ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Апрель—июнь 2005. Серия 1. Том 12, № 2, 78–99

УДК 519.718

КОМБИНАТОРНАЯ СЛОЖНОСТЬ РАЦИОНАЛЬНЫХ ЯЗЫКОВ*)

А. М. Шур

Комбинаторной сложностью языка L называется функция, сопоставляющая числу n число различных слов длины n в языке L. Точному вычислению и оценкам комбинаторной сложности различных языков посвящены десятки работ. В статье приводится классификация по сложности произвольных рациональных языков и доказывается, что все эти классы остаются нетривиальными при переходе к подклассу рациональных языков — языкам с конечными антисловарями. Для таких языков известен алгоритм оценки сложности. Этот алгоритм экспоненциален по памяти, но находит практическое применение. Обсуждаются приближения произвольных факторных языков языками с конечными антисловарями, доказывается теорема о приближениях языка Туэ–Морса и формулируется общая гипотеза о сложности таких приближений.

1. Введение

Функция $C_L : \mathbb{N}_0 \to \mathbb{N}_0$ называется *комбинаторной сложностью* формального языка L, если $C_L(n)$ равна числу различных слов длины n в языке L (см., например, [3]). Теоретическая база для исследований по комбинаторной сложности заложена в [9]. Комбинаторная сложность активно изучалась для двух классов языков. Во-первых, это языки, состоящие из всех конечных подслов бесконечных слов (обзор результатов в [6], см. также [4], [5]). Во-вторых, это языки, порожденные отношением избегаемости (например, языки, состоящие из всех слов в фиксированном алфавите, избегающие заданное множество слов). Впервые оценки сложности языков такого типа приведены в [3], обзор результатов приведен в [6], некоторые результаты содержатся в [8]. В то же время классы таких языков весьма ограничены, а классификация по сложности представляет интерес для любых классов языков, задаваемых естественными условиями. Мы приводим такую классификацию для рациональных

*)Исследование выполнено при финансовой поддержке гранта ведущих научных школ РФ (проект НШ-2227.2003.1) и программы "Университеты России"(проект УР 04.01.437).

языков; сложность рационального языка определяется свойствами конечного автомата, распознающего этот язык. Далее мы доказываем, что все классы из приводимой классификации остаются непустыми для собственного подкласса класса рациональных языков. В разделе 5 вводится понятие рационального приближения языка, вычисляется сложность рациональных приближений языка Туэ–Морса и формулируется общая гипотеза о сложности рациональных приближений.

2. Предварительные сведения

2.1. Оценки сложности. Все языки, для которых рассматривалась комбинаторная сложность, являются *факторными* (замкнутыми относительно подслов). Это существенно ограничивало множество функций, которые могут являться комбинаторной сложностью некоторого языка. А именно, такая функция либо ограничена, либо возрастает в каждой точке (см., например, [6]). Эти ограничения давали возможность пользоваться стандартными в теории сложности обозначениями: записи $f = O(g)$, $f = \Omega(g)$, $f = \Theta(g)$ означают соответственно, что

существует константа $C{>}0$ такая, что $f(n) \leqslant C \cdot g(n)$ при любом $n{\geqslant}0$;

существует константа $C{>}0$ такая, что $f(n) \geqslant C \cdot g(n)$ при любом $n{\geqslant}0$;

существуют константы $C_1, C_2{>}0$ такие, что $C_1 \cdot g(n) \leqslant f(n) \leqslant C_2 \cdot g(n)$ при любом $n{\geqslant}0$.

Если отказаться от свойства факторности, то класс функций значительно расширяется, допуская, например, функции с произвольным числом экстремумов. Такие функции могут являться комбинаторными сложностями даже рациональных языков (множество длин слов в рациональном языке может быть любым финально периодическим множеством (см., например, [1]), а значит, может существовать бесконечное число таких пар натуральных чисел m, n, что $C_L(n) > 0$, $C_L(m) = 0$ и $m > n$). Для удобства сравнения роста таких функций мы дадим новые определения символов Θ и Ω так, чтобы их можно было применять к любым неограниченным функциям, а для монотонных функций их значение не изменилось. Положим $f = \Omega(g)$, если существуют константа $C{>}0$ и последовательность $\{i_n\}_{n\geqslant 0}$ такие, что $f(i_n) \geqslant C \cdot g(i_n)$ при любом $n{\geqslant}0$, и $f = \Theta(g)$, если $f = O(g)$ и $f = \Omega(g)$. Таким образом, для оценки роста функции берется ее самая быстрорастущая подпоследовательность. Как обычно, функцию f мы называем *экспоненциальной*, если $f = \Omega(\alpha^n)$ для некоторого $\alpha > 1$, и *полиномиальной*, если $f = O(n^k)$ для некоторого $k \geqslant 0$. Очевидно, что максимальная сложность языка над алфавитом Σ равна $|\Sigma|^n$.

Для экспоненциальных функций основной характеристикой является *индекс роста* $\alpha = \lim\limits_{n\to\infty} \sqrt[n]{f(n)}$. Заметим, что если экспоненциальная функция f имеет индекс роста α, то не обязательно $f = \Theta(\alpha^n)$; например, возможен случай, когда $f = \Theta(p(n)\alpha^n)$, где $p(n)$ — произвольный полином. Следующая теорема дает ответ на вопрос о существовании индексов роста.

Теорема 1 [9]. *Для любого факторного языка $L \in \Sigma^*$ существует предел $\lim\limits_{n\to\infty} \sqrt[n]{C_L(n)} = \alpha$, при этом*

1) $\alpha = 0 \iff$ *L конечен*;
2) $\alpha = |\Sigma| \iff L = \Sigma^*$;
3) $\alpha > 1 \iff$ *C_L экспоненциальна*;
4) $\alpha = 1$ *в остальных случаях.*

Замечание. Индексом роста естественно было бы называть предел $\lim\limits_{n\to\infty} C_L(n{+}1)/C_L(n)$. Если этот предел существует, то он действительно равен индексу роста. Однако такой предел существует для более узкого класса функций и аналогов теоремы 1 не существует.

2.2. Конечные автоматы. *Детерминированным конечным автоматом* (ДКА) называется упорядоченная пятерка $A = (\Sigma, Q, \delta, s, t)$, где Σ — конечный алфавит, Q — непустое конечное *множество состояний*, $\delta : Q \times \Sigma \to Q$ — частичная *функция переходов*, $s \in Q$ — начальное состояние, $T \subseteq Q$ — множество заключительных состояний. ДКА называется *полным*, если функция δ всюду определена.

С каждым ДКА можно связать ориентированный помеченный граф, называемый *графом переходов*. Граф переходов автомата $A = (\Sigma, Q, \delta, s, t)$ определяется следующим образом: Q есть множество вершин, Σ — множество меток, а каждому равенству вида $\delta(q, a) = q'$ соответствует дуга (q, q'), помеченная буквой a. В дальнейшем мы рассматриваем автомат как граф переходов, используя стандартные понятия маршрута, цикла и т. п.

Словам над алфавитом Σ соответствуют маршруты в автомате, помеченные этими словами. Автомат A *читает* слово $W \in \Sigma^*$, если в A существует маршрут из начальной вершины в заключительную, помеченный этим словом. Язык $L(A)$, *распознаваемый* автоматом A, состоит из всех слов, читаемых A. Согласно классической теореме Клини класс языков, распознаваемых конечными автоматами, совпадает с классом рациональных языков.

Два автомата называются *эквивалентными*, если они распознают один и тот же язык. Преобразование автомата называется *эквивалент-*

ным, если его результатом является автомат, эквивалентный исходному.

Вершина t автомата называется *особой*, если через t не проходит ни один маршрут из начальной вершины в заключительную. Удаление особой вершины является эквивалентным преобразованием. Таким образом, всякий рациональный язык распознается некоторым ДКА, не содержащим особых состояний (вершин).

3. Классификация сложности рациональных языков

Теорема 2. *Пусть A — детерминированный конечный автомат, не содержащий особых состояний, и $L = L(A)$. Тогда*

1) *если автомат A ацикличен, то язык L конечен*;

2) *если A содержит циклы, но никакие два цикла не имеют общих вершин, то $C_L(n) = \Theta(n^{k-1})$, где k — максимальное число различных циклов, содержащихся в маршруте из начальной вершины в заключительную*;

3) *если в A имеется вершина, принадлежащая не менее чем двум циклам, то функция $C_L(n)$ экспоненциальна.*

Для доказательства введем следующее вспомогательное понятие. Через $P_q(n)$ обозначим число различных маршрутов длины n, ведущих из начальной вершины автомата A в вершину q (т. е. число различных слов длины n, «чтение» которых автоматом A заканчивается в вершине q). Функцию P_q назовем *посещаемостью* вершины q. Очевидно, что

$$C_L(n) = \sum_{q \in T} P_q(n). \tag{1}$$

Доказательство теоремы 2. Первое утверждение теоремы очевидно. В самом деле, длины маршрутов в ацикличном графе ограничены некоторой константой. Этой же константой ограничены и длины слов в языке L, откуда следует конечность L.

Докажем третье утверждение теоремы. Нетрудно видеть, что комбинаторная сложность языка L экспоненциальна тогда и только тогда, когда в автомате A найдется вершина с экспоненциальной посещаемостью. Действительно, если функция $C_L(n)$, являющаяся суммой конечного числа слагаемых, экспоненциальна, то хотя бы одно из слагаемых также должно расти экспоненциально. Обратно, если в графе есть такая вершина q, что $P_q(n) \geqslant C \cdot \alpha^n$, то для любой вершины q', достижимой из q по маршруту длины m, справедливо соотношение $P_{q'}(n) \geqslant C \cdot \alpha^{n-m} = (C\alpha^{-m}) \cdot \alpha^n$. Так как в G_A нет особых вершин, то найдется заключительная вершина, достижимая из q и имеющая экспоненциальную посещаемость. Поэтому $C_L(n)$ экспоненциальна.

Покажем, что вершина q, принадлежащая двум или более циклам,

имеет экспоненциальную посещаемость. Существует маршрут из начальной вершины в q; следовательно, $P_q(i) \geqslant 1$ для некоторого $i \geqslant 0$. Пусть q принадлежит циклу длины l и циклу длины m. Тогда $P_q(i + lm) \geqslant 2 \cdot P_q(i)$ (к каждому слову длины i из $P_q(i)$ можно дописать как результат l-кратного обхода цикла длины m, так и результат m-кратного обхода цикла длины l. Получается слово длины $i+lm$). Тогда для значений n, равных $i + klm$, $P_q(n) = \Omega((\sqrt[lm]{2})^n)$. Тем самым вершина q имеет экспоненциальную посещаемость. Третье утверждение теоремы доказано.

Докажем второе утверждение теоремы. В соответствии с равенством (1), надо показать, что посещаемость любой вершины графа не превосходит $O(n^{k-1})$ и хотя бы одна вершина имеет посещаемость $\Theta(n^{k-1})$. Назовем *уровнем* вершины q (обозначается $\mathrm{lev}(q)$) максимальное число циклов, содержащихся в пути из начальной вершины в q. Докажем, что посещаемость вершины равна $\Theta(n^{m-1})$, где m — уровень вершины (начиная с некоторого n, для вершин уровня 0 посещаемость равна 0 — см. первое утверждение теоремы). Проведем индукцию по m. Пусть $m = 1$. Рассмотрим сначала вершину q, лежащую в цикле K первого уровня. Ей предшествует вершина q', лежащая в этом же цикле, и некоторое число вершин уровня 0. Тогда

$$P_{q'}(n) \leqslant P_q(n+1) \leqslant P_{q'}(n) + \sum_{\mathrm{lev}(r)=0} P_r(n),$$

т. е. начиная с некоторого n_0 $P_q(n+1) = P_{q'}(n)$. Это означает, что

$$P_q(n) \leqslant \max_{r \in K} P_r(n_0).$$

Следовательно, $P_q(n) = \Theta(1)$. Если же вершина q не лежит в цикле, то величина $P_q(n)$ равна конечной сумме значений вида $P_{q_i}(n - l_i)$, где вершина q_i принадлежит циклу K_i первого уровня либо является вершиной нулевого уровня. Следовательно, и в этом случае $P_q(n) = \Theta(1)$.

Шаг индукции. Сначала рассмотрим цикл K уровня $m+1$ и суммарную посещаемость всех его вершин (обозначим ее через P_K). Тогда разность $P_K(n+1) - P_K(n)$ равна сумме значений $P_r(n)$ по всем вершинам r, предшествующим вершинам цикла. Поскольку уровень каждой такой вершины не превосходит m и среди них имеется вершина уровня m, то по предположению индукции $P_K(n+1) - P_K(n) = \Theta(n^{m-1})$, т. е. $P_K(n) = \Theta(n^m)$. Отсюда следует, что хотя бы одна вершина в цикле имеет посещаемость $\Theta(n^m)$. Но если вершина q имеет посещаемость $\Theta(n^m)$, то посещаемость следующей за ней в цикле вершины q' удовлетворяет

условию $P_{q'}(n+1) \geqslant P_q(n)$. Следовательно, $P_{q'}(n) = \Omega(n^m)$. Это означает, что каждая вершина цикла имеет посещаемость $\Theta(n^m)$. Если же вершина q уровня $m{+}1$ не лежит в цикле, то значение $P_q(n)$ получается (как в случае, когда $m = 1$) сложением конечного числа величин, не превосходящих $\Theta(n^m)$. Шаг индукции доказан.

Поскольку, как уже упоминалось выше, комбинаторная сложность обычно рассматривалась для факторных языков, возникает вопрос: сохранятся ли все классы в классификации из теоремы 1, если ограничиться рассмотрением факторных рациональных языков? Оказывается, сохраняется даже в случае, если рассматривать собственный подкласс класса факторных рациональных языков — *языки с конечными антисловарями* (см. [7]), применяемые в сжатии данных.

4. Языки с конечными антисловарями

Через $F(W)$ будем обозначать множество всех подслов слова W, а через $F(L)$ — объединение множеств $F(W)$ по всем $W \in L$. Таким образом, язык L является факторным тогда и только тогда, когда $F(L) = L$. Слово W называется *запрещенным* для языка L, если $W \notin F(L)$. Любое множество запрещенных для L слов называется *антисловарем* для L. Антисловарь M называется *полным*, если $L = \Sigma^* \backslash \Sigma^* M \Sigma^*$. Минимальный по включению полный антисловарь будем называть *стандартным*. Язык M называется *антифакторным*, если $F(W) \cap M = \{W\}$ для любого $W \in M$. Сформулируем простейшие свойства стандартных антисловарей.

Свойство 1. *Стандартный антисловарь является антифакторным языком.*

Доказательство. Если M — стандартный антисловарь и $U \in M$ является подсловом слова $W \in M$, то всякое слово, содержащее W, содержит и U. Значит, при удалении W антисловарь останется полным. Противоречие с минимальностью M.

Свойство 2. *Для любого факторного языка существует единственный стандартный антисловарь.*

Доказательство. Дополнение любого факторного языка является идеалом свободного моноида Σ^*. Минимальное порождающее множество этого идеала является стандартным антисловарем.

Свойство 3. *Для любого антифакторного языка M существует единственный язык, для которого M является стандартным антисловарем.*

Доказательство. Рассмотрим идеал моноида Σ^*, порожденный язы-

ком M. Его дополнение будет искомым языком.

Стандартный антисловарь для языка L обозначим через $\mathcal{AD}(L)$. В дальнейшем мы рассматриваем только стандартные антисловари и прилагательное «стандартный» будем опускать. Язык, для которого M является стандартным антисловарем, будем обозначать через $\mathcal{L}(M)$.

Если язык $\mathcal{AD}(L)$ конечен, то L рационален (см. определение полного антисловаря). Обратное неверно: так, нетрудно проверить, что если $L = b^*ab^*$, то $\mathcal{AD}(L) = ab^*a$, т. е. антисловарь рационального языка может быть бесконечным.

В работе [9] показано, что бесконечный язык с конечным антисловарем имеет либо полиномиальную, либо экспоненциальную сложность (этот результат перекрывается теоремой 2). В этом параграфе мы покажем, что существуют языки с конечными антисловарями, имеющие сложность $\Theta(n^k)$ при любом целом $k \geqslant 0$.

ДКА, распознающий конечный антисловарь M, является деревом, с корнем в качестве начальной вершины и листьями в качестве заключительных вершин. В [7] приведен алгоритм построения по этому дереву детерминированного автомата без особых вершин, распознающего $\mathcal{L}(M)$. Воспроизведем этот алгоритм, попутно демонстрируя его работу на примере (см. рис. 1).

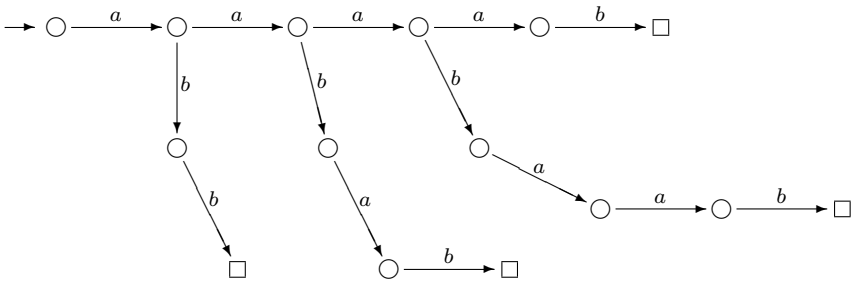
Возьмем ДКА, распознающий M (рис. 1 а)) и каждой вершине q поставим в соответствие слово W_q, которым помечен путь от корня до q. Доопределим функцию переходов δ по следующему правилу. Для каждой незаключительной вершины q и буквы c таких, что $\delta(q, c)$ не определено, в слове W_qc найдем длиннейший суффикс V такой, что $V = W_{q'}$ для некоторой вершины q', и положим $\delta(q, c) = q'$. Полученный автомат изображен на рис. 1 б). Теперь удалим все заключительные вершины, а все незаключительные вершины объявим заключительными. Результатом этого преобразования является искомый автомат (см. рис. 1 в)).

Теперь мы можем сформулировать результат о сложности языков с конечными антисловарями.

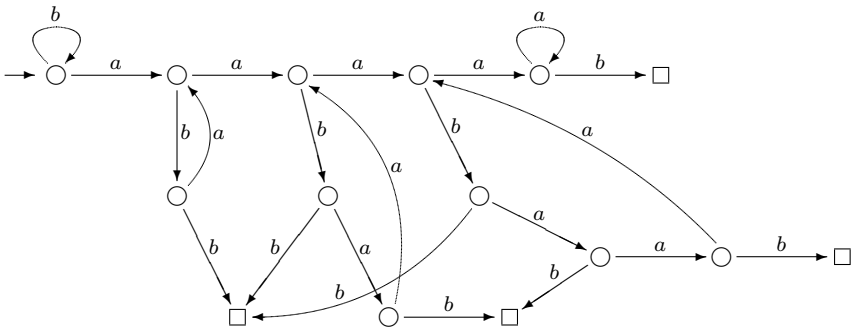
Теорема 3. *При любом целом $k \geqslant 0$ существует конечный антисловарь $M \in \{a, b\}^*$ такой, что комбинаторная сложность языка $\mathcal{L}(M)$ равна $\Theta(n^k)$.*

Доказательство. Случаи $k = 0$ и $k = 1$ рассмотрим отдельно. Для $k = 0$ возьмем антисловарь $M_0 = \{aa, bb\}$. Язык $\mathcal{L}(M_0)$ состоит из всех бинарных слов, в которых буквы чередуются. Следовательно, для любого $n \geqslant 1$ получаем $C_{\mathcal{L}(M_0)}(n) = 2$. Язык с константной сложностью найден. Для $k = 1$ возьмем антисловарь $M_1 = \{aab, bba\}$. Язык $\mathcal{L}(M_1)$

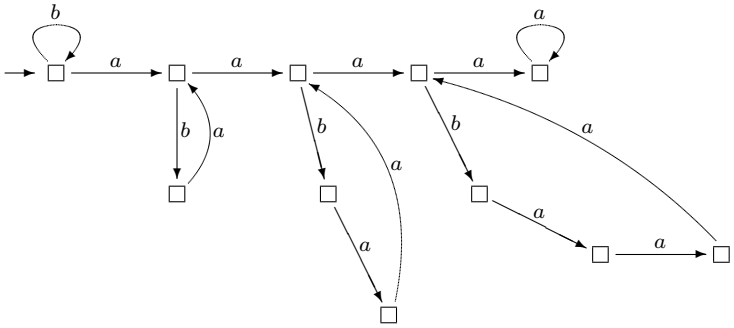
состоит из всех бинарных слов вида UV, где U состоит из чередующихся букв, а V — из одинаковых. Для каждого n существует $2n$ таких слов



а) *Автомат, распознающий антисловарь*



б) *Доопределение функции переходов*



в) *Автомат, распознающий язык с заданным антисловарем*

Рис. 1. Построение автомата, распознающего язык с антисловарем $M = \{aaaab, aaabaab, aabab, abb\}$ (заключительные вершины изображены квадратами)

длины n (именно столько имеется различных V, а каждое V однозначно определяет слово). Таким образом, мы построили язык с линейной сложностью.

Для каждого $k \geqslant 2$ рассмотрим антисловарь

$$M_k = \{a^k b,\ a^{k-1}ba^{k-2}b,\ a^{k-2}ba^{k-3}b,\ \ldots,\ abb\}$$

и докажем, что сложность языка $\mathcal{L}(M_k)$ равна $\Theta(n^k)$. Пример для $k = 4$ изображен на рис. 1. Рассмотрим общий случай. Возьмем автомат T_k, распознающий M_k, и применим к нему описанный выше алгоритм построения автомата, распознающего $\mathcal{L}(M_k)$. Доопределение функции переходов производится для внутренних вершин ровно с одним потомком. Это вершины, соответствующие словам λ, a^k, a^iba^j, где $1 \leqslant i < k$, $0 \leqslant j < i$. Нетрудно видеть, что дополнительная дуга с меткой b образует петлю при корневой вершине (соответствующей слову λ), а дополнительная дуга с меткой a — петлю при вершине, соответствующей слову a^k. Далее, если $j < i - 1$, то из вершины, соответствующей слову a^iba^j, нужно провести дугу, помеченную буквой b. Поскольку слово $a^{j+1}ba^jb$ соответствует листу дерева и является суффиксом a^iba^jb, эта дуга будет направлена в этот лист. Такие дуги будут удалены вместе с листьями на следующем шаге алгоритма. Если же $j = i - 1$, то из вершины, соответствующей слову a^iba^j, нужно провести дугу, помеченную a; длиннейший суффикс слова a^iba^i, соответствующий вершине дерева, равен a^i, в нее направлена требуемая дуга. Таким образом, после удаления терминальных вершин исходного автомата полученный автомат будет представлять собой последовательность непересекающихся циклов, «подвешенных» к каждой вершине пути, помеченного словом a^k (см. рис. 1 в)). Число таких циклов равно $k + 1$, поэтому из теоремы 2 следует искомая оценка комбинаторной сложности языка $\mathcal{L}(M_k)$. Теорема 3 доказана.

До сих пор неизвестно, насколько еще можно сузить класс факторных рациональных языков так, чтобы в этом классе выполнялась теорема 3. Приведем один из естественных вариантов сужения.

Вопрос. Верно ли, что для любого целого $k \geqslant 0$ найдется *симметричный* (т. е. замкнутый относительно переименования букв) конечный антисловарь $M \in \{a, b\}^*$ такой, что комбинаторная сложность языка $\mathcal{L}(M)$ равна $\Theta(n^k)$?

Симметричные антисловари имеют, в частности, все факторные языки, заданные отношением избегаемости (напомним, что слово U избегает слово V, если $F(U)$ не содержит гомоморфных образов V). Заметим, что

существуют языки константной и линейной сложности с симметричными антисловарями; примерами таких языков являются языки $\mathcal{L}(M_0)$ и $\mathcal{L}(M_1)$ из доказательства теоремы 3.

5. Рациональные приближения факторных языков

Оценка комбинаторной сложности факторного языка может представлять собой очень сложную задачу. В то же время для языков с конечными антисловарями техника оценки сложности в случае экспоненциальной сложности разработана Кобаяси [9]. Им доказана теорема о том, что индекс роста языка с конечным антисловарем равен доминантному собственному числу некоторой матрицы, которая эффективно строится по антисловарю. К сожалению, оценка этого собственного числа является трудоемкой вычислительной задачей, поскольку размер матрицы экспоненциально зависит от максимальной длины слова в антисловаре. Задача улучшения техники оценивания индекса роста актуальна, но находится за пределами данной статьи.

Естественной представляется идея «приближения» произвольного факторного языка языками с конечными антисловарями для получения оценки сложности исходного языка. Эта идея впервые была воплощена, по-видимому, в [3] для получения верхней оценки сложности языка бинарных бескубных слов. Кратко опишем общую схему. Пусть L — факторный язык и $M = \mathcal{AD}(L)$. Через M_k обозначим множество всех слов из M длины не более k. Тогда

$$L \subseteq \ldots \subseteq \mathcal{L}(M_k) \subseteq \ldots \subseteq \mathcal{L}(M_1),$$

причем для каждого n существует такое k, что $L \cap \Sigma^n = \mathcal{L}(M_k) \cap \Sigma^n$. Тогда

$$C_L(n) \leqslant \ldots \leqslant C_{\mathcal{L}(M_k)}(n) \leqslant \ldots \leqslant C_{\mathcal{L}(M_1)}(n)$$

для любого n.

Таким образом, последовательность функций $C_{\mathcal{L}(M_K)}(n)$ сходится к функции $C_L(n)$. Точность приближения увеличивается с ростом k. (Как уже упоминалось выше, сложность вычислений зависит от k экспоненциально).

Языки $\mathcal{L}(M_k)$ будем называть *рациональными приближениями* языка L. В этом разделе вычислена сложность всех рациональных приближений языка Туэ–Морса и сформулирована общая гипотеза о сложности рациональных приближений.

5.1. Антисловарь языка Туэ–Морса. Слова Туэ–Морса являются, по-видимому, самым популярным объектом исследования в комбинаторике слов. Напомним определения. Пусть φ — морфизм, определенный

на $\{a, b\}^*$ по правилу

$$\varphi(a) = ab, \qquad \varphi(b) = ba.$$

Слова $U_n = \varphi^n(a)$, $V_n = \varphi^n(b)$ называются словами Туэ–Морса; в дальнейшем мы называем их также *n-блоками*. Языком Туэ–Морса называется множество всех подслов слов Туэ–Морса. Следующее предложение описывает антисловарь языка Туэ–Морса. Здесь и далее мы используем следующие обозначения, связанные с произвольным словом $W \in \{a, b\}^*$: $Pr(W, k)$ обозначает k-буквенный префикс (начальный отрезок) слова W, $l(W)$ и $t(W)$ — соответственно первую и последнюю буквы в W, а W' — слово, полученное из W заменой всех букв a на b и наоборот.

Утверждение 1. *Пусть*

$$S_k = t(\varphi^k(a))\varphi^k(aba)l(\varphi^k(a)), \quad R_k = t(\varphi^k(a))\varphi^k(bab)l(\varphi^k(a)),$$

Тогда

$$\mathcal{AD}(TM) = \{aaa, bbb\} \cup \{S_k, S'_k, R_k, R'_k \mid k \geqslant 0\}.$$

В [10] было дано похожее описание множества всех слов, избегаемых языком TM. Доказательства утверждения из [10] и утверждения 1 опираются на алгоритм обработки слов из [11]. Приведем этот алгоритм и его основное свойство.

Алгоритм А [11].

Данные: слово W. Результат работы: слово W_A.

1. Положить $W_A = W$.
2. Если $W_A \in \{ab, ba\}$, остановиться.
3. Добавить не более одной буквы в конец слова W_A, затем не более одной буквы в начало слова W_A так, чтобы получилось слово $U \in \varphi(\{a, b\}^*)$. Если это невозможно, остановиться.
4. Положить $W_A = \varphi^{-1}(U)$ и вернуться на шаг 2.

Лемма 1 [11]. *Слово W принадлежит языку TM тогда и только тогда, когда $W_A \in \{ab, ba\}$.*

Доказательство утверждения 1. Пусть

$$M = \{aaa, bbb\} \cup \{S_k, S'_k, R_k, R'_k \mid k \geqslant 0\}.$$

Для доказательства равенства $M = \mathcal{AD}(TM)$ требуется проверить выполнение следующих четырех условий:

(1) M состоит из запрещенных слов;
(2) $F(M)\backslash M \in TM$;
(3) Любое запрещенное слово содержит подслово из M;
(4) M является антифакторным языком.

Убедимся в справедливости этих условий.

(1) Если $W = aaa$ или $W = aabaa$, то алгоритм A не может сделать ни одного шага, т. е. $W_A = aaa$ (соответственно $W_A = aabaa$). Если $W = ababa$, то $W_A = aaa$. Далее заметим, что при $k > 0$ один шаг алгоритма A преобразует слово S_k в S_{k-1}, а слово R_k в R_{k-1}; следовательно, в этих случаях имеем $W_A = aaa$ или $W_A = aabaa$. Симметричные рассмотрения справедливы для слов bbb, S'_k, R'_k. По лемме 1 все слова из M являются запрещенными.

(2) Рассмотрим максимальные собственные подслова слов из M. Слово aa очевидно принадлежит языку TM. Слово $t(\varphi^k(a))\varphi^k(aba)$ за k шагов преобразуется алгоритмом A в слово $aaba$ и далее последовательно в bab и ba, т. е. принадлежит языку TM по лемме 1. Остальные максимальные подслова рассматриваются аналогично.

(3) Если $W \notin TM$, то к слову W_A нельзя применить шаг алгоритма A, т. е. W_A нельзя представить в виде $cQ_1 \dots Q_p d$, где каждое из Q_i равно ab или ba, а $|c| \leqslant 1, |d| \leqslant 1$. Следовательно, существует такая пара чисел (i, j), что $W_A(2i{-}1) = W_A(2i)$ и $W_A(2j) = W_A(2j{+}1)$. Среди всех пар (i, j) с этим свойством выберем такую пару, в которой позиции $2i{-}1$ и $2j$ находятся ближе всего друг к другу. Тогда при $j = i$ и при $j = i{-}1$ слово W_A содержит подряд три одинаковые буквы, при $j = i{+}1$ и при $j = i{-}2$ слово W_A содержит подслово $aabaa$ или $bbabb$, а при $j > i{+}1$ и при $j < i{-}2$ слово W_A содержит подслово $ababa$ или $babab$. В первом случае либо $W = W_A$ (т. е. W содержит aaa (bbb)), либо $W \neq W_A$ и W содержит некоторое R_k (R'_k). Во втором случае W содержит некоторое S_k (S'_k), а в третьем случае — некоторое R_k (R'_k). Следовательно, каждое запрещенное слово содержит подслово из M.

(4) Слово из M не может быть собственным подсловом никакого слова из M по условиям (1) и (2). Непосредственно проверяется, что никакие два элемента из M не совпадают. Таким образом, M — антифакторный язык. Утверждение 1 доказано.

5.2. Порядки точек относительно слов Туэ–Морса. Для вычисления сложности произвольного рационального приближения языка Туэ–Морса нужно уметь определять структуру соответствующего автомата. Для этого мы адаптируем разработанную в [2] технику анализа бесконечных слов Туэ–Морса для конечных слов. Напомним, что *Z-слово*

есть бесконечная последовательность символов, индексированная всеми целыми числами. Произвольное Z-слово можно рассматривать как числовую прямую, на которой «расставлены» буквы: i-я буква занимает отрезок $[i-1, i]$. Z-слово Туэ–Морса над алфавитом $\{a, b\}$ есть Z-слово, все конечные подслова которого принадлежат языку Туэ–Морса TM. Приведем необходимые понятия и факты из [2].

Любое Z-слово Туэ–Морса единственным образом представимо в виде произведения n-блоков. Это представление называется *n-разбиением*. При этом $(n-1)$-разбиение является «измельчением» n-разбиения. Целочисленная точка x называется *граничной* для n-разбиения Z-слова W, если соседние с ней буквы слова W принадлежат разным блокам. *Порядок* точки x относительно W (обозначается $\deg_W(x)$) определяется как супремум множества чисел n таких, что x является граничной для n-разбиения W. Если W фиксировано, то индекс в обозначении порядка будем опускать. Порядки обладают следующими свойствами.

Лемма 2 [2]. *Для всякого Z-слова Туэ–Морса и всякой точки $x \in \mathbb{Z}$*
1) *если* $\deg(x) = k$ *и* $|x - y| < 2^k$, *то* $\deg(y) < k$;
2) *если* $\deg(x) = k$, *то* $\deg(x + m \cdot 2^k) \geqslant k$ *для любого* $m \in \mathbb{Z}$;
3) *если непосредственно справа (слева) от x расположены два k-блока, то* $\deg(x) \geqslant k$; *если эти k-блоки равны, то* $\deg(x) = k$;
4) *из двух соседних точек порядка не менее k одна имеет порядок k, а другая — порядок больше k.*

Приведенная лемма показывает, что определение порядков почти всех точек Z-слова Туэ–Морса может быть произведено на основании изучения «локальной» структуры слова в окрестности этой точки. Исключение составляют лишь точки порядка ∞. Любое конечное слово из TM содержится во *всех* Z-словах Туэ–Морса. Тем не менее можно указать порядок тех точек этого слова, для которых он не превосходит некоторого k (эти порядки не зависят от того, какое вхождение и в какое Z-слово мы рассматриваем); для оставшихся точек порядок не менее $k+1$ (точное значение порядка зависит от конкретного вхождения в конкретное Z-слово).

Итак, слова из TM можно представлять в виде отрезков с целочисленными концами на числовой прямой и вычислять или оценивать порядки точек на основании леммы 2. При доказательстве следующей теоремы мы воспользуемся тем, что любое слово из рационального приближения языка TM «локально» устроено как слово из TM.

5.3. Сложность рациональных приближений языка Туэ–Морса. В антисловаре языка Туэ–Морса присутствуют слова длины 3 и слова

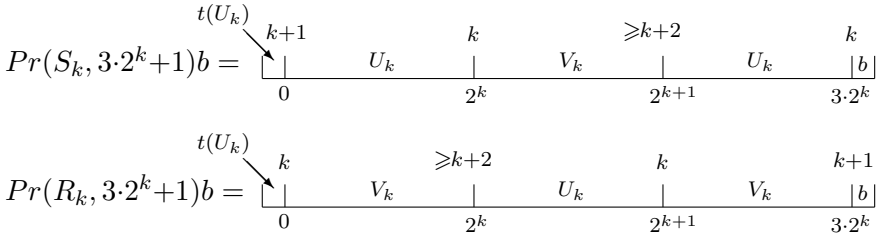
длины $3{\cdot}2^k{+}2$ для всех $k \geqslant 0$. Ниже через M_k обозначается подмножество антисловаря $M = \mathcal{AD}(TM)$, состоящее из всех слов длины не более $3{\cdot}2^k{+}2$. Для единообразия обозначений положим $M_{-1} = \{aaa, bbb\}$.

Теорема 4. *Рациональное приближение $\mathcal{L}(M_k)$ языка Туэ–Морса имеет экспоненциальную сложность с индексом роста* $\alpha = \sqrt[2^{k+1}]{\frac{1+\sqrt{5}}{2}}$.

Доказательство. Обозначим через T_k и A_k автоматы, распознающие языки M_k и $\mathcal{L}(M_k)$ соответственно. Для доказательства нужно установить структуру автомата A_k. Автомат A_k строится по дереву T_k при помощи алгоритма, приведенного в предыдущем разделе. Сначала выясним, куда ориентированы дополнительные дуги от внутренних вершин дерева. Для удобства каждую вершину q дерева T_k будем отождествлять со словом W_q, которым помечен путь из корня в эту вершину. Тогда (с точностью до переименования букв) любая внутренняя вершина имеет вид $Pr(S_n, l)$ или $Pr(R_n, l)$ при некоторых $0 \leqslant n \leqslant k$ и $1 \leqslant l \leqslant 3 \cdot 2^n + 1$.

Каждую ветку дерева T_k рассмотрим как отрезок числовой прямой с размещенными буквами. Вершины дерева соответствуют целочисленным точкам отрезков; при этом корню соответствует число -1 (дерево T_1 изображено на рис. 2).

По утверждению 1 все собственные префиксы листьев дерева T_k принадлежат языку TM. Так как любое слово из TM можно продолжить вправо, оставаясь в TM, то слово, полученное из листа заменой последней буквы, также принадлежит языку TM. Ниже на основании леммы 2 вычислены порядки «ключевых» точек относительно таких слов:



Очевидно, что порядки точек останутся теми же самыми при замене $Pr(S_k, 3{\cdot}2^k{+}1)b$ на $Pr(S'_k, 3{\cdot}2^k{+}1)a$ и $Pr(R_k, 3{\cdot}2^k{+}1)b$ на $Pr(R'_k, 3{\cdot}2^k{+}1)a$. Так как из леммы 2 следует, что середина отрезка, концы которого суть соседние точки порядка не менее n, имеет порядок $n{-}1$, то порядки всех непомеченных на рисунке точек вычисляются однозначно k-кратным применением этого замечания.

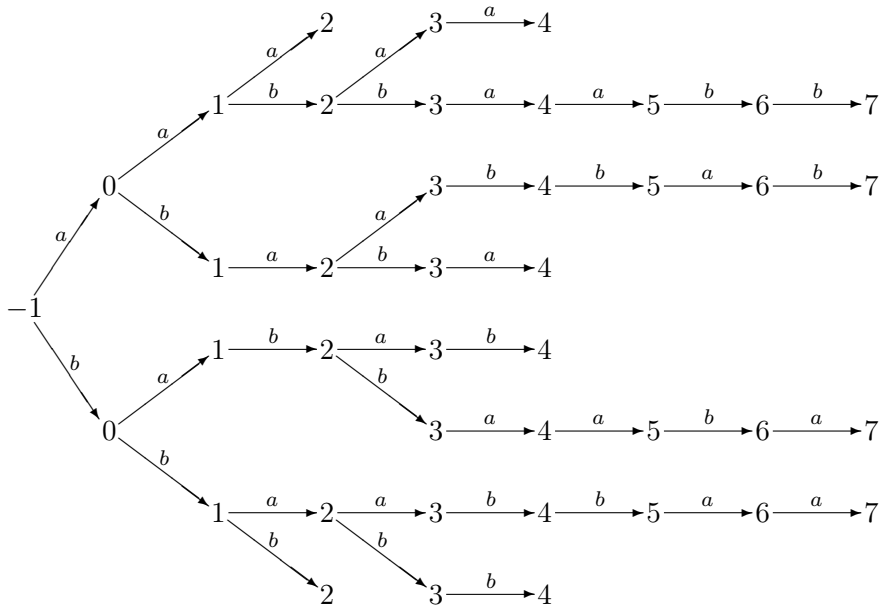


Рис. 2. Дерево T_1, распознающее антисловарь языка Туэ–Морса, «усеченный» до длины $8 = 3 \cdot 2^1 + 2$

Поскольку вершинам сопоставлены целочисленные точки, мы иногда будем говорить «порядок вершины» вместо «порядок точки». *Веткой типа* n будем называть ветку, помеченную одним из слов S_n, S'_n, R_n, R'_n. Определим, куда направлена дополнительная дуга, исходящая из вершины W порядка i на данной ветке типа n в зависимости от i.

Пусть $i < n$. В этом случае порядок равен i только для точек вида $m2^i$ при нечетном m.

Сначала рассмотрим случай $m \geqslant 5$. Слева от точки $m{\cdot}2^i$ в слове S_n или R_n находится m i-блоков. Рассмотрим последние пять из них. Так как порядок точки $(m{-}1){\cdot}2^i$ не менее $i{+}1$ по лемме 2, то $(m{-}1)$-й и $(m{-}2)$-й, а также $(m{-}3)$-й и $(m{-}4)$-й блоки образуют $(i{+}1)$-блоки, т. е. различны. Выбранные пять блоков составляют собственное подслово в слове из антисловаря, т. е. по утверждению 1 не могут образовывать не встречающийся в словах из TM фрагмент вида $UVUVU$. Следовательно, совпадают либо m-й и $(m{-}1)$-й, либо $(m{-}3)$-й и $(m{-}2)$-й блоки. Таким образом, мы получаем, что слово W заканчивается либо на U_iU_i (V_iV_i), либо на $U_iU_iV_iU_i$ ($V_iV_iU_iV_i$). В первом случае дополнительная дуга из W ведет в лист R_{i-1} или R'_{i-1} (если $i = 0$, то в лист aaa или bbb), во втором случае — в лист S_{i-1} или S'_{i-1}.

Пусть теперь $m = 3$. В словах S_n и R_n справа от точки 0 находится

$(n{+}1)$-блок; он содержит не менее четырех i-блоков (поскольку $i < n$), второй и третий i-блоки в нем совпадают. Таким образом, в этом случае слово W заканчивается на U_iU_i (V_iV_i), т. е. дополнительная дуга снова ведет в лист.

Осталось рассмотреть случай $m = 1$. При $i = 0$ имеем $W = ab$ или $W = ba$ (см. рис. 2). Дополнительные дуги направлены в вершины bb и aa соответственно. Докажем, что при $i > 0$ вершина W в дереве является полной (вершина *полная,* если у нее имеется два потомка). Очевидны равенства $t(U_i) = t(V_{i-1})$, $t(V_i) = t(U_{i-1})$ и $t(U_i) = (t(U_{i-1}))'$. Поэтому слова aU_i и bU_i являются длиннейшими общими префиксами пар слов S_i и R'_{i-1}, S_{i-1} и R'_i (какое слово является префиксом какой пары, зависит от четности i). Аналогично, aV_i и bV_i являются длиннейшими общими префиксами пар слов S'_i и R_{i-1}, S'_{i-1} и R_i.

Пусть теперь $i \geqslant n$. Такой порядок имеют точки 0, 2^n, 2^{n+1} и $3{\cdot}2^n$. Вершины, соответствующие точке 0, очевидно полны. Если вершина соответствует либо точке 2^{n+1} при $n < k$, либо точке 2^n, то можно воспользоваться доказательством полноты, приведенном в предыдущем абзаце. Поэтому осталось исследовать точки $3{\cdot}2^n$, а также точки 2^{n+1} при $n = k$.

Согласно алгоритму построения автомата A_k дополнительная дуга из вершины W по символу c ведет в вершину V такую, что V — длиннейший суффикс слова Wc, присутствующий в дереве. Поиск слова V будем осуществлять при помощи сравнения порядков точек относительно Wc и относительно слов $Pr(S_n, 3{\cdot}2^n{+}1)b$, $Pr(S'_n, 3{\cdot}2^n{+}1)a$, $Pr(R_n, 3{\cdot}2^n{+}1)b$, $Pr(R'_n, 3{\cdot}2^n{+}1)a$.

Пусть $W = Pr(S_n, 3{\cdot}2^n{+}1)$. Если на ветке типа i в дереве присутствует суффикс слова Wb длины не менее $2^{n+1}{+}2$, то $i \geqslant n$. Точка 0 на ветке типа i имеет порядок не менее i. Таким образом, после первой буквы в искомом суффиксе слова Wb находится точка порядка не менее i. Поэтому $n = i$ и длина суффикса равна $2^{n+1}{+}2$. Рассматривая префиксы длины $2^{n+1}{+}2$ веток типа n, замечаем, что префикс R_n равен искомому суффиксу. Следовательно, для каждого n, $0 \leqslant n \leqslant k$, соответствующий переход в автомате A_k определен как

$$\delta(Pr(S_n, 3{\cdot}2^n{+}1)\,,\, b) = Pr(R_n, 2^{n+1}{+}2). \tag{2}$$

Пусть $W = Pr(R_n, 3{\cdot}2^n{+}1)$, $n < k$. Как и в предыдущем случае получаем, что максимально возможная длина суффикса слова Wb в дереве не менее $2^{n+1}{+}2$. После первой буквы в данном суффиксе слова Wb находится точка порядка $n{+}2$. Следовательно, точка 0 на нужной ветке типа i также должна иметь порядок не менее $n{+}2$. Такая ветка типа $(n{+}1)$

существует, а соответствующие переходы в автомате A_k записываются для всех n как

$$\delta(Pr(R_n, 3{\cdot}2^n{+}1)\,,\, b) = Pr(S'_{n+1}, 2^{n+1}{+}2). \tag{3}$$

Отдельно разберем случай $W = Pr(R_k, 3{\cdot}2^k{+}1)$. Поскольку условие $\deg(0) \geqslant k{+}2$ не выполнимо на ветках дерева T_k, длиннейший суффикс слова Wb, присутствующий в дереве, короче чем $2^{k+1}{+}2$. Пусть длина этого суффикса не меньше $2^k{+}2$. Тогда он расположен на ветке типа $k{-}1$ или k. Точка 0 на такой ветке имеет порядок не менее $k{-}1$. После первой буквы в требуемом суффиксе слова Wb находится точка порядка не менее $k{-}1$. Если эта точка имеет порядок $k{-}1$ (середина второго k-блока в R_k), то соответствующая ветка помечена словом R_{k-1} или R'_{k-1}. Однако в этих словах точка 2^{k-1} имеет порядок не менее $k{+}1$, а соответствующая точка в Wb — порядок k, что невозможно. Следовательно, после первой буквы в требуемом суффиксе слова Wb находится точка порядка не менее k, т. е. длина суффикса равна $2^k{+}2$. Ветка, содержащая суффикс Wb такой длины, действительно существует:

$$\delta(Pr(R_k, 3{\cdot}2^k{+}1)\,,\, b) = Pr(S'_{k-1}, 2^k{+}2). \tag{4}$$

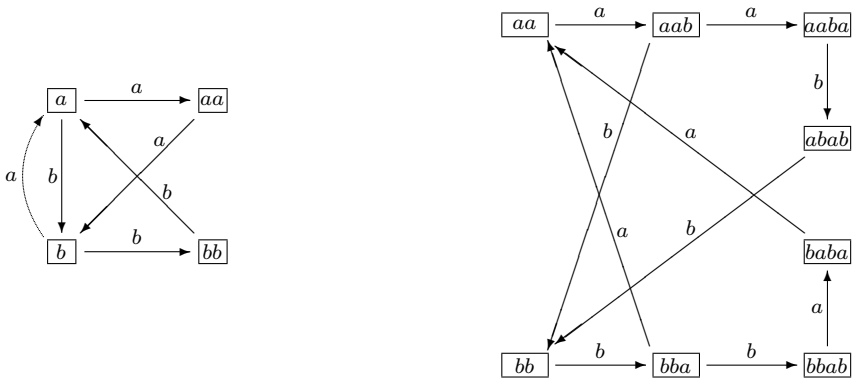
Точно такие же рассуждения проводятся для оставшихся случаев $W = Pr(S_k, 2^{k+1}{+}1)$ и $W = Pr(R_k, 2^{k+1}{+}1)$. В результате получаем

$$\delta(Pr(S_k, 2^{k+1}{+}1)\,,\, b) = Pr(S'_{k-1}, 2^k{+}2), \tag{5}$$

$$\delta(Pr(R_k, 2^{k+1}{+}1)\,,\, b) = Pr(S_{k-1}, 2^k{+}2). \tag{6}$$

Дополнительные дуги, начинающиеся на ветках, помеченных словами S'_n, R'_n, симметричны построенным. Первый этап доказательства завершен.

Согласно теореме 1 сложность языка $\mathcal{L}(M_k)$ зависит от взаимного расположения циклов в автомате A_k. На втором этапе доказательства мы покажем, что посещаемость любой вершины автомата A_k, не лежащей ни в одном цикле, с некоторого момента равна 0, и точно установим вид подграфа в A_k, индуцированного всеми остальными вершинами. Этот подграф будем обозначать через CG_{A_k}. Заметим, что необходимым условием принадлежности вершины циклу является наличие дополнительной дуги, ведущей в эту вершину или в ее префикс. На рис. 3 изображены графы $CG_{A_{-1}}$ и CG_{A_0}.



а) б)

Рис. 3. Все циклы автоматов а) A_{-1} и б) A_0

Пусть теперь $k \geqslant 1$. Пользуясь равенствами (2)–(6), нетрудно показать, что любая вершина дерева T_{k-2} не лежит в циклах автомата A_k. Сначала заметим, что вершина, в которую направлена дополнительная дуга, принадлежит ровно одной ветке дерева T_k, а длина слова, соответствующего этой вершине, не менее 4. Это, в частности, означает, что вершины ab и ba не принадлежат циклам, и дополнительные дуги из ab в bb и из ba в aa можно исключить из рассмотрения. Далее, никакая дополнительная дуга не направлена из вершины в ее префикс, а любой путь, содержащий две дополнительных дуги, соединяет вершину, принадлежащую дереву $T_i \backslash T_{i-1}$, с вершиной, принадлежащей $T_{i+1} \backslash T_i$ для всех $i < k$. В то же время, дополнительные дуги, ведущие из вершин $T_{i+1} \backslash T_i$ в вершины из T_i, существуют только при $i = k$. Тем самым, в циклах могут находиться только те вершины автомата, которые принадлежат дереву $T_k \backslash T_{k-2}$. Множество таких вершин образует в дереве T_k восемь изолированных цепей, соответствующих заключительным отрезкам веток типа $k-1$ и типа k. Эти цепи связаны дополнительными дугами (2)–(6), как показано на рис. 4.

Из рис. 4 видно, что в автомате A_k не существует пути из вершины, принадлежащей циклу, в вершину, циклам не принадлежащую. Таким образом, длины маршрутов в A_k, заканчивающихся в вершинах, не принадлежащих циклам, ограничены в совокупности. Следовательно, посещаемости всех таких вершин равны нулю, начиная с некоторой длины. Таким образом, на оценку комбинаторной сложности языка $\mathcal{L}(M_k)$ влияет только структура подграфа CG_{A_k}.

Из рис. 4 видно, что графу CG_{A_k} полностью принадлежат изобра-

женные фрагменты шести веток дерева T_k, а также, начиная с длины префикса, равной $2^{k+1}+2$, фрагменты веток с листьями R_k и R'_k. Следовательно, граф CG_{A_k} является циклом длины 2^{k+3} с двумя хордами, которые начинаются в максимально удаленных друг от друга вершинах данного цикла и стягивают непересекающиеся дуги длины $2^{k+1}+1$ (рис. 5). На оставшемся этапе доказательства нас более не интересуют слова, соответствующие вершинам данного графа; нумерация вершин на рис. 5 приведена для удобства. Заметим, что графы CG_{A_0} и $CG_{A_{-1}}$ имеют такую же структуру, в чем можно убедиться непосредственно из рис. 3. Второй этап доказательства завершен.

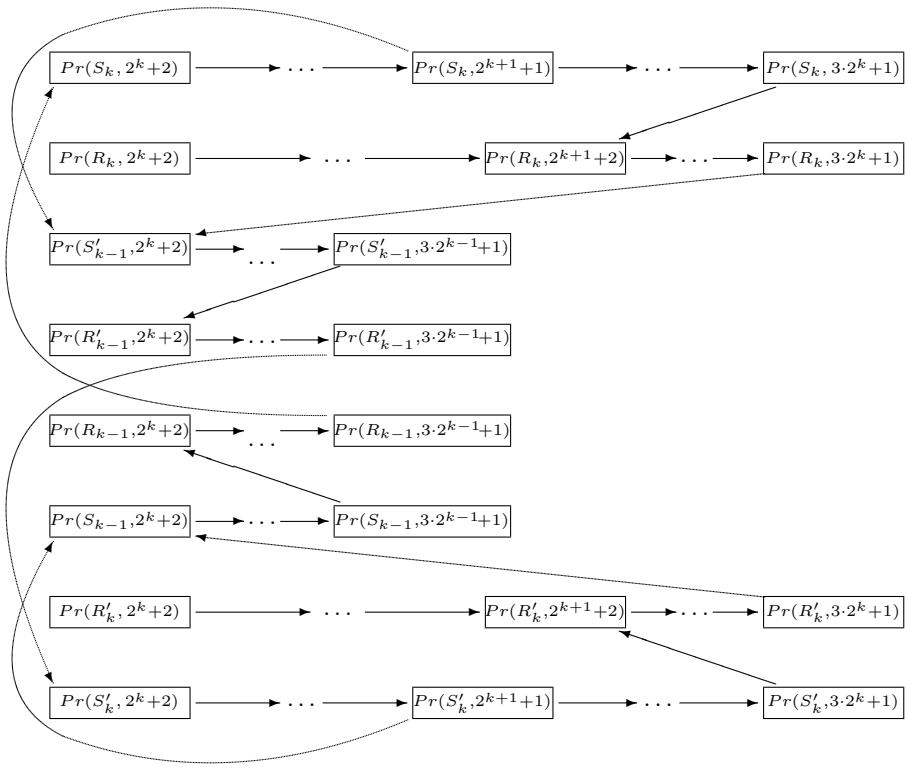


Рис. 4. Фрагмент автомата A_k, содержащий все его циклы

Перейдем к заключительному этапу доказательства. Выше мы показали существование такой константы N, что в автомате A_k все маршруты длины не менее N заканчиваются в вершине из CG_{A_k}. Тогда для всех

$n \geqslant N$ имеем

$$C_{\mathcal{L}(M_k)}(n) = \sum_{q \in CG_{A_k}} P_q(n).$$

Выведем рекуррентную формулу для вычисления $C_{\mathcal{L}(M_k)}(n)$. Из струк-

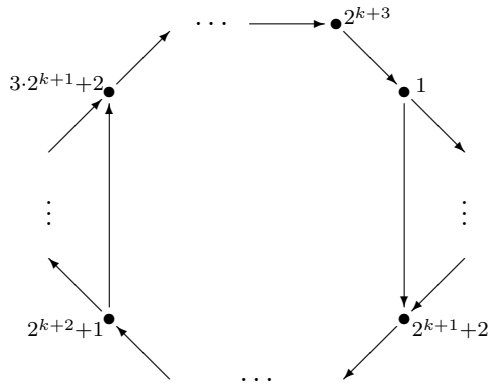


Рис. 5. Граф CG_{A_k}

туры графа CG_{A_k} (рис. 5) непосредственно выводятся соотношения

$$P_{2^{k+1}+2}(n{+}1) = P_1(n) + P_{2^{k+1}+1}(n),$$
$$P_{3\cdot 2^{k+1}+2}(n{+}1) = P_{2^{k+2}+1}(n) + P_{3\cdot 2^{k+1}+1}(n),$$

а также $P_1(n{+}1) = P_{2^{k+3}}(n)$ и $P_{i+1}(n{+}1) = P_i(n)$ для всех остальных i. Таким образом, $C_{\mathcal{L}(M_k)}(m)$ при $m \geqslant n$ является линейной комбинацией величин $P_1(n), \ldots, P_{2^{k+3}}(n)$ с целыми коэффициентами. Значения коэффициентов сведены в приводимую ниже таблицу. Взяв строки таблицы, соответствующие длинам n, $n{+}2^{k+1}$ и $n{+}2^{k+2}$, получаем рекуррентное соотношение

$$C_{\mathcal{L}(M_k)}(n{+}2^{k+2}) = C_{\mathcal{L}(M_k)}(n) + C_{\mathcal{L}(M_k)}(n{+}2^{k+1}).$$

Длина	1 $\ldots$ 2^{k+1}	2^{k+2}	$3{\cdot}2^{k+1}$	2^{k+3}
n	1 1 1 $\ldots$ 1	1 1 $\ldots$ 1	1 1 1 $\ldots$ 1	1 1 $\ldots$ 1
$n{+}1$	2 1 1 $\ldots$ 1	1 1 $\ldots$ 1	2 1 1 $\ldots$ 1	1 1 $\ldots$ 1
$n{+}2$	2 2 1 $\ldots$ 1	1 1 $\ldots$ 1	2 2 1 $\ldots$ 1	1 1 $\ldots$ 1
$\vdots$	$\vdots$ $\vdots$ $\vdots$ $\ddots$ $\vdots$	$\vdots$ $\vdots$ $\ddots$ $\vdots$	$\vdots$ $\vdots$ $\vdots$ $\ddots$ $\vdots$	$\vdots$ $\vdots$ $\ddots$ $\vdots$
$n{+}2^{k+1}$	2 2 2 $\ldots$ 2	1 1 $\ldots$ 1	2 2 2 $\ldots$ 2	1 1 $\ldots$ 1
$n{+}2^{k+1}{+}1$	3 2 2 $\ldots$ 2	2 1 $\ldots$ 1	3 2 2 $\ldots$ 2	2 1 $\ldots$ 1
$\vdots$	$\vdots$ $\vdots$ $\vdots$ $\ddots$ $\vdots$	$\vdots$ $\vdots$ $\ddots$ $\vdots$	$\vdots$ $\vdots$ $\vdots$ $\ddots$ $\vdots$	$\vdots$ $\vdots$ $\ddots$ $\vdots$
$n{+}2^{k+2}$	3 3 3 $\ldots$ 3	2 2 $\ldots$ 2	3 3 3 $\ldots$ 3	2 2 $\ldots$ 2

Возьмем любую бесконечную арифметическую прогрессию $\{m_i\}$ из натуральных чисел с разностью 2^{k+1}; пусть $m_i = m + i{\cdot}2^{k+1}$. Тогда последовательность $\{C_{\mathcal{L}(M_k)}(m_i)\}$ будет последовательностью Фибоначчи. Ее индекс роста известен и равен золотому сечению:

$$\lim_{i\to\infty} \sqrt[i]{C_{\mathcal{L}(M_k)}(m_i)} = \frac{1+\sqrt{5}}{2}.$$

Тогда

$$\lim_{i\to\infty} \sqrt[m_i]{C_{\mathcal{L}(M_k)}(m_i)} = \lim_{i\to\infty} \sqrt[i\cdot 2^{k+1}]{C_{\mathcal{L}(M_k)}(m_i)}$$
$$= \sqrt[2^{k+1}]{\lim_{i\to\infty} \sqrt[i]{C_{\mathcal{L}(M_k)}(m_i)}} = \sqrt[2^{k+1}]{\frac{1+\sqrt{5}}{2}}.$$

Поскольку число таких арифметических прогрессий конечно, каждое натуральное число принадлежит одной прогрессии, стандартные ε-рассуждения из математического анализа дают требуемую формулу

$$\alpha = \lim_{i\to\infty} \sqrt[n]{C_{\mathcal{L}(M_k)}(n)} = \sqrt[2^{k+1}]{\frac{1+\sqrt{5}}{2}}.$$

Теорема 4 доказана.

Известно, что сложность языка Туэ–Морса растет линейно и это минимальный порядок роста сложности языка с бесконечным антисловарем. Тем самым теорема 4 служит весомым аргументом в пользу следующей гипотезы.

Гипотеза. Если L — факторный язык и антисловарь $\mathcal{AD}(\mathcal{L})$ бесконечен, то сложность любого рационального приближения языка L экспоненциальна.

ЛИТЕРАТУРА

1. **Гинзбург С.** Математическая теория контекстно-свободных языков. М.: Мир, 1970.
2. **Шур А. М.** Структура множества бескубных Z-слов в двухбуквенном алфавите // Изв. РАН. Сер. матем. 2000. Т. 64, № 4. С. 201–224.
3. **Brandenburg F.-J.** Uniformly growing k-th power free homomorphisms // Theoret. Comput. Sci. 1983. V. 23, N 1. P. 69–82.
4. **Cassaigne J.** Special factors of sequences with linear subword complexity // Developments in language theory, II. Singapore: World Scientific, 1996. P. 25–34.

5. Cassaigne J. Constructing infinite words of intermediate complexity // Developments in languages theory. Berlin: Springer, 2002. P. 173–184. (Lecture Notes in Comput. Sci.; V. 2295).

6. Choffrut C., Karhumäki J. Combinatorics of words // Handbook of formal languages. Vol. 1. Words, languages, grammar. Berlin: Springer, 1997. P. 329–438.

7. Crochemore M., Mignosi F., Restivo A., Salemi S. Data compression using antidictionaries // Proc. of the IEEE. 2000. V. 88, N 11. P. 1756–1768.

8. Karhumäki J., Shallit J. Polinomial versus exponential growth in repetition-free binary words // J. Combin. Theory. Ser. A. 2004. V. 105, N 2. P. 335–347.

9. Kobayashi Y. Repetition-free words // Theoret. Comput. Sci. 1986. V. 44. P. 175–197.

10. Shur A. M. Binary words avoided by the Thue–Morse sequence // Semigroup Forum. 1996. V. 53, N 2. P. 212–219.

11. Shur A. M. Overlap-free words and Thue–Morse sequences // Int. J. Algebra and Comput. 1996. V. 6, N 3. P. 353–367.

Адрес автора:

Уральский гос. университет,
пр.Ленина, 51,
620083 Екатеринбург, Россия.
E-mail: Arseny.Shur@usu.ru

Статья поступила
18 августа 2004 г.

Переработанный вариант —
27 декабря 2004 г.