

# Лабораторная работа 1

Гуров Вячеслав ИУ9-52Б

## 1 Исходная система правил

Исходная система правил переписывания (SRS)  $\mathcal{T}$ :

1.  $apbc \rightarrow caqdbapbar$
2.  $paqd \rightarrow daqdbapbar$
3.  $ccpp \rightarrow adaqdqa$
4.  $dpr \rightarrow prp$

## 2 Доказательство завершимости системы переписывания

Докажем завершимость системы от противного, последовательно анализируя каждое правило.

### 2.1 Анализ правила 3 ( $ccpp \rightarrow adaqdqa$ )

Данное правило строго уменьшает количество символов  $c$  в строке на 2. Никакие другие правила не увеличивают количество  $c$ . Следовательно, правило 3 может быть применено лишь конечное число раз, определяемое начальным количеством  $c$ . Поэтому для построения гипотетической бесконечной цепочки это правило можно не учитывать.

### 2.2 Анализ правила 1 ( $apbc \rightarrow caqdbapbar$ ) и инвариант сдвига

Рассмотрим применение правила 1 к строке вида  $w_1apbcw_2 \rightarrow w_1caqdbapbarw_2$ .

**Случай 2.1:** Если в конце  $w_1$  имеется  $(apb)^n$ , то после серии применений правила 1 получим:

$$w'_1c(aqdbapbar)^{n+1}w_2$$

В процессе могут образовываться сочетания  $paqd$ , но их переписывание не создаёт новых возможностей для применения правил.

**Случай 2.2:** Если в начале  $w_2$  имеется  $(bc)^n$ , то после преобразований получим:

$$w_1(caqdbapb)^{n+1}apw_2$$

При этом каждая  $c$  сдвигается влево один раз. Если переписать все новые  $apbc$ , получившиеся на стыках, то каждая  $c$  сдвинется ещё раз, но новые сочетания для применения правила не появятся.

**Вывод:** Количество применений правила 1 ограничено количеством символов  $c$  в слове и длиной строки. Другие правила не могут существенно повлиять на этот процесс.

## 2.3 Анализ правил 2 ( $\text{paqd} \rightarrow \text{daqdbapbar}$ ) и 4 ( $\text{dpd} \rightarrow \text{pdp}$ )

Рассмотрим применение правила 2:  $w_1 \text{paqd} w_2 \rightarrow w_1 \text{daqdbapbar} w_2$ .

Первая  $\text{d}$  может быть использована:

- Для правила 4, если в конце  $w_1$  есть  $\text{dp}$
- Для правила 2, если в конце  $w_1$  есть  $\text{paq}$

Вторая  $\text{d}$  может быть использована только для правила 2, и то если первая  $\text{d}$  станет  $\text{p}$ .

**Вариант 1:**

$$\begin{aligned} & w'_1 \text{dp daqdbapbar} w_2 \\ & \rightarrow w'_1 \text{pdpaqdbapbar} w_2 \quad (\text{применяем правило 2}) \\ & \rightarrow w'_1 \text{pddaqaqdbapbarbarbar} w_2 \end{aligned}$$

Для дальнейшего применения правила 2 требуется  $\text{d}$  из  $w_1$ , что ограничено.

**Вариант 2:**

$$\begin{aligned} & w'_1 \text{paq daqdbapbar} w_2 \\ & \rightarrow w'_1 \text{daqdbapbaraqdbapbar} w_2 \quad (\text{применяем правило 2}) \\ & \rightarrow w'_1 \text{daqdbapbadaqdbapbarbarbar} w_2 \end{aligned}$$

Аналогично, количество “полезных”  $\text{d}$  уменьшается.

Правило 4 ( $\text{dpd} \rightarrow \text{pdp}$ ) не может применяться бесконечно, так как количество  $\text{d}$  в подходящих контекстах ограничено.

## 2.4 Заключение

Все правила системы либо уменьшают количество критических символов ( $\text{c}$ ), либо сдвигают их в направлении, в котором их движение не может быть бесконечным (правило 1), либо уменьшают потенциал для дальнейших преобразований (правила 2 и 4). Никакая их комбинация не может привести к бесконечной последовательности переписываний. Следовательно, система является завершимой.

## 3 Конечность классов эквивалентности по нормальной форме

Число классов эквивалентности бесконечно. Рассмотрим слова вида  $q^n$  для  $n \geq 1$ . Эти слова являются нормальными формами, так как ни одно правило не может быть применено к ним: они не содержат подстрок, которые являются левыми частями правил. Поскольку для разных  $n$  слова  $q^n$  различны и не могут быть преобразованы друг в друга, существует бесконечно много классов эквивалентности.

## 4 Локальная конфлюэнтность

Локальной конфлюэнтности нет. Рассмотрим слово  $\text{dpdpd}$ . Применим к нему правило 4 двумя способами:

- Применяем к первым трём символам:  $\text{dpd pd} \rightarrow \text{pdp pd}$  — получаем  $\text{pdpdpd}$ .
- Применяем к последним трём символам:  $\text{dp dpd} \rightarrow \text{dp pdp}$  — получаем  $\text{dppdp}$ .

Слова  $\text{pdpdpd}$  и  $\text{dppdp}$  являются нормальными формами, так как ни одно правило не может быть применено к ним. Поскольку они различны, система не является локально конфлюэнтной.

## 5 Пополняемость по Кнуту–Бендиксу

Система не является пополняемой по Кнуту–Бендиксу, так как процесс пополнения добавляет бесконечное число правил. Введём порядок: сначала по длине, затем по алфавиту ( $a < b < c < d < p < q$ ). Ориентируем правила:

1.  $caqdbarbar \rightarrow apbc$
2.  $daqdbarbar \rightarrow paqd$
3.  $adaqdqa \rightarrow ccpp$
4.  $pdp \rightarrow dpd$

Правило 4 создаёт бесконечную серию перекрытий с самим собой. Например, для слова  $pdpdp$  получаем критическую пару  $(dpddp, pddpd)$ . Эти слова не сводятся к общему терму, поэтому добавляем правило  $pddpd \rightarrow dpddp$ . Аналогично, для слова  $pddpdp$  получаем новую критическую пару и добавляем правило  $pdddpdp \rightarrow dpddpp$ . В общем случае возникают правила вида:

$$pd^{n+1}pd \rightarrow dpddp^n \quad \text{для } n \geq 1,$$

что приводит к бесконечному процессу пополнения.

## 6 Инварианты

Найдены следующие инварианты:

$$\begin{aligned} I_1(w) &= 2|w|_a + 3|w|_c - 2|w|_d - 2|w|_p, \\ I_2(w) &= 2|w|_a - 2|w|_b + |w|_c - 2|w|_q, \\ I_3(w) &= |w|_c \bmod 2, \end{aligned}$$

где  $|w|_x$  — количество символов  $x$  в слове  $w$ .

Первые два инварианта  $I_1$  и  $I_2$  являются линейно независимыми и образуют базис пространства всех целочисленных линейных инвариантов данной системы переписывания. Иными словами, любой другой линейный инвариант можно выразить как их целую линейную комбинацию.