ФГАОУ ВПО «УрФУ имени первого президента России Б.Н.Ельцина»
Институт математики и компьютерных наук

Кафедра алгебры и дискретной математики

Вычислительный эксперимент с разбиениями Линдона и Лемпеля-Зива

Отчет по учебной практике
студента 2 курса группы МК-240014
Борзунова Александра Александровича

Научный руководитель:
доктор физико-математических наук
профессор
Шур Арсений Михайлович

Екатеринбург
2016

1. Поставленная задача

Пусть $Z(S)$ - число сегментов в разбиении Лемпеля-Зива определённой строки, а $L(S)$ - число *различных* сегментов в разбиении Линдона. Существует следующая гипотеза:

$$L(S) = O(Z(S))$$

Была известна только одна серия бинарных строк, для которой $L(S) > Z(S)$. Предлагалось перебором коротких строк найти ещё серии с этим свойством.

2. Алгоритмы и техники, используемые в переборе

Для написания программы для перебора был выбран язык программирования Go. Это сравнительно новый язык (появился в 2009 году), который:

- Является компилируемым (программы на нём могут достигать производительности программ, написанных на Си);
- Безопасно управляет памятью (это упрощает отладку программ по сравнению с Си);
- Позволяет легко распараллелить вычисления между ядрами процессора.

Реализованная программа параллельно на всех ядрах процессора перебирает и проверяет все строки, начиная от меньших длин к большим. Выводятся все строки, где $L(S) > Z(S)$. С программой для перебора прилагается вспомогательная утилита, которая показывает разбиения произвольных строк и величины $L(S)$, $Z(S)$ для них.

Для вычисления количества различных сегментов в разбиении Линдона используется алгоритм Дюваля [1].

Для вычисления количества сегментов в разбиении Лемпеля-Зива используется алгоритм [2] (Goto, Bannai, 2012). Для его использования требуется предварительно построить суффиксный массив строки, что делается одним из следующих способов:

- Алгоритм [3] (Larsson, Sadakane, 2007), строящий суффиксный массив за $O(n \log n)$, где n — длина строки. Этот алгоритм уже реализован в стандартной библиотеке Go. Данная реализация была модифицирована так, чтобы работать только со статическими буферами и избежать регулярных выделений памяти (что значительно ускорило программу).
- Для коротких бинарных строк (чья длина не больше 64) можно использовать другой способ. Бинарную строку можно представить в виде переменной, содержащей одно целое 64-битное число. Суффиксы такой строки можно быстро сравнивать между собой, если выделить их с помощью операции битового сдвига влево (справа они дополнятся нулями), а затем сравнить как 64-битные числа (если они равны, следует также учесть длины суффиксов). Тогда суффиксы будут сравниваться лексикографически с помощью простых для процессора операций за $O(1)$.

  Используя описанный метод сравнения суффиксов по их смещениям, достаточно отсортировать массив со смещениями суффиксов за $O(n \log n)$ и получить

искомый суффиксный массив. На практике этот метод оказался существенно быстрее предыдущего.

В итоге первый метод использовался для вычисления разбиений длинных строк (их понадобилось строить, чтобы проверить свойства полученных серий), а второй метод — для разбиения коротких строк во время перебора.

Построение разбиения Лемпеля-Зива всё равно являлось узким местом в программе (оно выполняется значительно медленнее, чем построение разбиения Линдона). Для ускорения программы был использован тот факт, что $Z(S) = Z(\bar{S})$, где $\bar{S}$ — строка, полученная из S заменой всех 0 на 1 и всех 1 на 0. Это позволяет вычислять в два раза меньше разбиений Лемпеля-Зива, что ускорило программу ещё в 1,5–2 раза.

Итоговая версия программы требует $O(2^n \cdot n \log n)$ времени, чтобы проверить все строки длины n. На четырёхядерном процессоре ноутбука Intel Core i5-3317U проверка всех бинарных строк длины 25 занимает около 22 секунд.

Исходный код доступен по ссылке: `https://github.com/borzunov/course-work-2016`

3. Полученные результаты

3.1. Слова, где $L(S) > Z(S)$

С помощью перебора, описанного в предыдущем разделе, были проверены все бинарные слова длиной до 34 символов включительно. Оказалось, что самое короткое слово с данным свойством имеет длину 14. На проверенных длинах разность $L(S) - Z(S)$ не превысила 2. Таблицы с кратчайшими словами, где $L(S) > Z(S)$, и их количеством представлены в приложении А.

Список всех найденных слов с данным свойством доступен по ссылке: `https://raw.githubusercontent.com/borzunov/course-work-2016/master/bruteforce/result.txt`

Простого закона для зависимости кол-ва слов длины n с данным свойством от числа n (рис. 1 ниже) найдено не было. По графику на рис. 2 можно предположить, что доля слов с данным свойством среди всех слов определённой длины стремится к нулю.

3.2. Построенные серии строк

После поиска закономерностей в кратчайших словах с заданным свойством по аналогии были построены несколько серий строк, где $L(S)$ существенно превышает $Z(S)$. Тем не менее, во всех сериях не нарушается гипотеза о том, что $L(S) = O(Z(S))$. Серии перечислены ниже:

1)

$$S_n = \prod_{i=n}^{1} (10)^i 0$$

Для $n = 2k$ имеем:

$$L(S_n) = 2k + 2 \qquad Z(S_n) = k + 3 \qquad \frac{L(S_n)}{Z(S_n)} \to 2$$

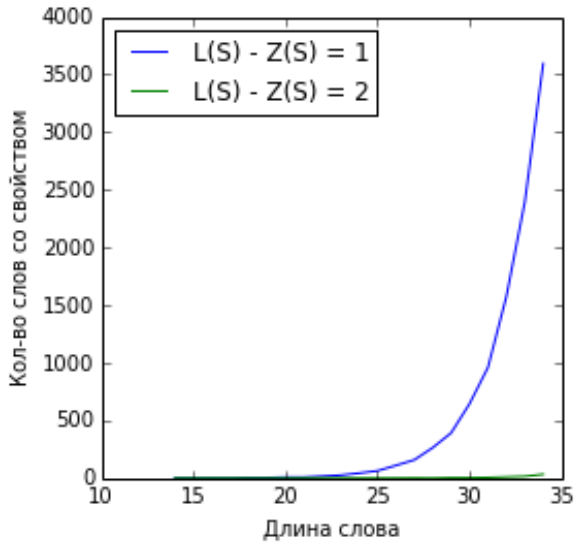

Рис. 1. Кол-во слов с указанными свойствами.

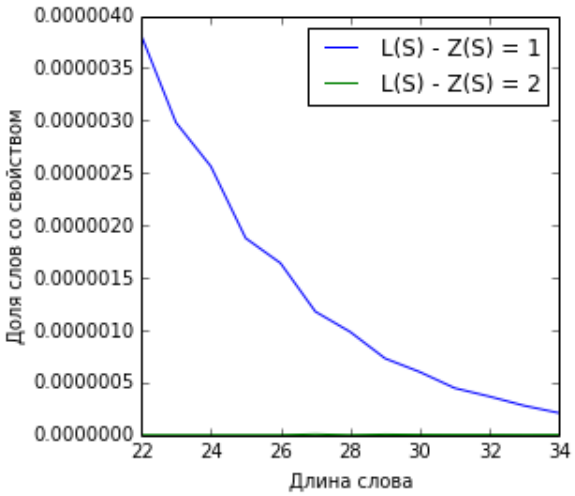

Рис. 2. Доля слов с указанными свойствами среди всех слов заданной длины.

Пример для $n = 6$:

- Слово:
  `101010101010 0 1010101010 0 10101010 0 101010 0 1010 0 10 0`
- Разбиение Линдона (8 различных сегментов):
  `1 01 01 01 01 01 00101010101 001010101 0010101 00101 001 0 0`
- Разбиение Лемпеля-Зива (6 сегментов):
  `1 0 1010101010 01010101010010101010 010101001010 0100`

Похожая серия с чуть более сложной формулой при $n = 2$ даёт слово наименьшей длины, у которого $L(S) > Z(S)$:

$$S_n = \left(\prod_{i=n+1}^{2} (10)^i 0 \right) \cdot 10$$

Упомянутое слово имеет длину 14:

- Слово:
  `101010 0 1010 0 10`
- Разбиение Линдона (5 различных сегментов):
  `1 01 01 00101 001 0`
- Разбиение Лемпеля-Зива (4 сегмента):
  `1 0 1010 01010010`

2)

$$S_n = \prod_{k=n}^{1} \prod_{i=k}^{1} (10)^i 0$$

Для $n = 2k$ имеем:

$$L(S_n) = 4k \qquad Z(S_n) = 3k \qquad \frac{L(S_n)}{Z(S_n)} = \frac{4}{3}$$

Пример для $n = 4$:

- Слово:
  `10101010 0 101010 0 10100 10 0  101010 0 1010 0 10 0  1010 0 10 0  10 0`
- Разбиение Линдона (8 различных сегментов):
  `1 01 01 01 0010101 00101 001001010100101 00100101 001 001 0 0`
- Разбиение Лемпеля-Зива (6 сегментов):
  `1 0 101010 010101001010 0100101010010100100 1010 0100100`

3)

$$S_n = \prod_{k=n}^{1} \prod_{j=k}^{1} \prod_{i=j}^{1} (10)^i 0$$

Для $n = 2k$ имеем:

$$L(S_n) = 6k - 2 \qquad Z(S_n) = 5k - 3 \qquad \frac{L(S_n)}{Z(S_n)} \to \frac{6}{5}$$

Пример для $n = 4$:

- Слово:
  `1010101001010100101001001010100101001001010010010100101001010010010100 1001001010010010010`
- Разбиение Линдона (10 различных сегментов):
  `1 01 01 01 0010101 00101 001001010100101 00100101 0010010010101001010010010 00100100101 001 001 001 0 0`
- Разбиение Лемпеля-Зива (7 сегментов):
  `1 0 101010 010101001010 0100101010010100100 1010 0100100100 1001001010100101001001010 0100100100`

4) Аналогично предыдущим сериям, можно составить:

$$S_n = \prod_{m=n}^{1} \prod_{k=m}^{1} \prod_{j=k}^{1} \prod_{i=j}^{1} (10)^i 0$$

Для $n = 2k$ имеем:

$$L(S_n) = 8k - 4 \qquad Z(S_n) = 7k - 6 \qquad \frac{L(S_n)}{Z(S_n)} \to \frac{8}{7}$$

Список литературы

[1] Duval, Jean-Pierre (1988), "Génération d'une section des classes de conjugaison et arbre des mots de Lyndon de longueur bornée Theoretical Computer Science (in French) 60 (3): 255–283, doi:10.1016/0304-3975(88)90113-2, MR 979464

[2] Keisuke Goto, Hideo Bannai (2012), Simpler and Faster Lempel Ziv Factorization, arXiv:1211.3642

[3] N. Jesper Larsson, Kunihiko Sadakane (2007), Faster suffix sorting, doi:10.1016/j.tcs.2007.07.017

Приложение. Таблицы с кратчайшими словами, где $L(S) > Z(S)$, и их количеством

Таблица 1. Кратчайшие слова, где $L(S) - Z(S) = 1$.

Длина	Слово	$L(S)$	$Z(S)$
14	10101001010010	5	4
16	10101010010010	5	4
18	101001000100100010	6	5
	101010101001010010	5	4
	101010010100100010	6	5
	101010100101010010	5	4
19	1010110100110100110	6	5
	1010010001001000100	6	5
	1010100101001010010	5	4
	1010100101001000100	6	5
20	10101101001011010010	6	5
	10101001010010001000	6	5
	10101010010101001010	5	4
	10101010101001010010	5	4
	10101001000100100010	6	5
	10101010010100100010	6	5
	10101001010010001010	6	5
	10101010100101010010	5	4

Таблица 2. Кратчайшие слова, где $L(S) - Z(S) = 2$.

Длина	Слово	$L(S)$	$Z(S)$
27	101010010100100010100100010	7	5
29	10101001010010001010010001010	7	5
	10101101001101001100100110010	8	6
	10101010010100100010100100010	7	5
30	101010100101010010100010100010	7	5
	101010010100100010100100010100	7	5
	101011010011010011001001100100	8	6
31	1010101001010010001010010001010	7	5
	1010101010010100100010100100010	7	5
	1010100101001000101000100100010	8	6
	1010101101001101001100100110010	8	6
	1010101001010100101001001010010	7	5

Таблица 3. Кол-во слов определённой длины с исследуемыми свойствами.

Длина	Слов, где $L(S) - Z(S) = 1$	Слов, где $L(S) - Z(S) = 2$
14	1	0
15	0	0
16	1	0
17	0	0
18	4	0
19	4	0
20	8	0
21	8	0
22	16	0
23	25	0
24	43	0
25	63	0
26	110	0
27	158	1
28	264	0
29	392	3
30	645	3
31	960	5
32	1575	10
33	2395	15
34	3596	33