Содержание

1 Задача 1

Определить регулярность языка $L = \{w \;\Big|\; |\,w\,|_{aba} = |\,w\,|_{ab} \;\&\; w \in \{a, b, c\}^*\}$

1.1 Решение

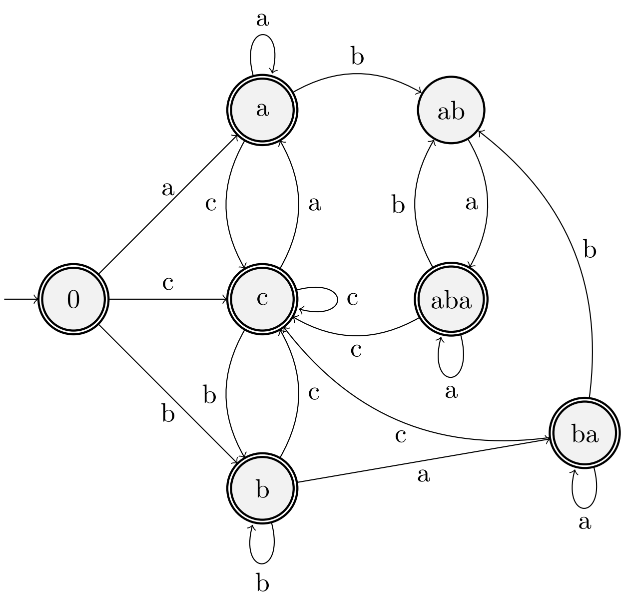

Рис. 1: Автомат для языка L

Автомат построен на основе суффиксов слов. Если ниразу не встретилось подслово ab, то такое слово нам подходит. Если встречается суфикс ab, то за ним же должен следовать символ a, чтобы уравновесить ab и aba, иначе сделать равным количество вхождений aba и ab не получится, т.к. каждое вхождение aba влечет за собой вхождение ab. Поэтому из состояния ab по b, c мы попадаем в ловушку, а по символу a попадаем в конечное состояние aba.

2 Задача 2

Проанализировать язык истинных выражений, представляющих собой утверждение вида $N_1 + N_2 > N_0$, где N_0, N_1 и N_2 - двоичные числа.

Алфавитом языка является множество: $\{0, 1, +, >\}$. Обозначим данный язык L. Рассмотрим следующей слово w, принадлежащее языку L:

$$w = 11^n + 11^n00^n > 11^{2n}0$$

Здесь $N_1 = 11^n$, $N_2 = 11^n00^n$, $N_3 = 11^{2n}0$. Пусть задана n - длина накачки. Возьмем описанное вышел слово, длина которого $> n$. Рассмотрим варианты разбиения этого слова на подслова u, x, v, y, w.

1. Если x или y содержат символы $+, >$, то при любой накачке полученное слово не будет принадлежать языку L. Поэтому стоит рассматривать лишь такие разбиения, в которых $x, y \in N_i,\ i = \overline{0, 2}$

2. Пусть xvy лежит в N_1. Тогда при отрицательной накачке, полученное слово не будет принадлежать языку L, т.к. изначально левая и правая часть отличались на единицу.

3. Пусть x лежит в N_1, а y в N_2. Тогда при отрицательной накачке, полученное слово не будет принадлежать языку L(аналогично пункту 2).

4. Пусть xvy лежит в N_2. Тогда при отрицательной накачке, полученное слово не будет принадлежать языку L(аналогично пункту 2).

5. Пусть xvy лежит в N_0. Тогда с помощью накачки N_0 можно сделать сколь угодно большим и неравенство станет неверным L.

6. Пусть $x \in N_2$, $y \in N_0$, тогда x имеет вид 0^k, а $y = 1^m$. Необходимо рассмотреть следующие случаи:

   6.1. $m = k$; Т.к. $x = 0^m$, а $y = 1^m$, то при положительной накачке, полученное слово не будет принадлежать языку L, т.к порядки левой и правой частей равны, но левая часть была накачана нулями, а правая единицами, и из-за этого число справа будет больше суммы чисел слева, поскольку в некотором разряде i в числе N_0 будет 1, а в сумме $N_1 + N_2$ в том же разряде будет стоять 0.

   6.2. $m > k$; При положительной накачке, число N_0 будет расти быстрее, чем N_2, поэтому существует длина накачки i, при которой порядок N_0 станет больше, чем $N_1 + N_2$, и слово не будет принадлежать языку.

   6.3. $m < k$; Если k - m $>$ 1, выполним отрицательную накачку и при сложении чисел слева, разряд суммы будет меньше чем разряд числа справа, т.к. при сложении число может увеличиться лишь на один разряд. Есл m + 1 = k, то при отрицательной накачке порядки чисел слева и справа будут равны, но число N_0 будет больше, т.к. при солжении N_1 и N_2 в получившемся числе на позиции i появится 0, причем $i > 0$. Т.о. неравенство не будет верным

Таким образом для любого n можно подобрать слово, которое не накачивается, значит данный в условии язык не является контекстно-свободным.

3 Задача 3

Определить, описывает ли данная грамматика регулярный язык

$$
\begin{array}{ll}
S \to TSTa & T \to a \\
S \to SS & T \to b \\
S \to bb & T \to TT
\end{array}
$$

3.1 Решение

Язык нетерминала T совпадает с языком задаваемым регулярными выражением $(a|b)^+$. Промежуточное представление для S можно записать следующим образом:

$$
\begin{aligned}
&S \to (a|b)^+ S (a|b)^+ a \\
&S \to SS \\
&S \to bb
\end{aligned}
$$

Гипотеза: Язык нетерминала S описывается регулярным выражением

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

4 Задача 4

Пусть $h(w)$ - слово, получающееся из w удвоением каждой буквы. Например, $h(aba^2) = a^2b^2a^4$. Запишем эти слова друг под другом так, чтобы первые буквы w и $h(w)$ образовали пару, вторые - следующую за ней, и т.д. Недостающуую длину в w дополним "решетками".

Исследовать язык пар слов $(w, h(w))$, поступающих на вход анализатора разбитыми таким образом на пары букв, т.е. поступающих параллельно (т.е. элементы входного алфавита - вектора $\begin{pmatrix} w_i \\ v_i \end{pmatrix}$, где $w_i, v_i \in \{a, b, \#\}$).

В нашем примере вход анализатора будет представлять собой следующую последовательность пар:

$$\begin{pmatrix} a \\ a \end{pmatrix} \begin{pmatrix} b \\ a \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \begin{pmatrix} \# \\ a \end{pmatrix} \begin{pmatrix} \# \\ a \end{pmatrix} \begin{pmatrix} \# \\ a \end{pmatrix} \begin{pmatrix} \# \\ a \end{pmatrix}$$