4
П-71
12
Виноградов

ИНСТИТУТ РУССКОГО ЯЗЫКА АН СССР

Проблемная группа по экспериментальной
и прикладной лингвистике

ПРЕДВАРИТЕЛЬНЫЕ ПУБЛИКАЦИИ

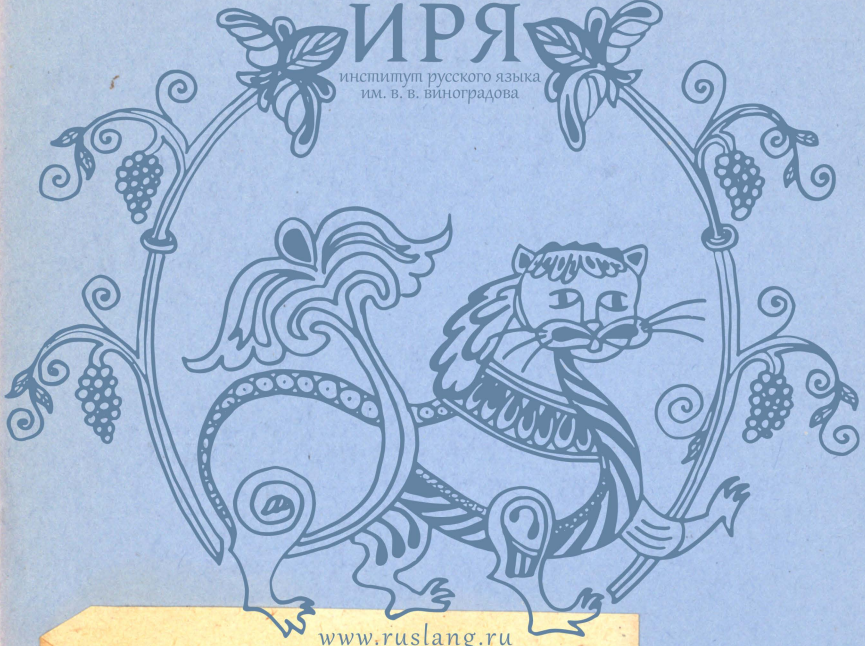
/12/

Б. В. СУХОТИН

НЕКОТОРЫЕ НЕЛОКАЛЬНЫЕ АЛГОРИТМЫ ЛИНГВИСТИЧЕСКОЙ ДЕШИФРОВКИ

Москва — 1970

ИРЯ
институт русского языка
им. в. в. виноградова



www.ruslang.ru

66100

ИНСТИТУТ РУССКОГО ЯЗЫКА АН СССР
Проблемная группа по экспериментальной и
прикладной лингвистике

ПРЕДВАРИТЕЛЬНЫЕ ПУБЛИКАЦИИ

Б.В. Сухотин

НЕКОТОРЫЕ НЕЛОКАЛЬНЫЕ АЛГОРИТМЫ ЛИНГВИСТИЧЕСКОЙ ДЕШИФРОВКИ

Москва - 1970

4
Д-71/12
БИБЛИОТЕКА
ИН-ТА РУССКОГО ЯЗЫКА
ИН-ТА ЯЗЫКОЗНАНИЯ АН СССР
66100

I. Дешифровочный алгоритм.

Дешифровочным алгоритмом называется алгоритм, предназначенный для опознавания некоторого лингвистического явления в текстах заданного вида. Дешифровочный алгоритм обычно имеет следующую структуру:

1) множество возможных решений (интерпретаций) P;

2) функция качества Q, определённая на этом множестве;

3) вычислительная процедура (или собственно алгоритм) A, имеющая целью поиск экстремума функции Q, т.е. наилучшего решения.

В статье разбираются, главным образом, три алгоритма, которые будут называться, соответственно, алгоритмами I, II и III.

2. Алгоритм I.

Хорошим примером дешифровочного алгоритма является алгоритм I, обнаруживающий в фонематическом тексте гласные и согласные. Для этого алгоритма множество возможных решений есть множество разбиений алфавита букв L на два непересекающихся подмножества C и V.

Функция Q строится, исходя из предположения о редкой встречаемости рядом букв одного и того же класса, следующим образом:

$$Q = \sum_i \sum_j f(l_i, l_j) + \sum_p \sum_q f(l_p, l_q). \qquad (B1)$$

Здесь $l_i \in V, l_j \in V, l_p \in C, l_q \in C$;
числа $f(l_i, l_j)$ и $f(l_p, l_q)$ представляют собой количества вхождений пар соответствующих букв без различия порядка (но расположающихся рядом в тексте); суммирование ведётся по всем таким парам. Наилучшему решению соответствует минимум Q.

Как найти минимум Q ? Помимо тривиальной вычислительной процедуры (перебора) может быть предложен ряд практичных методов, если и не находящих истинный минимум Q, то, во всяком случае, дающих разбиение, для которого Q достаточно мало. Таков, напри-мер, нижеследующая процедура:

1.для данного текста построить таблицу чисел f(x,y) , где f(x,y) означает количество появлений пары букв x,y рядом в тексте,в безразличном порядке.Текст можно считать написанным без пробелов;
2.вычеркнуть числа вида f(x,x);
3.подсчитать сумму чисел в каждой строке таблицы;
4.поставить на первое место (от левого верхнего угла) строку, для которой сумма чисел максимальна.Так же переставить соответствующий столбец;
5.отделить линиями переставленные строки и столбцы от непереставленных;
6.для строк,лежащих ниже горизонтальной линии подсчитать сумму чисел правее вертикальной линии и сумму чисел левее вертикальной линии;вторую сумму вычесть из первой.Полученные числа назовём решающими разностями;
7.определить,имеются ли положительные решающие разности.Если имеются,то перейти к выполнению команды 9;
8.конец.Строки,лежащие выше горизонтальной линии,соответствуют первому классу (обычно -гласных),ниже - второму классу (обычно-согласных);
9.выбрать строку,которой соответствует максимальная положительная решающая разность и перенести через горизонтальную линию, а соответствующий столбец - через вертикальную.Перейти к выполнению пункта 6.

Можно привести небольшой пример,показывающий как работает этот алгоритм.Определим гласные текста,представленного русским словом "собака".Как отмечалось,для работы этого алгоритма не нужно знать,где проходят границы между словами.В частности,будем считать,что неизвестно,с какой буквы текст начинается,а какой кончается,как если бы текст был написан по кругу:

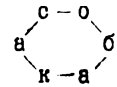


Таблица чисел f(x,y) для этого текста имеет следующий вид:

	с	о	б	а	к
с		I		I	
о	I		I		
б		I		I	
а	I		I		2
к				2	

(В эту таблицу нули, для наглядности, не вписаны).

При выполнении команды 2 таблица не изменяется. Выполнение команды 3 даёт:

	с	о	б	а	к	
с		I		I		2
о	I		I			2
б		I		I		2
а	I		I		2	4
к				2		2

Команды 4 и 5:

	а	с	о	б	к
а		I		I	2
с	I		I		
о		I		I	
б	I		I		
к	2				

Команда 6:

	а	с	о	б	к	
а		I		I	2	
с	I		I			0
о		I		I		2
б	I		I			0
к	2					-2

Седьмая команда требует перехода к девятой.Выполнение команд 9 и 6 даёт:

	а	о	с	б	к	
а			I	I	2	
о			I	I		
с	I	I				-2
б	I	I				-2
к	2					-2

Седьмая команда требует перехода к восьмой,т.е.окончить вычисления.Ответ: первый класс: а,о; второй: с,б,к.

Этот алгоритм дал очень хорошие результаты в экспериментах, проведённых на текстах объёмом в 10000 букв.Эксперименты проводились на ЭВМ.Для некоторых языков (русского,французского,испанского) результаты были практически безошибочны,для остальных (английского и немецкого) - содержали небольшие ошибки,вызванные особенностями правописания этих языков.

Удивительным образом,несовершенство алгоритма не сказалось на результатах.Это несовершенство можно обнаружить при использовании специально подобранных таблиц,например:

	a	b	c	d	e
a		II	30	5	5
b	II		3	10	10
c	30	3		I	2
d	5	10	I		
e	5	10	2		

Приведённая выше процедура,если её применить к этой таблице, даёт следующий результат:

	a	b	c	d	e
a		II	30	5	5
b	II		3	10	10
e	30	3		I	2
d	5	10	I		
c	5	10	2		

При этом сумма чисел в диагональных квадратах равна 28,однако правильным ответом является следующее разбиение:

	b	c	a	d	e
b		3	II	I0	I0
c	3		30	I	2
a	II	30		5	5
d	I0	I	5		
e	I0	2	5		

При этом сумма частот в диагональных квадратах равна 26. Назовём соседними разбиениями разбиения, отличающиеся переносом одной буквы из одного класса в другой. Локальным экстремумом функции качества назовём её значение для разбиения p в том случае, если значение функции для любого соседнего разбиения меньше (больше) её значения от p.

Оказывается, что процедура, изложенная на стр.2 не обеспечивает даже получения локального минимума. Приведём опровергающий пример:

	a	b	c	d	e	f	g	h	i	j
a		6	I0	I	I	I	I	I	I	I
b	6			I	I	I	I	2	2	2
c	I0			2	2	2	2	I	I	I
d	I	I	2							
e	I	I	2							
f	I	I	2							
g	I	I	2							
h	I	2	I							
i	I	2	I							
j	I	2	I							

Её разбиение процедурой на стр.2 даёт классы : a,b,c ; d,e, f,h,g,i,j ,тогда как соседнее разбиение b,c ; a,d,e,f,g,h,i,j обеспечивает меньшее значение функции качества.

Назовём x-разбиениями разбиения, содержащие x элементов в меньшем классе. Наилучшее (в смысле минимизации функции качества) x-разбиение назовём x-наилучшим.

Некоторую матрицу назовём регулярной, если меньший класс каждого x-наилучшего разбиения содержит меньший класс (x - I)-наилучшего разбиения; (x не больше половины мощности алфавита).

Очевидно, что алгоритм I верен в случае регулярности матрицы частот двухбуквенных сочетаний (см.стр. 13).

Нетрудно достроить алгоритм I так,чтобы обеспечивалось получение локального минимума Q. Для этого надо разрешить обратные переносы букв из первого класса во второй.Критерий переноса можно сформулировать так:в противоположный класс переносится буква a,для которой справедливо неравенство

$$\sum_x f(a,l_x) - \sum_x f(a,l_y) > 0.$$

Здесь l_y означает произвольную букву из того же класса,что и a, l_x - произвольную букву из противоположного класса.

Казалось бы,что при разрешении обратных переносов не исключена опасность зацикливания,т.е.бесконечных переносов из одного класса в другой.Простое рассуждение показывает,что такой опасности нет.В самом деле,перенос осуществляется лишь тогда,когда происходит уменьшение значения Q ;но сумма всех уменьшений не может превзойти сумму всех чисел в таблице,поэтому процесс рано или поздно оборвется.

Весьма хорошие результаты экспериментов,изложенные в I и 2 не свидетельствуют непосредственно о регулярности реальных таблиц частот двухбуквенных сочетаний.Эти результаты говорят о том,что справедлива не простая гипотеза о малой встречаемости гласных с гласными и согласных с согласными,а другая,которую можно сформулировать так:

а)самая частая буква - гласная;

б)если выделено x гласных букв,то невыделенная буква,частота сочетаний которой с выделенными буквами f_1 меньше частоты сочетаний с невыделенными буквами f_2, является потенциальной гласной.

в)потенциальная гласная,которой соответствует наибольшая разность $f_2 - f_1$, является x + I -ой гласной.

Эта гипотеза совпадала бы с гипотезой о малой встречаемости гласных с гласными и согласных с согласными в случае регулярности таблиц частот.

3.Место данной задачи среди родственных задач.

Поиск минимума Q можно представить как задачу минимизации квадратичной функции на вершинах n-мерного куба.

Будем рассматривать буквы данного алфавита $l_1, l_2, \ldots l_n$ как переменные,принимающие значение I,если буква гласная,и 0 -

если согласная. Введём ещё переменные $k_1, k_2, \ldots, k_n$, связанные с прежними соотношениями $l_i + k_i = 1$; $k_i = 1 - l_i$.
Частоту двухбуквенной цепочки l_i, l_j (или k_i, k_j) обозначим через f_{ij}.

Тогда минимизируемую функцию Q можно будет записать так:

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} l_i l_j + \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} k_i k_j.$$

Подставим вместо k_i и k_j их выражения через l_i и l_j:

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} l_i l_j - \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} (1 - l_i)(1 - l_j);$$

раскрывая скобки имеем:

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} l_i l_j + \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} - \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} l_i -$$

$$- \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} l_j + \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} l_i l_j.$$

Приводя подобные, учтём, что $f_{ij} l_j$ заменяется на $f_{ij} l_i$ ввиду того, что $l_j = l_i$ при $i = j$. Получаем:

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} + 2 \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} l_i l_j - 2 \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} l_i,$$

или, отбросив постоянную часть,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} l_i l_j - \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} l_i. \quad 1)$$

Итак, мы имеем квадратическую функцию от целочисленных переменных $l_1, l_2, \ldots, l_n$ на которые наложены ограничения, записываемые ввиде неравенств:

$$0 \leqslant l_1 \leqslant 1$$
$$0 \leqslant l_2 \leqslant 1$$
$$\ldots\ldots\ldots$$
$$\ldots\ldots\ldots$$
$$\ldots\ldots\ldots$$
$$0 \leqslant l_n \leqslant 1$$

Насколько известно автору, удовлетворительное решение подобных задач не найдено. Алгоритмическая разрешимость, однако, очевидна:

1) Максимизация полученной функции приводит к тому же ответу, что и минимизация предыдущей.

ответ всегда можно получить перебором всех разбиений, число которых конечно и равно $2^{n-1}-1$ (где n число букв в алфавите).

4. Система обозначений.

Для решения различных задач, связанных с построением алгоритмов и функций качества, нужна удобная система обозначений. Ниже мы предложим некоторую систему операций над подмножествами множества букв, основанную на векторном представлении этих множеств.

Пусть у нас имеется множество букв L, множество пар элементов этого множества L^2 и числовая функция F на L^2. Упорядочив произвольным образом L мы получим алфавит A - кортеж длины n, где n - количество букв в L. Теперь каждому подмножеству M из L сопоставим вектор длины n, состоящий из нулей и единиц. Этот вектор получается из алфавита L заменой букв из M на единицы и остальных букв на нули.

Функция F будет изображаться квадратной таблицей T, строки и столбцы которой упорядочены в соответствии с A.

Вектор, все элементы которого равны 0, будем обозначать символом $\vec{0}$. Вектор, все координаты которого кроме i-той равны 0, а i-тая координата равна 1, назовём единичным вектором e_i. Символом $\vec{A}$ в дальнейшем будем называть вектор, все координаты которого равны 1.

Над представляющими векторами может быть введён ряд операций. Сложение (+), вычитание (-) и умножение на число (·) определим обычным для векторов способом, т.е. как покоординатные сложение, вычитание и умножение на число.

Нетрудно заметить, что при выполнении этих действий могут получиться векторы, координаты которых отличаются от нулей и единиц. Такие векторы не соответствуют непосредственно подмножествам из L, и будут допускаться лишь на промежуточных этапах вычислений.

Множество векторов, на котором определены подобные операции, представляет собой аффинное пространство. В аффинном пространстве произвольный вектор x можно представить в виде суммы произведений единичных векторов на некоторые числа следующим образом:

$$x = a_1 \cdot e_1 + a_2 \cdot e_2 + \ldots + a_n \cdot e_n.$$

Своеобразие предлагаемой системы состоит в операции "скалярного произведения". Специальным символом оно обозначается не

будет. Скалярным произведением двух единичных векторов мы назовём число $f(e_i,e_j)$,стоящее на пересечении строки e_i и столбца e_j в таблице T .Скалярное произведение двух произвольных векторов x и y определяется так:

$$xy = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i \cdot b_j \cdot f(e_i e_j).$$

Здесь векторы x и y представляются их разложениями:

$$x = a_1 \cdot e_1 + a_2 \cdot e_2 + \ldots + a_n \cdot e_n$$
$$y = b_1 \cdot e_1 + b_2 \cdot e_2 + \ldots + b_n \cdot e_n.$$

Векторное сложение,как известно,коммутативно и ассоциативно; скалярное произведение дистрибутивно относительно сложения и вычитания и коммутативно в случае симметричности функции F. Дистрибутивность скалярного произведения относительно сложения и вычитания доказывается с помощью разложений по единичным векторам.Пусть,например,имеются следующие векторы:

$$a = a_1 \cdot e_1 + a_2 \cdot e_2 + \ldots + a_n \cdot e_n$$
$$b = b_1 \cdot e_1 + b_2 \cdot e_2 + \ldots + b_n \cdot e_n$$

$$c = c_1 \cdot e_1 + c_2 \cdot e_2 + \ldots + c_n \cdot e_n$$
$$a + b = (a_1 + b_1) \cdot e_1 + \ldots + (a_n + b_n) \cdot e_n$$

Доказывается справедливость равенства

$$(a + b)\,c = ac + bc.$$

Имеем: $ac = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i \cdot c_j \cdot f(e_i e_j)$; $bc = \sum_{i=1}^{n} \sum_{j=1}^{n} b_i \cdot c_j \cdot f(e_i e_j)$.

Суммируя ac и bc и объединяя слагаемые с одинаковыми сомножителями вида $c_j \cdot f(e_i e_j)$, получим выражение

$$\sum_{i=1}^{n} \sum_{j=1}^{n} (a_i + b_i) \cdot c_j \cdot f(e_i e_j),$$

равное,по определению,произведению a + b на c.

Если U и W - множества букв,и $U \subset W$,то символом $\subset$ мы будем соединять и векторы,представляющие U и W. Если при этом U - одноэлементное множество,то мы будем употреблять (не вполне корректно) символ $\in$.

Разбиение множества L на два класса мы будем обозначать буквой D с различными индексами, например, D_j - j-ое разбиение на два класса, D_{ji} - j-ое i-разбиение, D_{Hi} - i-наилучшее разбиение; обычным будет также обозначение разбиения указанием его классов, например: D = V,C = V,C или $D_j = k_{j1},k_{j2}$.

Функции качества будут обозначаться буквой Q с различными индексами; скалярное произведение вектора a самого на себя (скалярный квадрат) будет обозначаться a^2. Таким образом, функция качества алгоритма I будет записываться так:

$$Q = V^2 + C^2.$$

Простота этого выражения по сравнению с (B1)

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{n} f(l_i,l_j) + \sum_{p=1}^{n} \sum_{q=1}^{n} f(l_p,l_q)$$

очевидна.

Пусть у нас имеются два разбиения $D_i = k_1,k_2$ и $D_j = q_1,q_2$. Класс q_1 всегда можно получить из класса k_1 выбросив "лишние" и добавив "недостающие" буквы. Иначе говоря, выбрав вектор r должным образом, мы сможем записать

$$q_1 = k_1 + r;$$

то же относится и к вторым классам, только те элементы, которые были "лишними" в k_1 "недостают" в k_2 а те, которых "недоставало" в k_1 являются "лишними" в k_2. Поэтому можно записать:

$$q_2 = k_2 - r.$$

Разумеется, вектор r в общем случае содержит как положительные так и отрицательные координаты. Пару векторов r,-r мы назовём разностью разбиений D_j и D_i.

Разность $Q(D_j) - Q(D_i)$ мы назовём приращением качества для разбиения D_i на r,-r и будем обозначать через dQ или $dQ(D_i,r)$.

Приведём примеры использования введённой системы обозначений.

Пример I.

Задачу минимизации суммы частот сочетаний гласных с гласными и согласных с согласными можно записать следующим образом:

Найти такое $D_H = V_H,C_H$, что

$$V_H^2 + C_H^2 = \min_j (V_j^2 + C_j^2),$$

где V_j, C_j - произвольная классификация на два класса.

Пример 2

Докажем справедливость алгоритма I в случае регулярности таблицы частот.

Условие регулярности можно записать следующим образом:

$$V_{Hi}, C_{Hi} = V_{H(i-1)} + e, C_{H(i-1)} - e.$$

Здесь индексы Hi характеризуют i-наилучшее разбиение, e - буква, переносимая из большего класса (i-1) -наилучшего разбиения в меньший класс. В алгоритме I поиск минимума функции $V^2 + C^2$ заменяется максимизацией функции VC. Покажем справедливость этой замены:

$$V + C = A$$
$$(V + C)^2 = A^2$$
$$V^2 + 2 \cdot VC + C^2 = A^2$$
$$VC = \frac{A^2 - (V^2 + C^2)}{2}$$

Так как A^2 (сумма чисел во всей таблице частот) и 2 суть постоянные величины, то максимум функции VC обеспечивается минимумом $V^2 + C^2$.

Все квадраты векторов, представляющих одну букву, мы можем всегда считать равными 0, поскольку сумма их либо не входит в функцию качества (например, в VC), либо входит в неё для любого разбиения (например, в $V^2 + C^2$), и, следовательно, не влияет на поиск экстремума этих функций.

Функцию VC будем также обозначать через Q_1.

Введём важное понятие "тупикового разбиения". Разбиение назовём "тупиковым", если перенос любой буквы из большего класса в меньший не увеличивает значения Q_1.

Доказать алгоритм I в предположении регулярности таблицы частот означает следующее:

I) Показать, что предлагаемая процедура обеспечивает получение D_{Hi} если ранее получено $D_{H(i-1)}$ ($D_{H(i-1)}$ - не тупиковое разбиение);

2) Показать, что если D_{Hit} - i-наилучшее и, к тому же, тупиковое разбиение, то не существует разбиения D_{Hj} (где $j > i$), более хорошего - с точки зрения Q_1 - чем D_{Hit}.

Утверждение I.

Поскольку $Q_1(D_{Hi}) \geqslant Q_1(D_{ji})$, (где D_{ji} есть i-разбиение, не равное D_{Hi}), то

$$Q_1(D_{Hi}) - Q_1(D_{H(i-1)}) \geqslant Q_1(D_{ji}) - Q_1(D_{H(i-1)}) \qquad (B2)$$

Разбиения $D_{H(i-1)}, D_{Hi}, D_{ji}$ и соответствующие значения Q_1 можно записать следующим образом:

$D_{H(i-1)} = V, C$ $\qquad Q_1(D_{H(i-1)}) = VC$

$D_{Hi} = V + a, C - a$ $\qquad Q_1(D_{Hi}) = (V + a)(C - a) =$

$VC + aC - aV = VC + a(C - V)$

$D_{ji} = V + b, C - b$ $\qquad Q_1(D_{ji}) = VC + b(C - V).$

Подставляя эти выражения в (B2) получим:

$VC + a(C - V) - VC \geqslant VC + b(C - V) - VC,$

т.е. $a(C - V) \geqslant b(C - V),$

или иначе $aC - aV \geqslant bC - bV,$ $\qquad (B3)$

т.е. для получения D_{Hi} из $D_{H(i-1)}$ следует перенести из класса C в класс V букву a, для которой разность частот aC и aV максимальна.

Признаком того, что разбиение $D_{H(i-1)}$ – нетупиковое, является выполнение неравенства $aC - aV > 0$.

Нарушение этого неравенства означает, что даже наилучшее i-разбиение ухудшает качество $D_{H(i-1)}$.

Утверждение 2.

Пусть $D_{HiT} = V_T, C_T$. Разбиение D_{Hj} можно представить в виде $V_T + q, C_T - q$, где $q \subset C_T$ (q, в общем случае, не единичный вектор).

Допустим, что $V_T C_T < (V_T + q)(C_T - q)$,

т.е. $V_T C_T < V_T C_T - V_T q + C_T q - q^2; 0 < -V_T q + C_T q - q^2$,

или $C_T q - V_T q - q^2 > 0$, а значит $C_T q - V_T q > 0$.

Последнее означает, что найдётся по крайней мере один единичный вектор a[2], такой, что $C_T a - V_T a > 0$, т.е. букву a можно перенести в меньший класс D_{HiT}, что противоречит тупиковости D_{HiT}. Следовательно, $Q_1(D_{HiT}) \geqslant Q_1(D_{Hj})$.

---

2) $a \in q$

Пример 3.

Пусть дано разбиение $D = k_1, k_2$ и буква a ($a \in k_i$, $i = 1$ или $i = 2$.) Условие переноса для обводных переносов (см.стр.8) можно выразить следующим неравенством:

$$ak_{3-i} - ak_i > 0.$$

Пример 4.

Можно предложить функцию качества на множестве разбиений, которая выглядит не только отличной от Q (и Q_1), но и более правильной.

Пусть имеется разбиение $D = V, C$. Назовём однородной цепочкой отрезок текста, состоящий либо только из букв класса V, либо только из букв класса C. Однородную цепочку назовём максимальной, если добавление к ней справа или слева одной буквы превращает её в неоднородную.

Функцию качества Q_2 определим теперь как количество максимальных однородных цепочек, порождаемых данным разбиением. Очевидно, наилучшему разбиению должен соответствовать максимум Q_2. Данная функция кажется более правдоподобной, чем Q_1, поскольку здесь принимаются во внимание не только ближайшие, но и отдалённые окрестности букв. Тем не менее, Q_2 оказывается эквивалентной Q_1, т.е. их экстремум приходится на одинаковые разбиения.

Если, как обычно, мы будем считать, что за последней буквой текста идёт первая, то количества максимальных однородных цепочек обоих классов равны. Обозначим число таких цепочек, состоящих только из гласных через $Q_2(S_i^V)$.

Попробуем определить $Q_2(S_i^V)$, используя сведения о частоте гласных. Справедливо, что $Q_2(S_i^V) \leqslant VA$,

или $Q_2(S_i^V) = VA - R \quad (R \geqslant 0)$,

поскольку каждая цепочка S_i^V состоит, по меньшей мере, из одной гласной. Обозначим максимальную однородную цепочку гласных, содержащую n букв через S_{in}^V, а их количество через $Q_2(S_{in}^V)$. Имеет место выражение: $Q_2(S_1^V) = VA - Q_2(S_{12}^V) - 2Q_2(S_{13}^V) - \ldots$, поскольку при появлении максимальной однородной цепочки из двух гласных VA увеличивается на 2, а $Q_2(S_i^V)$ – на 1, и т.д. В то же время $Q_2(S_{12}^V) = V^2 - 2Q_2(S_{13}^V) - \ldots - (n-1)Q_2(S_{in}^V)$,

Поскольку при появлении максимальной однородной цепочки из двух букв $Q_2(s^V_{i2})$, так же, как и V^2 возрастает на 1, а при появлении максимальной однородной цепочки из трёх букв V^2 растёт на 2, а $Q_2(s^V_{i2})$ совсем не растёт, и т.д.

Подставим правую часть последнего равенства вместо $Q_2(s^V_{i2})$ предыдущего. Получим:

$$Q_2(s^V_i) = VA - V^2 + 2Q_2(s^V_{i3}) + \ldots + (n-1)Q_2(s^V_{in}) - 2Q_2(s^V_{i3}) - \ldots - (n-1)Q_2(s^V_{in}).$$

Можно видеть, что все слагаемые, начиная с третьего, уничтожаются:

$$Q_2(s^V_i) = VA - V^2.$$

Подобным образом получаем для $Q_2(s^C_i)$:

$$Q_2(s^C_i) = CA - C^2.$$

Полученные равенства складываем:

$$Q_2(s) = VA + CA - V^2 - C^2;$$

Далее:

$$Q_2(s) = (V + C)A - V^2 - C^2;$$

$$Q_2(s) = A^2 - V^2 - C^2; \quad Q_2(s) = A^2 - (V^2 + C^2).$$

Так как A^2 - величина постоянная, то максимум $Q_2(s)$ обеспечивается минимумом $V^2 + C^2$.

Алгоритм II.

Алгоритм II, как и излагаемый далее алгоритм III, предназначен для отыскания минимума функции $V^2 + C^2$ в случае нерегулярных таблиц частот. Алгоритм II служит для переборов всех локальных минимумов с последующим нахождением глобального.

Мы будем говорить, что для разбиения V,C функция $V^2 + C^2$ принимает локально-минимальное значение, если для всякого разбиения V_1, C_1 связанного с V,C соотношением

$$V_1, C_1 = V - a, C + a \quad \text{или}$$

$$V_1, C_1 = V + b, C - b \quad (a \in V, b \in C)$$

(такие разбиения называются соседними) справедливо, что $V_1^2 + C_1^2 \geqslant V^2 + C^2$. Разбиение V,C назовём допустимым.

Абсолютно минимальным значением функции $V^2 + C^2$ назовём такое значение $V_H^2 + C_H^2$ от разбиения V_H, C_H, что $V_H^2 + C_H^2 \leqslant V^2 + C^2$ для любого V,C.

Разумеется, абсолютно минимальное значение является и локально минимальным. Поэтому среди всех допустимых разбиений найдёт-

ся и наилучшее.

Признаками, определяющими локальную минимальность значения $V^2 + C^2$ можно считать соотношения: $aC - aV \geqslant 0$,

$$bV - bC \geqslant 0,$$

где $a \in V, b \in C$.

Покажем, что при этом $V_i^2 + C_i^2 \geqslant V^2 + C^2$ где V_i, C_i – соседнее к V, C разбиение. Пусть, например, $V_i, C_i = V + b, C - b$.

Предположим, что $(V + b)^2 + (C - b)^2 < V^2 + C^2$; раскроем скобки: $V^2 + 2 \cdot Vb + b^2 + C^2 - 2 \cdot Cb + b^2 < V^2 + C^2$; полагая, как обычно, $b^2 = 0$, имеем $Vb - Cb < 0$, что противоречит второму признаку. Так же доказывается, что $V_i^2 + C_i^2 \geqslant V^2 + C^2$ в случае, если $V_i, C_i = V - a, C + a$.

Вышеприведённые признаки можно считать признаками допустимости класса гласных и согласных. Вообще, назовём допустимым классом класс k_i (безразлично, гласных или согласных), если для всякого $a (a \in k_i)$ имеет место $a(A - k_i) - ak_i \geqslant 0$, или, раскрывая скобки, $aA - 2 \cdot ak_i \geqslant 0$.

В соответствии с этим, разбиение допустимо, если каждый класс его допустим. Докажем теорему:

Т.I. Если класс k_i не допустим, то любой класс k_j $(k_j \supset k_i)$ также не допустим.

Недопустимость класса k_i означает, что найдётся буква a, такая, что $aA - 2 \cdot ak_i < 0$. Из того, что $a \in k_i$ и $k_i \subset k_j$, следует, что $a \in k_j$. Проверим допустимость k_j по букве a:

$$aA - 2 \cdot ak_j;$$

Подставим вместо k_j сумму $k_i + R$, где $R = k_j - k_i$:

$$aA - 2 \cdot ak_i - 2 \cdot aR.$$

Однако, поскольку R есть сумма положительных единичных векторов (т.к. $k_j \supset k_i$), слагаемое $-2 \cdot aR$ либо отрицательно, либо равно 0. Значит $aA - 2 \cdot ak_j < aA - 2 \cdot ak_i$, а, следовательно и подавно $aA - 2 \cdot ak_j < 0$.

Алгоритм получения всех разбиений, дающих локальные минимумы, можно строить по следующему плану: сначала отыскиваются все допустимые классы, затем подбираются такие пары допустимых классов, которые в сумме дают A. Теорема Т.I позволяет сократить перебор.

66100

www.ruslang.ru

Вначале следует отыскивать допустимые классы, состоящие из двух элементов. Для этого просматриваются все пары элементов $a_i a_j$ (без различия порядка), т.е. если $|A| = n$, то $n(n-1):2$ пар. Для каждой пары необходимо проверить соблюдение двух неравенств: $a_i A - 2 \cdot a_i a_j \geqslant 0, a_j A - 2 \cdot a_i a_j \geqslant 0$.

Если хотя бы одно из них не соблюдается, то пара недопустима. Затем построим т.н. "потенциально доустимые тройки". Потенциально допустимые тройки суть классы из трёх элементов, такие, что любая пара, входящая в потенциально допустимую тройку, допустима. Потенциально допустимых троек, в общем случае, меньше, чем всех возможных троек; в этом проявляется смысл Т.I. Каждая из потенциально допустимых троек проверяется на допустимость, и недопустимые вычёркиваются из списка.

Аналогично, при получении списка потенциально допустимых m-ок требуется, чтобы они содержали лишь допустимые n-ки, где $n = m - 1$.

При условии $a_i^2 = 0$ все одноэлементные классы допустимы, т.к.

$$a_i A - 2 \cdot a_i^2 = a_i A, \; a_i A \geqslant 0.$$

Максимальная длина допустимых m-ок равна $n - 1$ ($A = n$). Далее выбираются все непересекающиеся пары допустимых классов, в сумме исчерпывающие алфавит; эти пары и образуют разбиения, для которых $V^2 + C^2$ локально минимальна. Определяя для каждого разбиения значение Q, находим (перебором) V_H, C_H.

Например, допустимые пары для элементов таблицы на стр.6 суть: a,b; a,d; a,e; b,c; c,e; c,d; d,e.
Потенциально допустимые тройки: a,d,e; c,d,e.
Обе они допустимы. Допустимых четвёрок нет.

Так как $A = 5$, а допустимых четвёрок нет, то искомые разбиения содержат по одному классу мощности 2, и одному - 3. Так как классов мощности 3 всего 2, то искомых разбиений не более двух. Оба дополнения, b,c и a,b допустимы. Следовательно, мы обнаружили два локальных минимума: для разбиения : b,c ; a,d,e и a,b ; c,d,e со значениями 26 и 28 соответственно.

Алгоритм III.

Алгоритм III сходен с алгоритмом I;в нём не перебираются все локальные экстремумы.[5)]

Как в алгоритме I,минимизацию $V^2 + C^2$ заменим максимизацией VC. Подставив вместо C равное ему выражение A - V ,выразим Q_1 в форме $AV - V^2$. Пусть $V = a_1 + a_2 + a_3 + \ldots + a_p$. Подстановкой в $AV - V^2$ правой части этого выражения и приравнивая нулю слагаемые вида a_1^2 ,получим:

$$AV - V^2 = a_1A + a_2A + a_3A + \ldots + a_pA - 2\cdot a_1a_2 - 2\cdot a_2a_3 - \ldots - 2\cdot a_1a_p - 2\cdot a_2a_3 - \ldots - 2\cdot a_2a_p - \ldots\ldots\ldots\ldots\ldots - 2\cdot a_{p-1}a_p$$

Правая часть этого равенства расположена в форме треугольной таблицы.Составим слагаемые g_1 ,соответствующие столбцам этой таблицы:

$$g_1 = a_1A$$
$$g_2 = a_2A - 2\cdot a_2a_1$$
$$g_3 = a_3A - 2\cdot a_3a_1 - 2\cdot a_3a_2$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$g_p = a_pA - 2\cdot a_pa_1 - 2\cdot a_pa_2 - \ldots - 2\cdot a_pa_{p-1}$$

Здесь скалярные произведения $a_1A,\ldots,a_pA$ размещены в первом столбце,скалярные произведения $-2\cdot a_1a_2-\ldots-2\cdot a_1a_p$ во втором,и т.д.

Переставляя элементы в выражении $V = a_1 + a_2 + \ldots + a_p$ и,соответственно,в других выражениях,содержащих $a_1,\ldots,a_p$, мы можем получить другое разбиение на слагаемые указанного вида. Тем самым,разбиение на слагаемые $g_1,\ldots,g_p$, которое мы будем называть путём к классу V есть функция от перестановки букв $a_1,\ldots,a_p$. Буквы $a_1,\ldots a_p$ будем называть образующими пути $g_1,\ldots,g_p$.

Назовём положительным путём к классу k путь $g_1,\ldots,g_p$

[5)] Данный алгоритм укладывается в схему т.н."метода ветвей и границ".

в котором каждый элемент ε_i пути не меньше нуля. Имеет место следующая теорема:

Т.2. Любой путь к производному допустимому классу есть положительный путь.

Будем употреблять вместо V и C эквивалентные выражения k и A-k. Если разбиение, состоящее из классов k и A - k образует локальный минимум Q, то для каждой буквы a из k справедливо соотношение $aA - 2 \cdot ak \geqslant 0$ по признаку локального минимума. Это соотношение представляет собой выражение последнего элемента ε_p такого пути к классу k, который во всех скалярных произведениях содержит a (данный путь "кончается" на a). Если буква a есть образующая непоследнего элемента данного пути, например, ε_i (где $i < p$), то ε_p меньше или равно ε_i, поскольку ε_i содержит меньше отрицательных слагаемых; поэтому ε_i и подавно больше или равно нулю.

Назовём путь монотонным, если $\varepsilon_i \geqslant \varepsilon_{i+1}$ для любой пары $\varepsilon_i, \varepsilon_{i+1}$ элементов пути. Докажем теорему:

Т.3. К произвольному допустимому классу существует положительный монотонный путь.

Как доказано выше, любой путь к такому классу есть положительный путь. Докажем, что среди этих путей найдётся и монотонный.

Построим этот путь так: выберем в качестве ε_1 такое a_1A, что $a_1A \geqslant a_iA$ $(a_i \in k - a_1)$, в качестве ε_2 такое $a_2A - 2 \cdot a_1a_2$, что $a_2A - 2 \cdot a_1a_2 \geqslant a_iA - 2 \cdot a_1a_i$ $(a_i \in k - a_1 - a_2)$, в качестве ε_3 такое $a_3A - 2\,a_3a_1 - 2\,a_3a_2$, что $a_3A - 2 \cdot a_3a_1 - 2 \cdot a_3a_2 \geqslant a_iA - 2 \cdot a_ia_1 - 2 \cdot a_ia_2$ $(a_i \in k - a_1 - a_2 - a_3)$, и т.д. Из того, что $a_1A \geqslant a_2A$ следует, что и подавно $a_1A \geqslant a_2A - 2 \cdot a_1a_2$ и т.д.

Эту процедуру, доказывающую существование монотонного пути, не следует смешивать с алгоритмом I. В алгоритме I требуется максимальность выражений, содержащих $a_1, a_2, \ldots$ и т.д. среди всех выражений вида $a_iA - 2\,a_ia_1 - \ldots - 2\,a_ia_n$, (где n - число букв в алфавите); в настоящей же процедуре a_i входят в множество k; класс k предполагается известным заранее.

Назовём положительный монотонный путь тупиковым, если $P_T = g_1,\ldots,g_p$ и нельзя построить g_{p+1} ,такое,что $g_{p+1} > 0$. Докажем теорему:

Т.4.К каждому классу допустимого разбиения $D = k_1,k_2$ существует тупиковый путь.

Положительный монотонный путь ккаждому из этих классов существует в силу Т.3.Этот же путь является и тупиковым.Пусть положительный монотонный путь к k_1 есть $P_T^1 = g_1,\ldots.g_p$. Попробуем построить g_{p+1} от образующей a ($a \in k_2$):

$$g_{p+1} = aA - 2\cdot a\, k_1;$$

но a допустима в k_2 ,т.е. $aA - 2\cdot ak_2 \geqslant 0, (k_2 = A - k_1)$, следовательно $aA - 2\cdot a(A - k_1) \geqslant 0$; $-aA + 2\cdot ak_1 \geqslant 0$ и $aA - 2\cdot ak_1 \leqslant 0$, т.е. $g_{p+1} \leqslant 0$.

Невозможность построения такого g_{p+1} ,что $g_{p+1} > 0$ назовём сильным условием тупиковости.

Итак,отыскание допустимых разбиений мы можем заменить отысканием тупиковых путей.Последующие выкладки позволяют ещё сильнее сузить множество перебираемых путей.

Назовём длиной пути количество t элементов или образующих пути.Очевидно,что нет смысла рассматривать пути,длина которых превышает $[\frac{n}{2}]$,где $[\frac{n}{2}]$ есть целая часть от деления количества букв в алфавите на два,потому что всегда можно отыскивать путь к меньшему классу (который в крайнем случае равен большему).

Если уже найдено некоторое значение Q_1 от разбиения D ,т.е. $Q_1(D)$, то дальнейший перебор можно сократить. В самом деле, имеет смысл искать лишь такие пути,сумма элементов которых превышает $Q_1(D)$.

Пусть,например,имеется такой путь P ,($P = g_1(a_1),g_2(a_2),\ldots$ $\ldots,g_p(a_p)$), что $g_1(a_1) + g_2(a_2) +\ldots+ g_p(a_p) > Q_1(D)$. Элементы такого пути обладают некоторыми полезными свойствами. Имеем: $g_{m+1}(a_{m+1}) +\ldots.+ g_p(a_p) > Q_1(D) - (g_1(a_1) +\ldots+g_m(a_m))$ (здесь $1 \leqslant m \leqslant p$).Пусть $g_q(a_q)$ - самое большое слагаемое среди слагаемых левой части.Тогда $(p - m)\cdot g(a) > Q(D) -$ $(g_1(a_1)+\ldots.)$, где $p-m$ есть количество слагаемых в левой части предыдущего неравенства.Далее:

$$g_q(a_q) > \frac{Q_1(D) - (g_1(a_1) + \ldots + g_m(a_m))}{p - m} \qquad (B4)$$

Вместо $p - m$ мы можем подставить число $w = [\frac{n}{2}] - m$ (целую часть от деления $\frac{n}{2}$ минус m),где n - мощность алфавита. Это число есть количество элементов в самом длинном пути к меньшему классу минус m ;оно всегда больше или равно $p - m$. Если данный путь - монотонный,то в качестве следующего элемента следует выбирать лишь элементы,удовлетворяющие (B4) .В частности,для первых элементов путей неравенство имеет вид:

$$g_1(a_1) > \frac{Q_1(D)}{[\frac{n}{2}]}$$

Возможен случай,когда правая часть (B4) меньше нуля.Тогда (B4) теряет смысл для выбора следующего элемента пути.Кроме того,при $w = 0$ (т.е.в том случае,когда объём класса k_1 уже достиг $[\frac{n}{2}]$),выражение (B4) становится неопределённым.Поэтому очередной элемент пути следует выбирать среди элементов,удовлетворяющих трём неравенствам:

$$w > 0;\ g_q(a_q) > 0;$$

$$g_q(a_q) > \frac{Q_1(D) - (g_1(a_1) + \ldots + g_m(a_m))}{w} \qquad (B5)$$

Путь,каждый элемент которого удовлетворяет (B5) назовём нормальным,а нормальный путь,для которого нельзя построить ещё один элемент,удовлетворяющий (B5) ,назовём условно-тупиковым.

Существенное сокращение перебора достигается за счёт того, что можно назвать "эквивалентностью путей".Поскольку нас интересуют классы,являющиеся неупорядоченными множествами,порядок выбора элементов класса не имеет значения.Естественно поэтому считать эквивалентными любые два пути,ведущие к одному и тому же классу.

Будем говорить,что путь P_j содержит путь P_i ,если множество образующих первых элементов P_j совпадает с множеством образующих пути P_i .Имеет место теорема:

Т5.Если P_i и P_j эквивалентны(ведут к одному и тому же классу) и все условно-тупиковые пути,содержащие P_i рассмотрены в процессе перебора,то условно-тупиковые пути,содержащие P_j можно не рассматривать.

Пусть некоторый условно тупиковый путь P_f содержит P_j. Докажем, что существует эквивалентный ему условно тупиковый путь P_f^*, содержащий P_i. Длину пути P_i (а также P_j) будем считать равной t.

Рассмотрим элемент g_{t+1}, входящий в P_f. Его можно выразить так: $g_{t+1} = a_{t+1}A - 2 \cdot a_{t+1}k$. Этот элемент должен удовлетворять условиям (B5): $g_{t+1} > 0$; $[\frac{n}{2}] - t > 0$;

$$g_{t+1} > \frac{Q_1(D) - \sum_{v=1} g_v}{[n:2] - t}$$

Рассмотрим теперь некоторый элемент g'_{t+1}, входящий в некоторый условно-тупиковый путь, содержащий P_i. Его можно выразить аналогично: $g'_{t+1} = a'_{t+1}A - 2\,a'_{t+1}k$. Он также должен удовлетворять (B5): $g'_{t+1} > 0$; $[n:2] - t > 0$

$$g'_{t+1} > \frac{Q_1(D) - \sum_{u=1} g'_u}{[n:2] - t}$$

Образующие a'_{t+1} и a_{t+1} входят в одно и тоже множество суммы Σg_v и $\Sigma g'_u$ равны по определению пути. Поэтому вместо a'_{t+1} мы можем подставить a_{t+1}.

Таким образом, существует путь P'_s, входящий в некоторый условно-тупиковый путь P_f^* и ведущий к классу $k + a_{t+1}$, а также путь P_s ($P_s = P_j + g_{t+1}$), причём P'_s эквивалентен P_s. Построение можно продолжить, заменив P_i и P_j соответственно на P'_s и P_s. Доказательство обрывается на g_{t+x} и g'_{t+x}, для которых не выполняется (B5).

Т6. Если есть произвольный путь P_j, и его множество образующих $M(P_j)$ есть собственное подмножество образующих $M(P)$ некоторого условно-тупикового пути P, ведущего к допустимому классу,[4] то для пути P_j можно построить ещё один элемент, удовлетворяющий (B5).

Пусть путь P построен по образующим $a_1, \ldots, a_u$ и сумма его элементов $g_1(a_1) + \ldots + g_m(a_m) + \ldots + g_u(a_u)$ равна x

4) Условно-тупиковый путь (и просто тупиковый путь) может и не вести к допустимому классу, см. пример на стр. 6.

$(x > q)$,где q - значение Q_1 для некоторого разбиения. Путь P_j построен по образующим $b_1,\ldots,b_m$,где $m < u$ и $b_1,\ldots,b_m$ есть собственное подмножество образующих пути P .Пусть $c_{m+1},\ldots,c_u$ суть остальные образующие пути P ,расположенные так,что $g_r(c_r) \geq g_{r+1}(c_{r+1})$ $(m < r < u)$.

Из того,что P ведёт к допустимому классу M(P) следует,что любой элемент вида $g_{m+1}(c_s)$ $(m < s \leq u)$ больше нуля.(Элемент вида $g_{m+1}(c_s)$ есть продолжение пути P_j ;его образующей является $c_s, c_s \in M(P) \setminus M(P_j)$.

В самом деле,из того,что M(P) есть допустимый класс,следует,что $c_sA - 2\cdot c_sM(P) \geq 0$. Но $g_{m+1}(c_s) = c_sA - 2\cdot c_sM(P)+2c_sR$ и не меньше $c_sA - 2\cdot c_sM(P)$, за счёт неотрицательного слагаемого $2\cdot c_sR$,где $R = M(P) \setminus M(P_j)$.

Теперь докажем,что среди $\{ g_{m+1}(c_s)\}$ найдётся элемент,удовлетворяющий (B5) .По определению пути имеем:

$$g_1(a_1) + \ldots + g_m(a_m)+\ldots+g_u(a_u) =$$
$$= g_1(b_1) + \ldots + g_m(b_m) + g_{m+1}(c_{m+1}) + \ldots + g_u(c_u) > q$$

Отсюда получаем:

$$g_{m+1}(c_{m+1}) + \ldots + g_u(c_u) > q - (g_1(b_1) + \ldots + g_m(b_r)).$$

Как обычно,если $g_t(c_t) = \max(g_r(c_r))$ $(m < r \leq u)$, то $(u - m)\, g_t(c_t) > q - (g_1(b_1) + \ldots + g_m(b_m))$,

$$w\cdot g_t(c_t) > q - (g_1(b_1) +\ldots+ g_m(b_m)) \quad (w = [n:2] - m);$$

$$g_t(c_t) > \frac{q - (g_1(b_1) + \ldots + g_m(b_m))}{w}$$

Неравенство нулю числа w вытекает из того,что объём P больше объёма P_j .Таким образом,для элемента $g_t(c_t)$ выполняются условия (B5).

Следствие:если для пути P_j нет продолжений и множество $M(P_j)$ его образующих входит в множество образующих пути P ,то путь P не является условно тупиковым и ведущим к допустимому классу. В противном случае для P_j нашлось бы продолжение,что противоречит Т6.

Как следует из теоремы Т3 можно ввести ещё один принцип сокращения перебора,а именно,не проверять пути,у которых после-

дующий элемент больше предыдущего. Мы покажем, что проверку на этот признак можно не производить, если, особым образом, уточнить порядок перебора.

Это уточнение сводится к следующему: пусть рассмотрены все монотонные пути вида $P, g_{t+1}(a_x)$ (т.е. отличающиеся друг от друга последними элементами). Расположим эти пути в порядке убывания величины элементов $g_{t+1}(a_x)$:

$$P, g_{t+1}(a_1);$$
$$P, g_{t+1}(a_2);$$
$$\dots\dots\dots\dots\dots$$
$$P, g_{t+1}(a_z).$$

Образующие элементов вида $g_{t+1}(a_x)$ назовём кончающими для пути P. Имеет место теорема:

Т7. Для обеспечения монотонности достаточно рассматривать продолжения пути $P, g_{t+1}(a_k)$ $(1 < k \leqslant z)$ не ранее проверки всех путей, начинающихся с $P, g_{t+1}(a_p)$ $(1 \leqslant p < k)$.

Обозначим множество образующих пути P через S, а множество образующих последних элементов $a_1, \dots, a_z$ через T. Пусть $1 \notin S + T$.

Допустим, продолжение пути $P, g_{t+1}(a_p)$ есть $g_{t+2}(1)$; имеет место

$$a_p A - 2\, a_p S > \frac{Q(D) - \sum_{u}^{t} g_u}{[n:2] - t} \quad \text{и} \quad 1A - 2 \cdot 1S \leqslant \frac{Q(D) - \sum_{u}^{t} g_u}{[n:2] - t},$$

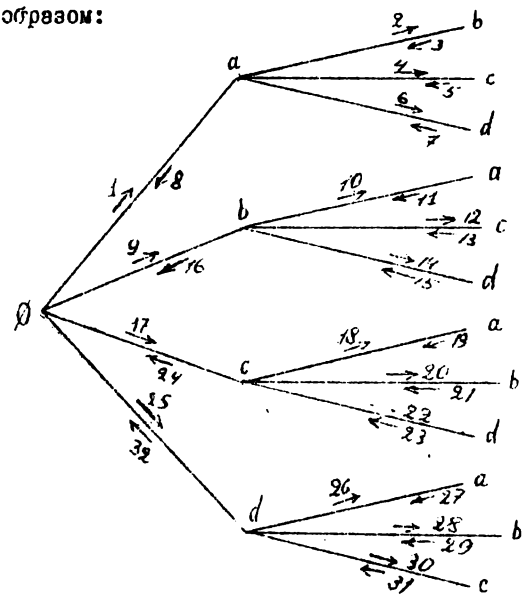
а следовательно $g_{t+1}(1) < g_{t+1}(a_p)$. Но $g_{t+2}(1) \leqslant g_{t+1}(1)$ поскольку $g_{t+2}(1) = 1A - 2 \cdot 1(S + a_p)$, т.е. $g_{t+2}(1) = g_{t+1}(1) - 2 \cdot 1 a_p$ и, значит, $g_{t+2}(1) \leqslant g_{t+1}(a_p)$. Поэтому в случае монотонности пути $P, g_{t+1}(a_p)$ монотонность пути $P, g_{t+1}(a_p), g_{t+2}(1)$ обеспечена. Далее рассмотрим пути вида

$$P, g_{t+1}(a_p), g_{t+2}(a_k).$$

По условию, $g_{t+1}(a_k) \leqslant g_{t+1}(a_p)$, но $g_{t+2}(a_k) \leqslant g_{t+1}(a_k)$ подобно только что рассмотренному случаю; следовательно $g_{t+2}(a_k) \leqslant g_{t+1}(a_p)$, что завершает доказательство.

Доказанные теоремы позволяют построить алгоритм,излагаемый ниже.Этот алгоритм перебирает монотонные условно-тупиковые пути, среди которых,как мы знаем,всегда найдётся путь,ведущий к меньшему классу наилучшего разбиения.Теорема Т.5 и следствие из Т.6 позволяют этот перебор сократить.

Множество путей удобно представлять деревом,вершины которого помечены буквами из A,ветви изображают цепочки образующих путей, причем общее начало двух путей изображается одной ветвью.Все ветви можно считать исходящими из "пустой" вершины $\emptyset$.Если i – тые элементы некоторых двух путей различны,то даже если i + k – тые элементы совпадают,ветви,соответствующие этим путям мы будем считать расходящимися по крайней мере с i– тых вершин. Длина ветви не превышает половину числа букв в алфавите. Например,для алфавита a,b,c,d дерево путей выглядит следующим образом:



Стрелки с цифрами поясняют порядок полного перебора путей. Образующие a,b,c и d считаются кончающими для пути (см.Т.7) и упорядочены от a к d по убыванию величины соответствующих элементов,сверху вниз на схеме;аналогично упорядочены тройки b,c,d; a,c,d; a,b,d; a,b,c.

В предлагаемом алгоритме осуществляется неполный перебор путей.Например,могут не рассматриваться все пути,начинающиеся с $\emptyset$,b, если элемент вида g(b) не удовлетворяет (B5) .Это будет соответствовать на схеме тому,что осуществив переход по стрелке 9 мы вернёмся в исходную вершину по стрелке I6.

Возврат к вершине,более близкой к корню $\emptyset$ происходит ещё в двух случаях:если множество образующих полученного пути P_i совпадает с множеством образующих некоторого пути P ,рассмотренного ранее (Т.5),или содержит это множество (Т.6).Монотонность путей обеспечивается порядком перебора (Т.7).При получении условно-тупикового пути,для которого значение Q_1 выше некоторого полученного ранее,в выражения (B5) подставляется это новое значение. Тем самым,множество перебираемых путей суживается.

Ниже мы приводим текст алгоритма.
Предварительно объясним применяемые обозначения.

Алгоритм строит список т.наз. "проверочных выражений" x_i = ПВ. Каждое x_i есть пара кортежей u_i, w_i, где u_i есть кортеж образующих некоторого пути,а w_i -кортеж возможных продолжений для этого пути.Буквы кортежа w_i расположены по убыванию величин, этих продолжений.Как u_i так и w_i могут быть пустыми.Это обстоятельство отмечается знаком $\emptyset$,ставящимся вместо пустого кортежа.Пустота w_i означает,что возможных продолжений либо нет,либо они все рассмотрены.

К кортежам справа и слева могут присоединяться или отниматься буквы.Уничтожение s букв слева у кортежа w_i будет обозначаться так: w_i - le(s); w_i - re(s).
Начальную букву некоторого кортежа w_i обозначим через $b(w_i)$ Поэтому вместо слов:добавить к u_i справа начальную букву из w_p мы будем писать: $u_i := u_i + re(b(w_p))$.

Проверочные выражения не остаются неизменными.Новые значения

проверочных выражений, а так же и прочих переменных вводятся символом := заимствованном из Алгола.

Текст алгоритма. [9]

I. Алгоритмом I найти некоторое разбиение $D = k_1, k_2$. [6]

2. Для пути P с кортежем образующих u_i найти все пути вида $P, g(l_j)$, где $g(l_j)$ удовлетворяют соотношениям

$[ni2] - d > 0$; $g(l_j) > 0$;

$$g(l_j) > \frac{Q_1(k_1,k_2) - \sum_{s=1}^{d} g_s(P)}{[ni2] - d}$$

Образующие l_j полученных элементов $g(l_j)$ расположить в порядке убывания $g(l_j)$, для равных $g(l_j)$ - произвольно; полученный кортеж есть $w_i : x_i$ $(x_i = u_i, w_i)$ вписать в список ПВ.

3. Пусто ли w_i ?да - делать 7.

4. $u_{i+1} := u_i + re(b(w_i))$.

5. Входит ли множество букв некоторого непустого кортежа u_t такого, что $w_t = \emptyset$ в множество букв из u_{i+1} ?да - делать I5.

6. $i := i + 1$, делать 2.

7. Если $Q_1(k_1,k_2) - \sum_{s=1}^{d} g_s(P) < 0$,

то считать, что k_1 равно множеству образующих из $u_i, i=1,$ $k_2 := A \setminus k_1$,

8. Найти x_p такое, что $u_p = u_i - re(a)$.

9. $u_p := u_p$; $w_p := w_p - le(1)$.

I0. Пусто ли w_p ?нет - делать I2.

II. Пусто ли u_p ?да - делать I6, нет - I4.

I2. $u_{i+1} := u_p + re(b(w_p))$.

I3. Входит ли множество букв некоторого непустого u_t, такого, что $w_t =$ в множество букв из u_{i+1} ?да - делать 9, нет - 6.

I4. $a := a + 1$, делать 8.

---

9) Исходные значения некоторых переменных:

$i = 0$; $x_0 = \emptyset, \emptyset$; $u_0 = \emptyset$; $w_0 = \emptyset$; $d = |k_1|$.

6) Этот пункт можно опустить, приняв $k_1 = \emptyset$ и $k_2 = A \setminus k_1$

I5. $w_1 := w_1 - 1e(1)$; Пусто ли w_i?
нет - делать 4,да - s := 1, делать 8.
I6.Конец.Ответ - текущее значение k_1, k_2.

В данной процедуре блок 2 определяет возможные продолжения для некоторого пути с образующими из кортежа u_1, l_j .Эти продолжения упорядочиваются в соответствии с Т.7.

Блок 4 образует новый путь путём прибавления самого большого продолжения к элементам пути P (это регистрируется прибавлением новой буквы к кортежу образующих).То же,после отхода,делает блок I2.

Блок 8 производит отход на нужное число букв (s),чтобы начать проверку следующих по величине продолжений.Блок 5 проверяет путь с помощью критериев,вытекающих из Т.5 и Т.6 (следствие), блок I3 делает то же самое,если данный путь образован сразу после отхода.

При совершении отхода назад к x_p, то,что предыдущее продолжение рассмотрено,отмечается его вычёркиванием в блоке 9.Вычёркивание продолжения,избыточного,с точки зрения Т.5 и следствия из Т.6 осуществляется блоком I5,после отхода - блоком 9.

Блок 3 указывает,когда начинать отход.Блок 7 регистрирует новое значение k_1, k_2. Блоки I0 и II определяют момент окончания процедуры,который наступает,когда рассмотрены все продолжения для x_0 ; если вычисления не закончены,то блок I0 отсылает к формированию нового пути,а блок II - к дальнейшему отходу.

В качестве примера приведём вычисления по таблице на стр.6. Символ H обозначает неопределённость значения.

+ + +

Автор глубоко благодарен А.Е.Гладкому за ценные советы.

Команды	q_1	k_1, k_2	i	s	$\frac{q_1(D) - g(P)}{[n;2] - d}$	$[n;2] - d$	ПВ	$u_1; w_1$	$u_b; w_b$	u_{i+1}
1	63	a,b;c,d,e	0	Н	36,5	2	x_0:Ø;Ø	Ø;Ø	Н	Н
2							x_0:Ø;a,c,b	Ø;a,c,b		
3		выполнить					команду	№4		
4										a
5		выполнить					команду	№6		
6			1					a;H		H
2					12	1	x_0:Ø;a,c,b x_1:a;Ø	a;Ø		
3		выполнить					команду	№7		
7				1						
8									x_0:Ø;a,c,b	
9							x_0:Ø;c,b x_1:a;Ø		x_0:Ø;c,b	
10		выполнить					команду	№12		
12										c
13		выполнить					команду	№6		

Команды	q_1	k_1, k_2	i	s	$\frac{g_1(D) - g(P)}{[h:2] - d}$	[h:2] - d	ШВ	$u_1 ; w_1$	$u_p ; w_p$	u_{i+1}
6			2					c;H		H
2					27	1	x_0:Ø;c,b x_1:a;Ø x_2:c;b	c;b		
3		выполнять команду №4								
4										c,b
5		выполнять команду №6								
6			3					c,b;H		H
2					H	0	x_0:Ø;c,b x_1:a;Ø x_2:c;b x_3:c,b;Ø	c,b;Ø		
3		выполнять команду №7								
7	64	c,b;a,d,e		1						
8									x_2:c;b	

Команда	Q_1	k_1, k_2	i	s	$\frac{Q_1(D) - g(P)}{[n:2] - d}$	$[n:2] - d$	ПВ	$u_i; w_i$	$u_p; w_p$	u_{i+1}
9							x_0:Ø;c,b x_1:a;Ø x_2:c;Ø x_3:c,b;Ø		x_2:c;Ø	
10		в ы п о	л н	я	т ь	к	с м а н д у	№II		
11		в ы п о	л н	я	т ь	к	о м а н д у	№I4		-
14				2						
8									x_0:Ø;c,b	
9							x_0:Ø;b x_1:a;Ø x_2:c;Ø x_3:c,b;Ø		x_0:Ø;b	
10		в ы п о	л н	я	т ь	к	о м а н д у	№I2		
12										b
13		в ы п о	л н	я	т ь	к	о м а н д у	№6		
6			4					b;H		H

Команды	φ_1	k_1, k_2	i	s	$\frac{\varphi_1(D) - g(P)}{[a:2]} - d$	$[a:2] - d$	ПВ	$u_i; w_i$	$u_p; w_p$	u_{i+1}
2					31	1	x_0:Ø;b x_1:a;Ø x_2:c;Ø x_3:c,b;Ø x_4:b;Ø	b;Ø		
3		в ы п о л	н	я	т ь к о	м	а н д у	№7		
7				1						
8									x_0:Ø;b	
9							x_0:Ø;Ø x_1:a;Ø x_2:c;Ø x_3:c,b;Ø x_4:b;Ø		x_0:Ø;Ø	
10		в ы п о л	н	я	т ь к о	м	а н д у	№II		
11		в ы п о л	н	я	т ь к о	м	а н д у	№I6		
16		к о н е ц								

Т-19141 от 21.12.70 100 экз. 2,25 п.л. Зак. 3

Тип. МГПИИЯ

-30 R