Лабораторная работа 1

Гуров Вячеслав ИУ9-52Б

1 Исходная система правил

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

1. $apbc \to caqdbapbap$
2. $paqd \to daqdbapbap$
3. $ccpp \to adaqdqa$
4. $dpd \to pdp$

2 Доказательство завершимости системы переписывания

Докажем завершимость системы от противного, последовательно анализируя каждое правило.

2.1 Анализ правила 3 (`ccpp → 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 → caqdbapbap`) и инвариант сдвига

Рассмотрим применение правила 1 к строке вида $w_1\texttt{apbc}w_2 \to w_1\texttt{caqdbapbap}w_2$.

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

$$w_1'\texttt{c}(\texttt{aqdbapbap})^{n+1}w_2$$

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 $(\texttt{bc})^n$, то после преобразований получим:

$$w_1(\texttt{caqdbapb})^{n+1}\texttt{ap}w_2$$

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 (paqd → daqdbapbap) и 4 (dpd → pdp)

Рассмотрим применение правила 2: $w_1\texttt{paqd}w_2 \to w_1\texttt{daqdbapbap}w_2$.

Первая d может быть использована:

- Для правила 4, если в конце w_1 есть dp
- Для правила 2, если в конце w_1 есть paq

Вторая d может быть использована только для правила 2, и то если первая d станет p.

Вариант 1:

$$\begin{aligned} & w_1'\texttt{dp daqdbapbap}w_2 \\ \to\ & w_1'\texttt{pdpaqdbapbap}w_2 \quad \text{(применяем правило 2)} \\ \to\ & w_1'\texttt{pddaqdbapbapbapbap}w_2 \end{aligned}$$

Для дальнейшего применения правила 2 требуется d из w_1, что ограничено.

Вариант 2:

$$\begin{aligned} & w_1'\texttt{paq daqdbapbap}w_2 \\ \to\ & w_1'\texttt{daqdbapbapaqdbapbap}w_2 \quad \text{(применяем правило 2)} \\ \to\ & w_1'\texttt{daqdbapbadaqdbapbapbapbap}w_2 \end{aligned}$$

Аналогично, количество "полезных" d уменьшается.

Правило 4 (dpd → pdp) не может применяться бесконечно, так как количество d в подходящих контекстах ограничено.

2.4 Заключение

Все правила системы либо уменьшают количество критических символов (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 $dpdpd$. Применим к нему правило 4 двумя способами:

- Применяем к первым трём символам: $dpd\,pd \to pdp\,pd$ — получаем $pdppd$.
- Применяем к последним трём символам: $dp\,dpd \to dp\,pdp$ — получаем $dppdp$.

Слова $pdppd$ и $dppdp$ 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:

$$\begin{aligned}
&1. \quad caqdbapbap \to apbc \\
&2. \quad daqdbapbap \to paqd \\
&3. \quad adaqdqa \to ccpp \\
&4. \quad pdp \to dpd
\end{aligned}$$

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 $pddpd \to dpddp$. Аналогично, для слова $pddpdp$ получаем новую критическую пару и добавляем правило $pdddpd \to dpddpp$. В общем случае возникают правила вида:

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

что приводит к бесконечному процессу пополнения.

6 Инварианты

Найдены следующие инварианты:

$$\begin{aligned}
I_1(w) &= 2|w|_a + 3|w|_c - 2|w|_d - 2|w|_p, \\
I_2(w) &= 2|w|_a - 2|w|_b + |w|_c - 2|w|_q, \\
I_3(w) &= |w|_c \bmod 2, \\
I_4(w) &= \mathrm{Tr}(\Phi(w))
\end{aligned}$$

где $|w|_x$ — количество символов x в слове w, а $\Phi : \Sigma^* \to \mathrm{Mat}_3(\mathbb{Z})$ — гомоморфизм, определённый на символах алфавита как:

$$\Phi(a) = \Phi(c) = \mathbf{0}, \quad \Phi(b) = \Phi(q) = I, \quad \Phi(d) = S_1, \quad \Phi(p) = S_2,$$

где $\mathbf{0}$ — нулевая матрица, I — единичная матрица, а S_1, S_2 — матрицы транспозиций:

$$S_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad S_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Инвариант $I_4(w)$ вычисляется как след (сумма диагональных элементов) матрицы $\Phi(w)$.

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.