфавитом Σ — это выражение над алфавитом $\Sigma \cup \{[_{x_i},]_{x_i}, x_i | i \in \mathbb{N}\}$, где x_i — переменная, а $[_{x_i},]_{x_i}$ — скобки, выделяющие подвыражение для переменной x_i. При этом если выражение $[_{x_i}\omega]_{x_i}$ — ref-word, то ω не содержит x_i.

$$([_{x_1}a[_{x_2}b]_{x_1}c]_{x_2})$$

Регулярные выражения над рекурсивными образцами

Регулярные выражения над алфавитом с переменными, каждая из который в свою очередь соответствует регулярному выражению над рекурсивными образцами.

$$cy^*cy^*, y = bx^*, x = a^* \quad c(ba^n)^m c(ba^n)^m$$

$$(xax)^*, x \in b^*$$

$$(x_1 a x_1)^* x_2 a x_2 \ldots x_i a x_i \ldots, \; x_i \in b^*$$

Re2 vs обратных ссылок

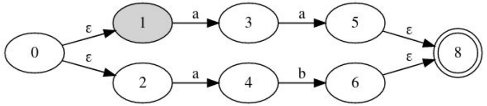
Подходы к сопоставлению строки с регулярным выражением

Переход от регулярного выражения к НКА и последующее использование механизма разбора по НКА с возвратами.

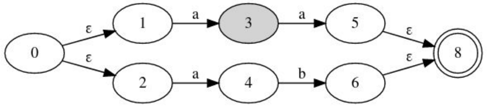
Безопасные реализация без возвратов, с отсутствием поддержки расширенного синтаксиса. Примеры: модули в *Go* и *Rust*, *Re2*.

Разбор с возвратами и без

ab

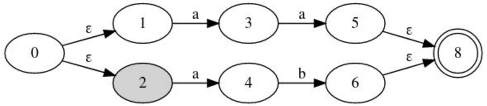


ab

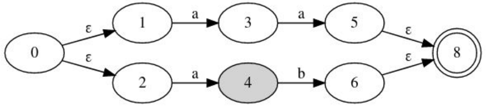


fails, backtrace

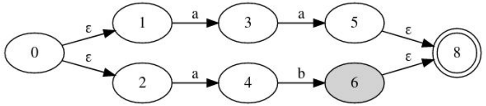
ab

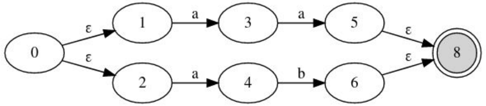


ab



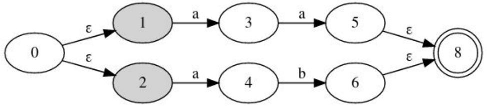
ab



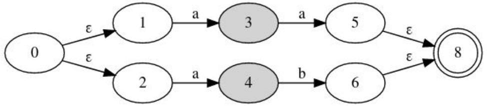


matched successfully

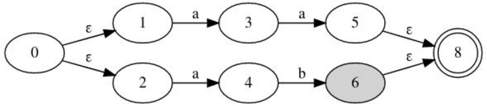
ab

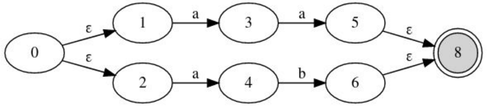


ab



ab





matched successfully

1-однозначность

Под 1-однозначностью понимается свойство регулярных выражений, когда существует не более, чем один вариант успешного сопоставления для любого входного слова.

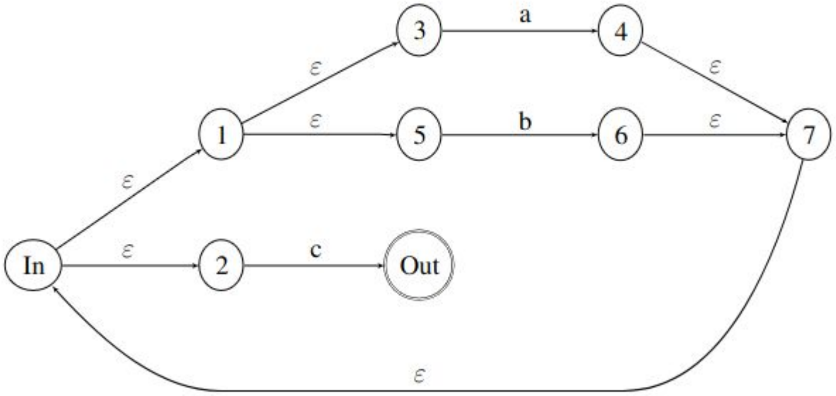
Входную строку $\mathtt{aaa}$ *по не 1-однозначному регулярному выражению* $(\mathtt{aa})^*\mathtt{a}^*$ *можно разобрать двумя разными способами:*

- $\mathtt{aa}$ *соответствует* $(\mathtt{aa})^*$, $\mathtt{a}$ *соответствует* $\mathtt{a}^*$;
- $\mathtt{aaa}$ *соответствует* $\mathtt{a}^*$, *а шаблон* $(\mathtt{aa})^*$ *сопоставляется с пустым словом.*

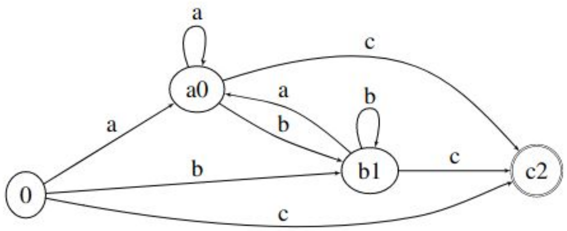
Автомат Глушкова

$$(a \mid b)^* c$$

Автомат Томпсона



Автомат Глушкова



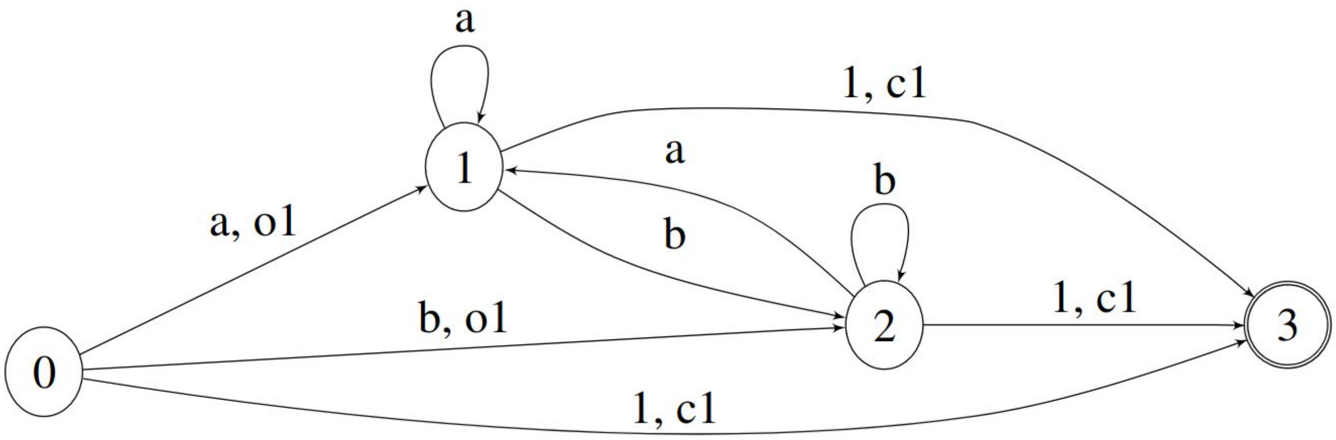
MFA

Формально автомат с памятью, или MFA, определяется как пятерка элементов $< Q, \Sigma, \delta, q_0, F >$, где Q — конечное множество состояний, Σ — это алфавит, $q_0 \cup Q$ — начальное состояние, $F \subseteq Q$ — множество конечных состояний и $\delta : Q \times (\Sigma \cup \{\varepsilon\} \cup \{1, 2, ..., k\}) \to \mathcal{P}(Q \times \{o, c, \diamond\}^k)$ это функция переходов. Элементы $o, c, \diamond$ называются инструкциями над памятью (o — открытие памяти, c — закрытие памяти, $\diamond$ — сохранение памяти в состоянии, в котором она была).

**автомат с памятью считается детерминированным, если он детерминирован в классическом смысле над алфавитом и списком переменных*

MFA

$$[_1\mathtt{a}^* \mid \mathtt{b}^*]_1 \& \mathtt{1}$$



Обращение классических регулярных выражений

- $\beta = \varepsilon \Rightarrow \mathrm{reverse}(\beta) = \varepsilon$;
- $\beta = \mathrm{a} \Rightarrow \mathrm{reverse}(\beta) = \mathrm{a}$;
- $\beta = A|B \Rightarrow \mathrm{reverse}(\beta) = (\mathrm{reverse}(A)|\mathrm{reverse}(B))$;
- $\beta = AB \Rightarrow \mathrm{reverse}(\beta) = \mathrm{reverse}(B)\mathrm{reverse}(A)$;
- $\beta = A^* \Rightarrow \mathrm{reverse}(\beta) = (\mathrm{reverse}(A))^*$.

Регулярное выражение $(\mathtt{a}|\mathtt{b})^*\mathtt{a}(\mathtt{a}|\mathtt{b})$ *недетерминировано. Однако при обращении получается детерминированное выражение* $(\mathtt{a}|\mathtt{b})\mathtt{a}(\mathtt{a}|\mathtt{b})^*$.

Ациклические регулярные выражения с обратными ссылками

- ε, $\emptyset$, а также все буквы алфавита Σ принадлежат $\mathcal{P}_{\mathbf{ACREG}}$;
- если $\tau_1, \tau_2 \in \mathcal{P}_{\mathbf{ACREG}}$, тогда $\tau_1\tau_2$, $\tau_1 \mid \tau_2$, а также $\tau_1{}^*$ принадлежат $\mathcal{P}_{\mathbf{ACREG}}$;
- если $i \in \mathbb{N}$, $\tau \in \mathcal{P}_{\mathbf{ACREG}}$, и τ не содержит $[_i\tau']_i$ ни для какого τ', тогда $\&i$ и $[_i\tau]_i$ принадлежат $\mathcal{P}_{\mathbf{ACREG}}$.

- $j \propto_r i$ — если в r встречается хотя бы одно подвыражение $[_i\tau]_i$ такое, что $\&j$ входит в τ; транзитивное замыкание отношения $\propto_r$ антирефлексивно.
- Для всякого $\&i$, входящего в r найдётся такое r', что $[_ir']_i$ входит в r.

ACREG != Алгебра Клини

$$\forall x, y, z \in \mathcal{A}(xy = yz \Rightarrow x^* y = yz^*)$$

- *В* ε*-семантике положим* $x = [_1 a]_1$, $y = \&1$, $z = aa$. *Тогда* xy *и* yz *оба задают язык* $\{aa\}$, *но* $\mathscr{L}(x^* y) = \{\varepsilon, a^{n+2}\}$, $\mathscr{L}(yz^*) = \{a^{2 \cdot n}\}$ *(*$n \geqslant 0$*).*
- *В* $\emptyset$*-семантике положим* $x = [_1 ba]_1$, $y = \&1 \mid b$, $z = ab \mid aba$. *Тогда* xy *и* yz *оба задают язык* $\{bab, baba\}$, *но* $\mathscr{L}(x^* y) = \{b, ba^{n+1}b, ba^{n+2}\}$, $\mathscr{L}(yz^*) = \{b(ab \mid aba)^n\}$ *(*$n \geqslant 0$*).*

Неоднозначные чтения

В регулярном выражении $([_1\mathtt{a}^*]_1|[_1\mathtt{b}^*]_1|\mathtt{c})\&\mathtt{1}$ *ссылка* $\&\mathtt{1}$ *может относиться как к либо первому либо второму вариантам альтернативы, и соответственно соответствовать* $\mathtt{a}^*$ *либо* $\mathtt{b}^*$*, так и к третьему варианту и соответствовать пустому слову.*

lastinit-неоднозначность

Множество $\mathrm{last}_{i:\mathrm{init}}(\mathtt{r})$ — это множество возможных выражений, инициализирующих ячейку памяти $\mathtt{i}$ после чтения выражения $\mathtt{r}$.

Степень неоднозначности регулярного выражения $\mathtt{r}$ с обратными ссылками по переменной $\mathtt{i}$ — $\mathsf{N}_{\mathrm{ambi}}\mathtt{r}$, равна $\mathtt{n}$, если для $\mathtt{k}$-го оператора чтения $\&\mathtt{i}$ в $\mathtt{r}$ и предшествующего ему выражения $\mathtt{r}'$ $|\mathrm{last}_{i:\mathrm{init}}\mathtt{r}'| = \mathtt{n}_\mathtt{k}$, и $\sum_\mathtt{k} \mathtt{n}_\mathtt{k} = \mathtt{n}$.

Степень $\mathrm{last}_{:\mathrm{init}}$–неоднозначности регулярного выражения $\mathtt{r}$ с обратными ссылками — $\mathsf{N}_{\mathrm{amb}}\mathtt{r}$, равна $\mathtt{n}$, если $\mathtt{n} = \prod_\mathtt{i} \mathsf{N}_{\mathrm{ambi}}\mathtt{r}$, где $\&\mathtt{i}$ входит в $\mathtt{r}$.

Пусть $\mathtt{r} = [_1\mathtt{ba}^*]_1[_2\mathtt{ca}^*]_2([_1\&2\mathtt{ab}]_1 \mid [_2\mathtt{bb}^*]_2)^*$. *Тогда* $\mathrm{last}_{2:\mathrm{init}}\mathtt{r} = \{\mathtt{ca}^*, \mathtt{bb}^*\}$.

СНФ

Ациклическое регулярное выражение $\mathtt{r}$ — в слабой ссылочной нормальной форме, если для каждого оператора чтения по ссылке $\&\mathtt{i}$ и предшествующего ему выражения $\mathtt{r}'$ множество $\mathrm{last}_{\mathrm{i:init}}\mathtt{r}'$ содержит единственный элемент.

Скажем, что $\mathtt{r}$ — в ссылочной нормальной форме (СНФ), если дополнительно к этому каждый каждый оператор записи в память $[_{\mathrm{i}}\mathtt{r}_0]_{\mathrm{i}}$ инициализирует выражение $\mathtt{r}_0$, которое входит в $\mathrm{last}_{\mathrm{i:init}}\mathtt{r}'$, где $\mathtt{r}'$ предшествует оператору чтения $\&\mathtt{i}$.

$$([_1\mathtt{a}^*]_1 \mid [_1\mathtt{b}^*]_1 \mid \mathtt{c})\&1 \to ([_1\mathtt{a}^*]_1\&1 \mid [_1\mathtt{b}^*]_1\&1 \mid \mathtt{c})$$

$$(\mathtt{c}(\&1|[_1\mathtt{a}^*]_1\&1)\mathtt{b})* \to (\mathtt{cb}) * (\mathtt{c}[_1\mathtt{a}^*]_1\&1\mathtt{b}(\mathtt{c}\&1\mathtt{b})*)*$$

Преобразование в СНФ

- $(r_1 \mid r_2)r_3 \to (r_1r_3 \mid r_2r_3)$, $r_1(r_2 \mid r_3) \to (r_1r_2 \mid r_1r_3)$
- $r_0{r_1}^*r_2 \to (r_0r_2 \mid r_0{r_1}^*r_1r_2)$
- $(r_1 \mid r_2)^*r_3 \to ({r_1}^*r_2)^*{r_1}^*r_3$
- $[_i(a \mid b)]_i \to ([_ia]_i \mid [_ib]_i)$
- $(r_1r_2)^*r_1 \to r_1(r_2r_1)^*$

Обращение ACREG в СНФ

- $\beta = [_i A]_i$ и переменная уже была инициализована при обращении $\Rightarrow$ $\mathsf{reverse}(\beta) = \&i$,
- $\beta = [_i A]_i$ и переменная не была инициализована при обращении $\Rightarrow$ $\mathsf{reverse}(\beta) = [_i \mathsf{reverse}(A)]_i$ и переменная добавляется в список инициализированных,
- $\beta = \&i$ и переменная уже была инициализована при обращении $\Rightarrow$ $\mathsf{reverse}(\beta) = \&i$,
- $\beta = \&i$ и переменная не была инициализована при обращении $\Rightarrow$ $\mathsf{reverse}(\beta) = [_i \mathsf{reverse}(A)]_i$ и переменная добавляется в список инициализированных.

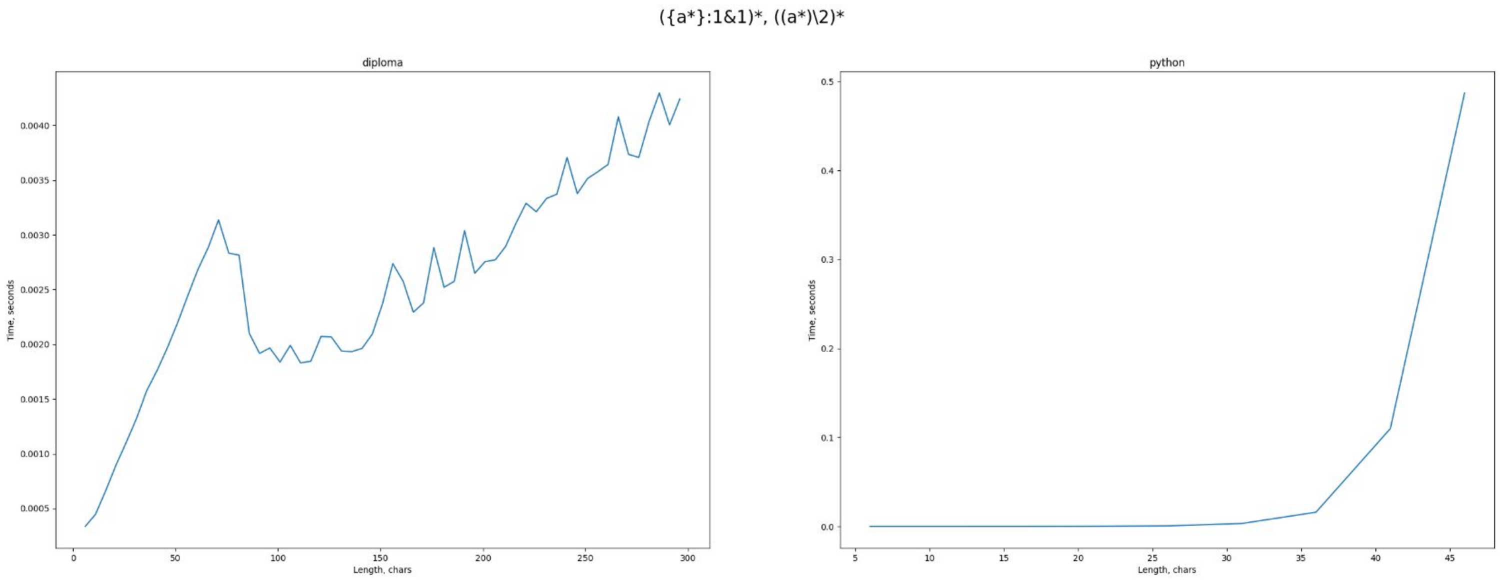
RW блоки

- $[_1\mathtt{b}^*]_1(\&1\,[_1\mathtt{a}^*]_1)^* \to (\mathtt{b}^*|[_1\mathtt{b}^*]_1\&1([_1\mathtt{a}^*]_1\&1)*\mathtt{a}^*)$
- $([_2\mathtt{b}^*]_2\&1\,[_1\mathtt{a}^*]_1\&2)^*$
- $[_1\mathtt{a}^*]_1(\&1\,[_1\mathtt{a}^*]_1)^* \to (\mathtt{a}^*|[_1\mathtt{a}^*]_1(\&1\,[_1\mathtt{a}^*]_1)*\&1\mathtt{a}^*)$
- $[_2\mathtt{a}^*\mathtt{d}]_2([_1\mathtt{a}^*]_1\mathtt{b}\&1\,[_2\&1\mathtt{c}*\mathtt{c}]_2)^*$

Примеры

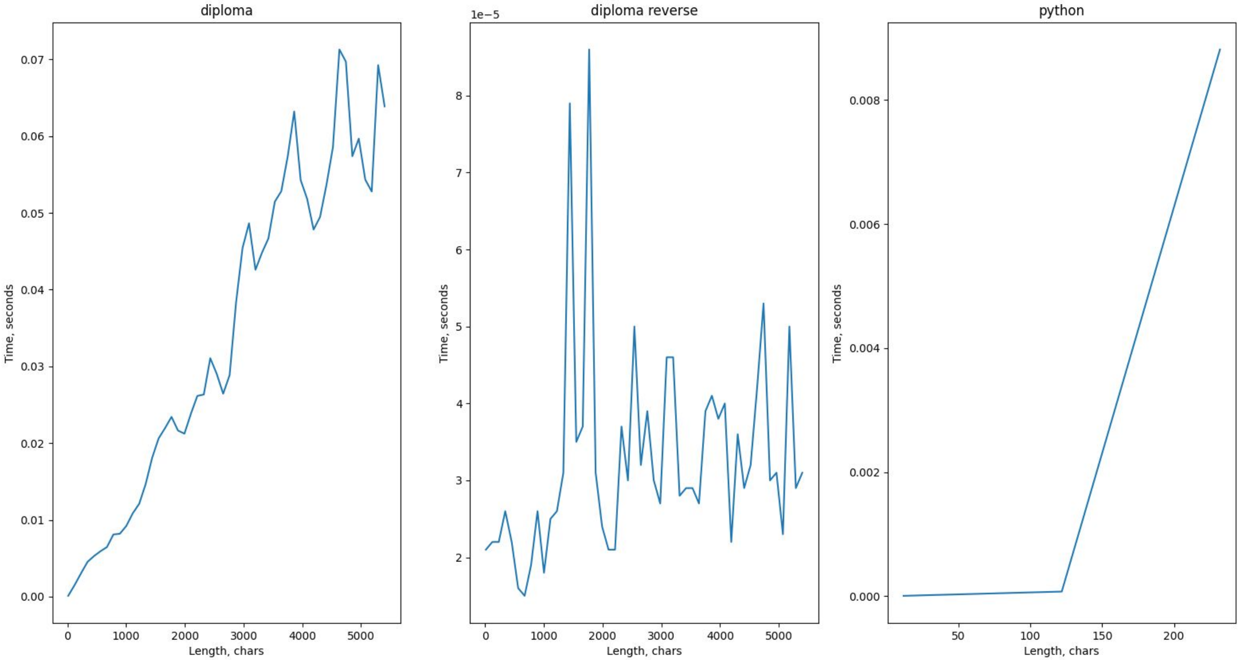
```
({a*}:1|b)&1
BNF: ({a*}:1&1|b&1)
Reverse: ({a*}:1&1|&1b)
({a*}:1|{b*}:1|c)&1
BNF: ({a*}:1&1|{b*}:1&1|c&1)
Reverse: ({a*}:1&1|{b*}:1&1|&1c)
{a*}:1(&1{a*}:1)*
BNF: (a*|{a*}:1(&1{a*}:1)*&1a*)
Reverse: (a*|a*{a*}:1(&1{a*}:1)*&1)
({b*}:2&1{a*}:1&2)*
BNF: (ε|{b*}:2(a*&2|({a*}:1&2{b*}:2&1)*{a*}:1&2{b*}:2&1a*&2))
Reverse: (ε|({b*}:2a*|{b*}:2a*{a*}:1&2{b*}:2&1({a*}:1&2&2&1)*)&2)
(c(&1|{a}:1&1)b)*
BNF: (cb)*(c{a}:1&1b(c&1b)*)*
Reverse: ((b{a}:1c)*b&1&1c)*(bc)*
(({a*}:1|b)(&1|b))*
BNF: (b|bb)*({a*}:1&1(bb)*|a*b(bb)*|{a*}:1&1(bb)*b&1(bb)*(b&1(bb)*)*|{a*}:1b(bb)*b&1(bb)*(b&1(bb)*)*)*
Reverse: ((bb)*{a*}:1&1|(bb)*ba*|((bb)*{a*}:1b)*(bb)*&1b(bb)*&1&1|((bb)*{a*}:1b)*(bb)*&1b(bb)*b&1)*(b|bb)*
```

Результаты


({a*}:1&1)*, ((a*)\2)*
diploma
Time, seconds
Length, chars
0.0005
0.0010
0.0015
0.0020
0.0025
0.0030
0.0035
0.0040
0
50
100
150
200
250
300
python
Time, seconds
Length, chars
0.0
0.1
0.2
0.3
0.4
0.5
5
10
15
20
25
30
35
40
45

Результаты

({a*}:1b|&1)*, ((a*)b|\2)*



Список литературы

1. Kozen, Dexter. 1994. “A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events.” Information and Computation 110 (2): 366–90. https://doi.org/https://doi.org/10.1006/inco.1994.1037.
2. Campeanu, Cezar & Salomaa, Kai & Yu, Sheng. (2003). A Formal Study Of Practical Regular Expressions.. Int. J. Found. Comput. Sci.. 14. 1007-1018. 10.1142/S012905410300214X.
3. Schmid, Markus. 2016. “Characterising REGEX Languages by Regular Languages Equipped with Factor-Referencing.” Information and Computation (February). https://doi.org/10.1016/j.ic.2016.02.003.
4. Schmid, Markus. (2012). Inside the class of REGEX languages. International Journal of Foundations of Computer Science. 24. 73-84. 10.1007/978-3-642-31653-1_8.
5. Brüggemann-Klein, Anne, and Derick Wood. 1998. “One-Unambiguous Regular Languages.” Information and Computation 140 (2): 229–53. https://doi.org/https://doi.org/10.1006/inco.1997.2688.
6. Gruber, Hermann, and Stefan Gulan. 2010. “Simplifying Regular Expressions.” In Language and Automata Theory and Applications, edited by Adrian-Horia Dediu, Henning Fernau, and Carlos Martín-Vide, 285–96. Berlin, Heidelberg: Springer Berlin Heidelberg.
7. Glushkov, V M. 1961. “THE ABSTRACT THEORY OF AUTOMATA.” Russian Mathematical Surveys 16 (5): 1. https://doi.org/10.1070/RM1961v016n05ABEH004112.
8. Freydenberger, Dominik D., and Markus L. Schmid. 2018. “Deterministic Regular Expressions with Back-References.” CoRR abs/1802.01508. http://arxiv.org/abs/1802.01508.
9. Hazel, Philip. n.d. “Официальное Руководство По PCRE2 (электронный ресурс).” https://www.pcre.org/current/doc/html/index.html.
10. Google “Репозиторий библиотеки Re2 (электронный ресурс).” https://github.com/google/re2.