СЛОЖНОСТЬ РАСПОЗНАВАНИЯ ПРИНАДЛЕЖНОСТИ СЛОВА РЕГУЛЯРНОМУ ЯЗЫКУ, ЗАДАННОМУ НАБОРОМ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

Александров Д.Е.

22 апреля 2014 года

Регулярные выражения

Далее рассматриваются PCRE-совместимые регулярные выражения над языком Σ со следующими обозначениями:

- "$R_1 R_2$" — конкатенация выражений R_1 и R_2
- "$(R_1 \mid R_2)$" — объединение (ИЛИ) выражений R_1 и R_2
- "`R*`" — конкатенация произвольного числа выражений R
- "`R{m,n}`" — конкатенация от m до n раз выражений R
- "`.`" — объединение всех символов языка
- "$[a_1 a_2 \ldots a_n]$" — объединение символов $a_1, a_2 \ldots a_n$
- "$[\hat{} a_1 a_2 \ldots a_n]$" — объединение всех символов языка кроме символов $a_1, a_2 \ldots a_n$

Варианты решения задачи поиска

Построение конечного автомата:

- НКА (NFA)
- ДКА (DFA)
- МДКА (MDFA)
- Д2КА (D^2FA)
- Дуальный КА (DualFA)
- Расширенный ДКА (XFA)

Недетерминированный конечный автомат

Заданы:

- Q — множество состояний
- $A \subseteq Q$ — множество конечных состояний
- $q_0 \in Q$ — начальное состояние
- $\delta\colon Q \times \Sigma \to 2^Q$ — функция перехода

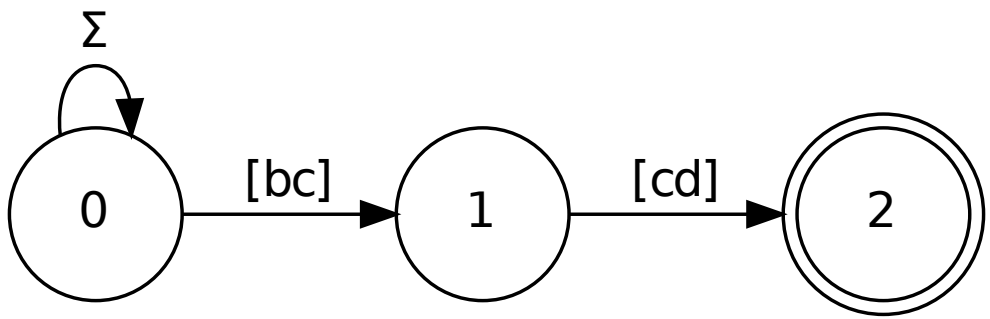
Определим функцию δ^*:

- $\forall q \in Q, a \in \Sigma\colon \delta^*(q, a) = \delta(q, a)$
- $\forall q \in Q, a \in \Sigma, \alpha \in \Sigma^*\colon \delta^*(q, \alpha a) = \bigcup\limits_{p \in \delta^*(q,\alpha)} \delta(p, a)$

Слово α представимо НКА, если $\delta^*(q_0, \alpha) \cap A \neq \emptyset$

Недетерминированный конечный автомат

Пример НКА для выражения ".*[bc][cd]":



$$\{0\} \xrightarrow{a} \{0\} \xrightarrow{b} \{0,1\} \xrightarrow{c} \{0,1,2\} \xrightarrow{d} \{0,2\}.$$

Недетерминированный конечный автомат

Хорошо:

- малое число состояний

Плохо:

- высокая сложность вычисления

Детерминированный конечный автомат

Заданы:

- Q — множество состояний
- $A \subseteq Q$ — множество конечных состояний
- $q_0 \in Q$ — начальное состояние
- $\delta\colon Q \times \Sigma \to Q$ — функция перехода

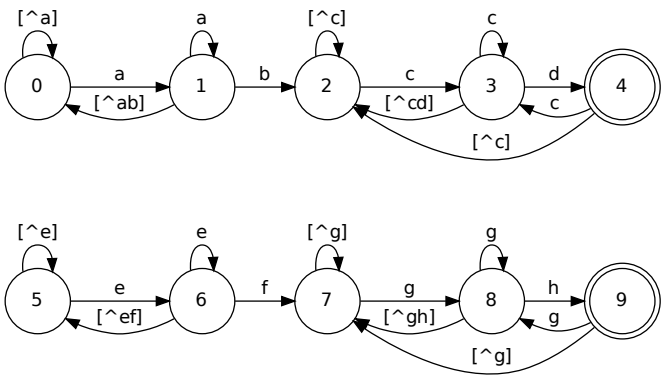
Определим функцию δ^*:

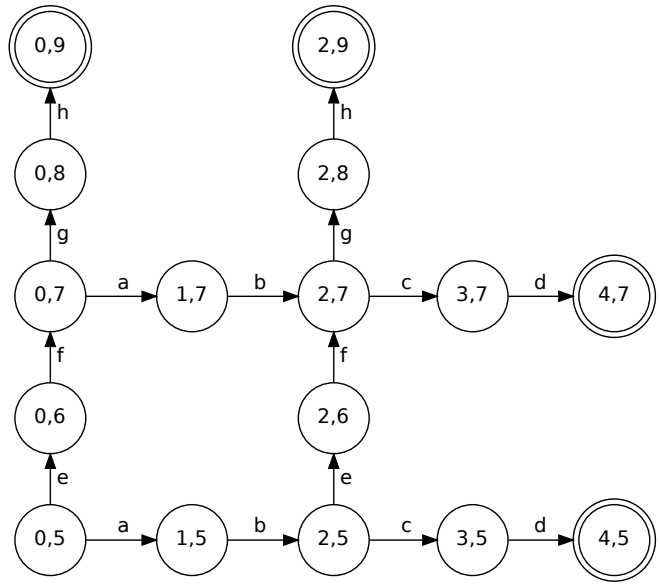
- $\forall q \in Q, a \in \Sigma\colon \delta^*(q, a) = \delta(q, a)$
- $\forall q \in Q, a \in \Sigma, \alpha \in \Sigma^*\colon \delta^*(q, \alpha a) = \delta\left(\delta^*(q, \alpha), a\right)$

Слово α представимо ДКА, если $\delta^*(q_0, \alpha) \in A$

Детерминированный конечный автомат

Пример ДКА для выражений ".*ab.*cd" и ".*ef.*gh":





Детерминированный конечный автомат

Хорошо:

- низкая сложность вычисления

Плохо:

- большое число состояний

МДКА

Исходные регулярные выражения делятся на группы, по которым строятся отдельные ДКА.
Причем группировка производится так, чтобы число состояний каждого автомата не превышало заданного значения.

МДКА

Хорошо:

- фиксированная сложность вычисления
- возможно небольшое число состояний

Плохо:

- возможна высокая сложность вычисления
- нетривиальное построение

Yu F. et al. Fast and memory-efficient regular expression matching for deep packet inspection //Architecture for Networking and Communications systems, 2006. ANCS 2006. ACM/IEEE Symposium on. – IEEE, 2006. – С. 93-102.

Д2КА

Заданы:

- Q — множество состояний
- $A \subseteq Q$ — множество конечных состояний
- $q_0 \in Q$ — начальное состояние
- $\delta\colon G \to Q$ — функция перехода, где $G \subseteq Q \times \Sigma$
- $\delta^e\colon Q \to Q$ — функция перехода на случай “ошибки”

Алгоритм работы:

Дано: текущее состояние $q \in Q$, входной символ $a \in \Sigma$

Надо: новое состояние q'

$q' := q$

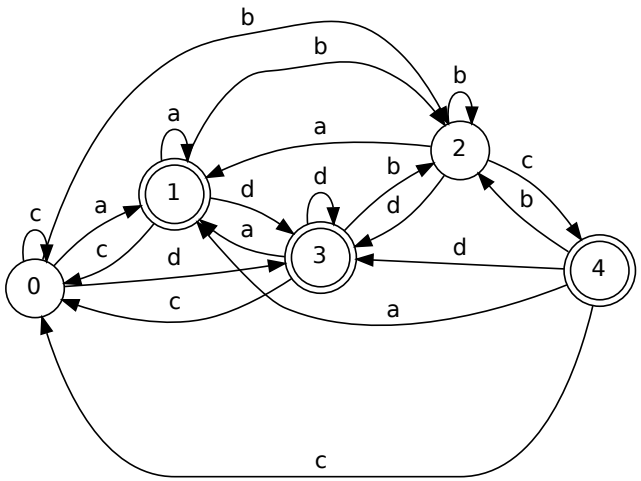
Пока $(q', a) \notin G$

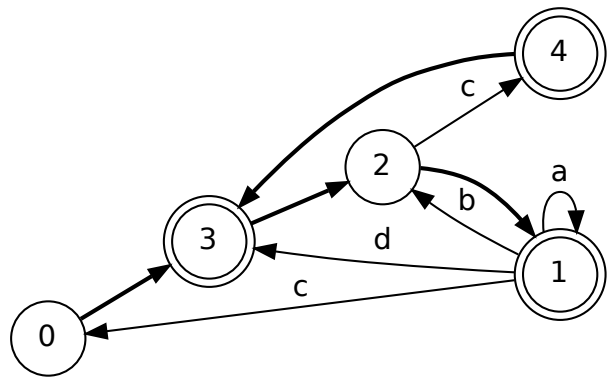
$q' := \delta^e(q')$

$q' := \delta(q', a)$

Д2КА

Пример ДКА и Д2КА для выражений ".*a+", ".*b+c" и ".*c*d+":





Д2КА

Хорошо:

- сокращение общего числа переходов

Плохо:

- возможна высокая сложность вычисления

Kumar S. et al. Algorithms to accelerate multiple regular expressions matching for deep packet inspection //ACM SIGCOMM Computer Communication Review. – ACM, 2006. – Т. 36. – №. 4. – С. 339-350.
Becchi M., Crowley P. An improved algorithm to accelerate regular expression evaluation //Proceedings of the 3rd ACM/IEEE Symposium on Architecture for networking and communications systems. – ACM, 2007. – С. 145-154.

Дуальный автомат

Заданы:

- ДКА с дополнительными “условными” и “расширенными” переходами
- Линейный конечный автомат — НКА, в котором из любого состояния можно попасть в не более чем 1 состояние, отличное от данного, а также не существует двух различных состояний, из которых можно перейти в третье.

Дуальный автомат

Алгоритм работы:

Дано: ДКА в состоянии $q \in Q$, функция перехода δ,
ЛКА в состоянии $q_l \subseteq Q_l$, функция перехода δ_l,
входной символ $a \in \Sigma$,

Надо: новые состояния q' и q'_l

$q' := \delta(q, a)$

$q'_l := \delta_l(q_l, a)$

Если переход $q \xrightarrow{a} q'$ — расширенный с меткой $p \subseteq Q_l$, то

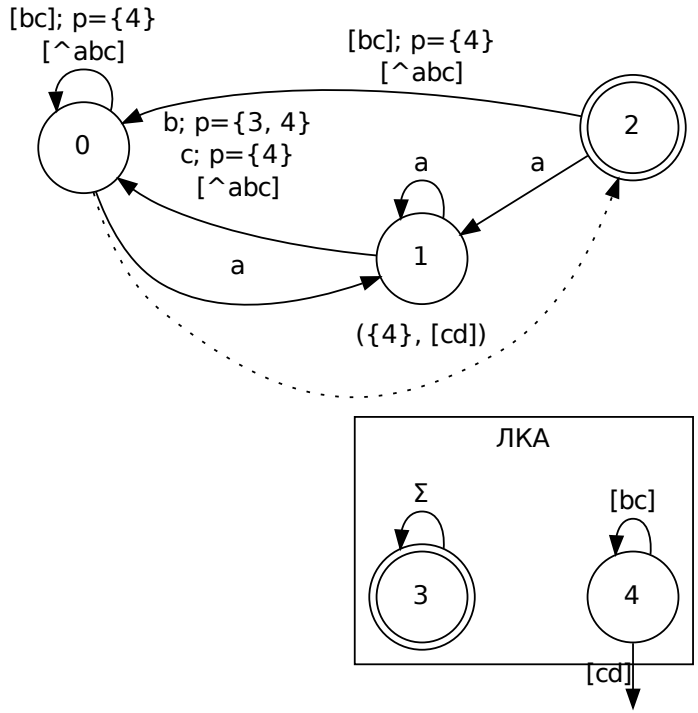
$q'_l := q'_l \cup p$

Если есть условный переход $q' \xrightarrow{(q_l, a)} q''$, то

$q' := q''$

Дуальный автомат

Пример дуального автомата для выражения ".*ab.*" и ".*[bc]+[cd]":



Дуальный автомат

Хорошо:

- возможно небольшое число состояний
- ЛКА реализуется через простейшие битовые операции и использует относительно немного памяти

Плохо:

- возможна высокая сложность вычисления

Liu C., Wu J. Fast Deep Packet Inspection with a Dual Finite Automata. – 2013.

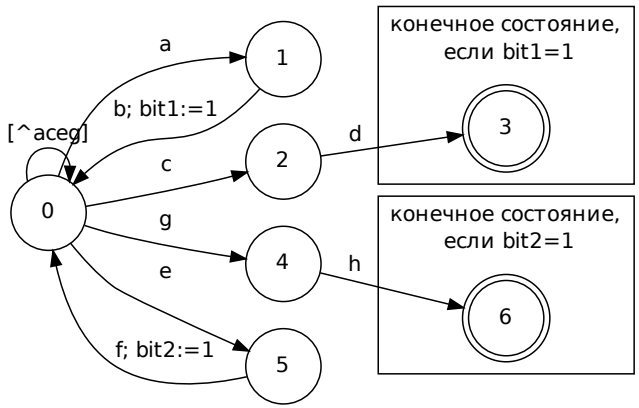
РДКА

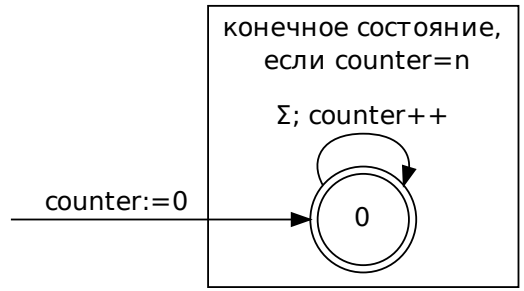
Заданы:

- набор счетчиков
  - битовых
  - числовых
- ДКА с “расширенными” переходами, изменяющими значения счетчиков

РДКА

Пример РДКА для выражений ".*ab.*cd", ".*ef.*gh" и ".{n}":





РДКА

Хорошо:

- возможно небольшое число состояний

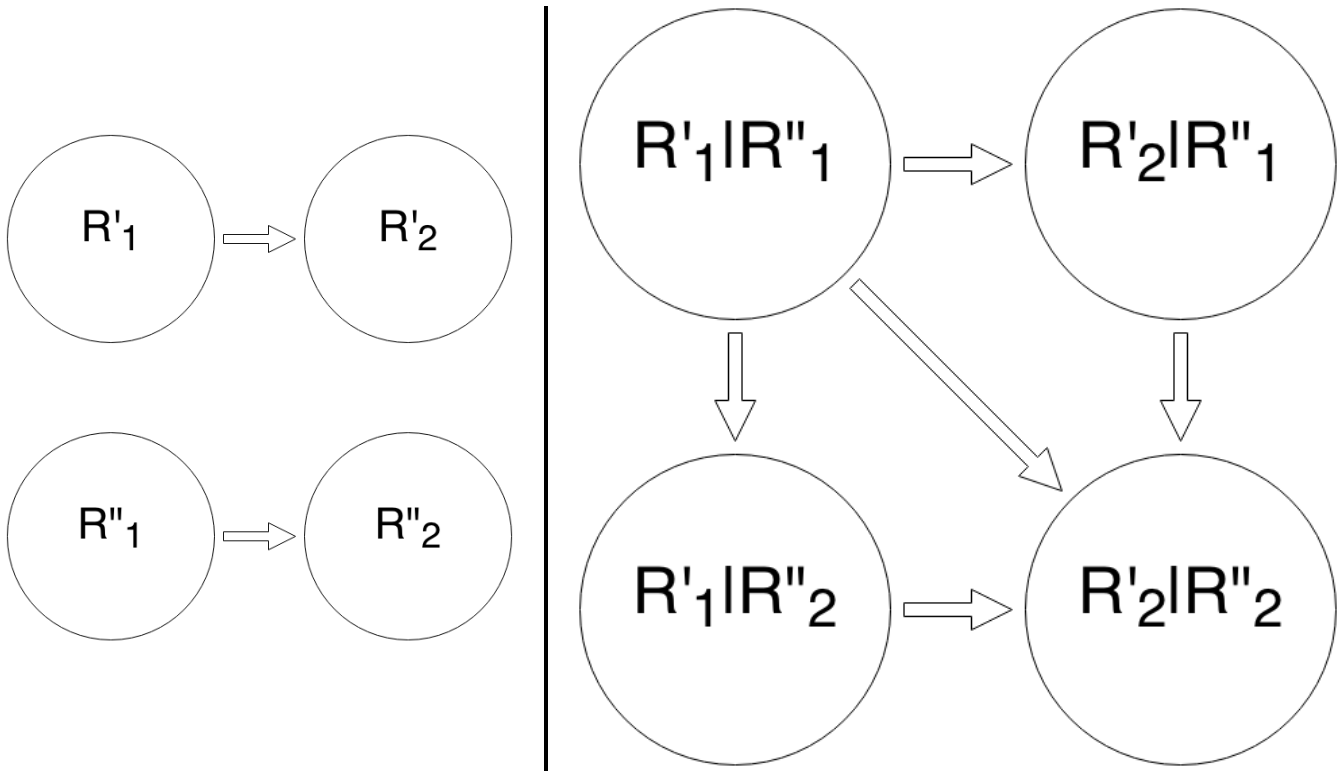
Плохо:

- возможна высокая сложность вычисления
- нетривиальное построение

Smith R. et al. Deflating the big bang: fast and scalable deep packet inspection with extended finite automata //ACM SIGCOMM Computer Communication Review. – ACM, 2008. – Т. 38. – №. 4. – С. 207-218.

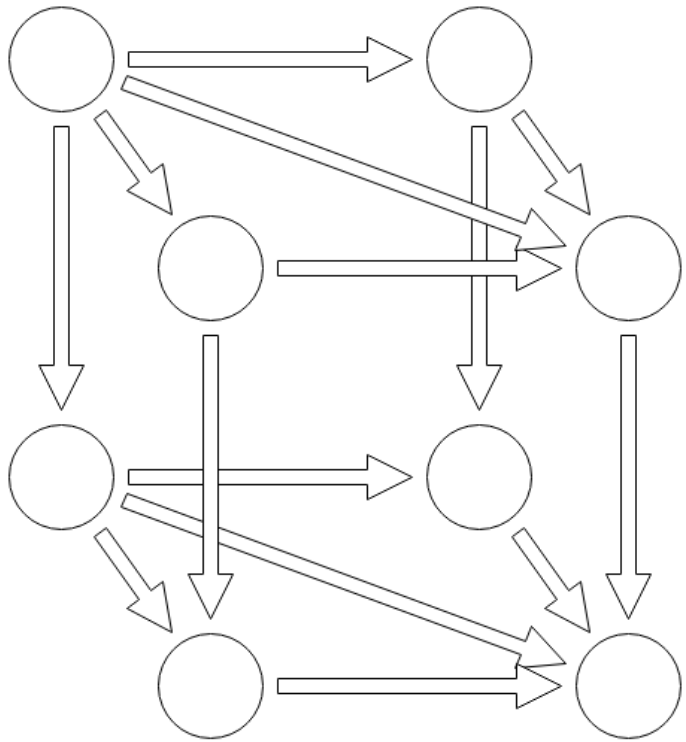
Изменение выражений

Проблема:
объединение выражений вида ".$*R'_1.*R'_2.*$" и ".$*R''_1.*R''_2.*$"
ведет к быстрому росту числа состояний



Изменение выражений

Схема КА для трех выражений



Изменение выражений

Дано:

набор регулярных выражений $R^i = .*\mathtt{R}_1^{\mathtt{i}}.*\mathtt{R}_2^{\mathtt{i}}.*$

Предложение:

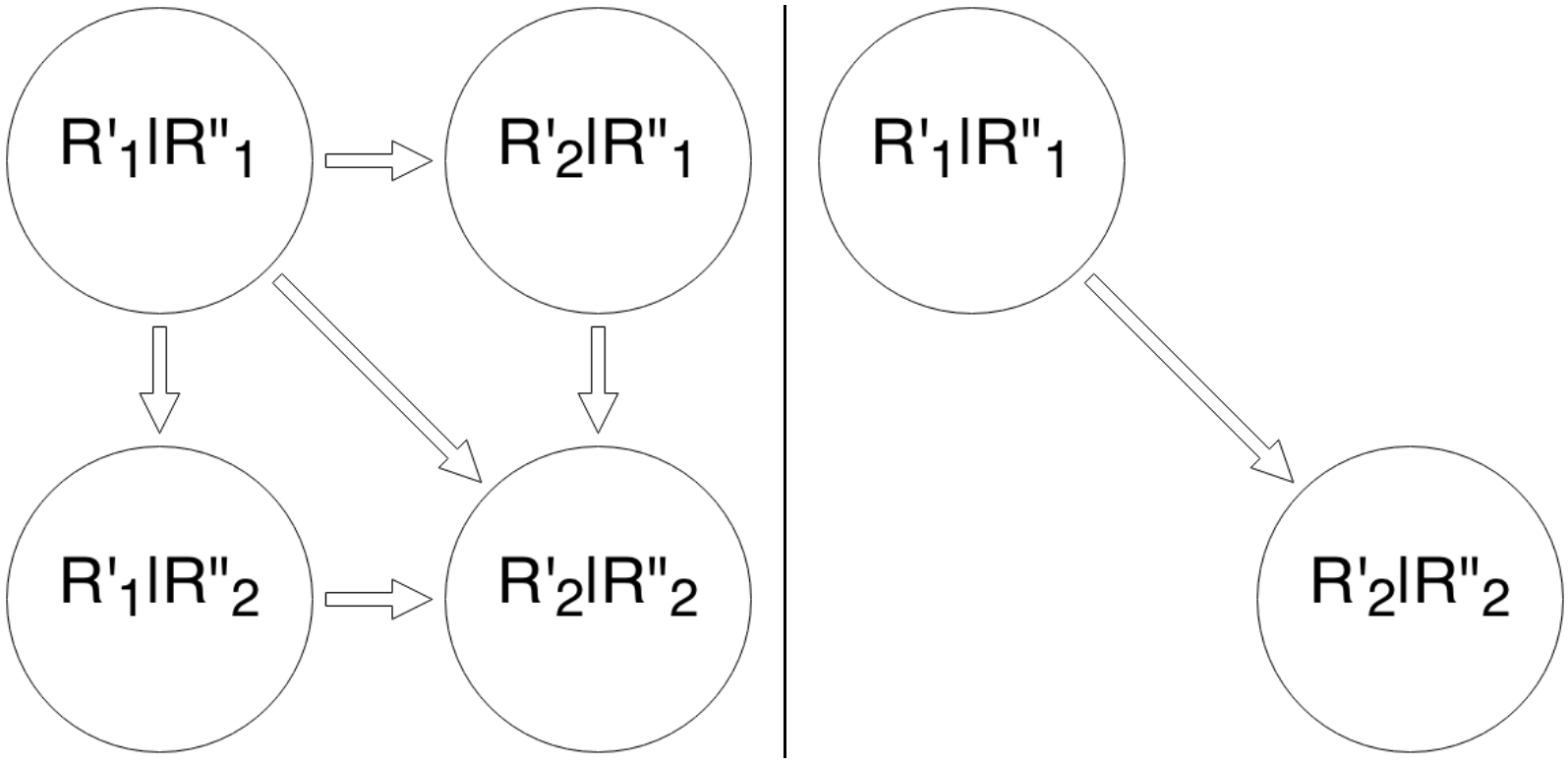
выбираем два выражения R^{i_1} и R^{i_2}

заменяем их на одно: $.*(\mathtt{R}_1^{\mathtt{i}_1}|\mathtt{R}_1^{\mathtt{i}_2}).*(\mathtt{R}_2^{\mathtt{i}_1}|\mathtt{R}_2^{\mathtt{i}_2}).*$

Результат:

- сокращение роста состояний KA при объединении
- ошибки при поиске (лишние слова представимы автоматом)

Изменение выражений



Изменение выражений

Вопрос:

Как можно оценивать ошибку при таком подходе?

Через $G_i(R)$ обозначим число слов длины i, задаваемых регулярным выражением R

Тогда величину ошибки можно оценить для произвольной длины l:

$$Err_l\left(R^1, R^2\right) = \frac{G_l(.*(\mathrm{R}_1^1|\mathrm{R}_1^2).*(\mathrm{R}_2^1|\mathrm{R}_2^2).*) - G_l(\mathrm{R}^1|\mathrm{R}^2)}{G_l(\mathrm{R}^1|\mathrm{R}^2)}$$

Изменение выражений

Как оценить ошибку $Err_l\left(R^1, R^2\right)$, не вычисляя в явном виде?

Пусть далее все регулярные подвыражения $\mathtt{R}_\mathtt{i}^\mathtt{j}$ таковы, что $\forall \alpha \in L(\mathtt{R}_\mathtt{i}^\mathtt{j}) \nexists \beta \in L(\mathtt{R}_\mathtt{i}^\mathtt{j}), \beta \neq \alpha\gamma_1, \gamma_2 \in \Sigma^* \colon \alpha = \gamma_1\beta\gamma_2$

Ведем функцию $P_i(R)$:

$$P_i(R) = \frac{G_i(R)}{|\Sigma|^i}$$

Тогда, если $\dfrac{\sum\limits_{i=0}^{l} P_i\left(\mathtt{R}_1^1\right)}{\sum\limits_{i=0}^{l} P_i(\mathtt{R}_1^2)} \leq \varepsilon$ и $\dfrac{\sum\limits_{i=0}^{l} P_i\left(\mathtt{R}_2^1\right)}{\sum\limits_{i=0}^{l} P_i(\mathtt{R}_2^2)} \leq \varepsilon$, то

$$Err_l(R^1, R^2) \leq \varepsilon$$

Изменение выражений

Кроме того, если $\frac{\sum_{i=0}^{\infty} P_i(\mathtt{R}_1^1)}{\sum_{i=0}^{\infty} P_i(\mathtt{R}_1^2)} \leq \varepsilon$ и $\frac{\sum_{i=0}^{\infty} P_i(\mathtt{R}_2^1)}{\sum_{i=0}^{\infty} P_i(\mathtt{R}_2^2)} \leq \varepsilon$, то

$$Err_l(R^1, R^2) \leq \varepsilon$$