

Теория формальных языков. Рубежный контроль №2

Вариант №23

Киселев Кирилл

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

Содержание

1	Задача 1	2
1.1	Решение	2
2	Задача 2	3
2.1	Решение	3
3	Задача 3	4
3.1	Решение	4

1 Задача 1

Язык SRS $a \rightarrow bab, a^3 \rightarrow a^2, ba \rightarrow ac$ над множеством базисных слов $b^n a^n$

1.1 Решение

2 Задача 2

Язык $\{w \mid |w|_{ab} = |w|_{baa} \ \& \ w = w^R\}$. Алфавит $\{a, b\}$

2.1 Решение

Пусть $L_1 = \{w \mid |w|_{ab} = |w|_{baa}\}$, $L_2 = \{w \mid w = w^R\}$. Язык L_1 регулярный, а язык L_2 контекстно-свободный. Значит исходный язык L является КС, как пересечение КС и регулярного языков.

Докажем недетерминированность L . Пусть n - длина накачки. Тогда возьмем следующие слова: $w_1 = a^{2n}b^{2n}a^{2n}$, $w_2 = a^{2n}b^{2n}baabb^{2n}a^{2n}$. Пусть $x = a^{2n}b^{2n}$, $y = a^{2n}$, $z = baabb^{2n}a^{2n}$. Необходимо рассмотреть 2 случая:

1. Рассмотрим общий префикс x . Пусть $x = x_0x_1x_2x_3x_4$. Если $x_1 = a^k$ и $x_3 = a^p$, либо $x_1 = a^k$ и $x_3 = b^p$, то отрицательная накачка выводит оба слова из языка, т.к. полученные слова уже не будут являться палиндромами. Если $x_1 = b^k$ и $x_3 = b^p$, то отрицательная накачка в w_2 выводит слово из языка, т.к. полученное слово не будет являться палиндромом. Если $x_1 = a^{k_1}b^{k_2}$, либо $x_2 = a^{k_1}b^{k_2}$, то отрицательная накачка выводит оба слова из языка
2. Пусть $x = x_0x_1x_2$, $y = y_0y_1y_2$, $z = z_0z_1z_2$. Т.к по условию леммы $|x_1x_2| \leq n$, то $x_1 = b^{k_1}$ и $x_2 = b^{k_2}$, $k_1 + k_2 \leq n$, $k_1 > 0$. Также y_1 в любом случае равно a^{k_3} , тогда слово $x_0x_1^ix_2y_0y_1^iy_2$ при любом $i \neq 1$ не принадлежит L , т.к. не является палиндромом.

Следовательно, данный язык не является детерминированным КС языком.

3 Задача 3

Язык атрибутивной грамматики для регулярных:

$$\begin{aligned} [S] &\rightarrow [Regex] && ; \\ [Regex] &\rightarrow [Regex][Regex] && ; \quad Regex_1.val \neq \varepsilon, Regex_2.val \neq \varepsilon \\ &&& \quad Regex_0.val := Regex_1.val + +Regex_2.val \\ [Regex] &\rightarrow ([Regex]|[Regex]) && ; \quad Regex_1.val \neq \varepsilon \vee Regex_2.val \neq \varepsilon, \\ &&& \quad Regex_1.val \neq |, Regex_0.val := | \\ [Regex] &\rightarrow ([Regex])^* && ; \quad Regex_1.val \neq \varepsilon \\ &&& \quad Regex_1.val \neq *, Regex_0.val := * \\ [Regex] &\rightarrow \varepsilon && ; \quad Regex.val := \varepsilon \\ [Regex] &\rightarrow a && ; \quad Regex.val := a \\ [Regex] &\rightarrow b && ; \quad Regex.val := b \end{aligned}$$

3.1 Решение

Рассмотрим подвыражения, которые запрещены согласно ограничениям налагаемым условиями на атрибуты:

1. $(\varepsilon|\varepsilon)$
2. $((\cdot | \cdot) | \cdot)$
3. $(\varepsilon)^*$
4. $((\cdot)^*)^*$