Homework №6

Вадим Шабашов

Задание 1

Равны ли данные регулярные выражения над алфавитом $\{a, b\}$:

- $b^*a((a|b)b^*a)^*$
- $((a|b)^*ba|a)(aa)^*$

Решение:

1. Кажется, что данные регулярки равны. Довольно сложно строго доказать, не строя при этом автоматы, но уж очень не хочется строить их :)

   Попробую аккуратно объяснить с помощью того, что все строки первого принимаются вторым и наоборот.

2. Поймем, какие слова кодирует вторая регулярка:

$$((a|b)^*ba|a)(aa)^* \Rightarrow \left[\begin{matrix}(a|b)^*ba(aa)^* \\ a(aa)^*\end{matrix}\right. \Rightarrow \left[\begin{matrix}vba^m, & v = \{a, b\}^* \text{ и } m \text{ — нечетное} \\ a^m, & m \text{ — нечетное}\end{matrix}\right.$$

   То есть можно понимать так, что кодируем все слова, в которых число всех a в конце нечетно.

3. Для первой регулярки сначала проверим, что принимаем все слова второй.

   - a^m, где m — нечетное:
     Регулярка принимает, т.к. $b^*a((a|b)b^*a)^* = a(aa)^*$
   - $vba^m, \quad v = \{a, b\}^*$ и m — нечетное.
     Пусть $m = 2k + 1$. Тогда раскроем регулярку так:

$$b^*a((a|b)b^*a)^* = b^*a((a|b)b^*a)^* \underbrace{(aa)...(aa)}_{k}$$

     Тогда конец с a^{2k} как раз покроется. Тогда останется показать, что можем принять слово vba ($v \in \{a, b\}^*$) регуляркой $b^*a((a|b)b^*a)^*$.
     В самом общем виде слово $vba = b^{j_1}a^{i_1}...b^{j_n}a^{i_n}b^{j_{n+1}}a$, где $n \geq 0$, $j_l \geq 1$, $i_l \geq 1$.

     – Если $n = 0$, то слово $b^{j_1}a^m$ принимается, т.к. $b^*a((a|b)b^*a)^* = b^*a(aa)^*$
     – Если $n > 0$.
       Сначала с помощью b^*a частью от регулярки уничтожим $b^{j_1}a$.
       Остается $a^{i_1-1}b^{j_2}a^{i_2}...b^{j_n}a^{i_n}b^{j_{n+1}}a$ и хотим понять, можно ли его принять регуляркой $((a|b)b^*a)^*$.
       Заметим, что дальше можно все свести к индуктивному процессу, где просто будет уничтожать на каждом шаге $a^{i_l-1}b^{j_l}a$ спереди. Как это сделать покажу на примере $a^{i_1-1}b^{j_2}a$:

* Сначала уничтожаем лишние a.
Если $(i_1 - 1)$ — четное и равно $2p$, то $((a|b)b^*a)^* = \underbrace{(aa)...(aa)}_{p}((a|b)b^*a)^*$.
Остается $b^{j_2}a^{i_2}...b^{j_n}a^{i_n}b^{j_{n+1}}a$.
Если $(i_1 - 1)$ — нечетное и равно $2p+1$, то опять берем $((a|b)b^*a)^* = \underbrace{(aa)...(aa)}_{p}((a|b)b^*a)^*$.
Остается $ab^{j_2}a^{i_2}...b^{j_n}a^{i_n}b^{j_{n+1}}a$.
* Далее в зависимости от того, что у нас спереди после предыдущего шага, a или b, берем соответственно $((a|b)b^*a)^* = ab^*a$ или $((a|b)b^*a)^* = bb^*a$.
Тогда остается показать, что $a^{i_2-1}...b^{j_n}a^{i_n}b^{j_{n+1}}a$ принимается регуляркой $((a|b)b^*a)^*$. То есть в точности показали, как избавиться спереди от $a^{i_1-1}b^{j_2}a$.
* Заметим, что на последнем шаге будем уничтожать $a^{i_n-1}b^{j_{n+1}}a$. И, действуя точно по алгориму-индукции выше, мы целиком все уничтожим регуляркой.

4. Теперь надо показать, что первой регуляркой не принимаются другие слова. А другие слова — это у которых на конце четное число a.

   * a^m, где m — четное.
   Его принять нельзя, т.к. без b регулярка упрощается к $b^*a((a|b)b^*a)^* = a(aa)^*$, а здесь обязательно нечетное число a.
   * vba^m, где $v \in \{a,b\}^*$ и m — четное.
   Попробуем получить хоть какое-нибудь слово с четным числом a на конце. Возможны на две ситуации:
     - Самый правый b берется из b^* из регулярки $b^*a((a|b)b^*a)^*$.
     Тогда все a^m получились из $a((a|b)b^*a)^*$. Там не может быть b, поэтому регулярка упрощается к $a(aa)^*$, а это опять штука, которая не может дать a^m для m — четного.
     - Самый правый b берется из $((a|b)b^*a)^*$. Не важно, откуда именно b берется: из b^* или из $(a|b)$. Главное, что в любом из этих случаев правее стоит одна a в $(a|b)b^*a$. После этого идет $((a|b)b^*a)^*$. Это значит, что a^m должно взяться из $a((a|b)b^*a)^* = a(aa)^*$. Опять получили эту ситуацию, которая не выполняется.

   Таким образом показали, что слово, которое кончается на четное число a не кодируется первой регуляркой.

5. Итак, явно нашел, какие именно слова кодирует вторая регулярка, а затем показал, что первая регулярка кодирует их же и ничего кроме них.

Задание 2

* $L = \{a^nb^ma^l | l = n+m\}$
* $L = \{a^n | n$ — простое число$\}$
* $L = \{wbva || w|_a > |v|_b, w \in \{a,b\}^*, v \in \{a,b\}^*\}$

Решение:

* 1. Пусть язык регулярный.
  Тогда по лемме о накачке существует n, что выполняются свойства из леммы.
  Рассмотрим слово $w = b^na^n$. Оно принадлежит языку и длины больше n. Тогда по лемме должно быть разбиение на x, y, z. Причем т.к. $|xy| < n$, то $y = b^i$, где $1 \le i < n$. Но тогда при $k = 0$ слово должно принадлежать языку, но это не так: $b^{n-i}a^n$ — слово не принадлежи языку, т.к. $n - i \ne n$. Противоречие.

- 1. Пусть язык регулярный.

     Тогда по лемме о накачке существует n, что выполняются свойства из леммы.

     Рассмотрим слово $w = a^m$, где $m > n$ и m — простое. Оно принадлежит языку и длины больше n. Тогда по лемме должно быть разбиение на x, y, z.

     В этом разбиении $y = a^i$, где $1 \leq i < n$. По лемме слово $a^{m+(k-1)i}$ должно принадлежать языку $\forall k \in \mathbb{N}_0$. Возьмем $k = m+1$: слово имеет вид $a^{m+m\cdot i} = a^{m(i+1)}$; но такое слово не может принадлежать языку, т.к. $m(i+1)$ — не простое (явно выделились два делителя как минимум; и, т.к. $i \geq 1$ и $m > n \geq 1$, то ни один из них не равен 1). Противоречие.

- 1. Язык регулярен. Покажу явным приведением регулярки.

     Но сначала надо понять структуру слов, которые принимаются языком.

  2. Пусть есть какое-то слово $wbva$, которое принимается. Заметим, что мы можем изменить разбиение на части и слово по-прежнему будет с ним приниматься: мы можем выбрать самое правое b во всем слове. Оно окажется либо тем, которое уже было, либо где-то в v. Обозначим новое слово так: $\tilde{w}b\tilde{v}a$. Заметим, что в новом разбиении могли только уменьшить число b ($|v|_b \geq |\tilde{v}|_b = 0$) и только увеличить число a ($|w|_a \geq |\tilde{w}|_a$). Раз слово исходно принималось, то $||\tilde{w}|_a| > |\tilde{v}|_b$ — тем более будет верно. То есть мы показали, что если слово принимается, то обязательно при разбиении с помощью последнего b оно тоже должно приниматься. Это значит, что принадлежность к языку можно проверять только с помощью приведенного разбиения.

  3. Теперь поймем, какая структура может быть у слова, которое принимаем. Разбиваем его способом, указанным до этого (с выделением последнего b). Тогда, чтобы слово было принято, должны выполниться следующие условия:

     - v — состоит только из a.
     - После v обязательно есть хотя бы одна a.
     - В w есть хотя бы одна a (т.к. хотим $|w|_a \geq 1 > |v|_b = 0$).

     А из этих условий легко составить регулярку:

     $$(a|b)^*a(a|b)^*ba^+$$

     Здесь просто первые два пункта комбинируются в a^+. Последний пункт, что в w есть хотя бы одна a: $(a|b)^*a(a|b)^*$.

  4. Раз есть регулярка, то есть и ДКА; то есть регулярный язык.

Задание 3

Построить КС грамматику для языка арифметических выражений с операциями + и * над натуральными числами в десятичной записи.

Решение:

1. Введем:

   - N — терминал для обозначения натурального числа
   - A — произвольное арифметическое выражение
   - B — выражение, где не может быть + на верхнем уровне (имеет смысл слагаемого)
   - C — выражение, где не может быть ни +, ни * на верхнем уровне (либо произвольное выражение в скобках, либо число).

Тогда:

$$
\begin{array}{ll}
V_T & = \{+, *, N, (,)\} \\
V_N & = \{A, B, C\} \\
S & = A \\
P: & A \to B + A \\
 & A \to B \\
 & B \to C \cdot B \\
 & B \to C \\
 & C \to (A) \\
 & C \to N
\end{array}
$$