Теория формальных языков. Рубежный контроль №1

Вариант №11

Банников Арсений

Теоретическая информатика и компьютерные технологии
МГТУ им. Н.Э. Баумана
ноябрь 2023

Содержание

Задача 1

Условие

Язык всех ref-слов с единственной памятью, которые порождают языки в которых есть слово aa. Алфавит $\{a, b, |, (,), [_1,]_1, \&1\}$

Решение

Работа кипит...

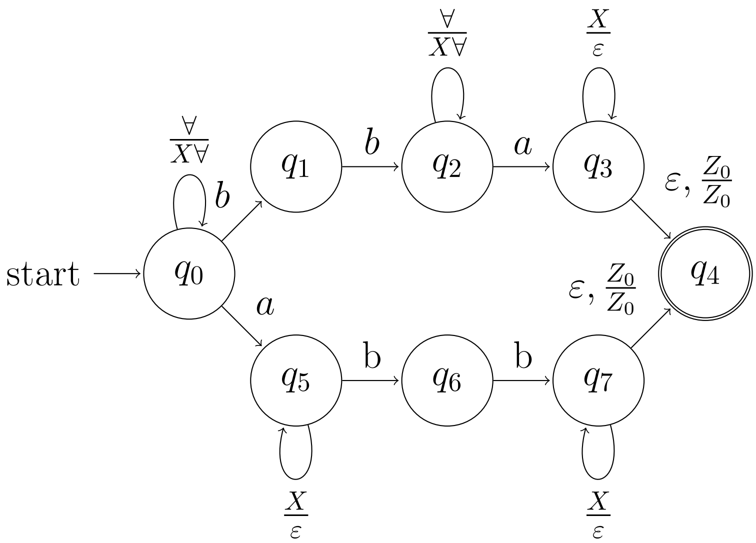
Задача 2

Условие

Язык $\left\{ w_1 bb w_2 \;\middle|\; w_1 w_2 = w_3 a w_4 \;\&\; |w_3| = |w_4| \right\}$

Решение

Исходный язык является недетерминированным КС. Соответствующий автомат:



Докажем недетерминированность. Рассмотрим слова $w_1 = b^n a b^{n+2}$ и $w_2 = b^n a b^{n+2} a b^{2n+1}$. У них общий префикс $x = b^n a b^{n+1}$, и суффиксы $y = b$, $z = bab^{2n+1}$ соответственно, начинающиеся с одной и той же буквы. Для упрощения доказательства выполним пересечение с $b^* a b^2 b^* (ab^*)?$. Если накачиваемый фрагмент в префиксе, то при любом разбиении x слово w_2 "накачкой"можно вывести из языка — теряется баланс букв справа и слева от a. При синхронной накачке вновь возникает дисбаланс числа букв, но уже в слове w_1.

Задача 3

Условие

$$
\begin{array}{ll}
[S] \to [Pred] & ; \\
[Pred] \to = \_[Expr]\_[Expr] & ; Expr_1.val == Expr_2.val \\
[Expr] \to [Op].[Expr] & ; Expr_0.val := (Op.fun)(Expr_1.val) \\
[Expr] \to [Op]\_[Val] & ; Expr.val := (Op.fun)(Val.val) \\
[Op] \to Sq & ; Op.fun := (\lambda x \to x^2) \\
[Op] \to Double & ; Op.fun := (\lambda x \to x \cdot 2) \\
[Val] \to 1 & ; Val.val := 1 \\
[Val] \to 1[Val] & ; Val_0.val := (Val_1.val) \cdot 2 + 1
\end{array}
$$

Решение

Опишем что происходит в данной грамматике. Очевидно, что:

1. Val — некоторое двоичное число вида $2^n - 1, n > 0$
2. $Expr$ — композиция функций Sq и $Double$, примененная к значению Val
3. $Pred$ — предикат проверяющий, что значение одного $Expr$ равно значению другого

Язык является не КС.

Рассмотрим слово $= \_Sq^n Double^n\_1\_Sq^n Double^n\_1$. Очевидно что мы не можем накачивать фрагменты содержащие _, так как их количество всегда равно четырем. Далее заметим, что мы не можем выбрать фрагменты только из одного $Expr$, так как если до накачивания значения $Expr.val$ были равны, то после накачивания они будут, очевидно, не равны. Также мы не можем выбрать фрагмент содержащий 1, так как отрицательная накачка выведет слово из языка. Следовательно накачивать нужно фрагменты из разных $Expr$, а это очевидно будут: в левом $Expr$ — $Double^{n_1}$, в правом $Expr$ — Sq^{n_3}. Далее проведем некоторую положительную накачку

i — получим слово $= \_Sq^n Double^{n_0+in_1}\_1\_Sq^{n_2+in_3} Double^n\_1$. Далее перенесем все $Double$ влево, для каждого $Expr$, получим слово $= \_Double^{(n_0+in_1)^{2^n}} Sq^n\_1\_Double^{n^{2^{n_2+in_3}}} Sq^{n_2+in_3}\_1$. Заметим, что при достаточно больших i количество $Double$ в $Expr$ справа больше чем в $Expr$ слева, что означает, что при разложении $Expr.val$ на простые множители мы получим для разных $Expr$ разное количество двоек, а значит $Expr.val$ перестанут быть равны и слово выйдет из языка. Таким образом мы никак не можем накачать данное слово. Из этого и следует, что язык не КС.