Постановка задачи

Пусть *А* = (Σ, Q, δ, s, T) – детерминированный конечный автомат, где Σ – входной алфавит, из которого формируются слова, принимаемые автоматом, Q – множество состояний автомата, δ – функция переходов, определенная как отображение δ: Q × Σ → Q, s ­– начальное состояние, T – множество терминальных (конечных) состояний.

Автомат принимает слово, если по окончании его обработки он находится в терминальном состоянии.

Пусть u и v – слова над алфавитом Σ.

Говорят, что *А* различает слова u и v, если он принимает одно из них и не принимает другое. Пару (u, v) будем называть тождеством.

Обозначим за sep(u, v) – количество состояний в минимальном детерминированном конечном автомате, различающем u и v.

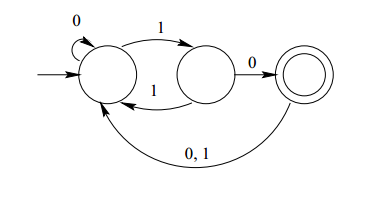
Например, автомат на рисунке (1) различает слова 0010 и 1000.

Рисунок 1.

Однако, ни один автомат из двух состояний различить эти слова не сможет. Значит, sep(0010, 1000) = 3.

Очевидно, что sep(u, v) = sep(v, u).

Пусть S(n) =

Задача о различении слов, известная как Separating Words Problem, состоит в том, чтобы найти хорошую асимптотическую оценку функции S(n).

Другими словами, сколько состояний должно быть в автомате, различающем две строки длины n?

В 1986 году эту задачу сформулировали Павел Горальчик и Вацлав Коубек. Они же доказали, что S(n) = o(n). Позже этой задачей занимался Джон Робсон. В 1989 году он доказал, что S(n) = O(n2/5(log n)3/5) для произвольных автоматов, и в 1996 году опубликовал статью, в которой показал, что S(n) = O(n1/2) для перестановочных автоматов. Однако зазор между верхней и нижней границей до сих пор остается открытой задачей.

В терминах алгебры……………..

Исследования данной задачи сводятся к поиску тождеств для автоматов.

Цель моих экспериментов – наблюдение за тем, как различают слова автоматы из 4 – 6 состояний, и поиск тождеств для них среди строк длины

40 – 60.

Используемые факты

Длина рассматриваемых строк.

Рассматриваемые мной строки одинаковой длины. Этот случай более интересен, поскольку доказано, что для слов u и v разной длины всегда существует маленькое простое число p (р = O(log n), где n – минимум из длин u и v), такое, что длины u и v различны по модулю р. Чтобы различить строки в данном случае, можно использовать автомат, считающий длину своего входа по модулю р. Другими словами, sep(u, v) = O(log n), и существует бесконечное количество случаев, когда sep(u, v) = Ω(log n).

В дальнейшем, я буду рассматривать пары строк одинаковой длины.

Независимость от размера алфавита.

Без ограничения общности можно рассматривать строки над двухбуквенным алфавитом. Для строк над алфавитом большего размера, различных для некоторого автомата, существует гомоморфизм, сопоставляющий им строки той же длины над двухбуквенным алфавитом, различающиеся автоматом с тем же количеством состояний. Любой автомат, различающий двоичные строки, можно преобразовать в автомат, различающий строки над исходным алфавитом, не меняя количество его состояний.

Формально говоря, Sk(n) = S2(n) для всех k ≥ 2, где k – длина алфавита. Значение интересующей нас функции не зависит от размера алфавита, над которым рассматриваются строки. Поэтому в дальнейшем я буду рассматривать строки над алфавитом {0, 1}.

Одинаковое количество нулей и единиц.

Весом Хэмминга называют количество ненулевых символов в строке.

Если две строки имеют различное количество нулей и единиц, они могут быть различены автоматом, считающим вес Хэмминга у входной строки по модулю р, где р – небольшое простое число, р = O(log n).

Счетчик подстрок.

Если некая подстрока длины k встречается в строках разное количество раз, то эти строки можно различить автоматом с количеством состояний

O(k log n).

Случайные строки.

Для двух случайно выбранных строк u и v sep(u, v) = O(1).

Различие вблизи начала строки.

Если слова u и v различаются в позиции d, считая с начала, то

sep(u, v) ≤ d + 2.

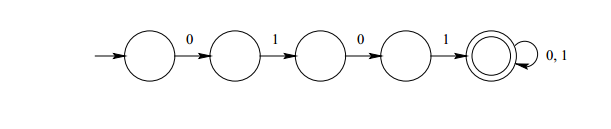
Например, чтобы различить слова 0101101010101010 и 0100101010101010 можно построить следующий автомат:

Рисунок 2.

На данном рисунке опущены переходы в «мертвое» состояние.

Стоит заметить, что автомат, различающий слова, отличающиеся близко к началу, не является перестановочным. Это утверждение нам понадобится чуть позже.

Различие вблизи конца строки.

Если наибольший общий суффикс u и v имеет длину k, то u и v можно различить, используя автомат с k+2 состояниями.Отсюда вытекает следующее.Если слова u и v различаются в позиции d, считая с конца, то sep(u, v) ≤ d + 1.

Автомат, различающий строки, отличающиеся близко к концу, также не является перестановочным.

Расстояние Хэмминга.

Пусть u и v – строки одинаковой длины. u и v имеют расстояние Хэмминга d, если они различаются в d позициях и совпадают в остальных.

Если две строки имеют расстояние Хэмминга d, то существует простое число p = O(d log n) и позиция i, в которой строки отличаются, и которая по модулю p не равна никакой другой такой позиции. Путем вычисления четности входных символов в позициях, сравнимых с i по модулю p, можно различить строки, используя автомат с O(d log n) состояниями.

Известные тождества.?

Поставленные эксперименты

Рассматриваемые автоматы и строки

В своих экспериментах я буду мыслить автомат не как пятерку, а как четверку составляющих, опуская множество терминальных состояний, однако считая, что все состояния терминальные. *А* = (Σ, Q, δ, s, T). В связи с этим, будем говорить, что автомат различает слова u и v, если он заканчивает их чтение в разных состояниях, то есть s.u ≠ s.v, где s.u – состояние, в которое приходит автомат, читая слово u из состояния s. Опустить множество терминальных состояний удобно для перебора автоматов, это значительно сокращает количество автоматов.

Также, имеет смысл рассматривать только автоматы, все состояния которых достижимы, поскольку если в автомате есть недостижимые состояния, его можно рассматривать как автомат с меньшим количеством состояний.

Если пара (u, v) является тождеством для автомата из N состояний, она будет тождеством для всех автоматов из M < N состояний.

Напомню также, что я рассматриваю строки одинаковой длины над алфавитом {0, 1}, количества нулей (и единиц соответственно) в строках совпадают.

Фильтрация строк случайными автоматами

При проведении экспериментов может возникнуть ситуация, когда пар-кандидатов на проверку на тождество оказывается очень много. Если у автомата совсем маленькое количество состояний (скажем, 2 или 3), справиться за разумное время конечно можно. Но уже для автоматов из 4 и более состояний процесс может затянуться.

Однако, список кандидатов можно сократить. Для этого можно каждую пару-кандидат попытаться различить небольшим количеством случайных автоматов, получив при этом список «трудноразличимых» пар. Как показала практика, полученный список действительно меньше исходного, а значит и его обработка пройдет быстрее.

FilterByRandom(pairs, numberOfStates, getRandomAutomata, filtersNumber)

Вход: pairs – список пар-кандидатов на тождества, n – количество состояний в автомате, getRandomAutomata – функция, возвращающая случайный автомат с n состояниями, filtersNumber – число случайных автоматов-фильтров.

Выход: отфильтрованный список пар.

filteredList ← Ø

foreach pair ∈ pairs

separated ← false

for i from 0 to filtersNumber-1

automata = getRandomAutomata(n)

if(automata.Separates(pair)

separated ← true

break

if(not separated)

filteredList.Add(pair)

return filteredList

К примеру, на рисунке … изображен график зависимости времени различения от количества пар строк. Верхняя линия отвечает за различение без использования фильтрации, нижняя – за различение с предварительной фильтрацией. Различение проводилось произвольными автоматами с пятью состояниями. Строки-кандидаты – тождества для произвольных автоматов с четырьмя состояниями. Количество случайных автоматов-фильтров – 20.

Перестановочные автоматы

Чтобы сократить количество перебираемых автоматов, большую часть экспериментов я посвятила различению слов перестановочными автоматами с маленьким количеством состояний и поиску тождеств для них.

Напомним, что *А* = (Σ, Q, δ, s, T) – детерминированный конечный автомат.

Автомат *А* – перестановочный, если переход из любого состояния по любому символу является перестановкой состояний, или, что тоже самое, для любого символа x из Q и любых состояний q и p δ(q, x) ≠ δ(p, x).

Так как автоматы, которые различают строки, отличающиеся в позиции, близкой к началу или концу, не являются перестановочными, строки, имеющие общий префикс или суффикс, в данном контексте не интересны. Поэтому для перестановочных автоматов имеет смысл рассматривать те пары, где строки начинаются с разных символов, а также заканчиваются разными символами.

Эксперимент на основе графа Рози.

Известно, что автоматы-счетчики подслов – это перестановочные автоматы с kp состояниями, где k - длина подслова и p - модуль, по которому количества подслов не равны. Таким образом, первой идеей было рассмотреть пары строк с одинаковым набором подстрок длины не больше 5. Для этого я использовала граф Рози пятого порядка.

Граф Рози порядка k – это граф, вершинами которого являются подстроки длины k, а вершины u и v соединены ребром тогда и только тогда, когда

u[2, k] = v[1, k – 1], где u[i, j] – подстрока строки u, начинающаяся в позиции i и заканчивающаяся в позиции j.

Идея эксперимента.

Для случайно выбранной строки длины N (N = 40 ÷ 50 в проведенных мною экспериментах) строится соответствующий ей граф Рози 5 порядка. Затем в нем я нахожу два цикла с общей вершиной и меняю их местами. Таким образом получается пара выбранной случайно строке. Такие пары я пробую отфильтровать способом, описанным в первом разделе, 10 случайными перестановочными автоматами из 5 состояний. Такой эксперимент на 100000 различных пар строк дал примерно 700 пар трудноразличимых. Однако в ходе такого эксперимента тождеств для перестановочных автоматов длины 5 мне найти не удалось. Возможно, это связано с тем, что получаемые мной строки имели маленькое расстояние Хэмминга. Известно, что если два слова u и v имеют расстояние Хэмминга меньше d, то sep(u, v) ≤ O(d log n), где n – длина рассматриваемых слов.

В поиске других идей я обратилась к уже известным тождествам для перестановочных автоматов из 5 и 6 состояний. Они были найдены среди пар, где второй строкой бралась обратная запись первой строки.

Я заметила, что строка из пары состоит из двух блоков четной длины, и в каждом таком блоке количество 0 и 1 совпадает. Более того, в них не содержалось строк 000 и 111. Ограничив таким образом область, из которой бралась случайная строка, и принимая в пару обратную запись строки, я получила более существенный результат. Сначала я перебрала все строки описанного выше вида длины 40 и получила 10 тождеств, которые, однако, оказались модификациями уже известных. После чего я взялась за строки длины 48, перебрать из оказалось уже сложно, поэтому я брала лишь случайные строки. После серии экспериментов мне удалось найти три тождества, которые не получить модификацией уже известных: два тождества длин 44 и 48 для перестановочных автоматов из 5 состояний и одно тождество длины 48 для автоматов с 6 состояниями. Подобным образом я нашла еще некоторое количество тождеств.

Произвольные автоматы

Автоматы, которые различают слова, отличающиеся в самом начале или в самом конце, не являются перестановочными. Поэтому все тождества для перестановочных автоматов имеет смысл искать среди пар строк, начинающихся и заканчивающихся по-разному. Соответственно, чтобы превратить тождество для перестановочных автоматов в тождество для произвольных, нужно дописать к строкам пары одинаковые префикс и суффикс. Притом, учитывая факты 4 и 5 (см. в разделе используемые факты), префикс должен быть длины не менее k-1, а суффикс - длины не менее k, чтобы исключить различение автоматами из k состояний около начала и конца строк.

Таким образом, я провела следующие эксперименты.

В первом случае я брала тождества для перестановочных автоматов с 5 состояниями, дописывала к ним одинаковые префикс и суффикс, и пыталась различить их всеми автоматами с 4 состояниями. Суффикс брала как обратную запись префикса. В ходе такого эксперимента удалось получить несколько сотен тождеств для произвольных автоматов из 4 состояний.

Во втором случае я взяла найденное в первом эксперименте тождество для произвольных автоматов с 4 состояниями и провела следующий процесс.

На каждой итерации, перебирая все автоматы с 5 состояниями по порядку, находила первый автомат, который различал пару. Затем я искала такие суффикс и префикс для строк, чтобы при добавлении их к строкам, этот автомат перестал их различать. И далее искала уже следующий автомат, который их различит. Таким жадным алгоритмом я нашла тождество для произвольных автоматов с 5 состояниями длины 128.

Надеюсь, что полученные мною тождества в дальнейшем помогут узнать больше о различении строк автоматами.