

# Лабораторная работа №1

Чистяков Илья ИУ9-52Б

3 января 2026 г.

## Изначальная система переписывания

Рассматривается система переписывания строк (вариант 24), где алфавит

$$\Sigma = \{a, b, c\},$$

а множество правил задано следующим образом:

$$aa \rightarrow \varepsilon$$

$$bb \rightarrow cccc$$

$$cc \rightarrow acb$$

$$abc \rightarrow aabbcc$$

$$baabaac \rightarrow cbba$$

## Завершимость

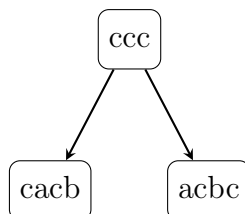
Рассмотрим количество применений каждого из правил для произвольной строки. Количество применений 4 и 5 правила конечно, так как система не может сама породить ни  $abc$  ни  $baabaac$ . Следовательно для их применения нужно использовать буквы из начальной строки, а их количество конечно. Количество применений первого и второго правила также конечно, так как строки  $aa$  и  $bb$  можно бесконечно порождать только четвертым и пятым правилом, количество применений которых, как мы выяснили, конечно. Отсюда количество применений третьего правила также конечно и вся система завершима.

## Конечность

Возьмём семейство слов  $w_n = (ac)^n$ . Каждое  $(w_n)$  принадлежит отдельному классу эквивалентности.

## Конфлюэнтность

Изначальная система не является конфлюэнтной ни локально, ни глобально. Это можно увидеть на примере строки  $ccc$ :



Дополним систему по алгоритму Кнута-Бендикса. Сначала введём лексикографический порядок в систему и переупорядочим правила:

$$\begin{aligned}aa &\rightarrow \varepsilon \\ cccc &\rightarrow bb \\ acb &\rightarrow cc \\ aabbcc &\rightarrow abc \\ baabaac &\rightarrow cbba\end{aligned}$$

Её можно упростить до

$$\begin{aligned}aa &\rightarrow \varepsilon \\ cccc &\rightarrow bb \\ acb &\rightarrow cc \\ bbcc &\rightarrow abc \\ cbba &\rightarrow bbc\end{aligned}$$

Пополняя по кнуту бендиксу получаем

$$\begin{aligned}aa &\rightarrow \varepsilon \\ acb &\rightarrow cc \\ acc &\rightarrow cb \\ cbb &\rightarrow bbb \\ bbbb &\rightarrow bb \\ abb &\rightarrow bb \\ ccc &\rightarrow bbb \\ ccb &\rightarrow bbb \\ bc &\rightarrow bb \\ bba &\rightarrow bb\end{aligned}$$

## Тестирование

### Фаззинг

В фаззинг тестировании, сгенерированное случайное слово я приводил сначала к нормальной форме в  $\tau$ . А после этого приводил эту нормальную форму и изначальное слово к нормальной форме в  $\tau'$ . Эквивалентность этих двух нормальных форм означает эквивалентность двух SRS. Несмотря на то, что цепочки переписывания в данном случае получаются детерминированными, основную свою функцию тестирование выполняет.

### Метаморфное тестирование

В этом тестировании придумал, как реализовать недетерминированные цепочки переписывания на рефале. В качестве инвариантов были выбраны две характеристики:

1. Четность количества  $b + c$  в слове
2. Длина слова