Московский Государственный Университет
имени М.В. Ломоносова
Российская Академия Наук
Международная Академия Технологических Наук
Российская Академия Естественных Наук

Интеллектуальные Системы.

Теория и приложения

ТОМ **29** ВЫПУСК 1 * **2025**

МОСКВА

УДК 519.95; 007:159.955 ISSN 2411–4448
ББК 32.81 Издаётся с 1996 г.

В журнале «Интеллектуальные системы. Теория и приложения» публикуются научные достижения в области теории и приложений интеллектуальных систем, новых информационных технологий и компьютерных наук.
Издание журнала осуществляется под эгидой МГУ имени М.В. Ломоносова, Научного Совета по комплексной проблеме «Кибернетика» РАН, Отделения «Математическое моделирование технологических процессов» МАТН.
Учредитель журнала: ООО «Интеллектуальные системы».
Журнал входит в список изданий, включенных ВАК РФ в реестр публикаций материалов по кандидатским и докторским диссертациям по математике и механике.

Индекс подписки на журнал: 64559 в каталоге НТИ «Роспечать».

Адрес редакции: 119991, Москва, ГСП-1, Ленинские Горы, д. 1, механико-математический факультет, комн. 12-01.
Адрес издателя: 115230, Россия, Москва, Хлебозаводский проезд, д. 7, стр. 9, офис 9. Тел. +7 (495) 939-46-37, e-mail: mail@intsysmagazine.ru

*) Прежнее название журнала: «Интеллектуальные системы».

ОГЛАВЛЕНИЕ

Часть 1
Общие проблемы теории интеллектуальных систем

Математические основы теоретической информатики. Теория нечёткой семантической информации о точке. Семантика естественного языка

А. В. Чечкин[1]

Цель. Исследовать семантику естественного языка.

Методы. Математическое моделирование. Системный и ультрасистемный анализ и синтез.

Результаты. Строится математическая модель семантики языковых конструкций, понятий естественного языка, математическая основа теоретической информатики. Определяется количественная мера семантической информации о точке. Делаются общие семантические выводы.

Ключевые слова: естественный язык и нечёткая семантическая информация о точке, семантика естественного языка.

1. Введение

Информатика – наука об информации, методах и процессах её сбора, хранения, анализа, синтеза, передачи, и использования в различных целях. Термин “информация” происходит от латинского слова “informatio”, что означает сведение или сообщение о чем-либо или о ком-либо. Для человека информация связана с его естественным языком и, следовательно, с его интеллектом. Кибернетика тоже - наука об информации. Разница здесь в предназначении информации. В ***кибернетике*** имеют дело с информацией для управления одним объектом, то есть с информацией, относящейся только к одной точке, которая в силу единственности не требует себе уникального семантического указателя. Кибернетика обслуживает системы автоматического регулирования. ***Информатика*** использует информацию принципиально о разных точках, что требует принципиально разных, автономных уникальных семантических указателей этих точек. У человека кибернетика отвечает за безусловные рефлексы первой сигнальной системы, за подсознательное регулирование биомеханизмами жизнеобеспечения человека. Информатика у человека отвечает за сознательное

[1] *Чечкин Александр Витальевич* — доктор физ.-мат. наук, профессор, Военная академия РВСН имени Петра Великого, Финансовый университет при правительстве Российской Федерации, e-mail: a.chechkin@mail.ru.

Chechkin Alexander Vitalievich — Doctor of Phys.-Math. Sci., Professor, Strategic Missile Forces Military Academy named after Peter the Great, Financial University under the Government of the Russian Federation.

когнитивное поведение человека, за условные (приобретённые) рефлексы, связанные в первую очередь с естественным языком, с обучением и со второй сигнальной системой человека (с языком), [1-2]. Информатика имеет дело с интеллектуальными системами, использующие информацию о различных точках.

Понятие "чёткой семантической информации о точке" было введено и изучено в [3-4]. Это понятие строится на базе теории фильтров и ультрафильтров А. Картана и только в рамках теории классических чётких множеств. Тогда как понятие "нечёткой семантической информации о точке" требует специальной формализации, о которой пойдет речь в настоящей статье.

2. Алгебра де Моргана нечётких множеств — основа семантики естественного языка

Определим *нечёткие множества* по L.A.Zade [5-6].

Определение 2.1. Нечёткое множество A определяется *функцией принадлежности*:

$$\mu_A : X \to [0;1] \tag{1}$$

Здесь X *опорное* или *универсальное* множество, из элементов которого определяется нечёткое множество $A = \{x : \mu_A(x) > 0, x \in X\}$. Функция принадлежности задает экспертную оценку принадлежности к нечеткому множеству A точек X. На прикладном языке нечёткое множество A определяется некоторым нечётким свойством, которым все точки A обладают, но в разной степени. Точки с $\mu_A(x) = 0, x \in X$ не принадлежат нечеткому множеству. Они *полностью* (*единогласно с точки зрения экспертной группы)* не обладают свойством A. Точки считаются принадлежащими нечеткому множеству в максимальной степени, если для них $\mu_A(x) = 1, x \in X$. Про такие точки говорят, что они *полностью или безоговорочно принадлежат A.* В этом случае подразумевается единогласное оценивание виртуальной экспертной группой точек из X на наличие у них некоторого нечёткого свойства A. Еще про такие точки говорят, что они образуют *ядро* A и пишут $KerA = \{x : \mu_A(x) = 1, x \in X\}$ Наличие ядра нечёткого множества является важным свойством нечёткого множества. Если в A имеется непустое ядро $KerA \neq \emptyset$ то нечёткое множество A называют *нормальным.* В противном случае нечёткое множество называется *ненормальным.* Наконец, про точки с дробными значениями функции принадлежности, у которых $0 < \mu_A(x) < 1, x \in X$, говорят, что точки принадлежат нечёткому множеству A *частично* и *образуют размытую границу нечёткого множества.* Другими словами, эти точки

обладают свойством A, но *с оговорками, неединогласно с точки зрения экспертной группы*. На практике размытая граница нечёткого множества часто вызвана объективным наличием некоторого *пограничного эффекта* нечёткого свойства A.

Все точки нечёткого множества образуют *носитель нечёткого множества* и пишут $SuppA = \{x : \mu_A(x) > 0, x \in X\}$

Согласно определению 2.1. обычные классические чёткие подмножества X так же относятся к нечётким подмножествам и составляют *подсемейство чётких подмножеств* в семействе всех нечётких подмножеств X. Для классических чётких подмножеств выполняется определяющее их равенство $KerA = SuppA$, еще они не имеют размытой границы.

Определение 2.2. В определении 2.1 нечёткого множества можно вместо экспертной оценки (1) перейти к вероятностной оценке выполнения у точек нечёткого свойства A, согласованной с данной функцией принадлежности (1), т.е. перейти к нормированной вероятностной мере некоторой воображаемой случайной величины, отражающей субъективную статистику экспертной группы оценивания свойства A. Заметим, что в математической статистике особо выделяют гауссовские процессы с высокими (большими) выбросами [7].

Для случая конечного или счетного опорного множества X исходов дискретное распределение плотности вероятности, соответствующее данной функции принадлежности Заде (1), определяется формулой

$$p_A(x) = \frac{\mu_A(x)}{\sum_{x \in X} \mu_A(x)}, x \in X, \sum_{x \in X} p_A(x) = 1 \tag{2}$$

Для случая континуального измеримого опорного множества исходов X вероятностная мера, соответствующая измеримой функции принадлежности (1), определяется формулой

$$p_A(x) = \frac{\mu_A(x)}{\int_X \mu_A(x)dx}, x \in X, \int_X p_A(x)dx = 1 \tag{3}$$

Пример 2.1. Рассмотрим устойчивое словосочетание естественного языка $A =$ *"Начало недели"*. ***Семантика*** этого лингвистического понятия является нечёткое подмножество A точек опорного множества X семи дней недели

$$X = \{\text{Пн}, \text{Вт}, \text{Ср}, \text{Чт}, \text{Пт}, \text{Сб}, \text{Вс}\}$$

Нечёткое множество A определяет объём лингвистического понятия *"Начало недели"*, который, в свою очередь определяется функцией принадлежности $\mu_A : X \to [0; 1]$. Пусть, например, функция принадлежности задана в табличном виде

$$\mu_A(x) = \{\frac{\mu_A(x)}{x} : x \in X\} = \{\frac{1}{\text{Пн}}, \frac{0.8}{\text{Вт}}, \frac{0}{\text{Ср}}, \frac{0}{\text{Чт}}, \frac{0}{\text{Пт}}, \frac{0}{\text{Сб}}, \frac{0}{\text{Вс}}\} \qquad (4)$$

Для конечного или счётного нечёткого множества (4) принята *польская запись* в форме суммы формальных дробе, отражающей табличное задание ее функции принадлежности (4). При этом в сумме члены с нулевыми значениями функции принадлежности принято опускать (их явно не писать, считая по умолчанию их присутствие)

$$A = \frac{1}{\text{Пн}} + \frac{0.8}{\text{Вт}} + \frac{0}{\text{Ср}} + \frac{0}{\text{Чт}} + \frac{0}{\text{Пт}} + \frac{0}{\text{Сб}} + \frac{0}{\text{Вс}} = \frac{1}{\text{Пн}} + \frac{0.8}{\text{Вт}} \qquad (5)$$

Конкретное значение функции принадлежности $\mu_A(\text{Вт}) = 0.8$ для вторника здесь отражает, например, некоторую частную статистику опроса: “Является ли вторник началом недели?”. В данном примере из каждых десяти опрошенных восемь считают вторник началом недели, а двое нет. Перейдём от функции принадлежности (5) к согласованной с (5) вероятностной (нормированной) мере (2). Тогда от экспертной оценки принадлежности точек к нечёткому множеству перейдём к случайному процессу отбора дней, относящихся к началу недели членами экспертной группы. Получим распределение вероятностей частной статистики, согласованной с (5)

$$p_{\text{Пн}} = \frac{1}{1+0.8} = \frac{5}{9}, p_{\text{Вт}} = \frac{0.8}{1+0.8} = \frac{4}{9}, p_{\text{Пн}} + p_{\text{Вт}} = 1$$

Здесь вероятности удовлетворяют полной группе событий и соответствуют некоторой частной статистике. Например, телефонному опросу людей, от которых просили: “Позвоните обязательно в начале недели”. Конечно, эта статистика частная и отражает особенность группы опрашиваемых. Семантика устойчивого слова сочетания “Начало недели” является нечёткой и представляет собой нечёткое множество (5). Здесь имеем ядро и носитель нечеткого множества

$KerA = \{\text{Пн}\}$, $SuppA = \{\text{Пн}, \text{Вт}\}$ Размытой границей этого нечеткого множества является множество $\{\text{Вт}\}$

Заметим, что в примере 2.1. опорное множество X является чётким и задаётся списком дней. Оно может рассматриваться как нечёткое множество с функцией принадлежности, совпадающей с характеристической функцией множества X, т.е. равной 1 во всех точках чёткого множества (6) или равномерным распределением вероятностей соответствующего случайного процесса $p_X(x) \equiv \frac{1}{7}$

$$A = \frac{1}{\text{Пн}} + \frac{1}{\text{Вт}} + \frac{1}{\text{Ср}} + \frac{1}{\text{Чт}} + \frac{1}{\text{Пт}} + \frac{1}{\text{Сб}} + \frac{1}{\text{Вс}} \qquad (6)$$

У чётких множеств справедливо равенство $KerX = SuppX$ и нет размытых границ.

Утверждение 2.1. Если некоторое свойство A проявляется для точек опорного множества исходов X с заранее известной плотностью распределения вероятности $p_A(x) > 0, x \in X$ задающей вероятностную меру на X то существует экспертная оценка для виртуальной экспертной группы, согласованная с данным распределением плотности вероятности, в форме функции принадлежности (1) $\mu_A : X \to [0;1]$ следующего вида:

$$\mu_A(x) = \frac{p_A(x)}{\sup_A p_A(x)}, x \in X \tag{7}$$

Доказательство. Функция принадлежности (7) безусловно (очевидно) отражает экспертную оценку нечеткого свойства A соответствующую данной статистике с плотностью распределения $p_A(x) > 0, x \in X$
Определение 2.3. Пусть имеются три нечётких множестваA, B и C заданные функциями принадлежности $\mu_A : X \to [0;1]$, $\mu_B : X \to [0;1]$ и $\mu_C : X \to [0;1]$. Тогда будем говорить и писать следующее.

1. *Отрицанием* или *дополнением* нечёткого множества A до X называется нечёткое множество $\bar{A}$ с функцией принадлежности вида

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x), x \in X \tag{8}$$

2. Нечёткое множество C называется *объединением* нечётких множеств

A, B и пишут $A \cup B = C$ если

$$\mu_C(x) = \sup_{x \in X}\{\mu_A(x); \mu_B(x)\} \tag{9}$$

3. Нечёткое множество C называется *пересечением* нечётких множеств A, Bи пишут $A \cap B = C$ если

$$\mu_C(x) = \inf_{x \in X}\{\mu_A(x); \mu_B(x)\} \tag{10}$$

4. Нечёткое множество A называется *надмножеством* для нечёткого множества B или нечёткое множество B называется *подмножеством* нечёткого множества A и пишут $A \supseteq B$ если

$$\mu_A(x) \geq \mu_B(x), x \in X \tag{11}$$

Замечание. В случае строгого включения нечётких подмножеств, когда $A \supset B$, $\mu_A > \mu_B$ и $SuppA = SuppB$, выполняется $A \cup B = A$, $A \cap B = B$.

Пример 2.2. Пусть нечёткое множество A и опорное чёткое множество X будут те же самые, что в примере 2.1. Тогда имеем, согласно (11)

$A \subseteq X$ что соответствует семантике понимания смысла: "*неделя*" $= X$ является надмножеством для $A =$ "*начало недели*". Найдем отрицание $\bar{A} =$ "*не начало недели*". Согласно (5) и (8) имеем нечёткое (ненормальное) множество

$$\bar{A} = \frac{0}{\text{Пн}} + \frac{0.2}{\text{Вт}} + \frac{1}{\text{Ср}} + \frac{1}{\text{Чт}} + \frac{1}{\text{Пт}} + \frac{1}{\text{Сб}} + \frac{1}{\text{Вс}} \tag{12}$$

Здесь нечёткое множество является семантикой нечёткого лингвистического устойчивого словосочетания "*Не начало недели*". Найдем теперь объединение двух нечётких множеств $\bar{A}, A$. Согласно (5), (9) и (12) будем иметь семантику устойчивого словосочетания естественного языка "Начало или не начало недели":

$$A \cup \bar{A} = \frac{1}{\text{Пн}} + \frac{0.8}{\text{Вт}} + \frac{1}{\text{Ср}} + \frac{1}{\text{Чт}} + \frac{1}{\text{Пт}} + \frac{1}{\text{Сб}} + \frac{1}{\text{Вс}}; \text{Здесь} A \cup \bar{A} \neq X \tag{13}$$

где $\mu_{A \cup \bar{A}}(\text{Вт}) = \sup\{\mu_A(\text{Вт}); \mu_A(\text{Вт})\} = \sup\{0,8; 0,2\} = 0,8.$

Найдем пересечение нечётких множеств $\bar{A}; A$ Согласно (5), (10) и (12) будем иметь семантику устойчивого словосочетания естественного языка "Начало и не начало недели":

$$A \cap \bar{A} = \frac{0}{\text{Пн}} + \frac{0.2}{\text{Вт}} + \frac{0}{\text{Ср}} + \frac{0}{\text{Чт}} + \frac{0}{\text{Пт}} + \frac{0}{\text{Сб}} + \frac{0}{\text{Вс}}; \text{Здесь} A \cap \bar{A} \neq \emptyset \tag{14}$$

Вывод. Формулы (13) и (14) иллюстрируют нарушение "Закона исключенного третьего", который в логике Моргана нечётких множеств не имеет место. Тогда как в булевой алгебре чётких множеств "Закон исключенного третьего" справедлив, там имеют место равенства

$$A \cup \bar{A} = X \text{и} A \cap \bar{A} = \emptyset \tag{15}$$

означающие: "Третьего не дано". В алгебре де Моргана нечётких множеств этот закон не выполняется. Интересно, что закон нарушается в точках размытой границы нечётких множеств $\bar{A}; A$, (5), (12), (13), (14).

Утверждение 2.2. Об алгебре и решётке нечётких множеств. Пусть X — чёткое опорное множество. Рассмотрим семейство всех нечётких подмножеств $A \subseteq X$, которое определяется семейством всех функций принадлежности $\mu_A : X \to [0; 1]$. В семействе всех нечётких подмножеств рассмотрим три алгебраические операции. Две бинарные операции, *объединение* подмножеств (9) и *пересечение* подмножеств (10), а так же одну унарную операцию, *отрицание* или *дополнение* нечёткого подмножества до X (8). В этом семействе всех нечётких подмножеств рассмотрим частичный порядок (11).

Тогда семейство нечётких подмножеств будет образовывать *алгебру Моргана*, в которой выполняются все законы алгебры Буля чётких подмножеств кроме одного, "Закона исключенного третьего", т.е. будет

$$A \cup \bar{A} \neq X \text{и} A \cap \bar{A} \neq \emptyset \tag{16}$$

Семейство нечётких подмножеств является дистрибутивной решёткой частичного порядка (11).

Доказательство. Докажем первый закон де Моргана для любых двух нечётких подмножеств $A\bar{\cup}B = \bar{A}\cap\bar{B}$. Перейдем к фукциям принадлежности. Для любых двух действительных чисел, $\mu_A(x); \mu_B(x), x \in X$ имеем очевидно

$$\begin{aligned} \mu_{A\bar{\cup}B}(x) = 1 - \mu_{A\cup B} = 1 - \sup\{\mu_A(x); \mu_B(x)\} = \\ = \inf\{1 - \mu_A(x); 1 - \mu_B(x)\} = \mu_{\bar{A}\cap\bar{B}} \end{aligned} \tag{17}$$

Аналогично можно доказать все другие законы алгебры Моргана для нечётких множеств. Докажем дистрибутивность решетки нечётких множеств. Для любых трех действительных чисел $\mu_A(x); \mu_B(x); \mu_C(x)$, $x \in X$ легко проверяется перебором равенство

$$\begin{aligned} \mu_{A\cup(B\cap C}(x) = \sup\{\mu_A(x); \inf\{\mu_B(x); \mu_C(x)\}\} = \\ = \inf\{\sup\{\mu_A(x); \mu_B(x)\}; \sup\{\mu_A(x); \mu_C(x)\}\} = \\ = \inf\{\mu_{A\cup B}(x); \mu_{A\cup C}(x)\} = \mu_{(A\cup B)\cap(A\cup C)}(x), x \in X\square \end{aligned}$$

3. Нечёткие сведения и нечёткая информация о точке

Сформулируем с информационной точки зрения факт принадлежности точки $x_0 \in X$ нечёткому множеству $A \subseteq X$, т.е. определим нечёткое сведение о точке $x_0 \in A \subseteq X$.

Определение 3.1. Пусть задано нечеткое множество $A \subseteq X$ опорного множества X с функцией принадлежности $\mu_A : X \to [0; 1]$. Рассмотрим факт принадлежности $x_0 \in A \subseteq X$. Будем этот факт называть *нечётким сведением о точке* $x_0 \in X$ и записывать любой из двух равносильных формул

$$(\mu_A)A(x_0) = (p_A)A(x_0), x_0 \in X \tag{19}$$

Здесь функция принадлежности $\mu_A(x_0)$ выражает экспертную оценку сведения, а функция плотности вероятности $p_A(x_0)$ вероятность (достоверность) сведения, предикатный символ A в (19) совпадает с обозначением нечёткого множества A и, следовательно, совпадает с обозначением нечёткого свойства A точек этого множества.

Утверждение 3.1. Корректности определения нечёткого сведения о точке и его нечёткой семантики.

Определение 3.1 и формула (17) корректно определяют понятие нечёткого сведения о точке $x_0 \in X$ и семантики словосочетания "точка $x_0 \in X$ обладает нечётким свойством A"

Доказательство. Сопоставим (19) с формализацией обычного чёткого сведния о точке $x_0 \in \delta = A \subseteq X$ вида $(1)\delta(x) = \delta(x_0)$, где $\delta = A$ — чёткое подмножество [3-4]. Так как чёткое сведение является частным случаем нечёткого сведения с функцией принадлежности, совпадающей с характеристической функцией чёткого множества A, то имеем

$$\mu_A(x) = \chi_A(x) = \chi_\delta(x); (\mu_A)A(x_0) = (1)A(x_0) = \delta(x_0), x_0 \in X \qquad (20)$$

Это полностью согласует определение 2.4 и формулу (19) с определением чёткого сведения [3-4]. □

Определение 3.2. Семейство нечётких сведений *о точке* $x_0 \in X$ опорного множества X назовём нечёткой информацией о точке $x_0 \in X$ и будем это записывать в любой из двух равносильных формул

$$\{(\mu_A)A(x_0) = \{(p_A)A\}(x_0) = \mathfrak{I}(x_0), x \in X, \text{если} \qquad (21)$$

$\mathfrak{I} = \{(\mu_A)A\} = \{(p_A)A\}$ – фильтр нечётких подмножеств, содержащих точку $x_0 \in X$. При этом базис фильтра, подсемейство $\mathfrak{B} \subseteq \mathfrak{I}$, определяет носитель $\mathfrak{B}(x_0)$ информации $\mathfrak{I}(x_0)$.

Пример 3.1. Рассмотрим устойчивые словосочетания естественного языка, относящиеся к бинарной иерархической классификации живых систем шведского естествоиспытателя Карла Линнея или же устойчивые словосочетания языка, относящиеся к периодическому закону химических элементов русского химика Дмитрия Ивановича Менделеева. Например, такие устойчивые словосочетания как "медведь бурый", "человек разумный Homo Sapiens)" или "берёза белая" имеют общепризнанную мировую научную семантику (описательные биологические особенности строения или научно подробные ДНК-коды). Аналогично, устойчивые словосочетания "Закон Менделеева о зависимости свойств веществ от их атомного веса", "Благородные газы". Семантика этих общепринятых во всем мире научных словосочетаний известна, доступна и всегда очень интересна.

4. Заключение

Уточним базовые для теоретической информатики определения “фильтра” и “базиса фильтра” для нечётких подмножеств X. Фильтром нечётких подмножеств называется такое семейство нечётких подмножеств, для которого выполняются три аксиомы:

1) *Каждое подмножество фильтра непустое* $A \neq \emptyset$

2) *Пересечение любых двух подмножеств фильтра принадлежит фильтру* $A \cap B \in \mathfrak{I}$

3) *Каждое надмножество* $B \supset A$ любого нечёткого подмножества $A \in \mathfrak{I}$ принадлежит фильтру $B \in \mathfrak{I}$

Наконец, базисом $\mathfrak{B}$ *фильтра* $\mathfrak{I}$ называется часть фильтра $\mathfrak{B} \subseteq \mathfrak{I}$, в которой по любому подмножеству $A \in \mathfrak{I}$ найдётся подмножество $B \in \mathfrak{B}$ такое, что $B \subseteq A$. *Любой базис фильтра однозначно определяет свой фильтр.*

Сделаем общие семантические выводы.

Уточним понятие “семантика естественного языка”. В философии, в семиотике, в рамках концепции “треугольника Фреге” семантикой слова (знака) называют контент (сигнификат, смысл) и противопоставляют её денотанту (референту). В теоретической информатике под семантикой сведения о точке понимают принесённую этим сведением информацию о точке. Выделим два методологических принципа описания семантики словарных слов-понятий. Принцип неединственности описания (Семантика, взятая из разных словарей) и принцип дополнительных описаний (Семантика дополнений из разных словарей). Термины “устойчивое словосочетание или устойчивое слово-понятие” подчёркивают только широкую, именно всеми признанную их семантику. Заметим, что у человека в центральной нервной системе имеются две сигнальные системы сенсориума. Первая система — это когда его рецепторы органов чувств воспринимают сигналы от естественных объектов или процессов природы и далее происходит подсознательное реактивное жизнеобеспечивающее поведение человека. Вторая система — это когда его рецепторы органов чувств воспринимают сигналы от искусственных символов, языковых и происходит восприятие семантики этих языковых символов как подсознательного сенсорного образа точки первичной сигнальной системы и далее когнитивное целенаправленное поведение человека, требующее использования разнообразных сведений и знаний о разных точках (объектах и процессах). При этом ведущей в семантике слов и словосочетаний естественного языка

выступают подсознательные образы первой сигнальной системы (первичного сенсориума), для описания которой используется общепризнанная чёткая или нечёткая семантика устойчивых слов-понятий и устойчивых словосочетаний естественного языка, например, частная относительно экспертной группы (явной или воображаемой). В обеих сигнальных системах человека ведущую роль играют специфические нейроны — аттракторы (сборщики) сведений только об одной точке. Например, так называемые, нейроны "Моей бабушки", "Моего дома" и т.п. [1-2]. Отметим важную особую роль в когнитивных свойствах естественного языка семантического указателя точки или уникального собственного имени точки (x_0). Именно, благодаря указателю точки, происходит процесс выделения целостного восприятия той или иной сущности (некоторого объекта или некоторого процесса). Происходит сборка частностей в единое целое. Это подчёркивает объективную системную структуру мира природы.

Список литературы

[1] И.П. Павлов, *Лекции о работе больших полушарий головного мозга. Полное собрание трудов в 5-и т.: Т.1, 2, 5.*, изд. "Культура", М., 1973.

[2] Г.С. Воронков, А.В. Чечки, "Проблемы моделирования сенсориума и языковой системы естественного интеллекта индивидуума", *Интеллектуальные системы*, **2**:1–4 (1997), 35–54.

[3] А.В. Чечкин, *Математическая информатика*, Наука, М., 1991.

[4] А.В. Чечкин, "Математические основы теоретической информатики. Теория чёткой сематической информации о точке", Труды XI Международной научно-практической конференции: "Современная математика и концепции инновационного математического образования", **11**:1 (2024), 249–271.

[5] L. Zade, "Fuzzy Sets", *Information and Control*, **8** (1965), 338–353.

[6] Л.А. Заде, *Понятие лингвистической переменной и его применение к принятию приближённых решений*, Мир, М., 1976.

[7] И.А. Козик, В.И. Питербарг, "Большие выбросы гауссовских нестационарных процессов в дискретном времени", *Фундаментальная и прикладная математика*, **22**:2 (2018), 159–169.

Mathematical Foundations of Theoretical Computer Science (Informatics). The Theory of Fuzzy Semantic Information About a Point. Semantics of Natural Language

Introduction.

In cybernetics, they deal with information for managing a single object, that is, with information related to only one point, which, due to its uniqueness, does not require a unique semantic pointer. Cybernetics serves automatic control systems. **Computer science**(**Informatics**) uses information about fundamentally different points, which requires fundamentally different, autonomous, unique semantic pointers for these points. In humans, cybernetics is responsible for the unconditioned reflexes of the first signaling system, for the subconscious regulation of human life support biomechanisms. Computer science in humans is responsible for conscious cognitive behavior of humans, for conditioned (acquired) reflexes associated primarily with natural language, with learning and with the second human signaling system (with language), [1-2]. Computer science deals with intelligent systems that use information about various points. The concept of "clear semantic information about a point" was introduced and studied in [3-4]. This concept is based on the theory of filters and ultrafilters by A. Cartan and only within the framework of the theory of classical distinct sets. Whereas the concept of "fuzzy semantic information about a point" requires special formalization, which will be discussed in this article.

Conclusion.

Let's draw general semantic conclusions. Let us clarify the concept of "natural language semantics". In philosophy, in semiotics, within the framework of the concept of the "Frege triangle", the semantics of a word (sign) is called content (signification, meaning) and is opposed to its denotation (the referent). In theoretical computer science, the semantics of information about a point is understood as the information about a point brought by this information. Let us single out two methodological principles for describing the semantics of dictionary words-concepts. The principle of non-uniqueness of descriptions (Semantics taken from different dictionaries) and the principle of additional descriptions (Semantics of additions from different dictionaries). The terms "stable phrase or stable word-concept" emphasize only their broad, universally recognized semantics. Note that humans have two sensory signaling systems in their central nervous system. The first system is when his sensory receptors perceive signals from natural objects or natural processes, and then subconscious reactive life–sustaining behavior occurs. The second system is when his sensory receptors perceive signals from artificial symbols, language symbols, and the semantics of these language symbols are perceived as a subconscious sensory image of a point in the primary signaling system, followed by cognitive purposeful human behavior that requires the use of a variety of information and knowledge about different points (objects and processes). At the same time, the subconscious images of the first signaling system (primary sensorium) are the leading ones in the semantics of words and phrases of natural language, for which the

generally recognized clear or fuzzy semantics of stable words-concepts and stable phrases of natural language is used, for example, a private relative to an expert group (explicit or imaginary). In both human signaling systems, specific neurons play a leading role - attractors (collectors) of information about only one point. For example, the so-called neurons of "My grandmother", "My house", etc. [1-2]. Note the important special role of the semantic pointer of a point or the unique proper name of a point (x_0) in the cognitive properties of natural language. It is thanks to the dot pointer that the process of highlighting the holistic perception of an entity (some object or some process) takes place. The details are assembled into a single whole. This highlights the objective systemic structure of the natural world.

Keywords: almost complete predicting, predicting machine, prediction of superwords by a machine, criterion for predicting.

References

[1] I. P. Pavlov, *Lectures on the work of the cerebral hemispheres. Complete works in 5 volumes: Vol. 1, 2, 5*, Publishing house Culture, Moscow, 1973 (In Russian).

[2] G. S. Voronkov, A. V. Chechkin, "Problems of modeling the sensorium and the language system of an Individual's natural intelligence", *Intellektual'nye sistemy*, **2**:1–4 (1997), 35–54 (In Russian).

[3] A. V. Chechkin, *Mathematical computer science*, Nauka, Moscow, 1991 (In Russian).

[4] A. V. Chechkin, "Mathematical foundations of theoretical computer science. The theory of clear semantic information about a point", Proceedings of the XI International Scientific and Practical Conference: "Modern mathematics and concepts of innovative mathematical education", **11**:1 (2024), 249–271 (In Russian).

[5] L. Zade, "Fuzzy Sets", *Information and Control*, **8** (1965), 338–353.

[6] L. Zade, *The concept of a linguistic variable and its application to making approximate decisions*, Mir, Moscow, 1976 (In Russian).

[7] I. A. Kozik, V. I. Peterbarg, "Large outliers of Gaussian nonstationary processes in discrete time", *Fundamental and applied mathematics*, **22**:2 (2018), 159–169 (In Russian).

Часть 2
Специальные вопросы теории интеллектуальных систем

Согласование формул Вильсона и уравнения состояния для расчета фазового равновесия нефти и газа

Е. В. Колдоба[1]

При расчете фазового равновесия многокомпонентных углеводородных растворах используются итерационные методы. Для вычисления начальных приближений применяется формула Вильсона, а при вычислении химических потенциалов или летучестей используется уравнение состояния. В работе предлагается способ настройки формулы Вильсона на используемое уравнение состояния. Показано, что коэффициенты рассогласования для тяжелых компонент больше, чем для легкий. Подход позволяет построить термодинамически согласованную систему, и устранить некоторые нефизические решения и численные неустойчивости.

Ключевые слова: уравнение состояния, формула Вильсона, фазовое равновесие, многокомпонентные растворы, нефть, численное моделирование.

1. Введение

Для прогнозирования разработки нефтяных и газовых месторождений широко применяются методы численного моделирования. Расчеты могут идти от нескольких часов до нескольких месяцев в зависимости от сложности задачи и значительную часть этого времени занимает расчет фазового равновесия. Многие численные проблемы, возникающие на счете также связаны с термодинамической частью задачи.

Природные газы и нефть — это многокомпонентные растворы. Фазовые переходы в них имеют ряд особенностей: переход начинается при одном давлении, а заканчивается при другом, и в процессе непрерывно меняется компонентный состав фаз. Эта особенность значительно усложняет расчеты фазового равновесия растворов, а именно: константы фазового равновесия K_i или K-values (отношение концентраций для каждой компоненты в газе и жидкости) превращаются в сложные функции давления, температуры и состава. Для каждой компоненты многокомпонентного раствора по уравнению состояния вычисляются химические

[1] *Колдоба Елена Валентиновна* — доцент каф. вычислительная механика мех.-мат. ф-та МГУ, e-mail: elenakoldoba@mail.ru.

Koldoba Elena Valentinovna — Associate Professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Computational Mechanics.

потенциалы (или летучести) в газе и жидкости, записывается их равенство. Получившуюся систему нелинейных алгебраических уравнений решают итерационными методами, например, методом Ньютона [1-3], сходимость которых иногда проблематична.

Исследования численных неустойчивостей показали, что возможны случаи, в которых уравнение состояния и функции K_i, задающие начальные концентрации в фазах, оказываются рассогласованы настолько, что это приводит к численным неустойчивостям на счете и нефизическим решениям. Так, например, тяжелая нефть при нормальных условиях может при расчете вся оказаться в газовом состоянии и т.д.

Для повышения надежности расчетов фазового равновесия обычно предлагаются разные виды функций K_i [4-5], используются разные уравнения состояния [6-7]. Но это может помочь в одном диапазоне термобарических условий и ухудшить ситуацию в другом. В данной работе обращается внимание на необходимость согласования функций K_i с используемым уравнением состояния, предлагается способ настройки, демонстрируются эффекты рассогласованности. В качестве примера выполнено согласование уравнения состояния Редлиха-Квонга и формулы Вильсона, которые часто используются в современных гидродинамических симуляторах при расчете фазового равновесия на нефтяных и газовых месторождениях.

2. Постановка задачи

Пусть N-компонентный раствор с полной молярной концентрацией (молярной долей) $\{z_i\}$ находится в двухфазном состоянии, разделившись на газ с концентрацией $\{y_i\}$ и жидкость с концентрацией $\{x_i\}$, где i– номер компоненты, $i = 1, 2...N$. Для смеси и каждой фазы в отдельности для молярных концентраций (молярных долей) должны выполняться следующие условия:

$$\sum_i^N z_i = 1, \qquad \sum_i^N x_i = 1, \qquad \sum_i^N y_i = 1$$

Коэффициенты распределения или константы фазового равновесия (K-values) связаны с концентрациями i-ой компоненты в газовой и жидкой фазах следующим образом:

$$K_i = y_i/x_i$$

3. Уравнение состояния Редлиха-Квонга (РК)

Часто используется в современных гидродинамических симуляторах для расчета фазового равновесия N-компонентных углеводородных растворов. Уравнение РК, задающее связь температуры T, давления P, молярного объема V, имеет вид:

$$P = \frac{RT}{V-b} - \frac{a}{\sqrt{T}V(V+b)} \tag{1}$$

где R – газовая постоянная; a, b – параметры, зависящие от критической температуры T_{ci}, критического P_{ci} , ацентрического фактора ω_i каждой i-ой компоненты и их концентраций. Процедуры расчета a, b для растворов хорошо известны [1].

Зная уравнение состояния, можно вычислить химические потенциалы μ_i или летучести i-ой компонента в каждой из фаз, записать их равенство для каждой компоненты в газе и жидкости и получить систему из N нелинейных алгебраических уравнений для расчета фазового равновесия:

$$\mu_{i,G}(T, P, \{y_i\}) = \mu_{i,L}(T, P, \{x_i\}) \qquad i = 1, 2...N,$$

где индексами G и L отмечены величины, относящиеся к газовой и жидкой фазам.

Для практических применений обычно используют равенство летучестей, записанных через коэффициенты летучести φ_i и функции K_i, и решают следующую систему:

$$K_i = \frac{\varphi_{i,L}(T, P, \{x_i\})}{\varphi_{i,G}(T, P, \{y_i\})} \qquad i = 1, 2...N \tag{2}$$

Для расчета фазового равновесия N-компонентного раствора при заданных давлении и температуре необходимо решить систему уравнений (2) и уравнение Рэчфорда-Райса для определения F_V–молярной доли газовой фазы[8]:

$$\sum_i^N \frac{(K_i - 1)z_i}{1 + (K_i - 1)F_V} = 0 \tag{3}$$

Уравнение (3) решается численно итерационными методами [1-3]: на первом шаге используются начальные приближения K_i, затем на каждой итерации находится значение F_V, потом по известным формулам вычисляются равновесные концентрации в газе y_i и в жидкости x_i, находятся новые значения K_i, которые подставляются в уравнение (3), и так далее до выполнения некоторых условий сходимости. Как и для любых итерационных методов, результат и сходимость алгоритма зависит от выбора начального приближения.

4. Формула Вильсона

В современных гидродинамических симуляторах для решения уравнения (3) начальные приближения K_i часто задаются по формуле Вильсона [7]:

$$K_i = \frac{P_{ci}}{P} exp\left(5.31(1+\omega_i)\left(1-\frac{T_{ci}}{T}\right)\right) \tag{4}$$

Согласно этой формуле каждая функция K_i зависит только от характеристик i-ой компоненты: критической температуры T_{ci}, критического давления P_{ci} , ацентрического фактора ω_i и не зависит от концентраций и свойств других компонент. Это связано с тем, что при выводе формулы использовались свойства идеальных растворов. Понятно, что это приближение лучше работает для легких компонент, полученные результаты подтверждают это. Например, при выводе формулы (4) использовалось приближение для идеальных растворов, следующего вида [7]:

$$K_i = \frac{P_{si}}{P}, \tag{5}$$

где P_{si} – давление насыщения (давление фазового перехода жидкость-газ) чистой i-ой компоненты. Т.е используя формулу (5), мы тем самым задаем значение P_{si}. Это же давление для каждой чистой компоненты можно найти из уравнения состояния РК, но полученное значение P_{si} будет несколько другим. Пусть P_{si} – это давление насыщения i-ой компоненты, рассчитанное из уравнения РК по правилу Максвелла, а P'_{si} – давление насыщения, полученное по формуле (4), их отношение через коэффициент $\gamma_i = P_{si}/P'_{si}$.

5. Рассогласованность значений давления фазового перехода

По уравнению состояния РК вычислялись P_{si} для ряда углеводородных компонент (метан, этан... декан) и сравнивались со значением P'_{si}. Проведенные вычисления показали, что для метана значения совпадают с хорошей точностью, поэтому согласование не требуется. Для более тяжелых компонент рассогласование становилось все значительнее по мере увеличения углеродного числа молекулы (для декана рассогласование максимальное). Так для декана при $t = 0^oC$ величины отличаются на 736% (см. таблицу 1).

6. Процедура согласования

Рассматривался следующий способ настройки функций K_i на уравнение РК: коэффициент рассогласования γ_i добавляется в формулу (4). Способ корректировки оказался эффективным во всей области фазовых переходов,.

Следует подчеркнуть, что для изотермического случая коэффициенты γ_i нужно рассчитывать только один раз, т.к. они не зависят от давления и концентраций, а зависят только от используемого уравнения состояния и температуры.

$\gamma_1(C1)$	$\gamma_2(C6)$	$\gamma_3(C10)$
1.0	2.2	7.36

Таблица 1. Коэффициенты γ_i для метана (C1), гексана (C6) и декана(C10) при температуре $t = 0^oC$.

В таблице 1. представлены рассчитанные коэффициенты рассогласования γ_i для метана (C1), гексана (C6) и декана(C10) при температуре $t = 0^oC$. Представленные данные показывают, что согласование необходимо проводить при наличии средних и тяжелых компонент в растворе (чем тяжелее, тем больше поправочный коэффициент), и не нужно проводить для метана. Таким образом формула Вильсона, согласованная с уравнением состояния, имеет вид:

$$K_i = \gamma_i \frac{P_{ci}}{P} exp\left(5.31(1+\omega_i)\left(1-\frac{T_{ci}}{T}\right)\right) \tag{6}$$

Следует заметить, что существуют разные корректировки формулы Вильсона, например, делающие её более точной для описания свойств реальных, а не идеальных растворов. Но в данной работе поправочные коэффициенты устраняют именно рассогласованность формул Вильсона и используемого уравнения состояния, что необходимо для расчета фазового равновесия итерационными методами.

7. Фазовые диаграммы «давление-состав»

Известно, что зная функции $\{K_i\}$ можно построить фазовые диаграммы. На Рис.1 представлены три фазовые диаграммы «давление-состав» двухкомпонентного раствора C6–C10 для изотермического случая, на которых можно наглядно продемонстрировать эффекты рассогласования. Так на Рис.1 двумя пунктирными линиями (W'_L, W'_G) обозначена диаграмма, построенная по формуле Вильсона; точечными линиями (W_L, W_G) —

по формуле Вильсона с поправкой; а сплошными линиями (C_L, C_G) — «точное» численное решение. Хорошо видно, что фазовая диаграмма, построенная по формуле с поправкой значительно ближе к «точному» решению, поэтому можно сделать вывод, что формула с поправкой дает лучшее начальное приближение для итерационных методов расчета фазового равновесия.

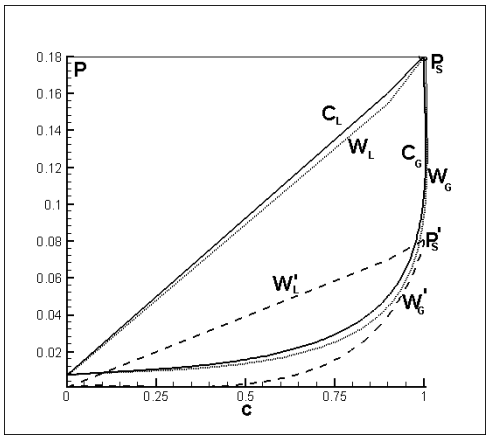

Рис. 1. фазовые диаграммы «давление-состав» двухкомпонентного раствора C6–C10 для изотермического случая.

8. Заключение

Таким образом, для численного расчета фазового равновесия многокомпонентного раствора с помощью уравнения состояния итерационным методом формула Вильсона должна быть согласована с этим уравнением. Для рассматриваемых примеров модифицированная формула Вильсона настолько хорошо приближает функции K_i к точным значениям, что в некоторых случаях не требуется дальнейшее итерационное уточнение K_i. Предлагаемый подход актуален для расчета фазового равновесия растворов, содержащих средние и тяжелые углеводородные компоненты (тяжелее гексана), т.к. чем тяжелее компонента, тем больше поправочный коэффициент. Выполненное согласование формул и уравнения состояния помогает устранить некоторые численные неустойчивости на счете и нефизические решения.

Список литературы

[1] Брусиловский А.И., *Фазовые превращения при разработке месторождений нефти и газа*, Грааль, Москва, 2002, 575 с.

[2] Michelsen M.L., "Calculation of Phase Envelopes and Critical Points for Multicomponent Mixtures", *Fluid Phase Equilibria*, **4**:1 (1980), 1–10

[3] Michelsen M.L., "The Isothermal Flash Problem. Part.I. Stability", *Fluid Phase Equilibria*, **9**:1 (1982), 1–19

[4] Almehaideb R.A., Ashour I., El-Fattah K.A., "Improved K-value correlation for UAE crude oil components at low pressures using PVT laboratory data.", *Fuel*, **82** (2003), 1057–1065

[5] Колдоба Е.В., "Эффективный термодинамически согласованный подход для численного моделирования процессов вытеснения нефти", *Математическое моделирование*, **10** (2009), 7–12.

[6] Whitson C. H, Brulé M. R., *Phase behavior*, SPE, Richardson, Texas, 2000, 240 с.

[7] Wilson G.A., "A modified Redlich–Kwong EOS. Application to general physical data calculations.", American Institute of Chemical Engineers 65th National Meeting, 1968, Paper No. 15C.

[8] Rachford H. H. Jr, Rice J. D., "Procedure for Use of Electronic Digital Computers in Calculating Flash Vaporization Hydrocarbon Equilibrium", *Journal of Petroleum Technology*, **4**:10 (1952), 19.

Reconciliation of Wilson formulas and the equation of state for calculating the phase equilibrium of oil and gas

Koldoba E.V.

Iterative methods are used to compute phase equilibrium in multicomponent hydrocarbon solutions. Wilson's formula is taken to calculate initial approximations, and the equation of state is taken to calculate chemical potentials or volatility. The paper proposes a method for adjusting Wilson's formula to the equation of state used. It is shown that the consistent coefficients for heavy components are greater than for light ones. The approach allows one to construct a thermodynamically consistent system and eliminate some non-physical solutions and numerical instabilities.

Keywords: equations of state, Wilson's formula, compution of phase equilibrium, multicomponent solutions, oil, numerical modeling.

References

[1] Brusilovskiy A.I., *Phase transformations in the development of oil and gas fields*, Graal, Moscow, 2002 (In Russian), 575 pp.

[2] Michelsen M.L., "Calculation of Phase Envelopes and Critical Points for Multicomponent Mixtures", *Fluid Phase Equilibria*, **4**:1 (1980), 1–10

[3] Michelsen M.L., "The Isothermal Flash Problem. Part.I. Stability", *Fluid Phase Equilibria*, **9**:1 (1982), 1–19

[4] Almehaideb R.A., Ashour I., El-Fattah K.A., "Improved K-value correlation for UAE crude oil components at low pressures using PVT laboratory data.", *Fuel*, **82** (2003), 1057–1065

[5] Koldoba E.V., "An efficient thermodynamically consistent approach for numerical modeling of oil displacement processes", *Matematicheskoe Modelirovanie*, **10** (2009), 7–12 (In Russian)

[6] Whitson C. H, Brulé M. R., *Phase behavior*, SPE, Richardson, Texas, 2000, 240 pp.

[7] Wilson G.A., "A modified Redlich–Kwong EOS. Application to general physical data calculations.", American Institute of Chemical Engineers 65th National Meeting, 1968, Paper No. 15C.

[8] Rachford H. H. Jr, Rice J. D., "Procedure for Use of Electronic Digital Computers in Calculating Flash Vaporization Hydrocarbon Equilibrium", *Journal of Petroleum Technology*, **4**:10 (1952), 19.

Empirical study of manifold learning techniques on forecasting stock price trend

Xinghao Niu[1]

In this work, five popular manifold learning techniques, PCA, ISOMAP, Locally Linear Embedding, Laplacian Eigenmaps and t-SNE, are examined on improving prediction accuracy of stock price trend. Effect of examined manifold learning techniques on classification and clustering task is proved to be different. Examined techniques tend to often worsen performance in clustering task. In classification task, observed improvement by all methods is slight, usually less than 1 percent. And only Laplacian Eigenmaps can more often stably improve classification accuracy at all number of components while other methods can't. Experiment results also suggest that there is no general effective technique for different stock price data set.
Keywords: dimension reduction, manifold learning, stock price trend, classification, clustering, neural network, k-means

1. Introduction

Real-world data usually has high dimensionality. In order to handle such real-world data adequately, its dimensionality needs to be reduced [3]. Manifold learning techniques are used to project original higher dimensional data to lower dimensional manifold by keeping structure of original dataset. It's also reasonable to represent financial market data as high-dimensional data. So, reducing high dimensionality of financial market data can be important for improving prediction accuracy.

State of the art manifold learning techniques include PCA (Principal Component Analysis), Locally Linear Embedding, multidimensional scaling (MDS) [11], diffusion maps, ISOMAP, Laplacian Eigenmaps, Maximum Variance Unfolding, t-SNE and autoencoder. Principal Component Analysis ([8, 7]) is aimed to project original high dimensional data to lower dimensional manifold based on main directions of datapoints. ISOMAP [10] determines neighborhood information on the manifold to obtain a weighted graph for data points based on certain distance measurement. Graph is then updated by calculating geodesic distances among data points. Lower-dimensional embedding is determined by solving optimization problem. Disadvantage of ISOMAP and MDS is higher computation requirement. Locally Linear Embedding [12] generally consists of three steps. First step is selecting

[1]Niu Xinghao — Ph.D. student, Department of economic informatics, Faculty of Economics, Lomonosov Moscow State University

neighbors, from which reconstruction occurs and weights are computed by solving optimization problem in second step. In final step, lower dimensional embedding is determined by minimizing error of reconstructing from weights. Laplacian Eigenmaps [13] also generally consists of three steps. In first step, adjacency matrix is constructed for determining whether data points are connected. In second step, weights among connected data points are calculated (other weights are set zero). In third step, Laplacian matrix is calculated and solving related eigenvector problem is needed for determining low dimensional manifold. Spectral clustering ([14, 15]) incorporating k-means usually uses Laplacian Eigenmaps as dimension reduction method. How Laplacian matrix needs to be derived from data set remains a question because structure of data set can be various. Diffusion maps was firstly proposed in [17]. And diffusion semigroups are used to generate multiscale geometries in order to organize and represent complex structures. Later, in ([18, 19]), the proposed diffusion map is integrated into one framework with the eigenmaps and random walk. Stiefel manifold from Grassmannian is used to project data to lower-dimensional space in the work of proposing visClust [16], which is a newly proposed clustering algorithm. Effectiveness of this novel clustering algorithm tends to decrease with rising of dimension while good result can be achieved on low dimensional data. Stochastic Neighbor Embedding (SNE) was initially proposed in [27]. SNE uses conditional probabilities to represent similarities. Final low dimensional embedding is a result of updating initialized low dimensional embedding by using gradient of sum of Kullback-Leibler divergences. For solving certain problems including difficulty of optimization in original SNE, t-SNE [26] is proposed with simpler gradient and computing the similarity by student-t distribution rather than a Gaussian distribution. Autoencoder ([20, 28]) is a multilayer neural network with a small central layer for reconstructing high-dimensional input vectors. Thus, by training the neural network, it's assumed that high-dimensional data can be effectively converted to low-dimensional representation. The idea of maximizing the trace of the inner product matrix has the effect of maximizing the variance of the low dimensional representation [23] was originally proposed as "semidefinite embedding" in ([25, 22]). Later "semidefinite embedding" is described as "maximum variance unfolding" [24].

There are works for comparing different manifold learning techniques. In the work [3], a comprehensive experiment for comparing twelve manifold learning techniques is conducted. In experiment manifold learning techniques are tested on five artificial datasets and five natural datasets. Main conclusion is that nonlinear techniques for dimensionality reduction are often not capable of outperforming traditional linear techniques such as PCA. In another work [5], PCA is examined on clustering gene expression data.

Experiment shows that clustering with the PCs does not necessarily improve, and often degrades, clustering quality.

So far, few works have focused on investigating effectiveness of manifold learning techniques on financial market data, especially stock price data. In [1], experiment is carried out on examining PCA, kernel PCA (KPCA) [31] and FRPCA [4]. Main reached conclusion is that three PCAs do not give significantly different results in average and standard PCA performs slightly better than FRPCA, and FRPCA performs slightly better than KPCA in average based on the P-values. One advantage of the work is the idea for construction of feature space. Limitation of the work is that only methods relating with PCA are examined. And in experiment there is no comparison for result obtained without dimension reduction. In work of [2], principal component analysis (PCA), autoencoder, deep belief network [28] are examined on stock selection. Authors argue that except for fitting nonlinear relations, autoencoder, deep belief network show no superiority to principal component analysis on Sharpe ratio and the advantage of dimensionality reduction is mainly reflected in trend situations (trend of moving up or down). Limitation of the work is related with the fact that risk free rate is not considered and only limited methods are examined.

It's very difficult to make any sufficient conclusion based on conducted researches on financial market data. Effect of manifold learning techniques on reducing dimensionality of financial market data is very unclear.

In this work, five popular manifold learning techniques, PCA, ISOMAP, Locally Linear Embedding, Laplacian Eigenmaps and t-SNE are compared for stock price trend clustering and classification task. The idea in [1] for construction of feature space is also used in this work. Manifold learning techniques are tested on datasets consisting of stock prices of five companies and high dimensional feature spaces including important macroeconomic indicators (for example, non-farm payroll, CPI, M1, M2, etc.) and other important indicators. Original datasets (without dimension reduction) and projected datasets are set as input for fully connected neural network and k-means to reveal effect of five popular manifold learning techniques.

The paper is organized into four sections. Section 1 is introduction part of total work. Section 2 reviews mathematical formulation of five manifold learning techniques. Section 3 presents experiment for comparison and discussion of experiment result. Section 4 is conclusion.

2. Manifold learning techniques

2.1. PCA

Goal of Principal Component Analysis [9] is to extract the important information from the original high dimensional data set and to express this information based on main directions of datapoints. It is likely to be the oldest manifold learning technique.

High dimensional data set with N datapoints is denoted as $\boldsymbol{X}$, with $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_N \end{bmatrix} \in R^{D\times N}$.Vector $\boldsymbol{\mu_X}$ is the central point of all data points:

$$\boldsymbol{\mu_X}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i,\ \boldsymbol{\mu_X} \in R^{D\times 1} \tag{1}$$

Matrix $\boldsymbol{M}$ from $\boldsymbol{\mu_X}$, with $\boldsymbol{M} = \begin{bmatrix} \boldsymbol{\mu_X} & \cdots & \boldsymbol{\mu_X} \end{bmatrix} \in R^{D\times N}$, is built. Original high dimensional data $\boldsymbol{X}$ needs to be centered firstly:

$$\boldsymbol{X}_c=\boldsymbol{X}-\boldsymbol{M} \tag{2}$$

Based on centered data $\boldsymbol{X}_c$, main directions need to be found through Singular Value Decomposition (SVD):

$$\boldsymbol{X}_c = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{3}$$

Low dimensional representation $\boldsymbol{D}$ is obtained through:

$$\boldsymbol{D} = \boldsymbol{U}^T\boldsymbol{X}_c \tag{4}$$

2.2. ISOMAP

Original high dimensional data set $\boldsymbol{X}$, with $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_N \end{bmatrix} \in R^{D\times N}$. Dimension of low dimensional manifold is denoted as d. Initially neighborhood relations of data points are represented as a weighted graph $\boldsymbol{G}$, with distances $d_X(i,j)$ between pairs of points i, j in the input space $\boldsymbol{X}$.

$$\boldsymbol{G} = (d_G\,(i,j)) = (\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) \tag{5}$$

By sorting every column of $\boldsymbol{G}$, if distance between pairs of points i, j is smaller than certain threshold, then two points are considered as neighbors. On the contrary, two points are not considered as neighbors. Next step for updating weight is conducted based on:

$$d_G(i,j)=\begin{cases} d_X(i,j), if\ \|\boldsymbol{x}_i - \boldsymbol{x}_j\| < \epsilon \\ \infty, \qquad if\ \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \geq \epsilon \end{cases} \tag{6}$$

For $k \in \{1, 2, \ldots, N\}$, geodesic distances are calculated as followings:

$$d^c{}_G(i,j) = \min\{d_G(i,j),\ d_G(i,k) + d_G(k,j)\} \tag{7}$$

For finding the shortest path between every two nodes, Floyd's algorithm can be used ([29, 30]). Apply MDS to $\boldsymbol{D}_G = \{d_G(i,j)\}$, $Y = \{\boldsymbol{y}_i\}$,$\boldsymbol{y}_i \in R^d$ and minimize the cost function by setting the coordinates $\boldsymbol{y}_i$ to the top d eigenvectors of the matrix $\tau(\boldsymbol{D}_G)$.

$$E = \|\tau(\boldsymbol{D}_G) - \tau(\boldsymbol{D}_Y)\|_{L^2} \tag{8}$$

where: $\tau(\boldsymbol{D}_G) = -\frac{\boldsymbol{HSH}}{2}$
$\boldsymbol{S} = (s_{ij}) = \left(d^c{}_G(i,j)^2\right)$
$\boldsymbol{H} = (h_{ij}),\ h_{ij} = \delta_{ij} - \frac{1}{N}$
p-th eigenvalue: λ_p.
p-th eigenvector: $\boldsymbol{v}_p$.
p-th component of coordinate of $\boldsymbol{y}_p$, with $\boldsymbol{y}_p = \boldsymbol{v}_p\sqrt{\lambda_p}$. Low dimensional manifold is then: $\boldsymbol{Y} = \begin{bmatrix}\boldsymbol{y}_1 & \cdots & \boldsymbol{y}_d\end{bmatrix}^T$.

2.3. Locally Linear Embedding

High dimensional input data, N samples with dimensionality D is denoted as $\boldsymbol{X}$, with $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}_1 \\ \cdots \\ \boldsymbol{x}_N\end{bmatrix} \in R^{N\times D}$.

For first step in Locally Linear Embedding, the set of neighbors for each data point can be determined by: choosing the K nearest neighbors by Euclidean distance, or choosing points within a fixed radius, or using prior knowledge [12].

For second step in Locally Linear Embedding, thus reconstruction with weights, matrix of weights $\boldsymbol{W}$ is calculated by minimizing reconstruction error:

$$\varepsilon(\boldsymbol{W}) = \sum_i \left|\boldsymbol{x}_i - \sum_j W_{ij}\boldsymbol{x}_j\right|^2 \tag{9}$$

Each data point is reconstructed from its neighbors and sum of each row of the weight matrix is one:

$$\sum_j W_{ij} = 1 \tag{10}$$

In the final step of Locally Linear Embedding, low dimensional manifold $\boldsymbol{Y}$ is determined by minimizing the embedding cost function:

$$\Phi(\boldsymbol{Y}) = \sum_i \left|\boldsymbol{y}_i - \sum_j W_{ij}\boldsymbol{y}_j\right|^2 \tag{11}$$

2.4. Laplacian Eigenmaps

Data in high dimensional space is denoted as $\boldsymbol{X}$, with $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_k \end{bmatrix} \in R^{l \times k}$.After constructing adjacency graph, a weight matrix is needed and denoted as $\boldsymbol{W}_{k \times k}$, with $\boldsymbol{W}_{k \times k} = (W_{ij})$.Weight between two points can be calculated either by a heat kernel or simple rule. By heat kernel, weight is calculated as:

$$W_{ij} = \begin{cases} e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}}, if \ \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 < \epsilon \\ 0 \quad , if \ \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \geq \epsilon \end{cases} \tag{12}$$

By simple rule, weight is calculated as:

$$W_{ij} = \begin{cases} 1 \, , if \ \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 < \epsilon \\ 0 \, , if \ \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \geq \epsilon \end{cases} \tag{13}$$

Diagonal weight matrix is denoted as $\boldsymbol{D}_{k \times k}$ and:

$$D_{ii} = \sum_j W_{ji} \tag{14}$$

Laplacian matrix is obtained by following way:

$$\boldsymbol{L}_{k \times k} = \boldsymbol{D}_{k \times k} - \boldsymbol{W}_{k \times k} \tag{15}$$

Eigenvectors $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{k-1}$ (corresponding eigenvalues $\lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{k-1}$) are solutions of equation 15, ordered according to their eigenvalues:

$$\boldsymbol{L}\boldsymbol{v} = \lambda \boldsymbol{D}\boldsymbol{v} \tag{16}$$

Where $\boldsymbol{v} \in R^{k \times 1}$. By removing $\boldsymbol{v}_0$ corresponding to $\lambda_0 = 0$, next m eigenvectors are used for embedding. Then dimensionality of k datapoints is reduced to m and $F_{k \times m} = (\mathbf{f}_\lambda)$,$\lambda \in \{1, 2, \ldots, m\}$.

2.5. t-SNE

As mentioned above, t-SNE is proposed based on SNE. For solving certain problems, such as difficulty of optimization in original SNE, t-SNE is proposed with computing the similarity by student-t distribution bot not Gaussian distribution. And a simpler gradient is used for minimizing sum of Kullback-leibler divergences.

Data in high dimensional space is denoted as $\boldsymbol{X}$, with $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_n \end{bmatrix} \in R^{h \times n}$. Low dimensional representation at step t is denoted

as $\boldsymbol{Y}^{(t)}$, with $\boldsymbol{Y}^{(t)} = \begin{bmatrix} \boldsymbol{y}_1^{(t)} & \cdots & \boldsymbol{y}_n^{(t)} \end{bmatrix} \in R^{d \times n}$. The similarity of datapoint $\boldsymbol{x}_j$ to datapoint $\boldsymbol{x}_i$ in high dimensional space is defined as:

$$p_{j|i} = \frac{\exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma_i^2})}{\sum_{k \neq i} \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_k\|^2}{2\sigma_i^2})} \tag{17}$$

Where σ_i is the variance of the distribution that is centered on datapoint $\boldsymbol{x}_i$. Further:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{18}$$

The perplexity is defined as:

$$Perp\left(P_i\right) = 2^{H(P_i)} \tag{19}$$

Shannon entropy is:

$$H\left(P_i\right) = -\sum_j p_{j|i} \log_2 p_{j|i} \tag{20}$$

The similarity of $\boldsymbol{y}_i$ to $\boldsymbol{y}_j$ in manifold:

$$q_{ij} = \frac{(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\boldsymbol{y}_k - \boldsymbol{y}_l\|^2)} \tag{21}$$

The sum of Kullback-leibler divergences over all datapoints is:

$$C = \sum_i KL(P|\,|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{22}$$

Gradient for minimizing C is then:

$$\frac{\partial C}{\partial \boldsymbol{y}_i} = 4 \sum_j (p_{ij} - q_{ij})(\boldsymbol{y}_i - \boldsymbol{y}_j)(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1} \tag{23}$$

For first step in t-SNE, compute p_{ij} of datapoint $\boldsymbol{x}_j$ to datapoint $\boldsymbol{x}_i$, and initiate $\boldsymbol{Y}^{(0)}$ randomly from $\mathcal{N}(0, 10^{-4}\boldsymbol{I})$. Then for $t \in \{1, 2, \ldots, T\}$, at every iteration, firstly, compute q_{ij}, then result is used for computing C and $\frac{\partial C}{\partial \boldsymbol{Y}^{(t)}}$, following update is then made on $\boldsymbol{Y}^{(t)}$:

$$\boldsymbol{Y}^{(t)} = \boldsymbol{Y}^{(t-1)} + \eta \frac{\partial C}{\partial \boldsymbol{Y}^{(t)}} + \alpha(t)(\boldsymbol{Y}^{(t-1)} - \boldsymbol{Y}^{(t-2)}) \tag{24}$$

Where:
η: learning rate,
T: number of iterations,
$\alpha(t)$: momentum.

3. Experiment

3.1. Methodology

3.1.1. Data sets

Stock prices of five companies, Amazon (AMZN), CME Group (CME), Microsoft (MSFT), Netflix (NFLX), Texas Instruments Incorporated (TXN) are chosen to build data sets. Totally, 5 datasets are used for conducting experiment. Raw data from 2021.05.26 to 2023.12.29, is obtained from following sources:

Raw data	**Source**
Stock price: Stock high prices Stock low prices	https://finance.yahoo.com/
Stock index: Dow Jones Industrial Average NASDAQ Composite	https://finance.yahoo.com/
Monetary and interest rate data: Federal Funds Effective Rate Monetary Base Bank Prime Loan Rate M1 M2 90-Day AA Financial Commercial Paper Interest Rate 30-Day, 60-Day Nonfinancial Commercial Paper Interest Rate 4-Week, 3-Month, 6-Month, 1-Year Treasury Bill Secondary Market Rate, Discount Basis Market Yield on U.S. Treasury Securities at 1-Month, 3-Month, 6-Month, 1-Year, 2-Year, 3-Year, 5-Year, 7-Year, 10- Year, 30-Year Constant Maturity	https://fred.stlouisfed.org/series

Market Yield on U.S. Treasury Securities at 5-Year, 7-Year, 10-Year, 20-Year, 30-Year Constant Maturity, Quoted on an Investment Basis, Inflation-Indexed	
Inflation, labor data: Consumer Price Index Unemployment Rate All Employees, Total Nonfarm	https://fred.stlouisfed.org/series
Exchange rate: EUR/USD GBP/USD USD/JPY USD/CNY	https://finance.yahoo.com/
Commodities: Gold Dec 24 (future) Crude Oil Dec 24 (future)	https://finance.yahoo.com/
Net income and Earnings per share	https://www.sec.gov/submit-filings

Table 1: Raw data and sources.

Five data sets are created based on collected raw data. Description of data sets is provided as the following:

Data set	number of samples	dimension
Stock AMZN	948	141
Stock CME	948	141
Stock MSFT	948	141
Stock NFLX	948	140
Stock TXN	948	141

Table 2. Description of data sets.

TimeSeriesSplit from sklearn is used for splitting original dataset to training and test data. For experiment, train test data size split is 80-20.

Data set	number of samples for training	number of samples for test
Stock AMZN	758	190
Stock CME	758	190
Stock MSFT	758	190
Stock NFLX	758	190

Stock TXN	758	190

Table 3: Training and test data.

Price trend label at day t is determined by:

$$s_t = \begin{cases} 1, \ if \ p_t^h > p_{t-1}^h, \ p_t^l > p_{t-1}^l \\ 0, \ if \ \left(p_t^h - p_{t-1}^h \right) \left(p_t^l - p_{t-1}^l \right) \leq 0 \\ -1, if \ p_t^h < p_{t-1}^h, \ p_t^l < p_{t-1}^l \end{cases} \tag{25}$$

Where:
p_t^h: high price at day t,
p_t^l: low price at day t.

3.1.2. Computation

Experiment is conducted on google Colab under configuration T4 GPU with system RAM 51.0 GB and GPU RAM 15.0 GB.

3.1.3. Sources of codes

Following Python implementations are used for conducting experiment. PCA: Python implementation from scikit-learn is used. Assessment on number of components for PCA : Python implementation (cross_val_score, FactorAnalysis) from scikit-learn is used. ISOMAP: Python implementation from scikit-learn is used. Locally Linear Embedding: Python implementation from scikit-learn is used. Laplacian Eigenmaps: Python implementation from scikit-learn is used. T-SNE: Python implementation from developer's site[2] is used with one modification for avoiding appearance of error. In function Hbeta, type is set as numpy.float128. When sumP is equal to zero, value is replaced by 10^{-10}. When sumP is positive infinity, value is replaced by 10^{10}. When sumP is negative infinity, value is replaced by -10^{10}. K-means: Python implementation from scikit-learn is used. Fully connected neural network: Python implementation from Keras is used.

3.1.4. Process of conducting experiment

For every original dataset, k-means and fully connected neural network are used to conduct clustering and classification on daily trend of stock price. Then by applying PCA, ISOMAP, Locally Linear Embedding, Laplacian Eigenmaps and t-SNE, dimension of original dataset is reduced. And newly

[2]https://lvdmaaten.github.io/tsne/

projected datasets are then used as input for k-means and fully connected neural network. Number of clusters for k-means is set as 3, because of price trend definition above. Architecture of fully connected neural network is configured as following:

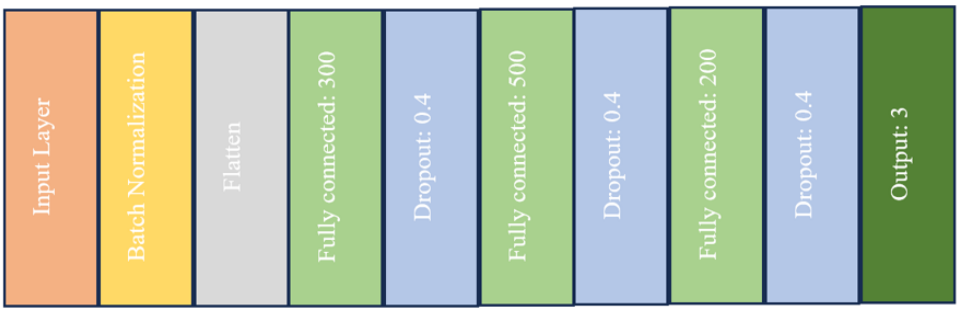

Fig. 1. Architecture of fully connected neural network

Neural network is trained by using method of gradient descent. Number of epochs for training neural network is 2000. Learning rate is 1e-06.

In experiment, dimension of manifold (n_c) is usually configured as 2, 50, 100, 140 except for PCA. Before conducting experiment for PCA, number of components is calculated based on two methods in 3.1.3. Calculated number of components is marked bold and represents recommended configuration of number of components on PCA. Choosing 2, 50, 100, 140 as dimension of manifold is because, firstly, for most examined manifold techniques, 2 is default configuration. This is because most manifold techniques are designed to achieve the goal of projecting high dimensional data on a 2D plane. Secondly, other dimension settings are used for revealing effect of manifold techniques more comprehensively. Every result on ACC (mean ± standard deviation) in experiment is obtained through 20 independent runs.

3.1.5. Metrics

ACC is used both for clustering and classification task. For clustering task, ACC is used instead of using ARI [6], which is a standard measurement for performance of clustering task. But for more straightforward comparison between clustering and classification task on prediction, ACC is used as a common measurement for performance. For clustering, best match between permutation of predicted labels and target labels is used for calculating ACC.

3.2. Experiment result

3.2.1. Result of k-means

Datasets	No dimension reduction	PCA			
Stock AMZN	0.47±0.008	**n_c = 3**	**n_c = 28**	n_c = 100	n_c = 140
		0.46±0.024	0.46±0.018	0.45±0.019	0.46±0.021
Stock CME	0.43±0.020	**n_c = 3**	**n_c = 29**	n_c = 100	n_c = 140
		0.42±0.017	0.43±0.017	0.43±0.019	0.43±0.018
Stock MSFT	0.45±0.013	**n_c = 3**	n_c = 28	n_c = 100	**n_c = 140**
		0.44±0.004	0.44±0.009	0.45±0.013	0.45±0.012
Stock NFLX	0.44±0.014	**n_c = 3**	**n_c = 31**	n_c = 100	n_c = 140
		0.44±0.012	0.44±0.016	0.44±0.012	0.45±0.007
Stock TXN	0.43±0.029	**n_c = 3**	**n_c = 29**	n_c = 100	n_c = 140
		0.43±0.022	0.46±0.018	0.45±0.024	0.44±0.033

Table 4. Clustering result on no dimension reduction and PCA.

Datasets	ISOMAP			
	n_c = 2	n_c = 50	n_c = 100	n_c = 140
Stock AMZN	0.44±0.008	0.43±0.029	0.43±0.034	0.42±0.032
Stock CME	0.38±0.005	0.39±0.014	0.39±0.017	0.38±0.016
Stock MSFT	0.43±0.010	0.45±0.016	0.44±0.019	0.44±0.018
Stock NFLX	0.45±1.110	0.42±0.033	0.42±0.027	0.42±0.027
Stock TXN	0.41±0.005	0.42±0.029	0.43±0.025	0.42±0.020

Table 5. Clustering result on ISOMAP.

Datasets	Locally Linear Embedding			
	n_c = 2	n_c = 50	n_c = 100	n_c = 140
Stock AMZN	0.44±0.004	0.43±0.011	0.43±0.006	0.43±0.005
Stock CME	0.39±1.110	0.40±0.009	0.40±0.011	0.40±0.006
Stock MSFT	0.41±0.001	0.43±0.018	0.43±0.011	0.43±0.006
Stock NFLX	0.41±0.004	0.41±0.011	0.41±0.007	0.41±0.007
Stock TXN	0.41±0.002	0.41±0.010	0.41±0.006	0.41±0.004

Table 6. Clustering result on Locally Linear Embedding.

Datasets	Laplacian Eigenmaps			
	n_c = 2	n_c = 50	n_c = 100	n_c = 140
Stock AMZN	0.45±0.002	0.43±0.014	0.43±0.013	0.43±0.011
Stock CME	0.41±5.551	0.40±0.015	0.40±0.014	0.39±0.014
Stock MSFT	0.44±0.002	0.42±0.021	0.41±0.019	0.42±0.023
Stock NFLX	0.42±0.001	0.40±0.014	0.41±0.019	0.41±0.014
Stock TXN	0.45±0.002	0.41±0.012	0.41±0.012	0.41±0.012

Table 7. Clustering result on Laplacian Eigenmaps.

Datasets	t-SNE			
	n_c = 2	n_c = 50	n_c = 100	n_c = 140
Stock AMZN	0.39±0.010	0.42±0.026	0.42±0.023	0.41±0.022
Stock CME	0.40±0.004	0.39±0.019	0.39±0.020	0.39±0.019
Stock MSFT	0.36±0.013	0.40±0.023	0.41±0.018	0.41±0.016
Stock NFLX	0.36±0.007	0.41±0.024	0.41±0.022	0.40±0.026
Stock TXN	0.39±0.014	0.42±0.024	0.43±0.029	0.43±0.020

Table 8. Clustering result on t-SNE.

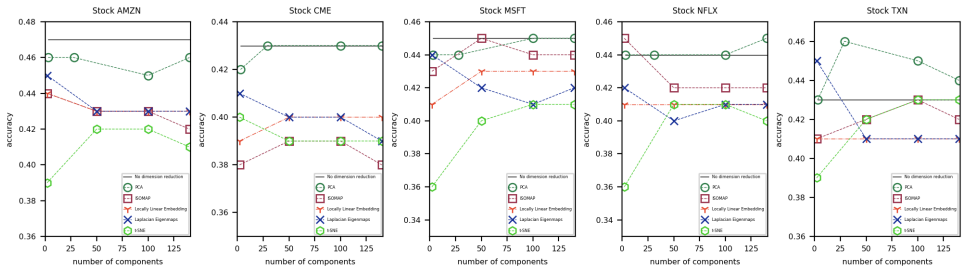

Fig. 2. Visualization of clustering results.

3.2.2. Results of neural network

Datasets	No dimension reduction	
	training	test
Stock AMZN	0.702±0.007	0.705±0.014
Stock CME	0.677±0.005	0.664±0.012
Stock MSFT	0.688±0.006	0.661±0.009
Stock NFLX	0.674±0.006	0.658±0.010
Stock TXN	0.684±0.007	0.689±0.011

Table 9: Classification result on no dimension reduction.

Datasets	PCA							
	training	test	training	test	training	test	training	test
Stock AMZN	**n_c = 3**		**n_c = 28**		n_c − 100		n_c − 140	
	0.670±0.003	0.667±0.001	0.682±0.009	0.679±0.008	0.694±0.007	0.698±0.012	0.697±0.008	0.707±0.011
Stock CME	**n_c = 3**		**n_c = 29**		n_c − 100		n_c − 140	
	0.668±0.002	0.663±0.000	0.670±0.006	0.668±0.007	0.674±0.006	0.664±0.009	0.675±0.007	0.659±0.008
Stock MSFT	**n_c = 3**		n_c − 28		n_c − 100		**n_c = 140**	
	0.666±0.002	0.664±0.003	0.676±0.008	0.643±0.009	0.686±0.006	0.657±0.009	0.688±0.008	0.657±0.008
Stock NFLX	**n_c = 3**		**n_c = 31**		n_c − 100		n_c − 140	
	0.666±0.002	0.664±0.001	0.667±0.006	0.654±0.006	0.671±0.007	0.659±0.008	0.675±0.007	0.656±0.009
Stock TXN	**n_c = 3**		**n_c = 29**		n_c − 100		n_c − 140	
	0.665±0.003	0.666±0.000	0.677±0.006	0.674±0.008	0.684±0.007	0.681±0.010	0.684±0.005	0.684±0.016

Table 10. Classification result on PCA.

Datasets	ISOMAP							
	n_c − 2		n_c − 50		n_c − 100		n_c − 140	
	training	test	training	test	training	test	training	test
Stock AMZN	0.670±0.003	0.667±0.003	0.677±0.006	0.666±0.012	0.684±0.007	0.666±0.011	0.684±0.007	0.668±0.007
Stock CME	0.666±0.002	0.666±0.002	0.663±0.008	0.660±0.007	0.666±0.005	0.657±0.008	0.664±0.007	0.656±0.008
Stock MSFT	0.669±0.003	0.662±0.002	0.671±0.006	0.637±0.012	0.672±0.007	0.640±0.010	0.672±0.006	0.638±0.011
Stock NFLX	0.667±0.003	0.664±0.003	0.664±0.008	0.648±0.008	0.666±0.008	0.648±0.009	0.668±0.010	0.645±0.007
Stock TXN	0.666±0.002	0.667±0.002	0.675±0.006	0.671±0.009	0.677±0.007	0.671±0.01	0.678±0.006	0.672±0.013

Table 11. Classification result on ISOMAP.

Datasets	Locally Linear Embedding							
	n_c − 2		n_c − 50		n_c − 100		n_c − 140	
	training	test	training	test	training	test	training	test
Stock AMZN	0.666±0.001	0.667±0.001	0.669±0.007	0.664±0.010	0.682±0.008	0.654±0.017	0.681±0.008	0.649±0.019
Stock CME	0.667±0.001	0.667±0.000	0.667±0.006	0.666±0.005	0.670±0.007	0.662±0.008	0.672±0.007	0.661±0.012
Stock MSFT	0.668±0.001	0.668±0.005	0.661±0.005	0.636±0.012	0.666±0.007	0.650±0.010	0.672±0.006	0.649±0.016
Stock NFLX	0.666±0.002	0.667±0.000	0.659±0.006	0.660±0.008	0.667±0.007	0.656±0.010	0.670±0.007	0.655±0.012
Stock TXN	0.665±0.001	0.667±0.001	0.666±0.006	0.666±0.010	0.669±0.007	0.662±0.012	0.666±0.008	0.649±0.014

Table 12. Classification result on Locally Linear Embedding.

Datasets	Laplacian Eigenmaps							
	n_c − 2		n_c − 50		n_c − 100		n_c − 140	
	training	test	training	test	training	test	training	test
Stock AMZN	0.667±0.000	0.667±0.000	0.666±0.002	0.667±0.000	0.662±0.006	0.667±0.000	0.658±0.008	0.666±0.000
Stock CME	0.667±0.000	0.667±0.000	0.666±0.001	0.667±0.000	0.666±0.001	0.667±0.000	0.664±0.003	0.667±0.000
Stock MSFT	0.667±0.000	0.667±0.000	0.666±0.003	0.666±0.001	0.662±0.005	0.666±0.002	0.663±0.006	0.667±0.001
Stock NFLX	0.667±0.000	0.667±0.000	0.666±0.002	0.667±0.000	0.664±0.004	0.667±0.000	0.663±0.004	0.667±0.000
Stock TXN	0.667±0.000	0.667±0.000	0.666±0.002	0.667±0.000	0.662±0.005	0.667±0.000	0.662±0.005	0.667±0.000

Table 13. Classification result on Laplacian Eigenmaps.

Datasets	t-SNE							
	n_c − 2		n_c − 50		n_c − 100		n_c − 140	
	training	test	training	test	training	test	training	test
Stock AMZN	0.664±0.004	0.667±0.000	0.681±0.010	0.678±0.011	0.682±0.009	0.684±0.009	0.688±0.009	0.685±0.008
Stock CME	0.665±0.002	0.667±0.000	0.661±0.007	0.666±0.010	0.662±0.007	0.669±0.013	0.665±0.005	0.658±0.012
Stock MSFT	0.664±0.004	0.666±0.004	0.668±0.006	0.654±0.010	0.670±0.006	0.647±0.012	0.673±0.006	0.651±0.010
Stock NFLX	0.666±0.003	0.665±0.005	0.669±0.007	0.637±0.011	0.669±0.007	0.638±0.008	0.667±0.006	0.634±0.011
Stock TXN	0.664±0.004	0.667±0.002	0.668±0.006	0.676±0.01	0.673±0.007	0.676±0.010	0.670±0.007	0.672±0.009

Table 14. Classification result on t-SNE.

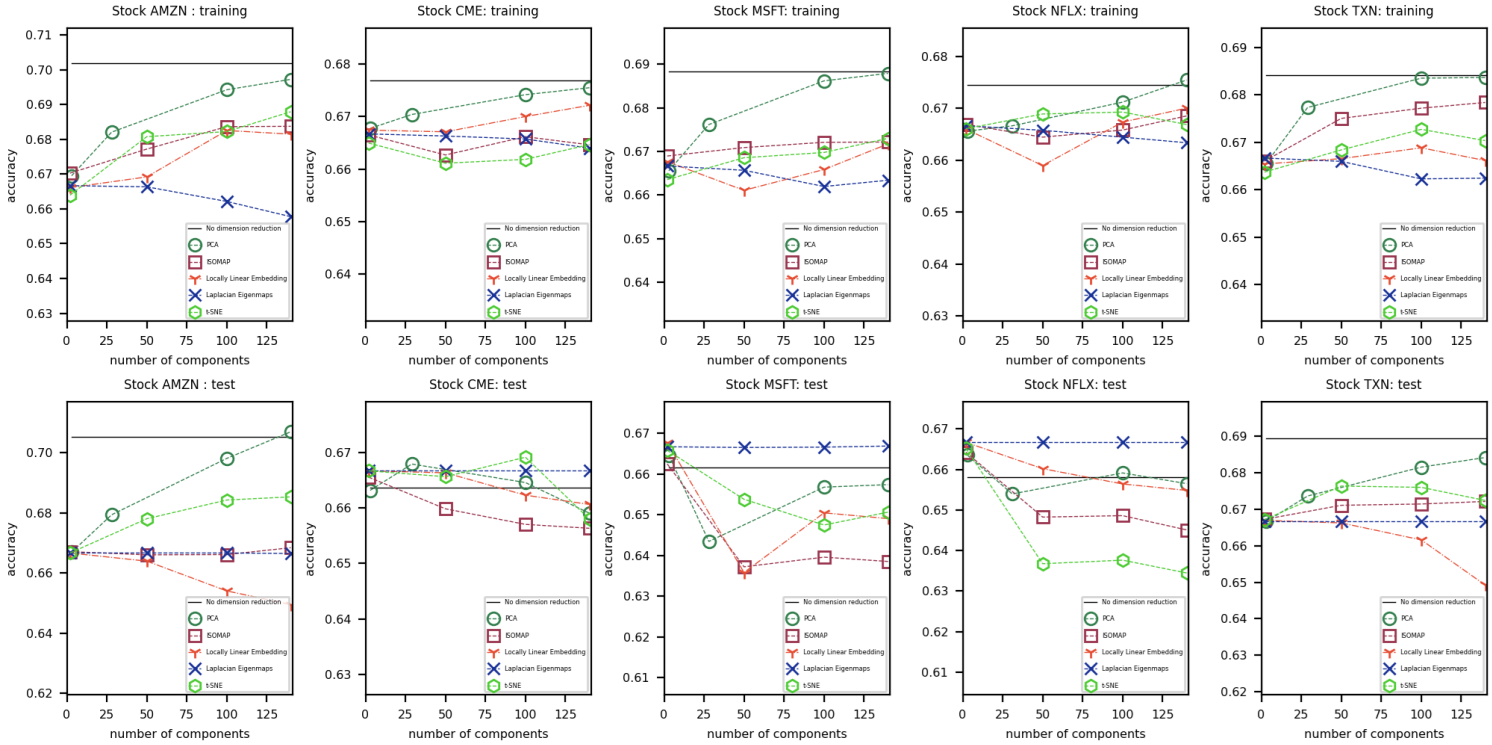

Fig. 3. Visualization of classification results

3.3. Discussion

3.3.1. K-means

Clustering results obtained from PCA generally can surpass other dimension reduction techniques. But it should be noticed that best result of PCA doesn't always correspondent to assessed number of components by methods mentioned in 3.1.3. Generally speaking, experiment results have shown that dimension reduction techniques often tend to worsen clustering performance of k-means. For all methods, there is at least one observation for not worsening clustering performance except Locally Linear Embedding. There is no observation for not worsening clustering performance for Locally Linear Embedding.

3.3.2. Neural network

For dataset AMZN, almost all methods tend to decrease neural network performance. Only PCA slightly improves neural network performance with number of components equal to 140. For dataset CME, most methods tend to slightly improve neural network performance. Improvement is generally less than 1 percent. For PCA, improvement of neural network performance is achieved at n_c= 29 and n_c = 100. For ISOMAP, improvement of performance is achieved at n_c = 2. Improvement of performance for Locally Linear Embedding is achived at n_c = 2 and n_c= 50. Improved performance for Laplacian Eigenmaps is achieved at all number of components. T-SNE improves performance of classification at n_c = 2, 50, 100 with best improvement among all method at n_c = 100. ISOMAP tend to more often decrease performance than other methods. For dataset MSFT, all methods tend to decrease classification accuracy except Laplacian Eigenmaps. Laplacian Eigenmaps can stably improve performance at all number of components. Other methods can all improve performance at lowest number of components but to worsen performance at other number of components. For dataset NFLX, all methods can all improve performance at lowest number of components. And PCA and Locally Linear Embedding can archive improvement correspondingly at n_c = 100 and n_c = 5. Again, Laplacian Eigenmaps can stably improve performance at all number of components. For dataset TXN, all methods worsen performance without exceptions. Generally speaking, improvement achieved by all methods is slight, usually less than 1 percent. For different dataset, improved performance by different methods is different. For best method, Laplacian Eigenmaps should be mentioned, as Laplacian Eigenmaps can stably improve classification accuracy for three datasets at all number of components.

3.3.3. Total result

The conclusion on PCA from clustering results of k-means supports the findings in [5]. Although result from PCA generally can be better than other dimension reduction techniques, but it's also observed that dimension reduction techniques often tend to worsen clustering performance of k-means. For classification task, observed improvement achieved by all methods is slight, usually less than 1 percent. Laplacian Eigenmaps can stably improve classification accuracy for three datasets at all number of components while other methods aren't able to achieve such performance. One assumption based on experiment can be raised: structures of data sets containing different stock prices can be very different. And there is no proof for that there exists one effective method for different stock data sets. Even the dimensions of data sets used in this experiment are almost same (except stock NFLX). And macroeconomic features of data sets are largely same. Thus, performance of examined techniques on stock data sets looks like the general performance of machine learning algorithms on different natural data sets.

3.3.4. Advantages and limitations

This work provides a more comprehensive study on manifold leaning techniques for financial data than previous works. Firstly, more methods are examined. Secondly, study has appropriately organized advantages of previous works, so conclusions are more convincing. In previous works, disadvantage on organizing experiment exists either in methodology of building dataset (for example, risk free rate is not considered) or type of examined methods (for example, only methods that are related with PCA are considered). Disadvantage also exists on the fact that only one of two tasks (clustering and classification) has been involved on drawing conclusion, but not both. These insufficiencies have negatively affected reached conclusions by these works. All mentioned insufficiencies have been properly addressed in this work. Future work can be improved further by involving more manifold leaning techniques that haven't been examined in this work.

4. Conclusion

In this work, effect of manifold learning techniques on predicting stock price trend is more carefully examined than previous works. Effect of manifold learning techniques on improving classification task result is slightly more obvious than that on clustering task. In clustering task, manifold learning techniques tend to often worsen clustering performance. In classification task, observed improvement achieved by all methods is slight, usually less than 1 percent. And only Laplacian Eigenmaps can more often stably

improve classification accuracy at all number of components while other methods aren't able to achieve such performance. There is no general effective technique for different stock data set. This result reminds about general performance of machine learning algorithms on different natural data sets, dimension of which can be very different.

Code and data availability

Code and data are available upon request.

Acknowledgments

This work is supported by the China Scholarship Council (grant no. 202308090103). I want to thank my scientific supervisor (Сидоренко Владимир Николаевич, к.ф.-м.н., к.э.н., к.ю.н.) for advices on improving quality of this work.

References

[1] Zhong, Xiao and Enke, David, "Forecasting daily stock market return using dimensionality reduction", *Expert Systems with Applications*, **67** (jan 2017), 126–139, http://dx.doi.org/10.1016/j.eswa.2016.09.027.

[2] Han, Jingti and Ge, Zhipeng, "Effect of dimensionality reduction on stock selection with cluster analysis in different market situations", *Expert Systems with Applications*, **147** (jun 2020), 113226, http://dx.doi.org/10.1016/j.eswa.2020.113226.

[3] Van Der Maaten, Laurens and Postma, Eric O and Van Den Herik, H Jaap and others, "Dimensionality reduction: A comparative review", *Journal of machine learning research*, **10**:66-71 (2009), 13.

[4] Luukka, Pasi, "A New Nonlinear Fuzzy Robust PCA Algorithm and Similarity Classifier in Classification of Medical Data Sets.", *International Journal of Fuzzy Systems*, **13**:3 (2011).

[5] Yeung, K. Y. and Ruzzo, W. L., "Principal component analysis for clustering gene expression data", *Bioinformatics*, **17**:9 (sep 2001), 763–774, http://dx.doi.org/10.1093/bioinformatics/17.9.763.

[6] Hubert, Lawrence and Arabie, Phipps, "Comparing partitions", *Journal of Classification*, **2**:1 (dec 1985), 193–218, http://dx.doi.org/10.1007/bf01908075.

[7] Hotelling, H., "Analysis of a complex of statistical variables into principal components.", *Journal of Educational Psychology*, **24**:6 (sep 1933), 417–441, http://dx.doi.org/10.1037/h0071325.

[8] Pearson, Karl, "LIII. On lines and planes of closest fit to systems of points in space", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **2**:11 (nov 1901), 559–572, http://dx.doi.org/10.1080/14786440109462720.

[9] Abdi, Hervé and Williams, Lynne J., "Principal component analysis", *WIREs Computational Statistics*, **2**:4 (jul 2010), 433–459, http://dx.doi.org/10.1002/wics.101.

[10] Tenenbaum, Joshua B. and Silva, Vin de and Langford, John C., "A Global Geometric Framework for Nonlinear Dimensionality Reduction", *Science*, **290**:5500 (dec 2000), 2319–2323, http://dx.doi.org/10.1126/science.290.5500.2319.

[11] Cox, M. A. A. and Cox, T. F., "Multidimensional Scaling on the Sphere", *Compstat*, Physica-Verlag HD, 1988, 323–328, ISBN: 9783642469008, http://dx.doi.org/10.1007/978-3-642-46900-8_44.

[12] Roweis, Sam T. and Saul, Lawrence K., "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science*, **290**:5500 (dec 2000), 2323–2326, http://dx.doi.org/10.1126/science.290.5500.2323.

[13] Belkin, Mikhail and Niyogi, Partha, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation", *Neural Computation*, **15**:6 (jun 2003), 1373–1396, http://dx.doi.org/10.1162/089976603321780317.

[14] von Luxburg, Ulrike, "A tutorial on spectral clustering", *Statistics and Computing*, **17**:4 (aug 2007), 395–416, http://dx.doi.org/10.1007/s11222-007-9033-z.

[15] Ding, Ling and Li, Chao and Jin, Di and Ding, Shifei, "Survey of spectral clustering based on graph theory", *Pattern Recognition*, **151** (jul 2024), 110366, http://dx.doi.org/10.1016/j.patcog.2024.110366.

[16] Breger, Anna and Karner, Clemens and Ehler, Martin, "visClust: A visual clustering algorithm based on orthogonal projections", *Pattern Recognition*, **148** (apr 2024), 110136, http://dx.doi.org/10.1016/j.patcog.2023.110136.

[17] Coifman, R. R. and Lafon, S. and Lee, A. B. and Maggioni, M. and Nadler, B. and Warner, F. and Zucker, S. W., "Geometric diffusions as

a tool for harmonic analysis and structure definition of data: Diffusion maps", *Proceedings of the National Academy of Sciences*, **102**:21 (may 2005), 7426–7431, http://dx.doi.org/10.1073/pnas.0500334102.

[18] Lafon, S. and Lee, A.B., "Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**:9 (sep 2006), 1393–1403, http://dx.doi.org/10.1109/tpami.2006.184.

[19] Nadler, Boaz and Lafon, Stéphane and Coifman, Ronald R. and Kevrekidis, Ioannis G., "Diffusion maps, spectral clustering and reaction coordinates of dynamical systems", *Applied and Computational Harmonic Analysis*, **21**:1 (jul 2006), 113–127, http://dx.doi.org/10.1016/j.acha.2005.07.004.

[20] DeMers, David and Cottrell, Garrison, "Non-linear dimensionality reduction", *Advances in neural information processing systems*, **5** (1992).

[21] Hinton, G. E. and Salakhutdinov, R. R., "Reducing the Dimensionality of Data with Neural Networks", *Science*, **313**:5786 (jul 2006), 504–507, http://dx.doi.org/10.1126/science.1127647.

[22] Weinberger, Kilian Q. and Saul, Lawrence K., "Unsupervised Learning of Image Manifolds by Semidefinite Programming", *International Journal of Computer Vision*, **70**:1 (may 2006), 77–90, http://dx.doi.org/10.1007/s11263-005-4939-z.

[23] Weinberger, Kilian Q. and Sha, Fei and Zhu, Qihui and Saul, Lawrence K., "Graph Laplacian Regularization for Large-Scale Semidefinite Programming", *Advances in Neural Information Processing Systems 19*, The MIT Press, sep 2007, 1489–1496, ISBN: 9780262256919, http://dx.doi.org/10.7551/mitpress/7503.003.0191.

[24] Sun, Jun and Boyd, Stephen and Xiao, Lin and Diaconis, Persi, "The Fastest Mixing Markov Process on a Graph and a Connection to a Maximum Variance Unfolding Problem", *SIAM Review*, **48**:4 (jan 2006), 681–699, http://dx.doi.org/10.1137/s0036144504443821.

[25] Weinberger, Kilian Q. and Sha, Fei and Saul, Lawrence K., "Learning a kernel matrix for nonlinear dimensionality reduction", *Twenty-first international conference on Machine learning - ICML '04*, ACM Press, 2004, 106, http://dx.doi.org/10.1145/1015330.1015345.

[26] Van der Maaten, Laurens and Hinton, Geoffrey, "Visualizing data using t-SNE.", *Journal of machine learning research*, **9**:11 (2008).

[27] Hinton, Geoffrey E and Roweis, Sam, "Stochastic neighbor embedding", *Advances in neural information processing systems*, **15** (2002).

[28] Hinton, Geoffrey E. and Osindero, Simon and Teh, Yee-Whye, "A Fast Learning Algorithm for Deep Belief Nets", *Neural Computation*, **18**:7 (jul 2006), 1527–1554, http://dx.doi.org/10.1162/neco.2006.18.7.1527.

[29] Floyd, Robert W., "Algorithm 97: Shortest path", *Communications of the ACM*, **5**:6 (jun 1962), 345–345, http://dx.doi.org/10.1145/367766.368168.

[30] Warshall, Stephen, "A Theorem on Boolean Matrices", *Journal of the ACM*, **9**:1 (jan 1962), 11–12, http://dx.doi.org/10.1145/321105.321107.

[31] Schölkopf, Bernhard and Smola, Alexander and Müller, Klaus-Robert, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem", *Neural Computation*, **10**:5 (jul 1998), 1299–1319, http://dx.doi.org/10.1162/089976698300017467.

Часть 3
Математические модели

Оценка числа правильных семейств функций k-значной логики

А. В. Галатенко[1]

Правильные семейства функций являются удобным средством для задания больших параметрических классов квазигрупп и d-квазигрупп. К. Д. Царегородцевым было установлено, что в булевом случае правильные семейства находятся в естественном взаимно-однозначном соответствии с одностоковыми ориентациями булева куба. Число таких ориентаций было оценено И. Матоушеком. В работе представлено обобщение нижней оценки Матоушека на случай логики произвольной значности, некоторые следствия из этого обобщения, а также показано, что свойство правильности является редким — доля правильных семейств размера n в классе всех семейств из n n-арных функций, для которых одноименная переменная фиктивна, стремится к 0 при $n \to \infty$.

Ключевые слова: правильные семейства функций, функции k-значной логики, гиперграф, паросочетание

1. Введение

Правильные семейства булевых функций были введены В. А. Носовым в работе [1]. В работе [2] было показано, что с помощью правильных семейств булевых функций размера n можно задавать параметрические классы квазигрупп порядка 2^n. В работе [3] В. А. Носов и А. Е. Панкратьев обобщили понятие правильного семейства на случай функций k-значной логики при $k \geq 3$. И. А. Плаксина показала, что с помощью правильных семейств можно задавать не только квазигруппы, но и d-квазигруппы при $d \geq 3$ [4]. В работе [5] конструкция Плаксиной была дополнительно усилена. Обзор результатов по правильным семействам приведен в статье [6].

Обозначим через $T_k(n)$ число правильных семейств размера n в k-значной логике. Известно, что $T_2(n) \geq n^{A \cdot 2^n}$, где A — некоторая положительная вещественная константа. Этот результат был получен следующим способом. К. Д. Царегородцев установил, что между правильными семействами булевых функций размера n и так называемыми одностоковыми ориентациями n-мерного булева куба существует взаимно однозначное соответствие [7]. Таким образом, в качестве нижней оценки на число

[1] *Галатенко Алексей Владимирович* — доцент каф. МаТИС мех.-мат. ф-та МГУ, e-mail: agalat@msu.ru.

Galatenko Alexei Vladimirovich — associate professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, MaTIS chair

правильных семейств булевых функций можно использовать оценку на число одностоковых ориентаций, полученную И. Матоушеком [8]. Приведенное ниже обобщение конструкции Матоушека на k-значный случай позволяет получить нижнюю оценку $T_k(n)$ для произвольного $k \geq 3$. Затем мы покажем, что свойство правильности является редким — доля правильных семейств среди всех семейств функций, фиктивно зависящих от одноименной переменной, стремится к 0.

Дальнейшее изложение имеет следующую структуру. В разделе 2 вводятся необходимые определения. В разделе 3 представлены основные результаты работы. Раздел 4 является заключением.

2. Основные определения и обозначения

Пусть $E_k = \{0, 1, \ldots, k-1\}$. Множество всех n-местных функций на E_k обозначим через P_k^n.

Определение 1. *Пусть $k, n \in \mathbb{N}$, $k \geq 2$. Семейство функций $(f_1, \ldots, f_n)$, $f_i \in P_k^n$, $i = 1, \ldots, n$, называется правильным, если для любых наборов $\alpha, \beta \in E_k^n$, $\alpha = (a_1, \ldots, a_n)$, $\beta = (b_1, \ldots, b_n)$, $\alpha \neq \beta$, найдется индекс i, $1 \leq i \leq n$, такой что $a_i \neq b_i$, но $f_i(\alpha) = f_i(\beta)$.*

Легко увидеть, что если семейство $(f_1, \ldots, f_n)$ является правильным, то переменная x_i фиктивна для функции $f_i(x_1, \ldots, x_n)$, $i = 1, \ldots, n$.

Пример 1 ([3, пример 2])**.** *Пусть семейство $F = (f_1, \ldots, f_n)$ таково, что переменные $x_i, \ldots, x_n$ фиктивны для функции $f_i(x_1, \ldots, x_n)$, $i = 1, \ldots, n$. Тогда это семейство является правильным. Для доказательства правильности достаточно рассмотреть первую позицию, на которой наборы α и β отличаются. Рассмотрим произвольную перестановку $\sigma \in S_n$ и применим ее к семейству F. Результат обозначим через $G = (g_1, \ldots, g_n)$, $g_i(x_1, \ldots, x_n) = f_{\sigma(i)}(x_{\sigma(1)}, \ldots, x_{\sigma(n)}$. Легко увидеть, что семейство G также является правильным. Такие семейства называются треугольными.*

Пример 2 ([6, замечание 1])**.** *Пусть семейство $F = (f_1, \ldots, f_n)$ удовлетворяет следующим условиям. Во-первых, переменная x_i фиктивна для функции $f_i(x_1, \ldots, x_n)$, $i = 1, \ldots, n$. Во-вторых, существует элемент $a \in E_k$, такой что для любого набора $\alpha \in E_k^n$ и любых i, j, $1 \leq i < j \leq n$, по крайней мере одно из значений $f_i(\alpha)$, $f_j(\alpha)$ равно a. Тогда семейство F правильное. Если во втором условии в качестве элемента a выбрать 0, получится аналог условия ортогональности, поэтому семейства такого вида мы будем называть обобщенно ортогональными.*

Значительное число других примеров можно найти в работе [6].

В завершение раздела приведем ряд определений из области теории графов, которые потребуются для доказательства утверждений.

Определение 2. *Гиперграфом называется пара (V, E), где V — конечное множество вершин, E — конечное множество гиперребер, элементами которого являются подмножества вершин. Если число вершин во всех гиперребрах равно одной и той же константе r, гиперграф называется r-однородным или просто однородным.*

Определение 3. *Пусть $H = (V, E)$ — некоторый гиперграф, $v \in V$ — вершина этого гиперграфа. Степенью вершины $d(v)$ называется число гиперрребер, содержащих v. Если для всех вершин степень равна одной и той же константе d, гиперграф называется d-регулярным или просто регулярным.*

Определение 4. *Паросочетанием в гиперграфе $H = (V, E)$ называется подмножество попарно непересекающихся гиперребер $M \subseteq E$.*

Пусть $H = (V, E)$ — гиперграф, $u, v \in V$, $u \neq v$. Через $d(u, v)$ обозначим число гиперребер, содержащих вершины u и v.

3. Основные результаты

Теорема 1. *Существует константа $A \in \mathbb{R}$, $A > 0$, такая что*

$$T_k(n) \geq n^{A \cdot k^n}.$$

Доказательство. Сперва построим взаимно однозначное соответствие между семействами функций $F_n = (f_1, \ldots, f_n)$, где $f_i \in P_k^n$, причем переменная x_i фиктивна для f_i, $i = 1, \ldots, n$, и классом нагруженных гиперграфов следующего вида. В качестве множества вершин выберем множество E_k^n. Содержательно каждая вершина соответствует набору значений переменных. Каждое подмножество вершин вида

$$\{(a_1, \ldots, a_{i-1}, c, a_{i+1}, \ldots, a_n) \mid c \in E_k\} \tag{1}$$

образуем гиперребро. Содержательно гиперребрами соединяются все наборы, отличающиеся ровно в одной заданной позиции. Дополнительно каждое гиперребро снабжается пометкой из множества E_k. В булевом случае роль пометок играла ориентация ребер.

Для определения отображения из множества семейств функций в множество гиперграфов достаточно задать правило, задающее значение меток гиперребер. Сделаем это следующим образом. Пусть F_n —

некоторое семейство. Тогда гиперребро (1) снабжается меткой, равной $f_i(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n)$. Заметим, что в силу фиктивной зависимости f_i от переменной x_i константа 0 может быть заменена на любую другую, при этом значение метки не изменится. Легко увидеть, что построенное соответствие является биекцией, так как разные семейства переходят в разные гиперграфы и любая комбинация меток реализуется как образ некоторого семейства.

Рассмотрим правильное семейство, такое что все функции тождественно равны константе $k-1$, и гиперграф H_n, соответствующий этому семейству. Метки всех гиперребер H_n равны $k-1$. Рассмотрим произвольное паросочетание M в H_n. Произвольным образом поменяем значения меток для гиперребер из M, выбрав для каждого гиперребра новое значение из множества $E_k \setminus \{k-1\}$. Обозначим получившийся гиперграф через H'_n, а прообраз этого гиперграфа относительно введенного соответствия через $F'_n = (f'_1, \ldots, f'_n)$. Покажем, что он является образом правильного семейства функций. Возможны следующие случаи.

1. $n = 1$. Тогда H'_n имеет ровно одно гиперребро с пометкой $c \in E_k$ и соответствует семейству, состоящему из одной функции, тождественно равной c, то есть является правильным.

2. $n = 2$. Тогда гиперграф H'_n имеет форму квадрата, а все гиперребра из множества M идут в одном направлении (любые два гиперребра, идущих разных направлениях, то есть одно горизонтальное и одно вертикальное, пересекаются). Рассмотрим подслучай, когда ребра в M вертикальные, что соответствует вариации второй компоненты набора. Тогда семейство F'_n имеет вид $((k-1), f(x_1))$, то есть является треугольным и, как следствие, правильным. Второй подслучай рассматривается полностью аналогично.

3. $n \geq 3$. В этом случае завершим доказательство индукцией по n. Рассмотрим пару входных наборов $\alpha = (a_1, \ldots, a_n)$, $\beta = (b_1, \ldots, b_n)$, $\alpha \neq \beta$. Если найдется такой индекс i, $1 \leq i \leq n$, что $a_i = b_i$, то рассмотрим семейство F^i_n, полученное из F'_n подстановкой значения a_i вместо переменной x_i и вычеркиванием функции f'_i. Семейству F^i_n соответствует гиперграф H^i_n, являющийся подграфом H'_n, порожденным множеством вершин, для которых компонента номер i равна a_i; этот подграф может быть получен из H_{n-1} и паросочетания, в которое входят те и только те гиперребра из M, для которых координата номер i фиксирована и равна a_i. По индуктивному предположению семейство F^i_n правильное, значит, найдется индекс j, такой что $a_j \neq b_j$, но $f'_j(\alpha) = f'_j(\beta)$. Пусть теперь все компоненты наборов α и β различны. Так как вершины, соответствующие наборам α и β, принадлежат не более чем одному гиперребру из M, а $n \geq 3$, найдется такой индекс i, что гиперребра, содержащие α и β и полученные вариацией компоненты номер i, не лежат в M, то есть в H'_n

оба этих ребра сохраняют пометку $k-1$. Значит, $f_i(\alpha) = f_i(\beta) = k-1$. Таким образом, семейство F'_n правильное по определению.

Легко увидеть, что семейства, являющиеся прообразами гиперграфов для различных паросочетаний, различны. Таким образом, число паросочетаний в графе H_n является нижней оценкой числа правильных семейств размера n.

Для завершения доказательства воспользуемся оценкой из работы А. С. Асратяна и Н. Н. Кузюрина [9]. Пусть r фиксированная константа, $H^n = (V^n, E^n)$ — , r-однородный и $d(n)$-регулярный гиперграф, такой что для каждой пары $u, v \in V$, $u \neq v$, выполнено $d(u,v) = o(n)$, $n \to \infty$. Тогда число паросочетаний $N(H^n)$ оценивается снизу следующим образом:

$$N(H^n) \geq d(n)^{(n-o(n))/r}, \quad n \to \infty. \tag{2}$$

Заметим, что в гиперграфе $H(n)$ степень каждой вершины равна n, каждое ребро состоит из k вершин, а пара различных вершин принадлежит не более чем одному гиперребру. Таким образом, можно применить оценку (2) с параметрами $r = k$, $d = n$, $n = n^k$, и получить следующее асимптотическое неравенство:

$$T_k(n) \geq n^{\frac{1}{k}\left(n^k - o\left(n^k\right)\right)}, \quad n \to \infty,$$

из которого вытекает справедливость теоремы. □

Замечание 1. *Так как нижняя оценка построена с помощью паросочетания, для любого входного набора α не более, чем одна функция нового семейства примет значение, отличное от $k-1$, откуда следует, что это семейство является обобщенно ортогональным.*

Используем полученную оценку для доказательства того, что семейства из примеров 1 составляют бесконечно малую долю в множестве всех правильных семейств.

Теорема 2. *Пусть $k \in \mathbb{N}$, $k \geq 3$. Тогда доля треугольных семейств размера n в множестве всех правильных семейств размера n стремится к 0 при $n \to \infty$.*

Доказательство. Сперва оценим сверху мощность множества треугольных семейств $(f_1, \ldots, f_n)$, таких что переменные $x_i, \ldots, x_n$ фиктивны для функции $f_i(x_1, \ldots, x_n)$, $i = 1, \ldots, n$. Из предположения на вид функции f_i следует, что она может быть выбрана $k^{k^{i-1}}$ способами. Значит, общее число треугольных семейств такого вида оценивается сверху следующим образом:

$$k^{k^0} \cdot k^{k^1} \cdot \ldots \cdot k^{k^{n-1}} = k^{\sum_{i=0}^{n-1} k^i} = k^{\frac{k^n-1}{k-1}} \leq k^{k^n}.$$

В результате согласованной перенумерации мощность вырастет не более, чем в $n!$ раз. Следовательно, доля треугольных семейств в множестве всех правильных семейств оценивается сверху величиной

$$\frac{n! \cdot k^{k^n}}{n^{A \cdot k^n}}. \tag{3}$$

Логарифм по основанию k отношения (3) в свою очередь оценивается величиной

$$\log_k (n!) + k^n - Ak^n \cdot \log_k n \leq n \cdot \log_k n + k^n - Ak^n \cdot \log_k n,$$

которая стремится к $-\infty$ при $n \to \infty$. Значит, искомое отношение стремится к 0. □

Следствие 1. *Отношение числа треугольных семейств размера n к числу обобщенно ортогональных семейств размера n стремится к 0 при $n \to \infty$.*

Замечание 2. *Аналог теоремы 2 для булева случая был установлен К. Д. Царегородцевым (см. [10, Теорема 6]). Следствие 1 в булевом случае также имеет место.*

Теперь покажем, что свойство правильности является редким: доля правильных семейств размера n среди всех семейств размера n, в которых функции фиктивно зависят от одноименной переменной, стремится к 0.

Теорема 3. *Пусть $k \in \mathbb{N}$, $k \geq 3$. Тогда*

$$\lim_{n \to \infty} \frac{T_k(n)}{N_k(n)} = 0, \tag{4}$$

где $N_k(n)$ — число семейств $(f_1, \ldots, f_n)$, $f_i \in P_k^n$, таких что переменная x_i фиктивна для функции $f_i(x_1, \ldots, x_n)$.

Доказательство. Функцию от n переменных, для которой заданная переменная фиктивна, очевидным образом можно выбрать $k^{k^{n-1}}$ способом, откуда следует равенство $N_k(n) = k^{n \cdot k^{n-1}}$.

Оценим сверху величину $T_k(n)$. Разделим функции семейства на пары: (f_1, f_2), (f_3, f_4), … Если n четно, последняя пара имеет вид (f_{n-1}, f_n), в противном случае последняя пара имеет вид (f_{n-2}, f_{n-1}), а последняя функция остается без пары. Оценим число вариантов для выбора пары (f_i, f_{i+1}). Сгруппируем входные наборы так, что в одну группу попадают все наборы, отличающихся только в позициях i или j. В каждой группе окажется k^{n-2} набора. На каждой группе пару (f_i, f_j) можно рассматривать как семейство функций от двух переменных x_i, x_j. Используя

утверждение о сохранении правильности при подстановке константы вместо переменной с удалением одноименной функции (см., например, [11, лемма 1]) легко показать, что это семейство правильное. Известно, что правильные семейства размера 2 являются треугольными (см., например, [11, утверждение 1]). Значит, на рассматриваемой группе входных наборов либо f_i является константой, а f_j есть функция от одной переменной x_i, либо наоборот. Значит, имеется не более $2 \cdot k \cdot k^k$ способов задать значения пары функций (f_i, f_j) на каждой группе входов, а общее число способов выбора (f_i, f_j) оценивается сверху величиной

$$\left(2 \cdot k \cdot k^k\right)^{k^{n-2}} \leq \left(k^{k+2}\right)^{k^{n-2}} = k^{(k+2)k^{n-2}}.$$

Если n четно, число пар равно $n/2$, поэтому число правильных семейств размера n оценивается сверху величиной

$$\left(k^{(k+2)k^{n-2}}\right)^{n/2};$$

если n нечетно, замечаем, что оставшаяся без пары функция f_n имеет фиктивную переменную x_n и поэтому может быть выбрана не более, чем $k^{k^{n-1}}$ способом, поэтому верхняя оценка принимает вид

$$\left(k^{(k+2)k^{n-2}}\right)^{(n-1)/2} \cdot k^{k^{n-1}}.$$

Поделим полученные оценки на $N_k(n)$ и прологарифмируем по основанию k. В первом случае с учетом неравенства $k \geq 3$ возникает следующая цепочка:

$$\frac{n}{2} \cdot (k+2) \cdot k^{n-2} - n \cdot k^{n-1} = \left(\frac{nk}{2} - kn + n\right) k^{n-2} = \\ = (2-k)\frac{n}{2} \cdot k^{n-2} \to -\infty, n \to \infty.$$

Во втором случае имеем

$$\frac{n-1}{2} \cdot (k+2) \cdot k^{n-2} + k^{n-1} - n \cdot k^{n-1} = \left(\frac{kn}{2} - \frac{k}{2} + n - 1 + k - nk\right) k^{n-2} = \\ = \left((2-k)\frac{n-1}{2}\right) k^{n-2} \to -\infty, n \to \infty.$$

Обе подпоследовательности сходятся к $-\infty$, откуда следует, что исходное отношение стремится к 0 при $n \to \infty$. □

Замечание 3. *Справедливость формулы (4) для случая $k = 2$ является очевидным следствием верхней оценки $T_2(n) \leq n^{B \cdot 2^n}$, полученной И. Матоушеком в работе [8]; здесь B — некоторая положительная вещественная константа.*

4. Заключение

В булевом случае в работе И. Матоушека [8] получены оценки мощности множества одностоковых ориентаций n-мерного булева куба. Из результатов К. Д. Царегородцева [7] вытекает, что эти оценки одновременно являются оценками мощности множества правильных семейств булевых функций размера n. В нашей работе изучается мощность множества правильных семейств функций k-значной логики при $k \geq 3$. В качестве нижней оценки предлагается прямое обобщение оценки Матоушека. Интересно, что она достигается на множестве обобщенно ортогональных правильных семейств, тогда как доля треугольных семейств размера n в классе всех правильных семейств размера n стремится к 0 при $n \to \infty$; более того, отношение мощности множества треугольных семейств размера n и мощности множества обобщенно ортогональных правильных семейств также стремится к 0.

Из определения правильности следует, что для каждой функции семейства одноименная переменная должна быть фиктивной. В качестве верхней оценки в работе показано, что доля правильных семейств размера n в классе всех семейств $(f_1, \ldots, f_n)$, где $f_i \in P_k^n$ и x_i фиктивна для $f_i(x_1, \ldots, x_n)$, $i = 1, \ldots, n$, стремится к 0 при $n \to \infty$. По всей видимости, верно и более сильное утверждение: $T_k(n) \leq n^{B \cdot k^n}$ для некоторой вещественной константы B. Доказательство этой гипотезы является направлением дальнейших исследований.

Автор благодарит К. Д. Царегородцева и А. Е. Панкратьева за плодотворные обсуждения и полезные замечания.

Список литературы

[1] В. А. Носов, “Критерий регулярности булевского неавтономного автомата с разделенным входом”, *Интеллектуальные системы*, **3**:3–4 (1998), 269–280.

[2] В. А. Носов, “Построение классов латинских квадратов в булевой базе данных”, *Интеллектуальные системы*, **4**:3–4 (1999), 307–320.

[3] В. А. Носов, А. Е. Панкратьев, “Латинские квадраты над абелевыми группами”, *Фундаментальная и прикладная математика*, **12**:3 (2006), 65–71.

[4] И. А. Плаксина, “Построение параметрического семейства многомерных латинских квадратов”, *Интеллектуальные системы. Теория и приложения*, **18**:2 (2014), 323–329.

[5] А. В. Галатенко, В. А. Носов, А. Е. Панкратьев, К. Д. Царегородцев, "О порождении n-квазигрупп с помощью правильных семейств функций", *Дискретная математика*, **35**:1 (2023), 35–53.

[6] A. V. Galatenko, V. A. Nosov, A. E. Pankratiev, K. D. Tsaregorodtsev, "Proper families of functions and their applications", *Матем. вопр. криптогр.*, **14**:2 (2023), 43–58.

[7] К. Д. Царегородцев, "О взаимно однозначном соответствии между правильными семействами булевых функций и реберными ориентациями булевых кубов", *Прикладная дискретная математика*, 2020, № 48, 16–21.

[8] J. Matoušek, "The number of unique-sink orientations of the hypercube", *Combinatorica*, **26** (2006), 91–99.

[9] A. S. Asratian, N. N. Kuzjurin, "On the number of nearly perfect matchings in almost regular uniform hypergraphs", *Discrete Mathematics*, **207**:1–3 (1999), 1–8.

[10] К. Д. Царегородцев, "О свойствах правильных семейств булевых функций", *Дискретная математиа*, **33**:1 (2021), 91–102.

[11] A. V. Galatenko, V. A. Nosov, A. E. Pankratiev, "Latin squares over quasigroups", *Lobachevskii Journal of Mathematics*, **41**:2 (2020), 194–203.

Bounds on the number of proper families of k-valued functions

Galatenko A.V.

Proper families of functions are a convenient apparatus for memory-efficient specification of large parametric families of quasigroups and d-quasigroups. K. D. Tsaregorodtsev proved that there exists a natural one-to-one correspondence between proper families of functions and unique sink orientations of a Boolean cube. The cardinality of such orientations was estimated by J. Matoušek. In our paper we extend Matoušek's lower bound to the case of k-valued logics for arbitrary $k > 2$, present a number of corollaries and prove that properness is a rare property, namely, the fraction of proper families of the size n in the class of all families of n-ary functions $(f_1, \ldots, f_n)$ such that x_i is dummy for $f_i(x_1, \ldots, x_n)$ tends to 0 as $n \to \infty$.

Keywords: proper families of functions, k-valued functions, hypergraph, matching.

References

[1] V. A. Nosov, "Regularity criterion for a non-autonomous Boolean automaton with split input", *Intellektualnye Systemy*, **3**:3–4 (1998), 269–280 (In Russian).

[2] V. A. Nosov, "Construction of classes of Latin squares in a Boolean database", *Intellektualnye Systemy*, **4**:3–4 (1999), 307–320 (In Russian).

[3] V. A. Nosov, A. E. Pankratiev, "Latin squares over Abelian groups", *Journal of Mathematical Sciences*, **149**:3 (2008), 1230–1234.

[4] I. A. Plaksina, "Construction of parametric families of multidimensional Latin squares", *Intellektualnye Systemy. Teoria i Prilojenia*, **18**:2 (2014), 323–329 (In Russian).

[5] A. V. Galatenko, V. A. Nosov, A. E. Pankratiev, K. D. Tsaregorodtsev, "Generation of n-quasigroups by the means of proper families of functions", *Diskrentnaya Matematika*, **35**:1 (2023), 35–53 (In Russian).

[6] A. V. Galatenko, V. A. Nosov, A. E. Pankratiev, K. D. Tsaregorodtsev, "Proper families of functions and their applications", *Matematicheskie Voprosy Kriptografii*, **14**:2 (2023), 43–58.

[7] K. D. Tsaregorodtsev, "One-to-one correspondence between proper families of Boolean functions and unique sink orientations of cubes", *Prikladnaya Diskrentnaya Matematika*, 2020, № 48, 16–21 (In Russian).

[8] J. Matoušek, "The number of unique-sink orientations of the hypercube", *Combinatorica*, **26** (2006), 91–99.

[9] A. S. Asratian, N. N. Kuzjurin, "On the number of nearly perfect matchings in almost regular uniform hypergraphs", *Discrete Mathematics*, **207**:1–3 (1999), 1–8.

[10] K. D. Tsaregorodtsev, "Properties of proper families of Boolean functions", *Discrete Mathematics and Applications*, **32**:5 (2022), 369–378.

[11] A. V. Galatenko, V. A. Nosov, A. E. Pankratiev, "Latin squares over quasigroups", *Lobachevskii Journal of Mathematics*, **41**:2 (2020), 194–203.

О функциональной системе, полученной из алгебры множеств добавлением индикаторов мощности

Ю. С. Капустин[1]

В данной работе исследуются свойства функциональной системы C_n с носителем $2^{\mathbb{Z}}$, порождённой теоретико-множественными функциями и индикаторами мощности $\mathbf{card}_0(x) \dots \mathbf{card}_n(x)$.

Ключевые слова: функциональная система, предполный класс, алгебра множеств, критерий полноты.

1. Введение

В работе [1] изучалась алгебраическая система с носителем $\mathbb{Z} \bigcup 2^{\mathbb{Z}}$, образованная отношениями и операциями, выразимыми с помощью логических связок, описателей и кванторов через отношение принадлежности. Было установлено, что любая кванторно определимая функция над множествами в этой системе может быть выражена через обычные теоретико-множественные функции и индикаторы мощности $\mathbf{card}_i(x)$, принимающие значение $\mathbb{Z}$, если множество x содержит ровно i элементов, и значение $\varnothing$ иначе.

Это привлекло внимание к изучению функциональной системы C_n с носителем $2^{\mathbb{Z}}$, порождённой теоретико-множественными функциями и индикаторами мощности $\mathbf{card}_0(x) \dots \mathbf{card}_n(x)$.

Получен ряд важных свойств этой функциональной системы. Найдено число функций в системе, предложен алгоритм решения уравнений в системе. Интересно, что число функций от заданного числа переменных в C_n имеет существенно больший порядок роста, чем для функций конечнозначных логик P_2 и P_k, в связи с чем её изучение не сводится к изучению указанных функциональных систем.

2. Основные понятия

Пусть $\mathbb{Z}$ — множество целых чисел. В качестве универсума возьмём $2^{\mathbb{Z}}$ — множество его подмножеств.

[1] *Капустин Юрий Сергеевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: kapustin.iu@yandex.ru

Kapustin Iurii Sergeevich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

На множестве $2^{\mathbb{Z}}$ естественным образом определены двухместные функции $a \bigcap b, a \bigcup b, a \backslash b$ и нульместная функция-константа $\mathbb{Z}$.

Запись $|x|$ обозначает число элементов в x.

Определим на этом множестве также счётное число функций $\mathbf{card}_k(a)$ (k — целый неотрицательный параметр) следующим образом:

$$\mathbf{card}_k(a) = \begin{cases} \mathbb{Z}, \text{ если } |a| = k \\ \varnothing, \text{ если } |a| \neq k. \end{cases}$$

Обозначим S_n — множество функций

$\{a \bigcap b, a \bigcup b, a \backslash b, \mathbb{Z}, \mathbf{card}_0(a), \ldots, \mathbf{card}_n(a)\}$, определённых на множестве $2^{\mathbb{Z}}$.

Обозначим через C_n функциональную систему с носителем $2^{\mathbb{Z}}$, порождённую функциями из S_n, то есть содержащую все функции, выразимые над S_n при помощи суперпозиции, и только их. Определение суперпозиции можно посмотреть в книге [3].

Обозначим S_C — множество функций $\{a \bigcap b, a \bigcup b, a \backslash b, \mathbb{Z}\}$. Функциональную систему с носителем $2^{\mathbb{Z}}$, порождённую функциями из S_C, обозначим через C. Как будет доказано далее, она изоморфна P_2.

Будем называть два терма равносильными, если они выражают одну и ту же функцию.

Через x^σ будем обозначать терм x, если σ — булева константа 1, и $(\mathbb{Z} \backslash x)$, если σ — булева константа 0. Обозначение $x_1^{\sigma_1} \ldots x_m^{\sigma_m}$ будет использоваться для терма $x_1^{\sigma_1} \bigcap \cdots \bigcap x_m^{\sigma_m}$.

Пусть рассматриваются функции из C_n от m переменных $x_1 \ldots x_m$, где n, m — фиксированные натуральные числа. Атомарным индикатором называется терм вида

$$\mathbf{card}_k(x_1^{\sigma_1} \ldots x_m^{\sigma_m})$$

для $0 \leq k \leq n$ или

$$\mathbb{Z} \backslash (\mathbf{card}_0(x_1^{\sigma_1} \ldots x_m^{\sigma_m}) \bigcup \cdots \bigcup \mathbf{card}_n(x_1^{\sigma_1} \ldots x_m^{\sigma_m})).$$

Для простоты будем обозначать $\mathbb{Z} \backslash (\mathbf{card}_0(x) \bigcup \cdots \bigcup \mathbf{card}_n(x))$ как $\mathbf{card}_{>n}(x)$. Значение этого выражения равно $\mathbb{Z}$, если множество x содержит более n элементов, и $\varnothing$ иначе.

Здесь и далее, когда упоминается функция $\mathbf{card}_l(x)$, где $l > n$, под ней подразумевается функция $\mathbf{card}_{>n}(x) \bigcap \mathbf{card}_{>n}(\mathbb{Z} \backslash x)$.

Например, если рассматриваются функции от переменных x_1, x_2 в C_2, терм

$$\mathbf{card}_1(x_1 \bigcap (\mathbb{Z} \backslash x_2))$$

— атомарный индикатор, а терм

$$\mathbf{card}_0(x_2)$$

— нет, так как не содержит переменной x_1.

Составным индикатором называется терм

$$\bigcap_{\sigma\in\{0,1\}^m} \mathbf{card}_{k_\sigma}(x_1^{\sigma_1}\dots x_m^{\sigma_m}),$$

где $\mathbf{card}_{k_\sigma}$ может означать $\mathbf{card}_0, \mathbf{card}_1, \dots, \mathbf{card}_n$ или $\mathbf{card}_{>n}$.

Например, если рассматриваются функции от переменной x_1 в C_2, терм

$$\mathbf{card}_1(x_1)\bigcap\mathbf{card}_2(\mathbb{Z}\backslash x_1)$$

— составной индикатор, а терм

$$\mathbf{card}_0(x_1)$$

— нет, так как не содержит атомарного индикатора для $\mathbb{Z}\backslash(x_1)$ (то есть для x_1^0).

Далее будет доказано, что если составной индикатор A_i не содержит $\mathbf{card}_{>n}$, то его значение — константа $\varnothing$ (поскольку объединение конечного число конечных множеств не может быть бесконечным множеством $\mathbb{Z}$).

Стандартной формой функции из C от переменных $x_1, \dots, x_m$ назовём терм вида $B_1\bigcup\dots\bigcup B_j$, где каждое B_i имеет вид $x_1^{\sigma_1}\dots x_m^{\sigma_m}$. Эта форма является аналогом ДНФ. Для функции-константы $\varnothing$ стандартной формой назовём терм $\mathbb{Z}\backslash\mathbb{Z}$. В дальнейшем этот терм будет обозначаться как $\varnothing$.

Нормальной формой m-местной функции из C_n называется терм вида $\bigcup_i(A_i\bigcap D_i)$, где A_i — составной индикатор, в терме участвуют все возможные составные индикаторы от m переменных, D_i — терм, выраженный в C. Если все термы D_i являются стандартной формой, назовём такую нормальную форму стандартной нормальной формой.

Пример: Пусть рассматривается функция от одной переменной x в C_0. Тогда терм

$$(\mathbf{card}_0(x)\bigcap\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap\mathbb{Z})\bigcup$$
$$(\mathbf{card}_0(x)\bigcap(\mathbb{Z}\backslash(\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap x))\bigcup$$
$$((\mathbb{Z}\backslash(\mathbf{card}_0(x)))\bigcap(\mathbb{Z}\backslash(\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap\mathbb{Z}))\bigcup$$
$$((\mathbb{Z}\backslash(\mathbf{card}_0(x)))\bigcap\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap\varnothing)$$

является нормальной формой, а терм

$$(\mathbf{card}_0(x)\bigcap\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap\mathbb{Z})\bigcup$$
$$(\mathbf{card}_0(x)\bigcap(\mathbb{Z}\backslash(\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap x))\bigcup$$
$$((\mathbb{Z}\backslash(\mathbf{card}_0(x)))\bigcap(\mathbb{Z}\backslash(\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap\mathbb{Z}))$$

— нет, так как содержит не все составные индикаторы.

3. Основные результаты

В данной статье доказываются следующие теоремы:

Теорема 1. *Любую функцию из C_n можно выразить термом в стандартной нормальной форме.*

Теорема 2. *Две стандартные нормальные формы $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ задают одну и ту же функцию тогда и только тогда, когда для каждого i (где индекс i параметризует всё множество составных индикаторов) верно одно из следующих утверждений:*

1) A_i не содержит $\mathbf{card}_{>n}$.

2) $D_i \equiv D'_i$

3) Все термы вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, где присутствуют все x_m, которые содержатся в только одном из термов D_i и D'_i, присутствуют в D_i в атомарном индикаторе $\mathbf{card}_0(x_1^{\sigma_1} \dots x_m^{\sigma_m})$.

Теорема 3. *Число функций от m переменных $x_1, ..., x_m$ в C_n равно $2^{(n+1)\cdot 2^m \cdot (n+2)^{2^m-1} - n\cdot 2^m \cdot (n+1)^{2^m-1}}$*

Пусть $O_1(x, y_1, \dots y_m), O_2(x, y_1, \dots y_m)$ — термы в C_n. Выражение $O_1(x, \overline{y}) = O_2(x, \overline{y})$ назовём уравнением относительно выбранной переменной x с параметрами $\overline{y}$. Пусть SP — некоторое множество предикатов. Будем говорить, что уравнение $O_1(x, \overline{y}) = O_2(x, \overline{y})$ имеет решение в множестве SP относительно переменной x, если предикат $O_1(x, \overline{y}) = O_2(x, \overline{y})$ выразим некоторой формулой над предикатами из SP. Эту формулу назовём решением уравнения $O_1(x, \overline{y}) = O_2(x, \overline{y})$ в множестве SP относительно переменной x.

Теорема 4. *Любое уравнение в C_n относительно переменной x с параметрами $x_1, \dots x_m$ имеет решение в множестве предикатов вида*

$$\mathbf{card}_{n_j}(x \bigcap F_j(x_1 \dots x_m)) = \mathbb{Z}$$

и

$$\mathbf{card}_{n_j}(F_j(x_1 \dots x_m) \backslash x) = \mathbb{Z},$$

где F_j — функция из C. При этом существует алгоритм, позволяющий найти это решение.

4. Число функций от m переменных в C_n

Чтобы определить число функций от m переменных, найдём стандартую форму, в которой выражается каждая функция из C, и нормальную форму, в которой выражается любая функция из C_n. Также найдём

необходимое и достаточное условие, при котором две стандартные формы задают одну и ту же функцию.

Во-первых, заметим, что проскольку

$$a\backslash b = a\bigcap(\mathbb{Z}\backslash b),$$

функции системы C порождаются также системой функций

$\{a\bigcap b, a\bigcup b, \mathbb{Z}\backslash b, \mathbb{Z}\}$. Рассмотрим формулы алгебры логики над набором функций:

$a\&b$, $a|b$, $\neg b$, 1

Рассмотрим оператор $G(f)$, сопоставляющий по индукции формуле алгебры логики над набором функций $\{a\&b,\ a|b,\ \neg b,\ 1\}$ терм из C_n над системой операций $\{a\bigcap b, a\bigcup b, \mathbb{Z}\backslash b, \mathbb{Z}\}$. Определим его (и обратное отображение) по индукции по длине формулы:

$G(x_i) = x_i, G^{-1}(x_i) = x_i$, если x — переменная.

$G(1) = \mathbb{Z}, G^{-1}(\mathbb{Z}) = 1$.

Если a, b — формулы алгебры логики над $\{a\&b,\ a|b,\ \neg b,\ 1\}$, c, d — термы над $\{a\bigcap b, a\bigcup b, \mathbb{Z}\backslash b, \mathbb{Z}\}$, то

$G(a|b) = G(a)\bigcup G(b), G^{-1}(c\bigcup d) = G^{-1}(c)|G^{-1}(d)$,

$G(a\&b) = G(a)\bigcap G(b), G^{-1}(c\bigcap d) = G^{-1}(c)\&G^{-1}(d)$,

$G(\neg a) = \mathbb{Z}\backslash G(a), G^{-1}(\mathbb{Z}\backslash c) = \neg G^{-1}(c)$.

Лемма 1. *Отображение G множества функций алгебры логики на множество функций в системе C, корректно определено и обратное отображение также корректно определено. То есть если две формулы f_1 и f_2 задают одну и ту же функцию алгебры логики, то термы $G(f_1)$ и $G(f_2)$ задают одну и ту же функцию. И наоборот, если два терма g_1 и g_2 в C_n задают одну и ту же функцию, то термы $G_1^{(-1)}(g_1)$ и $G_1^{(-1)}(g_2)$ задают одну и ту же функцию алгебры логики.*

Доказательство леммы.

Рассмотрим произвольную формулу алгебры логики $f_1(x_1 \dots x_n)$ над $\{\&, |, 1, \not\gamma$. Докажем индукцией по длине терма, что $e \in \mathbb{Z}$ принадлежит результату функции, выражаемой термом $G(f_1(x_1 \dots x_n))$ при тех и только тех значениях набора $x_1 \dots x_n$, при которых истинен результат формулы $f_1(y_1 \dots y_n)$, где $y_i = (e \in x_1)$.

База индукции — $(e \in x_i) \iff e \in (x_i)$ — очевидно выполняется.

Шаг индукции. Пусть a, b — два терма над $\{\bigcap, \bigcup, \backslash, \mathbb{Z}\}$. Тогда:

$e \in (a\bigcup b) \iff (e \in a)|(e \in b)$;

$e \in (a\bigcap b) \iff (e \in a)\&(e \in b)$;

$e \in \mathbb{Z} \iff 1$;

$e \in (\mathbb{Z}\backslash b) \iff \neg(e \in b)$ — эти утверждения также выполнены.

Следовательно, e принадлежит результату функции, выражаемой термом $G(f_1(x_1 \dots x_n))$ при тех и только тех значениях набора $x_1 \dots x_n$,

при которых истинен результат формулы $f_1(y_1 \dots y_n)$, где $y_i = (e \in x_1)$. Перейдём к доказательству самой леммы.

$\rightarrow$) Допустим, что две формулы f_1 и f_2 задают одну и ту же функцию. Результат функции, задаваемой термом $G(f_1)$ по доказанному ранее равен множеству тех элементов $e \in \mathbb{Z}$, для которых истинно значение формулы $f_1(y_1 \dots y_n)$, при $y_i = (e \in x_1)$, что полностью определает эту функцию. Результат функции, задаваемой термом $G(f_2)$ равен множеству тех e из $\mathbb{Z}$, для которых истинно значение формулы $f_2(y_1 \dots y_n)$ при $y_i = (e \in x_1)$. Поскольку формулы f_1 и f_2 задают одну и ту же функцию, то множества e из $\mathbb{Z}$, для которых значения $f_2(y_1 \dots y_n)$ и $f_1(y_1 \dots y_n)$ истинны, совпадают. Следовательно, термы $G(f_1)$ и $G(f_2)$ задают одну и ту же функцию.

$\leftarrow$) Допустим, что два терма g_1 и g_2 в C_n задают одну и ту же функцию, e — элемент $\mathbb{Z}$. Для любого набора $(x_1, \dots, x_n) \in \{0,1\}^n$ можно рассмотреть набор $(y_1, \dots, y_n) \in (2^{\mathbb{Z}})^n$:

$$y_i = \begin{cases} \{e\}, \text{если } x_1 = 1 \\ \varnothing, \text{если } x_1 = 0. \end{cases}$$

Тогда $G^{-1}(g_1(x_1 \dots x_n)) = e \in g_1(y_1 \dots y_n) = e \in g_2(y_1 \dots y_n) = G^{-2}(g_1(x_1 \dots x_n))$.

Отсюда из равенства функций, задаваемых термами g_1 и g_2 следует равенство значений функций, задаваемых формулами $G^{-1}(g_1)$ и $G^{-1}(g_2)$, на любом наборе $(x_1, \dots, x_n) \in \{0,1\}^n$. Следовательно, эти функции равны, ч.т.д.

Следствие 1. *В системе C 2^{2^m} различных m-местных функций.*

В моей статье [1] при доказательстве леммы 4 из теорем было доказано следующее утверждение:

Лемма 2. *(О разложении) Для любого терма T от переменных $x_1, \dots x_m$ в C можно найти такой набор $N(T)$ термов вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, что при любом значении набора переменных $x_1, \dots, x_m$ значения термов из $N(T)$ — непересекающиеся множества и значение их объединения равно значению T.*

1) Для любых $a_1, \dots a_m \in 2^{\mathbb{Z}}$ и двух различных термов $T = x_1^{\sigma_1} \dots x_m^{\sigma_m}$ и $T' = x_1^{\sigma'_1} \dots x_m^{\sigma'_m}$ значения этих двух термов - непересекающиеся множества.

Действительно, пусть, без ограничения общности, $\sigma'_i \neq \sigma_i$, $\sigma'_i = 1, \sigma_i = 0$. Тогда $T(a_1, \dots, a_n) = a_1^{\sigma_1} \dots a_m^{\sigma_m} \in \mathbb{Z} \backslash a_i$ $T'(a_1, \dots, a_n) = a_1^{\sigma'_1} \dots a_m^{\sigma'_m} \in a_i$. Так как a_i и $\mathbb{Z} \backslash a_i$ не пересекаются, $T'(a_1, \dots, a_n)$ и $T(a_1, \dots, a_n)$ не пересекаются.

2) Пусть T – терм в над S_C. К нему можно последовательно применить следующие преобразования:

– заменить все подтермы вида $a\backslash b$, где a, b - термы, $a \neq \mathbb{Z}$ на подтермы $a \bigcap (Z\backslash b)$.

– заменить все подтермы вида $Z\backslash(a \bigcup b)$ на $(Z\backslash a) \bigcap (Z\backslash b)$ и все подтермы вида $Z\backslash(a \bigcap b)$ на $(Z\backslash a) \bigcup (Z\backslash b)$. Повторять процедуру, пока не останется операций $Z\backslash$, внешних по отношению к $\bigcap, \bigcup$.

– заменить все подтермы вида $Z \bigcap (a \bigcup b)$ на $(Z \bigcup a) \bigcap (Z \bigcup b)$. Повторять процедуру, пока не останется операций $\bigcap$, внешних по отношению к $\bigcup$. Получим терм вида $B_1 \bigcup B_n$ (или B_1), где B_i имеет вид пересечения термов $x_i^{\sigma_{ij}}$ и $Z\backslash Z$.

– Если пересечение B_i содержит $\mathbb{Z}\backslash\mathbb{Z}$, то он равносилен $\varnothing$ и его можно убрать из пересечения. Если при этом B_i единственный, то исходный терм тождественно равен пустому объединению.

– Если пересечение B_i не содержит $x_i^{\sigma_i}$, заменить его на
$(B_i \bigcap x_i) \bigcup (B_i \bigcap (\mathbb{Z}\backslash x_i))$.

В результате получим терм-объединение термов вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$. Значение его на любом наборе значений переменных равно объединению их значений. Лемма доказана.

Лемма доказана.

Теперь найдём такую форму, что для каждой функции в C_n, найдётся терм в этой форме.

Обозначение x^σ будет использоваться для терма:
x, если $\sigma = 1$;
$(\mathbb{Z}\backslash x$, если $\sigma = 0$.
Обозначение $x_1^{\sigma_1} \dots x_m^{\sigma_m}$ будет использоваться для терма

$$x_1^{\sigma_1} \bigcap x_2^{\sigma_2} \cdots \bigcap x_m^{\sigma_m}.$$

Лемма 3. *Для каждой функции из C_n от переменных $x_1 \dots x_m$, найдётся выражающий её терм над S_n, в котором операция* **card** *применяется только к термам вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$.*

Доказательство.

Допустим, некоторая функция выражается термом g в C_n. Покажем, что её можно выразить термом, в котором нет вложенных **card**.

Действительно, если в терм g входит терм $\mathbf{card}_l(g')$, то терм g задаёт ту же функцию, что и терм

$$(\mathbf{card}_l(g') \bigcap g|_{\mathbf{card}_l(g')=\mathbb{Z}}) \bigcup (g|_{\mathbf{card}_l(g')=\varnothing} \backslash \mathbf{card}_l(g')),$$

где через

$$g|_{\mathbf{card}_l(g')=\mathbb{Z}}$$

обозначен терм, получающийся из g путём замены $\mathbf{card}_l(g')$ на $\mathbb{Z}$; константа $\varnothing$ выражается как $\mathbb{Z}\backslash\mathbb{Z}$. Назовём замену этого типа заменой вынесения $\mathbf{card}_l(g')$. Эта замена не меняет выражаемую термом функцию, поскольку для тех значений переменных, для которых значение терма g равно $\mathbb{Z}$, значение обоих термов равно значению терма $g|_{\mathbf{card}_l(g')=\mathbb{Z}}$, а для тех значений переменных, для которых значение терма g равно $\varnothing$, значение обоих термов равно значению терма $g|_{\mathbf{card}_l(g')=\varnothing}$.

Пусть максимальная вложенность операций **card** в терме T равна k, $k > 1$. Тогда если последовательно применить к терму T замену вынесения $\mathbf{card}_l(g')$ для каждого подтерма вида $\mathbf{card}_l(g')$, где g не содержит операций **card**, получим терм с максимальной вложенностью операций **card**, равной $k-1$. Повторяя эту процедуру, получим терм (обозначим его g''), в котором операция **card** будет применяться только к термам над S_C.

Например,

$$\mathbf{card}_0(x_1 \bigcup (\mathbf{card}_1(x_2))) = (\mathbf{card}_1(x_2) \bigcap \mathbf{card}_0(x_1 \bigcup \mathbb{Z})) \bigcup$$

$$(\mathbf{card}_0(x_1 \bigcup \varnothing) \backslash \mathbf{card}_1(x_2)).$$

Пусть терм g'' содержит подтерм $\mathbf{card}_l(g''')$. По лемме о разложении, существует множество $N(g''')$ термов вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, значения которых при любом значении переменных — непересекающиеся множества и в объединении дают значение g'''. Пусть $N(g''')$ содержит k термов g_j. Пусть $(l_{i1}, \dots, l_{ik})$ — все возможные наборы целых неотрицательных чисел, сумма которых равна l.

Тогда терм $\mathbf{card}_l(g''')$ равносилен терму

$$\bigcap_i (\mathbf{card}_{l_{i1}}(g_1) \bigcup \dots \bigcup \mathbf{card}_{l_{ik}}(g_k)).$$

Таким образом терм $\mathbf{card}_l(g''')$ равносилен терму, выразимому через термы $\mathbf{card}_{n_i}(g_i)$, где все g_i имеют вид $x_1^{\sigma_1} \dots x_m^{\sigma_m}$ и принадлежат $N(g''')$, а все n_i не больше l.

Например,

$$\mathbf{card}_2(x_2) = (\mathbf{card}_0(x_2 \bigcap x_1) \bigcap \mathbf{card}_2(x_2 \backslash x_1)) \bigcup (\mathbf{card}_1(x_2 \bigcap x_1)$$

$$\bigcap \mathbf{card}_1(x_2 \backslash x_1)) \bigcup (\mathbf{card}_2(x_2 \bigcap x_1) \bigcap \mathbf{card}_0(x_2 \backslash x_1)).$$

Таким образом каждую функцию в C_n можно выразить термом над S_n, в котором операция **card** применяется только к термам вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$. Лемма доказана.

Чтобы найти точное число функций в C_n от m переменных, найдём стандартную форму для таких термов.

Назовём терм $\mathbf{card}_k(x_1^{\sigma_1} \dots x_m^{\sigma_m})$ или $\mathbb{Z}\backslash(\mathbf{card}_0(x_1^{\sigma_1} \dots x_m^{\sigma_m}) \bigcup \dots \bigcup \mathbf{card}_n(x_1^{\sigma_1} \dots x_m^{\sigma_m}))$ атомарным индикатором, если $x_1 \dots x_m$ — все переменные. Для простоты будем обозначать $\mathbb{Z}\backslash(\mathbf{card}_0(x_1^{\sigma_1} \dots x_m^{\sigma_m}) \bigcup \dots \bigcup \mathbf{card}_n(x_1^{\sigma_1} \dots x_m^{\sigma_m}))$ как $\mathbf{card}_{>n}(x_1^{\sigma_1} \dots x_m^{\sigma_m})$. Поскольку существует 2^m возможных значений для $\sigma_1 \dots \sigma_m$ и $n+2$ различных операций $\mathbf{card_0}, \dots, \mathbf{card_n}, \mathbf{card}_{>\mathbf{n}}$, всего существует $(n+2) \cdot (2^m)$ атомарных индикаторов от m данных переменных в C_n.

Составным индикатором назовём терм $\bigcap_{\sigma \in \{0,1\}^m} \mathbf{card}_{k_\sigma}(x_1^{\sigma_1} \dots x_m^{\sigma_m})$, в котором функцией $\bigcap$ соединены атомарные индикаторы для всех значений параметров $\sigma \in \{0,1\}^m$; $\{0,1\}^m$ — булев куб; $\mathbf{card}_{k_\sigma}$ может означать $\mathbf{card}_0, \dots, \mathbf{card}_n$ или $\mathbf{card}_{>n}$. При этом будем считать равными составные индикаторы,которые различаются лишь порядком множителей во внешней операции пересечения. Всего существует $(n+2)^{(2^m)}$ различных составных индикаторов, поскольку каждый индикатор определяется 2^m параметрами k_σ, каждый из которых может принимать одно из $n+2$ значений — либо число от 0 до n, либо "$> n$".

Также заметим, что атомарные и составные индикаторы могут принимать только значения $\mathbb{Z}$ или $\varnothing$.

Пусть составные индикаторы параметризуются индексом i. Нормальной формой функции из C_n называется терм вида $\bigcup_i (A_i \bigcap D_i)$, где A_i — составной индикатор, соответствующий параметру i, D_i — формула, выраженная в C, в терме используются все возможные составные индикаторы. Если все D_i выражены в стандартной форме, назовём такую нормальную форму стандартной нормальной формой.

Лемма 4. *Любую функцию из C_n можно выразить термом в нормальной форме.*

Доказательство. Для каждого набора значений переменных $x'_1 \dots x'_m$ каждый из термов $x_1^{'\sigma_1} \dots x_m^{'\sigma_m}$ имеет одно значение — множество, имеющее определённое конечное или счётное число элементов $|x_1^{'\sigma_1} \dots x_m^{'\sigma_m}|$. Каждому набору значений переменных $x'_1 \dots x'_m$ таким образом можно сопоставить ровно один составной индикатор $\bigcap_{\sigma \in \{0,1\}^m} \mathbf{card}_{|x_1^{'\sigma_1} \dots x_m^{'\sigma_m}|}(x_1^{\sigma_1} \dots x_m^{\sigma_m})$, значение которого на этом наборе равно $\mathbb{Z}$ (здесь $\mathbf{card}_{|x_1^{'\sigma_1} \dots x_m^{'\sigma_m}|}$ считается принимающим значение или до n или значение $> n$)). Значение остальных составных индикаторов на этом наборе равно $\varnothing$. Следовательно, два различных составных индикатора не могут принимать значение, не равное $\varnothing$, на одном наборе значений переменных.

Пример. Набору значений переменных $x'_1 = \{1\}, x'_2 = \{1,2\}$ в C_1 соответствует составной индикатор

$$\mathbf{card}_0(x_1 \bigcap (\mathbb{Z} \backslash x_2)) \bigcap \mathbf{card}_1(x_1 \bigcap x_2) \bigcap \mathbf{card}_{>1}((\mathbb{Z} \backslash x_1)$$

$$\bigcap (\mathbb{Z} \backslash x_2)) \bigcap \mathbf{card}_1((\mathbb{Z} \backslash x_1) \bigcap x_2)$$

Составной индикатор — пересечение атомарных индикаторов, которые (поскольку для них **card** — внешняя функция) могут принимать только значения $\mathbb{Z}$ или $\varnothing$. Если атомарный индикатор $\mathbf{card}_l(x_1^{\sigma_1} \dots x_m^{\sigma_m})$ входит в составной индикатор I, то на наборе значений переменных $(x'_1, \dots, x'_m)$, на котором составной индикатор принимает значение $\mathbb{Z}$, атомарный индикатор принимает значение $\mathbb{Z}$.

Если же этот атомарный индикатор не входит в I, то в него входит другой индикатор $\mathbf{card}_{l_i}(x_1^{\sigma_1} \dots x_m^{\sigma_m})$, $l_i \neq l$. Поскольку $x_1^{\sigma_1} \dots x_m^{\sigma_m}$ не может иметь различное число элементов при одном и том же значении $x_1 \dots x_m$, то на наборе значений переменных, на котором составной индикатор принимает значение $\mathbb{Z}$, атомарный индикатор значение $\varnothing$.

Следовательно, для каждого составного индикатора и каждого атомарного индикатора на всех наборах значений переменных, на которых составной индикатор принимает значение $\mathbb{Z}$, атомарный индикатор принимает одно и то же значение. Это значение — $\mathbb{Z}$, если атомарный индикатор входит в составной, и $\varnothing$ иначе.

Рассмотрим функцию $O(x_1, \dots, x_m)$. По предыдущей лемме без ограничения общности можно считать, что она выражена термом $O'(x_1, \dots, x_m)$, в котором под **card** находятся только термы $x_1^{\sigma_1} \dots x_m^{\sigma_m}$. То есть все вхождения **card** в этот терм — атомарные индикаторы. Для каждого составного индикатора A_i обозначим за D_i терм, который получается из $O'(x_1, \dots, x_m)$ путём замены содержащихся в A_i атомарных индикаторов на $\mathbb{Z}$, а не содержащихся — на $\varnothing$. В этом случае каждое D_i выражено только через функции из S_C, и $\bigcup(A_i \bigcap D_i)$ — нормальная форма. Покажем, что $\bigcup(A_i \bigcap D_i)$ — нормальная форма для O, то есть выражает O.

Действительно, рассмотрим набор значений N переменых $x_1, \dots, x_m$. Пусть A_k — тот единственный составной индикатор, который на данном наборе принимает значение $\mathbb{Z}$. Тогда значение $(A_i \bigcap D_i)(N)$ равно $\varnothing$ при i не равном k, а значение $\bigcup(A_k \bigcap D_k)$ равно $D_k(N)$. Из определения D_k, $D_k(N) = O(N)$. Следовательно, на любом наборе N $O(N) = \bigcup(A_i \bigcap D_i)(N)$. То есть нормальная форма $\bigcup(A_i \bigcap D_i)$ задаёт функцию O, ч.т.д.

Будем называть две нормальные формы $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ равными, если термы D_i и D'_i выражают одну и ту же функцию для каждого

i. В противном случае будем называть их различными. Стандартной нормальной формой назовём такую форму, где каждое D_i выражено в виде, аналогичном ДНФ, то есть в виде $\bigcup(x_1^{\sigma_1}, \ldots, x_1^{\sigma_n})$. Несложно убедиться, что для любой нормальной формы существует равная ей стандартная нормальная форма.

Заметим, что две различные нормальные формы могут задавать одну и ту же функцию. Например, в C_0

$$\bigcup_i (A_i \bigcup \varnothing),$$

$$(\mathbf{card}_0(x) \bigcap \mathbf{card}_{>0}(\mathbb{Z}\backslash x) \bigcap x) \bigcup (\bigcup_j (A_j \bigcup \varnothing))$$

и

$$(\mathbf{card}_0(x) \bigcap \mathbf{card}_0(\mathbb{Z}\backslash x) \bigcap \mathbb{Z}) \bigcup (\bigcup_k (A_k \bigcup \varnothing))$$

задают одну и ту же функцию. Но можно показать, что подобные пары форм — единственные различные формы, выражающие одну и ту же функцию.

Лемма 5. $\bigcup_{(\sigma_1 \ldots \sigma_m) \in \{0,1\}^m} (x_1^{\sigma_1} \ldots x_m^{\sigma_m}) = \mathbb{Z}$ *для любого значения переменных* $x_1 \ldots x_m$

Докажем индукцией по числу m переменных. Если $m = 1$, $(\mathbb{Z}\backslash x_1) \bigcup x_1 = \mathbb{Z}$ — утверждение леммы верно.

Если утверждение леммы верно для m, то для $m' = m + 1$
$\bigcup_{(\sigma_1 \ldots \sigma_{m'}) \in \{0,1\}^{m'}} (x_1^{\sigma_1} \ldots x_{m'}^{\sigma_{m'}}) = \mathbb{Z} =$
$((\bigcup_{(\sigma_1 \ldots \sigma_m) \in \{0,1\}^m} (x_1^{\sigma_1} \ldots x_m^{\sigma_m})) \bigcap (x_{m'})) \bigcup$
$((\bigcup_{(\sigma_1 \ldots \sigma_m) \in \{0,1\}^m} (x_1^{\sigma_1} \ldots x_m^{\sigma_m})) \bigcap (\mathbb{Z}\backslash x_{m'})) =$
$((\bigcup_{(\sigma_1 \ldots \sigma_m) \in \{0,1\}^m} (x_1^{\sigma_1} \ldots x_m^{\sigma_m})) \bigcap (x_{m'} \bigcup (\mathbb{Z}\backslash x_{m'})) =$
$= ((\bigcup_{(\sigma_1 \ldots \sigma_m) \in \{0,1\}^m} (x_1^{\sigma_1} \ldots x_m^{\sigma_m})) = \mathbb{Z}$. Лемма доказана.

Лемма 6. *Если составной индикатор* A_i *не содержит* $\mathbf{card}_{>n}$, *то* $A_i \equiv \varnothing$

Доказательство. Поскольку по предыдущей лемме объединение конечного числа множеств $x_1^{\sigma_1} \ldots x_m^{\sigma_m}$ равно бесконечному множеству $\mathbb{Z}$. Следовательно, хотя бы одно из них — пусть это будет $x_1^{\sigma'_1} \ldots x_m^{\sigma'_m}$ бесконечно. Тогда соответствующий атомарный индикатор $\mathbf{card}_l(x_1^{\sigma'_1} \ldots x_m^{\sigma'_m})$ принимает значение $\varnothing$, и весь составной индикатор A_i принимает значение $\varnothing$.

Лемма 7. *Две стандартные нормальные формы* $\bigcup(A_i \bigcap D_i)$ *и* $\bigcup(A_i \bigcap D'_i)$ *задают одну и ту же функцию тогда и только тогда, когда для каждого* i *(где индекс* i *параметризует всё множество составных индикаторов) верно одно из следующих утверждений:*

1) A_i *не содержит* $\mathbf{card}_{>n}$.

2) $D_i \equiv D'_i$

3) Все термы вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$*, где присутствуют все* x_m*, которые содержатся в только одном из термов* D_i *и* D'_i*, присутствуют в* D_i *в атомарном индикаторе* $\mathbf{card}_0(x_1^{\sigma_1} \dots x_m^{\sigma_m})$.

Доказательство. $\leftarrow$) Пусть две формы $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ таковы, что для каждого i одно из утверждений (1) – (3) верно. Рассмотрим набор значений переменных N и соответствующий ему составной индикатор A_i, который принимает на нём значение $\mathbb{Z}$.

Для этого набора согласно предыдущей лемме не может выполняться 1).

Если для него выполняется 2), то, поскольку $D_i \equiv D'_i$, верно равенство $A_i(N) \bigcap D_i(N) = A_i(N) \bigcap D'_i(N)$.

Если для него выполняется 3), то $(A_i \bigcap D_i)(N) \equiv (A_i \bigcap D'_i)(N)$, поскольку обе части являются объединением одних и тех же непустых множеств и некоторого количества пустых.

Следовательно, $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ принимают одно и то же значение на любом значении переменных N, и выражаемые этими термами функции совпадают.

$\rightarrow$) От противного. Пусть две стандартные нормальные формы $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ задают одну и ту же функцию. Пусть существует i, для которого A_i содержит $\mathbf{card}_{>n}$, и, без ограничения общности, в терме D_i содержится терм $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, который не содержится в D'_i и присутствует в A_i в атомарном индикаторе $\mathbf{card}_k(x_1^{\sigma_1} \dots x_m^{\sigma_m})$, где k не равно 0.

Пусть $x_1 \dots x_m$ — набор значений переменных, соответствующий A_i (то есть тот, на котором $A_i = \mathbb{Z}$). Такой набор существует, поскольку можно найти 2^m непересекающихся множеств с заданным числом элементов у каждого (хотя бы одно из которых бесконечно), объединение которых равно $\mathbf{Z}$. $(A_i \bigcap D_i)(N)$ содержит элемент из множества $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, $(A_i \bigcap D'_i)(N)$ не содержит элемент из $x_1^{\sigma_1} \dots x_m^{\sigma_m}$. При этом никакие другие A_i не содержат элемент из $x_1^{\sigma_1} \dots x_m^{\sigma_m}$. Следовательно, $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ задают разные функции. Лемма доказана.

С учётом этой леммы найдём число функций в C_n от m переменных.

Теорема 1. *Число функций от* m *переменных* $x_1, ..., x_m$ *в* C_n *равно* $2^{(n+1)\cdot 2^m \cdot (n+2)^{2^m-1} - n \cdot 2^m \cdot (n+1)^{2^m-1}}$

Доказательство.

Как было указано ранее, всего существует $(n+2)^{(2^m)}$ различных составных индикаторов.

Зафиксируем A_i и найдём количество функций типа $(A_i \bigcap D)$. Обозначим его $F(i)$. Если A_i не содержит $\mathbf{card}_{>n}$, то значение A_i всегда равно $\varnothing$, $(A_i \bigcap D)$ — константа $\varnothing$, $F(i) = 1$. Если A_i содержит $\mathbf{card}_{>n}$, и j — число $\mathbf{card}_0$ в A_i, то $F(i) = 2^{k-j}$ (поскольку наличие или отсутствие в D подтерма $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, для которого A_i имеет подтерм $\mathbf{card}_0(x_1^{\sigma_1} \dots x_m^{\sigma_m})$, не влияет на значение функции), и $k - j = log_2(F(i))$.

Количество же всех m-местных функций $N(n,m)$ равно $\Pi_i F(i) = 2^{\Sigma_i log_2(F(i))}$. (*)

Обозначим $l = n + 2$ — число различных индексов для $\mathbf{card}$, включая $> n$.

Поскольку в составном индикаторе A_i при фиксированных j и k есть:

— C_k^j различных вариантов, где расположены j различных нулевых $\mathbf{card}$,

— $(l-1)^{k-j}$ варианта для значения ненулевых параметров $\mathbf{card}$,

— $(l-2)^{k-j}$ варианта для значения ненулевых параметров $\mathbf{card}$, в которых нет $\mathbf{card}_{>n}$,

— $(l-1)^{k-j} - (l-2)^{k-j}$ варианта для значения ненулевых параметров $\mathbf{card}$, среди которых есть хотя бы одно $\mathbf{card}_{>n}$,

получим:

$N(n,m) = 2^{\Sigma_i log_2(F(i))} = 2^{\Sigma_{j=0}^{k} C_k^j \cdot ((l-1)^{k-j} - (l-2)^{k-j}) \cdot (k-j)} =$

$2^{\Sigma_{j=0}^{k} C_k^{k-j} \cdot ((l-1)^{k-j} - (l-2)^{k-j}) \cdot (k-j)} = 2^{\Sigma_{j'=0}^{k} C_k^{j'} \cdot ((l-1)^{j'} - (l-2)^{j'}) \cdot (j')}$ (*)

Чтобы найти эту сумму, найдём значение суммы $\Sigma_{i=0}^{n}(C_n^i \cdot x^i \cdot i)$ для произвольного натурального i и вещественного x. Для этого воспользуемся фактом из математического анализа, что производная суммы дифференцируемых функций равна сумме их производных:

$\Sigma_{i=0}^{n}(C_n^i \cdot x^i \cdot i) = x \cdot \Sigma_{i=0}^{n}(C_n^i \cdot x^{i-1} \cdot i) = x \cdot \Sigma_{i=0}^{n}(C_n^i \cdot (x^i)'_x) = x \cdot (\Sigma_{i=0}^{n} C_n^i \cdot (x^i))'_x = x \cdot ((x+1)^n)'_x = x \cdot n \cdot (x+1)^{n-1}$ (**)

Таким образом, значение выражения (*) равно

$2^{(l-1)*k*l^{k-1} - (l-2)*k*(l-1)^{k-1}} = 2^{(n+1)\cdot 2^m \cdot (n+2)^{2^m-1} - n \cdot 2^m \cdot (n+1)^{2^m-1}}$

Теорема доказана.

5. Уравнения в C_n

Пусть $O_1(x, y_1, \dots y_m), O_2(x, y_1, \dots y_m)$ — термы в C_n. Выражение $O_1(x, \overline{y}) = O_2(x, \overline{y})$ назовём уравнением относительно выбранной переменной x с параметрами $\overline{y}$. Пусть SP — некоторое множество предикатов. Будем говорить, что уравнение $O_1(x, \overline{y}) = O_2(x, \overline{y})$ имеет решение в множестве SP относительно переменной x, если предикат $O_1(x, \overline{y}) = O_2(x, \overline{y})$

выразим некоторой формулой над предикатами из SP. Эту формулу назовём решением уравнения $O_1(x, \overline{y}) = O_2(x, \overline{y})$ в множестве SP относительно переменной x.

Теорема 2. *Любое уравнение в C_n относительно переменной x с параметрами $x_1, \dots x_m$ имеет решение в множестве предикатов вида*

$$\mathbf{card}_{n_j}(x \bigcap F_j(x_1 \dots x_m)) = \mathbb{Z}$$

и

$$\mathbf{card}_{n_j}(F_j(x_1 \dots x_m) \backslash x) = \mathbb{Z},$$

где F_j — функция из C. При этом существует алгоритм, позволяющий найти это решение.

Доказательство. Сначала докажем лемму

Лемма 8. *Существует алгоритм, с помощью которого любой терм $T(x, x_1, \dots, x_n)$ можно привести к стандартной форме (то есть найти терм в стандартной форме, выражающий ту же функцию).*

Один из возможных алгоритмов выглядит следующим образом:

1) Рассмотреть все составные индикаторы A_i от переменных $x, x_1, \dots, x_n$. Записать терм $\bigcup_i (A_i \bigcap T)$.

2) Преобразовать каждый терм $A_i \bigcap T_i$ следующим образом (терм T_i может меняться между шагами алгоритма):

— Пока в рассматриваемый терм T_i входит **card**:

—— Найти в нём вхождение вида $\mathbf{card}_k(T')$, где в T' не входит никакой другой **card** (то есть T' выражается над $\{\mathbb{Z} \backslash, \bigcap, \bigcup\}$)

—— Найти, объединением каких пересечений вида $x_1^{\sigma_1} \bigcap \dots \bigcap x_m^{\sigma_m}$ является T' (из изоморфизма C и P_2 это делается аналогично приведению формулы из P_2 к СКНФ), просуммировать по j индексы k_{ij} из входящих в A_i термов $\mathbf{card}_{k_{ij}}(a_i)$.

—— Если результат равен индексу k (или $> n$, если k — индекс "$> n$"), заменить в рассматриваемом терме $\mathbf{card}_k(T')$ на $\mathbb{Z}$. Иначе заменить его на $\varnothing$.

В результате получим равносильный T терм $\bigcup_i (A_i \bigcap T_i')$ в нормальной форме.

3) Аналогично алгоритму приведения функции алгебры логики к СКНФ, привести T_i к виду, аналогичному СКНФ.

В результате получится терм, равносильный T и имеющий стандартную нормальную форму, ч.т.д. Лемма доказана.

Как следует из леммы, без ограничения общности можно считать, что в выражении $O_1(x, x_1, \dots x_m) = O_2(x, x_1, \dots x_m)$ оба терма O_1 и O_2

записаны в стандартной форме. То есть достаточно решать уравнения вида $\bigcup(A_i \bigcap D_i) = \bigcup(A_i \bigcap D'_i)$.

Как было показано ранее, чтобы набор $x_1, \ldots, x_n$, на котором выполнено $A_j(x_1, \ldots, x_n) = \mathbb{Z}$, удовлетворял равенству $\bigcup(A_i \bigcap D_i) = \bigcup(A_i \bigcap D'_i)$, необходимо и достаточно, чтобы любой атомарный терм $x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}$, который входит в D_i, но не в D'_i, или наоборот, входил в A_i в виде $\mathbf{card}_0(x_1^{\sigma_1}, \ldots, x_n^{\sigma_n})$. Рассмотрим B — множество всех A_i, для которых любой атомарный терм $x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}$, который входит в D_i, но не в D'_i, или наоборот, входит в A_i в виде $\mathbf{card}_0(x_1^{\sigma_1}, \ldots, x_n^{\sigma_n})$.

Если на наборе $(x'_1, \ldots, x'_n)$ принимает значение $\mathbb{Z}$ такой A_i, то $\bigcup(A_j \bigcap D_j)(x'_1, \ldots, x'_n) = \varnothing \bigcup \varnothing \cdots \bigcup \varnothing \bigcup (A_i \bigcap D_i)(x'_1, \ldots, x'_n)) = D_i(x'_1, \ldots, x'_n) = D'_i(x'_1, \ldots, x'_n) = \bigcup(A'_j \bigcap D'_j)(x'_1, \ldots, x'_n)$.

Если же на наборе $(x'_1, \ldots, x'_n)$ принимает значение $\mathbb{Z}$ A_i, не удовлетворяющий этому свойству, то $\bigcup(A_j \bigcap D_j)(x'_1, \ldots, x'_n) = \varnothing \bigcup \varnothing \cdots \bigcup \varnothing \bigcup (A_i \bigcap D_i)(x'_1, \ldots, x'_n)) = D_i(x'_1, \ldots, x'_n) \neq D'_i(x'_1, \ldots, x'_n) = \bigcup(A'_j \bigcap D'_j)(x'_1, \ldots, x'_n)$.

Следовательно, равенство истинно на наборе $(x'_1, \ldots, x'_n)$ если и только если $(x'_1, \ldots, x'_n) \in A_i$ и $A_i \in B$. Решение равносильно формуле $\bigvee_{A_i \in B}(A_i = \mathbb{Z})$.

Заменив $\bigvee(\bigcap(I_{ij}) = \mathbb{Z})$ на $\bigvee(\&(I_{ij} = \mathbb{Z}))$, где I_{ij} — атомарные индинаторы, получим решение уравнения.

6. Заключение

В следующих статьях будут описаны свойства функциональной системы C_n, представлена шефферова функции в ней. Будет представлена серия предполных классов, позволяющая получить критерий относительной полноты.

Автор выражает благодарность профессору А.С. Подколзину за постановку задачи и помощь в работе.

Список литературы

[1] Капустин Ю. С., “Об элементарной выразимости в логике предикатов”, *Интеллектуальные системы. Теория и приложения*, **23**:2 (2019), 135–158.

[2] Яблонский С.В., *Введение в дискретную математику*, «Высшая школа», М, 2003, 384 с.

[3] Яблонский С.В, Грврилов Г.П., Кудрявцев В. Б., *Функции алгебры логики и классы Поста*, «Наука», М, 1966, 120 с.

On algebraic system created from set algebra by adding the set power indicator

Kapustin I.S.

This paper concerns the properties of the functional system C_n. This system has the domain $2^{\mathbb{Z}}$, and is generated by functions $2^{\mathbb{Z}} \backslash x, x \bigcup y, x \bigcap x$ and power indicators $\mathbf{card}_0(x) \dots \mathbf{card}_n(x)$.

Keywords: functional system, precomplete class, set algebra, completion criteria.

References

[1] Kapustin I. S., "On the elementary expressibility in predicate logic", *Intelligent Systems. Theory and Applications*, **23**:2 (2019), 135–158

[2] Yablonsky S.V., *Introduction to discrete mathematics*, «Vysshaya shkola», M, 2003, 384 c.

[3] Yablonsky S.V., Gavrilov G.P., Kudryavtsev V. B., *Functions of the Algebra of Logic and the Post Classes*, «Nauka», M, 1966, 120 c.

Реализация алгоритмов схемами из функциональных элементов

М. В. Носов[1]

В работе построена схема из функциональных элементов для машины Тьюринга с сохранением условия полиномиальности, если таким свойством обладала машина.

Ключевые слова: машина Тьюринга, схема из функциональных элементов.

Пусть есть задача, которая всегда имеет определённый ответ. Есть алгоритм решения этой задачи, соответствующая машина Тьюринга всегда останавливается в заключительном состоянии и ответ задачи — содержимое соответствующей ячейки ленты. Задача имеет характеристики и входные параметры. Например, возьмём известную задачу: можно ли набор натуральных чисел разделить на два набора с одинаковой суммой чисел. Эта задача всегда имеет определённый ответ. В такой постановки имеется две характеристики: количество чисел в исходном наборе и верхняя граница чисел. Входные параметры — сами целые числа в диапазоне от 1 до верхней границы.

Сложность алгоритма — минимальное количество шагов машины Тьюринга при определённых характеристиках и любых допустимых наборах параметров; пусть сложность строго мажорируется величиной r, зависящей от характеристик. Пусть машина имеет алфавит из m символов, перенумеруем их числами $1, \ldots, m$. Пусть машина имеет k состояний, перенумеруем их числами $1, \ldots, k$, причём k — номер заключительного состояния. Имеется три движения: 1 — вправо, 2 — влево, 3 — отсутствие движения. В начальный момент головка стоит на самом левом непустом символе,тогда зона работы машины всегда находится между ячейкой, расположенной от начальной ячейки на r ячеек влево и на r ячеек вправо. Таким образом, головка может находится только в ячейках, расположенных среди этих $2r + 1$ ячейках, так и перенумеруем их слева направо $1, \ldots, 2r + 1$, но головка никогда не попадёт в ячейки с номерами 1 и $2r + 1$ в силу выбора r.

Первая задача состоит в построении схемы из функциональных элементов в классическом базисе, которая будет выдавать такой же результат, как и машина Тьюринга и, главное, если алгоритм полиномиален, то

[1] *Носов Михаил Васильевич* — с.н.с. каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: mvnosov@rambler.ru.

Nosov Michail Vasilevich-senior researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

полиномиальной будет схема. В начале изложим очень грубо идею построения схемы с некоторой вольностью в символах чтобы не загромождать формулы. Пусть машина Тьюринга имеет n команд $(q'aq''bd)_t, t = 1, \ldots, n$. В грубой схеме будет $2(2r+1)+5n+2$ входов, на первые $2(2r+1)$ входов подаются номера соответствующих ячеек и их содержимое, на вторые $5n$ входов подаются команды и на последние два входа подаются начальное состояние и номер ячейки в начальном состоянии.Грубая схема состоит из r пар рядов. Первый ряд каждой пары имеет $2r+1$ блоков, имеется соответствие между ячейкой и блоком с тем же номером. На вход каждого блока поступает следующее:

1. содержимое соответствующей ячейки (или выхода соответствующего блока) на текущий момент a_{in};

2. номер ячейки (блока) s;

3. текущее состояние управляющего устройства q_{in};

4. номер ячейки, на которой сейчас стоит головка l_{in};

5. все команды программы, т.е. $5n$ чисел.

Выходом каждого блока первого ряда пары являяется следующее:

1. содержимое соответствующей ячейки a_{out};

2. состояние управляющего устройства q_{out}, если оно работает в этой ячейк,и или число 0, если управляющее устройство не работает в этой ячейке, или число k, если попали в заключительное сосотояние;

3. номер новой ячейки нового положения головки l_{out}, если головка работает в текущей ячейке, иначе число 0 или положение головки, если попали в заключительное состояние;

Введём обозначение $\chi(u = v) = 1$, если $u = v$, иначе равно $\chi(u = v) = 0$. Функция выходов задаётся следующей формулой:

$$\begin{aligned} a_{out} = a_{in}(1-\chi(l_{in}=s))(1-\chi(q_{in}=k))+ \\ +\sum_{(q'aq''bd)_i, i=1,\ldots,n} \chi(l_{in}=s)\chi(q_{in}=q')\chi(a_{in}=a)b+ \\ +a_{in}\chi(q_{in}=k). \end{aligned} \tag{1}$$

Значит

$$a_{out} = \begin{cases} a_{in}, & q_{in} \neq k, l_{in} \neq s, \\ b, & q_{in} \neq k, l_{in} = s, \\ a_{in}, & q_{in} = k. \end{cases}$$

Функция состояния задаётся следующей формулой:

$$\begin{aligned} q_{out} = \sum_{(q'aq''bd)_i, i=1,\ldots,n} \chi(l_{in}=s)\chi(q_{in}=q')\chi(a_{in}=a)q''+ \\ +k\chi(q_{in}=k). \end{aligned} \tag{2}$$

Значит

$$q_{out} = \begin{cases} 0, & q_{in} \neq k, l_{in} \neq s, \\ q'', & q_{in} \neq k, l_{in} = s, \\ k, & q_{in} = k. \end{cases}$$

Функция номера новой ячейки положения головки (сдвиг вправо $d = 1$, сдвиг влево $d = -1$, отсутствие движения $d = 0$) задаётся следующей формулой:

$$l_{out} = \sum_{(q'aq''bd)_i, i=1,\ldots,n} \chi(l_{in} = s)\chi(q_{in} = q')\chi(a_{in} = a)(s + d) + \\ + l_{in}\chi(q_{in} = k). \tag{3}$$

Значит

$$l_{out} = \begin{cases} 0, & q_{in} \neq k, l_{in} \neq s, \\ s + d, & q_{in} \neq k, l_{in} = s, \\ l_{in}, & q_{in} = k. \end{cases}$$

Второй ряд каждой пары кроме последней представлен одним блоком, имеет две группы входов. Первая группа состоит из $2r + 1$ входов, куда постпает q_{out} с каждого блока первого ряда пары, вторая группа имеет $2r + 1$ входов, куда поступает l_{out} с каждого блока первого ряда пары. Пока машина не попала в заключительное состояние на $2r$ входов первой группы будут поступать 0 и только на один вход поступит q'',в блоке выберется максимум, т.е. то состояние, в котором находится сейчас машина. На входах второй группы аналогично произойдёт выбор нового положения головки машины. Состояние и положение головки — два выхода второго ряда пары. Если в попадаем в заключительное состояние, то дальше оно не меняется, аналогично не меняется номер ячейки положения головки. Так как пар рядов больше чем число шагов машины, то попадание в заключительное состояние произойдёт обязательно. Последний ряд последней пары выдаёт содержимое ячейки с номером l_{out}.Это можно представить следующей формулой

$$a_{out} = \sum_{s=1,\ldots,2r+1} \chi(l_{in} = s)a^s_{in} \tag{4}$$

Это и будет ответом задачи в соответствующих значениях параметров.

Переходим к построению схемы из функциональных элементов в классическом базисе. Закодируем алфавит машины следующим образом:

1-ый символ алфавита — $\underbrace{0\ldots001}_{\text{m}}$

2-ой символ алфавита — $\underbrace{0\ldots011}_{\text{m}}$

. . . . . .

m-ый символ алфавита — $\underbrace{1\ldots111}_{\text{m}}$

Аналогично закодируем k состояний:

1-ый символ множества состояний — $\underbrace{0\ldots001}_{\text{k}}$

2-ой символ множества состоянийа — $\underbrace{0\ldots011}_{\text{k}}$

. . . . . .

k-ый символ множества состояний — $\underbrace{1\ldots111}_{\text{k}}$

Закодируем сдвиги:

сдвиг вправо — 01

сдвиг влево — 10

отсутствие движения — 11

Закодируем номера ячеек машины Тьюринга:

1-ая ячейка ленты — $\underbrace{0\ldots001}_{2r+1}$

2-ая ячейка ленты — $\underbrace{0\ldots011}_{2r+1}$

. . . . . .

$(2r+1)$-ая ячейка ленты — $\underbrace{1\ldots111}_{2r+1}$

Головка в начальный момент находится в $(r+1)$-ой ячейке.Введём обозначение для команд программы

$$\{((q'_1,\ldots,q'_k),(a_1,\ldots,a_m),(q''_1,\ldots,q''_k)(b_1,\ldots,b_m),(d_1,d_2))_t, t=1,\ldots\ n\}$$

Схема будет иметь $m(2r+1)$ входов — это коды содержимых ячеек.Из первого входа создаём 0 и 1 и затем из них коды номеров всех ячеек, коды всех команд программы, код номера $(r+1)$ ячейки — положение головки в начальный момент и код номера начального состояния согласно кодировки.

Согласно описанной выше грубой модели на вход блока поступает по содержимому ячейки (или выхода блока) $a_{in} = (a_{in\,1},\ldots,a_{in\,m})$, тогда первое слагаемое в формуле (1) для a_{out} реализуется следующей формулой:

$$p_i^1 = a_{in\,i}\cdot\urcorner(\bigwedge_{j=1}^{2r+1}(l_{in\,j}\sim s_j))\cdot\urcorner(\bigwedge_{j=1}^{k}(q_{in\,j}\sim 1)),$$
$$i=1,\ldots,m.$$

Для реализации достаточно $m(14r+k+11)$ элементов. Второе слагаемое реализуется следующей формулой:

$$p_i^2 = \bigvee_{t=1}^{n} \Big(\bigwedge_{j=1}^{2r+1} (l_{in\,j} \sim s_j) \Big) \cdot \Big(\bigwedge_{j=1}^{k} (q_{in\,j} \sim q'_{jt}) \Big) \cdot \Big(\bigwedge_{j=1}^{m} (a_{in\,j} \sim a_{jt}) \Big) \cdot b_{it},$$
$$i = 1, \ldots, m.$$

Для реализации достаточно $m(14rn + 7kn + 7mn + 10n)$ элементов. Третье слагаемое реализуется формулой:

$$p_i^3 = a_{in\,i} \cdot \Big(\bigwedge_{j=1}^{k} (q_{in\,j} \sim 1) \Big),$$
$$i = 1, \ldots, m.$$

Для реализации достаточно mk элементов. Тогда

$$a_{out} = (p_1^1 \bigvee p_1^2 \bigvee p_1^3, \ldots, p_m^1 \bigvee p_m^2 \bigvee p_m^3).$$

Для реализации достаточно $m(14r + 2k + 14rn + 7kn + 7mn + 19n + 11)$ элементов.

Далее аналогично, согласно описанной выше грубой модели на вход блока поступает по текущему состоянию $q_{in} = (q_{in\,1}, \ldots, q_{in\,k})$, тогда первое слагаемое в формуле (2) для q_{out} реализуется следующей формулой:

$$u_i^1 = \bigvee_{t=1}^{n} \Big(\bigwedge_{j=1}^{2r+1} (l_{in\,j} \sim s_j) \Big) \cdot \Big(\bigwedge_{j=1}^{k} (q_{in\,j} \sim q'_{jt}) \Big) \cdot \Big(\bigwedge_{j=1}^{m} (a_{in\,j} \sim a_{jt}) \Big) \cdot q''_{it},$$
$$i = 1, \ldots, k.$$

Для реализации достаточно k(14rn+7kn+7mn+10n) элементов. Второе слагаемое реализуется формулой:

$$u_i^2 = \bigwedge_{j=1}^{k} (q_{in\,j} \sim 1),$$
$$i = 1, \ldots, k.$$

Для реализации достаточно k^2 элементов.Тогда

$$q_{out} = (u_1^1 \bigvee u_1^2, \ldots, u_k^1 \bigvee u_k^2).$$

Для реализации достаточно $k(14rn + 7kn + 7mn + 10n + k + 1)$ элементов.

Наконец, согласно описанной выше грубой модели на вход блока поступает по текущему положению головки $l_{in} = (l_{in\,1}, \ldots, l_{in\,(2r+1)})$, тогда первое слагаемое в формуле (3) для l_{out} согласно кодировке сдвига кодировки номеров ячеек реализуется последовательностью следующих формул:

$$g_i(s_1,\dots,s_{2r+1}) = s_i \bigvee s_{i+1}, \qquad i = 1,\dots,2r.$$
$$g_{2r+1}(s_1,\dots,s_{2r+1}) = s_{2r+1}.$$
$$h_i(s_1,\dots,s_{2r+1}) = s_{i-1} \bigwedge s_i, \qquad i = 2,\dots,2r+1.$$
$$h_1(s_1,\dots,s_{2r+1}) = s_1.$$
$$f_{it}(d_1,d_2,s_1,\dots,s_{2r+1}) = (\urcorner d_{1\,t}) d_{2\,t} g_i(s_1,\dots,s_{2r+1}) \bigvee$$
$$\bigvee d_{1\,t}(\urcorner d_{2\,t}) h_i(s_1,\dots,s_{2r+1}) \bigvee d_{1\,t} d_{2\,t} s_i,$$
$$v_i^1 = \bigvee_{t=1}^{n} \Big(\bigwedge_{j=1}^{2r+1} (l_{in\,j} \sim s_j) \Big) \cdot \Big(\bigwedge_{j=1}^{k} (q_{in\,j} \sim q'_{jt}) \Big) \cdot$$
$$\cdot \Big(\bigwedge_{j=1}^{m} (a_{in\,j} \sim a_{jt}) \Big) \cdot f_{it}(d_1,d_2,s_1,\dots,s_{2r+1}),$$
$$i = 1,\dots,2r+1.$$

Для реализации этой последовательности формул достаточно $(2r+1)n(18r+7k+7m+24)$ элементов. Второе слагаемое реализуется формулой:

$$v_i^2 = l_{in\,i} \cdot \bigwedge_{j=1}^{k} (q_{in\,j} \sim 1),$$
$$i = 1,\dots,2r+1.$$

Для реализации достаточно $(2r+1)k$ элементов.Тогда

$$l_{out} = (v_1^1 \bigvee v_1^2, \dots, v_{2r+1}^1 \bigvee v_{2r+1}^2).$$

Для реализации достаточно $(2r+1)(18rn+7kn+7mn+20n+k)$ элементов.

На вход второго блока поступают $q_{out}^w = (q_{out\,1}^w, \dots, q_{out\,k}^w)$ с каждого блока первого ряда пары, здесь w — номер блока, $w = 1,\dots,2r+1$,выбирается максимум Q_{out} — текущее состояние машины по следующей формуле:

$$Q_{out\,i} = \bigvee_{w=1}^{2r+1} q_{out\,i}^w, \qquad i = 1,\dots,k.$$

Для реализации достаточно $k(2r+1)$ элементов.

Аналогично, на вход второго блока поступают $l_{out}^w = (l_{out\,1}^w, \dots, l_{out\,k}^w)$ с каждого блока первого ряда пары, здесь w — номер блока, $w = 1,\dots,2r+1$,выбирается максимум L_{out} — текущее положение головки по следующей формуле:

$$L_{out\,i} = \bigvee_{w=1}^{2r+1} l_{out\,i}^w, \qquad i = 1,\dots,2r+1.$$

Для реализации достаточно $(2r+1)^2$ элементов.

Теперь второй ряд последней пары рядов схемы. Формула (4) для a_{out} реализуется следующей формулой:

$$a_{out\,i} = \bigvee_{s=1}^{2r+1} \bigwedge_{j=1}^{2r+1} (l_{in\,j} \sim s_j) a^s_{in\,i}$$
$$i = 1, \ldots, m.$$

Сложность полученной схемы есть полином от r, m, k.

Рассмотрим вторую задачу. Пусть есть алгоритм решения задачи. Пусть алфавит соответствующей машины Тьюринга содержит символы 0 и 1. В начальный момент на ленте присутствуют только 0, 1 и пустой символ. Как было сказано в начале, известна верхняя оценка числа шагов. Пусть при любых значениях параметров задачи на ленте входные символы занимают фиксированное число ячеек. Выбрана кодировка алфавита числами 0 и 1 как в первой задаче. Машина всегда останавливается в заключительном состоянии и содержимое соответствующей ячейки 0 или 1. Этой машине поставим в соответствие схему из функциональных элементов. Входов у схемы будет столько, сколько в начальный момент символов 0 и 1 на ленте. Далее в схеме производится переход к схеме описанной в первой задаче: кодируются 0, 1 и пустой символ согласно выбранной кодировке. Для кодировки выбирается количество ячеек ленты исходя из верхней оценки числа шагов. Далее строится схема, как в первой задаче. После получения ответа — кода 0 или 1, происходит переход от кода к кодируемому символу 0 или 1. Если алгоритм полиномиален, то и схема полиномиальна. Сложность этой схемы не меньше, чем сложность минимальной схемы. Следовательно, если алгоритм полиномиален, то и минимальная схема полиномиальна. Обратно, по любой схеме можно построить машину Тьюринга, причём, если схема полиномиальна, то и машина Тьюринга полиномиальна. Если минимальный алгоритм не полиномиален, то при установленной системы кодировки минимальная схема не может быть полиномиальной. Относительно построения по схеме машины можно посмотреть [1]. По вопросу определения сложности минимальной схемы для известной булевской функции можно посмотреть [2] и скорректировать на частичную булевскую функцию, которая появляется при кодировании.

Список литературы

[1] Носов М.В., “О соответствии сложности СФЭ и числа шагов машины Тьюринга”, *Интеллектуальные системы. Теория и приложения*, **25**:3, 189-190.

[2] Носов М.В., “Об аналитическом представлении функции сложности минимальной схемы в базисе из штриха Шеффера”, *Интеллектуальные системы. Теория и приложения*, **21**:2, 193-196.

Implementation of algorithms by schemes of functional elements
Nosov M.V.

In this paper, a scheme of functional elements for a Turing machine is constructed while maintaining the polynomial condition, if the machine possessed such a property.

Keywords: Turing machine, a scheme of functional elements.

References

[1] Nosov M.V., "On the correspondence between the complexity of the SFE and the number of steps of the Turing machine", *Intelligent systems. Theory and Applications*, **25**:3 (2021), 189–190 (In Russian).

[2] Nosov M.V., "On the analytical representation of the function of the complexity of the minimum scheme in the basis of the Sheffer stroke", *Intelligent systems. Theory and Applications*, **21**:2 (2017), 193–196 (In Russian).

К сведению авторов публикаций в журнале «Интеллектуальные системы. Теория и приложения»

В соответствии с требованиями ВАК РФ к изданиям, входящим в перечень ведущих рецензируемых научных журналов и изданий, в которых могут быть опубликованы основные научные результаты диссертаций на соискание ученой степени доктора и кандидата наук, статьи в журнал «Интеллектуальные системы. Теория и приложения» предоставляются авторами в следующей форме:

1. Статьи, набранные в пакете LaTeX, предоставляются к загрузке через WEB-форму http://intsysmagazine.ru/generator_form .
2. К статье прилагаются файлы, содержащие название статьи на русском и английском языках, аннотацию на русском и английском языках (не более 50 слов), список ключевых слов на русском и английском языках (не более 20 слов), информация об авторах: Ф.И.О. полностью, место работы, должность, ученая степень и/или звание (если имеется), для аспирантов ФИО научного руководителя, контактные телефоны (с кодом города и страны), e-mail, почтовый адрес с индексом города (домашний или служебный).
3. Список литературы оформляется в едином формате, установленном системой Российского индекса научного цитирования. Список на русском языке приводится в конце файла с текстом статьи, в то время как список, переведённый на английский язык, прилагается отдельным файлом.
4. За публикацию статей в журнале «Интеллектуальные системы. Теория и приложения» с авторов (в том числе аспирантов высших учебных заведений) статей, рекомендованных к публикации, плата не взимается. Авторам бесплатно предоставляется номер журнала, в котором вышла статья. Журнал распространяется по подписке, экземпляры журнала рассылаются подписчикам наложенным платежом. Условия подписки публикуются в каталоге НТИ «Роспечать», индекс журнала 64559.
5. Доступ к электронной версии последнего вышедшего номера осуществляется через НЭБ «Российский индекс научного цитирования». Номера, вышедшие ранее, размещаются на сайте

http://intsysmagazine.ru,

и доступ к ним бесплатный. Там же будут размещены полные тексты всех публикуемых статей.

Подписано в печать: 28.03.2025
Дата выхода: 15.04.2025
Тираж: 200 экз.
Цена свободная
Свидетельство о регистрации СМИ: ПИ № ФС77–58444 от 25 июня 2014 г., выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций(Роскомнадзор).