S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru

Том 15, стр. 1245–1259 (2018) УДК 519.165
DOI 10.17377/semi.2018.15.101 MSC 68R10

МЕТРИКИ НА РЕШЕТКЕ МУЛЬТИРУБРИК РУБРИКАТОРНОГО ДЕРЕВА

Н.А. ГАЙДАМАКИН, В.А. БАРАНСКИЙ

Abstract. The order Hamming distance on multirubric lattice of a weighed rubricator tree is introduced. Rubricator trees are useful for representation of hierarchical thematic classifiers in the field of organization and search for textual information. The rubricator tree is a root tree where any inner vertex has at least two sons. Any multirubric of a rubricator tree is an antichain of vertices not containing a complete set of sons of any vertex of the tree. The set of all multirubrics of a rubricator tree forms the lattice with the natural dominance relation. Based on the dominance relation, and lattice operations in the multirubric lattice are constructed effective algorithms for calculation the order Hamming distance for the case of an additively weighed rubricator tree. Distances on the multirubrics lattice are suggested to be applied for thematically indexed text information search mechanism as a measure of relevance.

Keywords: rubricator tree, multirubric, lattice, metrics, distance, algorithm, Hamming distance.

Введение

Иерархические тематические классификаторы (рубрикаторы) широко используются при организации хранения и поиска текстовой (документальной) информации, в частности, в библиотечной сфере. В [1] на корневых деревьях, которые соответствуют иерархическим тематическим классификаторам, введено понятие *мультирубрики* как антицепи, не содержащей полного набора

Gaydamakin, N.A., Baransky, V.A., Metrics on the multirubric lattice of a rubricator tree.

© 2018 Гайдамакин Н.А., Баранский В.А.

Поступила 20 сентября 2018 г., опубликована 23 октября 2018 г.

сыновей никакой вершины дерева, построена решетка мультирубрик и доказано, что она изоморфна булевой решетке всех подмножеств множества листовых вершин.

Мультирубрики являются результатами индексирования тематического содержания текстовой информации и, соответственно, на основе мультирубрик организуется размещение текстов (книг, документов, файлов). В свою очередь, пользователи свои поисковые потребности в простейшем случае выражают посредством мультирубрик (информация по каким рубрикам необходима). Результат поиска считается релевантным запросу, если мультирубрика запроса соответствует мультирубрикам найденных текстов. В связи с этим теоретический и практический интерес представляют метрики (расстояния) на множестве мультирубрик.

Будем использовать терминологию для графов и решеток в соответствии с [2] и [3]. Под графом будем понимать неориентированный граф без петель и кратных ребер, а под простой цепью — маршрут в графе, в котором не повторяются вершины и ребра.

Пусть $\boldsymbol{T}$ — *корневое дерево* с корнем r, т. е. дерево с выделенной вершиной r, называемой корнем. Рассмотрим на $\boldsymbol{T}$ естественный строгий частичный порядок $<$ такой, что для $x, y \in \boldsymbol{T}$ условие $x < y$ выполняется в том и только в том случае, когда $x \neq y$ и y принадлежит простой цепи с началом в x и концом в r. Если $x < y$, то будем говорить, что y является предком вершины x, а x — *потомком* вершины y. Как обычно, будем писать $x \leq y$, если $x = y$ или $x < y$. Если для вершин x и y оба условия $x \leq y$ и $y \leq x$ не выполняются, то будем говорить, что вершины x и y *не сравнимы*. Если $x < y$ и не существует такого z, что $x < z < y$, то будем говорить, что y является *отцом вершины* x, а x — *сыном* вершины y, будем писать $y = father(x)$. Заметим, что корень r дерева $\boldsymbol{T}$ является единственной вершиной, которая не имеет отца. *Антицепью* в дереве $\boldsymbol{T}$ будем называть любое подмножество попарно несравнимых элементов из $\boldsymbol{T}$. Поскольку корень r сравним с любой вершиной дерева $\boldsymbol{T}$, любая не одноэлементная антицепь не содержит корня r, т. е. r принадлежит точно одной антицепи, состоящей лишь из одного этого корня. Вершины дерева $\boldsymbol{T}$, не имеющие сыновей, будем называть *листьями*. Множество всех листьев дерева $\boldsymbol{T}$ будем обозначать через $\boldsymbol{L}(\boldsymbol{T})$ или, кратко, через $\boldsymbol{L}$. Вершины, не являющиеся листьями, будем называть *внутренними вершинами* дерева.

Непустое семейство $\boldsymbol{X}$ вершин дерева $\boldsymbol{T}$ будем называть *кланом*, если оно состоит из всех сыновей некоторой внутренней вершины этого дерева, которую будем обозначать через $father(\boldsymbol{X})$ и называть *отцом клана* $\boldsymbol{X}$. Через $Clan(x)$ будем обозначать клан, отцом которого является вершина x дерева $\boldsymbol{T}$.

Упоминавшиеся выше рубрикаторы (иерархические тематические классификаторы) строятся по таксономическому принципу, предполагающему в т. ч. разделение целого минимум на две части. В этом контексте корневое дерево $\boldsymbol{T}$ с корнем r будем называть *рубрикаторным деревом*, если любая его внутренняя вершина имеет не менее двух сыновей. Далее под $\boldsymbol{T}$ мы всюду будем понимать рубрикаторное дерево с корнем r.

В [1] на рубрикаторном дереве введено понятие *мультирубрики* $\boldsymbol{M}$ как антицепи дерева $\boldsymbol{T}$, не содержащей кланов ни одной вершины дерева $\boldsymbol{T}$. Отметим,

что по определению пустое множество $\varnothing$ является мультирубрикой. Кроме того, любая вершина x дерева $\boldsymbol{T}$ образует одноэлементную мультирубрику, которую мы будем обозначать также через x.

В [1] показано, что любая мультирубрика $\boldsymbol{M}$ совпадает с множеством всех максимальных вершин порождаемого ею *порядкового идеала* $(\boldsymbol{M}]$ в $\boldsymbol{T}$, который к тому же является *рубрикаторным идеалом*, т. е. идеалом, замкнутым относительно взятия отцов кланов, содержащихся в нем.

На множестве всех мультирубрик $MRL(\boldsymbol{T})$ определим (см. [1]) отношение частичного порядка $\leq$, полагая $\boldsymbol{M}_1 \leq \boldsymbol{M}_2$ для произвольных мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ в том и только в том случае, когда $(\boldsymbol{M}_1]$ содержится в $(\boldsymbol{M}_2]$ Ясно, что условие $\boldsymbol{M}_1 \leq \boldsymbol{M}_2$ для мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ выполняется тогда и только тогда, когда для любой вершины x_1 из $\boldsymbol{M}_1$ существует вершина x_2 из $\boldsymbol{M}_2$ такая, что $x_1 \leq x_2$. Отношение $\leq$ на множестве мультирубрик будем называть *отношением доминируемости*[1] и в случае, когда $\boldsymbol{M}_1 \leq \boldsymbol{M}_2$, будем говорить, что мультирубрика $\boldsymbol{M}_1$ *доминируется* мультирубрикой $\boldsymbol{M}_2$, а мультирубрика $\boldsymbol{M}_2$ *доминирует* мультирубрику $\boldsymbol{M}_1$.

В [1] установлено, что множество всех мультирубрик $MRL(\boldsymbol{T})$ рубрикаторного дерева $\boldsymbol{T}$ является *решеткой* относительно отношения доминируемости и указаны алгоритмы нахождения *объединения* $\vee$ и *пересечения* $\wedge$ элементов в этой решетке, т. е. нахождения наименьшей верхней и наибольшей нижней границ для произвольных пар мультирубрик.

Будем рассматривать отображение $List(x)$, которое каждой вершине x дерева $\boldsymbol{T}$ ставит в соответствие множество всех листьев b таких, что $b \leq x$. Для каждой вершины x дерева $\boldsymbol{T}$ множество $List(x)$ будем называть *листовым множеством вершины* x. Очевидно, $List(x)$ всегда является антицепью и, конечно, любое подмножество из $\boldsymbol{L}$ является антицепью.

Далее для любого подмножества $\boldsymbol{X}$ из $\boldsymbol{T}$ через $\boldsymbol{L}(\boldsymbol{X})$ будем обозначать множество всех листьев, содержащихся в $\boldsymbol{X}$.

Рассмотрим отображение $List(\boldsymbol{M})$ решетки мультирубрик $MRL(\boldsymbol{T})$ на булеву решетку $\mathfrak{B}(\boldsymbol{L})$ подмножеств листового множества $\boldsymbol{L}$, которое каждой мультирубрике $\boldsymbol{M}$ ставит в соответствие подмножество всех листьев из $(\boldsymbol{M}]$, т. е. $List(\boldsymbol{M}) = \boldsymbol{L}((\boldsymbol{M}])$. В [1] доказано, что отображение $List(\boldsymbol{M})$ является изоморфизмом решетки мультирубрик $MRL(\boldsymbol{T})$ на булеву решетку $\mathfrak{B}(\boldsymbol{L})$.

Множество $List(\boldsymbol{M})$ будем называть *листовым множеством мультирубрики* $\boldsymbol{M}$, а его мощность — *листовым весом мультирубрики* $\boldsymbol{M}$, который будем обозначать через $lweight(\boldsymbol{M})$.

Наша цель состоит в рассмотрении различных расстояний на решетке мультирубрик $MRL(\boldsymbol{T})$ и построении алгоритмов их вычисления.

---

[1]Поскольку мультирубрики определяются на рубрикаторных деревьях, отражающих иерархические тематические классификаторы, то отношение доминируемости можно трактовать как отношение «тематически уже-шире»; иначе говоря, тематическое содержание доминируемой мультирубрики с учетом всех ее подчиненных рубрик полностью входит в тематическое содержание доминирующей мультирубрики.

1. Порядковое расстояние Хемминга на множестве мультирубрик

Рубрикаторное дерево $\boldsymbol{T}$ будем называть *взвешенным*, если на нем задана весовая функция w, которая каждой вершине x этого дерева ставит в соответствие положительное действительное число $w(x)$, называемое *w-весом* (или просто *весом*) вершины x.

Рубрикаторное дерево $\boldsymbol{T}$ будем называть субаддитивно взвешенным, если его весовая функция w является субаддитивной, т. е. удовлетворяет свойству: для любого клана сумма w-весов его вершин не превосходит веса отца этого клана.

Субаддитивно взвешенное рубрикаторное дерево называется *аддитивно взвешенным*, если его весовая функция w является аддитивной, т. е. удовлетворяет свойству: для любого клана сумма w-весов всех его вершин равна весу отца этого клана.

Далее под $\boldsymbol{T}$ будем понимать взвешенное дерево $\boldsymbol{T}$ с субаддитивной весовой функцией w.

Для любой антицепи $\boldsymbol{X}$ дерева $\boldsymbol{T}$ через $w(\boldsymbol{X})$ будем обозначать ее w-вес (или просто вес), т. е. сумму w-весов вершин, входящих в $\boldsymbol{X}$. В частности, для любой мультирубрики $\boldsymbol{M}_4$ ее w-вес $w(\boldsymbol{M})$ равен сумме w-весов, входящих в нее вершин.

Дефектом $\delta(x)$ внутренней вершины x дерева будем называть число, равное разности веса вершины x и веса клана, отцом которого является x (отметим, что любой клан очевидно является антицепью), а дефектом $\delta(x)$ листа x дерева $\boldsymbol{T}$ по определению будем считать его вес. Заметим, что всегда дефект внутренней вершины неотрицателен, а дефект листа положителен. Для произвольного подмножества $\boldsymbol{X}$ из $\boldsymbol{T}$ через $\delta(\boldsymbol{X})$ будем обозначать сумму дефектов всех вершин из $\boldsymbol{X}$.

Ясно, что субаддитивная весовая функция w является аддитивной тогда и только тогда, когда дефект любой внутренней вершины равен нулю.

Далее под *высотой вершины* x дерева $\boldsymbol{T}$ будем понимать наибольшую из длин простых цепей от вершины x до листьев.

Тривиальной индукцией по высоте вершин нетрудно проверить, что вес любой вершины x равен сумме дефектов всех вершин порожденного ею порядкового идеала $(x]$, состоящего из меньших или равных x вершин дерева $\boldsymbol{T}$. Через $\delta((x])$ будем обозначать сумму дефектов всех вершин из $(x]$. В силу сказанного верно равенство $w(x) = \delta((x])$. Ясно, что субаддитивная весовая функция однозначно определяется набором дефектов вершин дерева.

В случае же аддитивной весовой функции вес любой вершины x дерева $\boldsymbol{T}$ равен сумме весов всех листьев множества $List(x)$. Действительно,

$$w(x) = \delta((x]) = \delta(List(x)) = w(List(x)).$$

Поэтому аддитивную весовую функцию будем называть также *листовой весовой функцией*. Конечно, особое значение имеет аддитивная весовая функция, для которой вес каждого листа равен 1. Такую весовую функцию будем называть *обыкновенной листовой весовой функцией*.

Для мультирубрики $\boldsymbol{M} = \{x_1, x_2, \ldots, x_m\}$ в силу выше сказанного имеем

$$w(\boldsymbol{M}) = w(x_1) + \ldots + w(x_m) = \delta((x_1]) + \ldots + \delta((x_m]) = \delta((\boldsymbol{M}]),$$

так как идеал $(\boldsymbol{M}]$ равен объединению непересекающихся идеалов

$$(x_1], \ldots, (x_m].$$

Таким образом,

$$w(\boldsymbol{M}) = \delta((\boldsymbol{M}]).$$

Отсюда следует, что в случае аддитивной весовой функции для любой мультирубрики $\boldsymbol{M}$ выполняется

$$w(\boldsymbol{M}) = \delta((\boldsymbol{M}]) = \delta(\boldsymbol{L}((\boldsymbol{M}])) = w(List(\boldsymbol{M})).$$

Пусть $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ — две мультирубрики взвешенного рубрикаторного дерева $\boldsymbol{T}$ с субаддитивной весовой функцией w. Определим порядковое расстояние Хемминга

$$h\text{-}odist(\boldsymbol{M}_1, \boldsymbol{M}_2; w)$$

от $\boldsymbol{M}_1$ до $\boldsymbol{M}_2$, отвечающее субаддитивной весовой функции w (или w-порядковое расстояние Хемминга), полагая его равным сумме дефектов элементов симметрической разности

$$(\boldsymbol{M}_1] \oplus (\boldsymbol{M}_2]$$

порядковых идеалов $(\boldsymbol{M}_1]$ и $(\boldsymbol{M}_2]$ (если симметрическая разность пуста, то полагаем расстояние равным нулю), которая по определению равна

$$((\boldsymbol{M}_1] \cup (\boldsymbol{M}_2]) \backslash ((\boldsymbol{M}_1] \cap (\boldsymbol{M}_2]) = ((\boldsymbol{M}_1] \backslash (\boldsymbol{M}_2]) \cup ((\boldsymbol{M}_2] \backslash (\boldsymbol{M}_1]).$$

Таким образом, по определению имеем

$$h\text{-}odist(\boldsymbol{M}_1, \boldsymbol{M}_2; w) = \delta((\boldsymbol{M}_1] \oplus (\boldsymbol{M}_2]). \tag{1}$$

В случае же аддитивной весовой функции порядковое расстояние Хемминга от $\boldsymbol{M}_1$ до $\boldsymbol{M}_2$ будем обозначать через

$$h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w)$$

и будем называть его листовым расстоянием Хемминга, отвечающим аддитивной весовой функции w (или w-листовым расстоянием Хемминга), поскольку в этом случае верно равенство

$$h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w) = w(List(\boldsymbol{M}_1) \oplus List(\boldsymbol{M}_2)).$$

Действительно, так как очевидно выполняются равенства

$$\boldsymbol{L}((\boldsymbol{M}_1] \oplus (\boldsymbol{M}_2]) = \boldsymbol{L}((\boldsymbol{M}_1]) \oplus \boldsymbol{L}((\boldsymbol{M}_2]) = List(\boldsymbol{M}_1) \oplus List(\boldsymbol{M}_2),$$

в случае аддитивной весовой функции мы получаем

$$h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w) = \delta((\boldsymbol{M}_1] \oplus (\boldsymbol{M}_2]) = \delta(\boldsymbol{L}((\boldsymbol{M}_1] \oplus (\boldsymbol{M}_2])) =$$
$$= \delta(List(\boldsymbol{M}_1) \oplus List(\boldsymbol{M}_2)) = w(List(\boldsymbol{M}_1) \oplus List(\boldsymbol{M}_2)).$$

В случае же, когда в качестве w рассматривается обыкновенная листовая весовая функция, т. е. такая функция, что $w(b) = 1$ для любой листовой вершины b и $\delta(x) = 0$ для любой внутренней вершины x, указанное расстояние будем называть обыкновенным листовым расстоянием Хемминга от $\boldsymbol{M}_1$ до $\boldsymbol{M}_2$ и будем обозначать его через $h\text{-}ldist(\boldsymbol{M}_1,\ \boldsymbol{M}_2)$.

Справедлива следующая

Теорема 1. *Порядковое расстояние Хемминга для мультирубрик обладает обычными свойствами расстояния, а именно для любых мультирубрик $\boldsymbol{M}_1$, $\boldsymbol{M}_2$ и $\boldsymbol{M}_3$ выполняется:*

1) *h-odist*$(\boldsymbol{M}_1, \boldsymbol{M}_2; w) > 0$, *если* $\boldsymbol{M}_1 \neq \boldsymbol{M}_2$, *и* *h-odist*$(\boldsymbol{M}_1, \boldsymbol{M}_2; w) = 0$, *если* $\boldsymbol{M}_1 = \boldsymbol{M}_2$.
2) *h-odist*$(\boldsymbol{M}_1, \boldsymbol{M}_2; w) =$ *h-odist*$(\boldsymbol{M}_2, \boldsymbol{M}_1; w)$.
3) *h-odist*$(\boldsymbol{M}_1, \boldsymbol{M}_3; w) \leq$ *h-odist*$(\boldsymbol{M}_1, \boldsymbol{M}_2; w) +$ *h-odist*$(\boldsymbol{M}_2, \boldsymbol{M}_3; w)$ *(неравенство треугольника).*

Доказательство. Доказательство. Условие 1 справедливо в силу того, что для любых двух различных мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ выполняется $List(\boldsymbol{M}_1) \neq$ $\neq List(\boldsymbol{M}_2)$ и поэтому существует лист, принадлежащий симметрической разности $(\boldsymbol{M}_1] \oplus (\boldsymbol{M}_2]$, а вес листа положителен.

Условие 2 очевидно.

Докажем условие 3.

Заметим сначала, что для любых множеств $\boldsymbol{X}$, $\boldsymbol{Y}$ и $\boldsymbol{Z}$ любой элемент множества $\boldsymbol{X}\backslash\boldsymbol{Z}$ содержится либо в $\boldsymbol{X}\backslash\boldsymbol{Y}$, либо в $\boldsymbol{Y}\backslash\boldsymbol{Z}$. Поэтому любой элемент множества $\boldsymbol{X} \oplus \boldsymbol{Z}$ содержится либо в $\boldsymbol{X} \oplus \boldsymbol{Y}$, либо в $\boldsymbol{Y} \oplus \boldsymbol{Z}$. Отсюда при $\boldsymbol{X} = (\boldsymbol{M}_1]$, $\boldsymbol{Y} = (\boldsymbol{M}_2]$ и $\boldsymbol{Z} = (\boldsymbol{M}_3]$ следует в силу (1), что каждое слагаемое выражения *h-odist*$(\boldsymbol{M}_1, \boldsymbol{M}_3; w)$ из правой части неравенства 3, отвечающее элементу из $\boldsymbol{X} \oplus \boldsymbol{Z}$, присутствует в соответствии с (1) также в качестве слагаемого в одном из выражений

$$h\text{-}odist(\boldsymbol{M}_1, \boldsymbol{M}_2; w) \text{ или } h\text{-}odist(\boldsymbol{M}_2, \boldsymbol{M}_3; w),$$

отвечая тому же элементу, который лежит либо в $\boldsymbol{X} \oplus \boldsymbol{Y}$, либо в $\boldsymbol{Y} \oplus \boldsymbol{Z}$. Отсюда получаем, что число *h-odist*$(\boldsymbol{M}_1, \boldsymbol{M}_3; w)$ меньше или равно суммы *h-odist*$(\boldsymbol{M}_1, \boldsymbol{M}_2; w) +$ *h-odist*$(\boldsymbol{M}_2, \boldsymbol{M}_3; w)$.

Теорема доказана. □

Пример 1. Рассмотрим рубрикаторное[2] дерево, представленное на рис. 1. Для удобства листовые вершины здесь пронумерованы цифрами, а внутренние вершины — буквами латинского алфавита.

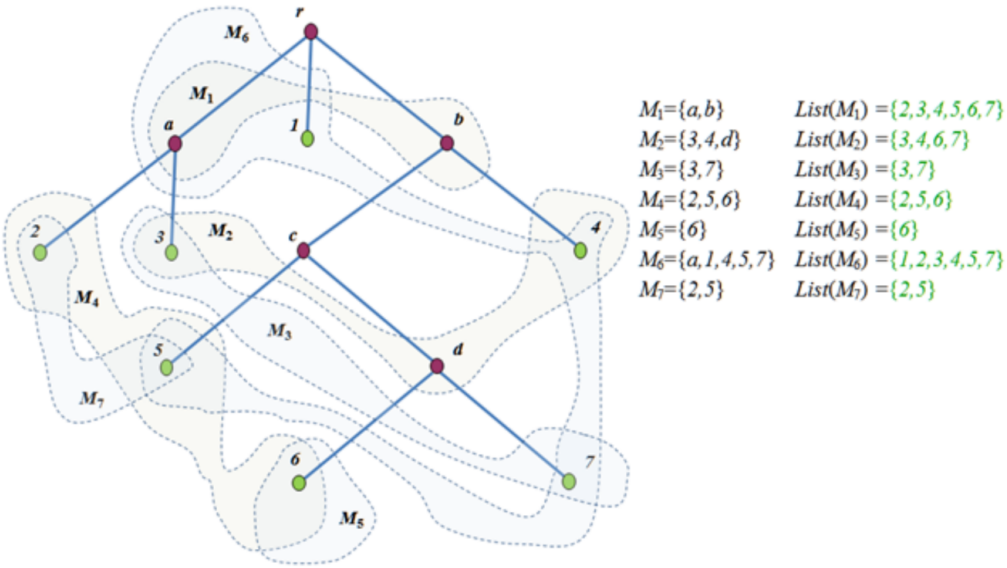
На дереве представлены мультирубрики $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, $\boldsymbol{M}_3$, $\boldsymbol{M}_4$, $\boldsymbol{M}_5$, $\boldsymbol{M}_6$ и $\boldsymbol{M}_7$. Для частичного порядка $\leq$ на множестве мультирубрик, операций объединения $\vee$ и пересечения $\wedge$ мультирубрик выполняется [3]: $\boldsymbol{M}_3 < \boldsymbol{M}_2 < \boldsymbol{M}_1$; $\boldsymbol{M}_5 < \boldsymbol{M}_2 < \boldsymbol{M}_1$; $\boldsymbol{M}_5 < \boldsymbol{M}_4 < \boldsymbol{M}_1$; $\boldsymbol{M}_3 < \boldsymbol{M}_6$; $\boldsymbol{M}_1 \parallel \boldsymbol{M}_6$; $\boldsymbol{M}_2 \parallel \boldsymbol{M}_4$; $\boldsymbol{M}_4 \parallel \boldsymbol{M}_6$; $\boldsymbol{M}_3 \parallel \boldsymbol{M}_5$; $\boldsymbol{M}_2 \parallel \boldsymbol{M}_6$; $\boldsymbol{M}_5 \parallel \boldsymbol{M}_6$; $\boldsymbol{M}_2 \vee \boldsymbol{M}_3 = \boldsymbol{M}_2$; $\boldsymbol{M}_2 \wedge \boldsymbol{M}_3 = \boldsymbol{M}_2$; $\boldsymbol{M}_2 \vee \boldsymbol{M}_4 = \boldsymbol{M}_1$; $\boldsymbol{M}_2 \wedge \boldsymbol{M}_4 = \boldsymbol{M}_5$; $\boldsymbol{M}_2 \vee \boldsymbol{M}_7 = \boldsymbol{M}_1$; $\boldsymbol{M}_2 \wedge \boldsymbol{M}_7 = \varnothing$.

В таблице рис. 1 приведены значения обыкновенного листового расстояния Хэмминга *h-ldist*$(M_i,\ M_j)$, вычисляемого как мощность симметрической разности листовых множеств соответствующих мультирубрик, а также указаны минимальное значение расстояния для разных из указанных мультирубрик, равное 1 (это расстояние между мультирубриками $\boldsymbol{M}_4$ и $\boldsymbol{M}_7$, симметрическая разность листовых множеств которых $List(\boldsymbol{M}_4) \oplus List(\boldsymbol{M}_7)$ состоит из одной листовой вершины 6), и максимальное значение расстояния, равное 7 (это расстояние между мультирубриками $\boldsymbol{M}_5$ и $\boldsymbol{M}_6$, симметрическая разность листовых множеств которых $List(\boldsymbol{M}_5) \oplus List(\boldsymbol{M}_6)$ совпадает с множеством $\boldsymbol{L}$ всех листовых вершин данного дерева).

---

[2]Напомним, что рубрикаторное дерево — это корневое дерево, в котором каждая внутренняя вершина имеет не менее двух сыновей.

[3]Знаком «$\parallel$» обозначено отношение несравнимости мультирубрик



$h\text{-}ldist(M_i, M_j) = |List(M_i) \oplus List(M_j)|$

	M_2	M_3	M_4	M_5	M_6	M_7
M_1	$\lvert\{2,5\}\rvert = \mathbf{2}$	$\lvert\{2,4,5,6\}\rvert = \mathbf{4}$	$\lvert\{3,4,7\}\rvert = \mathbf{3}$	$\lvert\{2,3,4,5,7\}\rvert = \mathbf{5}$	$\lvert\{1,6\}\rvert = \mathbf{2}$	$\lvert\{3,4,6,7\}\rvert = \mathbf{4}$
M_2		$\lvert\{4,6\}\rvert = \mathbf{2}$	$\lvert\{2,3,4,5,7\}\rvert = \mathbf{5}$	$\lvert\{3,4,7\}\rvert = \mathbf{3}$	$\lvert\{1,2,5,6\}\rvert = \mathbf{4}$	$\lvert\{2,3,4,5,6,7\}\rvert = \mathbf{6}$
M_3			$\lvert\{2,3,5,6,7\}\rvert = \mathbf{5}$	$\lvert\{3,6,7\}\rvert = \mathbf{3}$	$\lvert\{1,2,4,5\}\rvert = \mathbf{4}$	$\lvert\{2,3,5,7\}\rvert = \mathbf{4}$
M_4				$\lvert\{2,5\}\rvert = \mathbf{2}$	$\lvert\{1,3,4,6,7\}\rvert = \mathbf{5}$	$\lvert\{6\}\rvert = \mathbf{1}$
M_5					$\lvert\{1,2,3,4,5,6,7\}\rvert = \mathbf{7}$	$\lvert\{2,5,6\}\rvert = \mathbf{3}$
M_6						$\lvert\{1,3,4,7\}\rvert = \mathbf{4}$

Рис. 1. Пример рубрикаторного дерева, мультирубрик и обыкновенного листового расстояния Хемминга между ними

Подсчет расстояний в примере 1 также указывает на желательность построения эффективных алгоритмов нахождения расстояний между мультирубриками без вычисления их листовых подмножеств.

2. Алгоритмы вычисления листового расстояния Хемминга на решетке мультирубрик

Справедлива следующая

Теорема 2. *Пусть $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ — две мультирубрики взвешенного рубрикаторного дерева $\boldsymbol{T}$ с аддитивной весовой функцией w. Тогда*

$$h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w) = w(\boldsymbol{M}_1 \vee \boldsymbol{M}_2) - w(\boldsymbol{M}_1 \wedge \boldsymbol{M}_2) =$$

$$= w(\boldsymbol{M}_1) + w(\boldsymbol{M}_2) - 2w(\boldsymbol{M}_1 \wedge \boldsymbol{M}_2).$$

Доказательство. Поскольку отображение $List(\boldsymbol{M})$ является изоморфизмом решетки $MRL(\boldsymbol{T})$ на булеву решетку $\mathfrak{B}(\boldsymbol{L})$, для любых мультирубрик $\boldsymbol{M}_1$

и $\boldsymbol{M}_2$ выполняется

$$List(\boldsymbol{M}_1 \vee \boldsymbol{M}_2) = List(\boldsymbol{M}_1) \cup List(\boldsymbol{M}_2), List(\boldsymbol{M}_1 \wedge \boldsymbol{M}_2) = List(\boldsymbol{M}_1) \cap List(\boldsymbol{M}_2).$$

Отсюда следует, что

$$\begin{gathered} h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w) = w(List(\boldsymbol{M}_1) \oplus List(\boldsymbol{M}_2)) = \\ = w((List(\boldsymbol{M}_1) \cup \ List(\boldsymbol{M}_2)) \backslash (List(\boldsymbol{M}_1) \cap List(\boldsymbol{M}_2))) = \\ = w(List(\boldsymbol{M}_1) \cup List(\boldsymbol{M}_2)) - w(List(\boldsymbol{M}_1) \cap List(\boldsymbol{M}_2)) = \\ = w(List(\boldsymbol{M}_1 \vee \boldsymbol{M}_2)) - w(List(\boldsymbol{M}_1 \wedge \boldsymbol{M}_2)) = \\ = w(\boldsymbol{M}_1 \vee \boldsymbol{M}_2) - w(\boldsymbol{M}_1 \wedge \boldsymbol{M}_2). \end{gathered}$$

Аналогично получаем

$$\begin{gathered} h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w) = w(List(\boldsymbol{M}_1) \oplus List(\boldsymbol{M}_2)) = \\ = w((List(\boldsymbol{M}_1) \backslash List(\boldsymbol{M}_2)) \cup (List(\boldsymbol{M}_2) \backslash List(\boldsymbol{M}_1))) = \\ = w((List(\boldsymbol{M}_1)) - w(List(\boldsymbol{M}_1) \cap List(\boldsymbol{M}_2)) + \\ + w((List(\boldsymbol{M}_2) - w(List(\boldsymbol{M}_1) \cap List(\boldsymbol{M}_2)) = \\ = w(\boldsymbol{M}_1) + w(\boldsymbol{M}_2) - 2w(\boldsymbol{M}_1 \wedge \boldsymbol{M}_2). \end{gathered}$$

Теорема доказана. □

Из теоремы 2 вытекает

Следствие 1. *Для любых двух мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ взвешенного рубрикаторного дерева $\boldsymbol{T}$ с аддитивной весовой функцией w выполняется*

$$w(\boldsymbol{M}_1 \vee \boldsymbol{M}_2) = w(\boldsymbol{M}_1) + w(\boldsymbol{M}_2) - w(\boldsymbol{M}_1 \wedge \boldsymbol{M}_2).$$

Из теоремы 2 вытекает также следующее

Следствие 2. *Для цепи $\boldsymbol{M}_1 \leq \boldsymbol{M}_2 \leq \boldsymbol{M}_3$ мультирубрик неравенство треугольника 3 вырождается в равенство.*

Доказательство. Действительно, так как для любых двух сравнимых мультирубрик их объединение равно большей из них, а их пересечение — меньшей из них, три раза применяя первое из равенств теоремы 2, получаем

$$\begin{gathered} h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_3; w) = w(\boldsymbol{M}_3) - w(\boldsymbol{M}_1) = w(\boldsymbol{M}_3) - w(\boldsymbol{M}_2) + w(\boldsymbol{M}_2) - w(\boldsymbol{M}_1) = \\ = h\text{-}ldist(\boldsymbol{M}_2, \boldsymbol{M}_3; w) + h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w). \end{gathered}$$

Следствие доказано. □

Поскольку для любой мультирубрики $\boldsymbol{M} = \{x_1, x_2, \ldots, x_m\}$ выполняется

$$w(\boldsymbol{M}) = w(m_1) + w(m_2) + \ldots + w(x_m),$$

опираясь на теорему 2, теперь можно вычислить листовое расстояние

$$h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w)$$

от $\boldsymbol{M}_1$ до $\boldsymbol{M}_2$, используя лишь заданные w-веса вершин и не находя при этом листовые множества вершин и мультирубрик.

Из вышесказанного следует корректность следующего алгоритма вычисления листового расстояния Хемминга от $\boldsymbol{M}_1$ до $\boldsymbol{M}_2$.

Алгоритм 1. Пусть заданы две мультирубрики $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ взвешенного рубрикаторного дерева $\boldsymbol{T}$ с аддитивной весовой функцией w.

1) Вычисляем w-веса мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ как суммы w-весов, входящих в них вершин.
2) С помощью алгоритма 3 из [1] находим мультирубрику $\boldsymbol{M}_1 \wedge \boldsymbol{M}_2$, а затем вычисляем ее w-вес.
3) Наконец, вычисляем листовое расстояние Хемминга

$$h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w) = w(\boldsymbol{M}_1) + w(\boldsymbol{M}_2) - 2w(\boldsymbol{M}_1 \wedge \boldsymbol{M}_2)$$

от $\boldsymbol{M}_1$ до $\boldsymbol{M}_2$, отвечающее аддитивной весовой функции w.

Отметим, что в частном случае обыкновенной листовой весовой функции для любых мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ выполняется

$$ldist(\boldsymbol{M}_1, \boldsymbol{M}_2) = lweight(\boldsymbol{M}_1 \vee \boldsymbol{M}_2) - lweight(\boldsymbol{M}_1 \wedge \boldsymbol{M}_2).$$

Кроме того, в этом случае длина максимальной цепи в булевой решетке $\mathfrak{B}(\boldsymbol{L})$ от $List(\boldsymbol{M}_1) \cap List(\boldsymbol{M}_2)$ до $List(\boldsymbol{M}_1)$ очевидно равна мощности множества $List(\boldsymbol{M}_1) \backslash List(\boldsymbol{M}_2)$. Аналогично, длина максимальной цепи в булевой решетке $\mathfrak{B}(\boldsymbol{L})$ от $List(\boldsymbol{M}_1) \cap List(\boldsymbol{M}_2)$ до $List(\boldsymbol{M}_2)$ равна мощности множества $List(\boldsymbol{M}_2) \backslash List(\boldsymbol{M}_1)$. Таким образом, сумма длин этих максимальных цепей равна мощности множества $List(\boldsymbol{M}_1) \oplus List(\boldsymbol{M}_2)$, т. е. равна расстоянию $h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2)$. Поэтому легко видеть, что в силу изоморфизма решеток $MRL(\boldsymbol{T})$ и $\mathfrak{B}(\boldsymbol{L})$ обыкновенное листовое расстояние от $\boldsymbol{M}_1$ до $\boldsymbol{M}_2$ равно обычному расстоянию от $\boldsymbol{M}_1$ до $\boldsymbol{M}_2$ в графе, представляющем диаграмму Хассе решетки $MRL(\boldsymbol{T})$ (здесь под расстоянием в графе от $\boldsymbol{M}_1$ до $\boldsymbol{M}_2$ понимается длина кратчайшей простой цепи от $\boldsymbol{M}_1$ до $\boldsymbol{M}_2$).

На рис. 2 приведена диаграмма Хассе решетки мультирубрик рубрикаторного дерева, представленного в примере 1. Желтым цветом выделены мультирубрики, отмеченные на рис. 1. Простые цепи наименьшей длины, соединяющие на графе диаграммы Хассе соответствующие мультирубрики, выделены красным цветом. Цепи с наименьшей длиной (от $\boldsymbol{M}_4$ до $\boldsymbol{M}_7$) и наибольшей (от $\boldsymbol{M}_5$ до $\boldsymbol{M}_6$) выделены более жирным зеленым и фиолетовым цветом, соответственно.

Несмотря на иллюстрационную привлекательность вычисления расстояний между мультирубриками по диаграмме Хассе, в практическом плане такой способ является малопригодным ввиду его существенной временной сложности, поскольку построение самой диаграммы Хассе является весьма затратной процедурой, основанной на переборе всевозможных подмножеств листового множества дерева с одновременным иерархическим сжатием кланов вершин (см. [1]). Таким образом, рис. 2 так же указывает на необходимость построения алгоритмов вычисления листового расстояния Хемминга, не использующих листовых подмножеств мультирубрик.

В этом свете обсудим теперь возможные реализации алгоритма 1.

Ясно, что проблема заключается в эффективном вычислении объединения $\boldsymbol{M}_1 \vee \boldsymbol{M}_2$ и пересечения $\boldsymbol{M}_1 \wedge \boldsymbol{M}_2$ для произвольных мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$.

Будем предполагать наличие двух дополнительных естественных условий для рубрикаторного дерева $\boldsymbol{T}$ с аддитивной весовой функцией w:

- для каждой вершины x известен ее уровень $level(x)$, т. е. длина простой цепи от x до r в дереве $\boldsymbol{T}$;
- для каждой некорневой вершины x известен ее отец $father(x)$.

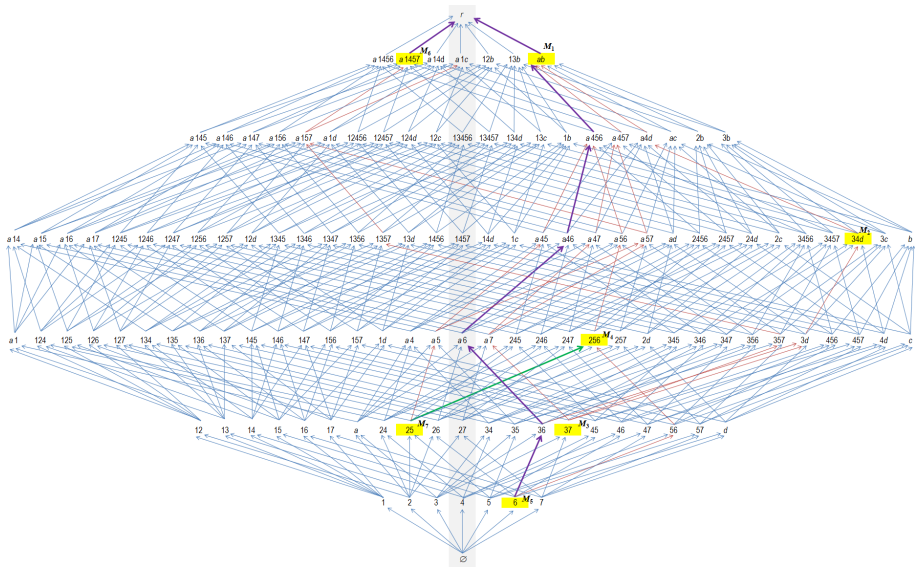


Рис. 2. Диаграмма Хассе решетки мультирубрик рубрикаторного дерева, представленного в примере 1

В соответствии с алгоритмом 3 из [1] мультирубрика $\boldsymbol{M}_1 \ \wedge \ \boldsymbol{M}_2$ состоит из всех вершин x_1 из $\boldsymbol{M}_1$, для каждой из которых существует вершина y_1 из $\boldsymbol{M}_2$ такая, что $x_1 \leq y_1$, и всех вершин y_2 из $\boldsymbol{M}_2$, для каждой из которых существует вершина x_2 из $\boldsymbol{M}_1$ такая, что $y_2 \leq x_2$. Поскольку $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ являются антицепями, нахождение мультирубрики $\boldsymbol{M}_1 \ \wedge \ \boldsymbol{M}_2$ существенно упрощается тем обстоятельством, что множество $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$ очевидно не содержит цепей длины два относительно порядка $\leq$. В силу этого замечания и комментариев, приведенных ниже, нетрудно убедиться в корректности следующего алгоритма.

Алгоритм 2. Пусть $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ — две мультирубрики рубрикаторного дерева $\boldsymbol{T}$.

1) Вершины мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ помещаем в двусвязные очереди S_1 и S_2, соответственно. Создаем
   - очередь S (в S будем накапливать ответ, т. е. будем накапливать вершины мультирубрики $\boldsymbol{M}_1 \wedge \boldsymbol{M}_2$) и делаем ее пустой;
   - двусвязную очередь CS_2 (CS_2 будет вспомогательной двусвязной очередью, которая во время работы алгоритма периодически совпадает с текущим значением двусвязной очереди S_2, а затем постепенно теряет свои элементы) и делаем ее пустой.
2) Если $S_1 = \varnothing$, то переходим к шагу 10. Если $S_1 \neq \varnothing$, то в качестве x берем первый элемент двусвязной очереди S_1.
3) Если $S_2 = \varnothing$, то переходим к шагу 10. Если $S_2 \neq \varnothing$, то копируем двусвязную очередь S_2 во вспомогательную двусвязную очередь CS_2.
4) В качестве y берем первый элемент двусвязной очереди CS_2.
5) Если $x = y$, то удаляем x из S_1 и из S_2, добавляем x в очередь S, делаем пустой двусвязную очередь CS_2 и переходим к шагу 2.

6) Если $level(x) > level(y)$ и y равен результату последовательных переходов, начиная с x, от вершин к их отцам $father$, то удаляем x из S_1, добавляем x в очередь S, делаем пустой двусвязную очередь CS_2 и переходим к шагу 2 (отметим, что в этом случае $x < y$ и требуется $level(x) - level(y)$ применений переходов к отцам вершин).
7) Если $level(x) < level(y)$ и x равен результату последовательных переходов, начиная с y, от вершин к их отцам $father$, то удаляем y из S_2 и из CS_2, добавляем y в очередь S и переходим к шагу 4 (отметим, что в этом случае $y < x$ и требуется $level(y) - level(x)$ применений переходов к отцам вершин).
8) Удаляем y из CS_2. Если $CS_2 \neq \varnothing$, то переходим к шагу 4.
9) Удаляем x из S_1 и переходим к шагу 2.
10) Очередь S состоит из вершин мультирубрики $\boldsymbol{M}_1 \wedge \boldsymbol{M}_2$. Завершение алгоритма.

Нетрудно заметить, что указанный алгоритм 2 имеет вычислительную сложность $O(t_1 t_2 t_3)$ (о временной сложности алгоритмов см., например, в [2] или [4]), где t_1 и t_2 — мощности мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$, соответственно, а t_3 — наибольшая из абсолютных величин разностей вида $level(x) - level(y)$, где x пробегает $\boldsymbol{M}_1$, а y — $\boldsymbol{M}_2$.

Теперь для нахождения с помощью алгоритма 1 листового расстояния Хемминга $h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w)$ мы можем воспользоваться алгоритмом 2. Нетрудно видеть, что с использованием такой организации вычислений алгоритм 1 будет работать быстрей, чем нахождение листовых множеств мультирубрик с помощью поиска в глубину (см. [2] или [4]) и последующее прямое вычисление листового расстояния Хемминга $h\text{-}ldist(\boldsymbol{M}_1, \boldsymbol{M}_2; w)$ через листовые множества в случаях, когда листовые множества достаточно велики.

Обсудим теперь возможную реализацию алгоритма 2 из [1], который находит мультирубрику $\boldsymbol{M}_1 \vee \boldsymbol{M}_2$ для заданных мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ рубрикаторного дерева $\boldsymbol{T}$. Будем предполагать наличие дополнительных естественных условий для рубрикаторного дерева $\boldsymbol{T}$:

- для каждой вершины x известен ее уровень $level(x)$;
- для каждой не листовой вершины x известна мощность $nClan(x)$ клана $Clan(x)$, отцом которого является вершина x.

В соответствии с алгоритмом 2 из [1] для получения мультирубрики $\boldsymbol{M}_1 \vee \vee \boldsymbol{M}_2$ необходимо

- сначала найти антицепь $\boldsymbol{A}$ максимальных элементов в $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$ относительно порядка $\leq$;
- а затем, начиная с $\boldsymbol{A}$, последовательно заменять в получаемых множествах (пока это возможно) содержащиеся в них кланы на отцов этих кланов, т. е. совершать сжатия кланов.

Поскольку $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ являются антицепями, нахождение мультирубрики $\boldsymbol{M}_1 \vee \boldsymbol{M}_2$ опять-таки существенно упрощается тем обстоятельством, что множество $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$ не содержит цепей длины два относительно порядка $\leq$.

Следующий алгоритм, корректность которого нетрудно установить в силу сделанного замечания и комментариев, приведенных ниже, находит антицепь $\boldsymbol{A}$ максимальных элементов из $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$ относительно порядка $\leq$. Во многом

этот алгоритм подобен алгоритму 2. В нем сначала фактически находится множество B не максимальных элементов из $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$, а затем искомое множество $\boldsymbol{A}$ получается в виде теоретико-множественной разности $(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \backslash B$.

Алгоритм 3. Пусть $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ — две мультирубрики рубрикаторного дерева $\boldsymbol{T}$.

1) Вершины мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ помещаем в двусвязные очереди S_1 и S_2. Создаем двусвязную очередь CS_2 (в CS_2 будем хранить вспомогательную двусвязную очередь, которая во время работы алгоритма периодически совпадает с текущим значением очереди S_2, а затем постепенно теряет свои элементы) и делаем ее пустой. Предполагаем, что вначале все вершины из S_1 и из S_2 имеют зеленый цвет.
2) Если $S_1 = \varnothing$, то переходим к шагу 10 (текущее множество $S_1 \cup S_2$ не содержит пар сравнимых различных элементов). Если $S_1 \neq \varnothing$, то в качестве x берем первый элемент двусвязной очереди S_1.
3) Если $S_2 = \varnothing$, то переходим к шагу 10 (текущее множество $S_1 \cup S_2$ не содержит пар сравнимых различных элементов). Если $S_2 \neq \varnothing$, то копируем двусвязную очередь S_2 во вспомогательную двусвязную очередь CS_2.
4) В качестве y берем первый элемент двусвязной очереди CS_2.
5) Если $x = y$, то красим вершину x в желтый цвет, удаляем x из S_1 и из S_2, делаем пустой двусвязную очередь CS_2 и переходим к 2 (желтая вершина x является общей вершиной для $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$; она максимальна в $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$, так как $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ — антицепи).
6) Если $level(x) > level(y)$ и y равен результату последовательных переходов, начиная с x, от вершин к их отцам $father$, то красим x в красный цвет, удаляем x из S_1, делаем пустой двусвязную очередь CS_2 и переходим к шагу 2 (отметим, что в этом случае $x < y$, требуется $level(x) - level(y)$ переходов от вершин к их отцам и вершина x не содержится в $\boldsymbol{M}_2$, т. е. вершина x не является общей для $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$, вершина x не является максимальной в $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$).
7) Если $level(x) < level(y)$ и x равен результату последовательных переходов, начиная с y, от вершин к их отцам $father$, то красим y в красный цвет, удаляем y из S_2 и из CS_2, переходим к шагу 4 (отметим, что в этом случае $y < x$, требуется $level(y) - level(x)$ переходов от вершин к их отцам и вершина y не содержится в $\boldsymbol{M}_1$, т. е. вершина y не является общей для $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$, вершина y не является максимальной в $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$).
8) Удаляем y из CS_2. Если $CS_2 \neq \varnothing$, то переходим к шагу 4.
9) Удаляем x из S_1 и переходим к шагу 2.
10) Удаляем из множества $\boldsymbol{M}_1$ вершины красного цвета, а из множества $\boldsymbol{M}_2$ — вершины желтого цвета и вершины красного цвета (нетрудно заметить, что красные вершины не являются общими для $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$, желтые вершины являются общими для $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$, любая общая вершина в процессе работы алгоритма будет окрашена в желтый цвет, а множество вершин красного цвета равно множеству B не максимальных элементов из $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$, поэтому множество максимальных вершин

из $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$ исчерпывается зелеными и желтыми вершинами). Полученные множества не пересекаются, а их объединение совпадает с антицепью $\boldsymbol{A}$ максимальных элементов из $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$ относительно порядка $\leq$. Завершение алгоритма.

Нетрудно заметить, что указанный алгоритм также имеет вычислительную сложность $O(t_1 t_2 t_3)$, где t_1 и t_2 — мощности мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$, соответственно, а t_3 — наибольшая из абсолютных величин разностей вида $level(x) - level(y)$, где x пробегает $\boldsymbol{M}_1$, а y — $\boldsymbol{M}_2$.

Следующий алгоритм, исходя из заданной антицепи $\boldsymbol{A}$, с помощью последовательного применения сжатий кланов строит антицепь, не содержащую кланов. Корректность этого алгоритма следует из комментариев, приведенных ниже, и того факта, что после выполнения процедуры сжатия клана в антицепи снова получается антицепь. Поскольку одноэлементные антицепи являются мультирубриками, нам достаточно рассмотреть случай не одноэлементных антицепей $\boldsymbol{A}$.

Алгоритм 4. Пусть $\boldsymbol{A}$ — не одноэлементная антицепь, состоящая из элементов рубрикаторного дерева $\boldsymbol{T}$ с корнем r.

1) Создаем очередь S, в которую копируем элементы антицепи $\boldsymbol{A}$, создаем также вспомогательную двусвязную очередь F (в которой будем хранить, по мере надобности, отцов вершин из S) и делаем ее пустой.
2) Если $S = \varnothing$, то переходим к шагу 5. Если $S_1 \neq \varnothing$, то в качестве x берем первый элемент очереди S. Если $x = r$, то (текущая антицепь одноэлементна и состоит лишь из корня r, поэтому $F = \varnothing$) переходим к шагу 5. Если $x \neq r$, то в качестве y берем отца $father(x)$ вершины x.
3) Если y не содержится в F, то помещаем y в конец двусвязной очереди F_4, создаем вспомогательную очередь $Sons(y)$, состоящую из одной вершины x (затем в ней будем накапливать сыновей вершины y, которые будут встречаться в S в процессе работы алгоритма), полагаем $n(y) = 1$ ($n(y)$ всегда будет равно числу элементов в очереди $Sons(y)$, т. е. будет равно длине очереди $Sons(y)$), удаляем x из S и переходим к шагу 2.
4) Если y содержится в F, то если $n(y) = nClan(y) - 1$, то удаляем из двусвязной очереди F, удаляем x из S, добавляем y в конец очереди S (фактически здесь происходит процесс удаления клана $Clan(y)$ из имеющейся антицепи и замена этого клана на его отца y для получения новой антицепи, т. е. выполняется процесс сжатия клана), удаляем вспомогательную очередь $Sons(y)$ и полагаем, что для вершины y значение числа $n(y)$ не определено, переходим к шагу 2; если $n(y) < nClan(y) - 1$, то добавляем x в конец очереди $Sons(y)$, увеличиваем значение величины $n(y)$ на 1, удаляем x из S и переходим к шагу 2.
5) Если $F = \varnothing$, то искомая антицепь одноэлементна и состоит лишь из одного корня r. Если $F \neq \varnothing$, то объединение множеств вершин очередей вида $Sons(y)$, где y пробегает F, (эти множества попарно не пересекаются) является искомой антицепью, не содержащей кланов. Завершение алгоритма.

Нетрудно заметить, что указанный алгоритм имеет временную сложность $O(t^2)$, где t — мощность антицепи $\boldsymbol{A}$. Действительно, в ходе выполнения алгоритма может произойти не более чем $t - 1$ сжатий кланов, так как после каждого сжатия клана длина изменяемой цепи уменьшается не менее чем на 1.

Поэтому в очередь S может добавиться не более чем $t-1$ новых вершин, т. е. в ходе работы алгоритма в качестве x на шаге 2 может быть выбрано не более чем $2t-1$ вершин. Осталось заметить, что длина двусвязной очереди F всегда не превосходит t.

Отметим, что имеются дополнительные возможности для ускорения работы алгоритма 4. Например, множество F можно представлять не в виде двусвязной очереди, а представлять его в виде дерева (или какой-либо другой структуры данных), обеспечивающего быстрые поиск, вставку и удаление вершин по какому-либо ключу, удобному для применения в конкретной области использования рубрикаторного дерева иерархического тематического классификатора для хранения и поиска информации.

Теперь для заданных мультирубрик $\boldsymbol{M}_1$ и $\boldsymbol{M}_2$ рубрикаторного дерева $\boldsymbol{T}$, последовательно применяя алгоритмы 3 и 4, можно найти мультирубрику $\boldsymbol{M}_1 \vee \vee \boldsymbol{M}_2$.

Сравнению временной сложности рассмотренных алгоритмов и некоторых иных алгоритмов для мультирубрик будет посвящена другая работа авторов.

Заключение

Конечно, метрика в виде порядкового расстояния Хемминга не является единственным видом расстояния между мультирубриками, которые представляют специальные подмножества вершин на корневых деревьях. Вместе с тем по отношению к рубрикаторным деревьям, выражающим иерархические тематические классификаторы, листовое расстояние Хемминга имеет ясный смысл, поскольку конкретная тематика предметной области, отображаемая классификаторами, сосредоточена именно в листовых рубриках.

В результате, как отмечалось во введении, листовое расстояние Хемминга обеспечивает механизмы поиска тематически проиндексированных текстов эффективной мерой, которая позволяет существенно повысить релевантность результатов поиска.

Следует отметить, что предложенное понятие порядкового расстояния Хемминга, основанное на взвешивании вершин рубрикаторного дерева, представляет широкие возможности для учета специфики различных предметных областей применения рубрикаторных деревьев. Вместо рассмотренных субаддитивного и аддитивного взвешивания вершин могут использоваться и другие подходы.

Кроме того, учет специфики предметных областей может обеспечиваться за счет спецификации вида функций, определяющих исходные w-веса листовых вершин. Например, w-веса листовых вершин b могут определяться в зависимости от их уровня $level(b)$, от количества вершин в клане, содержащем b, и т. п. Еще больше возможностей дает порядковое расстояние Хемминга в общем случае субаддитивных функций.

В отношении представленных алгоритмов вычисления листового расстояния Хэмминга следует отметить, что их эффективность, с одной стороны, очевидно существенным образом зависит от особенностей тех или иных видов рубрикаторных деревьев (сильно или слабо ветвящихся, многоуровневых или неглубоких и т.д.) а, с другой, от способа представления самих корневых деревьев. Возникающие здесь задачи являются предметом дальнейших исследований авторов. Другой круг задач связан с построением эффективных алгоритмов для

вычисления порядковых расстояний Хемминга в общем случае субаддитивных функций.

References

[1] N.A. Gaydamakin, V.A. Baransky, *Algebra of multirubric on root trees of hierarchical thematic classifiers*, Siberian Electronic Mathematical Reports, **14** (2017), 1030–1040. Zbl 06818286

[2] M.O. Asanov, V.A. Baransky, V.V. Rasin, *Diskretnaya matematika: grafyi, matroidyi, algoritmyi*, SPb.: Lan, 2010.

[3] V.A. Baransky, V.V. Kabanov, *Obschaya algebra i ee prilozheniya*, Ekaterinburg: UrGU, 2008.

[4] V. Lipsky, *Kombinatorika dlya programmistov*, M.: Mir, 1988.

Nikolay Aleksandrovich Gaydamakin
Ural Federal University,
pr. Lenina, 51,
620083, Ekaterinburg, Russia
E-mail address: haid2@bk.ru

Vitaly Anatolievich Baransky
Ural Federal University,
pr. Lenina, 51,
620083, Ekaterinburg, Russia
E-mail address: vitaly.baransky@urfu.ru